

Homework 1 for Exam This exercise is suitable to achieve a full exam grade. Please note that you need to achieve at least the grade of 2 (50% of work done), the passing grade of 1 is not possible with exercise. The points (and grades) are given based on how well the experiments in point 5 are done.

The exercise work is individual (no groups).

Submit the exercise by 31st of March (31.03.2024) to cstraba@utu.fi (see point 6)

Build a CNN from scratch and try to achieve the highest possible accuracy with the MNIST dataset.

The MNIST data set (<https://www.tensorflow.org/datasets/catalog/mnist>) is used to train handwritten digit recognition models. This is basically a classification model with 10 classes.

Here is a keras / tensorflow example (not CNN) for loading and training a model with MNIST: https://www.tensorflow.org/datasets/keras_example

The parameters for training a model can be found at https://keras.io/api/models/model_training_apis/

Your tasks:

1. Load the MNSIT data set and split into training and test sets.
2. Build a CNN with at least one convolutional layer and 2 or more hidden layers and a dense output layer for 10 classes.
3. Train your CNN on the MNIST training set (extracted in step 1).
4. Evaluate your trained model using the test data set. What is the accuracy of your model?
5. Do the following experiments to improve accuracy:
 - increase the size and depth of the inner layers, what is the effect on the model accuracy?
 - experiment with different activation functions in the inner layers (relu, sigmoid, softmax, etc), see the list of keras activation functions at <https://keras.io/api/layers/activations/>
 - what is the effect of using different activation functions? how about combining the activation function choice with different network size and depth?
 - experiment with various optimizers (<https://keras.io/api/optimizers/>) and learning rate. What is the effect on the resulting model accuracy?
 - with all the above variations, experiment with various batch sizes and epochs for training (see https://keras.io/api/models/model_training_apis/)
 - what happens if we don't use any CNN layer? (considering the model in https://www.tensorflow.org/datasets/keras_example).
6. Write a report on your observations on how the model performed with various experiments in step 5 and submit it along with the colab file.

Name: Rownita Tasneem ID# 2303926

```
from google.colab import drive
drive.mount("/content/drive")
```

Mounted at /content/drive

```
https://www.youtube.com/watch?v=bte8Er0QhDg&list=PLPAK2ZdXZScQGbkYAWzJUIULPXXxw5Gb6&index=1
https://www.kaggle.com/competitions/digit-recognizer/data?select=sample_submission.csv
https://www.tensorflow.org/datasets/keras_example
```

```
from sklearn.metrics import confusion_matrix, precision_score,
recall_score
import seaborn as sns

import os
import cv2
import numpy as np
import matplotlib.pyplot as plt
import tensorflow as tf

import pandas as pd
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense
from tensorflow.keras.optimizers import SGD
from tensorflow.keras.utils import to_categorical

import tensorflow as tf
print(tf.__version__)

2.12.0

#! pip install --upgrade tensorflow
```

<https://machinelearningmastery.com/how-to-develop-a-convolutional-neural-network-from-scratch-for-mnist-handwritten-digit-classification/>

<https://carpentries-incubator.github.io/intro-image-classification-cnn/03-build-cnn.html>

<https://www.analyticsvidhya.com/blog/2021/07/classification-of-handwritten-digits-using-cnn/>

<https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/>

Loading the Dataset

Loading the MNSIT dataset and splitting it into training and test sets.

```
mnist = tf.keras.datasets.mnist ### loading the dataset
(x_train, y_train), (x_test,y_test) = mnist.load_data() ### splitting
the datasets into training and testing

Downloading data from https://storage.googleapis.com/tensorflow/tf-
keras-datasets/mnist.npz
11490434/11490434 [=====] - 0s 0us/step

### normalizing the pixels of the training and testing sets. We will
normalize only the pixels but we will not normalize the labels

#x_train = tf.keras.utils.normalize(x_train, axis = 1)
#x_test = tf.keras.utils.normalize(x_test, axis = 1)

### printing the shapes of the training set

x_train.shape
(60000, 28, 28)

## printing the shapes of the testing set
x_test.shape
(10000, 28, 28)

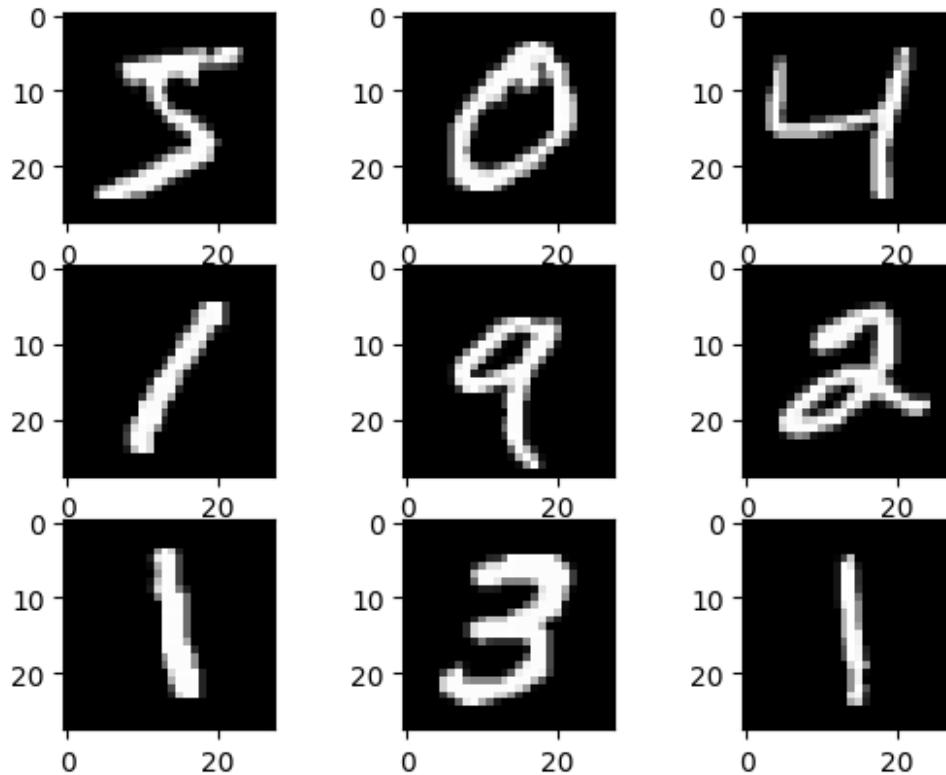
### Summarizing the loaded dataset

print(" Train: X = %s, y=%s" % (x_train.shape, y_train.shape))
print(" Test: X=%s, y = %s" %(x_test.shape, y_test.shape))

Train: X = (60000, 28, 28), y=(60000,)
Test: X=(10000, 28, 28), y = (10000,)
```

In the train dataset, we have 60000 training examples and 10000 testing examples. The size of the images is 28 X 28 pixels.

```
###plotting first few images
for i in range(9):
    ## defining the subplot
    plt.subplot(330+1+i)
    ##plotting raw pixel data
    plt.imshow(x_train[i], cmap = plt.get_cmap("gray"))
    ### Show the figure
plt.show()
```



Reshaping Data:

The dataset has grayscale images of digits. So, the channel dimension is 1. Particularly in image processing with CNNs, it is common practice to explicitly add this channel dimension to the data. This ensures compatibility with the architecture of CNNs.

The architecture of the CNN requires the data to be in the shape of (samples, height, width, channel).

This is a common preprocessing step when working with Convolutional Neural Networks (CNNs) on grayscale images.

In machine learning, particularly in image processing with CNNs, it's crucial that the input data has a consistent shape. For grayscale images, even though they inherently have only one color channel, it's common practice to explicitly add this channel dimension to the data. This ensures compatibility with the architecture of CNNs, which often expect data to be in the form of (samples, height, width, channels).

Here's what each part of the code does:

x_train.shape[0] and x_test.shape[0]: This accesses the first dimension of the x_train and x_test arrays, respectively, which represent the number of samples in each dataset.

reshape((x_train.shape[0], 28, 28, 1)) and reshape((x_test.shape[0], 28, 28, 1)): These commands reshape the datasets so that each image is explicitly represented as having a shape of (28, 28, 1). Here, 28, 28 represents the height and width of the images (assuming they are

squares of 28 pixels by 28 pixels), and 1 adds the single channel dimension for grayscale. The overall shape becomes:

(x_train.shape[0], 28, 28, 1) for the training dataset, and (x_test.shape[0], 28, 28, 1) for the testing dataset.

By reshaping the datasets this way, each image is represented as a 3D array, where the third dimension is the color channel. For grayscale images, there is only one color channel, hence the 1 in the reshape method.

This explicit inclusion of the channel dimension is necessary for compatibility with many TensorFlow/Keras functions, especially those related to CNNs, which expect input images to have both spatial dimensions (height and width) and a channel dimension, even if there's only one channel for grayscale images.

```
### reshape dataset to have a single channel

x_train = x_train.reshape((x_train.shape[0],28,28,1))
x_test = x_test.reshape((x_test.shape[0],28,28,1))

print(f" The size of the train dataset : {x_train.shape}")
print(f" The size of the test dataset: {x_test.shape}")

The size of the train dataset : (60000, 28, 28, 1)
The size of the test dataset: (10000, 28, 28, 1)
```

One hot encoding the dataset

One hot encoding transforms the integer target values into a binary matrix representation.

Integer Target Values --> Binary Class Matrix (One hot encoded Target Values)

Categorical cross-entropy loss functions often require one hot encoded target values for modeling multi-class classification problems.

This conversion is also essential for classification tasks where the model's output layer uses a softmax function, as it allows the model to predict the probability of each class independently.

```
from tensorflow.keras.utils import to_categorical

### One hot encode target values
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

Normalizing the Dataset

Normalizing the pixel values of grayscale images, rescale them to the range [0,1]. This involves first converting the data type from unsigned integers to floats, then dividing the pixel values by the maximum values.

```

### Converting the integers into float
def normalized_pixels(train, test):

    train_norm = train.astype("float32")
    test_norm = test.astype("float32")

    train_norm = train_norm / 255.0
    test_norm = test_norm / 255.0

    return train_norm, test_norm

train_norm, test_norm = normalized_pixels(x_train, x_test)

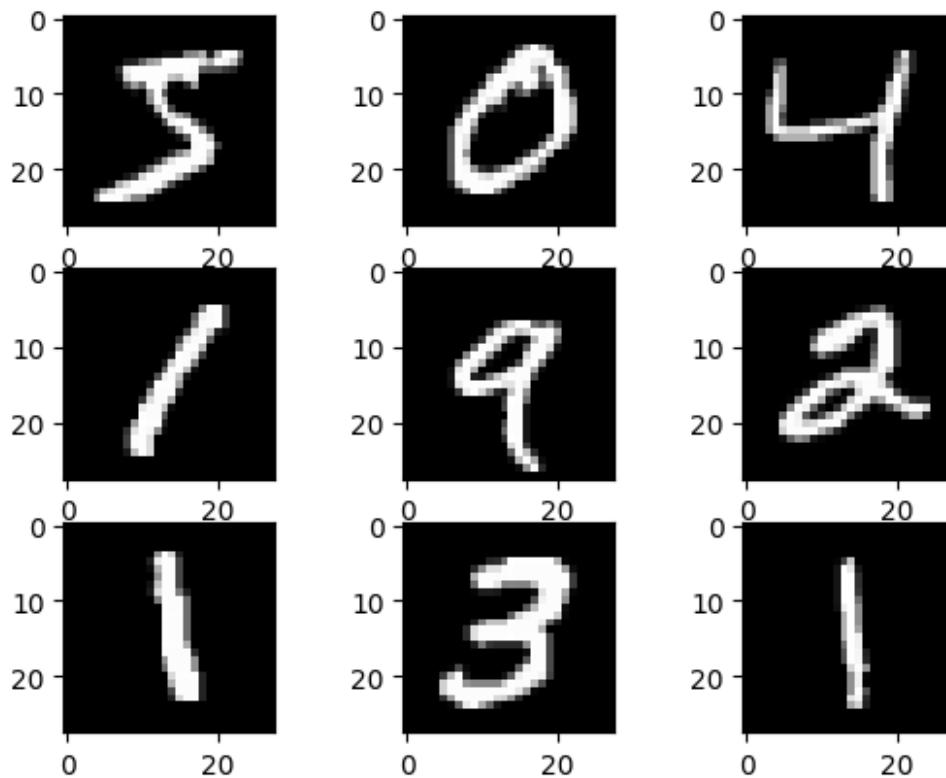
```

Plotting the images

```

###plotting first few images
for i in range(9):
    ## defining the subplot
    plt.subplot(330+1+i)
    ##plotting raw pixel data
    plt.imshow(train_norm[i], cmap = plt.get_cmap("gray"))
    ### Show the figure
plt.show()

```



Building a CNN with at least one convolutional layer and 2 or more hidden layers and a dense output layer for 10 classes.

Define the Model

```
def baseline_model():
    ## create model
    model = Sequential()

    ## Hidden layers
    ## Convolutional Layer
    model.add(Conv2D(32,(3,3), activation = "relu", kernel_initializer =
= "he_uniform", input_shape = (28,28,1)))
    ### Max Pooling Layer
    model.add(MaxPooling2D((2,2)))
    ## Flatten Layer
    model.add(Flatten())
    ## Dense Layer (1st Dense Layer)
    model.add(Dense(100, activation = "relu", kernel_initializer =
= "he_uniform"))

    ## Output layer
    model.add(Dense(10, activation = "softmax")) ## output layer

    ### Compile the model
    opt = SGD(learning_rate = 0.01, momentum = 0.9)
    model.compile(optimizer = opt, loss = "categorical_crossentropy",
metrics = ["accuracy"])
    return model
```

Training the Model

In the above CNN architecture , the hidden layers include everything between the input layer and the output layer. Specifically, for the architecture described, the hidden layers are:

Conv2D Layer: The first layer after the input, which applies 32 convolutional filters of size (3, 3) to the input images. Although it's the first layer and directly processes the input, in the context of the entire network, it acts as a hidden layer by extracting spatial features from the input images through trainable filters.

MaxPooling2D Layer: Follows the Conv2D layer and performs max pooling with a (2, 2) window. This layer reduces the spatial dimensions (height and width) of the feature maps outputted by the Conv2D layer, effectively downsampling the feature maps and making the representation smaller and more manageable.

Flatten Layer: This layer doesn't perform any learning itself but transforms the 2D feature maps from the preceding max pooling layer into a 1D tensor. This is necessary because the following dense layers require 1D input. While primarily a structural layer to enable transitioning from convolutional layers to dense layers, it's considered a part of the hidden structure.

First Dense Layer: A fully connected layer with 100 neurons, receiving input from the Flatten layer. Each neuron in this layer is connected to all neurons in the previous layer. It applies a ReLU activation function to introduce non-linearity, enabling the network to learn complex patterns.

These layers are termed "hidden" because they do not directly interact with the input or output but instead work on representations derived from the input. These representations are transformed and passed through the network until reaching the output layer.

The output layer in this model is the:

Second Dense Layer: A fully connected layer with 10 neurons (one for each class, assuming a 10-class classification problem) that uses a softmax activation function. This layer outputs the probability distribution over the 10 classes for each input image, which is not considered a hidden layer because it directly produces the model's output.

So, in summary, the hidden layers in the above model are the first Conv2D layer, the MaxPooling2D layer, the Flatten layer, and the first Dense layer.

```
### Create the model instance
model = baseline_model()

## Train the model

history = model.fit(train_norm,y_train, epochs = 10, batch_size = 32,
validation_split = 0.1, verbose = 2)

Epoch 1/10
1688/1688 - 36s - loss: 0.1934 - accuracy: 0.9406 - val_loss: 0.0734 -
val_accuracy: 0.9787 - 36s/epoch - 21ms/step
Epoch 2/10
1688/1688 - 35s - loss: 0.0653 - accuracy: 0.9796 - val_loss: 0.0595 -
val_accuracy: 0.9830 - 35s/epoch - 21ms/step
Epoch 3/10
1688/1688 - 34s - loss: 0.0415 - accuracy: 0.9874 - val_loss: 0.0482 -
val_accuracy: 0.9877 - 34s/epoch - 20ms/step
Epoch 4/10
1688/1688 - 38s - loss: 0.0288 - accuracy: 0.9913 - val_loss: 0.0452 -
val_accuracy: 0.9885 - 38s/epoch - 22ms/step
Epoch 5/10
1688/1688 - 36s - loss: 0.0202 - accuracy: 0.9936 - val_loss: 0.0572 -
val_accuracy: 0.9863 - 36s/epoch - 21ms/step
Epoch 6/10
1688/1688 - 35s - loss: 0.0134 - accuracy: 0.9963 - val_loss: 0.0500 -
val_accuracy: 0.9872 - 35s/epoch - 21ms/step
Epoch 7/10
1688/1688 - 35s - loss: 0.0102 - accuracy: 0.9974 - val_loss: 0.0433 -
val_accuracy: 0.9897 - 35s/epoch - 21ms/step
Epoch 8/10
1688/1688 - 34s - loss: 0.0067 - accuracy: 0.9984 - val_loss: 0.0513 -
val_accuracy: 0.9887 - 34s/epoch - 20ms/step
Epoch 9/10
```

```
1688/1688 - 36s - loss: 0.0043 - accuracy: 0.9992 - val_loss: 0.0489 -  
val_accuracy: 0.9883 - 36s/epoch - 21ms/step  
Epoch 10/10  
1688/1688 - 34s - loss: 0.0032 - accuracy: 0.9995 - val_loss: 0.0458 -  
val_accuracy: 0.9893 - 34s/epoch - 20ms/step
```

Evaluating the Model

```
test_loss, test_accuracy = model.evaluate(test_norm,y_test, verbose = 2)  
  
313/313 - 3s - loss: 0.0433 - accuracy: 0.9884 - 3s/epoch - 10ms/step  
  
print(f"Test Accuracy: {test_accuracy: .4f}")  
  
Test Accuracy: 0.9875
```

The test accuracy is 0.9875. This means that the model is able to correctly classify about 98% of the handwritten digits in the test set that it has not seen during the training phase.

```
from sklearn.metrics import confusion_matrix, precision_score,  
recall_score  
  
### Generate Probabilities for each class  
probabilities = model.predict(test_norm, verbose = 0)  
  
## converting probabilities to class predictions . Converting  
probabilities and y_test from one-hot encoded to class integers  
y_pred = np.argmax(probabilities, axis = 1)  
y_true = np.argmax(y_test, axis = 1)
```

Calculate and Print Confusion Matrix, Precision and Recall

```
### Confusion Matrix  
conf_matrix = confusion_matrix(y_true, y_pred)  
print("Confusion Matrix:\n", conf_matrix)
```

```
Confusion Matrix:  
[[ 974    0    1    0    0    0    2    2    1    0]  
 [  0 1129    0    1    0    0    2    1    2    0]  
 [  1    0 1019    0    1    0    2    6    3    0]  
 [  0    0    3 996    0    5    0    0    4    2]  
 [  0    0    0    0 976    0    1    0    1    4]  
 [  1    0    0    4    0 883    3    0    1    0]  
 [  5    2    0    1    2    2 946    0    0    0]  
 [  1    0    6    1    0    0    0 1017    3    0]  
 [  3    0    2    1    2    2    0    2 960    2]  
 [  0    0    0    3   11    3    0    5    3 984]]
```

`confusion_matrix(y_true, y_pred)`: Computes the confusion matrix from `y_true` and `y_pred`.

`plt.figure(figsize=(10, 7))`: Specifies the size of the figure that will display the confusion matrix (optional, but useful for larger matrices).

`sns.heatmap()`: Creates a heatmap to visualize the confusion matrix.

`cm`: The confusion matrix to be plotted.

`annot=True`: Annotates each cell in the heatmap with the numeric value.

`fmt='g'`: Ensures that the format is compact and doesn't use scientific notation.

`cmap='Blues'`: Uses a blue color map for the heatmap. `cbar=False`: Hides the color bar that would otherwise show the scale of the colors.

`plt.xlabel()`, `plt.ylabel()`, `plt.title()`: Set the labels for the x-axis, y-axis, and the title of the plot, respectively.

```
import seaborn as sns
cm = confusion_matrix(y_true, y_pred)
### plotting using seaborn
plt.figure(figsize = (10,7))
sns.heatmap(cm, annot = True, fmt = "g", cmap = "Blues", cbar = False)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```

Confusion Matrix										
	0	1	2	3	4	5	6	7	8	9
0	974	0	1	0	0	0	2	2	1	0
1	0	1129	0	1	0	0	2	1	2	0
2	1	0	1019	0	1	0	2	6	3	0
3	0	0	3	996	0	5	0	0	4	2
4	0	0	0	0	976	0	1	0	1	4
5	1	0	0	4	0	883	3	0	1	0
6	5	2	0	1	2	2	946	0	0	0
7	1	0	6	1	0	0	0	1017	3	0
8	3	0	2	1	2	2	0	2	960	2
9	0	0	0	3	11	3	0	5	3	984
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

The matrix is a 10 X 10 grid, as there are 10 classes representing the digits 0 through 9.

The number of true positive classifications are quite high than the number of misclassifications.

It appears that the classifier performs well for all digits, with most of the predictions concentrated along the diagonal, which indicates correct classifications.

The number of true positive classifications is 9884. In addition, the number of misclassifications is 116.

```
## Precision
precision = precision_score(y_true, y_pred, average = "weighted",
zero_division = 0) ## weighted account for label imbalance
print("Precision:", precision)

Precision: 0.9884198914864817

## Recall
recall = recall_score(y_true, y_pred, average = "weighted") ## weighted account for label imbalance
print("Recall:", recall)

Recall: 0.9884
```

The precision and recall score are quite high. Both of these scores are 0.988. This indicates the model performs very well in identifying the digits from the MNIST dataset accurately.

Precision measures the ratio of correctly predicted positive observations to the total predicted positives. A precision of 0.988 means that when the model predicts a certain digit, it is correct 98.8% of the time.

Recall, on the other hand, measures the ratio of correctly predicted positive observations in the actual class. A recall of 0.988 in this context means that the model successfully identifies 98.8% of all actual digits correctly.

Having both high precision and recall suggests that the model is both highly accurate and covers the majority of the relevant cases without many false positives or false negatives.

Increasing the Size and Depth of the Inner Layers and Effects on Model's Accuracy

Increasing Size and Depth of Inner Layers:

i. **Increase Depth:** If we **add more convolutional layers or dense layers** to make the model deeper, each additional layer can extract more complex features from the dataset.

ii. **Increase Size:** If we increase the **number of neurons in the dense layers or the number of filters** in the convolutional layers, our model will learn complex representations of the dataset.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense
from tensorflow.keras.optimizers import SGD

def modified_model():
    model = Sequential()
    ### Increase the filters numbers in convolutional layers
    model.add(Conv2D(64,(3,3), activation = "relu", kernel_initializer =
"he_uniform", input_shape = (28,28,1)))
    model.add(MaxPooling2D((2,2)))
    ## adding another convolutional layer
    model.add(Conv2D(128,(3,3), activation = "relu",kernel_initializer =
"he_uniform")) ## Additonal Layers
    model.add(MaxPooling2D((2,2)))

    model.add(Flatten())

    ### Increasing the number of neurons in the dense layers
    model.add(Dense(200, activation = "relu", kernel_initializer =
"he_uniform"))
    model.add(Dense(100, activation = "relu", kernel_initializer =
"he_uniform"))

    ### Output layer
    model.add(Dense(10,activation = "softmax"))
```

```

## Compile the model
opt = SGD(learning_rate = 0.01, momentum = 0.9)
model.compile(optimizer = opt, loss = "categorical_crossentropy",
metrics = ["accuracy"])
return model

improved_model = modified_model()
history = improved_model.fit(train_norm, y_train, epochs = 10,
batch_size = 32, validation_split = 0.1, verbose = 2)

Epoch 1/10
1688/1688 - 142s - loss: 0.1252 - accuracy: 0.9612 - val_loss: 0.0460
- val_accuracy: 0.9873 - 142s/epoch - 84ms/step
Epoch 2/10
1688/1688 - 140s - loss: 0.0416 - accuracy: 0.9871 - val_loss: 0.0497
- val_accuracy: 0.9863 - 140s/epoch - 83ms/step
Epoch 3/10
1688/1688 - 140s - loss: 0.0253 - accuracy: 0.9919 - val_loss: 0.0476
- val_accuracy: 0.9875 - 140s/epoch - 83ms/step
Epoch 4/10
1688/1688 - 137s - loss: 0.0182 - accuracy: 0.9941 - val_loss: 0.0523
- val_accuracy: 0.9878 - 137s/epoch - 81ms/step
Epoch 5/10
1688/1688 - 139s - loss: 0.0128 - accuracy: 0.9958 - val_loss: 0.0387
- val_accuracy: 0.9905 - 139s/epoch - 83ms/step
Epoch 6/10
1688/1688 - 137s - loss: 0.0083 - accuracy: 0.9974 - val_loss: 0.0367
- val_accuracy: 0.9905 - 137s/epoch - 81ms/step
Epoch 7/10
1688/1688 - 133s - loss: 0.0060 - accuracy: 0.9979 - val_loss: 0.0416
- val_accuracy: 0.9905 - 133s/epoch - 79ms/step
Epoch 8/10
1688/1688 - 134s - loss: 0.0031 - accuracy: 0.9990 - val_loss: 0.0515
- val_accuracy: 0.9913 - 134s/epoch - 79ms/step
Epoch 9/10
1688/1688 - 136s - loss: 0.0025 - accuracy: 0.9992 - val_loss: 0.0434
- val_accuracy: 0.9915 - 136s/epoch - 81ms/step
Epoch 10/10
1688/1688 - 140s - loss: 0.0019 - accuracy: 0.9994 - val_loss: 0.0486
- val_accuracy: 0.9905 - 140s/epoch - 83ms/step

test_loss, test_acc = improved_model.evaluate(test_norm, y_test,
verbose = 2)
print(f"Test Accuracy: {test_acc:.4f}")

313/313 - 7s - loss: 0.0342 - accuracy: 0.9920 - 7s/epoch - 22ms/step
Test Accuracy: 0.9920

```

The test accuracy for the improved model is 0.99. So, the accuracy increases as we add more convolutional layers and more neurons to the network. Therefore, the test accuracy increases as the width and depth of the network grows.

```
from sklearn.metrics import confusion_matrix, precision_score,
recall_score

### Generate probabilities for each class
probabilities_1 = improved_model.predict(test_norm, verbose = 0)

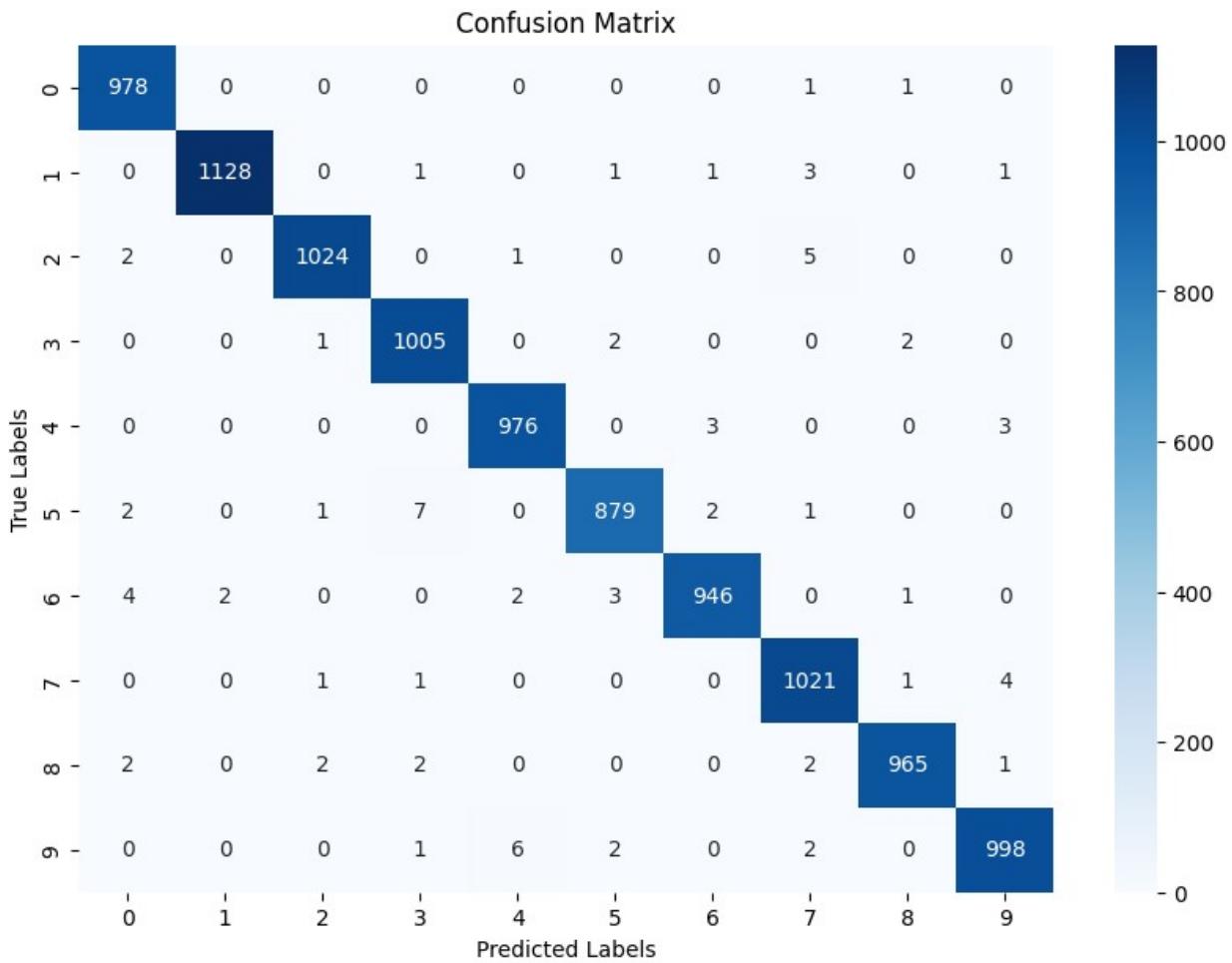
## Converting probabilities to class predictions. Converting
## probabilities and y_test from one-hot encoded to class integers
y_pred_1 = np.argmax(probabilities_1, axis = 1)
y_true_1 = np.argmax(y_test, axis = 1)

conf_matrix_1 = confusion_matrix(y_true, y_pred_1)
print("Confusion Matrix:\n", conf_matrix_1)

Confusion Matrix:
[[ 978   0   0   0   0   0   0   1   1   0]
 [  0 1128   0   1   0   1   1   3   0   1]
 [  2   0 1024   0   1   0   0   5   0   0]
 [  0   0   1 1005   0   2   0   0   2   0]
 [  0   0   0   0  976   0   3   0   0   3]
 [  2   0   1   7   0  879   2   1   0   0]
 [  4   2   0   0   2   3  946   0   1   0]
 [  0   0   1   1   0   0   0 1021   1   4]
 [  2   0   2   2   0   0   0   2  965   1]
 [  0   0   0   1   6   2   0   2   0  998]]
```

```
import seaborn as sns
cm_1 = confusion_matrix(y_true_1,y_pred_1)

###plotting using seaborn
plt.figure(figsize = (10,7))
sns.heatmap(cm_1, annot = True, fmt = "g", cmap = "Blues", cbar= True)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title("Confusion Matrix")
plt.show()
```



In the above confusion matrix, total number of true positive classifications is 9920 and the total number of misclassifications is 80. Moreover, the aggregate sum of the main diagonal is 9920. So, the model achieves a high level of model accuracy.

Therefore, the true positive rate increases as the number of layers and size of the deep neural network increases.

Hence, a more complex model characterized by a greater depth and breadth, has the potential to enhance the model's predictive power, possibly due to its ability to capture more intricate patterns in the data.

Generally, increasing a model's complexity does not invariably lead to improved performance and can result in overfitting, where model performs well on the training data and performs poorly on testing dataset.

```
precision_1 = precision_score(y_true_1,y_pred_1, average = "weighted",
zero_division = 0)
recall_1 = recall_score(y_true_1, y_pred_1, average = "weighted")
print("Recall:",recall_1)
print("Precision:",precision_1)
```

```
Recall: 0.992
Precision: 0.9920140833483724
```

The recall score is 0.99 and the precision score is 0.99. Therefore, the model correctly classifies positive classifications 99% of the times from the total positive cases. It can also classify actual positive classifications 99% of the times from total actual positive cases.

Therefore, the precision and the recall score increases if we add more layers and increase the depth and width of the network.

Experimenting with Different Activation Functions in the Inner Layers

```
from tensorflow.keras.activations import gelu, elu, relu, sigmoid, tanh,
softplus, exponential, hard_sigmoid, linear, softsign

def baseline_model_exp(conv_activation = "relu", dense_activation =
"relu"):
    ## create model
    model = Sequential()

    ## Hidden layers
    ## Convolutional Layer
    model.add(Conv2D(32,(3,3), activation = conv_activation,
kernel_initializer = "he_uniform", input_shape = (28,28,1)))
    ### Max Pooling Layer
    model.add(MaxPooling2D((2,2)))
    ## Flatten Layer
    model.add(Flatten())
    ## Dense Layer (1st Dense Layer)
    model.add(Dense(100, activation = dense_activation,
kernel_initializer = "he_uniform"))

    ## Output layer
    model.add(Dense(10, activation = "softmax")) ## output layer
activation function is always fixed for softmax activation function
for multiclass classification problem

    ### Compile the model
    opt = SGD(learning_rate = 0.01, momentum = 0.9)
    model.compile(optimizer = opt, loss = "categorical_crossentropy",
metrics = ["accuracy"])
    return model

act_functions =
["gelu", "elu", "relu", "sigmoid", "tanh", "softplus", "exponential", "softma
x", "softsign", "hard_sigmoid", "linear"]

for act in act_functions:
    model = baseline_model_exp(conv_activation = act, dense_activation =
act)
```

```

print(f"Training model with {act} activation...")
history = model.fit(train_norm,y_train,epochs = 10, batch_size =
32 ,validation_split = 0.1, verbose = 2)

test_loss,test_accuracy = model.evaluate(test_norm,y_test, verbose =
0)
print(f"Test accuracy with {act} activation:{test_accuracy:.4f}\n")

```

Training model with gelu activation...

Epoch 1/10
1688/1688 - 16s - loss: 0.1566 - accuracy: 0.9519 - val_loss: 0.0625 -
val_accuracy: 0.9825 - 16s/epoch - 9ms/step

Epoch 2/10
1688/1688 - 15s - loss: 0.0514 - accuracy: 0.9846 - val_loss: 0.0467 -
val_accuracy: 0.9867 - 15s/epoch - 9ms/step

Epoch 3/10
1688/1688 - 15s - loss: 0.0334 - accuracy: 0.9898 - val_loss: 0.0467 -
val_accuracy: 0.9867 - 15s/epoch - 9ms/step

Epoch 4/10
1688/1688 - 15s - loss: 0.0213 - accuracy: 0.9937 - val_loss: 0.0471 -
val_accuracy: 0.9873 - 15s/epoch - 9ms/step

Epoch 5/10
1688/1688 - 15s - loss: 0.0147 - accuracy: 0.9956 - val_loss: 0.0415 -
val_accuracy: 0.9892 - 15s/epoch - 9ms/step

Epoch 6/10
1688/1688 - 14s - loss: 0.0099 - accuracy: 0.9975 - val_loss: 0.0467 -
val_accuracy: 0.9887 - 14s/epoch - 8ms/step

Epoch 7/10
1688/1688 - 15s - loss: 0.0061 - accuracy: 0.9986 - val_loss: 0.0455 -
val_accuracy: 0.9895 - 15s/epoch - 9ms/step

Epoch 8/10
1688/1688 - 15s - loss: 0.0033 - accuracy: 0.9996 - val_loss: 0.0464 -
val_accuracy: 0.9903 - 15s/epoch - 9ms/step

Epoch 9/10
1688/1688 - 15s - loss: 0.0026 - accuracy: 0.9996 - val_loss: 0.0482 -
val_accuracy: 0.9898 - 15s/epoch - 9ms/step

Epoch 10/10
1688/1688 - 15s - loss: 0.0018 - accuracy: 0.9997 - val_loss: 0.0513 -
val_accuracy: 0.9907 - 15s/epoch - 9ms/step

Test accuracy with gelu activation:0.9894

Training model with elu activation...

Epoch 1/10
1688/1688 - 12s - loss: 0.1732 - accuracy: 0.9481 - val_loss: 0.0627 -
val_accuracy: 0.9827 - 12s/epoch - 7ms/step

Epoch 2/10
1688/1688 - 11s - loss: 0.0579 - accuracy: 0.9830 - val_loss: 0.0560 -
val_accuracy: 0.9823 - 11s/epoch - 7ms/step

```
Epoch 3/10
1688/1688 - 11s - loss: 0.0385 - accuracy: 0.9881 - val_loss: 0.0493 -
val_accuracy: 0.9858 - 11s/epoch - 7ms/step
Epoch 4/10
1688/1688 - 12s - loss: 0.0241 - accuracy: 0.9928 - val_loss: 0.0507 -
val_accuracy: 0.9865 - 12s/epoch - 7ms/step
Epoch 5/10
1688/1688 - 13s - loss: 0.0161 - accuracy: 0.9951 - val_loss: 0.0524 -
val_accuracy: 0.9867 - 13s/epoch - 8ms/step
Epoch 6/10
1688/1688 - 11s - loss: 0.0107 - accuracy: 0.9970 - val_loss: 0.0547 -
val_accuracy: 0.9870 - 11s/epoch - 7ms/step
Epoch 7/10
1688/1688 - 11s - loss: 0.0082 - accuracy: 0.9977 - val_loss: 0.0542 -
val_accuracy: 0.9872 - 11s/epoch - 7ms/step
Epoch 8/10
1688/1688 - 11s - loss: 0.0049 - accuracy: 0.9989 - val_loss: 0.0522 -
val_accuracy: 0.9882 - 11s/epoch - 6ms/step
Epoch 9/10
1688/1688 - 11s - loss: 0.0027 - accuracy: 0.9996 - val_loss: 0.0515 -
val_accuracy: 0.9893 - 11s/epoch - 6ms/step
Epoch 10/10
1688/1688 - 11s - loss: 0.0017 - accuracy: 0.9998 - val_loss: 0.0528 -
val_accuracy: 0.9897 - 11s/epoch - 7ms/step
Test accuracy with elu activation: 0.9882
```

```
Training model with relu activation...
Epoch 1/10
1688/1688 - 11s - loss: 0.1728 - accuracy: 0.9476 - val_loss: 0.0678 -
val_accuracy: 0.9797 - 11s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 10s - loss: 0.0604 - accuracy: 0.9818 - val_loss: 0.0589 -
val_accuracy: 0.9837 - 10s/epoch - 6ms/step
Epoch 3/10
1688/1688 - 10s - loss: 0.0402 - accuracy: 0.9878 - val_loss: 0.0637 -
val_accuracy: 0.9812 - 10s/epoch - 6ms/step
Epoch 4/10
1688/1688 - 10s - loss: 0.0284 - accuracy: 0.9912 - val_loss: 0.0479 -
val_accuracy: 0.9878 - 10s/epoch - 6ms/step
Epoch 5/10
1688/1688 - 11s - loss: 0.0204 - accuracy: 0.9941 - val_loss: 0.0403 -
val_accuracy: 0.9900 - 11s/epoch - 6ms/step
Epoch 6/10
1688/1688 - 10s - loss: 0.0139 - accuracy: 0.9959 - val_loss: 0.0436 -
val_accuracy: 0.9895 - 10s/epoch - 6ms/step
Epoch 7/10
1688/1688 - 10s - loss: 0.0085 - accuracy: 0.9979 - val_loss: 0.0487 -
val_accuracy: 0.9883 - 10s/epoch - 6ms/step
Epoch 8/10
```

```
1688/1688 - 11s - loss: 0.0073 - accuracy: 0.9983 - val_loss: 0.0617 -  
val_accuracy: 0.9863 - 11s/epoch - 6ms/step  
Epoch 9/10  
1688/1688 - 11s - loss: 0.0053 - accuracy: 0.9986 - val_loss: 0.0462 -  
val_accuracy: 0.9885 - 11s/epoch - 6ms/step  
Epoch 10/10  
1688/1688 - 10s - loss: 0.0036 - accuracy: 0.9992 - val_loss: 0.0572 -  
val_accuracy: 0.9868 - 10s/epoch - 6ms/step  
Test accuracy with relu activation:0.9843
```

Training model with sigmoid activation...

```
Epoch 1/10  
1688/1688 - 11s - loss: 0.6202 - accuracy: 0.8126 - val_loss: 0.2607 -  
val_accuracy: 0.9180 - 11s/epoch - 7ms/step  
Epoch 2/10  
1688/1688 - 11s - loss: 0.2838 - accuracy: 0.9138 - val_loss: 0.1981 -  
val_accuracy: 0.9402 - 11s/epoch - 6ms/step  
Epoch 3/10  
1688/1688 - 11s - loss: 0.2199 - accuracy: 0.9335 - val_loss: 0.1591 -  
val_accuracy: 0.9538 - 11s/epoch - 6ms/step  
Epoch 4/10  
1688/1688 - 11s - loss: 0.1832 - accuracy: 0.9439 - val_loss: 0.1242 -  
val_accuracy: 0.9652 - 11s/epoch - 6ms/step  
Epoch 5/10  
1688/1688 - 11s - loss: 0.1567 - accuracy: 0.9544 - val_loss: 0.1232 -  
val_accuracy: 0.9650 - 11s/epoch - 6ms/step  
Epoch 6/10  
1688/1688 - 11s - loss: 0.1368 - accuracy: 0.9593 - val_loss: 0.1055 -  
val_accuracy: 0.9705 - 11s/epoch - 7ms/step  
Epoch 7/10  
1688/1688 - 11s - loss: 0.1207 - accuracy: 0.9645 - val_loss: 0.1074 -  
val_accuracy: 0.9697 - 11s/epoch - 7ms/step  
Epoch 8/10  
1688/1688 - 11s - loss: 0.1082 - accuracy: 0.9678 - val_loss: 0.0920 -  
val_accuracy: 0.9748 - 11s/epoch - 7ms/step  
Epoch 9/10  
1688/1688 - 11s - loss: 0.0985 - accuracy: 0.9709 - val_loss: 0.0906 -  
val_accuracy: 0.9740 - 11s/epoch - 7ms/step  
Epoch 10/10  
1688/1688 - 11s - loss: 0.0877 - accuracy: 0.9745 - val_loss: 0.0750 -  
val_accuracy: 0.9775 - 11s/epoch - 7ms/step  
Test accuracy with sigmoid activation:0.9725
```

Training model with tanh activation...

```
Epoch 1/10  
1688/1688 - 11s - loss: 0.1948 - accuracy: 0.9422 - val_loss: 0.0933 -  
val_accuracy: 0.9742 - 11s/epoch - 7ms/step  
Epoch 2/10  
1688/1688 - 11s - loss: 0.0740 - accuracy: 0.9789 - val_loss: 0.0592 -
```

```
val_accuracy: 0.9837 - 11s/epoch - 6ms/step
Epoch 3/10
1688/1688 - 11s - loss: 0.0450 - accuracy: 0.9876 - val_loss: 0.0563 -
val_accuracy: 0.9843 - 11s/epoch - 6ms/step
Epoch 4/10
1688/1688 - 11s - loss: 0.0306 - accuracy: 0.9919 - val_loss: 0.0499 -
val_accuracy: 0.9857 - 11s/epoch - 6ms/step
Epoch 5/10
1688/1688 - 11s - loss: 0.0205 - accuracy: 0.9948 - val_loss: 0.0432 -
val_accuracy: 0.9888 - 11s/epoch - 7ms/step
Epoch 6/10
1688/1688 - 11s - loss: 0.0143 - accuracy: 0.9972 - val_loss: 0.0435 -
val_accuracy: 0.9877 - 11s/epoch - 7ms/step
Epoch 7/10
1688/1688 - 11s - loss: 0.0097 - accuracy: 0.9985 - val_loss: 0.0439 -
val_accuracy: 0.9867 - 11s/epoch - 6ms/step
Epoch 8/10
1688/1688 - 10s - loss: 0.0072 - accuracy: 0.9990 - val_loss: 0.0444 -
val_accuracy: 0.9880 - 10s/epoch - 6ms/step
Epoch 9/10
1688/1688 - 11s - loss: 0.0053 - accuracy: 0.9995 - val_loss: 0.0446 -
val_accuracy: 0.9883 - 11s/epoch - 6ms/step
Epoch 10/10
1688/1688 - 11s - loss: 0.0037 - accuracy: 0.9999 - val_loss: 0.0433 -
val_accuracy: 0.9892 - 11s/epoch - 6ms/step
Test accuracy with tanh activation: 0.9868
```

```
Training model with softplus activation...
Epoch 1/10
1688/1688 - 13s - loss: 0.6481 - accuracy: 0.7948 - val_loss: 0.2640 -
val_accuracy: 0.9207 - 13s/epoch - 8ms/step
Epoch 2/10
1688/1688 - 12s - loss: 0.2219 - accuracy: 0.9329 - val_loss: 0.1186 -
val_accuracy: 0.9645 - 12s/epoch - 7ms/step
Epoch 3/10
1688/1688 - 12s - loss: 0.1289 - accuracy: 0.9603 - val_loss: 0.0912 -
val_accuracy: 0.9760 - 12s/epoch - 7ms/step
Epoch 4/10
1688/1688 - 12s - loss: 0.1051 - accuracy: 0.9683 - val_loss: 0.0830 -
val_accuracy: 0.9760 - 12s/epoch - 7ms/step
Epoch 5/10
1688/1688 - 11s - loss: 0.0862 - accuracy: 0.9739 - val_loss: 0.0710 -
val_accuracy: 0.9800 - 11s/epoch - 7ms/step
Epoch 6/10
1688/1688 - 11s - loss: 0.0762 - accuracy: 0.9767 - val_loss: 0.0715 -
val_accuracy: 0.9795 - 11s/epoch - 7ms/step
Epoch 7/10
1688/1688 - 12s - loss: 0.0655 - accuracy: 0.9797 - val_loss: 0.0642 -
val_accuracy: 0.9817 - 12s/epoch - 7ms/step
```

```
Epoch 8/10
1688/1688 - 12s - loss: 0.0581 - accuracy: 0.9819 - val_loss: 0.0911 -
val_accuracy: 0.9770 - 12s/epoch - 7ms/step
Epoch 9/10
1688/1688 - 12s - loss: 0.0532 - accuracy: 0.9836 - val_loss: 0.0613 -
val_accuracy: 0.9835 - 12s/epoch - 7ms/step
Epoch 10/10
1688/1688 - 12s - loss: 0.0501 - accuracy: 0.9840 - val_loss: 0.0557 -
val_accuracy: 0.9858 - 12s/epoch - 7ms/step
Test accuracy with softplus activation:0.9824

Training model with exponential activation...
Epoch 1/10
1688/1688 - 12s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 12s/epoch - 7ms/step
Epoch 2/10
1688/1688 - 10s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 10s/epoch - 6ms/step
Epoch 3/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 4/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 5/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 6/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 7/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 8/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 7ms/step
Epoch 9/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 10/10
1688/1688 - 10s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 10s/epoch - 6ms/step
Test accuracy with exponential activation:0.0980

Training model with softmax activation...
Epoch 1/10
1688/1688 - 14s - loss: 2.3016 - accuracy: 0.1123 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 14s/epoch - 8ms/step
Epoch 2/10
```

```
1688/1688 - 13s - loss: 2.2998 - accuracy: 0.1149 - val_loss: 2.2982 -  
val_accuracy: 0.1050 - 13s/epoch - 8ms/step  
Epoch 3/10  
1688/1688 - 13s - loss: 2.2668 - accuracy: 0.1503 - val_loss: 2.0978 -  
val_accuracy: 0.2282 - 13s/epoch - 8ms/step  
Epoch 4/10  
1688/1688 - 13s - loss: 1.7063 - accuracy: 0.3311 - val_loss: 1.5060 -  
val_accuracy: 0.3558 - 13s/epoch - 8ms/step  
Epoch 5/10  
1688/1688 - 13s - loss: 1.4839 - accuracy: 0.3648 - val_loss: 1.4087 -  
val_accuracy: 0.3795 - 13s/epoch - 8ms/step  
Epoch 6/10  
1688/1688 - 13s - loss: 1.3992 - accuracy: 0.3944 - val_loss: 1.3345 -  
val_accuracy: 0.4353 - 13s/epoch - 8ms/step  
Epoch 7/10  
1688/1688 - 14s - loss: 1.3412 - accuracy: 0.4222 - val_loss: 1.3026 -  
val_accuracy: 0.4462 - 14s/epoch - 8ms/step  
Epoch 8/10  
1688/1688 - 13s - loss: 1.3061 - accuracy: 0.4330 - val_loss: 1.2596 -  
val_accuracy: 0.4625 - 13s/epoch - 8ms/step  
Epoch 9/10  
1688/1688 - 13s - loss: 1.2806 - accuracy: 0.4504 - val_loss: 1.2282 -  
val_accuracy: 0.4737 - 13s/epoch - 8ms/step  
Epoch 10/10  
1688/1688 - 13s - loss: 1.2498 - accuracy: 0.4674 - val_loss: 1.1960 -  
val_accuracy: 0.4900 - 13s/epoch - 8ms/step  
Test accuracy with softmax activation: 0.4838
```

```
Training model with softsign activation...  
Epoch 1/10  
1688/1688 - 12s - loss: 0.2608 - accuracy: 0.9244 - val_loss: 0.1195 -  
val_accuracy: 0.9680 - 12s/epoch - 7ms/step  
Epoch 2/10  
1688/1688 - 10s - loss: 0.1151 - accuracy: 0.9665 - val_loss: 0.0816 -  
val_accuracy: 0.9778 - 10s/epoch - 6ms/step  
Epoch 3/10  
1688/1688 - 11s - loss: 0.0778 - accuracy: 0.9778 - val_loss: 0.0744 -  
val_accuracy: 0.9783 - 11s/epoch - 6ms/step  
Epoch 4/10  
1688/1688 - 11s - loss: 0.0568 - accuracy: 0.9846 - val_loss: 0.0553 -  
val_accuracy: 0.9848 - 11s/epoch - 7ms/step  
Epoch 5/10  
1688/1688 - 11s - loss: 0.0440 - accuracy: 0.9881 - val_loss: 0.0522 -  
val_accuracy: 0.9875 - 11s/epoch - 7ms/step  
Epoch 6/10  
1688/1688 - 11s - loss: 0.0354 - accuracy: 0.9910 - val_loss: 0.0507 -  
val_accuracy: 0.9862 - 11s/epoch - 7ms/step  
Epoch 7/10  
1688/1688 - 11s - loss: 0.0287 - accuracy: 0.9930 - val_loss: 0.0500 -
```

```
val_accuracy: 0.9867 - 11s/epoch - 7ms/step
Epoch 8/10
1688/1688 - 11s - loss: 0.0232 - accuracy: 0.9948 - val_loss: 0.0446 -
val_accuracy: 0.9878 - 11s/epoch - 7ms/step
Epoch 9/10
1688/1688 - 11s - loss: 0.0190 - accuracy: 0.9958 - val_loss: 0.0452 -
val_accuracy: 0.9882 - 11s/epoch - 7ms/step
Epoch 10/10
1688/1688 - 11s - loss: 0.0158 - accuracy: 0.9970 - val_loss: 0.0442 -
val_accuracy: 0.9883 - 11s/epoch - 7ms/step
Test accuracy with softsign activation:0.9855

Training model with hard_sigmoid activation...
Epoch 1/10
1688/1688 - 16s - loss: 0.7087 - accuracy: 0.7854 - val_loss: 0.3235 -
val_accuracy: 0.9010 - 16s/epoch - 10ms/step
Epoch 2/10
1688/1688 - 15s - loss: 0.2992 - accuracy: 0.9093 - val_loss: 0.2219 -
val_accuracy: 0.9350 - 15s/epoch - 9ms/step
Epoch 3/10
1688/1688 - 15s - loss: 0.2284 - accuracy: 0.9310 - val_loss: 0.1609 -
val_accuracy: 0.9548 - 15s/epoch - 9ms/step
Epoch 4/10
1688/1688 - 15s - loss: 0.1859 - accuracy: 0.9446 - val_loss: 0.1363 -
val_accuracy: 0.9623 - 15s/epoch - 9ms/step
Epoch 5/10
1688/1688 - 15s - loss: 0.1570 - accuracy: 0.9534 - val_loss: 0.1211 -
val_accuracy: 0.9638 - 15s/epoch - 9ms/step
Epoch 6/10
1688/1688 - 15s - loss: 0.1363 - accuracy: 0.9594 - val_loss: 0.1051 -
val_accuracy: 0.9708 - 15s/epoch - 9ms/step
Epoch 7/10
1688/1688 - 15s - loss: 0.1185 - accuracy: 0.9642 - val_loss: 0.0933 -
val_accuracy: 0.9748 - 15s/epoch - 9ms/step
Epoch 8/10
1688/1688 - 15s - loss: 0.1049 - accuracy: 0.9685 - val_loss: 0.0889 -
val_accuracy: 0.9770 - 15s/epoch - 9ms/step
Epoch 9/10
1688/1688 - 15s - loss: 0.0946 - accuracy: 0.9720 - val_loss: 0.0894 -
val_accuracy: 0.9765 - 15s/epoch - 9ms/step
Epoch 10/10
1688/1688 - 15s - loss: 0.0847 - accuracy: 0.9747 - val_loss: 0.0734 -
val_accuracy: 0.9795 - 15s/epoch - 9ms/step
Test accuracy with hard_sigmoid activation:0.9729

Training model with linear activation...
Epoch 1/10
1688/1688 - 11s - loss: 0.2239 - accuracy: 0.9354 - val_loss: 0.1078 -
val_accuracy: 0.9687 - 11s/epoch - 6ms/step
```

```
Epoch 2/10
1688/1688 - 10s - loss: 0.1016 - accuracy: 0.9691 - val_loss: 0.0860 -
val_accuracy: 0.9760 - 10s/epoch - 6ms/step
Epoch 3/10
1688/1688 - 10s - loss: 0.0797 - accuracy: 0.9750 - val_loss: 0.0986 -
val_accuracy: 0.9755 - 10s/epoch - 6ms/step
Epoch 4/10
1688/1688 - 10s - loss: 0.0684 - accuracy: 0.9782 - val_loss: 0.0883 -
val_accuracy: 0.9775 - 10s/epoch - 6ms/step
Epoch 5/10
1688/1688 - 10s - loss: 0.0577 - accuracy: 0.9821 - val_loss: 0.0876 -
val_accuracy: 0.9763 - 10s/epoch - 6ms/step
Epoch 6/10
1688/1688 - 10s - loss: 0.0514 - accuracy: 0.9834 - val_loss: 0.0808 -
val_accuracy: 0.9793 - 10s/epoch - 6ms/step
Epoch 7/10
1688/1688 - 10s - loss: 0.0479 - accuracy: 0.9846 - val_loss: 0.0798 -
val_accuracy: 0.9792 - 10s/epoch - 6ms/step
Epoch 8/10
1688/1688 - 11s - loss: 0.0387 - accuracy: 0.9871 - val_loss: 0.0846 -
val_accuracy: 0.9788 - 11s/epoch - 6ms/step
Epoch 10/10
1688/1688 - 10s - loss: 0.0351 - accuracy: 0.9884 - val_loss: 0.0932 -
val_accuracy: 0.9790 - 10s/epoch - 6ms/step
Test accuracy with linear activation:0.9736
```

```
for act in act_functions:
    print(f"Training model with {act} activation...")
    model = baseline_model_exp(conv_activation = act, dense_activation = act)
    history = model.fit(train_norm,y_train, epochs = 10, batch_size = 32, validation_split = 0.1, verbose = 2)

    ##predictions
    y_pred_prob = model.predict(test_norm)
    y_pred = np.argmax(y_pred_prob, axis =1)

    ##Ensure y_test is not one-hot encoded for confusion matrix

    if y_test.ndim > 1:
        y_true = np.argmax(y_test, axis =1)

    else:
        y_true = y_test
```

```

### calculate the confusion matrix
cm = confusion_matrix(y_true,y_pred)

### Calculating precision and recall
precision = precision_score(y_true, y_pred, average = "weighted")
recall = recall_score(y_true, y_pred, average = "weighted")

##Printing results
print(f"Results for activation function: {act}")
print("Confusion Matrix")
print(cm)
print(f"Precision: {precision: .4f}")
print(f"Recall: {recall: .4f}")

###plotting the confusion matrix
plt.figure(figsize = (10,7))
sns.heatmap(cm, annot = True, fmt = "g", cmap = "Blues", cbar = False)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")

plt.title(f"Confusion Matrix for {act} Activation")
plt.show()

```

Training model with gelu activation...

Epoch 1/10
1688/1688 - 17s - loss: 0.1616 - accuracy: 0.9507 - val_loss: 0.0727 -
val_accuracy: 0.9813 - 17s/epoch - 10ms/step

Epoch 2/10
1688/1688 - 15s - loss: 0.0548 - accuracy: 0.9832 - val_loss: 0.0607 -
val_accuracy: 0.9833 - 15s/epoch - 9ms/step

Epoch 3/10
1688/1688 - 16s - loss: 0.0356 - accuracy: 0.9892 - val_loss: 0.0488 -
val_accuracy: 0.9855 - 16s/epoch - 9ms/step

Epoch 4/10
1688/1688 - 15s - loss: 0.0239 - accuracy: 0.9926 - val_loss: 0.0436 -
val_accuracy: 0.9890 - 15s/epoch - 9ms/step

Epoch 5/10
1688/1688 - 15s - loss: 0.0158 - accuracy: 0.9954 - val_loss: 0.0478 -
val_accuracy: 0.9888 - 15s/epoch - 9ms/step

Epoch 6/10
1688/1688 - 15s - loss: 0.0112 - accuracy: 0.9971 - val_loss: 0.0476 -
val_accuracy: 0.9883 - 15s/epoch - 9ms/step

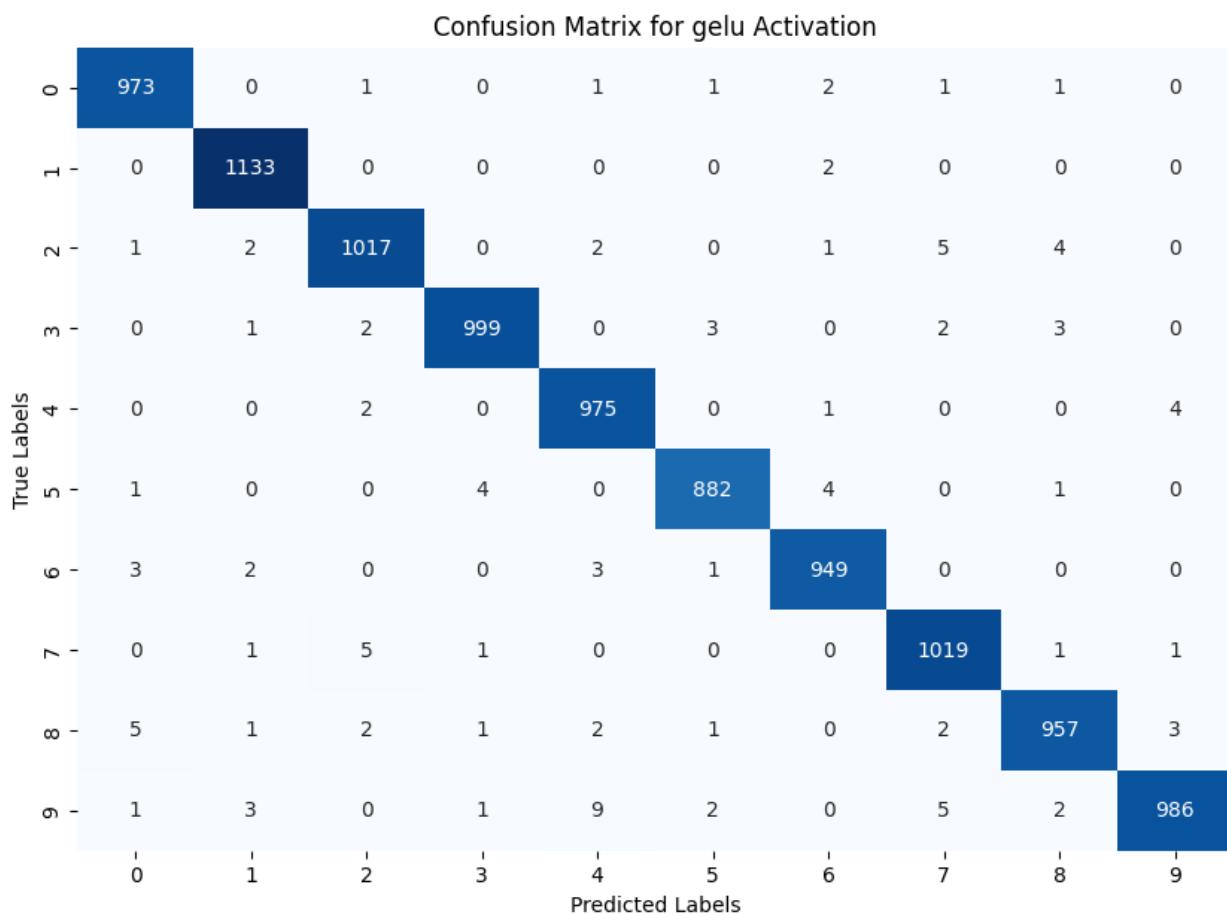
Epoch 7/10
1688/1688 - 15s - loss: 0.0077 - accuracy: 0.9979 - val_loss: 0.0511 -
val_accuracy: 0.9880 - 15s/epoch - 9ms/step

Epoch 8/10
1688/1688 - 15s - loss: 0.0048 - accuracy: 0.9989 - val_loss: 0.0477 -
val_accuracy: 0.9892 - 15s/epoch - 9ms/step

```

Epoch 9/10
1688/1688 - 16s - loss: 0.0031 - accuracy: 0.9994 - val_loss: 0.0519 -
val_accuracy: 0.9877 - 16s/epoch - 10ms/step
Epoch 10/10
1688/1688 - 16s - loss: 0.0020 - accuracy: 0.9997 - val_loss: 0.0489 -
val_accuracy: 0.9888 - 16s/epoch - 9ms/step
313/313 [=====] - 1s 4ms/step
Results for activation function: gelu
Confusion Matrix
[[ 973   0   1   0   1   1   2   1   1   0]
 [  0 1133   0   0   0   0   2   0   0   0]
 [  1   2 1017   0   2   0   1   5   4   0]
 [  0   1   2 999   0   3   0   2   3   0]
 [  0   0   2   0 975   0   1   0   0   4]
 [  1   0   0   4   0 882   4   0   1   0]
 [  3   2   0   0   3   1 949   0   0   0]
 [  0   1   5   1   0   0   0 1019   1   1]
 [  5   1   2   1   2   1   0   2 957   3]
 [  1   3   0   1   9   2   0   5   2 986]]
Precision: 0.9890
Recall: 0.9890

```



```

Training model with elu activation...
Epoch 1/10
1688/1688 - 13s - loss: 0.1795 - accuracy: 0.9450 - val_loss: 0.0677 -
val_accuracy: 0.9822 - 13s/epoch - 7ms/step
Epoch 2/10
1688/1688 - 12s - loss: 0.0629 - accuracy: 0.9811 - val_loss: 0.0572 -
val_accuracy: 0.9833 - 12s/epoch - 7ms/step
Epoch 3/10
1688/1688 - 11s - loss: 0.0383 - accuracy: 0.9881 - val_loss: 0.0558 -
val_accuracy: 0.9838 - 11s/epoch - 7ms/step
Epoch 4/10
1688/1688 - 13s - loss: 0.0257 - accuracy: 0.9921 - val_loss: 0.0551 -
val_accuracy: 0.9863 - 13s/epoch - 8ms/step
Epoch 5/10
1688/1688 - 12s - loss: 0.0168 - accuracy: 0.9951 - val_loss: 0.0539 -
val_accuracy: 0.9868 - 12s/epoch - 7ms/step
Epoch 6/10
1688/1688 - 12s - loss: 0.0113 - accuracy: 0.9970 - val_loss: 0.0589 -
val_accuracy: 0.9847 - 12s/epoch - 7ms/step
Epoch 7/10
1688/1688 - 12s - loss: 0.0078 - accuracy: 0.9979 - val_loss: 0.0513 -
val_accuracy: 0.9880 - 12s/epoch - 7ms/step
Epoch 8/10
1688/1688 - 12s - loss: 0.0047 - accuracy: 0.9991 - val_loss: 0.0517 -
val_accuracy: 0.9883 - 12s/epoch - 7ms/step
Epoch 9/10
1688/1688 - 12s - loss: 0.0034 - accuracy: 0.9995 - val_loss: 0.0543 -
val_accuracy: 0.9883 - 12s/epoch - 7ms/step
Epoch 10/10
1688/1688 - 11s - loss: 0.0021 - accuracy: 0.9997 - val_loss: 0.0531 -
val_accuracy: 0.9883 - 11s/epoch - 7ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: elu
Confusion Matrix
[[ 972    0    0    0    0    1    3    2    2    0]
 [  0 1129    1    1    0    0    2    1    1    0]
 [  1    2 1018    0    2    0    1    4    4    0]
 [  0    0    2 998    0    2    0    3    1    4]
 [  0    1    3    0 972    0    1    0    1    4]
 [  1    0    0    5    0 882    3    0    0    1]
 [  3    2    1    1    2    2 946    0    1    0]
 [  1    2    5    1    2    0    0 1013    1    3]
 [  4    1    3    2    0    1    0    1 958    4]
 [  1    3    0    3   10    2    0    4    1 985]]]
Precision: 0.9873
Recall: 0.9873

```

Confusion Matrix for elu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	972	0	0	0	0	1	3	2	2	0
0	972	0	0	0	0	0	2	1	1	0
1	0	1129	1	1	0	0	2	1	1	0
2	1	2	1018	0	2	0	1	4	4	0
3	0	0	2	998	0	2	0	3	1	4
4	0	1	3	0	972	0	1	0	1	4
5	1	0	0	5	0	882	3	0	0	1
6	3	2	1	1	2	2	946	0	1	0
7	1	2	5	1	2	0	0	1013	1	3
8	4	1	3	2	0	1	0	1	958	4
9	1	3	0	3	10	2	0	4	1	985
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

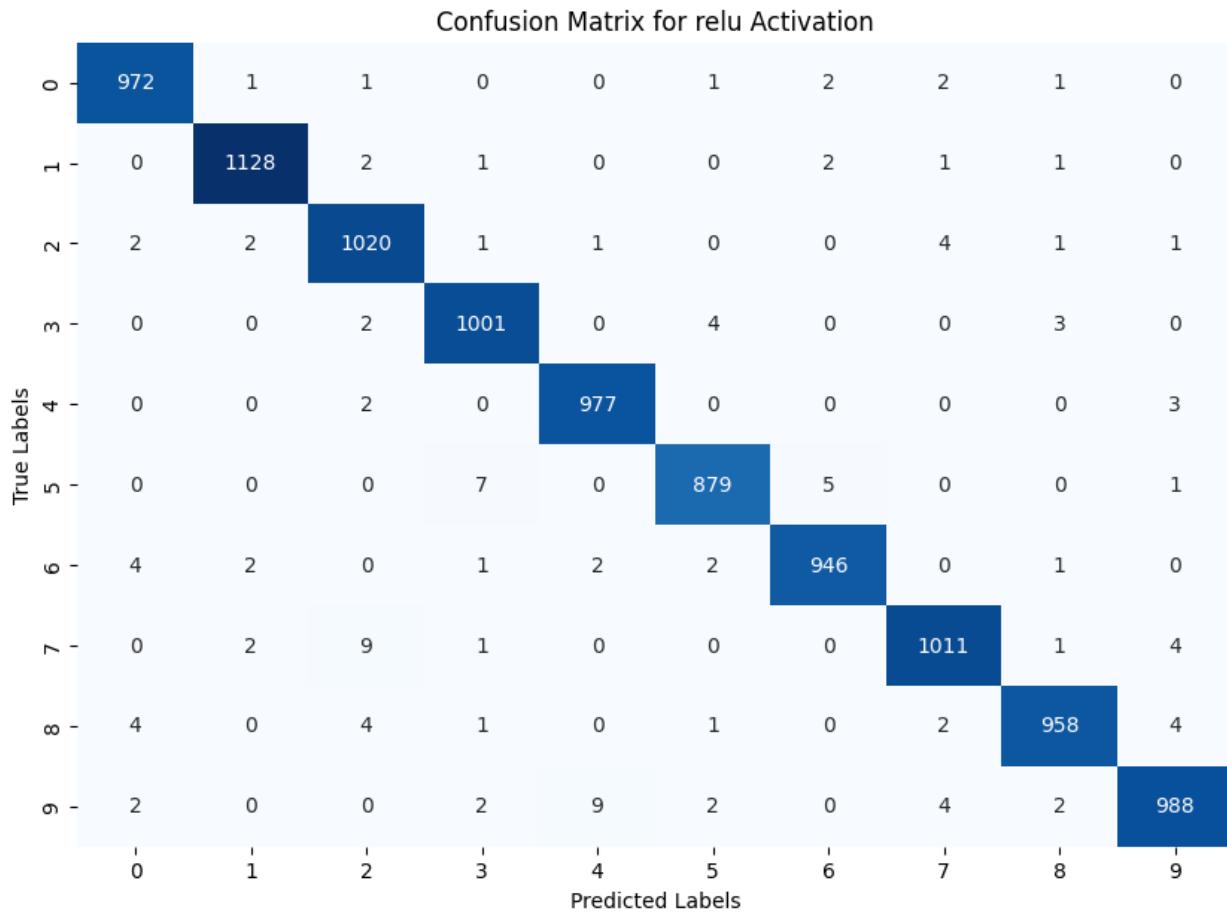
```

Training model with relu activation...
Epoch 1/10
1688/1688 - 12s - loss: 0.1646 - accuracy: 0.9502 - val_loss: 0.0654 -
val_accuracy: 0.9817 - 12s/epoch - 7ms/step
Epoch 2/10
1688/1688 - 11s - loss: 0.0576 - accuracy: 0.9832 - val_loss: 0.0489 -
val_accuracy: 0.9877 - 11s/epoch - 6ms/step
Epoch 3/10
1688/1688 - 11s - loss: 0.0379 - accuracy: 0.9886 - val_loss: 0.0464 -
val_accuracy: 0.9868 - 11s/epoch - 6ms/step
Epoch 4/10
1688/1688 - 11s - loss: 0.0260 - accuracy: 0.9920 - val_loss: 0.0409 -
val_accuracy: 0.9887 - 11s/epoch - 6ms/step
Epoch 5/10
1688/1688 - 11s - loss: 0.0181 - accuracy: 0.9946 - val_loss: 0.0533 -
val_accuracy: 0.9852 - 11s/epoch - 6ms/step
Epoch 6/10
1688/1688 - 11s - loss: 0.0119 - accuracy: 0.9967 - val_loss: 0.0434 -
val_accuracy: 0.9883 - 11s/epoch - 6ms/step
Epoch 7/10
1688/1688 - 10s - loss: 0.0080 - accuracy: 0.9980 - val_loss: 0.0473 -

```

```
val_accuracy: 0.9875 - 10s/epoch - 6ms/step
Epoch 8/10
1688/1688 - 11s - loss: 0.0057 - accuracy: 0.9988 - val_loss: 0.0428 -
val_accuracy: 0.9893 - 11s/epoch - 6ms/step
Epoch 9/10
1688/1688 - 11s - loss: 0.0032 - accuracy: 0.9994 - val_loss: 0.0429 -
val_accuracy: 0.9897 - 11s/epoch - 6ms/step
Epoch 10/10
1688/1688 - 11s - loss: 0.0023 - accuracy: 0.9996 - val_loss: 0.0462 -
val_accuracy: 0.9890 - 11s/epoch - 6ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: relu
Confusion Matrix
[[ 972    1    1    0    0    1    2    2    1    0]
 [  0  1128    2    1    0    0    2    1    1    0]
 [  2    2  1020    1    1    0    0    4    1    1]
 [  0    0    2  1001    0    4    0    0    3    0]
 [  0    0    2    0   977    0    0    0    0    3]
 [  0    0    0    7    0   879    5    0    0    1]
 [  4    2    0    1    2    2   946    0    1    0]
 [  0    2    9    1    0    0    0  1011    1    4]
 [  4    0    4    1    0    1    0    2   958    4]
 [  2    0    0    2    9    2    0    4    2   988]]
```

Precision: 0.9880
Recall: 0.9880



```

Training model with sigmoid activation...
Epoch 1/10
1688/1688 - 12s - loss: 0.6700 - accuracy: 0.7966 - val_loss: 0.2486 -
val_accuracy: 0.9288 - 12s/epoch - 7ms/step
Epoch 2/10
1688/1688 - 11s - loss: 0.2887 - accuracy: 0.9124 - val_loss: 0.2020 -
val_accuracy: 0.9395 - 11s/epoch - 7ms/step
Epoch 3/10
1688/1688 - 11s - loss: 0.2200 - accuracy: 0.9326 - val_loss: 0.1598 -
val_accuracy: 0.9553 - 11s/epoch - 7ms/step
Epoch 4/10
1688/1688 - 11s - loss: 0.1812 - accuracy: 0.9458 - val_loss: 0.1444 -
val_accuracy: 0.9575 - 11s/epoch - 7ms/step
Epoch 5/10
1688/1688 - 11s - loss: 0.1550 - accuracy: 0.9539 - val_loss: 0.1201 -
val_accuracy: 0.9687 - 11s/epoch - 7ms/step
Epoch 6/10
1688/1688 - 11s - loss: 0.1333 - accuracy: 0.9604 - val_loss: 0.1157 -
val_accuracy: 0.9687 - 11s/epoch - 7ms/step
Epoch 7/10
1688/1688 - 11s - loss: 0.1167 - accuracy: 0.9656 - val_loss: 0.0977 -

```

```
val_accuracy: 0.9745 - 11s/epoch - 7ms/step
Epoch 8/10
1688/1688 - 11s - loss: 0.1033 - accuracy: 0.9699 - val_loss: 0.0884 -
val_accuracy: 0.9762 - 11s/epoch - 7ms/step
Epoch 9/10
1688/1688 - 12s - loss: 0.0930 - accuracy: 0.9723 - val_loss: 0.0833 -
val_accuracy: 0.9763 - 12s/epoch - 7ms/step
Epoch 10/10
1688/1688 - 12s - loss: 0.0833 - accuracy: 0.9760 - val_loss: 0.0777 -
val_accuracy: 0.9795 - 12s/epoch - 7ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix
[[ 963    0    0    0    0    3    6    3    3    2]
 [  0 1125    3    0    0    2    1    2    2    0]
 [  4    5 1008    1    3    0    2    6    3    0]
 [  0    0    6  989    0    4    0    7    2    2]
 [  0    0    2    0  957    0    7    4    2   10]
 [  3    0    1    6    0  875    4    1    1    1]
 [  4    3    0    0    1    9  940    1    0    0]
 [  1    4   13    3    0    0    0 1005    0    2]
 [  4    0    7   13    3    8    5    5   927    2]
 [  4    5    2   10   14    6    1   13    1  953]]]
Precision: 0.9743
Recall: 0.9742
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	963	0	0	0	0	3	6	3	3	2	
0	963	0	0	0	0	3	6	3	3	2	
1	0	1125	3	0	0	2	1	2	2	0	
2	4	5	1008	1	3	0	2	6	3	0	
3	0	0	6	989	0	4	0	7	2	2	
4	0	0	2	0	957	0	7	4	2	10	
5	3	0	1	6	0	875	4	1	1	1	
6	4	3	0	0	1	9	940	1	0	0	
7	1	4	13	3	0	0	0	1005	0	2	
8	4	0	7	13	3	8	5	5	927	2	
9	4	5	2	10	14	6	1	13	1	953	
0	1	2	3	4	5	6	7	8	9	9	
Predicted Labels											

Training model with tanh activation...

Epoch 1/10

1688/1688 - 12s - loss: 0.1865 - accuracy: 0.9443 - val_loss: 0.0780 - val_accuracy: 0.9788 - 12s/epoch - 7ms/step

Epoch 2/10

1688/1688 - 12s - loss: 0.0712 - accuracy: 0.9795 - val_loss: 0.0660 - val_accuracy: 0.9808 - 12s/epoch - 7ms/step

Epoch 3/10

1688/1688 - 11s - loss: 0.0451 - accuracy: 0.9873 - val_loss: 0.0467 - val_accuracy: 0.9870 - 11s/epoch - 7ms/step

Epoch 4/10

1688/1688 - 11s - loss: 0.0305 - accuracy: 0.9922 - val_loss: 0.0491 - val_accuracy: 0.9863 - 11s/epoch - 7ms/step

Epoch 5/10

1688/1688 - 12s - loss: 0.0207 - accuracy: 0.9948 - val_loss: 0.0447 - val_accuracy: 0.9878 - 12s/epoch - 7ms/step

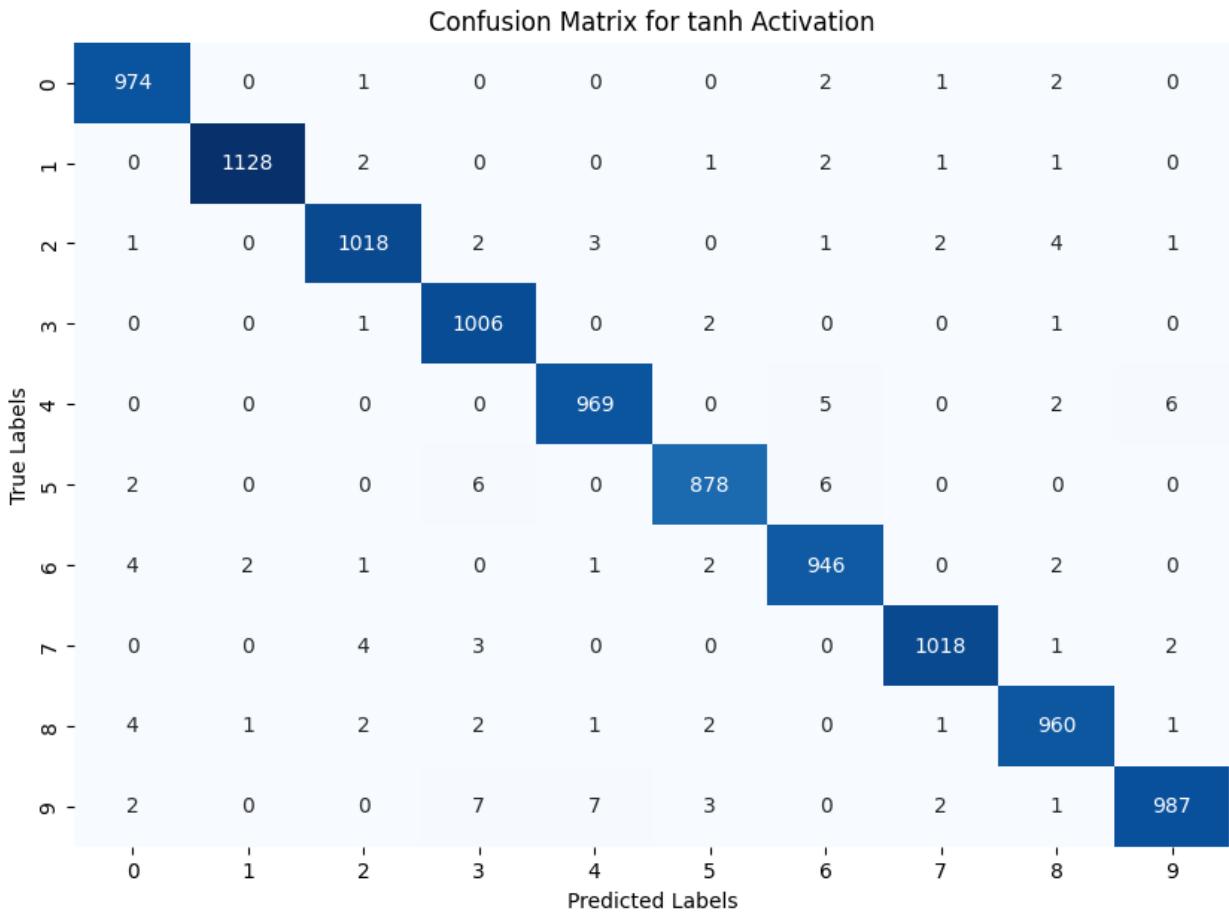
Epoch 6/10

1688/1688 - 11s - loss: 0.0140 - accuracy: 0.9972 - val_loss: 0.0436 - val_accuracy: 0.9885 - 11s/epoch - 7ms/step

Epoch 7/10

1688/1688 - 11s - loss: 0.0097 - accuracy: 0.9984 - val_loss: 0.0430 -

```
val_accuracy: 0.9872 - 11s/epoch - 7ms/step
Epoch 8/10
1688/1688 - 11s - loss: 0.0067 - accuracy: 0.9993 - val_loss: 0.0454 -
val_accuracy: 0.9882 - 11s/epoch - 6ms/step
Epoch 9/10
1688/1688 - 11s - loss: 0.0050 - accuracy: 0.9995 - val_loss: 0.0433 -
val_accuracy: 0.9887 - 11s/epoch - 7ms/step
Epoch 10/10
1688/1688 - 17s - loss: 0.0038 - accuracy: 0.9999 - val_loss: 0.0435 -
val_accuracy: 0.9888 - 17s/epoch - 10ms/step
313/313 [=====] - 2s 4ms/step
Results for activation function: tanh
Confusion Matrix
[[ 974    0    1    0    0    0    2    1    2    0]
 [  0 1128    2    0    0    1    2    1    1    0]
 [  1    0 1018    2    3    0    1    2    4    1]
 [  0    0    1 1006    0    2    0    0    1    0]
 [  0    0    0    0  969    0    5    0    2    6]
 [  2    0    0    6    0  878    6    0    0    0]
 [  4    2    1    0    1    2  946    0    2    0]
 [  0    0    4    3    0    0    0 1018    1    2]
 [  4    1    2    2    1    2    0    1  960    1]
 [  2    0    0    7    7    3    0    2    1  987]]]
Precision: 0.9884
Recall: 0.9884
```



```

Training model with softplus activation...
Epoch 1/10
1688/1688 - 13s - loss: 0.5014 - accuracy: 0.8410 - val_loss: 0.1979 -
val_accuracy: 0.9435 - 13s/epoch - 8ms/step
Epoch 2/10
1688/1688 - 12s - loss: 0.1982 - accuracy: 0.9407 - val_loss: 0.1605 -
val_accuracy: 0.9493 - 12s/epoch - 7ms/step
Epoch 3/10
1688/1688 - 12s - loss: 0.1287 - accuracy: 0.9604 - val_loss: 0.0948 -
val_accuracy: 0.9728 - 12s/epoch - 7ms/step
Epoch 4/10
1688/1688 - 12s - loss: 0.0994 - accuracy: 0.9698 - val_loss: 0.0986 -
val_accuracy: 0.9740 - 12s/epoch - 7ms/step
Epoch 5/10
1688/1688 - 12s - loss: 0.0827 - accuracy: 0.9749 - val_loss: 0.0771 -
val_accuracy: 0.9815 - 12s/epoch - 7ms/step
Epoch 6/10
1688/1688 - 12s - loss: 0.0705 - accuracy: 0.9781 - val_loss: 0.0706 -
val_accuracy: 0.9830 - 12s/epoch - 7ms/step
Epoch 7/10
1688/1688 - 12s - loss: 0.0629 - accuracy: 0.9804 - val_loss: 0.0843 -

```

```
val_accuracy: 0.9802 - 12s/epoch - 7ms/step
Epoch 8/10
1688/1688 - 12s - loss: 0.0535 - accuracy: 0.9839 - val_loss: 0.0908 -
val_accuracy: 0.9752 - 12s/epoch - 7ms/step
Epoch 9/10
1688/1688 - 12s - loss: 0.0463 - accuracy: 0.9853 - val_loss: 0.0640 -
val_accuracy: 0.9842 - 12s/epoch - 7ms/step
Epoch 10/10
1688/1688 - 12s - loss: 0.0422 - accuracy: 0.9862 - val_loss: 0.0624 -
val_accuracy: 0.9852 - 12s/epoch - 7ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softplus
Confusion Matrix
[[ 972    0    1    0    1    0    3    2    0    1]
 [  0 1122    3    0    3    1    2    3    1    0]
 [  5    1 1006    0    4    0    1    8    6    1]
 [  0    0    6 993    0    2    0    2    5    2]
 [  1    0    1    0 976    0    2    0    0    2]
 [  2    0    1    8    0 870    8    0    2    1]
 [  8    3    0    0    7    2 936    0    2    0]
 [  1    1    7    1    4    0    0 1008    1    5]
 [  7    0    2    0    4    0    4    5 950    2]
 [  2    0    1    6   19    0    0    4    4 973]]]
Precision: 0.9807
Recall: 0.9806
```

Confusion Matrix for softplus Activation											
	0	1	2	3	4	5	6	7	8	9	
0	972	0	1	0	1	0	3	2	0	1	
1	0	1122	3	0	3	1	2	3	1	0	
2	5	1	1006	0	4	0	1	8	6	1	
3	0	0	6	993	0	2	0	2	5	2	
4	1	0	1	0	976	0	2	0	0	2	
5	2	0	1	8	0	870	8	0	2	1	
6	8	3	0	0	7	2	936	0	2	0	
7	1	1	7	1	4	0	0	1008	1	5	
8	7	0	2	0	4	0	4	5	950	2	
9	2	0	1	6	19	0	0	4	4	973	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	

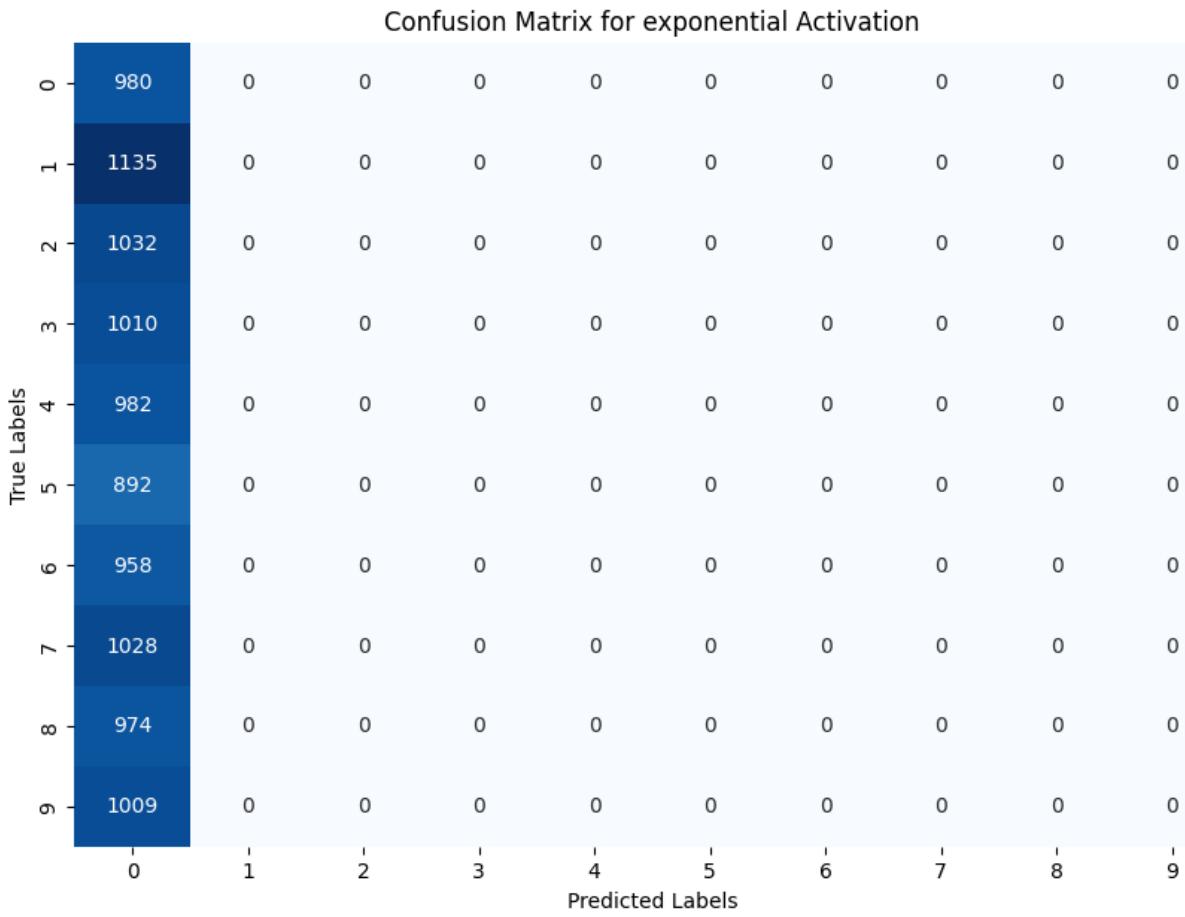
```

Training model with exponential activation...
Epoch 1/10
1688/1688 - 12s - loss: nan - accuracy: 0.0989 - val_loss: nan -
val_accuracy: 0.0978 - 12s/epoch - 7ms/step
Epoch 2/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 3/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 4/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 7ms/step
Epoch 5/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 7ms/step
Epoch 6/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 7ms/step
Epoch 7/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -

```

```
val_accuracy: 0.0978 - 11s/epoch - 7ms/step
Epoch 8/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 9/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 6ms/step
Epoch 10/10
1688/1688 - 11s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 11s/epoch - 7ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: exponential
Confusion Matrix
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
Precision: 0.0096
Recall: 0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```

Training model with softmax activation...
Epoch 1/10
1688/1688 - 14s - loss: 2.3015 - accuracy: 0.1133 - val_loss: 2.3008 -
val_accuracy: 0.1050 - 14s/epoch - 8ms/step
Epoch 2/10
1688/1688 - 13s - loss: 2.3003 - accuracy: 0.1134 - val_loss: 2.3011 -
val_accuracy: 0.1050 - 13s/epoch - 8ms/step
Epoch 3/10
1688/1688 - 13s - loss: 2.2977 - accuracy: 0.1175 - val_loss: 2.2960 -
val_accuracy: 0.1050 - 13s/epoch - 8ms/step
Epoch 4/10
1688/1688 - 14s - loss: 2.1019 - accuracy: 0.2132 - val_loss: 1.8186 -
val_accuracy: 0.3262 - 14s/epoch - 8ms/step
Epoch 5/10
1688/1688 - 14s - loss: 1.6039 - accuracy: 0.3362 - val_loss: 1.4958 -
val_accuracy: 0.3407 - 14s/epoch - 8ms/step
Epoch 6/10
1688/1688 - 14s - loss: 1.4876 - accuracy: 0.3599 - val_loss: 1.5193 -
val_accuracy: 0.3475 - 14s/epoch - 8ms/step
Epoch 7/10
1688/1688 - 14s - loss: 1.4327 - accuracy: 0.3886 - val_loss: 1.3654 -

```

```
val_accuracy: 0.3835 - 14s/epoch - 8ms/step
Epoch 8/10
1688/1688 - 14s - loss: 1.3598 - accuracy: 0.4193 - val_loss: 1.2968 -
val_accuracy: 0.4407 - 14s/epoch - 8ms/step
Epoch 9/10
1688/1688 - 14s - loss: 1.3066 - accuracy: 0.4374 - val_loss: 1.2516 -
val_accuracy: 0.4612 - 14s/epoch - 8ms/step
Epoch 10/10
1688/1688 - 14s - loss: 1.2677 - accuracy: 0.4543 - val_loss: 1.2412 -
val_accuracy: 0.4828 - 14s/epoch - 8ms/step
313/313 [=====] - 1s 4ms/step
Results for activation function: softmax
Confusion Matrix
[[ 953   0  22   2   2   0   0   1   0   0]
 [  0 1108   5   6   0   0   0   0  16   0]
 [ 580   2 317  99  18   1   0   5   8   2]
 [ 34   9 210 601  17   3   0   3 132   1]
 [ 19   9  62   1 307   8   0 277  35 264]
 [ 34  27 209 473  13   5   0   0 129   2]
 [ 835   1 107  12   1   0   0   1   1   0]
 [  4  11  19   4  61   1   0 872  13  43]
 [ 18  53 114 434  16   4   0   1 328   6]
 [  8   8  24   6 139   1   0 513  20 290]]]
Precision: 0.4276
Recall: 0.4781

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	0	953	0	22	2	2	0	0	1	0	0
1	0	1108	5	6	0	0	0	0	16	0	
2	580	2	317	99	18	1	0	5	8	2	
3	34	9	210	601	17	3	0	3	132	1	
4	19	9	62	1	307	8	0	277	35	264	
5	34	27	209	473	13	5	0	0	129	2	
6	835	1	107	12	1	0	0	1	1	0	
7	4	11	19	4	61	1	0	872	13	43	
8	18	53	114	434	16	4	0	1	328	6	
9	8	8	24	6	139	1	0	513	20	290	
0	1	2	3	4	5	6	7	8	9		
Predicted Labels											

```

Training model with softsign activation...
Epoch 1/10
1688/1688 - 11s - loss: 0.2497 - accuracy: 0.9284 - val_loss: 0.1057 -
val_accuracy: 0.9707 - 11s/epoch - 7ms/step
Epoch 2/10
1688/1688 - 11s - loss: 0.1066 - accuracy: 0.9692 - val_loss: 0.0847 -
val_accuracy: 0.9793 - 11s/epoch - 7ms/step
Epoch 3/10
1688/1688 - 11s - loss: 0.0713 - accuracy: 0.9797 - val_loss: 0.0602 -
val_accuracy: 0.9850 - 11s/epoch - 7ms/step
Epoch 4/10
1688/1688 - 11s - loss: 0.0534 - accuracy: 0.9851 - val_loss: 0.0555 -
val_accuracy: 0.9860 - 11s/epoch - 7ms/step
Epoch 5/10
1688/1688 - 11s - loss: 0.0422 - accuracy: 0.9887 - val_loss: 0.0515 -
val_accuracy: 0.9855 - 11s/epoch - 7ms/step
Epoch 6/10
1688/1688 - 11s - loss: 0.0331 - accuracy: 0.9914 - val_loss: 0.0504 -
val_accuracy: 0.9860 - 11s/epoch - 7ms/step
Epoch 7/10
1688/1688 - 11s - loss: 0.0270 - accuracy: 0.9932 - val_loss: 0.0472 -

```

```
val_accuracy: 0.9877 - 11s/epoch - 7ms/step
Epoch 8/10
1688/1688 - 11s - loss: 0.0216 - accuracy: 0.9951 - val_loss: 0.0485 -
val_accuracy: 0.9878 - 11s/epoch - 6ms/step
Epoch 9/10
1688/1688 - 11s - loss: 0.0179 - accuracy: 0.9963 - val_loss: 0.0490 -
val_accuracy: 0.9878 - 11s/epoch - 7ms/step
Epoch 10/10
1688/1688 - 11s - loss: 0.0149 - accuracy: 0.9971 - val_loss: 0.0465 -
val_accuracy: 0.9878 - 11s/epoch - 7ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softsign
Confusion Matrix
[[ 971    0    1    0    0    0    1    3    3    1
   0 1128    1    1    0    1    2    1    1    0
   3    3 1010    3    2    0    1    6    2    2
   0    0    2 1001    0    1    0    3    3    0
   1    1    1    0  965    0    2    1    1    10
   2    0    2    6    0  874    7    0    1    0
   8    2    2    1    1    2  939    0    3    0
   1    3    5    2    0    0    0 1015    0    2
   4    0    3    1    0    0    0    1  963    2
   2    1    0    4    8    1    0    8    2  983]]]
Precision: 0.9849
Recall: 0.9849
```

Confusion Matrix for softsign Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	971	0	1	0	0	0	1	3	3	1	
0	971	0	1	0	0	0	1	3	3	1	
1	0	1128	1	1	0	1	2	1	1	0	
2	3	3	1010	3	2	0	1	6	2	2	
3	0	0	2	1001	0	1	0	3	3	0	
4	1	1	1	0	965	0	2	1	1	10	
5	2	0	2	6	0	874	7	0	1	0	
6	8	2	2	1	1	2	939	0	3	0	
7	1	3	5	2	0	0	0	0	1015	0	2
8	4	0	3	1	0	0	0	1	963	2	
9	2	1	0	4	8	1	0	8	2	983	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```

Training model with hard_sigmoid activation...
Epoch 1/10
1688/1688 - 16s - loss: 0.6551 - accuracy: 0.8016 - val_loss: 0.2787 -
val_accuracy: 0.9133 - 16s/epoch - 10ms/step
Epoch 2/10
1688/1688 - 15s - loss: 0.2947 - accuracy: 0.9104 - val_loss: 0.2027 -
val_accuracy: 0.9417 - 15s/epoch - 9ms/step
Epoch 3/10
1688/1688 - 15s - loss: 0.2271 - accuracy: 0.9313 - val_loss: 0.1591 -
val_accuracy: 0.9553 - 15s/epoch - 9ms/step
Epoch 4/10
1688/1688 - 15s - loss: 0.1852 - accuracy: 0.9439 - val_loss: 0.1294 -
val_accuracy: 0.9642 - 15s/epoch - 9ms/step
Epoch 5/10
1688/1688 - 15s - loss: 0.1545 - accuracy: 0.9538 - val_loss: 0.1132 -
val_accuracy: 0.9673 - 15s/epoch - 9ms/step
Epoch 6/10
1688/1688 - 14s - loss: 0.1341 - accuracy: 0.9599 - val_loss: 0.1039 -
val_accuracy: 0.9717 - 14s/epoch - 9ms/step
Epoch 7/10
1688/1688 - 15s - loss: 0.1168 - accuracy: 0.9655 - val_loss: 0.0915 -

```

```
val_accuracy: 0.9767 - 15s/epoch - 9ms/step
Epoch 8/10
1688/1688 - 14s - loss: 0.1028 - accuracy: 0.9696 - val_loss: 0.0869 -
val_accuracy: 0.9758 - 14s/epoch - 9ms/step
Epoch 9/10
1688/1688 - 14s - loss: 0.0927 - accuracy: 0.9729 - val_loss: 0.0807 -
val_accuracy: 0.9775 - 14s/epoch - 8ms/step
Epoch 10/10
1688/1688 - 14s - loss: 0.0826 - accuracy: 0.9756 - val_loss: 0.0799 -
val_accuracy: 0.9792 - 14s/epoch - 8ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: hard_sigmoid
Confusion Matrix
[[ 968    0    0    1    0    2    3    2    1    3]
 [  0 1126    3    1    0    1    3    1    0    0]
 [  6    5 999    5    3    0    4    8    1    1]
 [  0    0    0 1001    0    1    0    3    3    2]
 [  0    0    3    0   961    0    5    1    1   11]
 [  6    1    0   18    1   850    8    2    3    3]
 [  8    3    1    1    3    4   935    0    3    0]
 [  1    5    8    8    0    0    0   997    1    8]
 [  5    2    3   18    2    1    6    4   929    4]
 [  6    5    1   16   10    1    1    8    1  960]]]
Precision: 0.9729
Recall: 0.9726
```

Confusion Matrix for hard_sigmoid Activation												
	0	1	2	3	4	5	6	7	8	9	10	11
True Labels	0	968	0	0	1	0	2	3	2	1	3	
1	0	1126	3	1	0	1	3	1	0	0		
2	6	5	999	5	3	0	4	8	1	1		
3	0	0	0	1001	0	1	0	3	3	2		
4	0	0	3	0	961	0	5	1	1	11		
5	6	1	0	18	1	850	8	2	3	3		
6	8	3	1	1	3	4	935	0	3	0		
7	1	5	8	8	0	0	0	997	1	8		
8	5	2	3	18	2	1	6	4	929	4		
9	6	5	1	16	10	1	1	8	1	960		
10	1	1	2	3	4	5	6	7	8	9		
11												

```

Training model with linear activation...
Epoch 1/10
1688/1688 - 11s - loss: 0.2174 - accuracy: 0.9370 - val_loss: 0.1007 -
val_accuracy: 0.9713 - 11s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 10s - loss: 0.1002 - accuracy: 0.9695 - val_loss: 0.0892 -
val_accuracy: 0.9740 - 10s/epoch - 6ms/step
Epoch 3/10
1688/1688 - 10s - loss: 0.0768 - accuracy: 0.9762 - val_loss: 0.0786 -
val_accuracy: 0.9810 - 10s/epoch - 6ms/step
Epoch 4/10
1688/1688 - 10s - loss: 0.0651 - accuracy: 0.9800 - val_loss: 0.0730 -
val_accuracy: 0.9798 - 10s/epoch - 6ms/step
Epoch 5/10
1688/1688 - 10s - loss: 0.0571 - accuracy: 0.9816 - val_loss: 0.0732 -
val_accuracy: 0.9822 - 10s/epoch - 6ms/step
Epoch 6/10
1688/1688 - 10s - loss: 0.0508 - accuracy: 0.9837 - val_loss: 0.0733 -
val_accuracy: 0.9810 - 10s/epoch - 6ms/step
Epoch 7/10
1688/1688 - 10s - loss: 0.0450 - accuracy: 0.9851 - val_loss: 0.0797 -

```

```
val_accuracy: 0.9823 - 10s/epoch - 6ms/step
Epoch 8/10
1688/1688 - 10s - loss: 0.0391 - accuracy: 0.9874 - val_loss: 0.0838 -
val_accuracy: 0.9805 - 10s/epoch - 6ms/step
Epoch 9/10
1688/1688 - 10s - loss: 0.0374 - accuracy: 0.9873 - val_loss: 0.0808 -
val_accuracy: 0.9807 - 10s/epoch - 6ms/step
Epoch 10/10
1688/1688 - 10s - loss: 0.0320 - accuracy: 0.9894 - val_loss: 0.0796 -
val_accuracy: 0.9818 - 10s/epoch - 6ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: linear
Confusion Matrix
[[ 964    0    5    0    1    2    3    3    0    2]
 [  1 1123    2    3    0    0    2    1    3    0]
 [  3    0 1003    6    4    0    3    5    6    2]
 [  1    0    3 997    0    5    0    1    3    0]
 [  1    0    1    0 969    0    1    1    4    5]
 [  3    0    2   12    0 865    8    0    1    1]
 [  8    2    2    0    1    2 940    0    3    0]
 [  1    3   10   11    1    0    0 998    4    0]
 [  7    0    5    4    0    3    0    5 948    2]
 [  3    1    1    8   12    6    0    8   11 959]]]
Precision: 0.9767
Recall: 0.9766
```

Confusion Matrix for linear Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	0	964	0	5	0	1	2	3	3	0	2
0	1	1123	2	3	0	0	0	2	1	3	0
1	3	0	1003	6	4	0	3	5	6	2	
2	1	0	3	997	0	5	0	1	3	0	
3	1	0	1	0	969	0	1	1	4	5	
4	3	0	2	12	0	865	8	0	1	1	
5	8	2	2	0	1	2	940	0	3	0	
6	1	3	10	11	1	0	0	998	4	0	
7	7	0	5	4	0	3	0	5	948	2	
8	3	1	1	8	12	6	0	8	11	959	
9	1	2	3	4	5	6	7	8	9	9	

Effects of Using Different Activation Functions:

In this experiment, we have combined different activation functions in the baseline model. These are gelu, elu, relu, sigmoid, tanh, softplus, exponential, softmax, softsign, hard_sigmoid, and linear. The experiments revealed a range of outcomes, with most activation functions achieving high precision and recall scores, indicating strong model performance.

Gelu: Precision: 0.9890 and Recall: 0.9890

Elu: Precision: 0.9873, Recall: 0.9873

Relu: Precision: 0.9880 , Recall: 0.9880

Sigmoid: Precision:0.9743 , Recall: 0.9743

Tanh: Precision: 0.9884 , Recall: 0.9884

Softplus: Precision: 0.9807, Recall: 0.9806

Exponential Activation: Precision: 0.0096, Recall: 0.0980, No classification

Softmax Activation: Precision: 0.4276, Recall: 0.4781

Softsign: Precision : 0.9849, Recall: 0.9849

Hard sigmoid: Precision: 0.9729, Recall: 0.9726

Linear: Precision: 0.9767, Recall: 0.9766

Analysis

Notably, GELU, ReLU, and Tanh functions led to the highest levels of precision and recall, closely followed by ELU and Softsign. Sigmoid and Hard Sigmoid, while still performing well, trailed slightly behind the leaders. Softplus showed a moderate decrease in performance. In contrast, Exponential Activation and Softmax Activation within the inner layers significantly underperformed, with the Exponential Activation function demonstrating almost no classification capability in this context.

Combining the Activation Function Choices with Different Network Sizes and Depths:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense, Activation
from tensorflow.keras.optimizers import SGD

act_functions = ["relu", "sigmoid", "tanh", "softmax", "exponential"]

layers_configs = [(1,1), (2,2), (3,3)] ## Tuple Format:
(num_conv_layers, num_dense_layers)
neuron_configs = [64, 128, 256]

def create_flexible_model_2(conv_activation='relu',
dense_activation='relu', num_conv_layers=4, num_dense_layers=4,
num_neurons=64):
    model = Sequential()

    # Adjust the Conv2D layer to use num_neurons for the number of
    filters
    for i in range(num_conv_layers):
        model.add(Conv2D(num_neurons, (3, 3),
activation=conv_activation, kernel_initializer="he_uniform",
padding='same'))
        if i % 2 == 1: # Add pooling layer after every two conv
layers instead of after every conv layer
            model.add(MaxPooling2D((2, 2)))

    model.add(Flatten())

    # Dense layers already use num_neurons
    for _ in range(num_dense_layers):
        model.add(Dense(num_neurons, activation=dense_activation,
kernel_initializer="he_uniform"))

    model.add(Dense(10, activation='softmax')) # The output layer
stays the same
```

```

# Use SGD as the optimizer
opt = SGD(learning_rate=0.01, momentum=0.9)
model.compile(optimizer=opt, loss="categorical_crossentropy",
metrics=["accuracy"])
return model

def create_flexible_model_3(conv_activation='relu',
dense_activation='relu', num_conv_layers=3, num_dense_layers=3):
    model = Sequential()

    for i in range(num_conv_layers):
        model.add(Conv2D(64, (3, 3), activation=conv_activation,
kernel_initializer="he_uniform", padding='same'))
        if i % 2 == 1: # Add pooling layer after every two conv
layers instead of after every conv layer
            model.add(MaxPooling2D((2, 2)))

    model.add(Flatten())

    for _ in range(num_dense_layers):
        model.add(Dense(100, activation=dense_activation,
kernel_initializer="he_uniform"))

    model.add(Dense(10, activation='softmax'))

    opt = SGD(learning_rate = 0.01, momentum = 0.9)
    model.compile(optimizer = opt, loss = "categorical_crossentropy",
metrics = ["accuracy"])
    return model

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense, Activation
from tensorflow.keras.optimizers import SGD

def create_flexible_model(conv_activation='relu',
dense_activation='relu', num_conv_layers=1, num_dense_layers=1,
num_neurons= 100):
    model = Sequential()

    # Add specified number of convolutional layers
    for _ in range(num_conv_layers):
        model.add(Conv2D(32, (3, 3), activation=conv_activation,
kernel_initializer="he_uniform", input_shape=(28, 28, 1)))
        model.add(MaxPooling2D((2, 2)))

    model.add(Flatten())

    # Add specified number of dense layers

```

```

        for _ in range(num_dense_layers):
            model.add(Dense(num_neurons, activation=dense_activation,
kernel_initializer="he_uniform"))

        model.add(Dense(10, activation='softmax')) # Output layer remains
the same
        opt = SGD(learning_rate = 0.01, momentum = 0.9)
        model.compile(optimizer=opt, loss='categorical_crossentropy',
metrics=['accuracy'])
        return model

for act in act_functions:
    for conv_layers, dense_layers in layers_configs:
        for num_neurons in neuron_configs:

            print(f"Training Model with {act} activation,
{conv_layers} conv_layers, {dense_layers} dense layers..")
            model = create_flexible_model(conv_activation=act,
dense_activation=act, num_conv_layers=conv_layers,
num_dense_layers=dense_layers, num_neurons = num_neurons)
            history = model.fit(train_norm, y_train, epochs=10,
batch_size=32, validation_split=0.1, verbose=2)

            # predictions
            y_pred_prob = model.predict(test_norm)
            y_pred = np.argmax(y_pred_prob, axis=1)

            # Ensure y_test is not one-hot encoded for confusion
matrix
            if y_test.ndim > 1:
                y_true = np.argmax(y_test, axis=1)
            else:
                y_true = y_test

            # Calculate the confusion matrix
            cm = confusion_matrix(y_true, y_pred)

            # Calculating precision and recall
            precision = precision_score(y_true, y_pred,
average="weighted")
            recall = recall_score(y_true, y_pred, average="weighted")

            # Printing results
            print(f"Results for activation function: {act}")
            print("Confusion Matrix:")
            print(cm)
            print(f"Precision: {precision:.4f}")
            print(f"Recall: {recall:.4f}")

            # Plotting the confusion matrix

```

```

plt.figure(figsize=(10, 7))
sns.heatmap(cm, annot=True, fmt="g", cmap="Blues",
cbar=False)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title(f"Confusion Matrix for {act} Activation")
plt.show()

Training Model with relu activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 38s - loss: 0.1981 - accuracy: 0.9399 - val_loss: 0.0698 -
val_accuracy: 0.9818 - 38s/epoch - 22ms/step
Epoch 2/10
1688/1688 - 32s - loss: 0.0676 - accuracy: 0.9792 - val_loss: 0.0582 -
val_accuracy: 0.9838 - 32s/epoch - 19ms/step
Epoch 3/10
1688/1688 - 32s - loss: 0.0455 - accuracy: 0.9862 - val_loss: 0.0569 -
val_accuracy: 0.9858 - 32s/epoch - 19ms/step
Epoch 4/10
1688/1688 - 33s - loss: 0.0320 - accuracy: 0.9903 - val_loss: 0.0505 -
val_accuracy: 0.9867 - 33s/epoch - 20ms/step
Epoch 5/10
1688/1688 - 32s - loss: 0.0233 - accuracy: 0.9930 - val_loss: 0.0520 -
val_accuracy: 0.9872 - 32s/epoch - 19ms/step
Epoch 6/10
1688/1688 - 32s - loss: 0.0188 - accuracy: 0.9941 - val_loss: 0.0490 -
val_accuracy: 0.9878 - 32s/epoch - 19ms/step
Epoch 7/10
1688/1688 - 34s - loss: 0.0128 - accuracy: 0.9961 - val_loss: 0.0516 -
val_accuracy: 0.9870 - 34s/epoch - 20ms/step
Epoch 8/10
1688/1688 - 32s - loss: 0.0094 - accuracy: 0.9975 - val_loss: 0.0511 -
val_accuracy: 0.9875 - 32s/epoch - 19ms/step
Epoch 9/10
1688/1688 - 32s - loss: 0.0059 - accuracy: 0.9988 - val_loss: 0.0561 -
val_accuracy: 0.9877 - 32s/epoch - 19ms/step
Epoch 10/10
1688/1688 - 35s - loss: 0.0039 - accuracy: 0.9993 - val_loss: 0.0576 -
val_accuracy: 0.9878 - 35s/epoch - 21ms/step
313/313 [=====] - 2s 8ms/step
Results for activation function: relu
Confusion Matrix:
[[ 976   0   0   0   0   0   1   0   2   1]
 [  1 1131   2   0   0   1   0   0   0   0]
 [  1   1 1021   0   2   0   0   6   1   0]
 [  0   0   3 998   0   4   0   3   2   0]
 [  0   0   1   0 975   0   0   0   0   6]
 [  1   0   1   5   0 880   4   0   0   1]
 [  7   3   0   1   1   3 942   0   1   0]
 [  0   1   6   0   0   0   0 1014   2   5]]
```

```
[ 7 1 2 1 1 0 0 4 952 6]
[ 2 0 0 3 8 1 0 3 1 991]]
```

Precision: 0.9880

Recall: 0.9880

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	976	0	0	0	0	0	1	0	2	1
0	976	0	0	0	0	0	1	0	2	1
1	1	1131	2	0	0	1	0	0	0	0
2	1	1	1021	0	2	0	0	6	1	0
3	0	0	3	998	0	4	0	3	2	0
4	0	0	1	0	975	0	0	0	0	6
5	1	0	1	5	0	880	4	0	0	1
6	7	3	0	1	1	3	942	0	1	0
7	0	1	6	0	0	0	0	1014	2	5
8	7	1	2	1	1	0	0	4	952	6
9	2	0	0	3	8	1	0	3	1	991
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

Training Model with relu activation, 1 conv_layers, 1 dense layers..

Epoch 1/10

1688/1688 - 40s - loss: 0.1752 - accuracy: 0.9461 - val_loss: 0.0567 - val_accuracy: 0.9842 - 40s/epoch - 24ms/step

Epoch 2/10

1688/1688 - 39s - loss: 0.0583 - accuracy: 0.9824 - val_loss: 0.0484 - val_accuracy: 0.9877 - 39s/epoch - 23ms/step

Epoch 3/10

1688/1688 - 38s - loss: 0.0356 - accuracy: 0.9893 - val_loss: 0.0420 - val_accuracy: 0.9882 - 38s/epoch - 23ms/step

Epoch 4/10

1688/1688 - 37s - loss: 0.0240 - accuracy: 0.9927 - val_loss: 0.0459 - val_accuracy: 0.9877 - 37s/epoch - 22ms/step

Epoch 5/10

1688/1688 - 38s - loss: 0.0164 - accuracy: 0.9953 - val_loss: 0.0438 - val_accuracy: 0.9885 - 38s/epoch - 22ms/step

```
Epoch 6/10
1688/1688 - 40s - loss: 0.0108 - accuracy: 0.9971 - val_loss: 0.0466 -
val_accuracy: 0.9888 - 40s/epoch - 23ms/step
Epoch 7/10
1688/1688 - 37s - loss: 0.0081 - accuracy: 0.9976 - val_loss: 0.0453 -
val_accuracy: 0.9893 - 37s/epoch - 22ms/step
Epoch 8/10
1688/1688 - 36s - loss: 0.0044 - accuracy: 0.9992 - val_loss: 0.0482 -
val_accuracy: 0.9877 - 36s/epoch - 21ms/step
Epoch 9/10
1688/1688 - 38s - loss: 0.0029 - accuracy: 0.9996 - val_loss: 0.0451 -
val_accuracy: 0.9892 - 38s/epoch - 22ms/step
Epoch 10/10
1688/1688 - 38s - loss: 0.0019 - accuracy: 0.9998 - val_loss: 0.0457 -
val_accuracy: 0.9900 - 38s/epoch - 22ms/step
313/313 [=====] - 3s 8ms/step
Results for activation function: relu
Confusion Matrix:
[[ 976   0   0   0   0   1   1   1   1   0]
 [  0 1130   1   1   0   0   1   0   2   0]
 [  0   2 1020   0   1   0   1   6   2   0]
 [  0   0   1 1004   0   1   0   1   2   1]
 [  0   0   1   0  974   0   1   0   0   6]
 [  1   0   0   6   0  880   4   0   1   0]
 [  5   3   0   1   2   2  944   0   1   0]
 [  0   2   6   2   0   0   0 1013   2   3]
 [  2   0   2   3   0   2   0   1  960   4]
 [  1   1   0   5   6   2   0   4   1  989]]
```

Precision: 0.9890
Recall: 0.9890

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	976	0	0	0	0	1	1	1	1	0
0	976	0	0	0	0	1	1	1	1	0
1	0	1130	1	1	0	0	1	0	2	0
2	0	2	1020	0	1	0	1	6	2	0
3	0	0	1	1004	0	1	0	1	2	1
4	0	0	1	0	974	0	1	0	0	6
5	1	0	0	6	0	880	4	0	1	0
6	5	3	0	1	2	2	944	0	1	0
7	0	2	6	2	0	0	0	1013	2	3
8	2	0	2	3	0	2	0	1	960	4
9	1	1	0	5	6	2	0	4	1	989
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

Training Model with relu activation, 1 conv_layers, 1 dense layers..

Epoch 1/10

1688/1688 - 54s - loss: 0.1689 - accuracy: 0.9487 - val_loss: 0.0743 - val_accuracy: 0.9782 - 54s/epoch - 32ms/step

Epoch 2/10

1688/1688 - 52s - loss: 0.0571 - accuracy: 0.9830 - val_loss: 0.0500 - val_accuracy: 0.9870 - 52s/epoch - 31ms/step

Epoch 3/10

1688/1688 - 53s - loss: 0.0339 - accuracy: 0.9897 - val_loss: 0.0474 - val_accuracy: 0.9887 - 53s/epoch - 31ms/step

Epoch 4/10

1688/1688 - 52s - loss: 0.0219 - accuracy: 0.9936 - val_loss: 0.0431 - val_accuracy: 0.9883 - 52s/epoch - 31ms/step

Epoch 5/10

1688/1688 - 53s - loss: 0.0137 - accuracy: 0.9960 - val_loss: 0.0443 - val_accuracy: 0.9883 - 53s/epoch - 31ms/step

Epoch 6/10

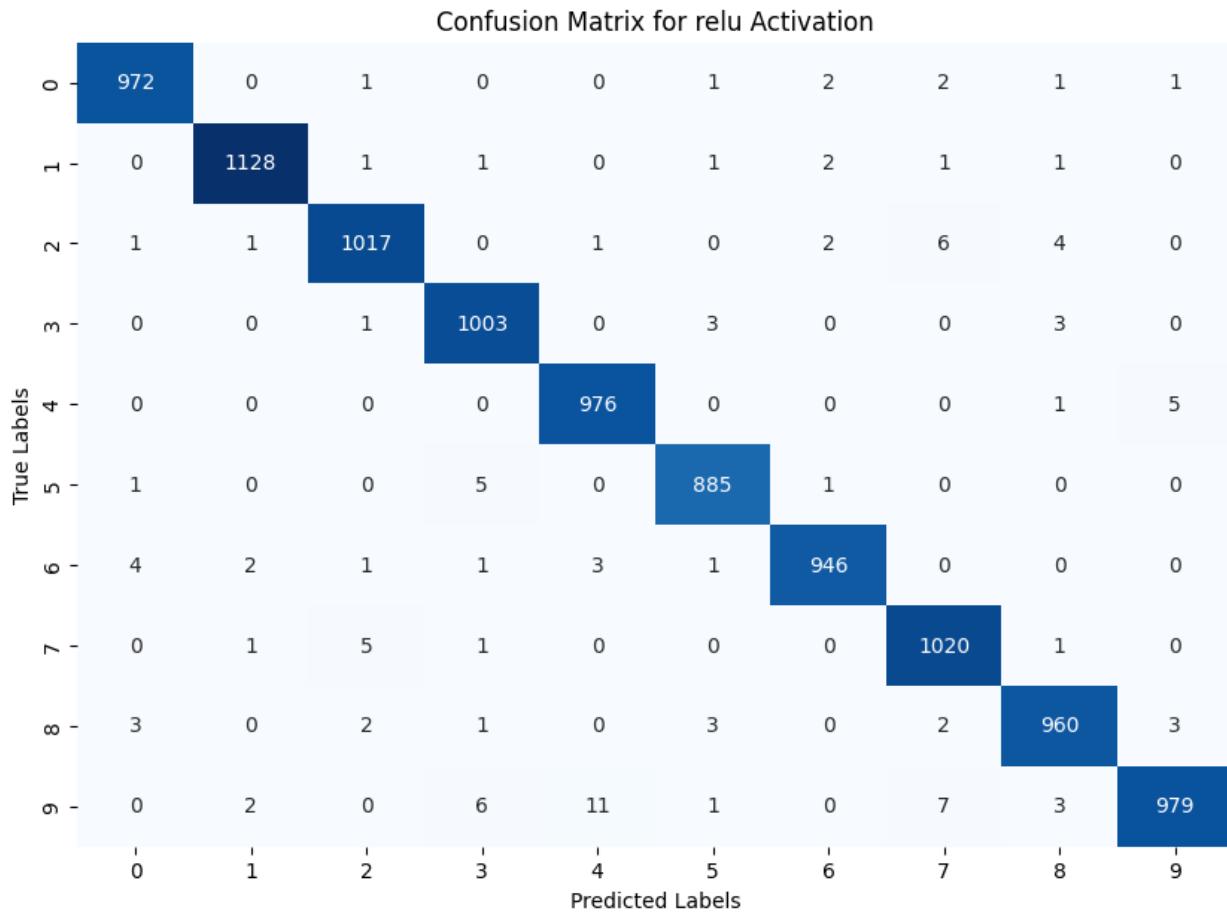
1688/1688 - 53s - loss: 0.0094 - accuracy: 0.9974 - val_loss: 0.0425 - val_accuracy: 0.9895 - 53s/epoch - 31ms/step

Epoch 7/10

1688/1688 - 53s - loss: 0.0062 - accuracy: 0.9986 - val_loss: 0.0432 -

```
val_accuracy: 0.9907 - 53s/epoch - 31ms/step
Epoch 8/10
1688/1688 - 50s - loss: 0.0040 - accuracy: 0.9992 - val_loss: 0.0426 -
val_accuracy: 0.9907 - 50s/epoch - 29ms/step
Epoch 9/10
1688/1688 - 53s - loss: 0.0025 - accuracy: 0.9996 - val_loss: 0.0438 -
val_accuracy: 0.9905 - 53s/epoch - 31ms/step
Epoch 10/10
1688/1688 - 52s - loss: 0.0014 - accuracy: 0.9998 - val_loss: 0.0484 -
val_accuracy: 0.9903 - 52s/epoch - 31ms/step
313/313 [=====] - 5s 15ms/step
Results for activation function: relu
Confusion Matrix:
[[ 972    0    1    0    0    1    2    2    1    1]
 [  0 1128    1    1    0    1    2    1    1    0]
 [  1    1 1017    0    1    0    2    6    4    0]
 [  0    0    1 1003    0    3    0    0    3    0]
 [  0    0    0    0  976    0    0    0    1    5]
 [  1    0    0    5    0  885    1    0    0    0]
 [  4    2    1    1    3    1  946    0    0    0]
 [  0    1    5    1    0    0    0 1020    1    0]
 [  3    0    2    1    0    3    0    2  960    3]
 [  0    2    0    6   11    1    0    7    3  979]]
```

Precision: 0.9886
Recall: 0.9886



```
Training Model with relu activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 47s - loss: 0.1576 - accuracy: 0.9510 - val_loss: 0.0809 -
val_accuracy: 0.9782 - 47s/epoch - 28ms/step
Epoch 2/10
1688/1688 - 47s - loss: 0.0556 - accuracy: 0.9823 - val_loss: 0.0533 -
val_accuracy: 0.9855 - 47s/epoch - 28ms/step
Epoch 3/10
1688/1688 - 46s - loss: 0.0383 - accuracy: 0.9880 - val_loss: 0.0414 -
val_accuracy: 0.9878 - 46s/epoch - 27ms/step
Epoch 4/10
1688/1688 - 46s - loss: 0.0279 - accuracy: 0.9911 - val_loss: 0.0377 -
val_accuracy: 0.9900 - 46s/epoch - 28ms/step
Epoch 5/10
1688/1688 - 46s - loss: 0.0240 - accuracy: 0.9923 - val_loss: 0.0442 -
val_accuracy: 0.9883 - 46s/epoch - 27ms/step
Epoch 6/10
1688/1688 - 46s - loss: 0.0166 - accuracy: 0.9948 - val_loss: 0.0405 -
val_accuracy: 0.9895 - 46s/epoch - 27ms/step
Epoch 7/10
1688/1688 - 47s - loss: 0.0141 - accuracy: 0.9952 - val_loss: 0.0495 -
```

```
val_accuracy: 0.9885 - 47s/epoch - 28ms/step
Epoch 8/10
1688/1688 - 46s - loss: 0.0126 - accuracy: 0.9958 - val_loss: 0.0429 -
val_accuracy: 0.9898 - 46s/epoch - 27ms/step
Epoch 9/10
1688/1688 - 48s - loss: 0.0087 - accuracy: 0.9972 - val_loss: 0.0468 -
val_accuracy: 0.9890 - 48s/epoch - 28ms/step
Epoch 10/10
1688/1688 - 48s - loss: 0.0078 - accuracy: 0.9976 - val_loss: 0.0547 -
val_accuracy: 0.9888 - 48s/epoch - 29ms/step
313/313 [=====] - 3s 9ms/step
Results for activation function: relu
Confusion Matrix:
[[ 967   0   1   0   1   3   5   1   1   1]
 [  0 1125   1   3   0   0   4   2   0   0]
 [  1   1 1021   0   0   0   2   7   0   0]
 [  0   0   2 1005   0   3   0   0   0   0]
 [  0   0   0   0  969   0   1   0   0  12]
 [  0   0   1   5   0  884   1   0   0   1]
 [  2   1   0   0   1   4  949   0   1   0]
 [  0   2   0   1   0   1   0 1020   0   4]
 [  1   0   2   2   0   3   3   0  961   2]
 [  0   0   0   2   2   2   0   0   3 1000]]]
Precision: 0.9901
Recall: 0.9901
```

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	967	0	1	0	1	3	5	1	1	1
0	967	0	1	0	1	3	5	1	1	1
1	0	1125	1	3	0	0	4	2	0	0
2	1	1	1021	0	0	0	2	7	0	0
3	0	0	2	1005	0	3	0	0	0	0
4	0	0	0	0	969	0	1	0	0	12
5	0	0	1	5	0	884	1	0	0	1
6	2	1	0	0	1	4	949	0	1	0
7	0	2	0	1	0	1	0	1020	0	4
8	1	0	2	2	0	3	3	0	961	2
9	0	0	0	2	2	2	0	0	3	1000
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

Training Model with relu activation, 2 conv_layers, 2 dense layers..

Epoch 1/10

1688/1688 - 50s - loss: 0.1434 - accuracy: 0.9548 - val_loss: 0.0601 - val_accuracy: 0.9812 - 50s/epoch - 30ms/step

Epoch 2/10

1688/1688 - 48s - loss: 0.0488 - accuracy: 0.9844 - val_loss: 0.0458 - val_accuracy: 0.9865 - 48s/epoch - 28ms/step

Epoch 3/10

1688/1688 - 48s - loss: 0.0326 - accuracy: 0.9894 - val_loss: 0.0405 - val_accuracy: 0.9885 - 48s/epoch - 28ms/step

Epoch 4/10

1688/1688 - 48s - loss: 0.0235 - accuracy: 0.9928 - val_loss: 0.0380 - val_accuracy: 0.9895 - 48s/epoch - 28ms/step

Epoch 5/10

1688/1688 - 48s - loss: 0.0170 - accuracy: 0.9946 - val_loss: 0.0433 - val_accuracy: 0.9890 - 48s/epoch - 28ms/step

Epoch 6/10

1688/1688 - 48s - loss: 0.0137 - accuracy: 0.9953 - val_loss: 0.0322 - val_accuracy: 0.9908 - 48s/epoch - 29ms/step

Epoch 7/10

1688/1688 - 48s - loss: 0.0096 - accuracy: 0.9967 - val_loss: 0.0473 -

```
val_accuracy: 0.9895 - 48s/epoch - 28ms/step
Epoch 8/10
1688/1688 - 49s - loss: 0.0085 - accuracy: 0.9974 - val_loss: 0.0313 -
val_accuracy: 0.9925 - 49s/epoch - 29ms/step
Epoch 9/10
1688/1688 - 48s - loss: 0.0058 - accuracy: 0.9981 - val_loss: 0.0412 -
val_accuracy: 0.9913 - 48s/epoch - 28ms/step
Epoch 10/10
1688/1688 - 47s - loss: 0.0058 - accuracy: 0.9981 - val_loss: 0.0336 -
val_accuracy: 0.9922 - 47s/epoch - 28ms/step
313/313 [=====] - 4s 12ms/step
Results for activation function: relu
Confusion Matrix:
[[ 978   0   0   0   0   0   1   1   0   0]
 [  1 1132   0   0   1   0   1   0   0   0]
 [  0   1 1029   1   0   0   0   1   0   0]
 [  0   0   3 1000   0   6   0   0   1   0]
 [  0   0   0   0  978   0   1   0   0   3]
 [  0   0   0   3   0  887   1   1   0   0]
 [  3   2   0   0   2   4  947   0   0   0]
 [  0   3   4   0   0   0   0 1017   0   4]
 [  2   0   4   1   0   0   1   1  963   2]
 [  0   0   0   0   2   4   0   0   1 1002]]
```

Precision: 0.9933
Recall: 0.9933

Confusion Matrix for relu Activation										
	0	1	2	3	4	5	6	7	8	9
True Labels	978	0	0	0	0	0	1	1	0	0
0	978	0	0	0	0	0	1	1	0	0
1	1	1132	0	0	1	0	1	0	0	0
2	0	1	1029	1	0	0	0	1	0	0
3	0	0	3	1000	0	6	0	0	1	0
4	0	0	0	0	978	0	1	0	0	3
5	0	0	0	3	0	887	1	1	0	0
6	3	2	0	0	2	4	947	0	0	0
7	0	3	4	0	0	0	0	1017	0	4
8	2	0	4	1	0	0	1	1	963	2
9	0	0	0	0	2	4	0	0	1	1002
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```

Training Model with relu activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 52s - loss: 0.1321 - accuracy: 0.9575 - val_loss: 0.0456 -
val_accuracy: 0.9863 - 52s/epoch - 31ms/step
Epoch 2/10
1688/1688 - 50s - loss: 0.0457 - accuracy: 0.9858 - val_loss: 0.0415 -
val_accuracy: 0.9875 - 50s/epoch - 30ms/step
Epoch 3/10
1688/1688 - 50s - loss: 0.0309 - accuracy: 0.9902 - val_loss: 0.0364 -
val_accuracy: 0.9903 - 50s/epoch - 29ms/step
Epoch 4/10
1688/1688 - 51s - loss: 0.0222 - accuracy: 0.9931 - val_loss: 0.0362 -
val_accuracy: 0.9910 - 51s/epoch - 30ms/step
Epoch 5/10
1688/1688 - 50s - loss: 0.0165 - accuracy: 0.9945 - val_loss: 0.0360 -
val_accuracy: 0.9917 - 50s/epoch - 30ms/step
Epoch 6/10
1688/1688 - 50s - loss: 0.0107 - accuracy: 0.9967 - val_loss: 0.0404 -
val_accuracy: 0.9910 - 50s/epoch - 30ms/step
Epoch 7/10
1688/1688 - 52s - loss: 0.0083 - accuracy: 0.9973 - val_loss: 0.0390 -

```

```
val_accuracy: 0.9912 - 52s/epoch - 31ms/step
Epoch 8/10
1688/1688 - 52s - loss: 0.0052 - accuracy: 0.9984 - val_loss: 0.0375 -
val_accuracy: 0.9928 - 52s/epoch - 31ms/step
Epoch 9/10
1688/1688 - 51s - loss: 0.0041 - accuracy: 0.9987 - val_loss: 0.0456 -
val_accuracy: 0.9910 - 51s/epoch - 30ms/step
Epoch 10/10
1688/1688 - 50s - loss: 0.0035 - accuracy: 0.9990 - val_loss: 0.0420 -
val_accuracy: 0.9923 - 50s/epoch - 30ms/step
313/313 [=====] - 3s 10ms/step
Results for activation function: relu
Confusion Matrix:
[[ 974    0    0    0    0    1    2    3    0    0]
 [  0 1126    1    4    0    1    3    0    0    0]
 [  0    0 1027    0    1    0    0    4    0    0]
 [  0    0    1 1002    0    5    0    0    1    1]
 [  0    0    0    0  976    0    3    0    0    3]
 [  1    0    0    4    0  885    1    0    0    1]
 [  0    2    0    0    1    2  952    0    1    0]
 [  0    2    6    0    0    0    0 1017    2    1]
 [  1    0    3    0    0    1    0    1  966    2]
 [  0    0    0    0    2    4    0    3    0 1000]]]
Precision: 0.9925
Recall: 0.9925
```

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	974	0	0	0	0	1	2	3	0	0
0	974	0	0	0	0	1	2	3	0	0
1	0	1126	1	4	0	1	3	0	0	0
2	0	0	1027	0	1	0	0	4	0	0
3	0	0	1	1002	0	5	0	0	1	1
4	0	0	0	0	976	0	3	0	0	3
5	1	0	0	4	0	885	1	0	0	1
6	0	2	0	0	1	2	952	0	1	0
7	0	2	6	0	0	0	0	1017	2	1
8	1	0	3	0	0	1	0	1	966	2
9	0	0	0	0	2	4	0	3	0	1000
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

Training Model with relu activation, 3 conv_layers, 3 dense layers..

Epoch 1/10

1688/1688 - 53s - loss: 0.2653 - accuracy: 0.9168 - val_loss: 0.0967 - val_accuracy: 0.9728 - 53s/epoch - 32ms/step

Epoch 2/10

1688/1688 - 49s - loss: 0.1016 - accuracy: 0.9700 - val_loss: 0.0745 - val_accuracy: 0.9795 - 49s/epoch - 29ms/step

Epoch 3/10

1688/1688 - 47s - loss: 0.0760 - accuracy: 0.9772 - val_loss: 0.0622 - val_accuracy: 0.9825 - 47s/epoch - 28ms/step

Epoch 4/10

1688/1688 - 48s - loss: 0.0655 - accuracy: 0.9805 - val_loss: 0.0647 - val_accuracy: 0.9818 - 48s/epoch - 28ms/step

Epoch 5/10

1688/1688 - 47s - loss: 0.0536 - accuracy: 0.9833 - val_loss: 0.0548 - val_accuracy: 0.9838 - 47s/epoch - 28ms/step

Epoch 6/10

1688/1688 - 47s - loss: 0.0486 - accuracy: 0.9851 - val_loss: 0.0488 - val_accuracy: 0.9865 - 47s/epoch - 28ms/step

Epoch 7/10

1688/1688 - 47s - loss: 0.0420 - accuracy: 0.9871 - val_loss: 0.0504 -

```
val_accuracy: 0.9857 - 47s/epoch - 28ms/step
Epoch 8/10
1688/1688 - 48s - loss: 0.0367 - accuracy: 0.9889 - val_loss: 0.0531 -
val_accuracy: 0.9863 - 48s/epoch - 29ms/step
Epoch 9/10
1688/1688 - 47s - loss: 0.0345 - accuracy: 0.9895 - val_loss: 0.0422 -
val_accuracy: 0.9880 - 47s/epoch - 28ms/step
Epoch 10/10
1688/1688 - 49s - loss: 0.0329 - accuracy: 0.9894 - val_loss: 0.0492 -
val_accuracy: 0.9863 - 49s/epoch - 29ms/step
313/313 [=====] - 3s 9ms/step
Results for activation function: relu
Confusion Matrix:
[[ 967   0   0   0   1   1   5   4   1   1]
 [  0 1133   0   0   0   1   1   0   0   0]
 [  2   1 1018   3   0   0   1   7   0   0]
 [  0   0   1 1007   0   1   0   1   0   0]
 [  1   0   0   0  978   0   3   0   0   0]
 [  2   2   0   9   0  871   5   1   2   0]
 [  3   3   0   1   2   1  948   0   0   0]
 [  0   5   19  11   1   1   0  988   1   2]
 [  2   1   3   2   1   0   2   0  960   3]
 [  3   0   0   1   5   3   1   3   5  988]]]
Precision: 0.9859
Recall: 0.9858
```

Confusion Matrix for relu Activation										
	0	1	2	3	4	5	6	7	8	9
0	967	0	0	0	1	1	5	4	1	1
1	0	1133	0	0	0	1	1	0	0	0
2	2	1	1018	3	0	0	1	7	0	0
3	0	0	1	1007	0	1	0	1	0	0
4	1	0	0	0	978	0	3	0	0	0
5	2	2	0	9	0	871	5	1	2	0
6	3	3	0	1	2	1	948	0	0	0
7	0	5	19	11	1	1	0	988	1	2
8	2	1	3	2	1	0	2	0	960	3
9	3	0	0	1	5	3	1	3	5	988
	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```

Training Model with relu activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 50s - loss: 0.2784 - accuracy: 0.9115 - val_loss: 0.0935 -
val_accuracy: 0.9732 - 50s/epoch - 30ms/step
Epoch 2/10
1688/1688 - 47s - loss: 0.1040 - accuracy: 0.9685 - val_loss: 0.0751 -
val_accuracy: 0.9793 - 47s/epoch - 28ms/step
Epoch 3/10
1688/1688 - 49s - loss: 0.0760 - accuracy: 0.9770 - val_loss: 0.0595 -
val_accuracy: 0.9832 - 49s/epoch - 29ms/step
Epoch 4/10
1688/1688 - 47s - loss: 0.0609 - accuracy: 0.9814 - val_loss: 0.0546 -
val_accuracy: 0.9828 - 47s/epoch - 28ms/step
Epoch 5/10
1688/1688 - 50s - loss: 0.0523 - accuracy: 0.9840 - val_loss: 0.0553 -
val_accuracy: 0.9812 - 50s/epoch - 29ms/step
Epoch 6/10
1688/1688 - 47s - loss: 0.0443 - accuracy: 0.9865 - val_loss: 0.0602 -
val_accuracy: 0.9830 - 47s/epoch - 28ms/step
Epoch 7/10
1688/1688 - 49s - loss: 0.0380 - accuracy: 0.9881 - val_loss: 0.0603 -

```

```
val_accuracy: 0.9820 - 49s/epoch - 29ms/step
Epoch 8/10
1688/1688 - 47s - loss: 0.0351 - accuracy: 0.9886 - val_loss: 0.0661 -
val_accuracy: 0.9810 - 47s/epoch - 28ms/step
Epoch 9/10
1688/1688 - 49s - loss: 0.0323 - accuracy: 0.9894 - val_loss: 0.0570 -
val_accuracy: 0.9843 - 49s/epoch - 29ms/step
Epoch 10/10
1688/1688 - 48s - loss: 0.0264 - accuracy: 0.9916 - val_loss: 0.0585 -
val_accuracy: 0.9845 - 48s/epoch - 28ms/step
313/313 [=====] - 4s 11ms/step
Results for activation function: relu
Confusion Matrix:
[[ 969   1   3   0   2   0   2   1   0   2]
 [  1 1127   2   1   0   0   2   1   1   0]
 [  0   0 1025   1   0   0   1   5   0   0]
 [  0   0    6 1003   0   0   0   1   0   0]
 [  1   0    4   0  961   0   6   1   4   5]
 [  4   0    2    7   0  870   6   1   2   0]
 [  2   2    1   0    4   3  945   0   1   0]
 [  1   1   28   2   0   0   0  987   3   6]
 [  1   0    2    2   1   0   1   2  964   1]
 [  1   0    1   1    2   5   0   3   3  993]]
```

Precision: 0.9845
Recall: 0.9844

Confusion Matrix for relu Activation

	0	1	2	3	0	2	0	2	1	0	2
True Labels	969	1	3	0	2	0	2	1	0	2	
0	969	1	3	0	2	0	2	1	0	2	
1	1	1127	2	1	0	0	2	1	1	1	0
2	0	0	1025	1	0	0	1	5	0	0	
3	0	0	6	1003	0	0	0	1	0	0	
4	1	0	4	0	961	0	6	1	4	5	
5	4	0	2	7	0	870	6	1	2	0	
6	2	2	1	0	4	3	945	0	1	0	
7	1	1	28	2	0	0	0	987	3	6	
8	1	0	2	2	1	0	1	2	964	1	
9	1	0	1	1	2	5	0	3	3	993	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

Training Model with relu activation, 3 conv_layers, 3 dense layers..

Epoch 1/10

1688/1688 - 51s - loss: 0.2539 - accuracy: 0.9201 - val_loss: 0.1145 - val_accuracy: 0.9657 - 51s/epoch - 30ms/step

Epoch 2/10

1688/1688 - 51s - loss: 0.0962 - accuracy: 0.9706 - val_loss: 0.0649 - val_accuracy: 0.9797 - 51s/epoch - 30ms/step

Epoch 3/10

1688/1688 - 49s - loss: 0.0690 - accuracy: 0.9788 - val_loss: 0.0552 - val_accuracy: 0.9850 - 49s/epoch - 29ms/step

Epoch 4/10

1688/1688 - 49s - loss: 0.0554 - accuracy: 0.9830 - val_loss: 0.0513 - val_accuracy: 0.9848 - 49s/epoch - 29ms/step

Epoch 5/10

1688/1688 - 51s - loss: 0.0465 - accuracy: 0.9854 - val_loss: 0.0680 - val_accuracy: 0.9798 - 51s/epoch - 30ms/step

Epoch 6/10

1688/1688 - 50s - loss: 0.0390 - accuracy: 0.9880 - val_loss: 0.0631 - val_accuracy: 0.9825 - 50s/epoch - 30ms/step

Epoch 7/10

1688/1688 - 50s - loss: 0.0338 - accuracy: 0.9889 - val_loss: 0.0579 -

```
val_accuracy: 0.9843 - 50s/epoch - 29ms/step
Epoch 8/10
1688/1688 - 51s - loss: 0.0293 - accuracy: 0.9911 - val_loss: 0.0573 -
val_accuracy: 0.9862 - 51s/epoch - 30ms/step
Epoch 9/10
1688/1688 - 50s - loss: 0.0276 - accuracy: 0.9912 - val_loss: 0.0504 -
val_accuracy: 0.9863 - 50s/epoch - 30ms/step
Epoch 10/10
1688/1688 - 49s - loss: 0.0247 - accuracy: 0.9918 - val_loss: 0.0553 -
val_accuracy: 0.9853 - 49s/epoch - 29ms/step
313/313 [=====] - 3s 10ms/step
Results for activation function: relu
Confusion Matrix:
[[ 960     0     2     0     3     0     5     4     1     5]
 [  0 1125     0     1     0     2     2     2     3     0]
 [  1     1 1007     6     1     0     1    10     5     0]
 [  0     0     1 999     0     7     0     2     1     0]
 [  0     0     0     1 974     0     2     3     0     2]
 [  1     0     1     2     0 886     1     1     0     0]
 [  3     2     0     0     2     5 940     0     5     1]
 [  0     3    14     1     0     1     0 1007     1     1]
 [  0     0     2     0     1     2     0     1   966     2]
 [  0     0     0     1     3     8     1     6     8 982]]
```

Precision: 0.9847
Recall: 0.9846

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	960	0	2	0	3	0	5	4	1	5
0	960	0	2	0	3	0	5	4	1	5
1	0	1125	0	1	0	2	2	2	3	0
2	1	1	1007	6	1	0	1	10	5	0
3	0	0	1	999	0	7	0	2	1	0
4	0	0	0	1	974	0	2	3	0	2
5	1	0	1	2	0	886	1	1	0	0
6	3	2	0	0	2	5	940	0	5	1
7	0	3	14	1	0	1	0	1007	1	1
8	0	0	2	0	1	2	0	1	966	2
9	0	0	0	1	3	8	1	6	8	982
0	1	2	3	4	5	6	7	8	9	9
Predicted Labels										

```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 36s - loss: 0.6942 - accuracy: 0.7981 - val_loss: 0.2502 -
val_accuracy: 0.9303 - 36s/epoch - 21ms/step
Epoch 2/10
1688/1688 - 34s - loss: 0.2788 - accuracy: 0.9175 - val_loss: 0.1769 -
val_accuracy: 0.9488 - 34s/epoch - 20ms/step
Epoch 3/10
1688/1688 - 35s - loss: 0.2142 - accuracy: 0.9359 - val_loss: 0.1451 -
val_accuracy: 0.9600 - 35s/epoch - 21ms/step
Epoch 4/10
1688/1688 - 33s - loss: 0.1741 - accuracy: 0.9486 - val_loss: 0.1281 -
val_accuracy: 0.9633 - 33s/epoch - 20ms/step
Epoch 5/10
1688/1688 - 35s - loss: 0.1458 - accuracy: 0.9566 - val_loss: 0.1137 -
val_accuracy: 0.9673 - 35s/epoch - 20ms/step
Epoch 6/10
1688/1688 - 33s - loss: 0.1253 - accuracy: 0.9627 - val_loss: 0.1021 -
val_accuracy: 0.9710 - 33s/epoch - 20ms/step
Epoch 7/10
```

```
1688/1688 - 35s - loss: 0.1097 - accuracy: 0.9683 - val_loss: 0.0879 -  
val_accuracy: 0.9758 - 35s/epoch - 21ms/step  
Epoch 8/10  
1688/1688 - 35s - loss: 0.0974 - accuracy: 0.9710 - val_loss: 0.0791 -  
val_accuracy: 0.9803 - 35s/epoch - 21ms/step  
Epoch 9/10  
1688/1688 - 34s - loss: 0.0867 - accuracy: 0.9753 - val_loss: 0.0812 -  
val_accuracy: 0.9785 - 34s/epoch - 20ms/step  
Epoch 10/10  
1688/1688 - 35s - loss: 0.0781 - accuracy: 0.9779 - val_loss: 0.0793 -  
val_accuracy: 0.9785 - 35s/epoch - 21ms/step  
313/313 [=====] - 3s 8ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 967  0  1  1  0  2  6  1  2  0]  
 [ 0 1127  1  1  0  1  4  0  1  0]  
 [ 5  4 995  9  4  0  3  6  6  0]  
 [ 0  0  0 1001  0  4  0  3  2  0]  
 [ 1  0  1  0 972  0  4  0  2  2]  
 [ 3  0  0  9  0 875  3  0  2  0]  
 [ 5  3  2  1  3  4 936  0  4  0]  
 [ 2  5  11  8  3  0  0 997  0  2]  
 [ 4  0  1 12  3  4  3  3 944  0]  
 [ 4  6  0 14  33 12  1  9  8 922]]  
Precision: 0.9740  
Recall: 0.9736
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	967	0	1	1	0	2	6	1	2	0	
0	0	1127	1	1	0	1	4	0	1	0	
1	5	4	995	9	4	0	3	6	6	0	
2	0	0	0	1001	0	4	0	3	2	0	
3	1	0	1	0	972	0	4	0	2	2	
4	3	0	0	9	0	875	3	0	2	0	
5	5	3	2	1	3	4	936	0	4	0	
6	2	5	11	8	3	0	0	997	0	2	
7	4	0	1	12	3	4	3	3	944	0	
8	4	6	0	14	33	12	1	9	8	922	
9	1	2	3	4	5	6	7	8	9	9	
Predicted Labels											

```
Training Model with sigmoid activation, 1 conv_layers, 1 dense
layers..
Epoch 1/10
1688/1688 - 42s - loss: 0.6547 - accuracy: 0.8011 - val_loss: 0.2732 -
val_accuracy: 0.9192 - 42s/epoch - 25ms/step
Epoch 2/10
1688/1688 - 41s - loss: 0.2847 - accuracy: 0.9126 - val_loss: 0.1837 -
val_accuracy: 0.9482 - 41s/epoch - 24ms/step
Epoch 3/10
1688/1688 - 39s - loss: 0.2164 - accuracy: 0.9349 - val_loss: 0.1504 -
val_accuracy: 0.9577 - 39s/epoch - 23ms/step
Epoch 4/10
1688/1688 - 40s - loss: 0.1731 - accuracy: 0.9481 - val_loss: 0.1368 -
val_accuracy: 0.9615 - 40s/epoch - 24ms/step
Epoch 5/10
1688/1688 - 39s - loss: 0.1456 - accuracy: 0.9566 - val_loss: 0.1085 -
val_accuracy: 0.9685 - 39s/epoch - 23ms/step
Epoch 6/10
1688/1688 - 41s - loss: 0.1233 - accuracy: 0.9634 - val_loss: 0.0961 -
val_accuracy: 0.9752 - 41s/epoch - 24ms/step
Epoch 7/10
```

```
1688/1688 - 41s - loss: 0.1088 - accuracy: 0.9674 - val_loss: 0.0838 -  
val_accuracy: 0.9792 - 41s/epoch - 24ms/step  
Epoch 8/10  
1688/1688 - 39s - loss: 0.0955 - accuracy: 0.9714 - val_loss: 0.0902 -  
val_accuracy: 0.9747 - 39s/epoch - 23ms/step  
Epoch 9/10  
1688/1688 - 40s - loss: 0.0852 - accuracy: 0.9747 - val_loss: 0.0834 -  
val_accuracy: 0.9787 - 40s/epoch - 24ms/step  
Epoch 10/10  
1688/1688 - 39s - loss: 0.0775 - accuracy: 0.9776 - val_loss: 0.0825 -  
val_accuracy: 0.9788 - 39s/epoch - 23ms/step  
313/313 [=====] - 3s 9ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 971  0  0  1  0  1  5  1  0  1]  
 [ 0 1126  2  1  0  1  4  1  0  0]  
 [ 4  3 1011  3  2  0  2  6  0  1]  
 [ 0  0  3 997  0  0  0  6  2  2]  
 [ 0  0  3  0 958  0  7  4  0 10]  
 [ 6  0  1 18  0 854  8  2  1  2]  
 [ 6  3  1  1  2  2 941  0  2  0]  
 [ 1  3 14  3  0  0  0 998  0  9]  
 [ 7  2 11 21  3  3  9  6 907  5]  
 [ 5  4  3 11 10  1  1  9  0 965]]  
Precision: 0.9732  
Recall: 0.9728
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	971	0	0	1	0	1	5	1	0	1
0	971	0	0	1	0	1	5	1	0	1
1	0	1126	2	1	0	1	4	1	0	0
2	4	3	1011	3	2	0	2	6	0	1
3	0	0	3	997	0	0	0	6	2	2
4	0	0	3	0	958	0	7	4	0	10
5	6	0	1	18	0	854	8	2	1	2
6	6	3	1	1	2	2	941	0	2	0
7	1	3	14	3	0	0	0	998	0	9
8	7	2	11	21	3	3	9	6	907	5
9	5	4	3	11	10	1	1	9	0	965
10	1	1	2	3	4	5	6	7	8	9

```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 55s - loss: 0.5990 - accuracy: 0.8146 - val_loss: 0.2289 - val_accuracy: 0.9325 - 55s/epoch - 32ms/step
Epoch 2/10
1688/1688 - 54s - loss: 0.2826 - accuracy: 0.9149 - val_loss: 0.2063 - val_accuracy: 0.9377 - 54s/epoch - 32ms/step
Epoch 3/10
1688/1688 - 54s - loss: 0.2123 - accuracy: 0.9349 - val_loss: 0.1419 - val_accuracy: 0.9605 - 54s/epoch - 32ms/step
Epoch 4/10
1688/1688 - 53s - loss: 0.1668 - accuracy: 0.9500 - val_loss: 0.1239 - val_accuracy: 0.9635 - 53s/epoch - 31ms/step
Epoch 5/10
1688/1688 - 54s - loss: 0.1375 - accuracy: 0.9590 - val_loss: 0.1061 - val_accuracy: 0.9723 - 54s/epoch - 32ms/step
Epoch 6/10
1688/1688 - 53s - loss: 0.1173 - accuracy: 0.9653 - val_loss: 0.1008 - val_accuracy: 0.9713 - 53s/epoch - 31ms/step
Epoch 7/10
```

```
1688/1688 - 54s - loss: 0.1005 - accuracy: 0.9699 - val_loss: 0.0872 -  
val_accuracy: 0.9757 - 54s/epoch - 32ms/step  
Epoch 8/10  
1688/1688 - 54s - loss: 0.0887 - accuracy: 0.9739 - val_loss: 0.0816 -  
val_accuracy: 0.9765 - 54s/epoch - 32ms/step  
Epoch 9/10  
1688/1688 - 53s - loss: 0.0780 - accuracy: 0.9766 - val_loss: 0.0725 -  
val_accuracy: 0.9808 - 53s/epoch - 32ms/step  
Epoch 10/10  
1688/1688 - 54s - loss: 0.0710 - accuracy: 0.9792 - val_loss: 0.0693 -  
val_accuracy: 0.9805 - 54s/epoch - 32ms/step  
313/313 [=====] - 5s 14ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 969 0 0 0 0 1 3 1 3 3]  
[ 0 1125 2 1 0 1 4 1 1 0]  
[ 1 2 1014 4 3 0 2 4 1 1]  
[ 0 0 0 1002 0 1 0 1 2 4]  
[ 0 0 5 0 964 0 1 1 2 9]  
[ 4 0 0 13 0 868 5 0 1 1]  
[ 6 2 1 1 2 5 936 0 5 0]  
[ 1 3 10 11 0 0 0 989 1 13]  
[ 4 0 3 15 3 3 2 1 939 4]  
[ 2 3 1 10 8 4 1 2 1 977]]  
Precision: 0.9785  
Recall: 0.9783
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	969	0	0	0	0	1	3	1	3	3	0
0	1125	2	1	0	1	4	1	1	1	0	1
1	1014	4	3	0	0	2	4	1	1	1	1
2	1002	0	1	0	1	0	0	1	2	4	4
3	964	0	5	0	0	1	1	1	2	9	9
4	868	0	13	0	0	5	0	1	1	1	1
5	936	5	2	5	0	0	0	0	5	0	0
6	989	0	11	0	0	0	0	1	1	13	13
7	939	1	10	0	0	3	2	1	1	4	4
8	977	1	3	15	3	3	2	1	1	1	9
9	9	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 50s - loss: 1.9029 - accuracy: 0.3266 - val_loss: 0.7447 -
val_accuracy: 0.7998 - 50s/epoch - 29ms/step
Epoch 2/10
1688/1688 - 51s - loss: 0.4714 - accuracy: 0.8674 - val_loss: 0.2305 -
val_accuracy: 0.9375 - 51s/epoch - 30ms/step
Epoch 3/10
1688/1688 - 50s - loss: 0.2246 - accuracy: 0.9353 - val_loss: 0.1468 -
val_accuracy: 0.9617 - 50s/epoch - 30ms/step
Epoch 4/10
1688/1688 - 49s - loss: 0.1464 - accuracy: 0.9567 - val_loss: 0.1012 -
val_accuracy: 0.9707 - 49s/epoch - 29ms/step
Epoch 5/10
1688/1688 - 50s - loss: 0.1122 - accuracy: 0.9668 - val_loss: 0.0789 -
val_accuracy: 0.9767 - 50s/epoch - 30ms/step
Epoch 6/10
1688/1688 - 50s - loss: 0.0935 - accuracy: 0.9720 - val_loss: 0.0785 -
val_accuracy: 0.9788 - 50s/epoch - 30ms/step
Epoch 7/10
```

```
1688/1688 - 49s - loss: 0.0806 - accuracy: 0.9752 - val_loss: 0.0680 -  
val_accuracy: 0.9805 - 49s/epoch - 29ms/step  
Epoch 8/10  
1688/1688 - 49s - loss: 0.0718 - accuracy: 0.9782 - val_loss: 0.0609 -  
val_accuracy: 0.9817 - 49s/epoch - 29ms/step  
Epoch 9/10  
1688/1688 - 51s - loss: 0.0652 - accuracy: 0.9802 - val_loss: 0.0533 -  
val_accuracy: 0.9840 - 51s/epoch - 30ms/step  
Epoch 10/10  
1688/1688 - 53s - loss: 0.0593 - accuracy: 0.9822 - val_loss: 0.0535 -  
val_accuracy: 0.9848 - 53s/epoch - 31ms/step  
313/313 [=====] - 3s 11ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 975  0  0  0  0  0  1  1  3  0]  
 [  0 1125  1  1  0  0  2  1  5  0]  
 [  3   3 1019  0  1  0  0  3  3  0]  
 [  1   0   2 992  0  8  0  3  4  0]  
 [  2   0   1  0 966  0  5  0  2  6]  
 [  5   0   1   3  0 870  5  1  4  3]  
 [  4   2   0   0  1  2 946  0  3  0]  
 [  2   2   10  2  0  0  0 1008  2  2]  
 [  6   0   2   1  2  0  2  2 957  2]  
 [  5   4   0   7  8  2  0  5  6 972]]  
Precision: 0.9831  
Recall: 0.9830
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	975	0	0	0	0	0	1	1	3	0
0	0	1125	1	1	0	0	2	1	5	0
1	3	3	1019	0	1	0	0	3	3	0
2	1	0	2	992	0	8	0	3	4	0
3	2	0	1	0	966	0	5	0	2	6
4	5	0	1	3	0	870	5	1	4	3
5	4	2	0	0	1	2	946	0	3	0
6	2	2	10	2	0	0	0	0	1008	2
7	6	0	2	1	2	0	2	2	957	2
8	5	4	0	7	8	2	0	5	6	972
9	1	2	3	4	5	6	7	8	9	9
Predicted Labels										

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 53s - loss: 1.9540 - accuracy: 0.2978 - val_loss: 0.6816 - val_accuracy: 0.8165 - 53s/epoch - 31ms/step
Epoch 2/10
1688/1688 - 51s - loss: 0.4305 - accuracy: 0.8719 - val_loss: 0.2172 - val_accuracy: 0.9407 - 51s/epoch - 30ms/step
Epoch 3/10
1688/1688 - 50s - loss: 0.2153 - accuracy: 0.9365 - val_loss: 0.1269 - val_accuracy: 0.9637 - 50s/epoch - 30ms/step
Epoch 4/10
1688/1688 - 50s - loss: 0.1452 - accuracy: 0.9564 - val_loss: 0.0987 - val_accuracy: 0.9723 - 50s/epoch - 29ms/step
Epoch 5/10
1688/1688 - 52s - loss: 0.1123 - accuracy: 0.9666 - val_loss: 0.0849 - val_accuracy: 0.9743 - 52s/epoch - 31ms/step
Epoch 6/10
1688/1688 - 52s - loss: 0.0958 - accuracy: 0.9706 - val_loss: 0.0749 - val_accuracy: 0.9788 - 52s/epoch - 31ms/step
Epoch 7/10
```

```
1688/1688 - 51s - loss: 0.0842 - accuracy: 0.9747 - val_loss: 0.0821 -  
val_accuracy: 0.9760 - 51s/epoch - 30ms/step  
Epoch 8/10  
1688/1688 - 50s - loss: 0.0748 - accuracy: 0.9770 - val_loss: 0.0616 -  
val_accuracy: 0.9825 - 50s/epoch - 30ms/step  
Epoch 9/10  
1688/1688 - 51s - loss: 0.0681 - accuracy: 0.9791 - val_loss: 0.0585 -  
val_accuracy: 0.9828 - 51s/epoch - 30ms/step  
Epoch 10/10  
1688/1688 - 50s - loss: 0.0628 - accuracy: 0.9808 - val_loss: 0.0566 -  
val_accuracy: 0.9837 - 50s/epoch - 30ms/step  
313/313 [=====] - 3s 11ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 966  0  4  0  0  1  2  3  3  1]  
 [ 0 1127  1  2  0  1  1  1  2  0]  
 [ 0  2 1023  0  1  0  0  5  1  0]  
 [ 0  0  1 997  0  5  0  5  2  0]  
 [ 0  0  2  0 970  0  1  3  2  4]  
 [ 1  0  0  9  0 876  2  1  1  2]  
 [ 2  3  1  0  5  5 941  0  1  0]  
 [ 0  1  9  3  0  0  0 1014  0  1]  
 [ 3  0  4  6  4  1  0  3 950  3]  
 [ 1  3  1 10  10  2  0 12  2 968]]  
Precision: 0.9833  
Recall: 0.9832
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	966	0	4	0	0	1	2	3	3	1	0
0	0	1127	1	2	0	1	1	1	2	0	0
1	0	2	1023	0	1	0	0	5	1	0	0
2	0	0	1	997	0	5	0	5	2	0	0
3	0	0	2	0	970	0	1	3	2	4	0
4	1	0	0	9	0	876	2	1	1	2	0
5	2	3	1	0	5	5	941	0	1	0	0
6	0	1	9	3	0	0	0	0	1014	0	1
7	3	0	4	6	4	1	0	3	950	3	0
8	1	3	1	10	10	2	0	12	2	968	0
9	1	2	3	4	4	5	6	7	8	9	0

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 56s - loss: 1.8669 - accuracy: 0.3331 - val_loss: 0.5325 -
val_accuracy: 0.8427 - 56s/epoch - 33ms/step
Epoch 2/10
1688/1688 - 53s - loss: 0.3621 - accuracy: 0.8926 - val_loss: 0.2075 -
val_accuracy: 0.9395 - 53s/epoch - 31ms/step
Epoch 3/10
1688/1688 - 54s - loss: 0.1953 - accuracy: 0.9416 - val_loss: 0.1224 -
val_accuracy: 0.9637 - 54s/epoch - 32ms/step
Epoch 4/10
1688/1688 - 53s - loss: 0.1373 - accuracy: 0.9575 - val_loss: 0.0952 -
val_accuracy: 0.9743 - 53s/epoch - 32ms/step
Epoch 5/10
1688/1688 - 55s - loss: 0.1091 - accuracy: 0.9670 - val_loss: 0.0787 -
val_accuracy: 0.9773 - 55s/epoch - 32ms/step
Epoch 6/10
1688/1688 - 54s - loss: 0.0944 - accuracy: 0.9717 - val_loss: 0.0811 -
val_accuracy: 0.9765 - 54s/epoch - 32ms/step
Epoch 7/10
```

```
1688/1688 - 55s - loss: 0.0822 - accuracy: 0.9749 - val_loss: 0.0674 -  
val_accuracy: 0.9795 - 55s/epoch - 33ms/step  
Epoch 8/10  
1688/1688 - 54s - loss: 0.0744 - accuracy: 0.9769 - val_loss: 0.0606 -  
val_accuracy: 0.9835 - 54s/epoch - 32ms/step  
Epoch 9/10  
1688/1688 - 54s - loss: 0.0681 - accuracy: 0.9792 - val_loss: 0.0586 -  
val_accuracy: 0.9820 - 54s/epoch - 32ms/step  
Epoch 10/10  
1688/1688 - 55s - loss: 0.0618 - accuracy: 0.9809 - val_loss: 0.0608 -  
val_accuracy: 0.9818 - 55s/epoch - 32ms/step  
313/313 [=====] - 4s 11ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 975  0  0  0  0  0  4  1  0  0]  
 [ 0 1125  2  1  0  0  5  0  2  0]  
 [ 3  4 1021  0  2  0  0  2  0  0]  
 [ 2  0  3  982  0  15  0  4  4  0]  
 [ 0  0  0  0  977  0  3  1  1  0]  
 [ 1  0  0  2  0  883  5  1  0  0]  
 [ 5  2  0  0  2  5  944  0  0  0]  
 [ 2  7  16  2  0  0  0  994  2  5]  
 [ 8  0  3  1  3  3  3  2  949  2]  
 [ 4  7  1  3  11  9  0  4  2  968]]  
Precision: 0.9819  
Recall: 0.9818
```

Confusion Matrix for sigmoid Activation

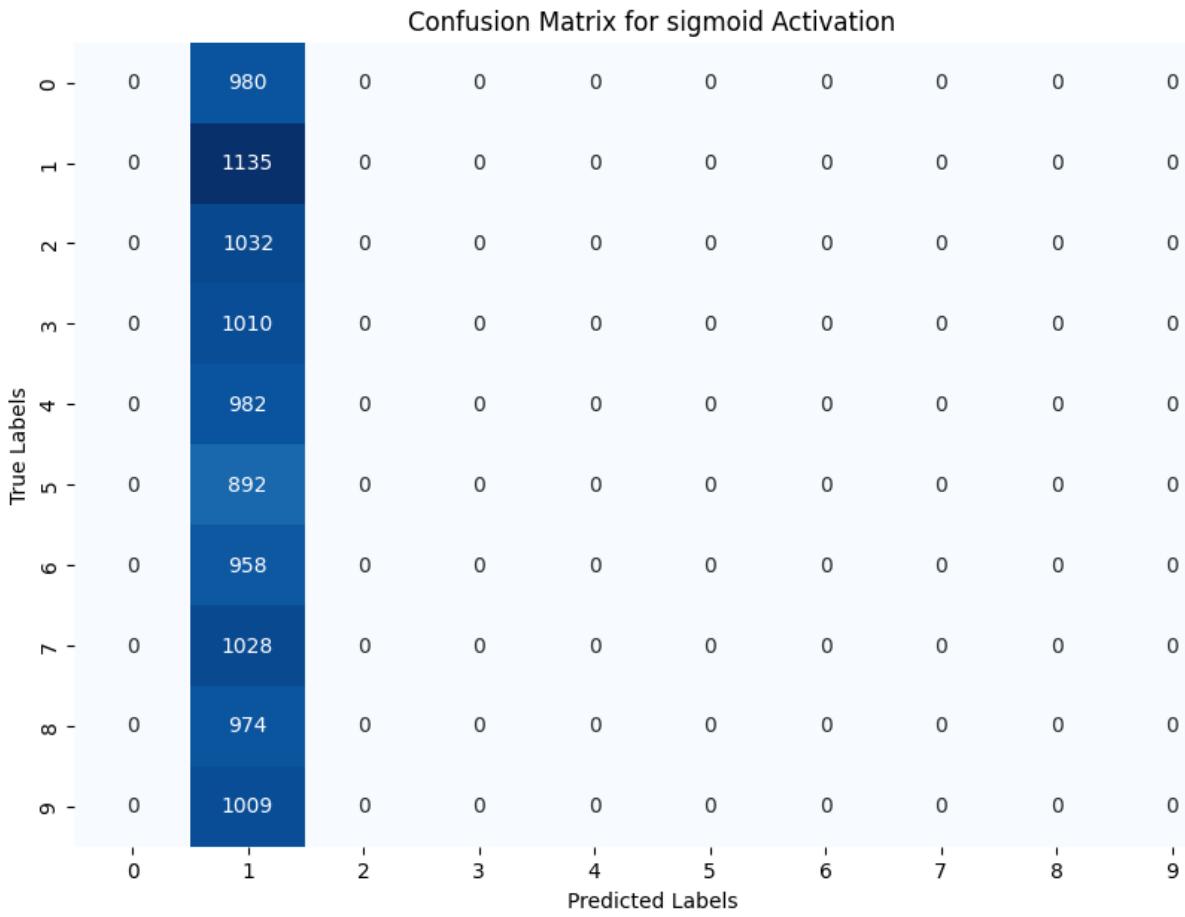
	0	1	2	3	4	5	6	7	8	9
True Labels	975	0	0	0	0	0	4	1	0	0
0	1125	2	1	0	0	5	0	2	0	0
1	3	4	1021	0	2	0	0	2	0	0
2	2	0	3	982	0	15	0	4	4	0
3	0	0	0	0	977	0	3	1	1	0
4	1	0	0	2	0	883	5	1	0	0
5	5	2	0	0	2	5	944	0	0	0
6	2	7	16	2	0	0	0	994	2	5
7	8	0	3	1	3	3	3	2	949	2
8	4	7	1	3	11	9	0	4	2	968
9	1	2	3	4	5	6	7	8	9	9

```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 53s - loss: 2.3140 - accuracy: 0.1034 - val_loss: 2.3101 -
val_accuracy: 0.1002 - 53s/epoch - 31ms/step
Epoch 2/10
1688/1688 - 51s - loss: 2.3084 - accuracy: 0.1084 - val_loss: 2.3078 -
val_accuracy: 0.0992 - 51s/epoch - 30ms/step
Epoch 3/10
1688/1688 - 50s - loss: 2.3054 - accuracy: 0.1054 - val_loss: 2.3050 -
val_accuracy: 0.0960 - 50s/epoch - 30ms/step
Epoch 4/10
1688/1688 - 52s - loss: 2.3042 - accuracy: 0.1082 - val_loss: 2.3064 -
val_accuracy: 0.1050 - 52s/epoch - 31ms/step
Epoch 5/10
1688/1688 - 51s - loss: 2.3032 - accuracy: 0.1080 - val_loss: 2.3050 -
val_accuracy: 0.1050 - 51s/epoch - 30ms/step
Epoch 6/10
1688/1688 - 50s - loss: 2.3028 - accuracy: 0.1107 - val_loss: 2.3025 -
val_accuracy: 0.1000 - 50s/epoch - 29ms/step
Epoch 7/10
```

```
1688/1688 - 50s - loss: 2.3023 - accuracy: 0.1105 - val_loss: 2.3050 -  
val_accuracy: 0.1050 - 50s/epoch - 30ms/step  
Epoch 8/10  
1688/1688 - 52s - loss: 2.3023 - accuracy: 0.1104 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 52s/epoch - 31ms/step  
Epoch 9/10  
1688/1688 - 50s - loss: 2.3022 - accuracy: 0.1106 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 50s/epoch - 30ms/step  
Epoch 10/10  
1688/1688 - 50s - loss: 2.3022 - accuracy: 0.1121 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 50s/epoch - 29ms/step  
313/313 [=====] - 3s 11ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```



```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 52s - loss: 2.3200 - accuracy: 0.1025 - val_loss: 2.3169 - val_accuracy: 0.1000 - 52s/epoch - 31ms/step
Epoch 2/10
1688/1688 - 52s - loss: 2.3101 - accuracy: 0.1030 - val_loss: 2.3055 - val_accuracy: 0.1050 - 52s/epoch - 31ms/step
Epoch 3/10
1688/1688 - 51s - loss: 2.3053 - accuracy: 0.1070 - val_loss: 2.3062 - val_accuracy: 0.1000 - 51s/epoch - 30ms/step
Epoch 4/10
1688/1688 - 50s - loss: 2.3041 - accuracy: 0.1066 - val_loss: 2.3049 - val_accuracy: 0.0995 - 50s/epoch - 30ms/step
Epoch 5/10
1688/1688 - 52s - loss: 2.3028 - accuracy: 0.1107 - val_loss: 2.3021 - val_accuracy: 0.1045 - 52s/epoch - 31ms/step
Epoch 6/10
1688/1688 - 51s - loss: 2.3027 - accuracy: 0.1112 - val_loss: 2.3015 - val_accuracy: 0.1050 - 51s/epoch - 30ms/step
Epoch 7/10
```

```
1688/1688 - 52s - loss: 2.3023 - accuracy: 0.1101 - val_loss: 2.3038 -  
val_accuracy: 0.1050 - 52s/epoch - 31ms/step  
Epoch 8/10  
1688/1688 - 49s - loss: 2.3017 - accuracy: 0.1139 - val_loss: 2.3041 -  
val_accuracy: 0.0978 - 49s/epoch - 29ms/step  
Epoch 9/10  
1688/1688 - 53s - loss: 2.3010 - accuracy: 0.1142 - val_loss: 2.2995 -  
val_accuracy: 0.1052 - 53s/epoch - 31ms/step  
Epoch 10/10  
1688/1688 - 53s - loss: 2.2950 - accuracy: 0.1319 - val_loss: 2.2762 -  
val_accuracy: 0.1767 - 53s/epoch - 31ms/step  
313/313 [=====] - 4s 14ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0  7  0 973  0  0  0  0  0  0]  
 [ 0 1123  0  12  0  0  0  0  0  0]  
 [ 0  337  0 695  0  0  0  0  0  0]  
 [ 0  302  0 708  0  0  0  0  0  0]  
 [ 0  17  0 965  0  0  0  0  0  0]  
 [ 0  98  0 794  0  0  0  0  0  0]  
 [ 0  51  0 907  0  0  0  0  0  0]  
 [ 0  295  0 733  0  0  0  0  0  0]  
 [ 0  94  0 880  0  0  0  0  0  0]  
 [ 0  32  0 977  0  0  0  0  0  0]]  
Precision: 0.0635  
Recall: 0.1831
```

Confusion Matrix for sigmoid Activation										
	0	1	2	3	4	5	6	7	8	9
True Labels	0	0	7	0	973	0	0	0	0	0
0	0	0	1123	0	12	0	0	0	0	0
1	0	0	337	0	695	0	0	0	0	0
2	0	0	302	0	708	0	0	0	0	0
3	0	0	17	0	965	0	0	0	0	0
4	0	0	98	0	794	0	0	0	0	0
5	0	0	51	0	907	0	0	0	0	0
6	0	0	295	0	733	0	0	0	0	0
7	0	0	94	0	880	0	0	0	0	0
8	0	0	32	0	977	0	0	0	0	0
9	0	1	2	3	4	5	6	7	8	9

```

Training Model with sigmoid activation, 3 conv_layers, 3 dense
layers..
Epoch 1/10
1688/1688 - 55s - loss: 2.3293 - accuracy: 0.1012 - val_loss: 2.3192 -
val_accuracy: 0.0978 - 55s/epoch - 33ms/step
Epoch 2/10
1688/1688 - 54s - loss: 2.3076 - accuracy: 0.1041 - val_loss: 2.3054 -
val_accuracy: 0.0960 - 54s/epoch - 32ms/step
Epoch 3/10
1688/1688 - 57s - loss: 2.3041 - accuracy: 0.1064 - val_loss: 2.3030 -
val_accuracy: 0.0952 - 57s/epoch - 34ms/step
Epoch 4/10
1688/1688 - 54s - loss: 2.3029 - accuracy: 0.1095 - val_loss: 2.3028 -
val_accuracy: 0.1050 - 54s/epoch - 32ms/step
Epoch 5/10
1688/1688 - 54s - loss: 2.3028 - accuracy: 0.1090 - val_loss: 2.3040 -
val_accuracy: 0.1050 - 54s/epoch - 32ms/step
Epoch 6/10
1688/1688 - 53s - loss: 2.3024 - accuracy: 0.1106 - val_loss: 2.3033 -
val_accuracy: 0.1050 - 53s/epoch - 31ms/step
Epoch 7/10

```

```
1688/1688 - 53s - loss: 2.3025 - accuracy: 0.1106 - val_loss: 2.3015 -  
val_accuracy: 0.1113 - 53s/epoch - 32ms/step  
Epoch 8/10  
1688/1688 - 53s - loss: 2.3021 - accuracy: 0.1112 - val_loss: 2.3041 -  
val_accuracy: 0.1050 - 53s/epoch - 31ms/step  
Epoch 9/10  
1688/1688 - 54s - loss: 2.3021 - accuracy: 0.1116 - val_loss: 2.3040 -  
val_accuracy: 0.1050 - 54s/epoch - 32ms/step  
Epoch 10/10  
1688/1688 - 55s - loss: 2.3023 - accuracy: 0.1114 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 55s/epoch - 32ms/step  
313/313 [=====] - 4s 13ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]  
 [ 0 1135 0 0 0 0 0 0 0 ]  
 [ 0 1032 0 0 0 0 0 0 0 ]  
 [ 0 1010 0 0 0 0 0 0 0 ]  
 [ 0 982 0 0 0 0 0 0 0 ]  
 [ 0 892 0 0 0 0 0 0 0 ]  
 [ 0 958 0 0 0 0 0 0 0 ]  
 [ 0 1028 0 0 0 0 0 0 0 ]  
 [ 0 974 0 0 0 0 0 0 0 ]  
 [ 0 1009 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135
```

Confusion Matrix for sigmoid Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	980	0	0	0	0	0	0	0	0	0
	1135	0	0	0	0	0	0	0	0	0
	1032	0	0	0	0	0	0	0	0	0
	1010	0	0	0	0	0	0	0	0	0
	982	0	0	0	0	0	0	0	0	0
	892	0	0	0	0	0	0	0	0	0
	958	0	0	0	0	0	0	0	0	0
	1028	0	0	0	0	0	0	0	0	0
	974	0	0	0	0	0	0	0	0	0
	1009	0	0	0	0	0	0	0	0	0

```

Training Model with tanh activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 34s - loss: 0.2028 - accuracy: 0.9402 - val_loss: 0.0897 -
val_accuracy: 0.9755 - 34s/epoch - 20ms/step
Epoch 2/10
1688/1688 - 37s - loss: 0.0774 - accuracy: 0.9780 - val_loss: 0.0682 -
val_accuracy: 0.9805 - 37s/epoch - 22ms/step
Epoch 3/10
1688/1688 - 35s - loss: 0.0480 - accuracy: 0.9864 - val_loss: 0.0564 -
val_accuracy: 0.9850 - 35s/epoch - 20ms/step
Epoch 4/10
1688/1688 - 35s - loss: 0.0330 - accuracy: 0.9912 - val_loss: 0.0495 -
val_accuracy: 0.9850 - 35s/epoch - 21ms/step
Epoch 5/10
1688/1688 - 34s - loss: 0.0224 - accuracy: 0.9946 - val_loss: 0.0452 -
val_accuracy: 0.9872 - 34s/epoch - 20ms/step
Epoch 6/10
1688/1688 - 34s - loss: 0.0153 - accuracy: 0.9970 - val_loss: 0.0467 -
val_accuracy: 0.9867 - 34s/epoch - 20ms/step
Epoch 7/10
1688/1688 - 35s - loss: 0.0109 - accuracy: 0.9984 - val_loss: 0.0470 -

```

```
val_accuracy: 0.9857 - 35s/epoch - 21ms/step
Epoch 8/10
1688/1688 - 33s - loss: 0.0076 - accuracy: 0.9989 - val_loss: 0.0434 -
val_accuracy: 0.9878 - 33s/epoch - 20ms/step
Epoch 9/10
1688/1688 - 36s - loss: 0.0057 - accuracy: 0.9995 - val_loss: 0.0434 -
val_accuracy: 0.9882 - 36s/epoch - 21ms/step
Epoch 10/10
1688/1688 - 33s - loss: 0.0044 - accuracy: 0.9998 - val_loss: 0.0417 -
val_accuracy: 0.9885 - 33s/epoch - 20ms/step
313/313 [=====] - 4s 12ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 972    0    2    1    0    2    1    1    1    0]
 [  0 1129    2    1    0    1    1    1    0    0]
 [  3    2 1016    0    3    0    2    5    0    1]
 [  0    1    2 997    0    4    0    3    3    0]
 [  0    0    1    0 972    0    2    0    1    6]
 [  2    0    0    5    0 879    5    0    0    1]
 [  6    2    0    1    1    6 941    0    1    0]
 [  1    2    5    0    0    0    0 1018    0    2]
 [  4    0    2    0    1    3    0    1 962    1]
 [  1    0    0    5    9    4    0    3    1 986]]
```

Precision: 0.9872
Recall: 0.9872

Confusion Matrix for tanh Activation											
	0	1	2	3	4	5	6	7	8	9	
0	972	0	2	1	0	2	1	1	1	0	0
1	0	1129	2	1	0	1	1	1	0	0	0
2	3	2	1016	0	3	0	2	5	0	1	
3	0	1	2	997	0	4	0	3	3	0	0
4	0	0	1	0	972	0	2	0	1	6	
5	2	0	0	5	0	879	5	0	0	1	
6	6	2	0	1	1	6	941	0	1	0	
7	1	2	5	0	0	0	0	1018	0	2	
8	4	0	2	0	1	3	0	1	962	1	
9	1	0	0	5	9	4	0	3	1	986	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	

```

Training Model with tanh activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 42s - loss: 0.1943 - accuracy: 0.9424 - val_loss: 0.0737 -
val_accuracy: 0.9787 - 42s/epoch - 25ms/step
Epoch 2/10
1688/1688 - 41s - loss: 0.0731 - accuracy: 0.9788 - val_loss: 0.0657 -
val_accuracy: 0.9815 - 41s/epoch - 24ms/step
Epoch 3/10
1688/1688 - 40s - loss: 0.0458 - accuracy: 0.9872 - val_loss: 0.0527 -
val_accuracy: 0.9867 - 40s/epoch - 24ms/step
Epoch 4/10
1688/1688 - 41s - loss: 0.0308 - accuracy: 0.9917 - val_loss: 0.0529 -
val_accuracy: 0.9852 - 41s/epoch - 25ms/step
Epoch 5/10
1688/1688 - 41s - loss: 0.0217 - accuracy: 0.9944 - val_loss: 0.0476 -
val_accuracy: 0.9860 - 41s/epoch - 24ms/step
Epoch 6/10
1688/1688 - 41s - loss: 0.0146 - accuracy: 0.9970 - val_loss: 0.0439 -
val_accuracy: 0.9882 - 41s/epoch - 24ms/step
Epoch 7/10
1688/1688 - 41s - loss: 0.0099 - accuracy: 0.9981 - val_loss: 0.0465 -

```

```
val_accuracy: 0.9875 - 41s/epoch - 24ms/step
Epoch 8/10
1688/1688 - 41s - loss: 0.0071 - accuracy: 0.9990 - val_loss: 0.0413 -
val_accuracy: 0.9890 - 41s/epoch - 24ms/step
Epoch 9/10
1688/1688 - 39s - loss: 0.0050 - accuracy: 0.9996 - val_loss: 0.0418 -
val_accuracy: 0.9885 - 39s/epoch - 23ms/step
Epoch 10/10
1688/1688 - 40s - loss: 0.0038 - accuracy: 0.9997 - val_loss: 0.0426 -
val_accuracy: 0.9888 - 40s/epoch - 23ms/step
313/313 [=====] - 3s 9ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 972    0    1    1    0    0    2    2    2    0]
 [  0 1128    3    1    0    1    0    1    1    0]
 [  6    1 1012    1    2    0    2    4    4    0]
 [  0    0    2  999    0    3    0    2    3    1]
 [  0    0    0    0  973    0    1    0    1    7]
 [  2    0    0    6    0  877    4    0    1    2]
 [  7    2    1    0    1    3  944    0    0    0]
 [  2    1    7    2    0    0    0 1015    0    1]
 [  6    0    2    0    2    1    0    3  958    2]
 [  1    0    0    4    8    1    0    4    2 989]]
```

Precision: 0.9867
Recall: 0.9867

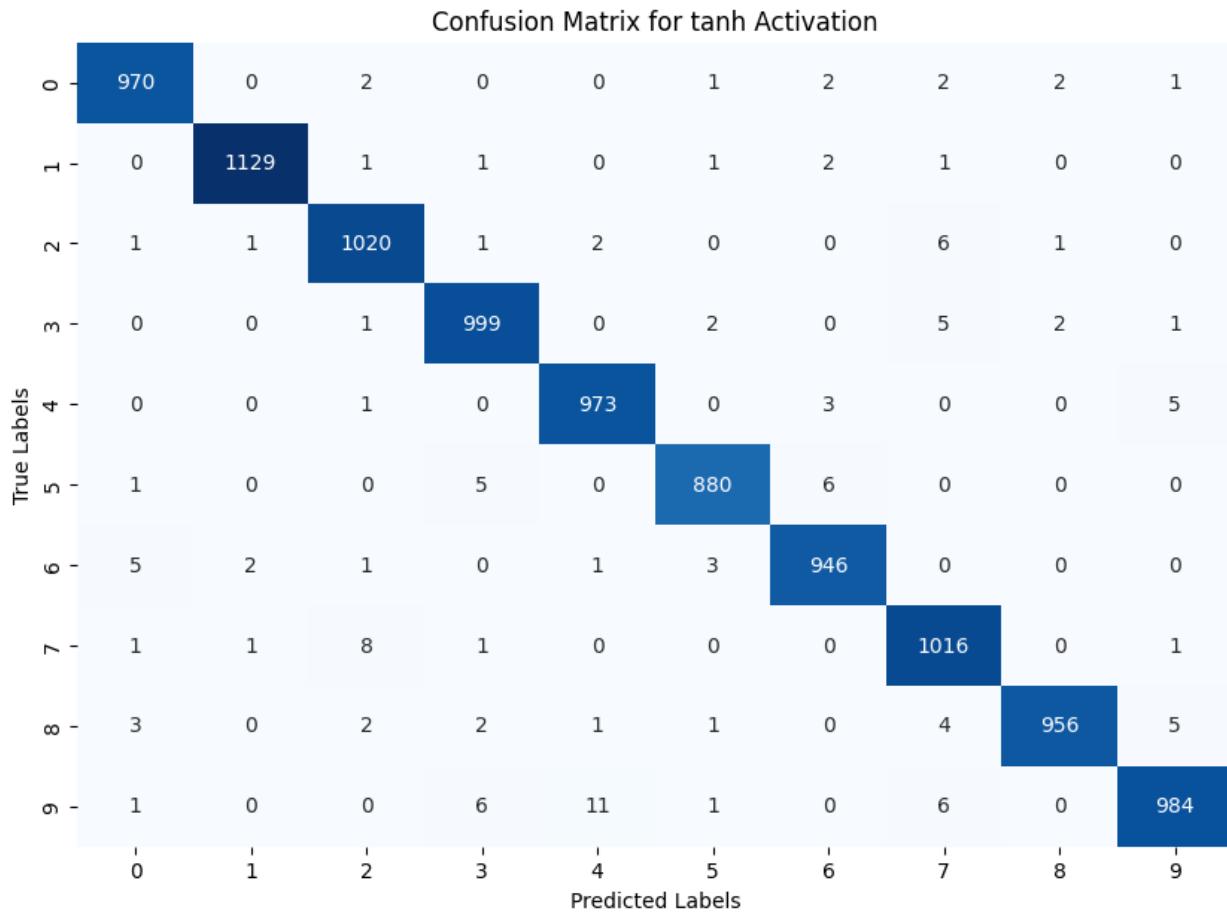
Confusion Matrix for tanh Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	972	0	1	1	0	0	2	2	2	0	
0	972	0	1	1	0	0	2	2	2	0	
1	0	1128	3	1	0	1	0	1	1	0	
2	6	1	1012	1	2	0	2	4	4	0	
3	0	0	2	999	0	3	0	2	3	1	
4	0	0	0	0	973	0	1	0	1	7	
5	2	0	0	6	0	877	4	0	1	2	
6	7	2	1	0	1	3	944	0	0	0	
7	2	1	7	2	0	0	0	0	1015	0	
8	6	0	2	0	2	1	0	3	958	2	
9	1	0	0	4	8	1	0	4	2	989	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```

Training Model with tanh activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 54s - loss: 0.1928 - accuracy: 0.9416 - val_loss: 0.0808 -
val_accuracy: 0.9802 - 54s/epoch - 32ms/step
Epoch 2/10
1688/1688 - 54s - loss: 0.0686 - accuracy: 0.9794 - val_loss: 0.0601 -
val_accuracy: 0.9838 - 54s/epoch - 32ms/step
Epoch 3/10
1688/1688 - 55s - loss: 0.0414 - accuracy: 0.9879 - val_loss: 0.0511 -
val_accuracy: 0.9867 - 55s/epoch - 33ms/step
Epoch 4/10
1688/1688 - 55s - loss: 0.0274 - accuracy: 0.9924 - val_loss: 0.0504 -
val_accuracy: 0.9865 - 55s/epoch - 33ms/step
Epoch 5/10
1688/1688 - 54s - loss: 0.0182 - accuracy: 0.9955 - val_loss: 0.0485 -
val_accuracy: 0.9865 - 54s/epoch - 32ms/step
Epoch 6/10
1688/1688 - 54s - loss: 0.0114 - accuracy: 0.9978 - val_loss: 0.0443 -
val_accuracy: 0.9888 - 54s/epoch - 32ms/step
Epoch 7/10
1688/1688 - 54s - loss: 0.0073 - accuracy: 0.9990 - val_loss: 0.0440 -

```

```
val_accuracy: 0.9885 - 54s/epoch - 32ms/step
Epoch 8/10
1688/1688 - 53s - loss: 0.0055 - accuracy: 0.9994 - val_loss: 0.0427 -
val_accuracy: 0.9892 - 53s/epoch - 31ms/step
Epoch 9/10
1688/1688 - 55s - loss: 0.0036 - accuracy: 0.9997 - val_loss: 0.0459 -
val_accuracy: 0.9892 - 55s/epoch - 33ms/step
Epoch 10/10
1688/1688 - 55s - loss: 0.0027 - accuracy: 0.9999 - val_loss: 0.0434 -
val_accuracy: 0.9892 - 55s/epoch - 32ms/step
313/313 [=====] - 4s 12ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 970    0    2    0    0    1    2    2    2    1]
 [  0 1129    1    1    0    1    2    1    0    0]
 [  1    1 1020    1    2    0    0    6    1    0]
 [  0    0    1 999    0    2    0    5    2    1]
 [  0    0    1    0 973    0    3    0    0    5]
 [  1    0    0    5    0 880    6    0    0    0]
 [  5    2    1    0    1    3 946    0    0    0]
 [  1    1    8    1    0    0    0 1016    0    1]
 [  3    0    2    2    1    1    0    4 956    5]
 [  1    0    0    6   11    1    0    6    0 984]]]
Precision: 0.9873
Recall: 0.9873
```



```
Training Model with tanh activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 51s - loss: 0.1488 - accuracy: 0.9565 - val_loss: 0.0578 -
val_accuracy: 0.9830 - 51s/epoch - 30ms/step
Epoch 2/10
1688/1688 - 50s - loss: 0.0508 - accuracy: 0.9845 - val_loss: 0.0492 -
val_accuracy: 0.9868 - 50s/epoch - 30ms/step
Epoch 3/10
1688/1688 - 49s - loss: 0.0332 - accuracy: 0.9898 - val_loss: 0.0478 -
val_accuracy: 0.9877 - 49s/epoch - 29ms/step
Epoch 4/10
1688/1688 - 50s - loss: 0.0232 - accuracy: 0.9929 - val_loss: 0.0470 -
val_accuracy: 0.9878 - 50s/epoch - 30ms/step
Epoch 5/10
1688/1688 - 48s - loss: 0.0150 - accuracy: 0.9963 - val_loss: 0.0483 -
val_accuracy: 0.9872 - 48s/epoch - 29ms/step
Epoch 6/10
1688/1688 - 49s - loss: 0.0104 - accuracy: 0.9974 - val_loss: 0.0440 -
val_accuracy: 0.9878 - 49s/epoch - 29ms/step
Epoch 7/10
1688/1688 - 50s - loss: 0.0064 - accuracy: 0.9987 - val_loss: 0.0482 -
```

```
val_accuracy: 0.9870 - 50s/epoch - 30ms/step
Epoch 8/10
1688/1688 - 49s - loss: 0.0047 - accuracy: 0.9992 - val_loss: 0.0444 -
val_accuracy: 0.9882 - 49s/epoch - 29ms/step
Epoch 9/10
1688/1688 - 49s - loss: 0.0035 - accuracy: 0.9995 - val_loss: 0.0430 -
val_accuracy: 0.9883 - 49s/epoch - 29ms/step
Epoch 10/10
1688/1688 - 50s - loss: 0.0021 - accuracy: 0.9999 - val_loss: 0.0423 -
val_accuracy: 0.9900 - 50s/epoch - 30ms/step
313/313 [=====] - 4s 14ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 976   0   0   0   0   0   1   1   2   0]
 [  0 1131   0   1   0   0   1   1   0   1]
 [  1   2 1025   0   1   0   0   2   1   0]
 [  0   0   0 1003   0   3   0   2   1   1]
 [  0   0   1   0  975   0   2   1   0   3]
 [  2   0   0    7   0  879   2   0   1   1]
 [  2   2   1   0    2   2  947   0   2   0]
 [  0   2   6   2   0   0   0 1013   0   5]
 [  4   0   2   0   1   0   1   1  963   2]
 [  1   0   0   2   3   4   1   0   1  997]]
```

Precision: 0.9909
Recall: 0.9909

Confusion Matrix for tanh Activation										
	0	1	2	3	4	5	6	7	8	9
0	976	0	0	0	0	0	1	1	2	0
1	0	1131	0	1	0	0	1	1	0	1
2	1	2	1025	0	1	0	0	2	1	0
3	0	0	0	1003	0	3	0	2	1	1
4	0	0	1	0	975	0	2	1	0	3
5	2	0	0	7	0	879	2	0	1	1
6	2	2	1	0	2	2	947	0	2	0
7	0	2	6	2	0	0	0	1013	0	5
8	4	0	2	0	1	0	1	1	963	2
9	1	0	0	2	3	4	1	0	1	997
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```

Training Model with tanh activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 50s - loss: 0.1430 - accuracy: 0.9568 - val_loss: 0.0530 -
val_accuracy: 0.9848 - 50s/epoch - 30ms/step
Epoch 2/10
1688/1688 - 51s - loss: 0.0478 - accuracy: 0.9849 - val_loss: 0.0408 -
val_accuracy: 0.9887 - 51s/epoch - 30ms/step
Epoch 3/10
1688/1688 - 50s - loss: 0.0286 - accuracy: 0.9912 - val_loss: 0.0412 -
val_accuracy: 0.9887 - 50s/epoch - 30ms/step
Epoch 4/10
1688/1688 - 48s - loss: 0.0194 - accuracy: 0.9941 - val_loss: 0.0404 -
val_accuracy: 0.9887 - 48s/epoch - 28ms/step
Epoch 5/10
1688/1688 - 50s - loss: 0.0120 - accuracy: 0.9969 - val_loss: 0.0383 -
val_accuracy: 0.9893 - 50s/epoch - 30ms/step
Epoch 6/10
1688/1688 - 49s - loss: 0.0079 - accuracy: 0.9981 - val_loss: 0.0445 -
val_accuracy: 0.9882 - 49s/epoch - 29ms/step
Epoch 7/10
1688/1688 - 48s - loss: 0.0056 - accuracy: 0.9987 - val_loss: 0.0421 -

```

```
val_accuracy: 0.9887 - 48s/epoch - 29ms/step
Epoch 8/10
1688/1688 - 50s - loss: 0.0032 - accuracy: 0.9995 - val_loss: 0.0401 -
val_accuracy: 0.9900 - 50s/epoch - 30ms/step
Epoch 9/10
1688/1688 - 49s - loss: 0.0021 - accuracy: 0.9998 - val_loss: 0.0391 -
val_accuracy: 0.9910 - 49s/epoch - 29ms/step
Epoch 10/10
1688/1688 - 49s - loss: 0.0014 - accuracy: 1.0000 - val_loss: 0.0395 -
val_accuracy: 0.9898 - 49s/epoch - 29ms/step
313/313 [=====] - 4s 11ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 977   0   1   0   0   0   1   1   0   0]
 [  0 1131   1   1   0   1   1   0   0   0]
 [  0   2 1024   0   1   0   0   3   2   0]
 [  0   1   0 1004   0   3   0   0   2   0]
 [  0   0   0   0  973   0   3   0   0   6]
 [  1   0   1   7   0  881   2   0   0   0]
 [  4   2   0   0   1   3  946   0   2   0]
 [  0   3   8   0   0   0   0 1013   1   3]
 [  2   0   3   2   1   0   1   2  961   2]
 [  1   0   0   2   4   6   0   2   0  994]]
```

Precision: 0.9904
Recall: 0.9904

Confusion Matrix for tanh Activation										
	0	1	2	3	4	5	6	7	8	9
0	977	0	1	0	0	0	1	1	0	0
1	0	1131	1	1	0	1	1	0	0	0
2	0	2	1024	0	1	0	0	3	2	0
3	0	1	0	1004	0	3	0	0	2	0
4	0	0	0	0	973	0	3	0	0	6
5	1	0	1	7	0	881	2	0	0	0
6	4	2	0	0	1	3	946	0	2	0
7	0	3	8	0	0	0	0	1013	1	3
8	2	0	3	2	1	0	1	2	961	2
9	1	0	0	2	4	6	0	2	0	994
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```

Training Model with tanh activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 54s - loss: 0.1406 - accuracy: 0.9566 - val_loss: 0.0578 -
val_accuracy: 0.9848 - 54s/epoch - 32ms/step
Epoch 2/10
1688/1688 - 53s - loss: 0.0456 - accuracy: 0.9864 - val_loss: 0.0474 -
val_accuracy: 0.9872 - 53s/epoch - 32ms/step
Epoch 3/10
1688/1688 - 53s - loss: 0.0280 - accuracy: 0.9911 - val_loss: 0.0407 -
val_accuracy: 0.9888 - 53s/epoch - 31ms/step
Epoch 4/10
1688/1688 - 53s - loss: 0.0166 - accuracy: 0.9951 - val_loss: 0.0425 -
val_accuracy: 0.9882 - 53s/epoch - 31ms/step
Epoch 5/10
1688/1688 - 53s - loss: 0.0100 - accuracy: 0.9974 - val_loss: 0.0365 -
val_accuracy: 0.9912 - 53s/epoch - 31ms/step
Epoch 6/10
1688/1688 - 52s - loss: 0.0064 - accuracy: 0.9988 - val_loss: 0.0382 -
val_accuracy: 0.9900 - 52s/epoch - 31ms/step
Epoch 7/10
1688/1688 - 54s - loss: 0.0039 - accuracy: 0.9994 - val_loss: 0.0437 -

```

```
val_accuracy: 0.9885 - 54s/epoch - 32ms/step
Epoch 8/10
1688/1688 - 53s - loss: 0.0026 - accuracy: 0.9997 - val_loss: 0.0396 -
val_accuracy: 0.9897 - 53s/epoch - 31ms/step
Epoch 9/10
1688/1688 - 54s - loss: 0.0016 - accuracy: 0.9999 - val_loss: 0.0391 -
val_accuracy: 0.9905 - 54s/epoch - 32ms/step
Epoch 10/10
1688/1688 - 53s - loss: 0.0012 - accuracy: 0.9999 - val_loss: 0.0393 -
val_accuracy: 0.9903 - 53s/epoch - 32ms/step
313/313 [=====] - 4s 12ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 975    0    0    0    0    0    2    1    2    0]
 [  0 1131    2    1    0    0    1    0    0    0]
 [  0    0 1023    0    1    0    0    5    2    1]
 [  0    0    0 1003    0    3    0    0    2    2]
 [  0    0    0    0  975    0    3    0    0    4]
 [  1    0    1    5    0  881    4    0    0    0]
 [  4    2    0    0    1    3  948    0    0    0]
 [  0    1    5    1    0    0    0 1017    1    3]
 [  3    0    1    2    0    1    0    1   964    2]
 [  0    0    0    0    3    3    1    3    3   996]]
```

Precision: 0.9913
Recall: 0.9913

Confusion Matrix for tanh Activation										
	0	1	2	3	4	5	6	7	8	9
True Labels	975	0	0	0	0	0	2	1	2	0
0	975	0	0	0	0	0	1	0	0	0
1	0	1131	2	1	0	0	1	0	0	0
2	0	0	1023	0	1	0	0	5	2	1
3	0	0	0	1003	0	3	0	0	2	2
4	0	0	0	0	975	0	3	0	0	4
5	1	0	1	5	0	881	4	0	0	0
6	4	2	0	0	1	3	948	0	0	0
7	0	1	5	1	0	0	0	1017	1	3
8	3	0	1	2	0	1	0	1	964	2
9	0	0	0	0	3	3	1	3	3	996
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```

Training Model with tanh activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 51s - loss: 0.2331 - accuracy: 0.9284 - val_loss: 0.0972 -
val_accuracy: 0.9730 - 51s/epoch - 30ms/step
Epoch 2/10
1688/1688 - 49s - loss: 0.0967 - accuracy: 0.9714 - val_loss: 0.0891 -
val_accuracy: 0.9758 - 49s/epoch - 29ms/step
Epoch 3/10
1688/1688 - 49s - loss: 0.0752 - accuracy: 0.9761 - val_loss: 0.0758 -
val_accuracy: 0.9792 - 49s/epoch - 29ms/step
Epoch 4/10
1688/1688 - 51s - loss: 0.0588 - accuracy: 0.9819 - val_loss: 0.0708 -
val_accuracy: 0.9817 - 51s/epoch - 30ms/step
Epoch 5/10
1688/1688 - 49s - loss: 0.0467 - accuracy: 0.9858 - val_loss: 0.0618 -
val_accuracy: 0.9837 - 49s/epoch - 29ms/step
Epoch 6/10
1688/1688 - 48s - loss: 0.0410 - accuracy: 0.9868 - val_loss: 0.0631 -
val_accuracy: 0.9832 - 48s/epoch - 29ms/step
Epoch 7/10
1688/1688 - 50s - loss: 0.0370 - accuracy: 0.9885 - val_loss: 0.0655 -

```

```
val_accuracy: 0.9820 - 50s/epoch - 30ms/step
Epoch 8/10
1688/1688 - 49s - loss: 0.0324 - accuracy: 0.9894 - val_loss: 0.0691 -
val_accuracy: 0.9815 - 49s/epoch - 29ms/step
Epoch 9/10
1688/1688 - 50s - loss: 0.0269 - accuracy: 0.9910 - val_loss: 0.0633 -
val_accuracy: 0.9838 - 50s/epoch - 30ms/step
Epoch 10/10
1688/1688 - 52s - loss: 0.0218 - accuracy: 0.9929 - val_loss: 0.0708 -
val_accuracy: 0.9820 - 52s/epoch - 31ms/step
313/313 [=====] - 5s 15ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 961   2   3   0   4   1   2   2   0   5]
 [  0 1128   4   1   0   0   0   1   1   0]
 [  2   0 1013   1   2   0   1  10   2   1]
 [  0   0   9 989   0   3   0   5   1   3]
 [  1   0   0   0 975   0   5   0   0   1]
 [  2   0   1   8   0 870   2   3   3   3]
 [  5   3   1   0   4   3 940   0   2   0]
 [  0   3  13   7   1   0   0 998   1   5]
 [  0   0   2   3   6   1   1   2 955   4]
 [  0   1   1   0 13   4   0   5   7 978]]
```

Precision: 0.9808
Recall: 0.9807

Confusion Matrix for tanh Activation												
	0	1	2	3	4	5	6	7	8	9	10	11
True Labels	961	2	3	0	4	1	2	2	0	5	0	5
0	961	2	3	0	4	1	2	2	0	5	0	5
1	0	1128	4	1	0	0	0	1	1	1	0	0
2	2	0	1013	1	2	0	1	10	2	1	1	1
3	0	0	9	989	0	3	0	5	1	3	1	3
4	1	0	0	0	975	0	5	0	0	0	1	1
5	2	0	1	8	0	870	2	3	3	3	3	3
6	5	3	1	0	4	3	940	0	2	0	2	0
7	0	3	13	7	1	0	0	998	1	5	1	5
8	0	0	2	3	6	1	1	2	955	4	4	4
9	0	1	1	0	13	4	0	5	7	978	978	978
10	1	2	3	4	5	6	7	8	9	9	9	9
Predicted Labels												

```

Training Model with tanh activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 52s - loss: 0.2357 - accuracy: 0.9270 - val_loss: 0.0860 -
val_accuracy: 0.9725 - 52s/epoch - 31ms/step
Epoch 2/10
1688/1688 - 51s - loss: 0.0955 - accuracy: 0.9711 - val_loss: 0.0775 -
val_accuracy: 0.9767 - 51s/epoch - 30ms/step
Epoch 3/10
1688/1688 - 51s - loss: 0.0735 - accuracy: 0.9773 - val_loss: 0.0670 -
val_accuracy: 0.9783 - 51s/epoch - 30ms/step
Epoch 4/10
1688/1688 - 52s - loss: 0.0570 - accuracy: 0.9821 - val_loss: 0.0571 -
val_accuracy: 0.9832 - 52s/epoch - 31ms/step
Epoch 5/10
1688/1688 - 51s - loss: 0.0469 - accuracy: 0.9853 - val_loss: 0.0612 -
val_accuracy: 0.9813 - 51s/epoch - 30ms/step
Epoch 6/10
1688/1688 - 50s - loss: 0.0391 - accuracy: 0.9874 - val_loss: 0.0568 -
val_accuracy: 0.9850 - 50s/epoch - 30ms/step
Epoch 7/10
1688/1688 - 52s - loss: 0.0381 - accuracy: 0.9881 - val_loss: 0.0577 -

```

```
val_accuracy: 0.9843 - 52s/epoch - 31ms/step
Epoch 8/10
1688/1688 - 52s - loss: 0.0300 - accuracy: 0.9906 - val_loss: 0.0555 -
val_accuracy: 0.9863 - 52s/epoch - 31ms/step
Epoch 9/10
1688/1688 - 51s - loss: 0.0241 - accuracy: 0.9924 - val_loss: 0.0635 -
val_accuracy: 0.9845 - 51s/epoch - 30ms/step
Epoch 10/10
1688/1688 - 51s - loss: 0.0212 - accuracy: 0.9930 - val_loss: 0.0713 -
val_accuracy: 0.9813 - 51s/epoch - 30ms/step
313/313 [=====] - 4s 12ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 962    0    3    1    2    4    5    3    0    0]
 [  0 1126    2    4    0    1    0    2    0    0]
 [  1    0 1016    7    2    0    1    3    1    1]
 [  0    0    3 999    0    6    0    0    1    1]
 [  2    0    0    1 965    0    2    3    0    9]
 [  1    1    0    5    0 878    1    2    4    0]
 [  3    2    3    0    2    9 938    0    1    0]
 [  1    1   30   13    0    2    0 977    1    3]
 [  0    0    8    3    0    7    0    1 954    1]
 [  3    0    0    4    6    9    2    4    7 974]]]
Precision: 0.9791
Recall: 0.9789
```

Confusion Matrix for tanh Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	0	962	0	3	1	2	4	5	3	0	0
0	962	0	3	1	2	4	5	3	0	0	0
1	0	1126	2	4	0	1	0	2	0	0	0
2	1	0	1016	7	2	0	1	3	1	1	1
3	0	0	3	999	0	6	0	0	1	1	1
4	2	0	0	1	965	0	2	3	0	9	9
5	1	1	0	5	0	878	1	2	4	0	0
6	3	2	3	0	2	9	938	0	1	0	0
7	1	1	30	13	0	2	0	977	1	3	1
8	0	0	8	3	0	7	0	1	954	1	1
9	3	0	0	4	6	9	2	4	7	974	9
0	1	2	3	4	5	6	7	8	9	9	9

```

Training Model with tanh activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 55s - loss: 0.2201 - accuracy: 0.9306 - val_loss: 0.0959 -
val_accuracy: 0.9718 - 55s/epoch - 32ms/step
Epoch 2/10
1688/1688 - 53s - loss: 0.0879 - accuracy: 0.9723 - val_loss: 0.0845 -
val_accuracy: 0.9750 - 53s/epoch - 31ms/step
Epoch 3/10
1688/1688 - 54s - loss: 0.0643 - accuracy: 0.9800 - val_loss: 0.0763 -
val_accuracy: 0.9782 - 54s/epoch - 32ms/step
Epoch 4/10
1688/1688 - 54s - loss: 0.0510 - accuracy: 0.9841 - val_loss: 0.0730 -
val_accuracy: 0.9798 - 54s/epoch - 32ms/step
Epoch 5/10
1688/1688 - 54s - loss: 0.0388 - accuracy: 0.9876 - val_loss: 0.0644 -
val_accuracy: 0.9823 - 54s/epoch - 32ms/step
Epoch 6/10
1688/1688 - 53s - loss: 0.0350 - accuracy: 0.9883 - val_loss: 0.0623 -
val_accuracy: 0.9823 - 53s/epoch - 31ms/step
Epoch 7/10
1688/1688 - 53s - loss: 0.0305 - accuracy: 0.9899 - val_loss: 0.0672 -

```

```
val_accuracy: 0.9840 - 53s/epoch - 31ms/step
Epoch 8/10
1688/1688 - 53s - loss: 0.0216 - accuracy: 0.9931 - val_loss: 0.0765 -
val_accuracy: 0.9815 - 53s/epoch - 31ms/step
Epoch 9/10
1688/1688 - 53s - loss: 0.0216 - accuracy: 0.9927 - val_loss: 0.0686 -
val_accuracy: 0.9818 - 53s/epoch - 31ms/step
Epoch 10/10
1688/1688 - 53s - loss: 0.0198 - accuracy: 0.9934 - val_loss: 0.0745 -
val_accuracy: 0.9803 - 53s/epoch - 31ms/step
313/313 [=====] - 5s 17ms/step
Results for activation function: tanh
Confusion Matrix:
[[ 955    0    4    0    3    3    5    3    7    0]
 [  1 1122    1    0    1    1    1    1    7    0]
 [  0    0 1014    6    0    0    1    7    3    1]
 [  0    0    6 995    0    4    0    0    4    1]
 [  0    0    1    0 965    0    2    0    7    7]
 [  1    0    3    7    0 872    3    2    2    2]
 [  2    2    1    0    4    3 940    0    6    0]
 [  0    2   23   19    0    0    0 977    3    4]
 [  0    0    1    4    0    1    0    1 966    1]
 [  2    0    3    2    2    1    1    3   12 983]]
```

Precision: 0.9792
Recall: 0.9789

Confusion Matrix for tanh Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	955	0	4	0	3	3	5	3	7	0	
0	955	0	4	0	3	3	5	3	7	0	
1	1	1122	1	0	1	1	1	1	7	0	
2	0	0	1014	6	0	0	1	7	3	1	
3	0	0	6	995	0	4	0	0	4	1	
4	0	0	1	0	965	0	2	0	7	7	
5	1	0	3	7	0	872	3	2	2	2	
6	2	2	1	0	4	3	940	0	6	0	
7	0	2	23	19	0	0	0	977	3	4	
8	0	0	1	4	0	1	0	1	966	1	
9	2	0	3	2	2	1	1	3	12	983	
0	1	2	3	4	5	6	7	8	9	9	
Predicted Labels											

```

Training Model with softmax activation, 1 conv_layers, 1 dense
layers..
Epoch 1/10
1688/1688 - 42s - loss: 2.2997 - accuracy: 0.1154 - val_loss: 2.2969 -
val_accuracy: 0.1057 - 42s/epoch - 25ms/step
Epoch 2/10
1688/1688 - 42s - loss: 2.1998 - accuracy: 0.1944 - val_loss: 1.9001 -
val_accuracy: 0.3330 - 42s/epoch - 25ms/step
Epoch 3/10
1688/1688 - 40s - loss: 1.4197 - accuracy: 0.4646 - val_loss: 1.1954 -
val_accuracy: 0.5207 - 40s/epoch - 24ms/step
Epoch 4/10
1688/1688 - 41s - loss: 1.1847 - accuracy: 0.5440 - val_loss: 1.0895 -
val_accuracy: 0.5742 - 41s/epoch - 24ms/step
Epoch 5/10
1688/1688 - 41s - loss: 1.1066 - accuracy: 0.5764 - val_loss: 1.0328 -
val_accuracy: 0.5993 - 41s/epoch - 24ms/step
Epoch 6/10
1688/1688 - 41s - loss: 1.0490 - accuracy: 0.6114 - val_loss: 0.9900 -
val_accuracy: 0.6387 - 41s/epoch - 24ms/step
Epoch 7/10

```

```
1688/1688 - 41s - loss: 0.9994 - accuracy: 0.6428 - val_loss: 0.9630 -  
val_accuracy: 0.6415 - 41s/epoch - 24ms/step  
Epoch 8/10  
1688/1688 - 40s - loss: 0.9499 - accuracy: 0.6654 - val_loss: 0.8667 -  
val_accuracy: 0.6968 - 40s/epoch - 24ms/step  
Epoch 9/10  
1688/1688 - 41s - loss: 0.8929 - accuracy: 0.6986 - val_loss: 0.8467 -  
val_accuracy: 0.6975 - 41s/epoch - 24ms/step  
Epoch 10/10  
1688/1688 - 41s - loss: 0.8400 - accuracy: 0.7265 - val_loss: 0.7603 -  
val_accuracy: 0.7603 - 41s/epoch - 24ms/step  
313/313 [=====] - 5s 17ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 914  0  0   8   1  10  43  2   1   1]  
 [ 0 1103  8  3   0   0   0   0  21  0]  
 [ 5  742 137 92   4   4   2   8  33  5]  
 [ 6  5  33 875   6  23  17  22  21  2]  
 [ 0  0  0   0  937  6   2   0  20  17]  
 [ 12  1  6  49 109 438 164  3  100 10]  
 [ 92  2  1  8  18 125 709  0   2  1]  
 [ 0  36  1  4   7   3   0  901  6  70]  
 [ 6  51  8  8 180  62  16  2  626 15]  
 [ 3  3  0   4 131  9  10  17  21 811]]  
Precision: 0.7582  
Recall: 0.7451
```

Confusion Matrix for softmax Activation

	0	1	2	3	4	5	6	7	8	9
0	914	0	0	8	1	10	43	2	1	1
1	0	1103	8	3	0	0	0	0	21	0
2	5	742	137	92	4	4	2	8	33	5
3	6	5	33	875	6	23	17	22	21	2
4	0	0	0	0	937	6	2	0	20	17
5	12	1	6	49	109	438	164	3	100	10
6	92	2	1	8	18	125	709	0	2	1
7	0	36	1	4	7	3	0	901	6	70
8	6	51	8	8	180	62	16	2	626	15
9	3	3	0	4	131	9	10	17	21	811
	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```
Training Model with softmax activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 48s - loss: 2.3013 - accuracy: 0.1125 - val_loss: 2.3008 - val_accuracy: 0.1050 - 48s/epoch - 28ms/step
Epoch 2/10
1688/1688 - 49s - loss: 2.2998 - accuracy: 0.1160 - val_loss: 2.2986 - val_accuracy: 0.2055 - 49s/epoch - 29ms/step
Epoch 3/10
1688/1688 - 47s - loss: 2.2952 - accuracy: 0.1335 - val_loss: 2.2881 - val_accuracy: 0.2117 - 47s/epoch - 28ms/step
Epoch 4/10
1688/1688 - 48s - loss: 2.0265 - accuracy: 0.2511 - val_loss: 1.6856 - val_accuracy: 0.3495 - 48s/epoch - 29ms/step
Epoch 5/10
1688/1688 - 46s - loss: 1.4083 - accuracy: 0.4505 - val_loss: 1.2255 - val_accuracy: 0.4962 - 46s/epoch - 27ms/step
Epoch 6/10
1688/1688 - 47s - loss: 1.2108 - accuracy: 0.5151 - val_loss: 1.1486 - val_accuracy: 0.5640 - 47s/epoch - 28ms/step
Epoch 7/10
```

```
1688/1688 - 45s - loss: 1.1405 - accuracy: 0.5792 - val_loss: 1.0687 -  
val_accuracy: 0.6387 - 45s/epoch - 27ms/step  
Epoch 8/10  
1688/1688 - 45s - loss: 1.0612 - accuracy: 0.6298 - val_loss: 0.9767 -  
val_accuracy: 0.6827 - 45s/epoch - 27ms/step  
Epoch 9/10  
1688/1688 - 46s - loss: 0.9765 - accuracy: 0.6743 - val_loss: 0.9028 -  
val_accuracy: 0.7230 - 46s/epoch - 27ms/step  
Epoch 10/10  
1688/1688 - 46s - loss: 0.9194 - accuracy: 0.7035 - val_loss: 0.8592 -  
val_accuracy: 0.7297 - 46s/epoch - 27ms/step  
313/313 [=====] - 6s 19ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 926   1   21   25    0    3    0    1    2    1]  
 [  1 1097   4    3    0    0    1    0   29    0]  
 [ 193    4 546 103    8   71   24    1   63   19]  
 [ 545    0   17 404    0   12    0   10   15    7]  
 [  0    1   11    0 787    0   23    0    9 151]  
 [ 206    2 182 332    3   86    9    2   57   13]  
 [  7    1   77    0    7    0 859    0    7    0]  
 [  1    1    2   20    1    2    0 945   11   45]  
 [ 26   33   51 111    7   35    4    5 678   24]  
 [  2    3    4   35   36    3    0 216    9 701]]  
Precision: 0.7052  
Recall: 0.7029
```

Confusion Matrix for softmax Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	926	1	21	25	0	3	0	1	2	1	
0	926	1	21	25	0	3	0	1	2	1	
1	1	1097	4	3	0	0	1	0	29	0	
2	193	4	546	103	8	71	24	1	63	19	
3	545	0	17	404	0	12	0	10	15	7	
4	0	1	11	0	787	0	23	0	9	151	
5	206	2	182	332	3	86	9	2	57	13	
6	7	1	77	0	7	0	859	0	7	0	
7	1	1	2	20	1	2	0	945	11	45	
8	26	33	51	111	7	35	4	5	678	24	
9	2	3	4	35	36	3	0	216	9	701	
0	1	2	3	4	5	6	7	8	9	9	
Predicted Labels											

```

Training Model with softmax activation, 1 conv_layers, 1 dense
layers..
Epoch 1/10
1688/1688 - 63s - loss: 2.3018 - accuracy: 0.1121 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 63s/epoch - 37ms/step
Epoch 2/10
1688/1688 - 60s - loss: 2.3016 - accuracy: 0.1130 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 60s/epoch - 35ms/step
Epoch 3/10
1688/1688 - 62s - loss: 2.3014 - accuracy: 0.1131 - val_loss: 2.3011 -
val_accuracy: 0.1113 - 62s/epoch - 36ms/step
Epoch 4/10
1688/1688 - 59s - loss: 2.3010 - accuracy: 0.1130 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 59s/epoch - 35ms/step
Epoch 5/10
1688/1688 - 63s - loss: 2.3003 - accuracy: 0.1150 - val_loss: 2.3006 -
val_accuracy: 0.1050 - 63s/epoch - 37ms/step
Epoch 6/10
1688/1688 - 60s - loss: 2.2989 - accuracy: 0.1154 - val_loss: 2.2989 -
val_accuracy: 0.1050 - 60s/epoch - 35ms/step
Epoch 7/10

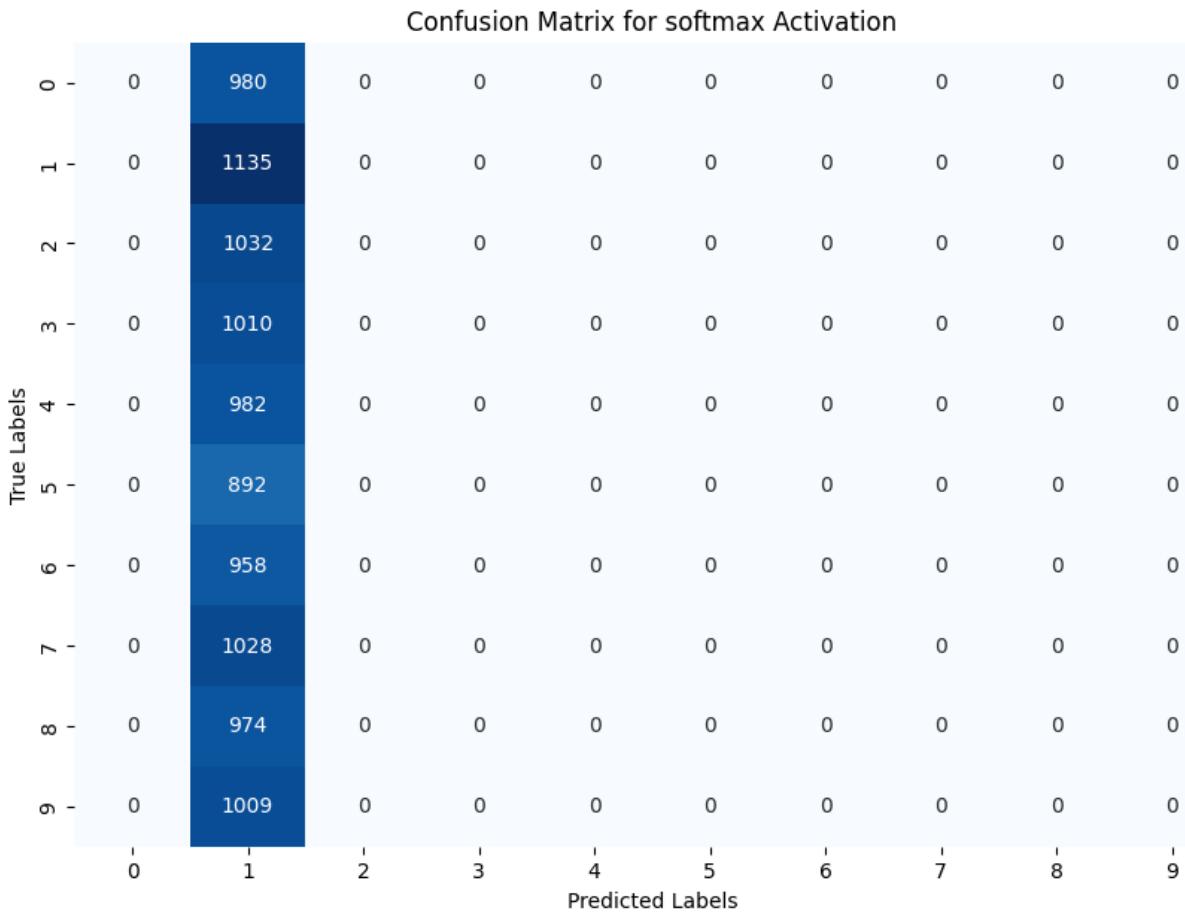
```

```
1688/1688 - 59s - loss: 2.2961 - accuracy: 0.1179 - val_loss: 2.2927 -  
val_accuracy: 0.1050 - 59s/epoch - 35ms/step  
Epoch 8/10  
1688/1688 - 59s - loss: 1.9608 - accuracy: 0.2650 - val_loss: 1.5333 -  
val_accuracy: 0.3552 - 59s/epoch - 35ms/step  
Epoch 9/10  
1688/1688 - 59s - loss: 1.3788 - accuracy: 0.4365 - val_loss: 1.2756 -  
val_accuracy: 0.4678 - 59s/epoch - 35ms/step  
Epoch 10/10  
1688/1688 - 61s - loss: 1.2372 - accuracy: 0.4818 - val_loss: 1.1679 -  
val_accuracy: 0.5018 - 61s/epoch - 36ms/step  
313/313 [=====] - 7s 22ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 901  0  29  26  2  3  2  0  14  3]  
 [ 2 1104  16  3  0  0  4  0  6  0]  
 [ 59  16  532  6  16  11  340  4  39  9]  
 [ 804  5  38  87  0  8  1  9  49  9]  
 [ 0  12  26  0  192  1  17  710  8  16]  
 [ 616  13  55  111  4  11  7  1  68  6]  
 [ 10  7  63  0  1  0  874  1  2  0]  
 [ 1  30  8  4  29  1  1  871  24  59]  
 [ 269  18  116  216  5  13  9  9  284  35]  
 [ 14  10  1  7  26  0  2  822  27  100]]  
Precision: 0.5055  
Recall: 0.4956
```

Confusion Matrix for softmax Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	901	0	29	26	2	3	2	0	14	3	
0	901	0	29	26	2	3	2	0	14	3	
1	2	1104	16	3	0	0	4	0	6	0	
2	59	16	532	6	16	11	340	4	39	9	
3	804	5	38	87	0	8	1	9	49	9	
4	0	12	26	0	192	1	17	710	8	16	
5	616	13	55	111	4	11	7	1	68	6	
6	10	7	63	0	1	0	874	1	2	0	
7	1	30	8	4	29	1	1	871	24	59	
8	269	18	116	216	5	13	9	9	284	35	
9	14	10	1	7	26	0	2	822	27	100	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

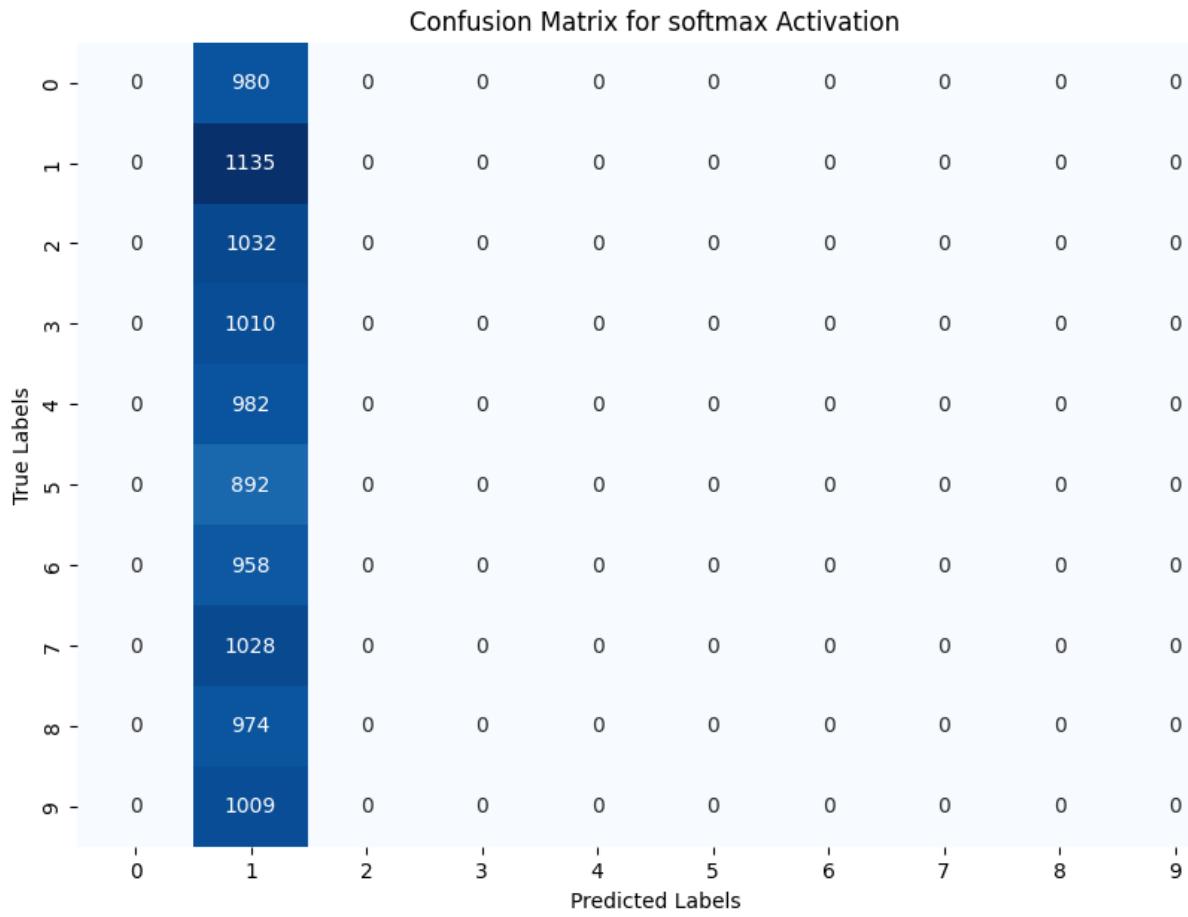
```
Training Model with softmax activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 58s - loss: 2.3018 - accuracy: 0.1105 - val_loss: 2.3026 - val_accuracy: 0.1050 - 58s/epoch - 34ms/step
Epoch 2/10
1688/1688 - 57s - loss: 2.3019 - accuracy: 0.1121 - val_loss: 2.3026 - val_accuracy: 0.1050 - 57s/epoch - 34ms/step
Epoch 3/10
1688/1688 - 60s - loss: 2.3019 - accuracy: 0.1125 - val_loss: 2.3019 - val_accuracy: 0.1050 - 60s/epoch - 35ms/step
Epoch 4/10
1688/1688 - 58s - loss: 2.3019 - accuracy: 0.1127 - val_loss: 2.3027 - val_accuracy: 0.1050 - 58s/epoch - 35ms/step
Epoch 5/10
1688/1688 - 58s - loss: 2.3019 - accuracy: 0.1116 - val_loss: 2.3032 - val_accuracy: 0.1050 - 58s/epoch - 35ms/step
Epoch 6/10
1688/1688 - 61s - loss: 2.3019 - accuracy: 0.1117 - val_loss: 2.3030 - val_accuracy: 0.1050 - 61s/epoch - 36ms/step
Epoch 7/10
```

```
1688/1688 - 57s - loss: 2.3019 - accuracy: 0.1120 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 57s/epoch - 34ms/step  
Epoch 8/10  
1688/1688 - 57s - loss: 2.3018 - accuracy: 0.1121 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 57s/epoch - 34ms/step  
Epoch 9/10  
1688/1688 - 57s - loss: 2.3019 - accuracy: 0.1126 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 57s/epoch - 34ms/step  
Epoch 10/10  
1688/1688 - 57s - loss: 2.3019 - accuracy: 0.1126 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 57s/epoch - 34ms/step  
313/313 [=====] - 5s 17ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]  
 [ 0 1135 0 0 0 0 0 0 0 ]  
 [ 0 1032 0 0 0 0 0 0 0 ]  
 [ 0 1010 0 0 0 0 0 0 0 ]  
 [ 0 982 0 0 0 0 0 0 0 ]  
 [ 0 892 0 0 0 0 0 0 0 ]  
 [ 0 958 0 0 0 0 0 0 0 ]  
 [ 0 1028 0 0 0 0 0 0 0 ]  
 [ 0 974 0 0 0 0 0 0 0 ]  
 [ 0 1009 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 61s - loss: 2.3017 - accuracy: 0.1124 - val_loss: 2.3027 - val_accuracy: 0.1050 - 61s/epoch - 36ms/step
Epoch 2/10
1688/1688 - 59s - loss: 2.3019 - accuracy: 0.1114 - val_loss: 2.3025 - val_accuracy: 0.1050 - 59s/epoch - 35ms/step
Epoch 3/10
1688/1688 - 58s - loss: 2.3019 - accuracy: 0.1126 - val_loss: 2.3026 - val_accuracy: 0.1050 - 58s/epoch - 35ms/step
Epoch 4/10
1688/1688 - 60s - loss: 2.3018 - accuracy: 0.1126 - val_loss: 2.3014 - val_accuracy: 0.1113 - 60s/epoch - 35ms/step
Epoch 5/10
1688/1688 - 58s - loss: 2.3018 - accuracy: 0.1128 - val_loss: 2.3014 - val_accuracy: 0.1050 - 58s/epoch - 35ms/step
Epoch 6/10
1688/1688 - 61s - loss: 2.3019 - accuracy: 0.1120 - val_loss: 2.3024 - val_accuracy: 0.1045 - 61s/epoch - 36ms/step
Epoch 7/10
```

```
1688/1688 - 58s - loss: 2.3020 - accuracy: 0.1121 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 58s/epoch - 35ms/step  
Epoch 8/10  
1688/1688 - 58s - loss: 2.3018 - accuracy: 0.1125 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 58s/epoch - 34ms/step  
Epoch 9/10  
1688/1688 - 59s - loss: 2.3018 - accuracy: 0.1126 - val_loss: 2.3033 -  
val_accuracy: 0.1050 - 59s/epoch - 35ms/step  
Epoch 10/10  
1688/1688 - 59s - loss: 2.3018 - accuracy: 0.1126 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 59s/epoch - 35ms/step  
313/313 [=====] - 5s 16ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
`zero_division` parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]  
 [ 0 1135 0 0 0 0 0 0 0 ]  
 [ 0 1032 0 0 0 0 0 0 0 ]  
 [ 0 1010 0 0 0 0 0 0 0 ]  
 [ 0 982 0 0 0 0 0 0 0 ]  
 [ 0 892 0 0 0 0 0 0 0 ]  
 [ 0 958 0 0 0 0 0 0 0 ]  
 [ 0 1028 0 0 0 0 0 0 0 ]  
 [ 0 974 0 0 0 0 0 0 0 ]  
 [ 0 1009 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135
```

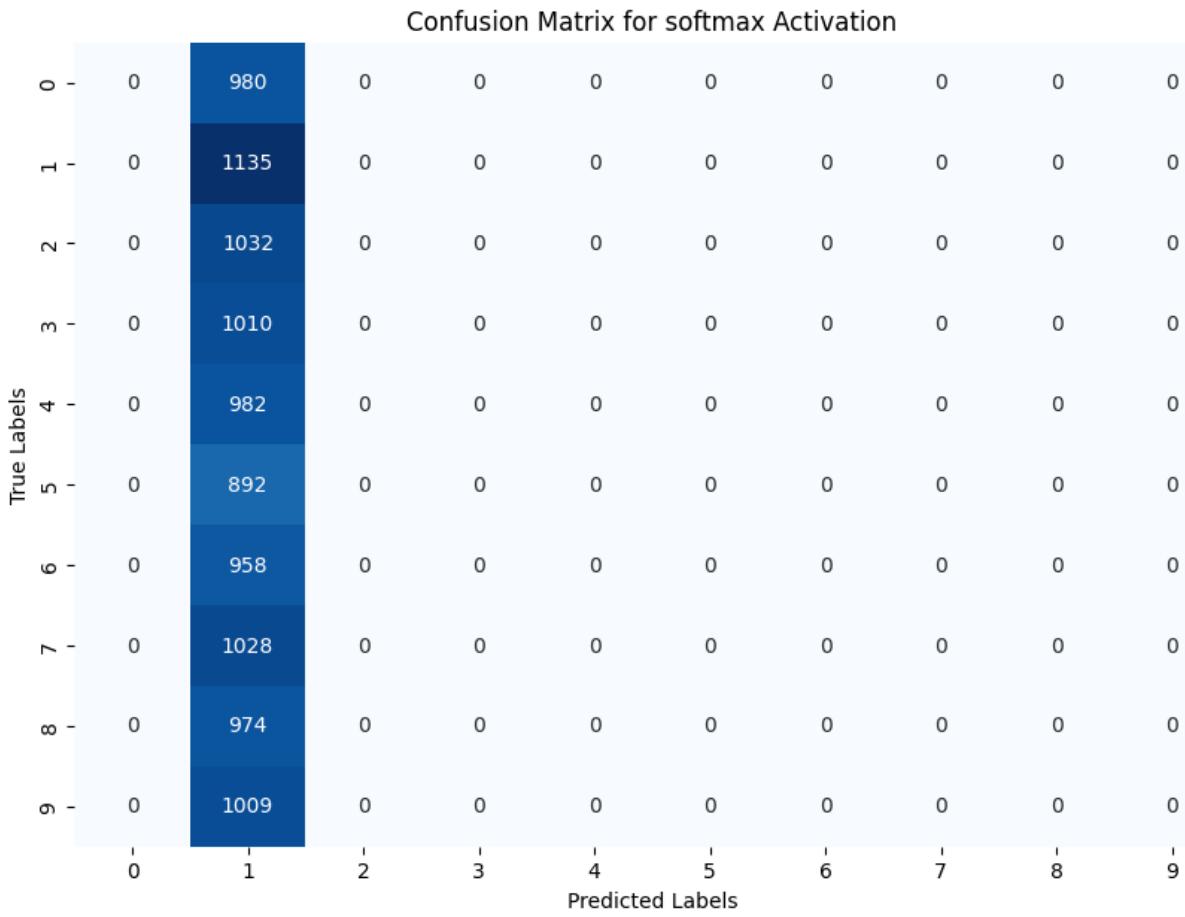


```
Training Model with softmax activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 62s - loss: 2.3019 - accuracy: 0.1125 - val_loss: 2.3025 - val_accuracy: 0.1050 - 62s/epoch - 36ms/step
Epoch 2/10
1688/1688 - 63s - loss: 2.3019 - accuracy: 0.1119 - val_loss: 2.3030 - val_accuracy: 0.1050 - 63s/epoch - 37ms/step
Epoch 3/10
1688/1688 - 63s - loss: 2.3019 - accuracy: 0.1126 - val_loss: 2.3032 - val_accuracy: 0.1050 - 63s/epoch - 37ms/step
Epoch 4/10
1688/1688 - 63s - loss: 2.3019 - accuracy: 0.1120 - val_loss: 2.3022 - val_accuracy: 0.1050 - 63s/epoch - 37ms/step
Epoch 5/10
1688/1688 - 64s - loss: 2.3018 - accuracy: 0.1126 - val_loss: 2.3035 - val_accuracy: 0.1050 - 64s/epoch - 38ms/step
Epoch 6/10
1688/1688 - 61s - loss: 2.3019 - accuracy: 0.1123 - val_loss: 2.3017 - val_accuracy: 0.1050 - 61s/epoch - 36ms/step
Epoch 7/10
```

```
1688/1688 - 62s - loss: 2.3018 - accuracy: 0.1122 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 62s/epoch - 37ms/step  
Epoch 8/10  
1688/1688 - 62s - loss: 2.3017 - accuracy: 0.1118 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 62s/epoch - 36ms/step  
Epoch 9/10  
1688/1688 - 63s - loss: 2.3019 - accuracy: 0.1123 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 63s/epoch - 37ms/step  
Epoch 10/10  
1688/1688 - 64s - loss: 2.3019 - accuracy: 0.1128 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 64s/epoch - 38ms/step  
313/313 [=====] - 8s 24ms/step

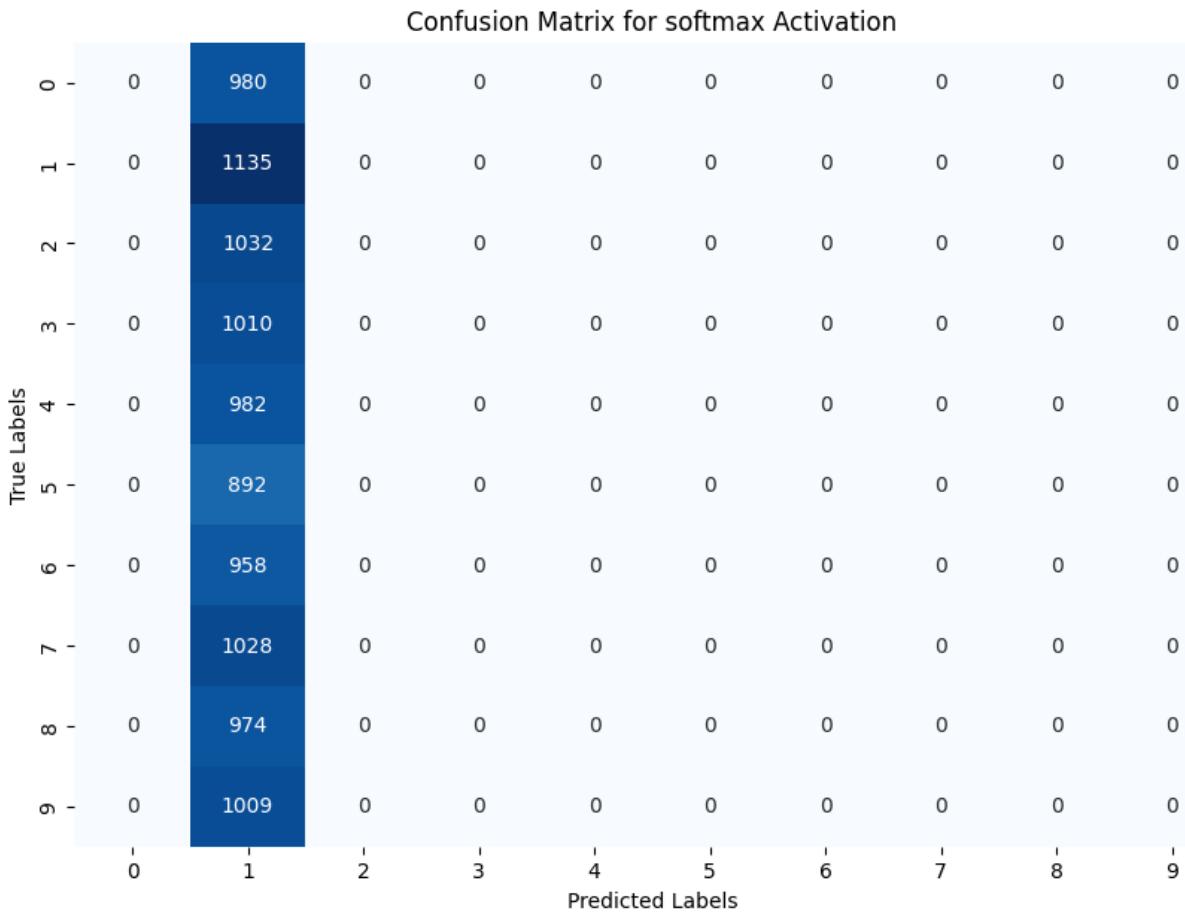
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))

Results for activation function: softmax
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
 [ 0 1135 0 0 0 0 0 0 0 0]  
 [ 0 1032 0 0 0 0 0 0 0 0]  
 [ 0 1010 0 0 0 0 0 0 0 0]  
 [ 0 982 0 0 0 0 0 0 0 0]  
 [ 0 892 0 0 0 0 0 0 0 0]  
 [ 0 958 0 0 0 0 0 0 0 0]  
 [ 0 1028 0 0 0 0 0 0 0 0]  
 [ 0 974 0 0 0 0 0 0 0 0]  
 [ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135
```



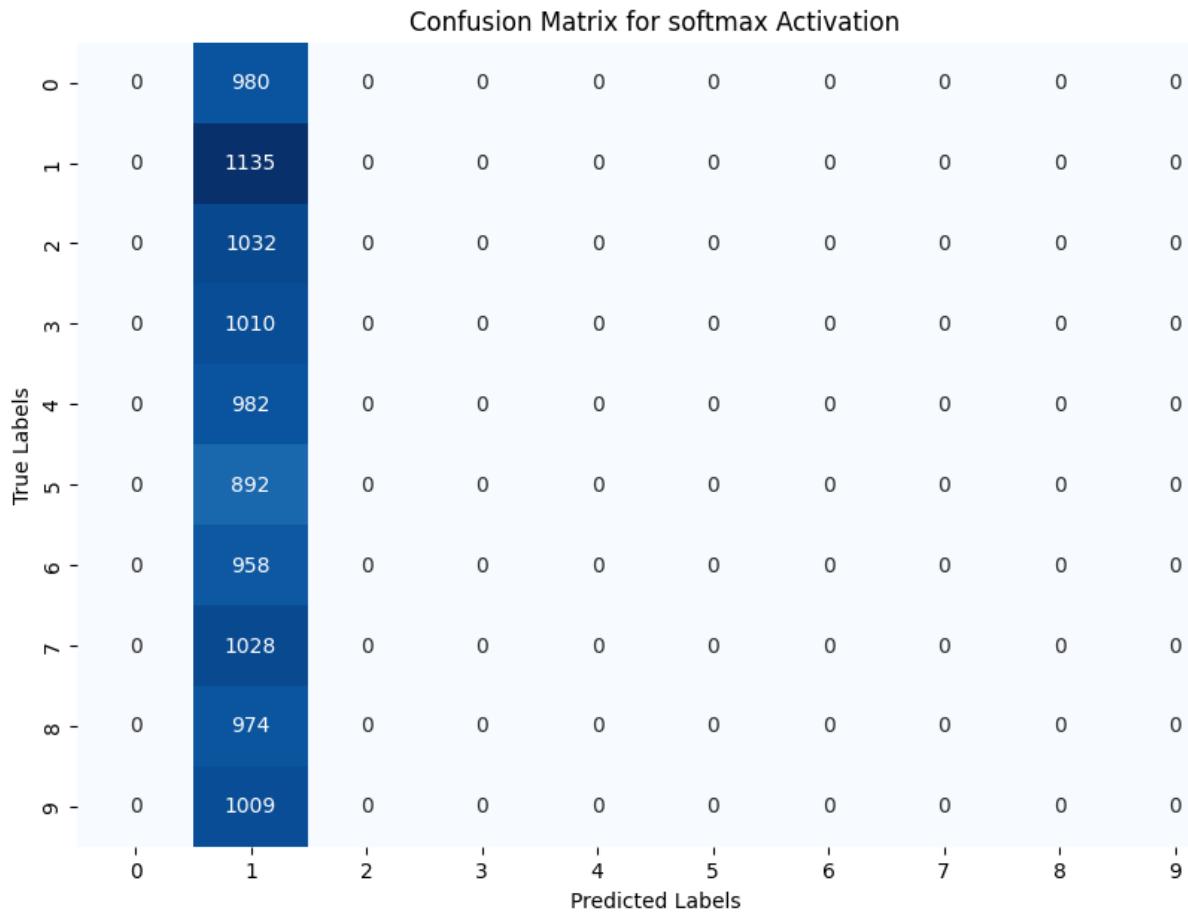
```
Training Model with softmax activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 60s - loss: 2.3020 - accuracy: 0.1116 - val_loss: 2.3020 - val_accuracy: 0.1113 - 60s/epoch - 36ms/step
Epoch 2/10
1688/1688 - 59s - loss: 2.3018 - accuracy: 0.1116 - val_loss: 2.3030 - val_accuracy: 0.1050 - 59s/epoch - 35ms/step
Epoch 3/10
1688/1688 - 59s - loss: 2.3018 - accuracy: 0.1116 - val_loss: 2.3029 - val_accuracy: 0.1050 - 59s/epoch - 35ms/step
Epoch 4/10
1688/1688 - 59s - loss: 2.3017 - accuracy: 0.1117 - val_loss: 2.3033 - val_accuracy: 0.1050 - 59s/epoch - 35ms/step
Epoch 5/10
1688/1688 - 60s - loss: 2.3018 - accuracy: 0.1121 - val_loss: 2.3022 - val_accuracy: 0.1113 - 60s/epoch - 35ms/step
Epoch 6/10
1688/1688 - 58s - loss: 2.3019 - accuracy: 0.1125 - val_loss: 2.3036 - val_accuracy: 0.1050 - 58s/epoch - 34ms/step
Epoch 7/10
```

```
1688/1688 - 58s - loss: 2.3018 - accuracy: 0.1124 - val_loss: 2.3028 -  
val_accuracy: 0.1050 - 58s/epoch - 34ms/step  
Epoch 8/10  
1688/1688 - 60s - loss: 2.3019 - accuracy: 0.1122 - val_loss: 2.3030 -  
val_accuracy: 0.1050 - 60s/epoch - 35ms/step  
Epoch 9/10  
1688/1688 - 59s - loss: 2.3019 - accuracy: 0.1126 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 59s/epoch - 35ms/step  
Epoch 10/10  
1688/1688 - 58s - loss: 2.3018 - accuracy: 0.1122 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 58s/epoch - 34ms/step  
313/313 [=====] - 7s 21ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
`zero_division` parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 59s - loss: 2.3017 - accuracy: 0.1116 - val_loss: 2.3023 - val_accuracy: 0.1050 - 59s/epoch - 35ms/step
Epoch 2/10
1688/1688 - 61s - loss: 2.3019 - accuracy: 0.1124 - val_loss: 2.3027 - val_accuracy: 0.1050 - 61s/epoch - 36ms/step
Epoch 3/10
1688/1688 - 59s - loss: 2.3019 - accuracy: 0.1120 - val_loss: 2.3020 - val_accuracy: 0.1050 - 59s/epoch - 35ms/step
Epoch 4/10
1688/1688 - 58s - loss: 2.3019 - accuracy: 0.1131 - val_loss: 2.3031 - val_accuracy: 0.1050 - 58s/epoch - 35ms/step
Epoch 5/10
1688/1688 - 59s - loss: 2.3019 - accuracy: 0.1127 - val_loss: 2.3021 - val_accuracy: 0.1050 - 59s/epoch - 35ms/step
Epoch 6/10
1688/1688 - 58s - loss: 2.3017 - accuracy: 0.1127 - val_loss: 2.3039 - val_accuracy: 0.1050 - 58s/epoch - 35ms/step
Epoch 7/10
```

```
1688/1688 - 60s - loss: 2.3020 - accuracy: 0.1129 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 60s/epoch - 35ms/step  
Epoch 8/10  
1688/1688 - 60s - loss: 2.3019 - accuracy: 0.1113 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 60s/epoch - 35ms/step  
Epoch 9/10  
1688/1688 - 59s - loss: 2.3020 - accuracy: 0.1116 - val_loss: 2.3032 -  
val_accuracy: 0.1050 - 59s/epoch - 35ms/step  
Epoch 10/10  
1688/1688 - 60s - loss: 2.3019 - accuracy: 0.1123 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 60s/epoch - 36ms/step  
313/313 [=====] - 6s 18ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
`zero_division` parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]  
 [ 0 1135 0 0 0 0 0 0 0 ]  
 [ 0 1032 0 0 0 0 0 0 0 ]  
 [ 0 1010 0 0 0 0 0 0 0 ]  
 [ 0 982 0 0 0 0 0 0 0 ]  
 [ 0 892 0 0 0 0 0 0 0 ]  
 [ 0 958 0 0 0 0 0 0 0 ]  
 [ 0 1028 0 0 0 0 0 0 0 ]  
 [ 0 974 0 0 0 0 0 0 0 ]  
 [ 0 1009 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135
```

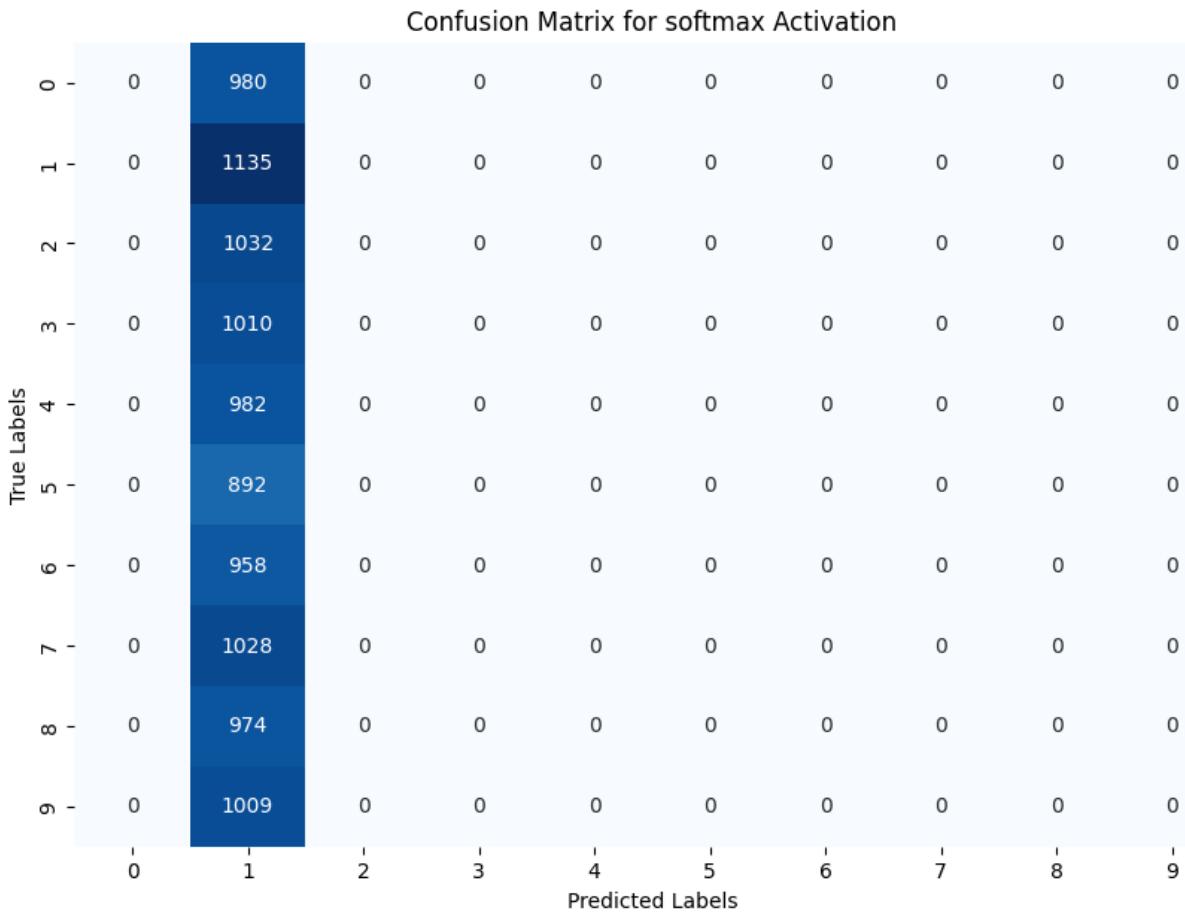


```

Training Model with softmax activation, 3 conv_layers, 3 dense
layers..
Epoch 1/10
1688/1688 - 62s - loss: 2.3017 - accuracy: 0.1118 - val_loss: 2.3031 -
val_accuracy: 0.1050 - 62s/epoch - 37ms/step
Epoch 2/10
1688/1688 - 64s - loss: 2.3020 - accuracy: 0.1130 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 64s/epoch - 38ms/step
Epoch 3/10
1688/1688 - 61s - loss: 2.3019 - accuracy: 0.1121 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 61s/epoch - 36ms/step
Epoch 4/10
1688/1688 - 62s - loss: 2.3020 - accuracy: 0.1124 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 62s/epoch - 37ms/step
Epoch 5/10
1688/1688 - 61s - loss: 2.3018 - accuracy: 0.1111 - val_loss: 2.3027 -
val_accuracy: 0.1050 - 61s/epoch - 36ms/step
Epoch 6/10
1688/1688 - 61s - loss: 2.3020 - accuracy: 0.1125 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 61s/epoch - 36ms/step
Epoch 7/10

```

```
1688/1688 - 61s - loss: 2.3019 - accuracy: 0.1114 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 61s/epoch - 36ms/step  
Epoch 8/10  
1688/1688 - 62s - loss: 2.3018 - accuracy: 0.1118 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 62s/epoch - 37ms/step  
Epoch 9/10  
1688/1688 - 60s - loss: 2.3019 - accuracy: 0.1123 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 60s/epoch - 35ms/step  
Epoch 10/10  
1688/1688 - 64s - loss: 2.3019 - accuracy: 0.1109 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 64s/epoch - 38ms/step  
313/313 [=====] - 7s 20ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
 [ 0 1135 0 0 0 0 0 0 0 0]  
 [ 0 1032 0 0 0 0 0 0 0 0]  
 [ 0 1010 0 0 0 0 0 0 0 0]  
 [ 0 982 0 0 0 0 0 0 0 0]  
 [ 0 892 0 0 0 0 0 0 0 0]  
 [ 0 958 0 0 0 0 0 0 0 0]  
 [ 0 1028 0 0 0 0 0 0 0 0]  
 [ 0 974 0 0 0 0 0 0 0 0]  
 [ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135
```

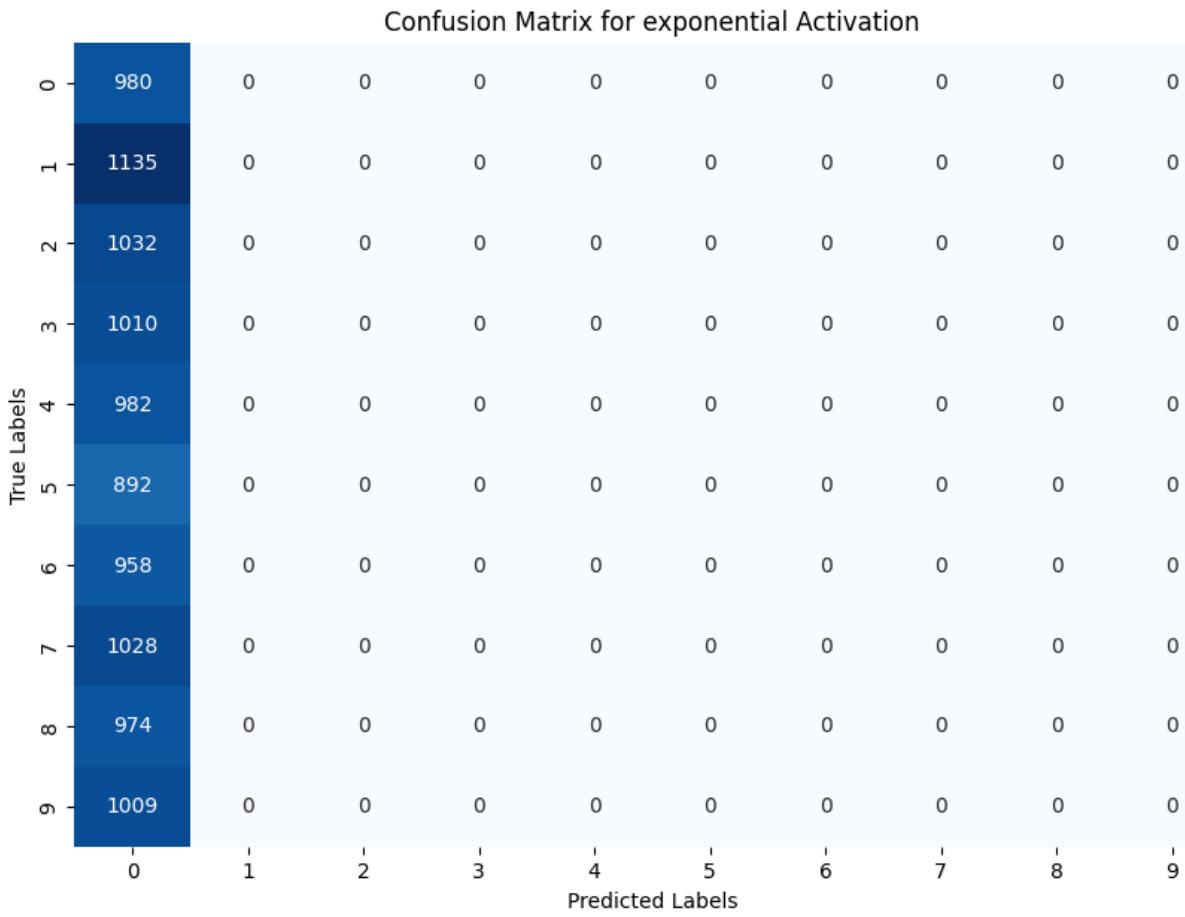


```
Training Model with exponential activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 34s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 34s/epoch - 20ms/step
Epoch 2/10
1688/1688 - 32s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 32s/epoch - 19ms/step
Epoch 3/10
1688/1688 - 34s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 34s/epoch - 20ms/step
Epoch 4/10
1688/1688 - 32s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 32s/epoch - 19ms/step
Epoch 5/10
1688/1688 - 34s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 34s/epoch - 20ms/step
Epoch 6/10
1688/1688 - 32s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 32s/epoch - 19ms/step
Epoch 7/10
```

```
1688/1688 - 32s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 32s/epoch - 19ms/step  
Epoch 8/10  
1688/1688 - 34s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 34s/epoch - 20ms/step  
Epoch 9/10  
1688/1688 - 32s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 32s/epoch - 19ms/step  
Epoch 10/10  
1688/1688 - 34s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 34s/epoch - 20ms/step  
313/313 [=====] - 4s 11ms/step  
Results for activation function: exponential  
Confusion Matrix:  
[[ 980  0  0  0  0  0  0  0  0  0]  
[1135  0  0  0  0  0  0  0  0  0]  
[1032  0  0  0  0  0  0  0  0  0]  
[1010  0  0  0  0  0  0  0  0  0]  
[ 982  0  0  0  0  0  0  0  0  0]  
[ 892  0  0  0  0  0  0  0  0  0]  
[ 958  0  0  0  0  0  0  0  0  0]  
[1028  0  0  0  0  0  0  0  0  0]  
[ 974  0  0  0  0  0  0  0  0  0]  
[1009  0  0  0  0  0  0  0  0  0]]  
Precision: 0.0096  
Recall: 0.0980  

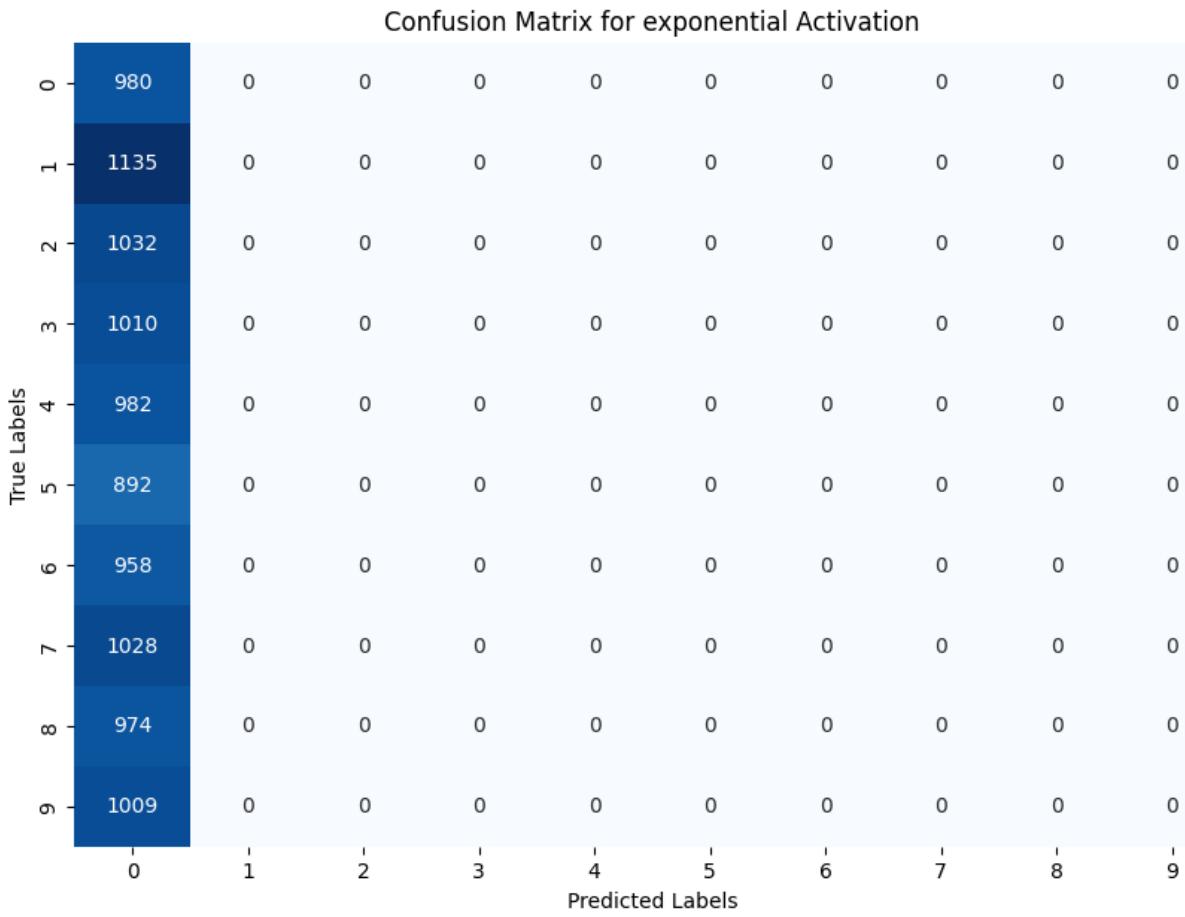

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```



```
Training Model with exponential activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 40s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 40s/epoch - 24ms/step
Epoch 2/10
1688/1688 - 38s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 38s/epoch - 23ms/step
Epoch 3/10
1688/1688 - 38s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 38s/epoch - 22ms/step
Epoch 4/10
1688/1688 - 39s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 39s/epoch - 23ms/step
Epoch 5/10
1688/1688 - 38s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 38s/epoch - 23ms/step
Epoch 6/10
1688/1688 - 37s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 37s/epoch - 22ms/step
Epoch 7/10
```

```
1688/1688 - 38s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 38s/epoch - 22ms/step  
Epoch 8/10  
1688/1688 - 39s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 39s/epoch - 23ms/step  
Epoch 9/10  
1688/1688 - 40s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 40s/epoch - 24ms/step  
Epoch 10/10  
1688/1688 - 39s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 39s/epoch - 23ms/step  
313/313 [=====] - 5s 14ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: exponential  
Confusion Matrix:  
[[ 980  0  0  0  0  0  0  0  0  0]  
 [1135  0  0  0  0  0  0  0  0  0]  
 [1032  0  0  0  0  0  0  0  0  0]  
 [1010  0  0  0  0  0  0  0  0  0]  
 [ 982  0  0  0  0  0  0  0  0  0]  
 [ 892  0  0  0  0  0  0  0  0  0]  
 [ 958  0  0  0  0  0  0  0  0  0]  
 [1028  0  0  0  0  0  0  0  0  0]  
 [ 974  0  0  0  0  0  0  0  0  0]  
 [1009  0  0  0  0  0  0  0  0  0]]  
Precision: 0.0096  
Recall: 0.0980
```



```
Training Model with exponential activation, 1 conv_layers, 1 dense layers..
Epoch 1/10
1688/1688 - 54s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 54s/epoch - 32ms/step
Epoch 2/10
1688/1688 - 54s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 54s/epoch - 32ms/step
Epoch 3/10
1688/1688 - 54s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 54s/epoch - 32ms/step
Epoch 4/10
1688/1688 - 54s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 54s/epoch - 32ms/step
Epoch 5/10
1688/1688 - 54s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 54s/epoch - 32ms/step
Epoch 6/10
1688/1688 - 55s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 55s/epoch - 32ms/step
Epoch 7/10
```

```
1688/1688 - 54s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 54s/epoch - 32ms/step
```

```
Epoch 8/10
```

```
1688/1688 - 53s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 53s/epoch - 31ms/step
```

```
Epoch 9/10
```

```
1688/1688 - 53s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 53s/epoch - 31ms/step
```

```
Epoch 10/10
```

```
1688/1688 - 53s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 53s/epoch - 31ms/step
```

```
313/313 [=====] - 4s 14ms/step
```

```
Results for activation function: exponential
```

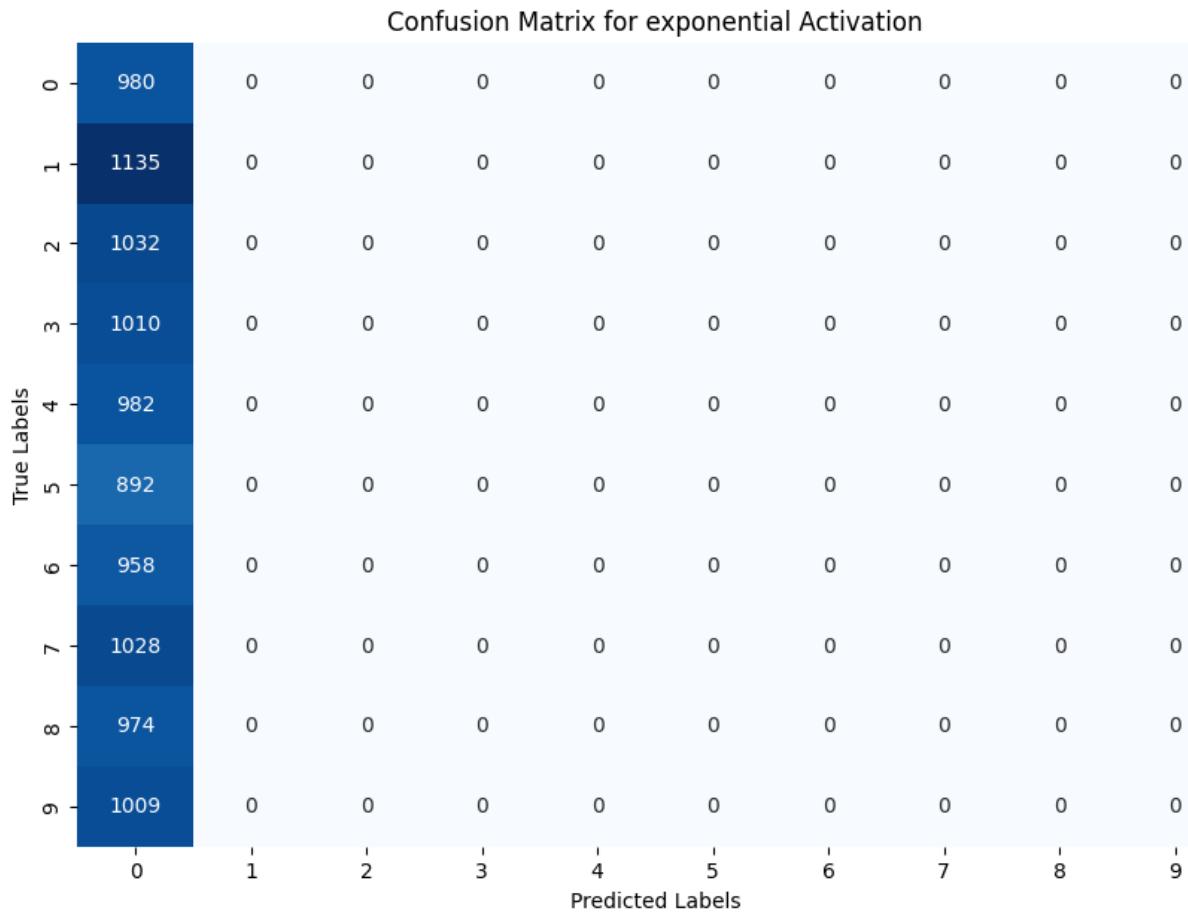
```
Confusion Matrix:
```

```
[[ 980   0   0   0   0   0   0   0   0   0]  
[1135   0   0   0   0   0   0   0   0   0]  
[1032   0   0   0   0   0   0   0   0   0]  
[1010   0   0   0   0   0   0   0   0   0]  
[ 982   0   0   0   0   0   0   0   0   0]  
[ 892   0   0   0   0   0   0   0   0   0]  
[ 958   0   0   0   0   0   0   0   0   0]  
[1028   0   0   0   0   0   0   0   0   0]  
[ 974   0   0   0   0   0   0   0   0   0]  
[1009   0   0   0   0   0   0   0   0   0]]
```

```
Precision: 0.0096
```

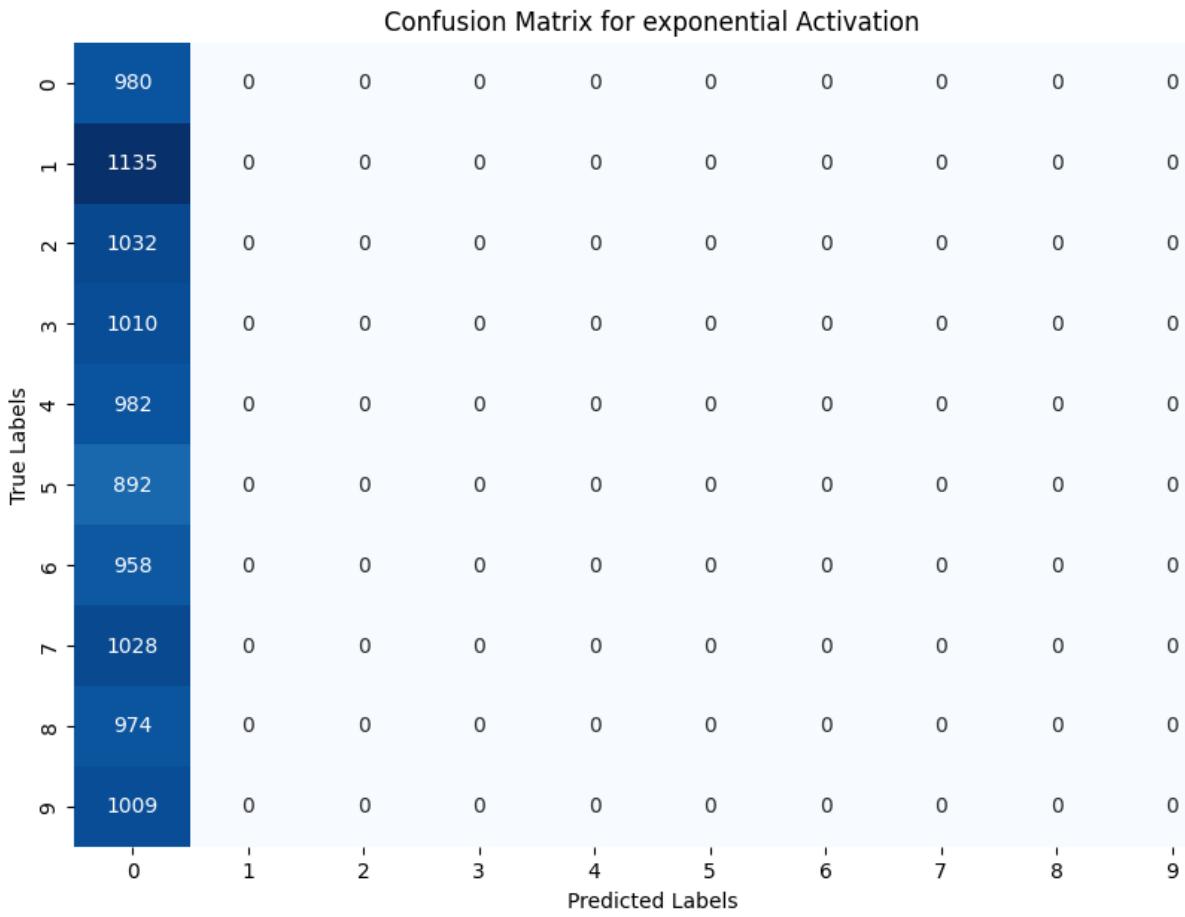
```
Recall: 0.0980
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with exponential activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 50s/epoch - 30ms/step
Epoch 2/10
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 48s/epoch - 28ms/step
Epoch 3/10
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 49s/epoch - 29ms/step
Epoch 4/10
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 49s/epoch - 29ms/step
Epoch 5/10
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 48s/epoch - 29ms/step
Epoch 6/10
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 49s/epoch - 29ms/step
Epoch 7/10
```

```
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 48s/epoch - 28ms/step  
Epoch 8/10  
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 48s/epoch - 29ms/step  
Epoch 9/10  
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 48s/epoch - 28ms/step  
Epoch 10/10  
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 48s/epoch - 29ms/step  
313/313 [=====] - 5s 16ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: exponential  
Confusion Matrix:  
[[ 980  0  0  0  0  0  0  0  0  0]  
 [1135  0  0  0  0  0  0  0  0  0]  
 [1032  0  0  0  0  0  0  0  0  0]  
 [1010  0  0  0  0  0  0  0  0  0]  
 [ 982  0  0  0  0  0  0  0  0  0]  
 [ 892  0  0  0  0  0  0  0  0  0]  
 [ 958  0  0  0  0  0  0  0  0  0]  
 [1028  0  0  0  0  0  0  0  0  0]  
 [ 974  0  0  0  0  0  0  0  0  0]  
 [1009  0  0  0  0  0  0  0  0  0]]  
Precision: 0.0096  
Recall: 0.0980
```

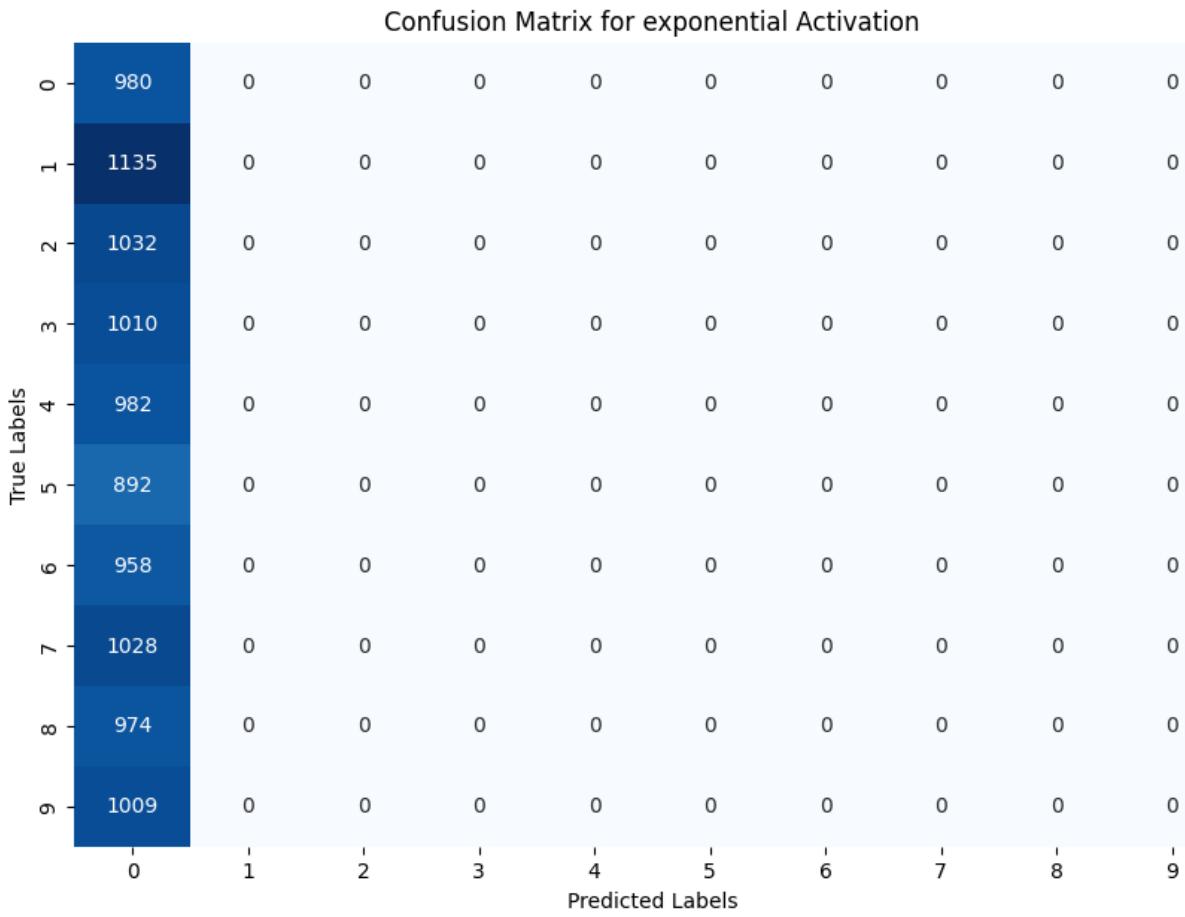


```

Training Model with exponential activation, 2 conv_layers, 2 dense
layers..
Epoch 1/10
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 49s/epoch - 29ms/step
Epoch 2/10
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 50s/epoch - 30ms/step
Epoch 3/10
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 49s/epoch - 29ms/step
Epoch 4/10
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 50s/epoch - 30ms/step
Epoch 5/10
1688/1688 - 51s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 51s/epoch - 30ms/step
Epoch 6/10
1688/1688 - 51s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 51s/epoch - 30ms/step
Epoch 7/10

```

```
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 50s/epoch - 30ms/step  
Epoch 8/10  
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 52s/epoch - 31ms/step  
Epoch 9/10  
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 50s/epoch - 29ms/step  
Epoch 10/10  
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 49s/epoch - 29ms/step  
313/313 [=====] - 5s 16ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: exponential  
Confusion Matrix:  
[[ 980  0  0  0  0  0  0  0  0  0]  
 [1135  0  0  0  0  0  0  0  0  0]  
 [1032  0  0  0  0  0  0  0  0  0]  
 [1010  0  0  0  0  0  0  0  0  0]  
 [ 982  0  0  0  0  0  0  0  0  0]  
 [ 892  0  0  0  0  0  0  0  0  0]  
 [ 958  0  0  0  0  0  0  0  0  0]  
 [1028  0  0  0  0  0  0  0  0  0]  
 [ 974  0  0  0  0  0  0  0  0  0]  
 [1009  0  0  0  0  0  0  0  0  0]]  
Precision: 0.0096  
Recall: 0.0980
```



```
Training Model with exponential activation, 2 conv_layers, 2 dense layers..
Epoch 1/10
1688/1688 - 54s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 54s/epoch - 32ms/step
Epoch 2/10
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 52s/epoch - 31ms/step
Epoch 3/10
1688/1688 - 53s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 53s/epoch - 31ms/step
Epoch 4/10
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 52s/epoch - 31ms/step
Epoch 5/10
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 52s/epoch - 31ms/step
Epoch 6/10
1688/1688 - 53s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 53s/epoch - 31ms/step
Epoch 7/10
```

```
1688/1688 - 53s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 53s/epoch - 32ms/step
```

```
Epoch 8/10
```

```
1688/1688 - 54s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 54s/epoch - 32ms/step
```

```
Epoch 9/10
```

```
1688/1688 - 53s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 53s/epoch - 31ms/step
```

```
Epoch 10/10
```

```
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 52s/epoch - 31ms/step
```

```
313/313 [=====] - 4s 14ms/step
```

```
Results for activation function: exponential
```

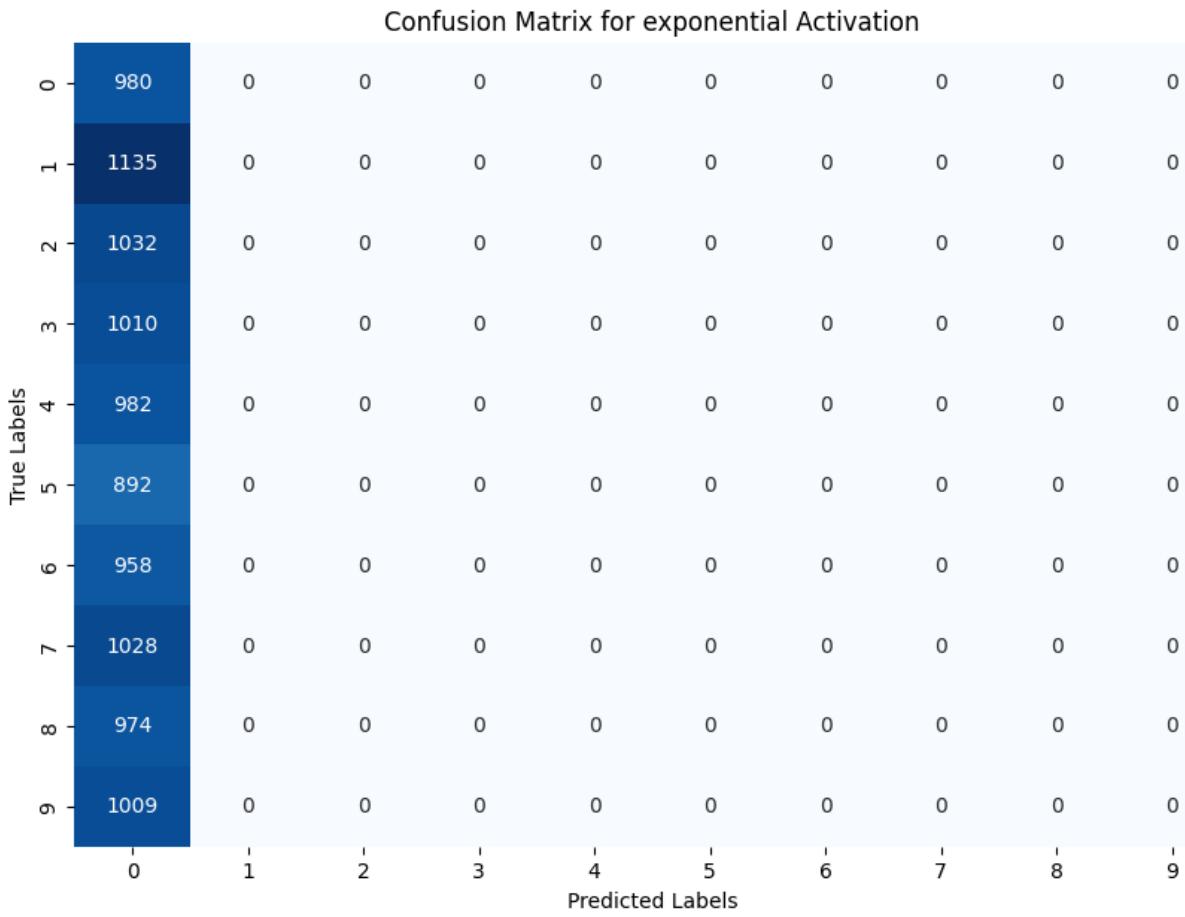
```
Confusion Matrix:
```

```
[[ 980   0   0   0   0   0   0   0   0   0]  
[1135   0   0   0   0   0   0   0   0   0]  
[1032   0   0   0   0   0   0   0   0   0]  
[1010   0   0   0   0   0   0   0   0   0]  
[ 982   0   0   0   0   0   0   0   0   0]  
[ 892   0   0   0   0   0   0   0   0   0]  
[ 958   0   0   0   0   0   0   0   0   0]  
[1028   0   0   0   0   0   0   0   0   0]  
[ 974   0   0   0   0   0   0   0   0   0]  
[1009   0   0   0   0   0   0   0   0   0]]
```

```
Precision: 0.0096
```

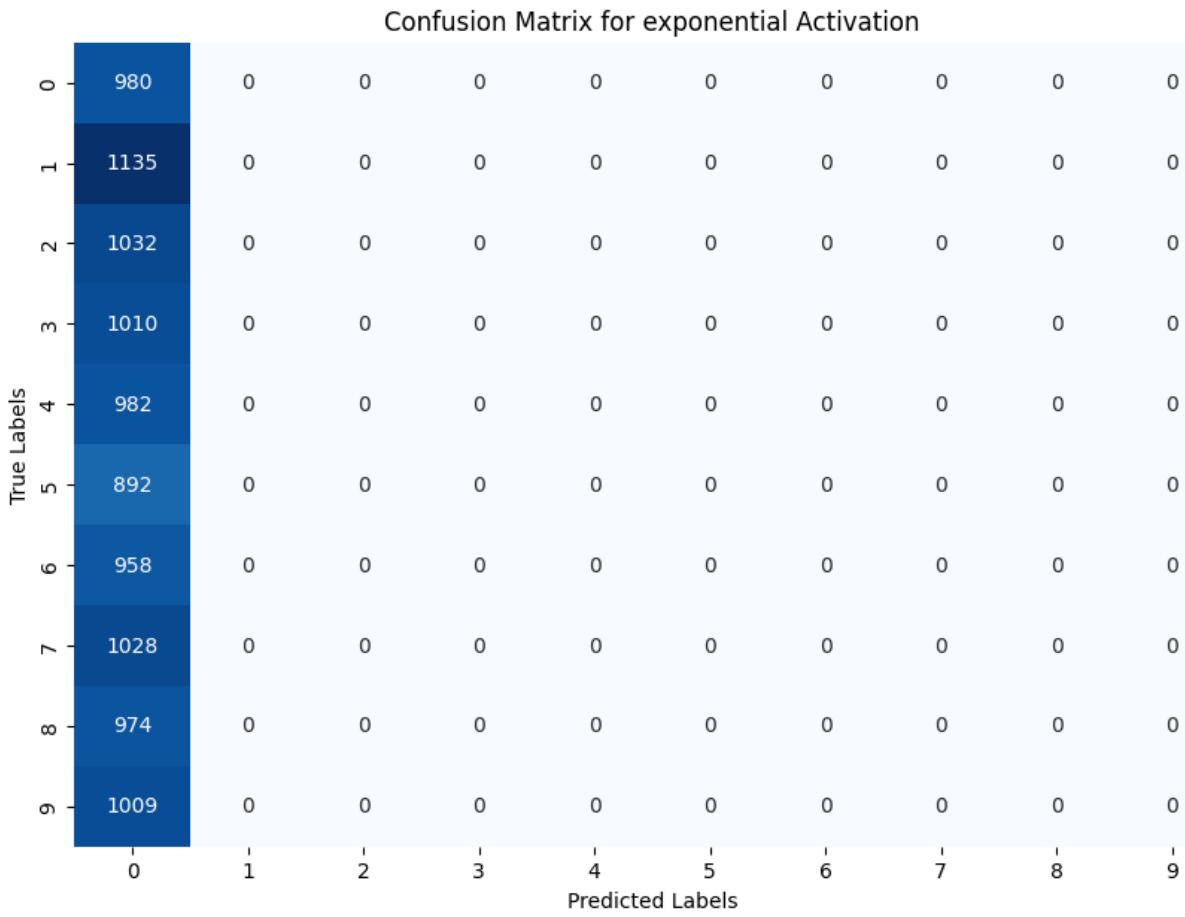
```
Recall: 0.0980
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with exponential activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 51s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 51s/epoch - 30ms/step
Epoch 2/10
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 49s/epoch - 29ms/step
Epoch 3/10
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 49s/epoch - 29ms/step
Epoch 4/10
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 50s/epoch - 30ms/step
Epoch 5/10
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 48s/epoch - 29ms/step
Epoch 6/10
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 50s/epoch - 29ms/step
Epoch 7/10
```

```
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 50s/epoch - 30ms/step  
Epoch 8/10  
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 49s/epoch - 29ms/step  
Epoch 9/10  
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 49s/epoch - 29ms/step  
Epoch 10/10  
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 48s/epoch - 28ms/step  
313/313 [=====] - 5s 15ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: exponential  
Confusion Matrix:  
[[ 980  0  0  0  0  0  0  0  0  0]  
 [1135  0  0  0  0  0  0  0  0  0]  
 [1032  0  0  0  0  0  0  0  0  0]  
 [1010  0  0  0  0  0  0  0  0  0]  
 [ 982  0  0  0  0  0  0  0  0  0]  
 [ 892  0  0  0  0  0  0  0  0  0]  
 [ 958  0  0  0  0  0  0  0  0  0]  
 [1028  0  0  0  0  0  0  0  0  0]  
 [ 974  0  0  0  0  0  0  0  0  0]  
 [1009  0  0  0  0  0  0  0  0  0]]  
Precision: 0.0096  
Recall: 0.0980
```



```
Training Model with exponential activation, 3 conv_layers, 3 dense layers..
Epoch 1/10
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 50s/epoch - 30ms/step
Epoch 2/10
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 50s/epoch - 29ms/step
Epoch 3/10
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 48s/epoch - 29ms/step
Epoch 4/10
1688/1688 - 48s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 48s/epoch - 28ms/step
Epoch 5/10
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 49s/epoch - 29ms/step
Epoch 6/10
1688/1688 - 47s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 47s/epoch - 28ms/step
Epoch 7/10
```

```
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 50s/epoch - 30ms/step
```

```
Epoch 8/10
```

```
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 49s/epoch - 29ms/step
```

```
Epoch 9/10
```

```
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 49s/epoch - 29ms/step
```

```
Epoch 10/10
```

```
1688/1688 - 49s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 49s/epoch - 29ms/step
```

```
313/313 [=====] - 4s 13ms/step
```

```
Results for activation function: exponential
```

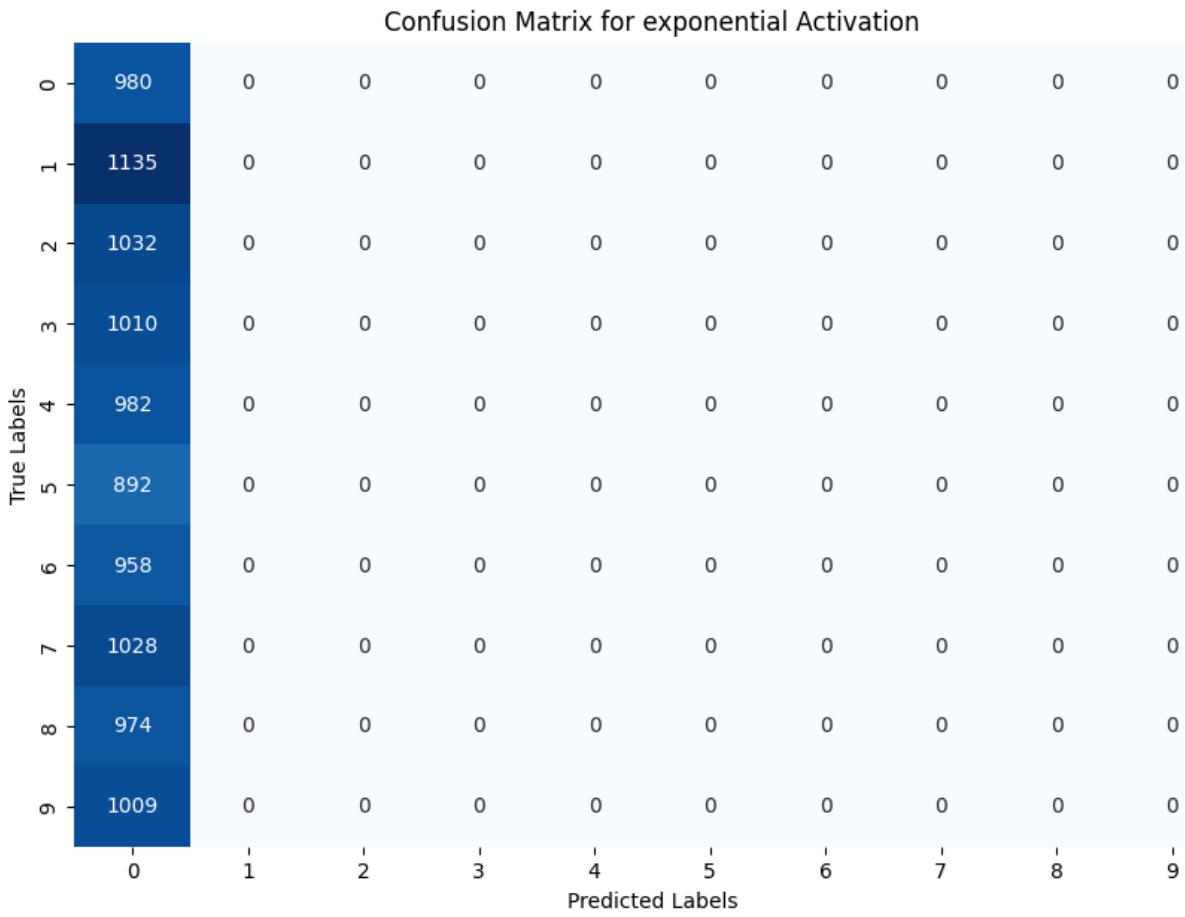
```
Confusion Matrix:
```

```
[[ 980  0  0  0  0  0  0  0  0  0]  
[1135  0  0  0  0  0  0  0  0  0]  
[1032  0  0  0  0  0  0  0  0  0]  
[1010  0  0  0  0  0  0  0  0  0]  
[ 982  0  0  0  0  0  0  0  0  0]  
[ 892  0  0  0  0  0  0  0  0  0]  
[ 958  0  0  0  0  0  0  0  0  0]  
[1028  0  0  0  0  0  0  0  0  0]  
[ 974  0  0  0  0  0  0  0  0  0]  
[1009  0  0  0  0  0  0  0  0  0]]
```

```
Precision: 0.0096
```

```
Recall: 0.0980
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))
```



```

Training Model with exponential activation, 3 conv_layers, 3 dense
layers..
Epoch 1/10
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 52s/epoch - 31ms/step
Epoch 2/10
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 52s/epoch - 31ms/step
Epoch 3/10
1688/1688 - 51s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 51s/epoch - 30ms/step
Epoch 4/10
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 50s/epoch - 30ms/step
Epoch 5/10
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 52s/epoch - 31ms/step
Epoch 6/10
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 52s/epoch - 31ms/step
Epoch 7/10

```

```
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 52s/epoch - 31ms/step  
Epoch 8/10  
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 50s/epoch - 30ms/step  
Epoch 9/10  
1688/1688 - 50s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 50s/epoch - 30ms/step  
Epoch 10/10  
1688/1688 - 52s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 52s/epoch - 31ms/step  
313/313 [=====] - 5s 15ms/step  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))  
  
Results for activation function: exponential  
Confusion Matrix:  
[[ 980  0  0  0  0  0  0  0  0  0]  
 [1135  0  0  0  0  0  0  0  0  0]  
 [1032  0  0  0  0  0  0  0  0  0]  
 [1010  0  0  0  0  0  0  0  0  0]  
 [ 982  0  0  0  0  0  0  0  0  0]  
 [ 892  0  0  0  0  0  0  0  0  0]  
 [ 958  0  0  0  0  0  0  0  0  0]  
 [1028  0  0  0  0  0  0  0  0  0]  
 [ 974  0  0  0  0  0  0  0  0  0]  
 [1009  0  0  0  0  0  0  0  0  0]]  
Precision: 0.0096  
Recall: 0.0980
```

		Confusion Matrix for exponential Activation									
		0	1	2	3	4	5	6	7	8	9
True Labels	0	980	0	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0	0

Key Observations:

ReLU Performance:

Exhibited high precision and recall across all network depths (1, 2, and 3 layers), with peak performance in deeper networks, indicating robustness and efficiency in learning complex patterns.

Sigmoid Activation:

Showed relatively lower precision and recall compared to ReLU, especially in shallower networks, but maintained consistent performance as network depth increased. This suggests a degree of saturation and slower learning, which is characteristic of sigmoid functions.

Tanh Activation:

Demonstrated competitive precision and recall, closely rivaling and sometimes surpassing ReLU, particularly in deeper networks. This highlights Tanh's capability in capturing and processing information efficiently, possibly due to its symmetric output range.

Softmax Activation:

Resulted in significantly lower precision and recall in most configurations, with a notable decrease in networks beyond a single layer. This is an expected outcome since softmax is

primarily suited for the output layer in classification tasks, not as an activation function in inner layers.

Exponential Activation:

Yielded the lowest precision and recall, indicating poor performance across all configurations. The exponential function likely led to exploding gradients, severely hindering the learning process.

Variation with Network Depth:

Increasing the network depth generally improved performance for ReLU and Tanh activations but had mixed impacts on Sigmoid and drastically negative effects on Softmax and Exponential functions. The poor performance of exponential activation across all depths underscores its unsuitability for internal layers due to instability in gradient propagation.

####Experiments with Various Optimizers and Learning Rates

```
optimizers = ["adam", "nadam", "adamw", "adadelta", "rmsprop"]

learning_rate_num = [0.1, 0.01, 0.001, 0.0001]

from tensorflow.keras.optimizers import AdamW, Adam, RMSprop, Adagrad,
Nadam, Adadelta

! pip install tensorflow_addons

Collecting tensorflow_addons
  Downloading tensorflow_addons-0.23.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (611 kB)
                                           611.8/611.8 kB 3.6 MB/s eta
0:00:00
  ent already satisfied: packaging in /usr/local/lib/python3.10/dist-
  packages (from tensorflow_addons) (24.0)
Collecting typeguard<3.0.0,>=2.7 (from tensorflow_addons)
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
Installing collected packages: typeguard, tensorflow_addons
Successfully installed tensorflow_addons-0.23.0 typeguard-2.13.3

import tensorflow_addons as tfa

/usr/local/lib/python3.10/dist-packages/tensorflow_addons/utils/
tfa_eol_msg.py:23: UserWarning:

TensorFlow Addons (TFA) has ended development and introduction of new
features.
TFA has entered a minimal maintenance and release mode until a planned
end of life in May 2024.
Please modify downstream libraries to take dependencies from other
repositories in our TensorFlow community (e.g. Keras, Keras-CV, and
Keras-NLP).
```

For more information see:
<https://github.com/tensorflow/addons/issues/2807>

```
warnings.warn(  
/usr/local/lib/python3.10/dist-packages/tensorflow_addons/utils/ensure  
_tf_install.py:53: UserWarning: Tensorflow Addons supports using  
Python ops for all Tensorflow versions above or equal to 2.13.0 and  
strictly below 2.16.0 (nightly versions are not supported).  
The versions of TensorFlow you are currently using is 2.12.0 and is  
not supported.  
Some things might work, some things might not.  
If you were to encounter a bug, do not file an issue.  
If you want to make sure you're using a tested and supported  
configuration, either change the TensorFlow version or the TensorFlow  
Addons's version.  
You can find the compatibility matrix in TensorFlow Addon's readme:  
https://github.com/tensorflow/addons  
warnings.warn(  
  
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,  
Dense  
from tensorflow.keras.optimizers import Adam, Nadam, Adadelta, RMSprop  
  
# If using AdamW, ensure tensorflow_addons is installed and imported  
import tensorflow_addons as tfa  
  
def baseline_model_opt_lr_exp_2(optimizer_name="adam", lr=0.001):  
    model = Sequential([  
        Conv2D(32, (3, 3), activation="relu",  
kernel_initializer="he_uniform", input_shape=(28, 28, 1)),  
        MaxPooling2D((2, 2)),  
        Flatten(),  
        Dense(100, activation="relu",  
kernel_initializer="he_uniform"),  
        Dense(10, activation="softmax")  
    ])  
  
    # Select the optimizer based on the optimizer_name parameter  
    if optimizer_name.lower() == "adam":  
        opt = Adam(learning_rate=lr)  
    elif optimizer_name.lower() == "nadam":  
        opt = Nadam(learning_rate=lr)  
    elif optimizer_name.lower() == "adadelta":  
        opt = Adadelta(learning_rate=lr)  
    elif optimizer_name.lower() == "rmsprop":  
        opt = RMSprop(learning_rate=lr, momentum=0.9)  
    elif optimizer_name.lower() == "adamw":  
        opt = tfa.optimizers.AdamW(learning_rate=lr,  
weight_decay=0.004)
```

```

else:
    raise ValueError("Unsupported Optimizer")

# Compile the model with the selected optimizer
model.compile(optimizer=opt, loss="categorical_crossentropy",
metrics=["accuracy"])

return model

for opt in optimizers:
    for learn_r in learning_rate_num:
        print(f"Training with {opt} optimizer and the learning_rate is
{learn_r}...")
        model = baseline_model_opt_lr_exp_2(optimizer_name = opt, lr =
learn_r)

        history = model.fit(train_norm, y_train, epochs = 10, batch_size
= 32, validation_split = 0.1, verbose = 2)

        ###predictions
        y_pred_prob = model.predict(test_norm)

        y_pred = np.argmax(y_pred_prob, axis = 1)

        ### ensuring y_test is not one-hot encoded for confusion matrix
        if y_test.ndim > 1:
            y_true = np.argmax(y_test, axis = 1)

        else:
            y_true = y_test

        ## calculating the confusion matrix
        cm = confusion_matrix(y_true, y_pred)

        precision = precision_score(y_true, y_pred, average =
"weighted")
        recall = recall_score(y_true, y_pred, average = "weighted")

        ### Printing the confusion matrix

        print("Confusion Matrix:")
        print(cm)
        print(f"Precision: {precision:.4f}")
        print(f"Recall: {recall:.4f}")

        # Plotting the confusion matrix
        plt.figure(figsize=(10, 7))

```

```

sns.heatmap(cm, annot=True, fmt="g", cmap="Blues", cbar=False)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title(f"Confusion Matrix for {opt} optimizer and
{learn_r} learning rate ")
plt.show()

```

Training with adam optimizer and the learning_rate is 0.1...

Epoch 1/10
1688/1688 - 9s - loss: 2.5857 - accuracy: 0.1036 - val_loss: 2.3069 -
val_accuracy: 0.1050 - 9s/epoch - 5ms/step

Epoch 2/10
1688/1688 - 8s - loss: 2.3135 - accuracy: 0.1032 - val_loss: 2.3216 -
val_accuracy: 0.0960 - 8s/epoch - 5ms/step

Epoch 3/10
1688/1688 - 8s - loss: 2.3135 - accuracy: 0.1039 - val_loss: 2.3111 -
val_accuracy: 0.1050 - 8s/epoch - 5ms/step

Epoch 4/10
1688/1688 - 8s - loss: 2.3145 - accuracy: 0.0999 - val_loss: 2.3191 -
val_accuracy: 0.1050 - 8s/epoch - 5ms/step

Epoch 5/10
1688/1688 - 8s - loss: 2.3138 - accuracy: 0.1050 - val_loss: 2.3075 -
val_accuracy: 0.1050 - 8s/epoch - 5ms/step

Epoch 6/10
1688/1688 - 8s - loss: 2.3132 - accuracy: 0.1018 - val_loss: 2.3197 -
val_accuracy: 0.1113 - 8s/epoch - 5ms/step

Epoch 7/10
1688/1688 - 8s - loss: 2.3133 - accuracy: 0.1031 - val_loss: 2.3170 -
val_accuracy: 0.1113 - 8s/epoch - 5ms/step

Epoch 8/10
1688/1688 - 8s - loss: 2.3147 - accuracy: 0.1029 - val_loss: 2.3073 -
val_accuracy: 0.0952 - 8s/epoch - 5ms/step

Epoch 9/10
1688/1688 - 8s - loss: 2.3138 - accuracy: 0.1039 - val_loss: 2.3143 -
val_accuracy: 0.1050 - 8s/epoch - 5ms/step

Epoch 10/10
1688/1688 - 8s - loss: 2.3147 - accuracy: 0.1034 - val_loss: 2.3047 -
val_accuracy: 0.1045 - 8s/epoch - 5ms/step

313/313 [=====] - 1s 2ms/step

Confusion Matrix:

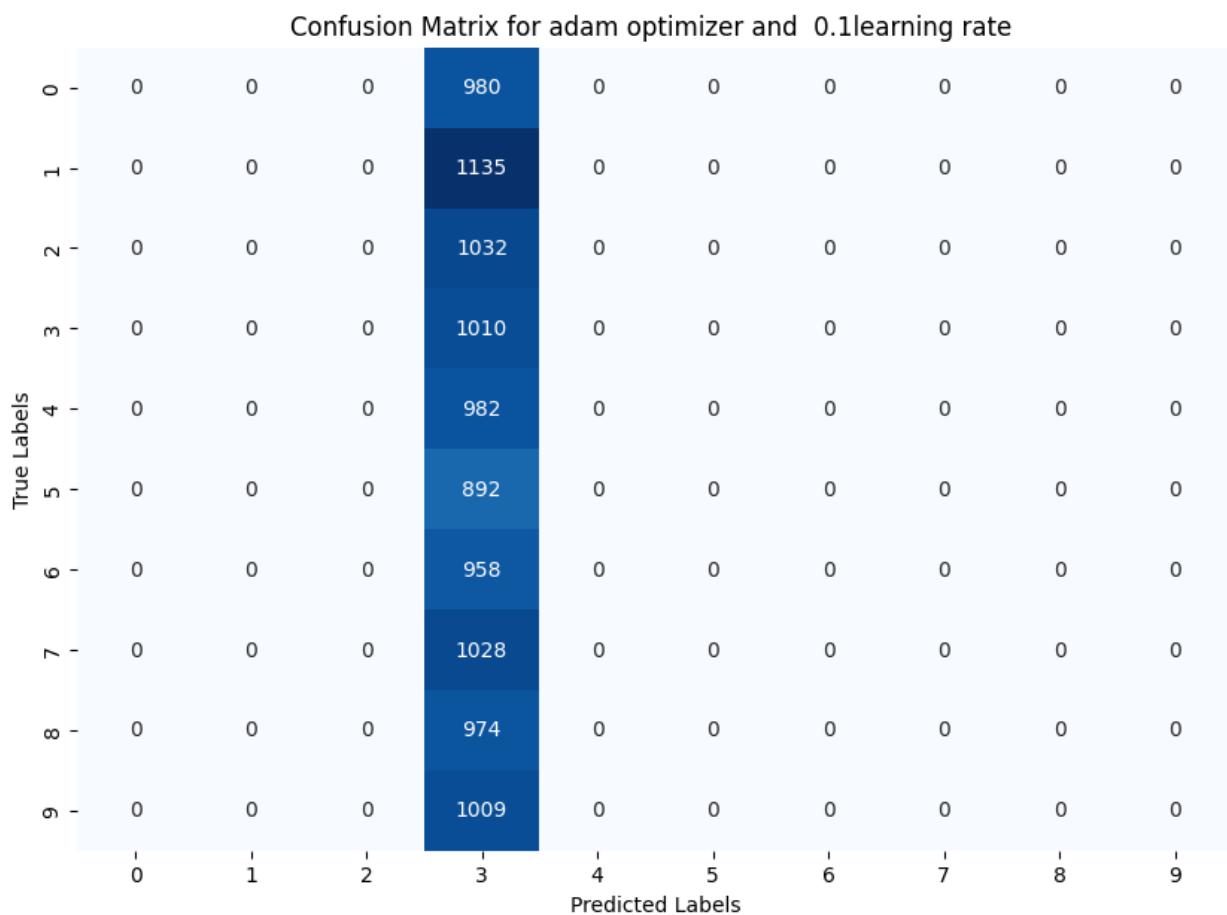
[0	0	0	980	0	0	0	0	0	0]
[0	0	0	1135	0	0	0	0	0	0]
[0	0	0	1032	0	0	0	0	0	0]
[0	0	0	1010	0	0	0	0	0	0]
[0	0	0	982	0	0	0	0	0	0]
[0	0	0	892	0	0	0	0	0	0]
[0	0	0	958	0	0	0	0	0	0]
[0	0	0	1028	0	0	0	0	0	0]

```

[ 0  0  0 974  0  0  0  0  0  0]
[ 0  0  0 1009  0  0  0  0  0  0]]
Precision: 0.0102
Recall: 0.1010

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))

```



```

Training with adam optimizer and the learning_rate is 0.01...
Epoch 1/10
1688/1688 - 9s - loss: 0.1769 - accuracy: 0.9540 - val_loss: 0.0675 -
val_accuracy: 0.9808 - 9s/epoch - 5ms/step
Epoch 2/10
1688/1688 - 8s - loss: 0.0684 - accuracy: 0.9797 - val_loss: 0.0926 -
val_accuracy: 0.9760 - 8s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 0.0531 - accuracy: 0.9836 - val_loss: 0.0646 -
val_accuracy: 0.9837 - 8s/epoch - 5ms/step

```

```
Epoch 4/10
1688/1688 - 8s - loss: 0.0485 - accuracy: 0.9856 - val_loss: 0.0882 -
val_accuracy: 0.9802 - 8s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 0.0418 - accuracy: 0.9885 - val_loss: 0.1022 -
val_accuracy: 0.9795 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.0360 - accuracy: 0.9901 - val_loss: 0.1225 -
val_accuracy: 0.9762 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 0.0369 - accuracy: 0.9898 - val_loss: 0.1290 -
val_accuracy: 0.9812 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 0.0342 - accuracy: 0.9918 - val_loss: 0.1276 -
val_accuracy: 0.9813 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 0.0308 - accuracy: 0.9928 - val_loss: 0.1529 -
val_accuracy: 0.9813 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 0.0313 - accuracy: 0.9928 - val_loss: 0.1383 -
val_accuracy: 0.9852 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 974   0   0   0   1   0   3   1   1   0]
 [  0 1127   3   0   0   2   0   3   0   0]
 [  5   1 1011   1   0   0   1   8   5   0]
 [  0   0   2 994   0   5   0   4   3   2]
 [  0   0   1   0 975   0   1   1   0   4]
 [  2   0   2   10   0 863   5   0   8   2]
 [  8   1   0   1   0   8 933   0   7   0]
 [  2   1   10   4   2   0   0 999   4   6]
 [  5   0   0   3   3   0   0   3 956   4]
 [  4   3   2   0 13   3   0   3   2 979]]
```

Precision: 0.9811
Recall: 0.9811

Confusion Matrix for adam optimizer and 0.01learning rate											
	0	1	2	3	4	5	6	7	8	9	
0	974	0	0	0	1	0	3	1	1	0	
1	0	1127	3	0	0	2	0	3	0	0	
2	5	1	1011	1	0	0	1	8	5	0	
3	0	0	2	994	0	5	0	4	3	2	
4	0	0	1	0	975	0	1	1	0	4	
5	2	0	2	10	0	863	5	0	8	2	
6	8	1	0	1	0	8	933	0	7	0	
7	2	1	10	4	2	0	0	999	4	6	
8	5	0	0	3	3	0	0	3	956	4	
9	4	3	2	0	13	3	0	3	2	979	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	

```

Training with adam optimizer and the learning_rate is 0.001...
Epoch 1/10
1688/1688 - 9s - loss: 0.1379 - accuracy: 0.9578 - val_loss: 0.0607 -
val_accuracy: 0.9843 - 9s/epoch - 5ms/step
Epoch 2/10
1688/1688 - 8s - loss: 0.0472 - accuracy: 0.9849 - val_loss: 0.0629 -
val_accuracy: 0.9832 - 8s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 0.0289 - accuracy: 0.9909 - val_loss: 0.0515 -
val_accuracy: 0.9865 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 0.0186 - accuracy: 0.9939 - val_loss: 0.0599 -
val_accuracy: 0.9872 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 0.0125 - accuracy: 0.9959 - val_loss: 0.0543 -
val_accuracy: 0.9873 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.0086 - accuracy: 0.9970 - val_loss: 0.0572 -
val_accuracy: 0.9873 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 0.0082 - accuracy: 0.9972 - val_loss: 0.0638 -

```

```
val_accuracy: 0.9858 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 0.0060 - accuracy: 0.9979 - val_loss: 0.0675 -
val_accuracy: 0.9862 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 0.0049 - accuracy: 0.9984 - val_loss: 0.0652 -
val_accuracy: 0.9862 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 0.0040 - accuracy: 0.9986 - val_loss: 0.0722 -
val_accuracy: 0.9865 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 975   0   0   0   0   0   3   1   1   0]
 [  0 1128   0   3   0   0   3   1   0   0]
 [  1   2 1021   0   1   0   0   1   6   0]
 [  0   0   1 1002   0   4   0   0   2   1]
 [  0   1   1   0  968   0   4   0   2   6]
 [  1   0   0   7   0  877   5   0   1   1]
 [  2   2   0   0   1   2  948   0   3   0]
 [  0   2   12  2   1   0   0 1001   2   8]
 [  3   0   1   1   1   2   2   2  958   4]
 [  0   2   0   1   4   2   0   1   1  998]]
```

Precision: 0.9876
Recall: 0.9876

Confusion Matrix for adam optimizer and 0.001 learning rate										
	0	1	2	3	4	5	6	7	8	9
0	975	0	0	0	0	0	3	1	1	0
1	0	1128	0	3	0	0	3	1	0	0
2	1	2	1021	0	1	0	0	1	6	0
3	0	0	1	1002	0	4	0	0	2	1
4	0	1	1	0	968	0	4	0	2	6
5	1	0	0	7	0	877	5	0	1	1
6	2	2	0	0	1	2	948	0	3	0
7	0	2	12	2	1	0	0	1001	2	8
8	3	0	1	1	1	2	2	2	958	4
9	0	2	0	1	4	2	0	1	1	998
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```

Training with adam optimizer and the learning_rate is 0.0001...
Epoch 1/10
1688/1688 - 9s - loss: 0.2848 - accuracy: 0.9203 - val_loss: 0.1134 - val_accuracy: 0.9700 - 9s/epoch - 5ms/step
Epoch 2/10
1688/1688 - 9s - loss: 0.1080 - accuracy: 0.9705 - val_loss: 0.0759 - val_accuracy: 0.9810 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 0.0735 - accuracy: 0.9797 - val_loss: 0.0836 - val_accuracy: 0.9748 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 8s - loss: 0.0552 - accuracy: 0.9842 - val_loss: 0.0628 - val_accuracy: 0.9825 - 8s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 0.0435 - accuracy: 0.9878 - val_loss: 0.0543 - val_accuracy: 0.9843 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.0359 - accuracy: 0.9900 - val_loss: 0.0495 - val_accuracy: 0.9870 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 0.0298 - accuracy: 0.9916 - val_loss: 0.0531 -

```

```
val_accuracy: 0.9863 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 0.0250 - accuracy: 0.9933 - val_loss: 0.0470 -
val_accuracy: 0.9883 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 0.0212 - accuracy: 0.9942 - val_loss: 0.0516 -
val_accuracy: 0.9865 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 0.0181 - accuracy: 0.9955 - val_loss: 0.0520 -
val_accuracy: 0.9863 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 974   1   0   0   0   1   2   2   0   0]
 [  1 1131   1   1   0   0   0   0   1   0]
 [  0   2 1013   4   1   0   1   8   3   0]
 [  0   0   0 1003   0   2   0   2   3   0]
 [  0   1   0   0  977   0   0   1   0   3]
 [  2   0   0   6   0  879   5   0   0   0]
 [  6   3   1   1   1   4  940   0   2   0]
 [  0   2   4   3   0   0   0 1019   0   0]
 [  3   0   3   5   3   3   0   4  949   4]
 [  2   3   0   6  13   1   0  11   1  972]]
```

Precision: 0.9858
Recall: 0.9857

Confusion Matrix for adam optimizer and 0.0001 learning rate

	0	1	2	3	4	5	6	7	8	9
True Labels	974	1	0	0	0	1	2	2	0	0
0	1	1131	1	1	0	0	0	0	1	0
1	0	2	1013	4	1	0	1	8	3	0
2	0	0	0	1003	0	2	0	2	3	0
3	0	1	0	0	977	0	0	1	0	3
4	2	0	0	6	0	879	5	0	0	0
5	6	3	1	1	1	4	940	0	2	0
6	0	2	4	3	0	0	0	1019	0	0
7	3	0	3	5	3	3	0	4	949	4
8	2	3	0	6	13	1	0	11	1	972
9	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training with nadam optimizer and the learning_rate is 0.1...
Epoch 1/10
1688/1688 - 10s - loss: 1.3203 - accuracy: 0.5841 - val_loss: 2.3128 - val_accuracy: 0.0917 - 10s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 9s - loss: 2.3136 - accuracy: 0.1031 - val_loss: 2.3143 - val_accuracy: 0.1050 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 2.3132 - accuracy: 0.1020 - val_loss: 2.3136 - val_accuracy: 0.1050 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 8s - loss: 2.3130 - accuracy: 0.1004 - val_loss: 2.3302 - val_accuracy: 0.1000 - 8s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 2.3128 - accuracy: 0.1035 - val_loss: 2.3147 - val_accuracy: 0.0952 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 9s - loss: 2.3134 - accuracy: 0.1050 - val_loss: 2.3112 - val_accuracy: 0.1050 - 9s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 2.3126 - accuracy: 0.1034 - val_loss: 2.3326 -
```

```
val_accuracy: 0.0915 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 2.3131 - accuracy: 0.1050 - val_loss: 2.3191 -
val_accuracy: 0.1000 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 9s - loss: 2.3135 - accuracy: 0.1012 - val_loss: 2.3146 -
val_accuracy: 0.1045 - 9s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 2.3122 - accuracy: 0.1041 - val_loss: 2.3117 -
val_accuracy: 0.0995 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 0   0   0   0   0   0   0   0   980   0]
 [ 0   0   0   0   0   0   0   0  1135   0]
 [ 0   0   0   0   0   0   0   0  1032   0]
 [ 0   0   0   0   0   0   0   0  1010   0]
 [ 0   0   0   0   0   0   0   0  982   0]
 [ 0   0   1   0   0   0   0   0  891   0]
 [ 0   0   0   0   0   0   0   0  958   0]
 [ 0   0   0   0   0   0   0   0  1028   0]
 [ 0   0   0   0   0   0   0   0  974   0]
 [ 0   0   0   0   0   0   0   0  1009   0]]
Precision: 0.0095
Recall: 0.0974

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for nadam optimizer and 0.1learning rate											
True Labels	0 -	1	2	3	4	5	6	7	8	9	
0 -	980	0	0	0	0	0	0	0	0	0	0
1 -	1135	0	0	0	0	0	0	0	0	0	0
2 -	1032	0	0	0	0	0	0	0	0	0	0
3 -	1010	0	0	0	0	0	0	0	0	0	0
4 -	982	0	0	0	0	0	0	0	0	0	0
5 -	891	0	0	1	0	0	0	0	0	0	0
6 -	958	0	0	0	0	0	0	0	0	0	0
7 -	1028	0	0	0	0	0	0	0	0	0	0
8 -	974	0	0	0	0	0	0	0	0	0	0
9 -	1009	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

```

Training with nadam optimizer and the learning_rate is 0.01...
Epoch 1/10
1688/1688 - 10s - loss: 0.1456 - accuracy: 0.9578 - val_loss: 0.0973 -
val_accuracy: 0.9697 - 10s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 9s - loss: 0.0685 - accuracy: 0.9791 - val_loss: 0.0634 -
val_accuracy: 0.9818 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 9s - loss: 0.0552 - accuracy: 0.9835 - val_loss: 0.1348 -
val_accuracy: 0.9693 - 9s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 0.0475 - accuracy: 0.9865 - val_loss: 0.0818 -
val_accuracy: 0.9818 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 9s - loss: 0.0428 - accuracy: 0.9886 - val_loss: 0.1326 -
val_accuracy: 0.9817 - 9s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 9s - loss: 0.0446 - accuracy: 0.9882 - val_loss: 0.0992 -
val_accuracy: 0.9825 - 9s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 0.0334 - accuracy: 0.9915 - val_loss: 0.1237 -

```

```
val_accuracy: 0.9815 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 9s - loss: 0.0359 - accuracy: 0.9913 - val_loss: 0.1399 -
val_accuracy: 0.9823 - 9s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 9s - loss: 0.0392 - accuracy: 0.9915 - val_loss: 0.1449 -
val_accuracy: 0.9818 - 9s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 9s - loss: 0.0315 - accuracy: 0.9928 - val_loss: 0.1465 -
val_accuracy: 0.9825 - 9s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 963   0   2   1   0   1   2   1   8   2]
 [  0 1112   3   1   1   2   3   2  11   0]
 [  0   3 1006   1   2   0   2   8  10   0]
 [  0   0   2 993   0   6   0   2   7   0]
 [  0   0   0   0 964   0   1   3   3  11]
 [  3   0   0   7   0 869   6   0   4   3]
 [  3   3   0   0   3   1 944   0   4   0]
 [  0   1   8   1   2   0   0 1006   7   3]
 [  4   0   1   2   2   1   0   0 962   2]
 [  3   0   1   1   4   2   0   6  19 973]]]
Precision: 0.9795
Recall: 0.9792
```

Confusion Matrix for nadam optimizer and 0.01learning rate											
	0	1	2	3	4	5	6	7	8	9	
0	963	0	2	1	0	1	2	1	8	2	
1	0	1112	3	1	1	2	3	2	11	0	
2	0	3	1006	1	2	0	2	8	10	0	
3	0	0	2	993	0	6	0	2	7	0	
4	0	0	0	0	964	0	1	3	3	11	
5	3	0	0	7	0	869	6	0	4	3	
6	3	3	0	0	3	1	944	0	4	0	
7	0	1	8	1	2	0	0	1006	7	3	
8	4	0	1	2	2	1	0	0	962	2	
9	3	0	1	1	4	2	0	6	19	973	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

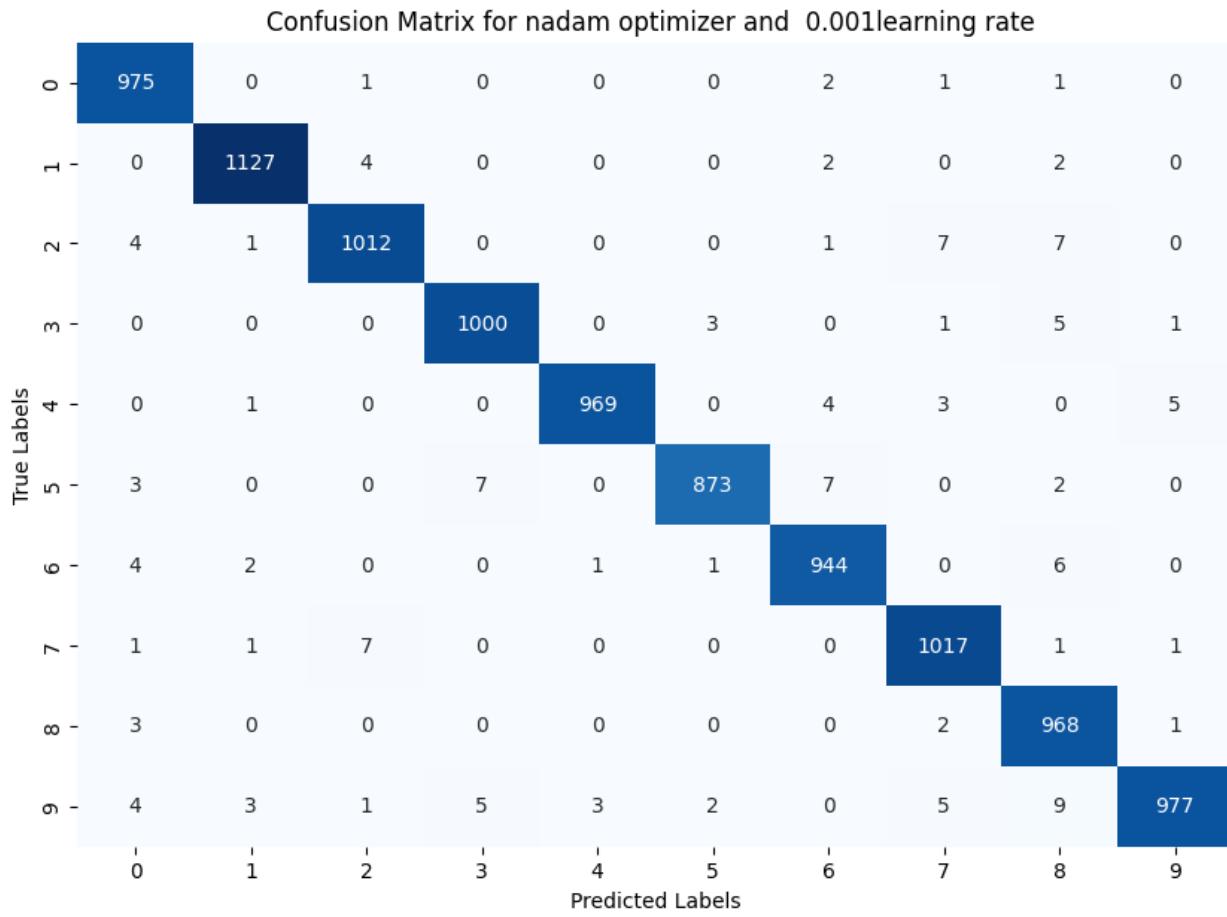
```

Training with nadam optimizer and the learning_rate is 0.001...
Epoch 1/10
1688/1688 - 10s - loss: 0.1419 - accuracy: 0.9564 - val_loss: 0.0624 -
val_accuracy: 0.9835 - 10s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 9s - loss: 0.0473 - accuracy: 0.9855 - val_loss: 0.0648 -
val_accuracy: 0.9832 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 9s - loss: 0.0283 - accuracy: 0.9911 - val_loss: 0.0629 -
val_accuracy: 0.9815 - 9s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 0.0185 - accuracy: 0.9941 - val_loss: 0.0447 -
val_accuracy: 0.9882 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 9s - loss: 0.0126 - accuracy: 0.9958 - val_loss: 0.0532 -
val_accuracy: 0.9877 - 9s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.0083 - accuracy: 0.9973 - val_loss: 0.0514 -
val_accuracy: 0.9898 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 9s - loss: 0.0077 - accuracy: 0.9974 - val_loss: 0.0505 -

```

```
val_accuracy: 0.9882 - 9s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 9s - loss: 0.0051 - accuracy: 0.9984 - val_loss: 0.0676 -
val_accuracy: 0.9847 - 9s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 9s - loss: 0.0051 - accuracy: 0.9983 - val_loss: 0.0607 -
val_accuracy: 0.9872 - 9s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 9s - loss: 0.0038 - accuracy: 0.9987 - val_loss: 0.0664 -
val_accuracy: 0.9897 - 9s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 975   0   1   0   0   0   2   1   1   0]
 [  0 1127   4   0   0   0   2   0   2   0]
 [  4   1 1012   0   0   0   1   7   7   0]
 [  0   0   0 1000   0   3   0   1   5   1]
 [  0   1   0   0  969   0   4   3   0   5]
 [  3   0   0   7   0  873   7   0   2   0]
 [  4   2   0   0   1   1  944   0   6   0]
 [  1   1   7   0   0   0   0 1017   1   1]
 [  3   0   0   0   0   0   0   2  968   1]
 [  4   3   1   5   3   2   0   5   9  977]]
```

Precision: 0.9863
Recall: 0.9862



```

Training with nadam optimizer and the learning_rate is 0.0001...
Epoch 1/10
1688/1688 - 10s - loss: 0.2924 - accuracy: 0.9212 - val_loss: 0.1133 - val_accuracy: 0.9715 - 10s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 9s - loss: 0.1052 - accuracy: 0.9704 - val_loss: 0.0819 - val_accuracy: 0.9778 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 9s - loss: 0.0714 - accuracy: 0.9800 - val_loss: 0.0593 - val_accuracy: 0.9842 - 9s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 0.0532 - accuracy: 0.9854 - val_loss: 0.0599 - val_accuracy: 0.9838 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 0.0426 - accuracy: 0.9886 - val_loss: 0.0532 - val_accuracy: 0.9847 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 9s - loss: 0.0340 - accuracy: 0.9907 - val_loss: 0.0485 - val_accuracy: 0.9862 - 9s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 9s - loss: 0.0279 - accuracy: 0.9927 - val_loss: 0.0485 -

```

```
val_accuracy: 0.9875 - 9s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 9s - loss: 0.0226 - accuracy: 0.9943 - val_loss: 0.0458 -
val_accuracy: 0.9868 - 9s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 9s - loss: 0.0188 - accuracy: 0.9952 - val_loss: 0.0480 -
val_accuracy: 0.9870 - 9s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 9s - loss: 0.0165 - accuracy: 0.9959 - val_loss: 0.0461 -
val_accuracy: 0.9878 - 9s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 967   0   2   0   1   1   5   2   0   2]
 [  0 1130   2   0   0   1   1   1   0   0]
 [  2   2 1017   1   1   0   2   7   0   0]
 [  0   0   3  996   0   5   0   3   3   0]
 [  0   2   2   0  965   0   2   2   0   9]
 [  2   0   0   4   0  882   4   0   0   0]
 [  5   2   1   0   1   5  941   0   3   0]
 [  0   2   7   1   0   0   0 1018   0   0]
 [  4   0   5   3   2   3   0   4  950   3]
 [  0   2   0   5   6   3   0   5   2  986]]]
Precision: 0.9852
Recall: 0.9852
```

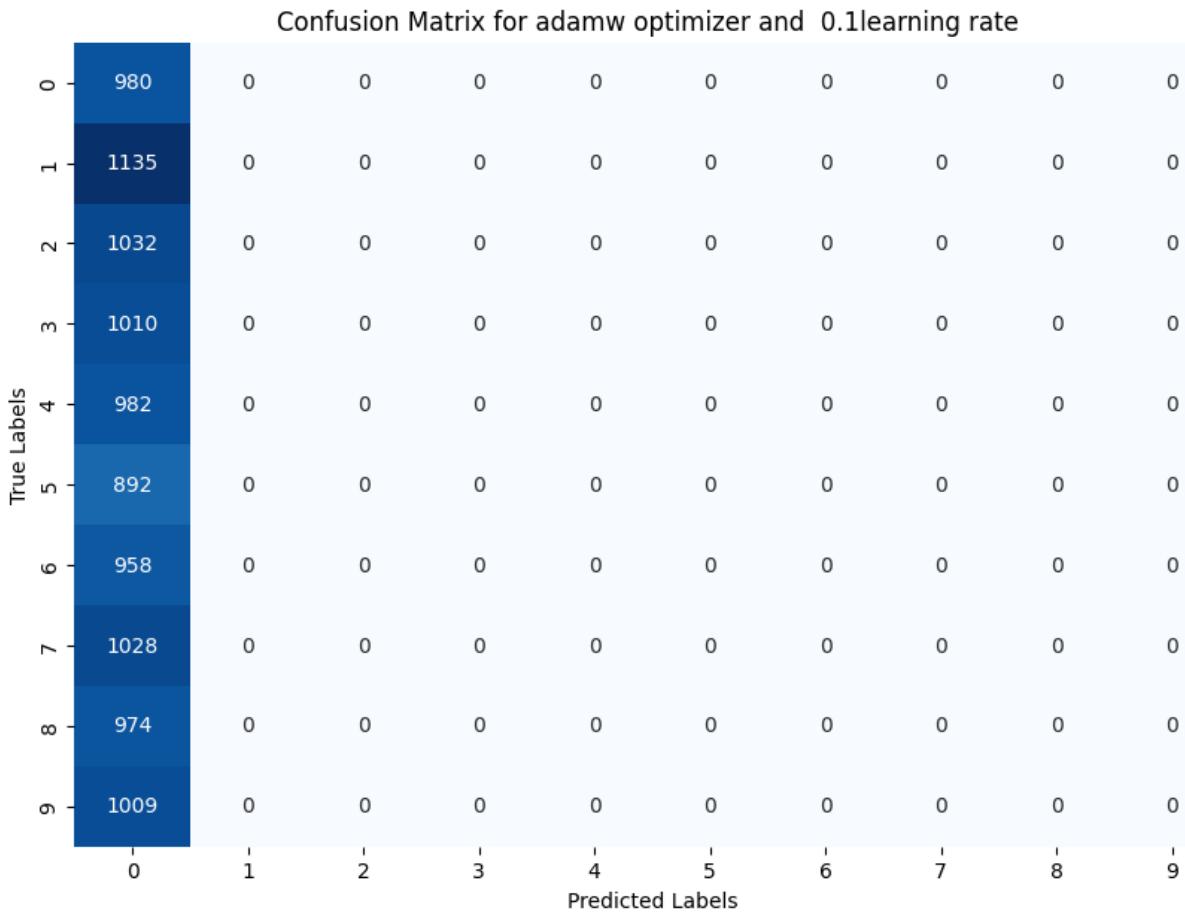
Confusion Matrix for nadam optimizer and 0.0001 learning rate

	0	1	2	3	4	5	6	7	8	9
True Labels	967	0	2	0	1	1	5	2	0	2
0	967	0	2	0	1	1	5	2	0	2
1	0	1130	2	0	0	1	1	1	0	0
2	2	2	1017	1	1	0	2	7	0	0
3	0	0	3	996	0	5	0	3	3	0
4	0	2	2	0	965	0	2	2	0	9
5	2	0	0	4	0	882	4	0	0	0
6	5	2	1	0	1	5	941	0	3	0
7	0	2	7	1	0	0	0	1018	0	0
8	4	0	5	3	2	3	0	4	950	3
9	0	2	0	5	6	3	0	5	2	986
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training with adamw optimizer and the learning_rate is 0.1...
Epoch 1/10
1688/1688 - 8s - loss: 2.4826 - accuracy: 0.1028 - val_loss: 2.3052 - val_accuracy: 0.1050 - 8s/epoch - 5ms/step
Epoch 2/10
1688/1688 - 8s - loss: 2.3135 - accuracy: 0.1043 - val_loss: 2.3057 - val_accuracy: 0.1113 - 8s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 2.3132 - accuracy: 0.1026 - val_loss: 2.3076 - val_accuracy: 0.0992 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 8s - loss: 2.3118 - accuracy: 0.1024 - val_loss: 2.3138 - val_accuracy: 0.0952 - 8s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 2.3132 - accuracy: 0.1028 - val_loss: 2.3091 - val_accuracy: 0.1113 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 2.3123 - accuracy: 0.1044 - val_loss: 2.3142 - val_accuracy: 0.0995 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 9s - loss: 2.3131 - accuracy: 0.1009 - val_loss: 2.3151 -
```

```
val_accuracy: 0.1050 - 9s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 2.3131 - accuracy: 0.1029 - val_loss: 2.3118 -
val_accuracy: 0.1050 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 2.3129 - accuracy: 0.1027 - val_loss: 2.3124 -
val_accuracy: 0.1045 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 2.3129 - accuracy: 0.1051 - val_loss: 2.3163 -
val_accuracy: 0.0978 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
Precision: 0.0096
Recall: 0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```

Training with adamw optimizer and the learning_rate is 0.01...
Epoch 1/10
1688/1688 - 9s - loss: 0.2243 - accuracy: 0.9376 - val_loss: 0.1232 - 
val_accuracy: 0.9640 - 9s/epoch - 5ms/step
Epoch 2/10
1688/1688 - 8s - loss: 0.1661 - accuracy: 0.9496 - val_loss: 0.1836 - 
val_accuracy: 0.9430 - 8s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 0.1692 - accuracy: 0.9486 - val_loss: 0.1388 - 
val_accuracy: 0.9598 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 8s - loss: 0.1689 - accuracy: 0.9484 - val_loss: 0.1320 - 
val_accuracy: 0.9605 - 8s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 0.1706 - accuracy: 0.9481 - val_loss: 0.1136 - 
val_accuracy: 0.9633 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.1734 - accuracy: 0.9479 - val_loss: 0.1693 - 
val_accuracy: 0.9418 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 0.1730 - accuracy: 0.9487 - val_loss: 0.1756 - 

```

```
val_accuracy: 0.9477 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 0.1719 - accuracy: 0.9480 - val_loss: 0.1796 -
val_accuracy: 0.9443 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 0.1732 - accuracy: 0.9473 - val_loss: 0.1144 -
val_accuracy: 0.9707 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 0.1718 - accuracy: 0.9475 - val_loss: 0.1677 -
val_accuracy: 0.9505 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 962   0   1   0   1   0   2   1   12   1]
 [  0 1122   3   0   0   0   7   0   3   0]
 [  6   7  972   1   0   0   5   4   37   0]
 [  1   0   23  896   0   21   2   4   55   8]
 [  2   7   4   0  866   0   15   0   14  74]
 [  3   1   0   4   0  847   5   1   30   1]
 [ 16   3   0   0   4   5  924   0   6   0]
 [  2  13   25   2   6   2   0  929   20  29]
 [  5   3   3   1   1   1   5   0  954   1]
 [  5   6   0   0   2   7   0   7   39  943]]]
Precision: 0.9454
Recall: 0.9415
```

Confusion Matrix for adamw optimizer and 0.01learning rate

	0	1	2	3	4	5	6	7	8	9	
0	962	0	1	0	1	0	2	1	12	1	
1	0	1122	3	0	0	0	7	0	3	0	
2	6	7	972	1	0	0	5	4	37	0	
3	1	0	23	896	0	21	2	4	55	8	
4	2	7	4	0	866	0	15	0	14	74	
5	3	1	0	4	0	847	5	1	30	1	
6	16	3	0	0	4	5	924	0	6	0	
7	2	13	25	2	6	2	0	929	20	29	
8	5	3	3	1	1	1	5	0	954	1	
9	5	6	0	0	2	7	0	7	39	943	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```
Training with adamw optimizer and the learning_rate is 0.001...
Epoch 1/10
1688/1688 - 9s - loss: 0.3764 - accuracy: 0.8995 - val_loss: 0.2826 - val_accuracy: 0.9328 - 9s/epoch - 5ms/step
Epoch 2/10
1688/1688 - 8s - loss: 0.3648 - accuracy: 0.8991 - val_loss: 0.3483 - val_accuracy: 0.8960 - 8s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 0.3590 - accuracy: 0.9008 - val_loss: 0.3110 - val_accuracy: 0.9163 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 8s - loss: 0.3544 - accuracy: 0.9015 - val_loss: 0.3221 - val_accuracy: 0.9063 - 8s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 0.3534 - accuracy: 0.9004 - val_loss: 0.3288 - val_accuracy: 0.8982 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.3532 - accuracy: 0.9004 - val_loss: 0.3157 - val_accuracy: 0.9212 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 0.3499 - accuracy: 0.9016 - val_loss: 0.3142 -
```

```
val_accuracy: 0.9195 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 0.3472 - accuracy: 0.9016 - val_loss: 0.2686 -
val_accuracy: 0.9318 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 0.3489 - accuracy: 0.9017 - val_loss: 0.2627 -
val_accuracy: 0.9358 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 0.3437 - accuracy: 0.9043 - val_loss: 0.2828 -
val_accuracy: 0.9328 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 938   0   0   3   2   5  15   2  14   1]
 [  0 1104   3   2   1   1   3   0  21   0]
 [  5   0  936  16  12   3  16   8  35   1]
 [  0   1  10  905   3  34   4  12  40   1]
 [  0   2   2   0  941   1  19   2   5  10]
 [  3   1   0  24  11  799  17   2  33   2]
 [  5   3   5   1  11  10  920   0   3   0]
 [  0  10  23   9  13   1   0  945   7  20]
 [  3   3   2  19   5  11  10   3  917   1]
 [  2   7   0  12  121  11   1  57  25  773]]
```

Precision: 0.9210
Recall: 0.9178

Confusion Matrix for adamw optimizer and 0.001 learning rate

	0	1	2	3	4	5	6	7	8	9	10
True Labels	938	0	0	3	2	5	15	2	14	1	0
0	0	1104	3	2	1	1	3	0	21	0	0
1	5	0	936	16	12	3	16	8	35	1	0
2	0	1	10	905	3	34	4	12	40	1	0
3	0	2	2	0	941	1	19	2	5	10	0
4	3	1	0	24	11	799	17	2	33	2	0
5	5	3	5	1	11	10	920	0	3	0	0
6	0	10	23	9	13	1	0	945	7	20	0
7	3	3	2	19	5	11	10	3	917	1	0
8	2	7	0	12	121	11	1	57	25	773	0
9	1	2	3	4	5	6	7	8	9	0	0
	0	1	2	3	4	5	6	7	8	9	10
Predicted Labels											

```
Training with adamw optimizer and the learning_rate is 0.0001...
```

```
Epoch 1/10
```

```
1688/1688 - 9s - loss: 1.7537 - accuracy: 0.6726 - val_loss: 1.8377 - val_accuracy: 0.6570 - 9s/epoch - 5ms/step
```

```
Epoch 2/10
```

```
1688/1688 - 8s - loss: 1.8555 - accuracy: 0.5796 - val_loss: 1.7170 - val_accuracy: 0.6332 - 8s/epoch - 5ms/step
```

```
Epoch 3/10
```

```
1688/1688 - 8s - loss: 1.7269 - accuracy: 0.6054 - val_loss: 1.7044 - val_accuracy: 0.6583 - 8s/epoch - 5ms/step
```

```
Epoch 4/10
```

```
1688/1688 - 8s - loss: 1.7212 - accuracy: 0.6055 - val_loss: 1.6842 - val_accuracy: 0.5875 - 8s/epoch - 5ms/step
```

```
Epoch 5/10
```

```
1688/1688 - 8s - loss: 1.7063 - accuracy: 0.6042 - val_loss: 1.6929 - val_accuracy: 0.5958 - 8s/epoch - 5ms/step
```

```
Epoch 6/10
```

```
1688/1688 - 8s - loss: 1.7041 - accuracy: 0.6009 - val_loss: 1.6875 - val_accuracy: 0.6473 - 8s/epoch - 5ms/step
```

```
Epoch 7/10
```

```
1688/1688 - 8s - loss: 1.7064 - accuracy: 0.6018 - val_loss: 1.6930 -
```

```
val_accuracy: 0.6053 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 1.7022 - accuracy: 0.6037 - val_loss: 1.6901 -
val_accuracy: 0.6293 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 1.7068 - accuracy: 0.6019 - val_loss: 1.6914 -
val_accuracy: 0.5982 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 1.7061 - accuracy: 0.5998 - val_loss: 1.6905 -
val_accuracy: 0.6153 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 942    0    5   25    1    0    6    1    0    0]
 [  0 1123    5    7    0    0    0    0    0    0]
 [ 68   71  645  150   12    0   74   10    0    2]
 [  5   41   46  883    2    0    0   32    0    1]
 [ 22   44    1    1  664    0   27   43    0  180]
 [ 116   20   34  588   39    0   14   79    0    2]
 [ 82   46   85   10    5    0  730    0    0    0]
 [  6   96    5   26   17    0    0  867    0   11]
 [ 45  154   40  599   23    0    5   93    0   15]
 [ 21   44    0   19  241    0    2  418    0  264]]
Precision: 0.5242
Recall: 0.6118

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adamw optimizer and 0.0001 learning rate

	0	1	2	3	4	5	6	7	8	9
True Labels	942	0	5	25	1	0	6	1	0	0
0	0	1123	5	7	0	0	0	0	0	0
1	68	71	645	150	12	0	74	10	0	2
2	5	41	46	883	2	0	0	32	0	1
3	22	44	1	1	664	0	27	43	0	180
4	116	20	34	588	39	0	14	79	0	2
5	82	46	85	10	5	0	730	0	0	0
6	6	96	5	26	17	0	0	867	0	11
7	45	154	40	599	23	0	5	93	0	15
8	21	44	0	19	241	0	2	418	0	264
9	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training with adadelta optimizer and the learning_rate is 0.1...
```

```
Epoch 1/10
```

```
1688/1688 - 9s - loss: 0.2660 - accuracy: 0.9274 - val_loss: 0.1140 - val_accuracy: 0.9697 - 9s/epoch - 6ms/step
```

```
Epoch 2/10
```

```
1688/1688 - 9s - loss: 0.1151 - accuracy: 0.9668 - val_loss: 0.0841 - val_accuracy: 0.9782 - 9s/epoch - 5ms/step
```

```
Epoch 3/10
```

```
1688/1688 - 8s - loss: 0.0838 - accuracy: 0.9757 - val_loss: 0.0724 - val_accuracy: 0.9805 - 8s/epoch - 5ms/step
```

```
Epoch 4/10
```

```
1688/1688 - 9s - loss: 0.0673 - accuracy: 0.9808 - val_loss: 0.0613 - val_accuracy: 0.9838 - 9s/epoch - 5ms/step
```

```
Epoch 5/10
```

```
1688/1688 - 9s - loss: 0.0566 - accuracy: 0.9841 - val_loss: 0.0669 - val_accuracy: 0.9817 - 9s/epoch - 5ms/step
```

```
Epoch 6/10
```

```
1688/1688 - 9s - loss: 0.0486 - accuracy: 0.9864 - val_loss: 0.0550 - val_accuracy: 0.9865 - 9s/epoch - 5ms/step
```

```
Epoch 7/10
```

```
1688/1688 - 9s - loss: 0.0425 - accuracy: 0.9883 - val_loss: 0.0513 -
```

```
val_accuracy: 0.9877 - 9s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 9s - loss: 0.0382 - accuracy: 0.9894 - val_loss: 0.0539 -
val_accuracy: 0.9852 - 9s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 9s - loss: 0.0346 - accuracy: 0.9904 - val_loss: 0.0525 -
val_accuracy: 0.9860 - 9s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 9s - loss: 0.0312 - accuracy: 0.9914 - val_loss: 0.0493 -
val_accuracy: 0.9872 - 9s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 976   0   1   1   0   0   0   1   1   0]
 [  0 1125   4   1   0   0   3   0   2   0]
 [  2   2 1021   1   2   0   0   3   0   1]
 [  1   0   0  997   0   3   0   3   3   3]
 [  0   0   2   0  969   0   3   0   0   8]
 [  3   0   1   4   0  877   5   0   1   1]
 [  7   3   1   0   1   2  942   0   2   0]
 [  1   3  14   3   0   0   0 1004   0   3]
 [  7   1   3   1   2   1   1   1  952   5]
 [  3   3   0   7   8   1   0   2   2  983]]]
Precision: 0.9846
Recall: 0.9846
```

Confusion Matrix for adadelta optimizer and 0.1learning rate										
	0	1	2	3	4	5	6	7	8	9
0	976	0	1	1	0	0	0	1	1	0
1	0	1125	4	1	0	0	3	0	2	0
2	2	2	1021	1	2	0	0	3	0	1
3	1	0	0	997	0	3	0	3	3	3
4	0	0	2	0	969	0	3	0	0	8
5	3	0	1	4	0	877	5	0	1	1
6	7	3	1	0	1	2	942	0	2	0
7	1	3	14	3	0	0	0	1004	0	3
8	7	1	3	1	2	1	1	1	952	5
9	3	3	0	7	8	1	0	2	2	983
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```

Training with adadelta optimizer and the learning_rate is 0.01...
Epoch 1/10
1688/1688 - 9s - loss: 0.6985 - accuracy: 0.8255 - val_loss: 0.2933 - val_accuracy: 0.9302 - 9s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 8s - loss: 0.3083 - accuracy: 0.9170 - val_loss: 0.2186 - val_accuracy: 0.9455 - 8s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 0.2523 - accuracy: 0.9300 - val_loss: 0.1865 - val_accuracy: 0.9523 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 0.2219 - accuracy: 0.9383 - val_loss: 0.1685 - val_accuracy: 0.9550 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 9s - loss: 0.2008 - accuracy: 0.9435 - val_loss: 0.1548 - val_accuracy: 0.9580 - 9s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.1843 - accuracy: 0.9483 - val_loss: 0.1434 - val_accuracy: 0.9633 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 9s - loss: 0.1712 - accuracy: 0.9515 - val_loss: 0.1342 -

```

```
val_accuracy: 0.9640 - 9s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 9s - loss: 0.1600 - accuracy: 0.9553 - val_loss: 0.1276 -
val_accuracy: 0.9652 - 9s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 9s - loss: 0.1501 - accuracy: 0.9576 - val_loss: 0.1211 -
val_accuracy: 0.9665 - 9s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 9s - loss: 0.1416 - accuracy: 0.9607 - val_loss: 0.1162 -
val_accuracy: 0.9682 - 9s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 968   0   1   1   0   1   4   2   2   1]
 [  0 1121   3   1   1   0   3   1   5   0]
 [  8   1  969   11   7   0   5   9  20   2]
 [  0   0   7  973   0   7   0   9   9   5]
 [  1   0   3   0  962   0   4   2   2   8]
 [  5   1   1   17   0  843   9   1   9   6]
 [  9   3   1   1   6   5  928   1   4   0]
 [  0   5   17   7   4   0   0  981   2  12]
 [  5   1   6   14   7   4   6   7  912  12]
 [  9   6   1   12   8   1   0   8   6  958]]]
Precision: 0.9616
Recall: 0.9615
```

Confusion Matrix for adadelta optimizer and 0.01learning rate

	0	1	2	3	4	5	6	7	8	9	10
0	968	0	1	1	0	1	4	2	2	1	
1	0	1121	3	1	1	0	3	1	5	0	
2	8	1	969	11	7	0	5	9	20	2	
3	0	0	7	973	0	7	0	9	9	5	
4	1	0	3	0	962	0	4	2	2	8	
5	5	1	1	17	0	843	9	1	9	6	
6	9	3	1	1	6	5	928	1	4	0	
7	0	5	17	7	4	0	0	981	2	12	
8	5	1	6	14	7	4	6	7	912	12	
9	9	6	1	12	8	1	0	8	6	958	
10	1	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
True Labels											Predicted Labels

```

Training with adadelta optimizer and the learning_rate is 0.001...
Epoch 1/10
1688/1688 - 10s - loss: 1.9317 - accuracy: 0.3979 - val_loss: 1.5360 - val_accuracy: 0.6773 - 10s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 9s - loss: 1.3279 - accuracy: 0.7250 - val_loss: 1.0616 - val_accuracy: 0.8127 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 0.9713 - accuracy: 0.8074 - val_loss: 0.7799 - val_accuracy: 0.8652 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 0.7585 - accuracy: 0.8432 - val_loss: 0.6140 - val_accuracy: 0.8885 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 9s - loss: 0.6299 - accuracy: 0.8629 - val_loss: 0.5128 - val_accuracy: 0.8993 - 9s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.5470 - accuracy: 0.8765 - val_loss: 0.4462 - val_accuracy: 0.9090 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 9s - loss: 0.4902 - accuracy: 0.8860 - val_loss: 0.3996 -

```

```
val_accuracy: 0.9128 - 9s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 9s - loss: 0.4483 - accuracy: 0.8935 - val_loss: 0.3651 -
val_accuracy: 0.9168 - 9s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 0.4167 - accuracy: 0.8993 - val_loss: 0.3386 -
val_accuracy: 0.9218 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 0.3918 - accuracy: 0.9037 - val_loss: 0.3178 -
val_accuracy: 0.9242 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 957   0   2   2   1   1  10   1   6   0]
 [  0 1111   2   4   0   1   5   1  11   0]
 [ 12    2  895  19  17   0  16  23  46   2]
 [  3    1  18  911   0  25   4  18  24   6]
 [  0    5   6   0  909   0  16   2   7  37]
 [  9    3   4  43    7  764  14   5  39   4]
 [ 17    5   5   0   9  16  897   1   8   0]
 [  4   14  20   3  12   0   0  923   6  46]
 [ 11    6   9  22   9  17   7  12  863  18]
 [  9    7   4  15  24   6   0  28   5  911]]]
Precision: 0.9143
Recall: 0.9141
```

Confusion Matrix for adadelta optimizer and 0.001learning rate

	0	1	2	3	4	5	6	7	8	9	
0	957	0	2	2	1	1	10	1	6	0	
1	0	1111	2	4	0	1	5	1	11	0	
2	12	2	895	19	17	0	16	23	46	2	
3	3	1	18	911	0	25	4	18	24	6	
4	0	5	6	0	909	0	16	2	7	37	
5	9	3	4	43	7	764	14	5	39	4	
6	17	5	5	0	9	16	897	1	8	0	
7	4	14	20	3	12	0	0	923	6	46	
8	11	6	9	22	9	17	7	12	863	18	
9	9	7	4	15	24	6	0	28	5	911	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

Training with adadelta optimizer and the learning_rate is 0.0001...

Epoch 1/10

1688/1688 - 10s - loss: 2.5126 - accuracy: 0.1246 - val_loss: 2.3987 - val_accuracy: 0.1423 - 10s/epoch - 6ms/step

Epoch 2/10

1688/1688 - 9s - loss: 2.3159 - accuracy: 0.1613 - val_loss: 2.2317 - val_accuracy: 0.1967 - 9s/epoch - 5ms/step

Epoch 3/10

1688/1688 - 9s - loss: 2.1757 - accuracy: 0.2311 - val_loss: 2.1007 - val_accuracy: 0.2993 - 9s/epoch - 5ms/step

Epoch 4/10

1688/1688 - 9s - loss: 2.0594 - accuracy: 0.3416 - val_loss: 1.9862 - val_accuracy: 0.4018 - 9s/epoch - 5ms/step

Epoch 5/10

1688/1688 - 9s - loss: 1.9553 - accuracy: 0.4278 - val_loss: 1.8821 - val_accuracy: 0.4780 - 9s/epoch - 5ms/step

Epoch 6/10

1688/1688 - 9s - loss: 1.8594 - accuracy: 0.4922 - val_loss: 1.7850 - val_accuracy: 0.5430 - 9s/epoch - 5ms/step

Epoch 7/10

1688/1688 - 9s - loss: 1.7690 - accuracy: 0.5495 - val_loss: 1.6936 -

```
val_accuracy: 0.6017 - 9s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 9s - loss: 1.6840 - accuracy: 0.5983 - val_loss: 1.6071 -
val_accuracy: 0.6482 - 9s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 9s - loss: 1.6034 - accuracy: 0.6403 - val_loss: 1.5255 -
val_accuracy: 0.6905 - 9s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 1.5277 - accuracy: 0.6756 - val_loss: 1.4491 -
val_accuracy: 0.7233 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 923   0   3  15   2   2  15   1   17   2]
 [  0 1002  28   3   0   2   5   1   91   3]
 [ 104   35  684  57  11  17  35   6   52  31]
 [  31   15   41  760   3  59   7   4   74  16]
 [   8   12    9   20  473   1  102  13  108 236]
 [  41   19   14  187   6  448   27   3  107  40]
 [  69   37   16   11   4  11  778   0   31   1]
 [  16   47   20   15  34   6  13  689   54 134]
 [  33   39   38   61  28  42  10   11  665  47]
 [  14   16    2   16 124   1   8   38  130 660]]
```

Precision: 0.7240
Recall: 0.7082

Confusion Matrix for adadelta optimizer and 0.0001 learning rate

	0	1	2	3	4	5	6	7	8	9
0	923	0	3	15	2	2	15	1	17	2
1	0	1002	28	3	0	2	5	1	91	3
2	104	35	684	57	11	17	35	6	52	31
3	31	15	41	760	3	59	7	4	74	16
4	8	12	9	20	473	1	102	13	108	236
5	41	19	14	187	6	448	27	3	107	40
6	69	37	16	11	4	11	778	0	31	1
7	16	47	20	15	34	6	13	689	54	134
8	33	39	38	61	28	42	10	11	665	47
9	14	16	2	16	124	1	8	38	130	660
	0	1	2	3	4	5	6	7	8	9
True Labels										Predicted Labels

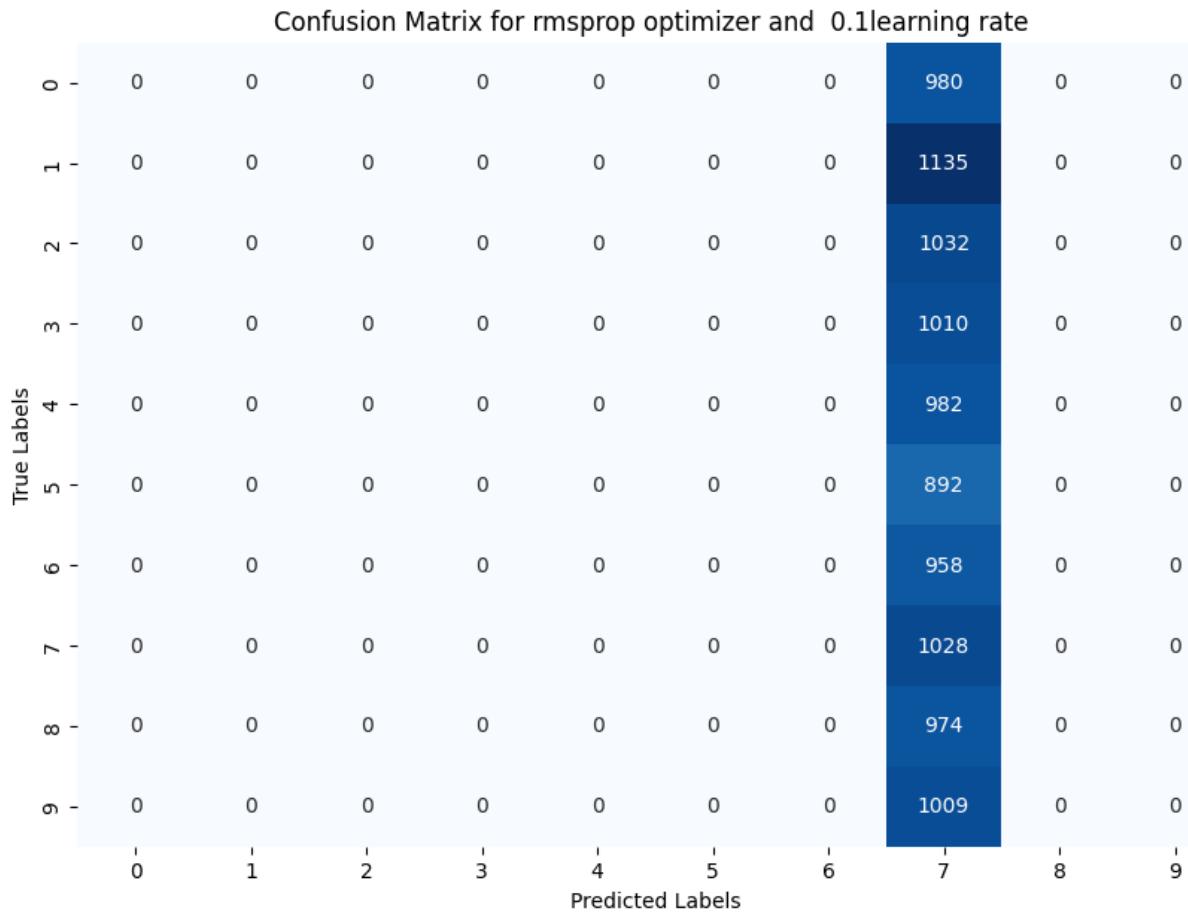
```

Training with rmsprop optimizer and the learning_rate is 0.1...
Epoch 1/10
1688/1688 - 9s - loss: 8.5883 - accuracy: 0.1011 - val_loss: 2.4173 - val_accuracy: 0.1000 - 9s/epoch - 5ms/step
Epoch 2/10
1688/1688 - 8s - loss: 2.4047 - accuracy: 0.0998 - val_loss: 2.3479 - val_accuracy: 0.0960 - 8s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 8s - loss: 2.4031 - accuracy: 0.1016 - val_loss: 2.3723 - val_accuracy: 0.1113 - 8s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 8s - loss: 2.4039 - accuracy: 0.0991 - val_loss: 2.4925 - val_accuracy: 0.1000 - 8s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 2.4091 - accuracy: 0.1016 - val_loss: 2.3680 - val_accuracy: 0.0992 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 2.4067 - accuracy: 0.1023 - val_loss: 2.3806 - val_accuracy: 0.1045 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 2.4032 - accuracy: 0.1005 - val_loss: 2.4068 -

```

```
val_accuracy: 0.0960 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 2.4046 - accuracy: 0.1008 - val_loss: 2.3784 -
val_accuracy: 0.1000 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 2.4031 - accuracy: 0.0997 - val_loss: 2.4673 -
val_accuracy: 0.0960 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 2.4049 - accuracy: 0.1012 - val_loss: 2.4212 -
val_accuracy: 0.1113 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 0  0  0  0  0  0  0  980  0  0]
 [ 0  0  0  0  0  0  0  1135  0  0]
 [ 0  0  0  0  0  0  0  1032  0  0]
 [ 0  0  0  0  0  0  0  1010  0  0]
 [ 0  0  0  0  0  0  0  982  0  0]
 [ 0  0  0  0  0  0  0  892  0  0]
 [ 0  0  0  0  0  0  0  958  0  0]
 [ 0  0  0  0  0  0  0  1028  0  0]
 [ 0  0  0  0  0  0  0  974  0  0]
 [ 0  0  0  0  0  0  0  1009  0  0]]
Precision: 0.0106
Recall: 0.1028

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



```

Training with rmsprop optimizer and the learning_rate is 0.01...
Epoch 1/10
1688/1688 - 9s - loss: 2.4127 - accuracy: 0.1024 - val_loss: 2.3108 - val_accuracy: 0.1113 - 9s/epoch - 5ms/step
Epoch 2/10
1688/1688 - 9s - loss: 2.3138 - accuracy: 0.1026 - val_loss: 2.3074 - val_accuracy: 0.1050 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 9s - loss: 2.3137 - accuracy: 0.1046 - val_loss: 2.3110 - val_accuracy: 0.1113 - 9s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 2.3131 - accuracy: 0.1015 - val_loss: 2.3060 - val_accuracy: 0.0978 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 8s - loss: 2.3137 - accuracy: 0.1032 - val_loss: 2.3189 - val_accuracy: 0.1000 - 8s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 2.3131 - accuracy: 0.1058 - val_loss: 2.3158 - val_accuracy: 0.1050 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 9s - loss: 2.3131 - accuracy: 0.1038 - val_loss: 2.3190 -

```

```
val_accuracy: 0.1050 - 9s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 2.3138 - accuracy: 0.1035 - val_loss: 2.3156 -
val_accuracy: 0.1050 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 2.3133 - accuracy: 0.1028 - val_loss: 2.3306 -
val_accuracy: 0.0915 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 2.3139 - accuracy: 0.1019 - val_loss: 2.3193 -
val_accuracy: 0.0995 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 0  0  0  0  0  0  0  0  980  0]
 [ 0  0  0  0  0  0  0  0  1135  0]
 [ 0  0  0  0  0  0  0  0  1032  0]
 [ 0  0  0  0  0  0  0  0  1010  0]
 [ 0  0  0  0  0  0  0  0  982  0]
 [ 0  0  0  0  0  0  0  0  892  0]
 [ 0  0  0  0  0  0  0  0  958  0]
 [ 0  0  0  0  0  0  0  0  1028  0]
 [ 0  0  0  0  0  0  0  0  974  0]
 [ 0  0  0  0  0  0  0  0  1009  0]]
Precision: 0.0095
Recall: 0.0974

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for rmsprop optimizer and 0.01learning rate											
True Labels	0 -	1	2	3	4	5	6	7	8	9	
0 -	980	0	0	0	0	0	0	0	0	0	0
1 -	1135	0	0	0	0	0	0	0	0	0	0
2 -	1032	0	0	0	0	0	0	0	0	0	0
3 -	1010	0	0	0	0	0	0	0	0	0	0
4 -	982	0	0	0	0	0	0	0	0	0	0
5 -	892	0	0	0	0	0	0	0	0	0	0
6 -	958	0	0	0	0	0	0	0	0	0	0
7 -	1028	0	0	0	0	0	0	0	0	0	0
8 -	974	0	0	0	0	0	0	0	0	0	0
9 -	1009	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

```

Training with rmsprop optimizer and the learning_rate is 0.001...
Epoch 1/10
1688/1688 - 9s - loss: 0.1567 - accuracy: 0.9578 - val_loss: 0.0801 - val_accuracy: 0.9797 - 9s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 9s - loss: 0.0706 - accuracy: 0.9823 - val_loss: 0.0632 - val_accuracy: 0.9867 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 9s - loss: 0.0533 - accuracy: 0.9874 - val_loss: 0.1059 - val_accuracy: 0.9790 - 9s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 0.0472 - accuracy: 0.9889 - val_loss: 0.1264 - val_accuracy: 0.9827 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 9s - loss: 0.0478 - accuracy: 0.9902 - val_loss: 0.0880 - val_accuracy: 0.9858 - 9s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 8s - loss: 0.0400 - accuracy: 0.9913 - val_loss: 0.1621 - val_accuracy: 0.9817 - 8s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 0.0396 - accuracy: 0.9929 - val_loss: 0.1269 -

```

```
val_accuracy: 0.9850 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 8s - loss: 0.0299 - accuracy: 0.9941 - val_loss: 0.1798 -
val_accuracy: 0.9847 - 8s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 0.0365 - accuracy: 0.9943 - val_loss: 0.1747 -
val_accuracy: 0.9862 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 9s - loss: 0.0265 - accuracy: 0.9955 - val_loss: 0.1687 -
val_accuracy: 0.9872 - 9s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 975   0   1   0   1   0   0   1   1   1]
 [  0 1127   0   2   0   0   3   2   1   0]
 [  6   4 1009   3   4   0   0   5   1   0]
 [  0   0   1 1001   0   3   0   1   3   1]
 [  0   2   0   0  970   0   2   1   0   7]
 [  2   0   0   8   1  873   3   0   3   2]
 [ 11   1   0   0   5   4  937   0   0   0]
 [  1   4   5   1   3   0   0 1011   0   3]
 [ 11   2   3   3   0   1   0   2  948   4]
 [  3   2   0   0   9   2   0   6   5  982]]
```

Precision: 0.9834
Recall: 0.9833

Confusion Matrix for rmsprop optimizer and 0.001 learning rate

	0	1	2	3	4	5	6	7	8	9
True Labels	975	0	1	0	1	0	0	1	1	1
0	975	0	1	0	1	0	0	1	1	1
1	0	1127	0	2	0	0	3	2	1	0
2	6	4	1009	3	4	0	0	5	1	0
3	0	0	1	1001	0	3	0	1	3	1
4	0	2	0	0	970	0	2	1	0	7
5	2	0	0	8	1	873	3	0	3	2
6	11	1	0	0	5	4	937	0	0	0
7	1	4	5	1	3	0	0	1011	0	3
8	11	2	3	3	0	1	0	2	948	4
9	3	2	0	0	9	2	0	6	5	982
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```

Training with rmsprop optimizer and the learning_rate is 0.0001...
Epoch 1/10
1688/1688 - 10s - loss: 0.1533 - accuracy: 0.9530 - val_loss: 0.0577 -
val_accuracy: 0.9855 - 10s/epoch - 6ms/step
Epoch 2/10
1688/1688 - 9s - loss: 0.0484 - accuracy: 0.9856 - val_loss: 0.0531 -
val_accuracy: 0.9868 - 9s/epoch - 5ms/step
Epoch 3/10
1688/1688 - 9s - loss: 0.0291 - accuracy: 0.9911 - val_loss: 0.0488 -
val_accuracy: 0.9890 - 9s/epoch - 5ms/step
Epoch 4/10
1688/1688 - 9s - loss: 0.0176 - accuracy: 0.9949 - val_loss: 0.0555 -
val_accuracy: 0.9878 - 9s/epoch - 5ms/step
Epoch 5/10
1688/1688 - 9s - loss: 0.0109 - accuracy: 0.9970 - val_loss: 0.0538 -
val_accuracy: 0.9890 - 9s/epoch - 5ms/step
Epoch 6/10
1688/1688 - 9s - loss: 0.0065 - accuracy: 0.9981 - val_loss: 0.0612 -
val_accuracy: 0.9873 - 9s/epoch - 5ms/step
Epoch 7/10
1688/1688 - 8s - loss: 0.0040 - accuracy: 0.9989 - val_loss: 0.0603 -

```

```
val_accuracy: 0.9897 - 8s/epoch - 5ms/step
Epoch 8/10
1688/1688 - 9s - loss: 0.0029 - accuracy: 0.9992 - val_loss: 0.0637 -
val_accuracy: 0.9887 - 9s/epoch - 5ms/step
Epoch 9/10
1688/1688 - 8s - loss: 0.0014 - accuracy: 0.9996 - val_loss: 0.0757 -
val_accuracy: 0.9885 - 8s/epoch - 5ms/step
Epoch 10/10
1688/1688 - 8s - loss: 9.2322e-04 - accuracy: 0.9998 - val_loss:
0.0710 - val_accuracy: 0.9903 - 8s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix:
[[ 974    1    2    0    0    1    1    1    0    0]
 [  0 1126    3    1    0    0    2    1    2    0]
 [  1    1 1016    1    2    0    1    5    5    0]
 [  0    0    2 1002    1    3    0    0    2    0]
 [  0    1    1    0   974    0    1    0    0    5]
 [  2    0    1    4    0   881    4    0    0    0]
 [  4    2    0    1    2    1   947    0    1    0]
 [  0    1    4    0    0    0    0 1022    1    0]
 [  4    1    3    0    1    0    0    1   961    3]
 [  2    3    0    4    9    3    0    8    2   978]]]
Precision: 0.9881
Recall: 0.9881
```

Confusion Matrix for rmsprop optimizer and 0.0001 learning rate										
	0	1	2	0	0	1	1	1	0	0
True Labels	974	1	2	0	0	1	1	1	0	0
0	974	1	2	0	0	1	1	1	0	0
1	0	1126	3	1	0	0	2	1	2	0
2	1	1	1016	1	2	0	1	5	5	0
3	0	0	2	1002	1	3	0	0	2	0
4	0	1	1	0	974	0	1	0	0	5
5	2	0	1	4	0	881	4	0	0	0
6	4	2	0	1	2	1	947	0	1	0
7	0	1	4	0	0	0	0	1022	1	0
8	4	1	3	0	1	0	0	1	961	3
9	2	3	0	4	9	3	0	8	2	978
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

Impact of Learning Rate:

A learning rate of 0.001 tends to yield optimal results across different optimizers, striking a balance between convergence speed and stability.

Higher learning rates generate lower performance values when we use adadelta and rmsprop optimizers in the model. Conversely, if we decrease the learning rates, then adam optimizer shows high classification performance of the model.

Adamw shows poor performance when the learning rate is too low and too high. When we gradually decrease the learning rate, the performance sharply increases and decreases further.

Optimizer Performance:

Adam and Nadam optimizers consistently provide high precision and recall across most configurations, suggesting their effectiveness in adapting the learning rate during training.

AdamW also shows promising results, particularly at lower learning rates, indicating its potential in regularization and preventing overfitting.

Adadelta and RMSprop exhibit variable performance, with instances of very low precision and recall at certain learning rates, pointing to their sensitivity to learning rate settings and possible difficulties in navigating the loss landscape.

Learning Rate Optimization: Fine-tuning the learning rate is essential, with lower rates (e.g., 0.001, 0.0001) often yielding better performance, particularly in combination with adaptive optimizers like Adam and Nadam.

Optimizer Efficacy: The choice of optimizer significantly affects model outcomes, where Adam, Nadam, and AdamW emerge as versatile and effective across a broad spectrum of configurations.

Experimenting with Various Batch Sizes and Epochs for Training:

In this section, we are going to experiment with various batch sizes and epochs for training our neural network. We are going to combine this experiment with various variations such as different activation functions, layer configurations, neuron configurations, various optimizer and different learning rates.

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense, Activation
from tensorflow.keras.optimizers import AdamW, Adam, RMSprop, Adagrad,
Nadam, Adadelta

! pip install tensorflow_addons

Collecting tensorflow_addons
  Downloading tensorflow_addons-0.23.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (611 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 611.8/611.8 kB 4.3 MB/s eta
0:00:00
ent already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from tensorflow_addons) (24.0)
Collecting typeguard<3.0.0,>=2.7 (from tensorflow_addons)
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
Installing collected packages: typeguard, tensorflow_addons
Successfully installed tensorflow_addons-0.23.0 typeguard-2.13.3

import tensorflow_addons as tfa

/usr/local/lib/python3.10/dist-packages/tensorflow_addons/utils/
tfa_eol_msg.py:23: UserWarning:

TensorFlow Addons (TFA) has ended development and introduction of new
features.
TFA has entered a minimal maintenance and release mode until a planned
end of life in May 2024.
Please modify downstream libraries to take dependencies from other
repositories in our TensorFlow community (e.g. Keras, Keras-CV, and
Keras-NLP).

For more information see:
```

<https://github.com/tensorflow/addons/issues/2807>

```
warnings.warn(  
/usr/local/lib/python3.10/dist-packages/tensorflow_addons/utils/ensure  
_tf_install.py:53: UserWarning: Tensorflow Addons supports using  
Python ops for all Tensorflow versions above or equal to 2.13.0 and  
strictly below 2.16.0 (nightly versions are not supported).  
The versions of TensorFlow you are currently using is 2.12.0 and is  
not supported.  
Some things might work, some things might not.  
If you were to encounter a bug, do not file an issue.  
If you want to make sure you're using a tested and supported  
configuration, either change the TensorFlow version or the TensorFlow  
Addons's version.  
You can find the compatibility matrix in TensorFlow Addon's readme:  
https://github.com/tensorflow/addons  
warnings.warn(
```

Increasing the Size and the Depth of the Inner layers with various Epochs and Batch Sizes:

```
from tensorflow.keras.models import Sequential  
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,  
Dense  
from tensorflow.keras.optimizers import SGD  
  
act_functions = ["relu", "sigmoid", "softmax"]  
layer_configs = [(1,1), (2,2), (3,3)]  
neuron_configs = [64, 128, 256]  
optimizers = ["adam", "nadam", "adamw"]  
batch_sizes = [64, 128, 256, 16]  
epochs_list = [5, 15, 20]  
learning_rate_num = [0.1, 0.01, 0.001]  
  
def modified_model_1():  
    model = Sequential()  
    ### Increase the filters numbers in convolutional layers  
    model.add(Conv2D(64, (3,3), activation = "relu", kernel_initializer =  
"he_uniform", input_shape = (28,28,1)))  
    model.add(MaxPooling2D((2,2)))  
    ## adding another convolutional layer  
    model.add(Conv2D(128, (3,3), activation = "relu", kernel_initializer =  
"he_uniform")) ## Addititonal Layers  
    model.add(MaxPooling2D((2,2)))  
  
    model.add(Flatten())  
  
    ### Increasing the number of neurons in the dense layers  
    model.add(Dense(200, activation = "relu", kernel_initializer =  
"he_uniform"))  
    model.add(Dense(100, activation = "relu", kernel_initializer =
```

```

"he_uniform"))

### Output layer
model.add(Dense(10,activation = "softmax"))

## Compile the model
opt = SGD(learning_rate = 0.01, momentum = 0.9)
model.compile(optimizer = opt, loss = "categorical_crossentropy",
metrics = ["accuracy"])
return model

for batch in batch_sizes:
    for epoch in epochs_list:
        model_1 = modified_model_1()
        history = model_1.fit(train_norm, y_train, epochs = epoch,
batch_size = batch, validation_split = 0.1, verbose = 2)
        print(f"Running model for {batch} batch size and {epoch} epochs....")
        # predictions and metrics
        y_pred_prob = model_1.predict(test_norm)
        y_pred = np.argmax(y_pred_prob, axis=1)

        y_true = np.argmax(y_test, axis=1) if y_test.ndim > 1 else y_test
        cm = confusion_matrix(y_true, y_pred)
        precision = precision_score(y_true, y_pred, average="weighted")
        recall = recall_score(y_true, y_pred, average="weighted")

                           # Printing metrics and plotting the
confusion matrix
        print("Confusion Matrix...")
        print(cm)
        print(f"Precision score: {precision:.4f}")
        print(f"Recall: {recall:.4f}")
        plt.figure(figsize=(10,6))
        sns.heatmap(cm, annot=True, fmt="g", cmap="Blues", cbar=False)
        plt.xlabel("Predicted Labels")
        plt.ylabel("True Labels")
        plt.title(f"Confusion Matrix for {batch} batch size, {epoch} epochs")
        plt.show()

Epoch 1/5
844/844 - 14s - loss: 0.1499 - accuracy: 0.9542 - val_loss: 0.0527 -
val_accuracy: 0.9860 - 14s/epoch - 17ms/step
Epoch 2/5
844/844 - 14s - loss: 0.0467 - accuracy: 0.9856 - val_loss: 0.0479 -
val_accuracy: 0.9880 - 14s/epoch - 16ms/step
Epoch 3/5

```

```
844/844 - 14s - loss: 0.0306 - accuracy: 0.9910 - val_loss: 0.0361 -  
val_accuracy: 0.9895 - 14s/epoch - 16ms/step
```

```
Epoch 4/5
```

```
844/844 - 13s - loss: 0.0221 - accuracy: 0.9926 - val_loss: 0.0379 -  
val_accuracy: 0.9898 - 13s/epoch - 16ms/step
```

```
Epoch 5/5
```

```
844/844 - 13s - loss: 0.0155 - accuracy: 0.9949 - val_loss: 0.0326 -  
val_accuracy: 0.9903 - 13s/epoch - 16ms/step
```

```
Running model for 64 batch size and 5 epochs....
```

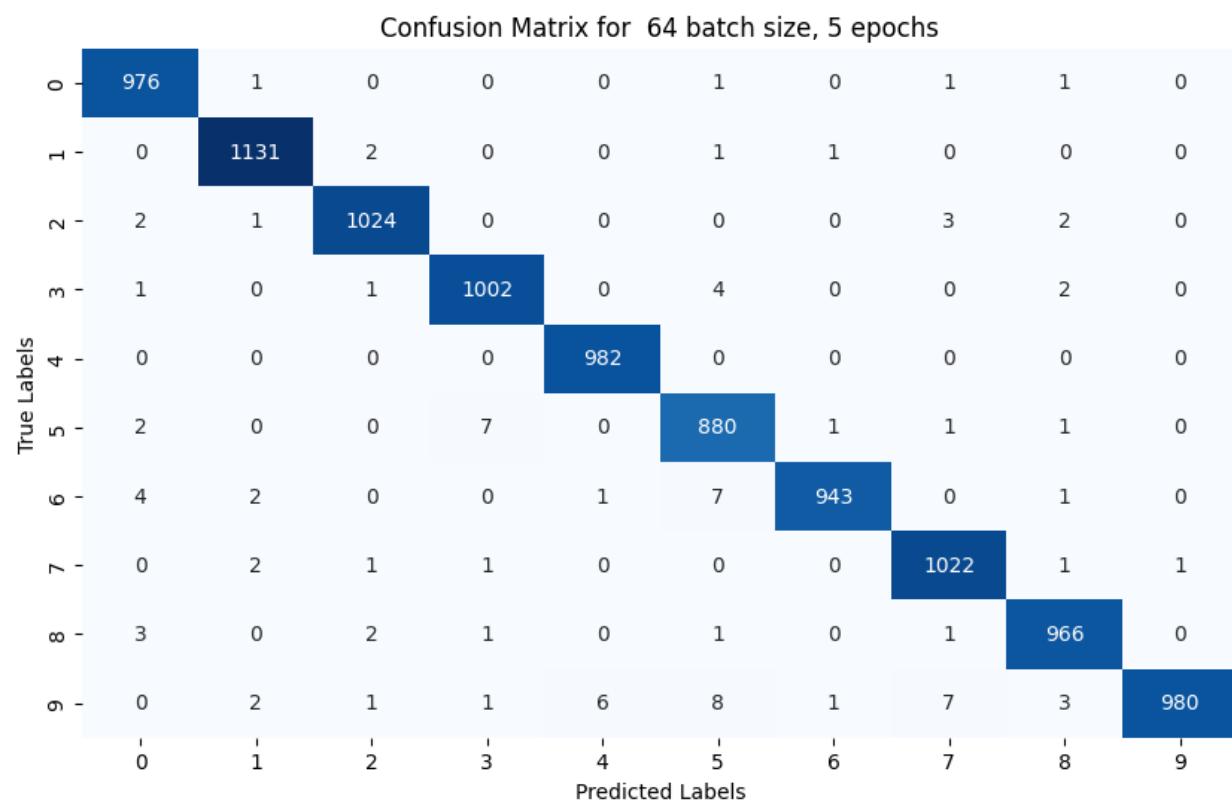
```
313/313 [=====] - 1s 4ms/step
```

```
Confusion Matrix...
```

```
[[ 976   1   0   0   0   1   0   1   1   0 ]  
 [  0 1131   2   0   0   1   1   0   0   0 ]  
 [  2   1 1024   0   0   0   0   3   2   0 ]  
 [  1   0   1 1002   0   4   0   0   2   0 ]  
 [  0   0   0   0  982   0   0   0   0   0 ]  
 [  2   0   0   7   0  880   1   1   1   0 ]  
 [  4   2   0   0   1   7  943   0   1   0 ]  
 [  0   2   1   1   0   0   0 1022   1   1 ]  
 [  3   0   2   1   0   1   0   1  966   0 ]  
 [  0   2   1   1   6   8   1   7   3  980 ]]
```

```
Precision score: 0.9906
```

```
Recall: 0.9906
```



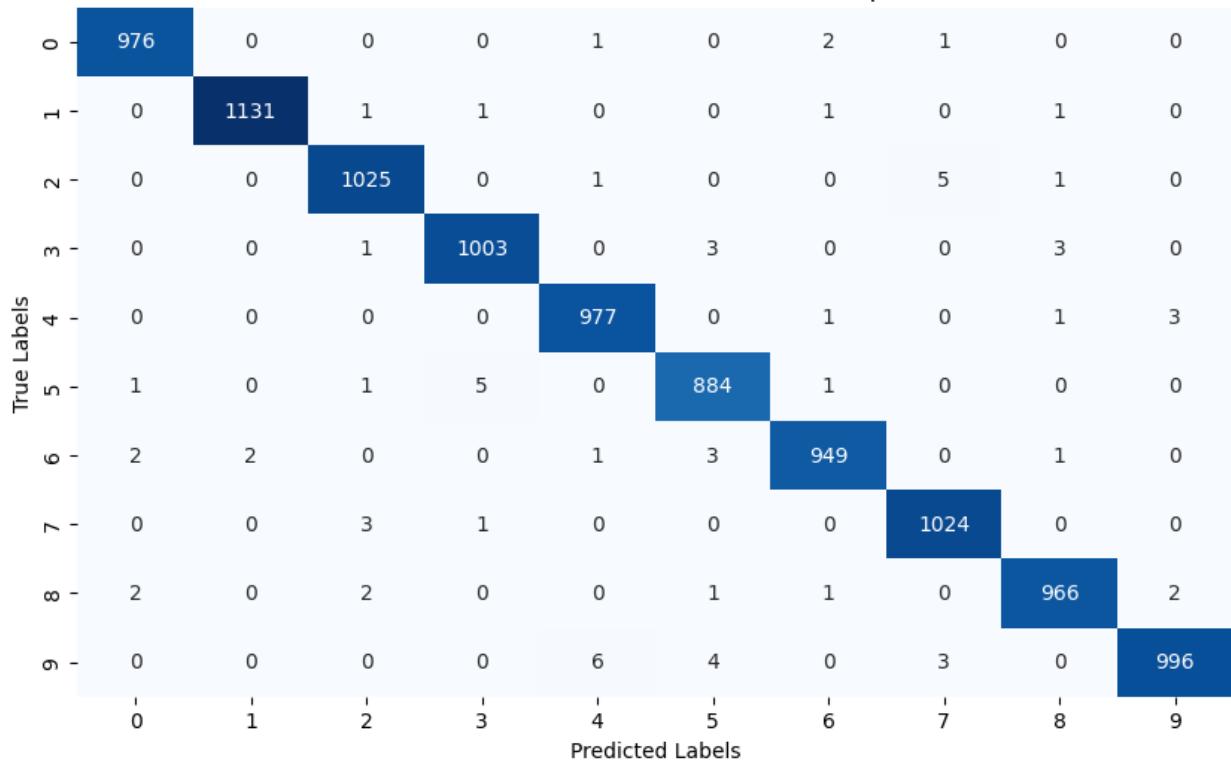
```
Epoch 1/15
844/844 - 15s - loss: 0.1513 - accuracy: 0.9521 - val_loss: 0.0550 -
val_accuracy: 0.9847 - 15s/epoch - 17ms/step
Epoch 2/15
844/844 - 14s - loss: 0.0482 - accuracy: 0.9845 - val_loss: 0.0421 -
val_accuracy: 0.9888 - 14s/epoch - 17ms/step
Epoch 3/15
844/844 - 13s - loss: 0.0325 - accuracy: 0.9893 - val_loss: 0.0399 -
val_accuracy: 0.9895 - 13s/epoch - 16ms/step
Epoch 4/15
844/844 - 14s - loss: 0.0228 - accuracy: 0.9927 - val_loss: 0.0418 -
val_accuracy: 0.9887 - 14s/epoch - 16ms/step
Epoch 5/15
844/844 - 13s - loss: 0.0165 - accuracy: 0.9945 - val_loss: 0.0400 -
val_accuracy: 0.9898 - 13s/epoch - 16ms/step
Epoch 6/15
844/844 - 14s - loss: 0.0116 - accuracy: 0.9960 - val_loss: 0.0356 -
val_accuracy: 0.9915 - 14s/epoch - 16ms/step
Epoch 7/15
844/844 - 14s - loss: 0.0091 - accuracy: 0.9971 - val_loss: 0.0380 -
val_accuracy: 0.9917 - 14s/epoch - 16ms/step
Epoch 8/15
844/844 - 14s - loss: 0.0063 - accuracy: 0.9982 - val_loss: 0.0375 -
val_accuracy: 0.9912 - 14s/epoch - 16ms/step
Epoch 9/15
844/844 - 14s - loss: 0.0042 - accuracy: 0.9988 - val_loss: 0.0385 -
val_accuracy: 0.9920 - 14s/epoch - 16ms/step
Epoch 10/15
844/844 - 14s - loss: 0.0025 - accuracy: 0.9995 - val_loss: 0.0393 -
val_accuracy: 0.9918 - 14s/epoch - 16ms/step
Epoch 11/15
844/844 - 13s - loss: 0.0017 - accuracy: 0.9997 - val_loss: 0.0408 -
val_accuracy: 0.9923 - 13s/epoch - 16ms/step
Epoch 12/15
844/844 - 14s - loss: 0.0012 - accuracy: 0.9998 - val_loss: 0.0430 -
val_accuracy: 0.9922 - 14s/epoch - 16ms/step
Epoch 13/15
844/844 - 13s - loss: 0.0014 - accuracy: 0.9996 - val_loss: 0.0475 -
val_accuracy: 0.9913 - 13s/epoch - 16ms/step
Epoch 14/15
844/844 - 13s - loss: 6.2666e-04 - accuracy: 0.9999 - val_loss: 0.0430
- val_accuracy: 0.9928 - 13s/epoch - 16ms/step
Epoch 15/15
844/844 - 14s - loss: 3.0740e-04 - accuracy: 1.0000 - val_loss: 0.0447
- val_accuracy: 0.9930 - 14s/epoch - 16ms/step
Running model for 64 batch size and 15 epochs....
313/313 [=====] - 1s 5ms/step
Confusion Matrix...
[[ 976   0   0   0   1   0   2   1   0   0]
 [  0 1131   1   1   0   0   1   0   1   0]]
```

```
[ 0  0 1025  0  1  0  0  5  1  0]
[ 0  0  1 1003  0  3  0  0  3  0]
[ 0  0  0  0 977  0  1  0  1  3]
[ 1  0  1  5  0 884  1  0  0  0]
[ 2  2  0  0  1  3 949  0  1  0]
[ 0  0  3  1  0  0  0 1024  0  0]
[ 2  0  2  0  0  1  1  0 966  2]
[ 0  0  0  0  6  4  0  3  0 996]]
```

Precision score: 0.9931

Recall: 0.9931

Confusion Matrix for 64 batch size, 15 epochs



Epoch 1/20

844/844 - 14s - loss: 0.1638 - accuracy: 0.9482 - val_loss: 0.0618 - val_accuracy: 0.9808 - 14s/epoch - 17ms/step

Epoch 2/20

844/844 - 13s - loss: 0.0512 - accuracy: 0.9844 - val_loss: 0.0513 - val_accuracy: 0.9853 - 13s/epoch - 16ms/step

Epoch 3/20

844/844 - 13s - loss: 0.0338 - accuracy: 0.9895 - val_loss: 0.0386 - val_accuracy: 0.9883 - 13s/epoch - 16ms/step

Epoch 4/20

844/844 - 13s - loss: 0.0249 - accuracy: 0.9920 - val_loss: 0.0347 - val_accuracy: 0.9897 - 13s/epoch - 15ms/step

Epoch 5/20

```
844/844 - 13s - loss: 0.0189 - accuracy: 0.9943 - val_loss: 0.0390 -  
val_accuracy: 0.9897 - 13s/epoch - 16ms/step  
Epoch 6/20  
844/844 - 13s - loss: 0.0133 - accuracy: 0.9959 - val_loss: 0.0376 -  
val_accuracy: 0.9907 - 13s/epoch - 16ms/step  
Epoch 7/20  
844/844 - 13s - loss: 0.0103 - accuracy: 0.9965 - val_loss: 0.0350 -  
val_accuracy: 0.9912 - 13s/epoch - 16ms/step  
Epoch 8/20  
844/844 - 13s - loss: 0.0079 - accuracy: 0.9972 - val_loss: 0.0389 -  
val_accuracy: 0.9915 - 13s/epoch - 16ms/step  
Epoch 9/20  
844/844 - 14s - loss: 0.0058 - accuracy: 0.9981 - val_loss: 0.0394 -  
val_accuracy: 0.9907 - 14s/epoch - 16ms/step  
Epoch 10/20  
844/844 - 13s - loss: 0.0049 - accuracy: 0.9985 - val_loss: 0.0473 -  
val_accuracy: 0.9900 - 13s/epoch - 16ms/step  
Epoch 11/20  
844/844 - 14s - loss: 0.0027 - accuracy: 0.9993 - val_loss: 0.0370 -  
val_accuracy: 0.9922 - 14s/epoch - 16ms/step  
Epoch 12/20  
844/844 - 14s - loss: 0.0022 - accuracy: 0.9995 - val_loss: 0.0399 -  
val_accuracy: 0.9923 - 14s/epoch - 16ms/step  
Epoch 13/20  
844/844 - 13s - loss: 0.0014 - accuracy: 0.9997 - val_loss: 0.0412 -  
val_accuracy: 0.9920 - 13s/epoch - 16ms/step  
Epoch 14/20  
844/844 - 14s - loss: 4.3692e-04 - accuracy: 1.0000 - val_loss: 0.0407  
- val_accuracy: 0.9925 - 14s/epoch - 16ms/step  
Epoch 15/20  
844/844 - 14s - loss: 8.4079e-04 - accuracy: 0.9998 - val_loss: 0.0430  
- val_accuracy: 0.9922 - 14s/epoch - 17ms/step  
Epoch 16/20  
844/844 - 14s - loss: 3.2015e-04 - accuracy: 0.9999 - val_loss: 0.0428  
- val_accuracy: 0.9922 - 14s/epoch - 16ms/step  
Epoch 17/20  
844/844 - 14s - loss: 1.7971e-04 - accuracy: 1.0000 - val_loss: 0.0428  
- val_accuracy: 0.9923 - 14s/epoch - 17ms/step  
Epoch 18/20  
844/844 - 14s - loss: 1.4577e-04 - accuracy: 1.0000 - val_loss: 0.0435  
- val_accuracy: 0.9923 - 14s/epoch - 16ms/step  
Epoch 19/20  
844/844 - 14s - loss: 1.3057e-04 - accuracy: 1.0000 - val_loss: 0.0441  
- val_accuracy: 0.9923 - 14s/epoch - 16ms/step  
Epoch 20/20  
844/844 - 14s - loss: 1.1655e-04 - accuracy: 1.0000 - val_loss: 0.0454  
- val_accuracy: 0.9927 - 14s/epoch - 16ms/step  
Running model for 64 batch size and 20 epochs....  
313/313 [=====] - 1s 4ms/step
```

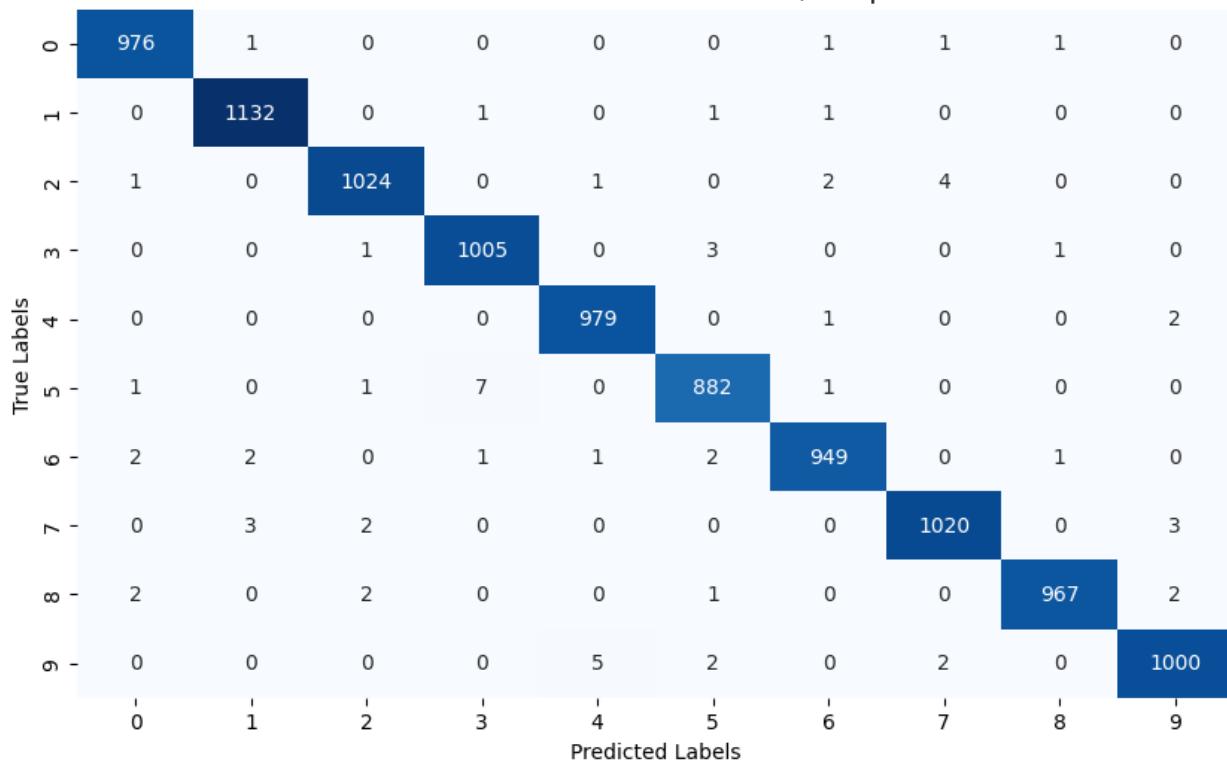
Confusion Matrix...

```
[[ 976   1   0   0   0   0   1   1   1   0]
 [  0 1132   0   1   0   1   1   0   0   0]
 [  1   0 1024   0   1   0   2   4   0   0]
 [  0   0   1 1005   0   3   0   0   1   0]
 [  0   0   0   0  979   0   1   0   0   2]
 [  1   0   1   7   0  882   1   0   0   0]
 [  2   2   0   1   1   2  949   0   1   0]
 [  0   3   2   0   0   0   0 1020   0   3]
 [  2   0   2   0   0   1   0   0   967   2]
 [  0   0   0   0   5   2   0   2   0 1000]]
```

Precision score: 0.9934

Recall: 0.9934

Confusion Matrix for 64 batch size, 20 epochs



Epoch 1/5

422/422 - 13s - loss: 0.2034 - accuracy: 0.9364 - val_loss: 0.0687 - val_accuracy: 0.9793 - 13s/epoch - 31ms/step

Epoch 2/5

422/422 - 11s - loss: 0.0612 - accuracy: 0.9809 - val_loss: 0.0484 - val_accuracy: 0.9850 - 11s/epoch - 27ms/step

Epoch 3/5

422/422 - 12s - loss: 0.0417 - accuracy: 0.9872 - val_loss: 0.0425 - val_accuracy: 0.9875 - 12s/epoch - 28ms/step

Epoch 4/5

```
422/422 - 12s - loss: 0.0309 - accuracy: 0.9904 - val_loss: 0.0357 -  
val_accuracy: 0.9918 - 12s/epoch - 28ms/step
```

Epoch 5/5

```
422/422 - 12s - loss: 0.0244 - accuracy: 0.9923 - val_loss: 0.0335 -  
val_accuracy: 0.9902 - 12s/epoch - 28ms/step
```

Running model for 128 batch size and 5 epochs....

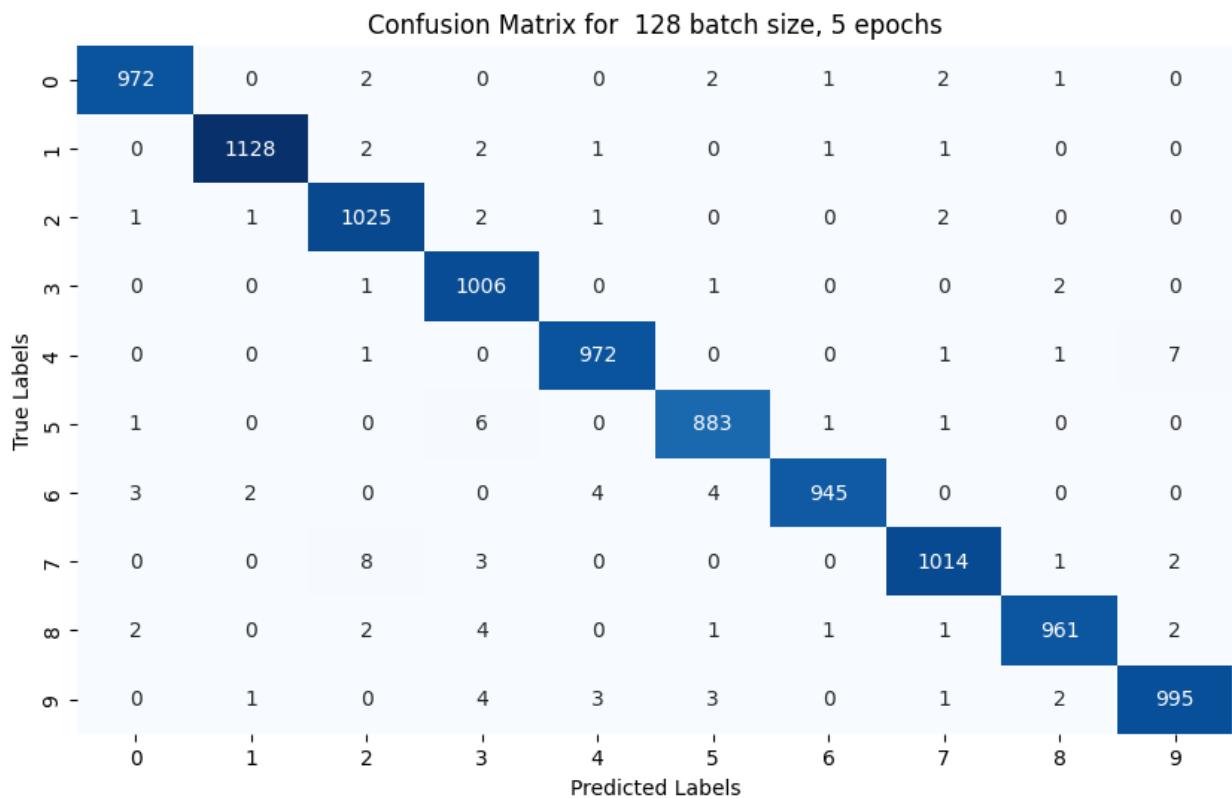
```
313/313 [=====] - 1s 4ms/step
```

Confusion Matrix...

```
[[ 972    0    2    0    0    2    1    2    1    0]  
 [ 0 1128    2    2    1    0    1    1    0    0]  
 [ 1    1 1025    2    1    0    0    2    0    0]  
 [ 0    0    1 1006    0    1    0    0    2    0]  
 [ 0    0    1    0 972    0    0    1    1    7]  
 [ 1    0    0    6    0 883    1    1    0    0]  
 [ 3    2    0    0    4    4 945    0    0    0]  
 [ 0    0    8    3    0    0    0 1014    1    2]  
 [ 2    0    2    4    0    1    1    1 961    2]  
 [ 0    1    0    4    3    3    0    1    2 995]]
```

Precision score: 0.9901

Recall: 0.9901



Epoch 1/15

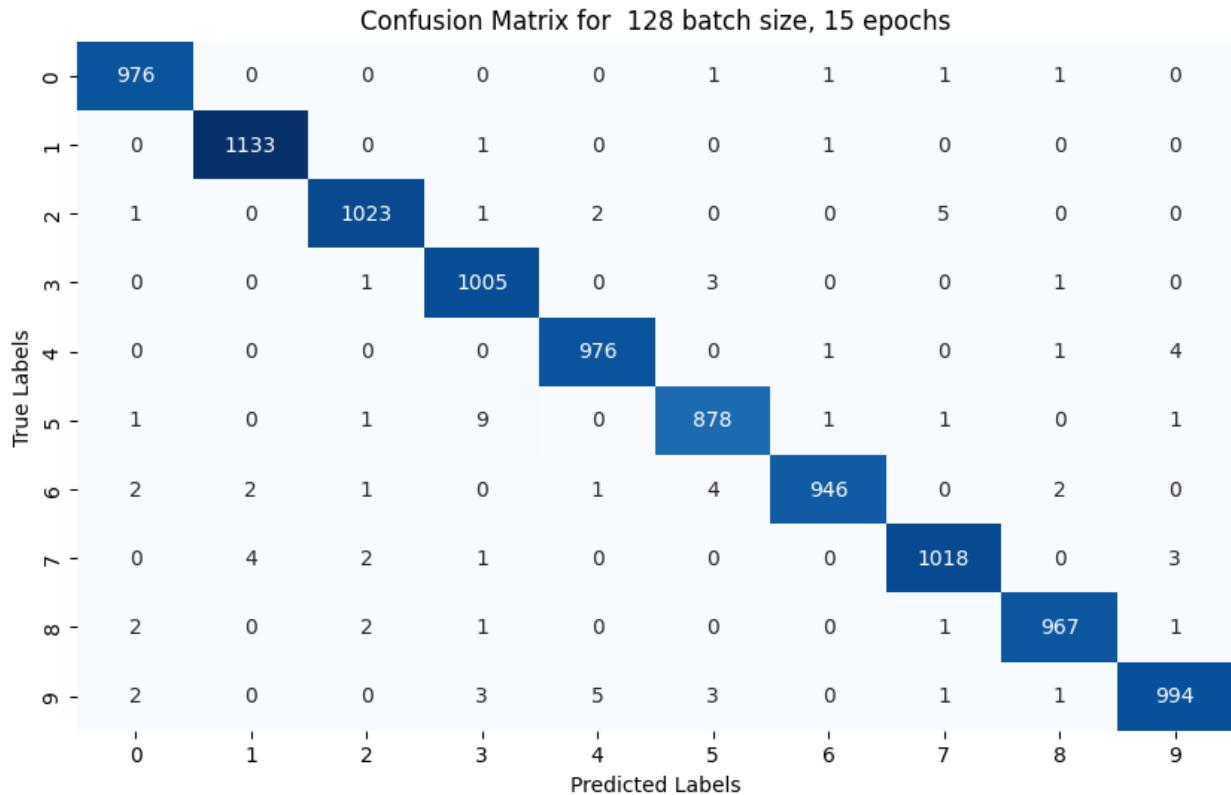
```
422/422 - 12s - loss: 0.2035 - accuracy: 0.9374 - val_loss: 0.0578 -  
val_accuracy: 0.9848 - 12s/epoch - 29ms/step
```

```
Epoch 2/15
422/422 - 12s - loss: 0.0596 - accuracy: 0.9819 - val_loss: 0.0474 -
val_accuracy: 0.9858 - 12s/epoch - 27ms/step
Epoch 3/15
422/422 - 12s - loss: 0.0405 - accuracy: 0.9872 - val_loss: 0.0528 -
val_accuracy: 0.9860 - 12s/epoch - 27ms/step
Epoch 4/15
422/422 - 12s - loss: 0.0301 - accuracy: 0.9903 - val_loss: 0.0401 -
val_accuracy: 0.9887 - 12s/epoch - 28ms/step
Epoch 5/15
422/422 - 12s - loss: 0.0233 - accuracy: 0.9929 - val_loss: 0.0401 -
val_accuracy: 0.9882 - 12s/epoch - 29ms/step
Epoch 6/15
422/422 - 12s - loss: 0.0202 - accuracy: 0.9939 - val_loss: 0.0383 -
val_accuracy: 0.9883 - 12s/epoch - 28ms/step
Epoch 7/15
422/422 - 12s - loss: 0.0155 - accuracy: 0.9953 - val_loss: 0.0370 -
val_accuracy: 0.9902 - 12s/epoch - 28ms/step
Epoch 8/15
422/422 - 12s - loss: 0.0119 - accuracy: 0.9964 - val_loss: 0.0339 -
val_accuracy: 0.9887 - 12s/epoch - 28ms/step
Epoch 9/15
422/422 - 12s - loss: 0.0095 - accuracy: 0.9974 - val_loss: 0.0377 -
val_accuracy: 0.9902 - 12s/epoch - 28ms/step
Epoch 10/15
422/422 - 12s - loss: 0.0077 - accuracy: 0.9978 - val_loss: 0.0351 -
val_accuracy: 0.9917 - 12s/epoch - 27ms/step
Epoch 11/15
422/422 - 12s - loss: 0.0058 - accuracy: 0.9984 - val_loss: 0.0380 -
val_accuracy: 0.9895 - 12s/epoch - 28ms/step
Epoch 12/15
422/422 - 12s - loss: 0.0046 - accuracy: 0.9988 - val_loss: 0.0354 -
val_accuracy: 0.9900 - 12s/epoch - 28ms/step
Epoch 13/15
422/422 - 12s - loss: 0.0034 - accuracy: 0.9993 - val_loss: 0.0352 -
val_accuracy: 0.9908 - 12s/epoch - 28ms/step
Epoch 14/15
422/422 - 12s - loss: 0.0025 - accuracy: 0.9995 - val_loss: 0.0352 -
val_accuracy: 0.9908 - 12s/epoch - 28ms/step
Epoch 15/15
422/422 - 12s - loss: 0.0020 - accuracy: 0.9997 - val_loss: 0.0367 -
val_accuracy: 0.9913 - 12s/epoch - 28ms/step
Running model for 128 batch size and 15 epochs....
313/313 [=====] - 1s 4ms/step
Confusion Matrix...
[[ 976   0   0   0   0   1   1   1   1   0]
 [  0 1133   0   1   0   0   1   0   0   0]
 [  1   0 1023   1   2   0   0   5   0   0]
 [  0   0   1 1005   0   3   0   0   1   0]]
```

```
[ 0  0  0  0 976  0  1  0  1  4]
[ 1  0  1  9  0 878  1  1  0  1]
[ 2  2  1  0  1  4 946  0  2  0]
[ 0  4  2  1  0  0  0 1018  0  3]
[ 2  0  2  1  0  0  0  1 967  1]
[ 2  0  0  3  5  3  0  1  1 994]]
```

Precision score: 0.9916

Recall: 0.9916



Epoch 1/20

422/422 - 13s - loss: 0.2064 - accuracy: 0.9356 - val_loss: 0.0624 - val_accuracy: 0.9818 - 13s/epoch - 32ms/step

Epoch 2/20

422/422 - 12s - loss: 0.0596 - accuracy: 0.9815 - val_loss: 0.0445 - val_accuracy: 0.9877 - 12s/epoch - 28ms/step

Epoch 3/20

422/422 - 12s - loss: 0.0414 - accuracy: 0.9868 - val_loss: 0.0425 - val_accuracy: 0.9890 - 12s/epoch - 28ms/step

Epoch 4/20

422/422 - 12s - loss: 0.0294 - accuracy: 0.9908 - val_loss: 0.0406 - val_accuracy: 0.9893 - 12s/epoch - 28ms/step

Epoch 5/20

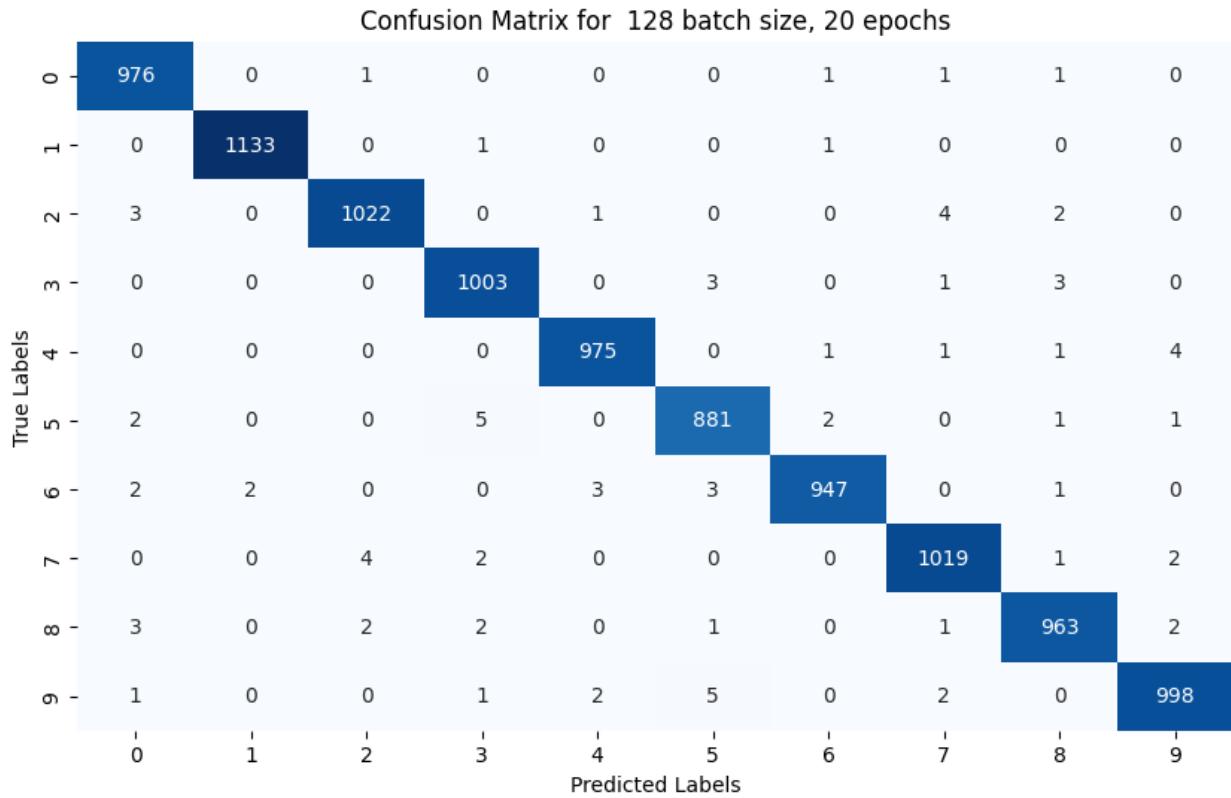
422/422 - 12s - loss: 0.0234 - accuracy: 0.9932 - val_loss: 0.0365 - val_accuracy: 0.9900 - 12s/epoch - 28ms/step

```
Epoch 6/20
422/422 - 12s - loss: 0.0184 - accuracy: 0.9944 - val_loss: 0.0392 -
val_accuracy: 0.9883 - 12s/epoch - 29ms/step
Epoch 7/20
422/422 - 12s - loss: 0.0142 - accuracy: 0.9958 - val_loss: 0.0380 -
val_accuracy: 0.9897 - 12s/epoch - 28ms/step
Epoch 8/20
422/422 - 12s - loss: 0.0119 - accuracy: 0.9964 - val_loss: 0.0335 -
val_accuracy: 0.9910 - 12s/epoch - 27ms/step
Epoch 9/20
422/422 - 11s - loss: 0.0088 - accuracy: 0.9975 - val_loss: 0.0378 -
val_accuracy: 0.9905 - 11s/epoch - 27ms/step
Epoch 10/20
422/422 - 12s - loss: 0.0078 - accuracy: 0.9977 - val_loss: 0.0370 -
val_accuracy: 0.9907 - 12s/epoch - 28ms/step
Epoch 11/20
422/422 - 12s - loss: 0.0056 - accuracy: 0.9985 - val_loss: 0.0353 -
val_accuracy: 0.9910 - 12s/epoch - 28ms/step
Epoch 12/20
422/422 - 12s - loss: 0.0041 - accuracy: 0.9990 - val_loss: 0.0333 -
val_accuracy: 0.9915 - 12s/epoch - 28ms/step
Epoch 13/20
422/422 - 12s - loss: 0.0032 - accuracy: 0.9992 - val_loss: 0.0401 -
val_accuracy: 0.9908 - 12s/epoch - 28ms/step
Epoch 14/20
422/422 - 12s - loss: 0.0025 - accuracy: 0.9994 - val_loss: 0.0391 -
val_accuracy: 0.9918 - 12s/epoch - 28ms/step
Epoch 15/20
422/422 - 12s - loss: 0.0018 - accuracy: 0.9997 - val_loss: 0.0376 -
val_accuracy: 0.9915 - 12s/epoch - 28ms/step
Epoch 16/20
422/422 - 12s - loss: 0.0019 - accuracy: 0.9996 - val_loss: 0.0395 -
val_accuracy: 0.9912 - 12s/epoch - 27ms/step
Epoch 17/20
422/422 - 12s - loss: 0.0013 - accuracy: 0.9998 - val_loss: 0.0405 -
val_accuracy: 0.9917 - 12s/epoch - 28ms/step
Epoch 18/20
422/422 - 12s - loss: 0.0010 - accuracy: 0.9999 - val_loss: 0.0403 -
val_accuracy: 0.9922 - 12s/epoch - 28ms/step
Epoch 19/20
422/422 - 12s - loss: 7.0752e-04 - accuracy: 0.9999 - val_loss: 0.0398 -
val_accuracy: 0.9927 - 12s/epoch - 28ms/step
Epoch 20/20
422/422 - 12s - loss: 5.0571e-04 - accuracy: 1.0000 - val_loss: 0.0414 -
val_accuracy: 0.9925 - 12s/epoch - 28ms/step
Running model for 128 batch size and 20 epochs....
313/313 [=====] - 1s 4ms/step
Confusion Matrix...
[[ 976  0  1  0  0  0  1  1  1  0]]
```

```
[ 0 1133 0 1 0 0 1 0 0 0]
[ 3 0 1022 0 1 0 0 4 2 0]
[ 0 0 0 1003 0 3 0 1 3 0]
[ 0 0 0 0 975 0 1 1 1 4]
[ 2 0 0 5 0 881 2 0 1 1]
[ 2 2 0 0 3 3 947 0 1 0]
[ 0 0 4 2 0 0 0 1019 1 2]
[ 3 0 2 2 0 1 0 1 963 2]
[ 1 0 0 1 2 5 0 2 0 998]]
```

Precision score: 0.9917

Recall: 0.9917



Epoch 1/5

211/211 - 14s - loss: 0.2972 - accuracy: 0.9079 - val_loss: 0.0783 - val_accuracy: 0.9770 - 14s/epoch - 65ms/step

Epoch 2/5

211/211 - 13s - loss: 0.0787 - accuracy: 0.9761 - val_loss: 0.0577 - val_accuracy: 0.9827 - 13s/epoch - 60ms/step

Epoch 3/5

211/211 - 12s - loss: 0.0553 - accuracy: 0.9834 - val_loss: 0.0506 - val_accuracy: 0.9862 - 12s/epoch - 58ms/step

Epoch 4/5

211/211 - 13s - loss: 0.0432 - accuracy: 0.9868 - val_loss: 0.0446 - val_accuracy: 0.9872 - 13s/epoch - 59ms/step

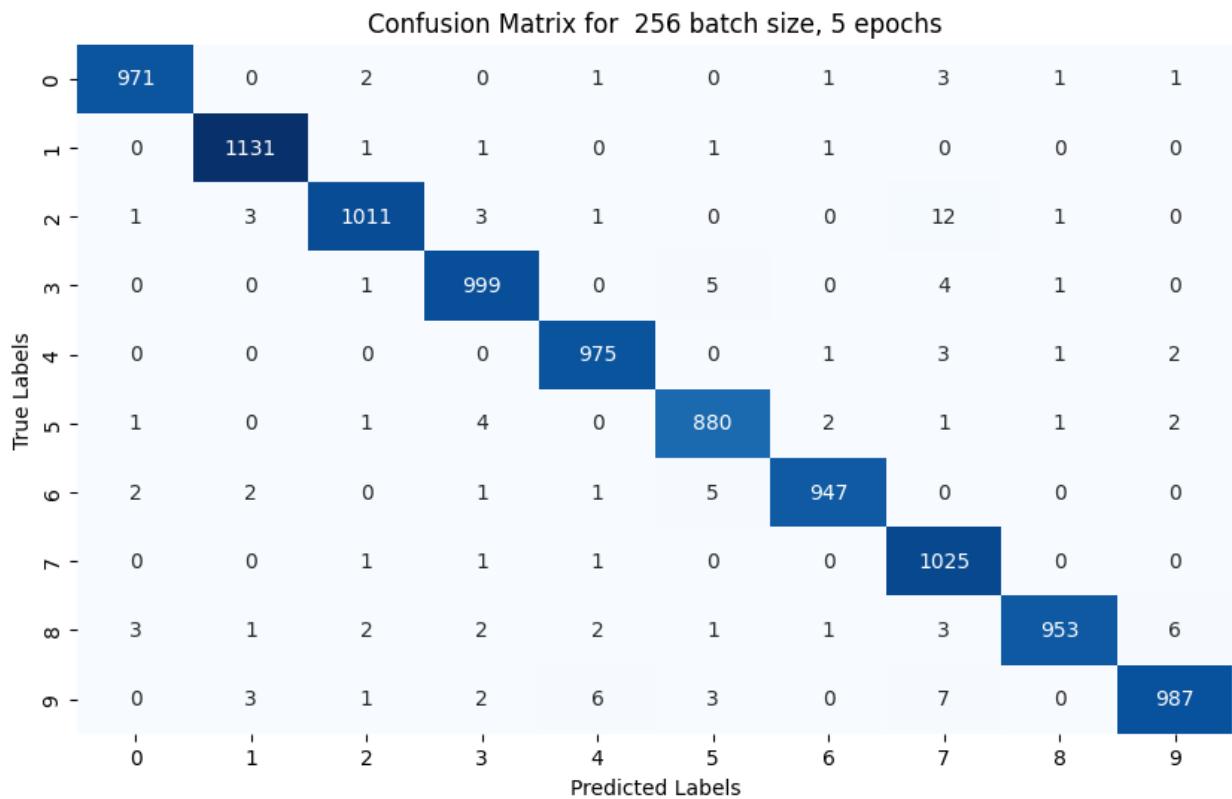
```

Epoch 5/5
211/211 - 12s - loss: 0.0365 - accuracy: 0.9887 - val_loss: 0.0384 -
val_accuracy: 0.9898 - 12s/epoch - 59ms/step
Running model for 256 batch size and 5 epochs....
313/313 [=====] - 1s 5ms/step
Confusion Matrix...
[[ 971   0   2   0   1   0   1   3   1   1]
 [ 0 1131   1   1   0   1   1   0   0   0]
 [ 1   3 1011   3   1   0   0   12   1   0]
 [ 0   0   1 999   0   5   0   4   1   0]
 [ 0   0   0   0 975   0   1   3   1   2]
 [ 1   0   1   4   0 880   2   1   1   2]
 [ 2   2   0   1   1   5 947   0   0   0]
 [ 0   0   1   1   1   0   0 1025   0   0]
 [ 3   1   2   2   2   1   1   3 953   6]
 [ 0   3   1   2   6   3   0   7   0 987]]

```

Precision score: 0.9880

Recall: 0.9879



```

Epoch 1/15
211/211 - 13s - loss: 0.3003 - accuracy: 0.9046 - val_loss: 0.0787 -
val_accuracy: 0.9802 - 13s/epoch - 63ms/step
Epoch 2/15
211/211 - 12s - loss: 0.0808 - accuracy: 0.9753 - val_loss: 0.0692 -

```

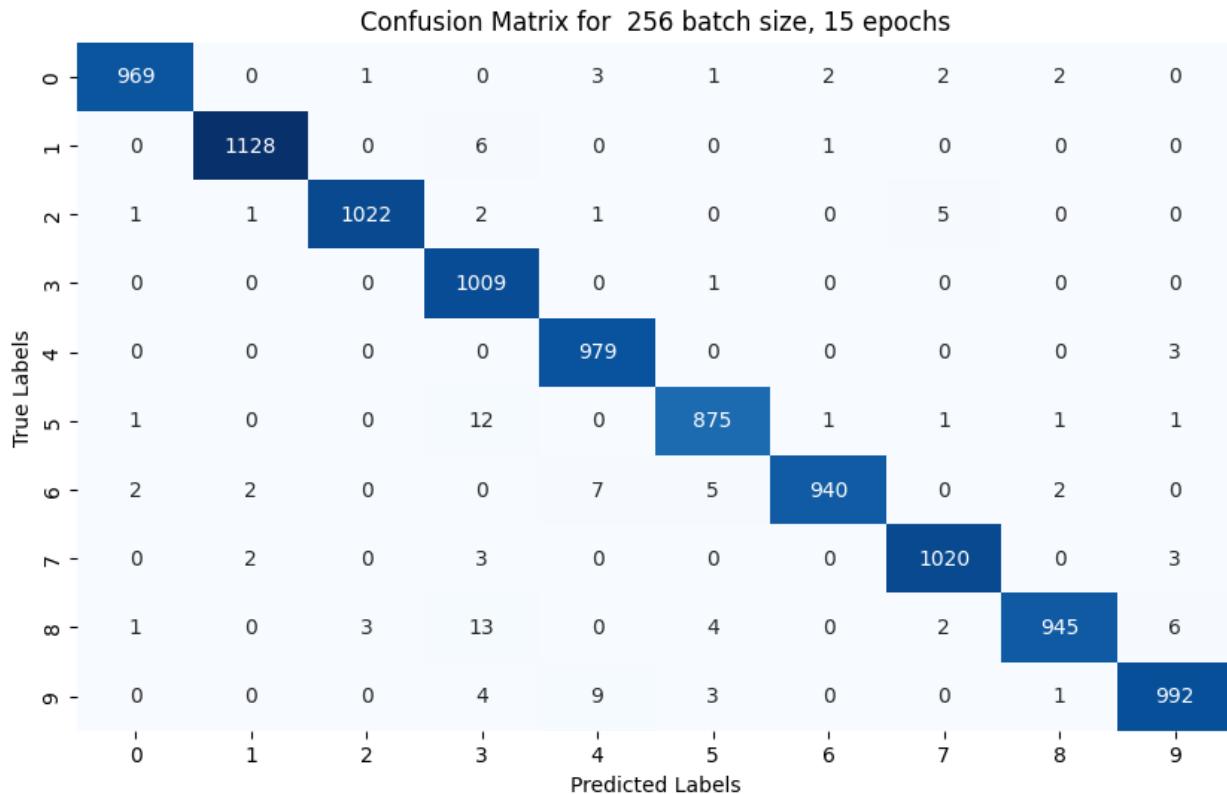
```

val_accuracy: 0.9807 - 12s/epoch - 58ms/step
Epoch 3/15
211/211 - 12s - loss: 0.0587 - accuracy: 0.9820 - val_loss: 0.0625 -
val_accuracy: 0.9828 - 12s/epoch - 57ms/step
Epoch 4/15
211/211 - 12s - loss: 0.0469 - accuracy: 0.9857 - val_loss: 0.0587 -
val_accuracy: 0.9828 - 12s/epoch - 56ms/step
Epoch 5/15
211/211 - 12s - loss: 0.0369 - accuracy: 0.9886 - val_loss: 0.0491 -
val_accuracy: 0.9865 - 12s/epoch - 58ms/step
Epoch 6/15
211/211 - 12s - loss: 0.0314 - accuracy: 0.9906 - val_loss: 0.0417 -
val_accuracy: 0.9877 - 12s/epoch - 59ms/step
Epoch 7/15
211/211 - 13s - loss: 0.0266 - accuracy: 0.9914 - val_loss: 0.0391 -
val_accuracy: 0.9905 - 13s/epoch - 60ms/step
Epoch 8/15
211/211 - 13s - loss: 0.0227 - accuracy: 0.9934 - val_loss: 0.0425 -
val_accuracy: 0.9882 - 13s/epoch - 60ms/step
Epoch 9/15
211/211 - 13s - loss: 0.0187 - accuracy: 0.9943 - val_loss: 0.0421 -
val_accuracy: 0.9880 - 13s/epoch - 59ms/step
Epoch 10/15
211/211 - 12s - loss: 0.0160 - accuracy: 0.9953 - val_loss: 0.0418 -
val_accuracy: 0.9897 - 12s/epoch - 59ms/step
Epoch 11/15
211/211 - 12s - loss: 0.0152 - accuracy: 0.9955 - val_loss: 0.0408 -
val_accuracy: 0.9890 - 12s/epoch - 59ms/step
Epoch 12/15
211/211 - 13s - loss: 0.0117 - accuracy: 0.9966 - val_loss: 0.0385 -
val_accuracy: 0.9897 - 13s/epoch - 60ms/step
Epoch 13/15
211/211 - 12s - loss: 0.0105 - accuracy: 0.9971 - val_loss: 0.0365 -
val_accuracy: 0.9895 - 12s/epoch - 59ms/step
Epoch 14/15
211/211 - 12s - loss: 0.0101 - accuracy: 0.9971 - val_loss: 0.0384 -
val_accuracy: 0.9897 - 12s/epoch - 58ms/step
Epoch 15/15
211/211 - 13s - loss: 0.0080 - accuracy: 0.9978 - val_loss: 0.0448 -
val_accuracy: 0.9887 - 13s/epoch - 60ms/step
Running model for 256 batch size and 15 epochs....
313/313 [=====] - 1s 5ms/step
Confusion Matrix...
[[ 969   0   1   0   3   1   2   2   2   0]
 [  0 1128   0   6   0   0   1   0   0   0]
 [  1   1 1022   2   1   0   0   5   0   0]
 [  0   0   0 1009   0   1   0   0   0   0]
 [  0   0   0   0  979   0   0   0   0   3]
 [  1   0   0   12   0  875   1   1   1   1]]
```

```
[ 2 2 0 0 7 5 940 0 2 0]
[ 0 2 0 3 0 0 0 1020 0 3]
[ 1 0 3 13 0 4 0 2 945 6]
[ 0 0 0 4 9 3 0 0 1 992]]
```

Precision score: 0.9880

Recall: 0.9879



Epoch 1/20

211/211 - 13s - loss: 0.2762 - accuracy: 0.9129 - val_loss: 0.0774 - val_accuracy: 0.9773 - 13s/epoch - 63ms/step

Epoch 2/20

211/211 - 13s - loss: 0.0758 - accuracy: 0.9769 - val_loss: 0.0556 - val_accuracy: 0.9852 - 13s/epoch - 60ms/step

Epoch 3/20

211/211 - 12s - loss: 0.0564 - accuracy: 0.9826 - val_loss: 0.0497 - val_accuracy: 0.9865 - 12s/epoch - 59ms/step

Epoch 4/20

211/211 - 13s - loss: 0.0442 - accuracy: 0.9866 - val_loss: 0.0455 - val_accuracy: 0.9867 - 13s/epoch - 60ms/step

Epoch 5/20

211/211 - 13s - loss: 0.0369 - accuracy: 0.9891 - val_loss: 0.0667 - val_accuracy: 0.9815 - 13s/epoch - 60ms/step

Epoch 6/20

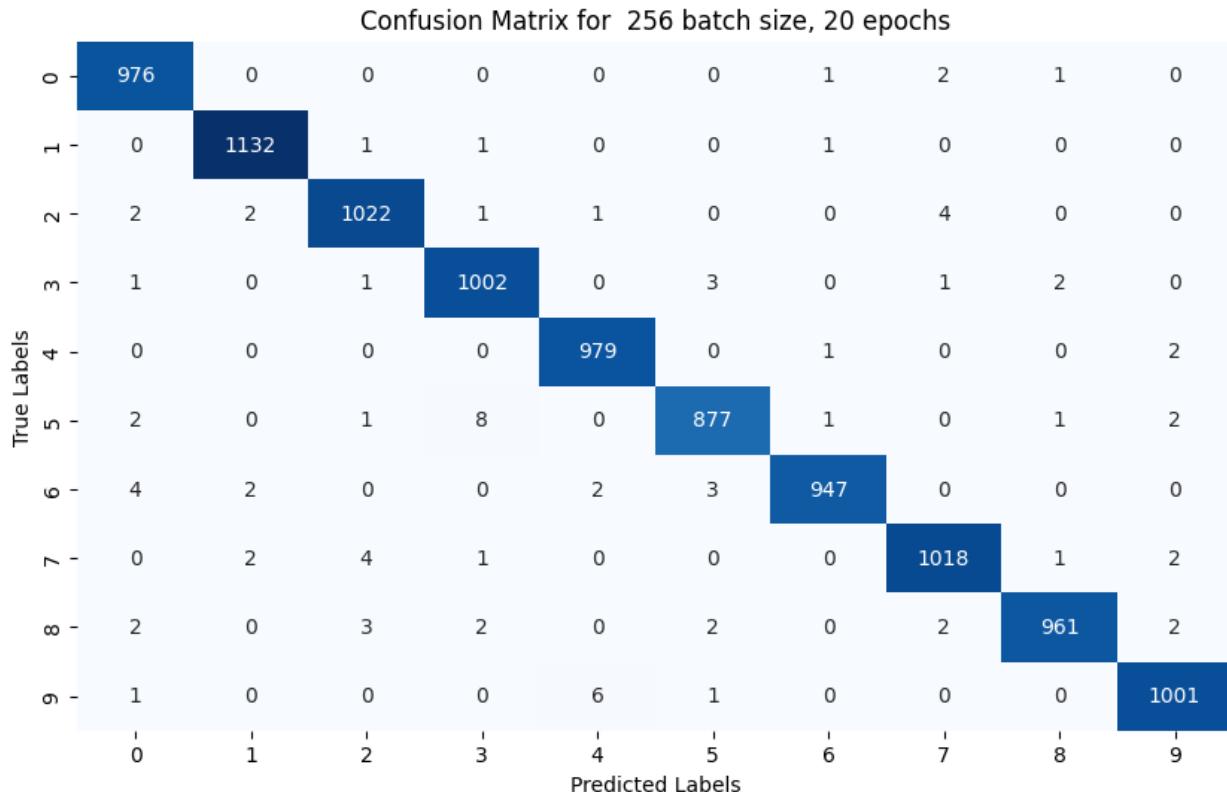
211/211 - 13s - loss: 0.0314 - accuracy: 0.9906 - val_loss: 0.0384 -

```
val_accuracy: 0.9902 - 13s/epoch - 60ms/step
Epoch 7/20
211/211 - 12s - loss: 0.0261 - accuracy: 0.9920 - val_loss: 0.0508 -
val_accuracy: 0.9873 - 12s/epoch - 58ms/step
Epoch 8/20
211/211 - 12s - loss: 0.0215 - accuracy: 0.9935 - val_loss: 0.0429 -
val_accuracy: 0.9883 - 12s/epoch - 59ms/step
Epoch 9/20
211/211 - 13s - loss: 0.0192 - accuracy: 0.9945 - val_loss: 0.0450 -
val_accuracy: 0.9870 - 13s/epoch - 62ms/step
Epoch 10/20
211/211 - 13s - loss: 0.0169 - accuracy: 0.9946 - val_loss: 0.0421 -
val_accuracy: 0.9885 - 13s/epoch - 60ms/step
Epoch 11/20
211/211 - 13s - loss: 0.0137 - accuracy: 0.9960 - val_loss: 0.0401 -
val_accuracy: 0.9907 - 13s/epoch - 62ms/step
Epoch 12/20
211/211 - 13s - loss: 0.0118 - accuracy: 0.9967 - val_loss: 0.0373 -
val_accuracy: 0.9907 - 13s/epoch - 62ms/step
Epoch 13/20
211/211 - 13s - loss: 0.0100 - accuracy: 0.9973 - val_loss: 0.0390 -
val_accuracy: 0.9913 - 13s/epoch - 61ms/step
Epoch 14/20
211/211 - 13s - loss: 0.0087 - accuracy: 0.9977 - val_loss: 0.0373 -
val_accuracy: 0.9905 - 13s/epoch - 62ms/step
Epoch 15/20
211/211 - 13s - loss: 0.0074 - accuracy: 0.9981 - val_loss: 0.0385 -
val_accuracy: 0.9910 - 13s/epoch - 61ms/step
Epoch 16/20
211/211 - 13s - loss: 0.0064 - accuracy: 0.9984 - val_loss: 0.0377 -
val_accuracy: 0.9910 - 13s/epoch - 60ms/step
Epoch 17/20
211/211 - 13s - loss: 0.0054 - accuracy: 0.9988 - val_loss: 0.0370 -
val_accuracy: 0.9898 - 13s/epoch - 60ms/step
Epoch 18/20
211/211 - 13s - loss: 0.0045 - accuracy: 0.9991 - val_loss: 0.0374 -
val_accuracy: 0.9913 - 13s/epoch - 60ms/step
Epoch 19/20
211/211 - 13s - loss: 0.0037 - accuracy: 0.9993 - val_loss: 0.0388 -
val_accuracy: 0.9917 - 13s/epoch - 60ms/step
Epoch 20/20
211/211 - 13s - loss: 0.0037 - accuracy: 0.9993 - val_loss: 0.0387 -
val_accuracy: 0.9910 - 13s/epoch - 59ms/step
Running model for 256 batch size and 20 epochs....
313/313 [=====] - 1s 4ms/step
Confusion Matrix...
[[ 976   0   0   0   0   0   1   2   1   0]
 [  0 1132   1   1   0   0   1   0   0   0]
 [  2   2 1022   1   1   0   0   4   0   0]]
```

```
[ 1 0 1 1002 0 3 0 1 2 0]
[ 0 0 0 0 979 0 1 0 0 2]
[ 2 0 1 8 0 877 1 0 1 2]
[ 4 2 0 0 2 3 947 0 0 0]
[ 0 2 4 1 0 0 0 1018 1 2]
[ 2 0 3 2 0 2 0 2 961 2]
[ 1 0 0 0 6 1 0 0 0 1001]]
```

Precision score: 0.9915

Recall: 0.9915



Epoch 1/5

3375/3375 - 32s - loss: 0.1323 - accuracy: 0.9591 - val_loss: 0.0667 - val_accuracy: 0.9815 - 32s/epoch - 9ms/step

Epoch 2/5

3375/3375 - 31s - loss: 0.0430 - accuracy: 0.9865 - val_loss: 0.0401 - val_accuracy: 0.9890 - 31s/epoch - 9ms/step

Epoch 3/5

3375/3375 - 32s - loss: 0.0262 - accuracy: 0.9916 - val_loss: 0.0417 - val_accuracy: 0.9892 - 32s/epoch - 9ms/step

Epoch 4/5

3375/3375 - 31s - loss: 0.0168 - accuracy: 0.9949 - val_loss: 0.0355 - val_accuracy: 0.9918 - 31s/epoch - 9ms/step

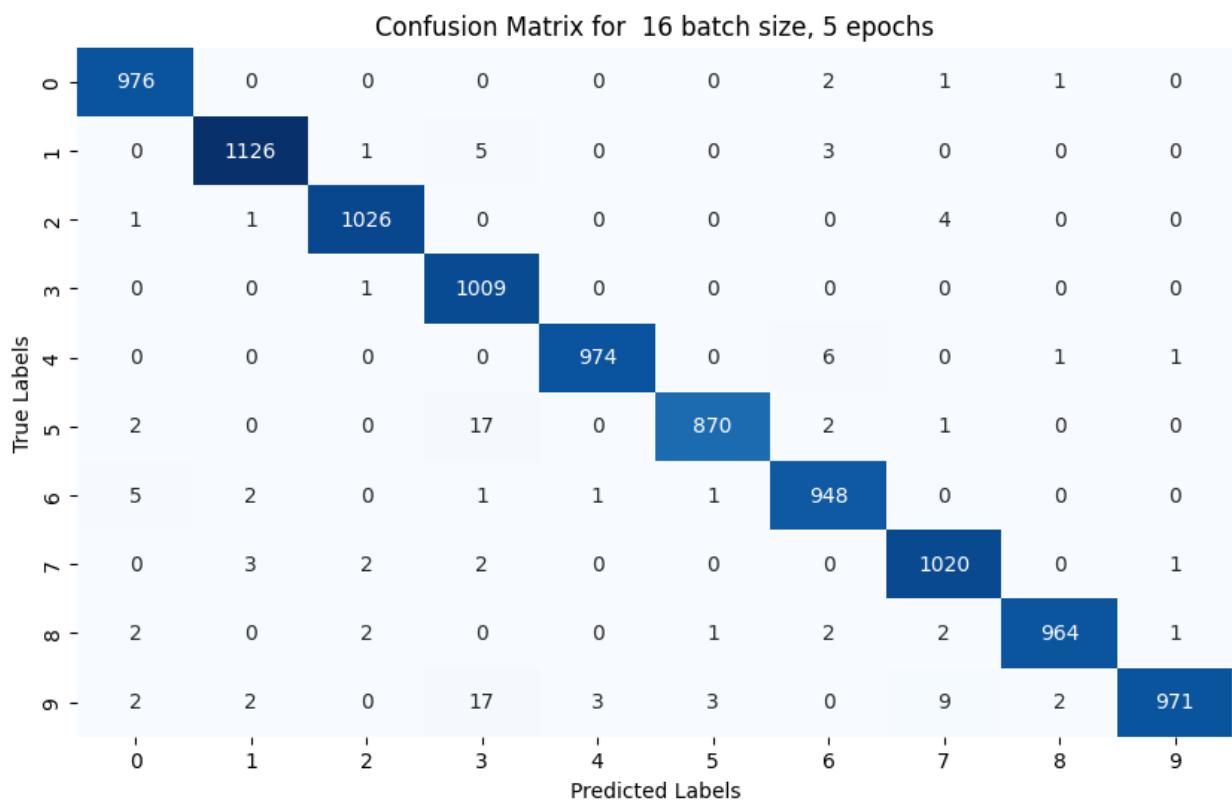
Epoch 5/5

3375/3375 - 31s - loss: 0.0143 - accuracy: 0.9955 - val_loss: 0.0492 -

```

val_accuracy: 0.9882 - 31s/epoch - 9ms/step
Running model for 16 batch size and 5 epochs....
313/313 [=====] - 1s 4ms/step
Confusion Matrix...
[[ 976 0 0 0 0 0 2 1 1 0]
 [ 0 1126 1 5 0 0 3 0 0 0]
 [ 1 1 1026 0 0 0 0 4 0 0]
 [ 0 0 1 1009 0 0 0 0 0 0]
 [ 0 0 0 0 974 0 6 0 1 1]
 [ 2 0 0 17 0 870 2 1 0 0]
 [ 5 2 0 1 1 1 948 0 0 0]
 [ 0 3 2 2 0 0 0 0 1020 0]
 [ 2 0 2 0 0 1 2 2 964 1]
 [ 2 2 0 17 3 3 0 9 2 971]]
Precision score: 0.9886
Recall: 0.9884

```



```

Epoch 1/15
3375/3375 - 32s - loss: 0.1260 - accuracy: 0.9602 - val_loss: 0.0477 -
val_accuracy: 0.9850 - 32s/epoch - 9ms/step
Epoch 2/15
3375/3375 - 30s - loss: 0.0405 - accuracy: 0.9877 - val_loss: 0.0398 -
val_accuracy: 0.9887 - 30s/epoch - 9ms/step
Epoch 3/15

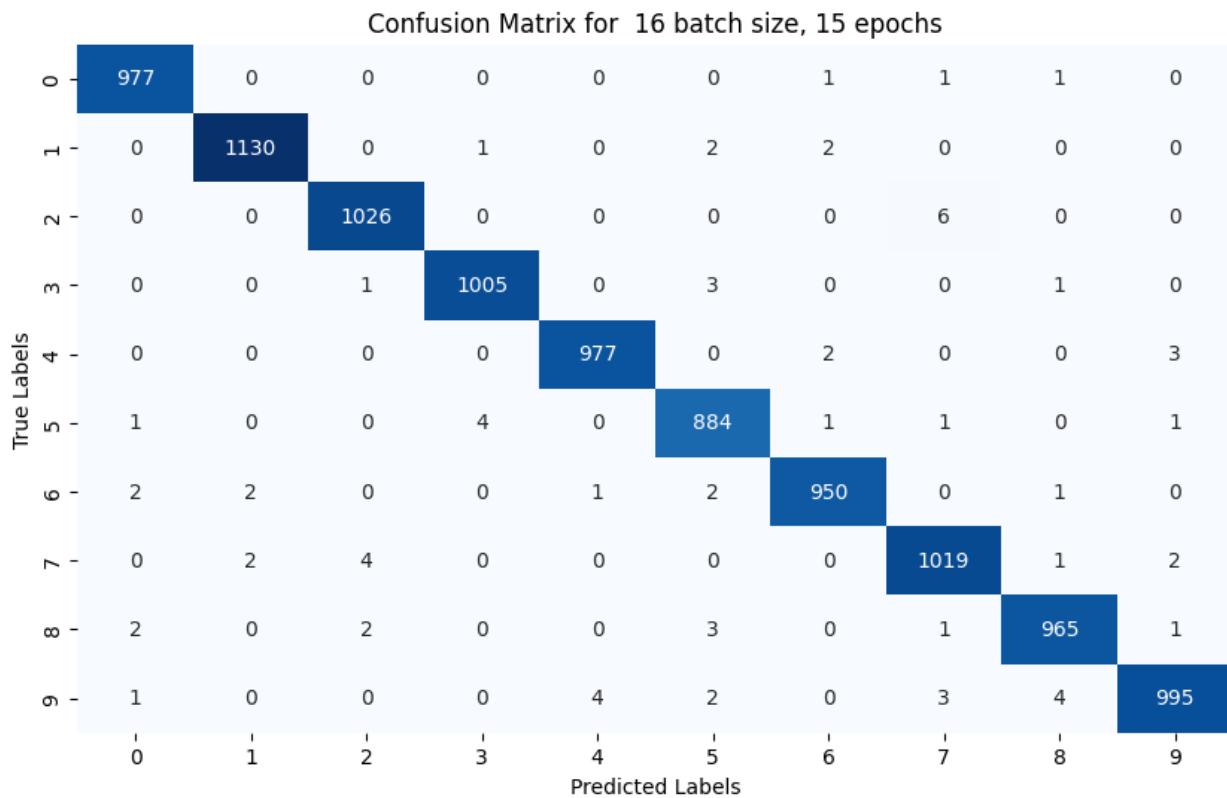
```

```
3375/3375 - 30s - loss: 0.0255 - accuracy: 0.9918 - val_loss: 0.0399 -  
val_accuracy: 0.9890 - 30s/epoch - 9ms/step  
Epoch 4/15  
3375/3375 - 31s - loss: 0.0175 - accuracy: 0.9943 - val_loss: 0.0422 -  
val_accuracy: 0.9902 - 31s/epoch - 9ms/step  
Epoch 5/15  
3375/3375 - 30s - loss: 0.0129 - accuracy: 0.9959 - val_loss: 0.0402 -  
val_accuracy: 0.9905 - 30s/epoch - 9ms/step  
Epoch 6/15  
3375/3375 - 30s - loss: 0.0083 - accuracy: 0.9973 - val_loss: 0.0459 -  
val_accuracy: 0.9905 - 30s/epoch - 9ms/step  
Epoch 7/15  
3375/3375 - 31s - loss: 0.0054 - accuracy: 0.9981 - val_loss: 0.0522 -  
val_accuracy: 0.9897 - 31s/epoch - 9ms/step  
Epoch 8/15  
3375/3375 - 30s - loss: 0.0047 - accuracy: 0.9986 - val_loss: 0.0450 -  
val_accuracy: 0.9912 - 30s/epoch - 9ms/step  
Epoch 9/15  
3375/3375 - 31s - loss: 0.0022 - accuracy: 0.9994 - val_loss: 0.0496 -  
val_accuracy: 0.9908 - 31s/epoch - 9ms/step  
Epoch 10/15  
3375/3375 - 31s - loss: 0.0028 - accuracy: 0.9992 - val_loss: 0.0646 -  
val_accuracy: 0.9880 - 31s/epoch - 9ms/step  
Epoch 11/15  
3375/3375 - 30s - loss: 0.0046 - accuracy: 0.9985 - val_loss: 0.0594 -  
val_accuracy: 0.9882 - 30s/epoch - 9ms/step  
Epoch 12/15  
3375/3375 - 30s - loss: 0.0017 - accuracy: 0.9996 - val_loss: 0.0438 -  
val_accuracy: 0.9930 - 30s/epoch - 9ms/step  
Epoch 13/15  
3375/3375 - 30s - loss: 4.4171e-04 - accuracy: 1.0000 - val_loss:  
0.0502 - val_accuracy: 0.9925 - 30s/epoch - 9ms/step  
Epoch 14/15  
3375/3375 - 31s - loss: 6.4443e-05 - accuracy: 1.0000 - val_loss:  
0.0509 - val_accuracy: 0.9928 - 31s/epoch - 9ms/step  
Epoch 15/15  
3375/3375 - 31s - loss: 3.4671e-05 - accuracy: 1.0000 - val_loss:  
0.0524 - val_accuracy: 0.9927 - 31s/epoch - 9ms/step  
Running model for 16 batch size and 15 epochs....  
313/313 [=====] - 1s 4ms/step  
Confusion Matrix...  
[[ 977  0  0  0  0  0  1  1  1  0]  
 [ 0 1130  0  1  0  2  2  0  0  0]  
 [ 0  0 1026  0  0  0  0  6  0  0]  
 [ 0  0  1 1005  0  3  0  0  1  0]  
 [ 0  0  0  977  0  2  0  0  0  3]  
 [ 1  0  0  4  0  884  1  1  0  1]  
 [ 2  2  0  0  1  2  950  0  1  0]  
 [ 0  2  4  0  0  0  0 1019  1  2]]
```

```
[ 2 0 2 0 0 3 0 1 965 1]
[ 1 0 0 0 4 2 0 3 4 995]]
```

Precision score: 0.9928

Recall: 0.9928



Epoch 1/20

```
3375/3375 - 33s - loss: 0.1235 - accuracy: 0.9607 - val_loss: 0.0508 -
val_accuracy: 0.9842 - 33s/epoch - 10ms/step
```

Epoch 2/20

```
3375/3375 - 29s - loss: 0.0427 - accuracy: 0.9866 - val_loss: 0.0447 -
val_accuracy: 0.9872 - 29s/epoch - 8ms/step
```

Epoch 3/20

```
3375/3375 - 27s - loss: 0.0276 - accuracy: 0.9916 - val_loss: 0.0372 -
val_accuracy: 0.9900 - 27s/epoch - 8ms/step
```

Epoch 4/20

```
3375/3375 - 27s - loss: 0.0183 - accuracy: 0.9940 - val_loss: 0.0354 -
val_accuracy: 0.9915 - 27s/epoch - 8ms/step
```

Epoch 5/20

```
3375/3375 - 27s - loss: 0.0135 - accuracy: 0.9957 - val_loss: 0.0380 -
val_accuracy: 0.9920 - 27s/epoch - 8ms/step
```

Epoch 6/20

```
3375/3375 - 27s - loss: 0.0095 - accuracy: 0.9966 - val_loss: 0.0427 -
val_accuracy: 0.9912 - 27s/epoch - 8ms/step
```

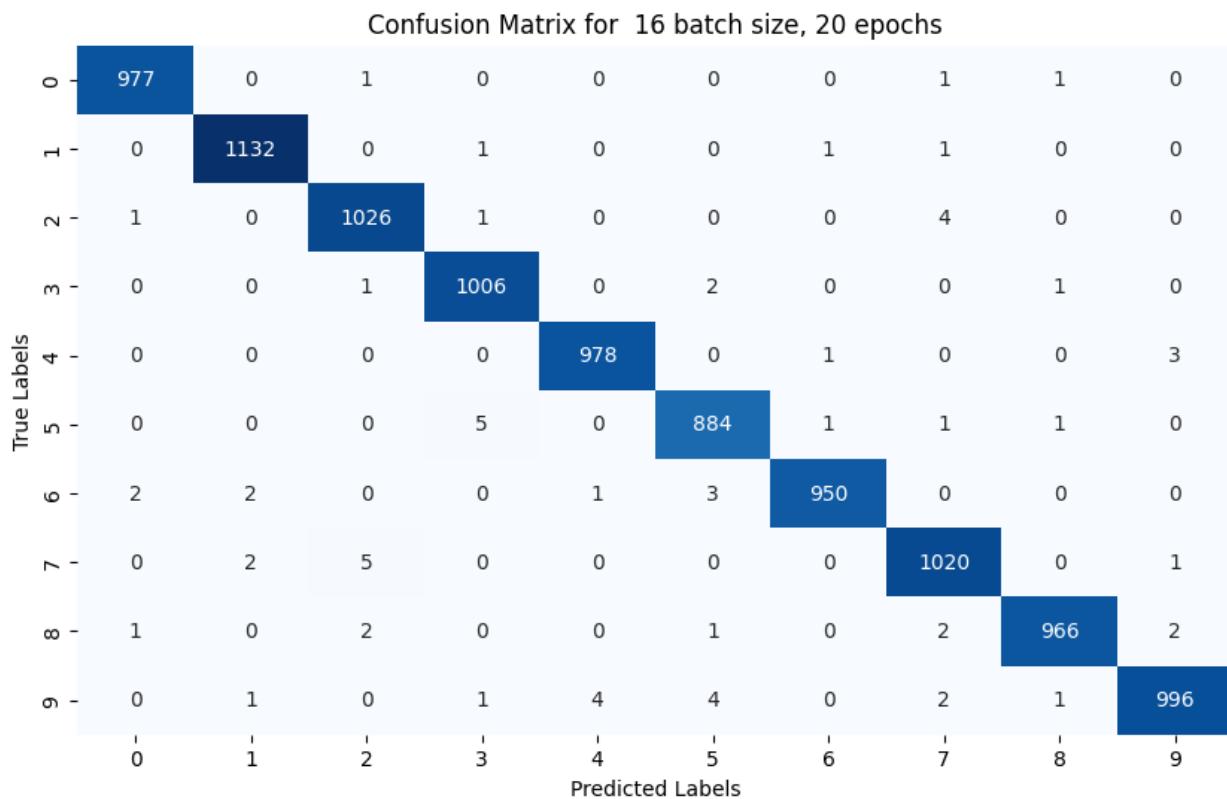
Epoch 7/20

```
3375/3375 - 27s - loss: 0.0072 - accuracy: 0.9977 - val_loss: 0.0472 -  
val_accuracy: 0.9900 - 27s/epoch - 8ms/step  
Epoch 8/20  
3375/3375 - 27s - loss: 0.0064 - accuracy: 0.9981 - val_loss: 0.0574 -  
val_accuracy: 0.9877 - 27s/epoch - 8ms/step  
Epoch 9/20  
3375/3375 - 28s - loss: 0.0059 - accuracy: 0.9978 - val_loss: 0.0439 -  
val_accuracy: 0.9922 - 28s/epoch - 8ms/step  
Epoch 10/20  
3375/3375 - 26s - loss: 0.0052 - accuracy: 0.9984 - val_loss: 0.0453 -  
val_accuracy: 0.9908 - 26s/epoch - 8ms/step  
Epoch 11/20  
3375/3375 - 27s - loss: 0.0045 - accuracy: 0.9987 - val_loss: 0.0382 -  
val_accuracy: 0.9915 - 27s/epoch - 8ms/step  
Epoch 12/20  
3375/3375 - 27s - loss: 0.0023 - accuracy: 0.9993 - val_loss: 0.0436 -  
val_accuracy: 0.9930 - 27s/epoch - 8ms/step  
Epoch 13/20  
3375/3375 - 26s - loss: 0.0029 - accuracy: 0.9992 - val_loss: 0.0366 -  
val_accuracy: 0.9935 - 26s/epoch - 8ms/step  
Epoch 14/20  
3375/3375 - 26s - loss: 7.2596e-04 - accuracy: 0.9999 - val_loss:  
0.0436 - val_accuracy: 0.9930 - 26s/epoch - 8ms/step  
Epoch 15/20  
3375/3375 - 26s - loss: 2.3782e-04 - accuracy: 0.9999 - val_loss:  
0.0438 - val_accuracy: 0.9937 - 26s/epoch - 8ms/step  
Epoch 16/20  
3375/3375 - 27s - loss: 4.3406e-05 - accuracy: 1.0000 - val_loss:  
0.0442 - val_accuracy: 0.9938 - 27s/epoch - 8ms/step  
Epoch 17/20  
3375/3375 - 27s - loss: 2.7584e-05 - accuracy: 1.0000 - val_loss:  
0.0451 - val_accuracy: 0.9935 - 27s/epoch - 8ms/step  
Epoch 18/20  
3375/3375 - 27s - loss: 2.1833e-05 - accuracy: 1.0000 - val_loss:  
0.0459 - val_accuracy: 0.9935 - 27s/epoch - 8ms/step  
Epoch 19/20  
3375/3375 - 26s - loss: 1.8409e-05 - accuracy: 1.0000 - val_loss:  
0.0465 - val_accuracy: 0.9935 - 26s/epoch - 8ms/step  
Epoch 20/20  
3375/3375 - 26s - loss: 1.5908e-05 - accuracy: 1.0000 - val_loss:  
0.0469 - val_accuracy: 0.9937 - 26s/epoch - 8ms/step  
Running model for 16 batch size and 20 epochs....  
313/313 [=====] - 1s 4ms/step  
Confusion Matrix...  
[[ 977  0  1  0  0  0  0  1  1  0]  
 [ 0 1132  0  1  0  0  1  1  0  0]  
 [ 1  0 1026  1  0  0  0  4  0  0]  
 [ 0  0  1 1006  0  2  0  0  1  0]  
 [ 0  0  0  0  978  0  1  0  0  3]]
```

```
[ 0  0  0  5  0 884  1  1  1  0]
[ 2  2  0  0  1  3 950  0  0  0]
[ 0  2  5  0  0  0  0 1020  0  1]
[ 1  0  2  0  0  1  0  2 966  2]
[ 0  1  0  1  4  4  0  2  1 996]]
```

Precision score: 0.9935

Recall: 0.9935



###Experimenting with Different Activation Functions in the Inner Layers and with Various Batch Sizes and Epochs:

```
from tensorflow.keras.activations import
relu,sigmoid,softmax,exponential,hard_sigmoid

def baseline_model_exp(conv_activation = "relu", dense_activation =
"relu"):
    ## create model
    model = Sequential()

    ## Hidden layers
    ## Convolutional Layer
    model.add(Conv2D(32,(3,3), activation = conv_activation,
kernel_initializer = "he_uniform", input_shape = (28,28,1)))
    ### Max Pooling Layer
    model.add(MaxPooling2D((2,2)))
```

```

## Flatten Layer
model.add(Flatten())
## Dense Layer (1st Dense Layer)
model.add(Dense(100, activation = dense_activation,
kernel_initializer = "he_uniform"))

## Output layer
model.add(Dense(10, activation = "softmax")) ## output layer
activation function is always fixed for softmax activation function
for multiclass classification problem

### Compile the model
opt = SGD(learning_rate = 0.01, momentum = 0.9)
model.compile(optimizer = opt, loss = "categorical_crossentropy",
metrics = ["accuracy"])
return model

act_func = ["relu", "sigmoid", "softmax", "exponential", "hard_sigmoid"]

for act in act_func:
    for batch in batch_sizes:
        for epoch in epochs_list:
            print(f"Training model with {act} activation function, {batch} batch sizes, {epoch} epochs_list")
            model = baseline_model_exp(conv_activation = act,
dense_activation = act)
            history = model.fit(train_norm,y_train, epochs = epoch,
batch_size = batch, validation_split = 0.1, verbose = 2)

###predictions
y_pred_prob = model.predict(test_norm)
y_pred = np.argmax(y_pred_prob, axis = 1)

##Ensure y_test is not one-hot encoded for confusion matrix

if y_test.ndim > 1:
    y_true = np.argmax(y_test, axis = 1)
else:
    y_true = y_test

###calculating the confusion matrix
cm = confusion_matrix(y_true, y_pred)
### Calculating precision and recall
precision = precision_score(y_true, y_pred, average =
"weighted")
recall = recall_score(y_true, y_pred, average = "weighted")

##Printing results
print(f"Results for {act} function, {batch} batch size and
{epoch} epochs...")

```

```

print(f"Confusion Matrix")
print(cm)
print(f"Precision: {precision: .4f}")
print(f"Recall: {recall: .4f}")

### plotting the confusion matrix
plt.figure(figsize = (10,7))
sns.heatmap(cm, annot = True, fmt = "g", cmap = "Blues", cbar
= False)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title(f"Confusion Matrix for {act} Activation, {batch}
batch size, {epoch} epochs...")
plt.show()

```

Training model with relu activation function, 64 batch sizes, 5 epochs_list

Epoch 1/5
 844/844 - 6s - loss: 0.2151 - accuracy: 0.9355 - val_loss: 0.0870 -
 val_accuracy: 0.9775 - 6s/epoch - 7ms/step

Epoch 2/5
 844/844 - 5s - loss: 0.0789 - accuracy: 0.9765 - val_loss: 0.0669 -
 val_accuracy: 0.9828 - 5s/epoch - 6ms/step

Epoch 3/5
 844/844 - 5s - loss: 0.0536 - accuracy: 0.9844 - val_loss: 0.0528 -
 val_accuracy: 0.9872 - 5s/epoch - 6ms/step

Epoch 4/5
 844/844 - 5s - loss: 0.0399 - accuracy: 0.9881 - val_loss: 0.0534 -
 val_accuracy: 0.9870 - 5s/epoch - 6ms/step

Epoch 5/5
 844/844 - 5s - loss: 0.0308 - accuracy: 0.9912 - val_loss: 0.0440 -
 val_accuracy: 0.9895 - 5s/epoch - 6ms/step

313/313 [=====] - 1s 2ms/step

Results for relu function, 64 batch size and 5 epochs...

Confusion Matrix

974	0	1	0	0	2	1	1	1	0
0	1126	3	1	0	0	2	1	2	0
1	2	1019	0	2	0	1	7	0	0
0	0	3	993	0	6	0	4	3	1
0	0	1	0	975	0	0	0	1	5
2	0	1	6	0	880	3	0	0	0
7	2	1	0	2	6	939	0	1	0
0	3	6	0	0	0	0	1017	1	1
6	1	2	0	2	3	0	5	953	2
3	3	0	3	6	2	0	15	3	974

Precision: 0.9851
 Recall: 0.9850

Confusion Matrix for relu Activation, 64 batch size, 5 epochs...

	0	1	2	3	4	5	6	7	8	9
0	974	0	1	0	0	2	1	1	1	0
1	0	1126	3	1	0	0	2	1	2	0
2	1	2	1019	0	2	0	1	7	0	0
3	0	0	3	993	0	6	0	4	3	1
4	0	0	1	0	975	0	0	0	1	5
5	2	0	1	6	0	880	3	0	0	0
6	7	2	1	0	2	6	939	0	1	0
7	0	3	6	0	0	0	0	1017	1	1
8	6	1	2	0	2	3	0	5	953	2
9	3	3	0	3	6	2	0	15	3	974
	0	1	2	3	4	5	6	7	8	9
True Labels										
Predicted Labels										

Training model with relu activation function, 64 batch sizes, 15 epochs_list
 Epoch 1/15
 844/844 - 6s - loss: 0.2391 - accuracy: 0.9270 - val_loss: 0.0982 - val_accuracy: 0.9740 - 6s/epoch - 7ms/step
 Epoch 2/15
 844/844 - 5s - loss: 0.0925 - accuracy: 0.9725 - val_loss: 0.0741 - val_accuracy: 0.9792 - 5s/epoch - 6ms/step
 Epoch 3/15
 844/844 - 5s - loss: 0.0600 - accuracy: 0.9822 - val_loss: 0.0588 - val_accuracy: 0.9845 - 5s/epoch - 6ms/step
 Epoch 4/15
 844/844 - 5s - loss: 0.0437 - accuracy: 0.9867 - val_loss: 0.0597 - val_accuracy: 0.9832 - 5s/epoch - 6ms/step
 Epoch 5/15
 844/844 - 5s - loss: 0.0326 - accuracy: 0.9904 - val_loss: 0.0509 - val_accuracy: 0.9863 - 5s/epoch - 6ms/step
 Epoch 6/15

```
844/844 - 5s - loss: 0.0255 - accuracy: 0.9924 - val_loss: 0.0500 -  
val_accuracy: 0.9855 - 5s/epoch - 6ms/step  
Epoch 7/15  
844/844 - 5s - loss: 0.0189 - accuracy: 0.9949 - val_loss: 0.0507 -  
val_accuracy: 0.9878 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.0152 - accuracy: 0.9956 - val_loss: 0.0531 -  
val_accuracy: 0.9860 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.0117 - accuracy: 0.9972 - val_loss: 0.0499 -  
val_accuracy: 0.9875 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.0095 - accuracy: 0.9978 - val_loss: 0.0480 -  
val_accuracy: 0.9877 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.0070 - accuracy: 0.9986 - val_loss: 0.0457 -  
val_accuracy: 0.9895 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.0050 - accuracy: 0.9993 - val_loss: 0.0501 -  
val_accuracy: 0.9885 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.0041 - accuracy: 0.9994 - val_loss: 0.0513 -  
val_accuracy: 0.9880 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.0030 - accuracy: 0.9997 - val_loss: 0.0527 -  
val_accuracy: 0.9882 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.0029 - accuracy: 0.9996 - val_loss: 0.0508 -  
val_accuracy: 0.9893 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Results for relu function, 64 batch size and 15 epochs...  
Confusion Matrix  
[[ 971  0  2  0  0  1  2  1  2  1]  
 [ 0 1127  2  1  0  1  1  1  2  0]  
 [ 2  1 1016  3  2  0  0  4  4  0]  
 [ 0  0  1 997  0  5  0  1  3  3]  
 [ 0  0  2  0 969  0  2  0  0  9]  
 [ 2  0  0  5  0 878  6  0  1  0]  
 [ 3  3  1  1  2  1 944  1  2  0]  
 [ 0  0  7  2  0  0  0 1017  1  1]  
 [ 5  0  2  0  2  2  0  2 958  3]  
 [ 1  2  0  2  8  2  0  5  1 988]]  
Precision: 0.9865  
Recall: 0.9865
```

Confusion Matrix for relu Activation, 64 batch size, 15 epochs...

	0	1	2	3	4	5	6	7	8	9
True Labels	971	0	2	0	0	1	2	1	2	1
0	971	0	2	0	0	1	2	1	2	1
1	0	1127	2	1	0	1	1	1	2	0
2	2	1	1016	3	2	0	0	4	4	0
3	0	0	1	997	0	5	0	1	3	3
4	0	0	2	0	969	0	2	0	0	9
5	2	0	0	5	0	878	6	0	1	0
6	3	3	1	1	2	1	944	1	2	0
7	0	0	7	2	0	0	0	1017	1	1
8	5	0	2	0	2	2	0	2	958	3
9	1	2	0	2	8	2	0	5	1	988
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```
Training model with relu activation function, 64 batch sizes, 20
epochs_list
Epoch 1/20
844/844 - 6s - loss: 0.2236 - accuracy: 0.9313 - val_loss: 0.0811 -
val_accuracy: 0.9785 - 6s/epoch - 7ms/step
Epoch 2/20
844/844 - 6s - loss: 0.0805 - accuracy: 0.9758 - val_loss: 0.0627 -
val_accuracy: 0.9823 - 6s/epoch - 7ms/step
Epoch 3/20
844/844 - 6s - loss: 0.0544 - accuracy: 0.9834 - val_loss: 0.0508 -
val_accuracy: 0.9863 - 6s/epoch - 7ms/step
Epoch 4/20
844/844 - 5s - loss: 0.0395 - accuracy: 0.9886 - val_loss: 0.0526 -
val_accuracy: 0.9858 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: 0.0304 - accuracy: 0.9912 - val_loss: 0.0468 -
val_accuracy: 0.9898 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 0.0237 - accuracy: 0.9934 - val_loss: 0.0469 -
val_accuracy: 0.9883 - 5s/epoch - 6ms/step
Epoch 7/20
```

```

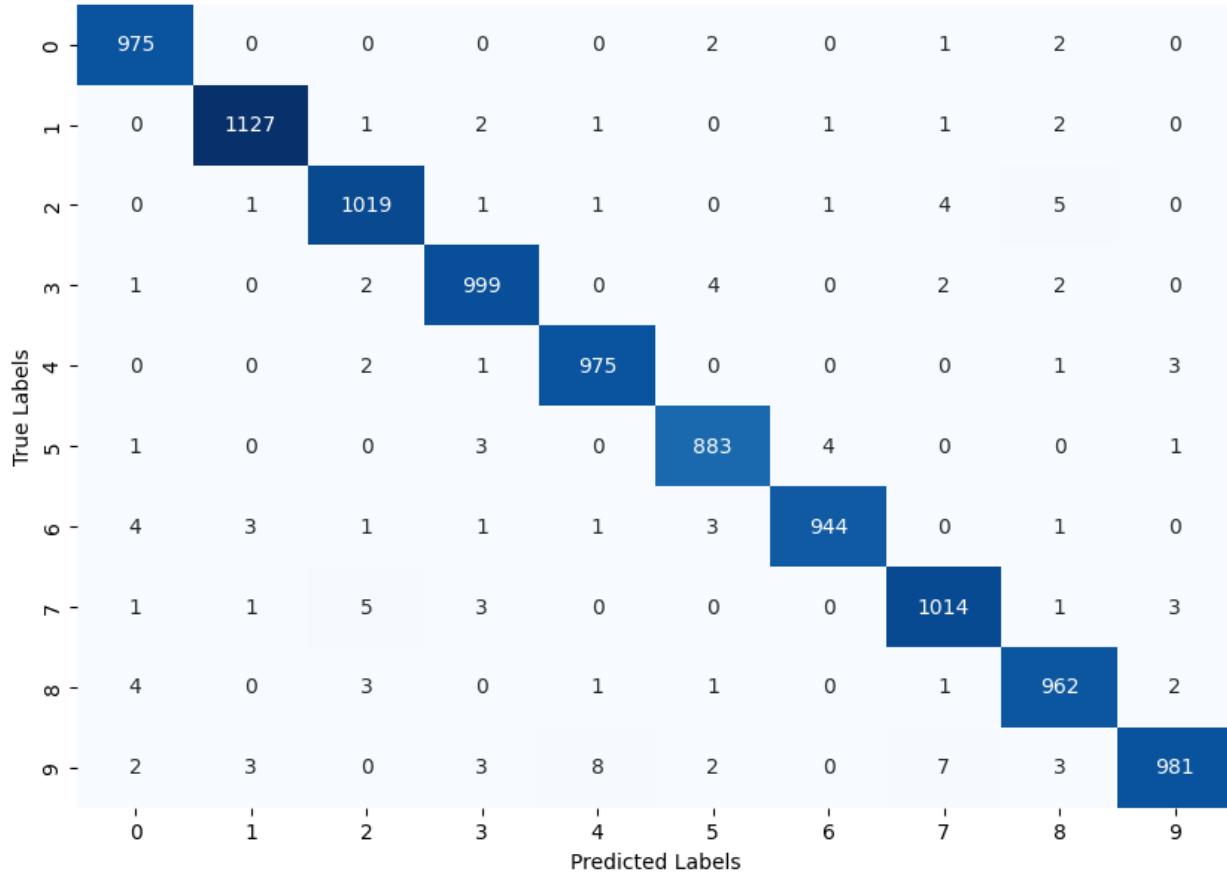
844/844 - 5s - loss: 0.0184 - accuracy: 0.9948 - val_loss: 0.0465 -
val_accuracy: 0.9887 - 5s/epoch - 6ms/step
Epoch 8/20
844/844 - 5s - loss: 0.0141 - accuracy: 0.9963 - val_loss: 0.0424 -
val_accuracy: 0.9897 - 5s/epoch - 6ms/step
Epoch 9/20
844/844 - 5s - loss: 0.0116 - accuracy: 0.9969 - val_loss: 0.0525 -
val_accuracy: 0.9865 - 5s/epoch - 6ms/step
Epoch 10/20
844/844 - 5s - loss: 0.0089 - accuracy: 0.9980 - val_loss: 0.0479 -
val_accuracy: 0.9885 - 5s/epoch - 6ms/step
Epoch 11/20
844/844 - 5s - loss: 0.0070 - accuracy: 0.9988 - val_loss: 0.0476 -
val_accuracy: 0.9885 - 5s/epoch - 6ms/step
Epoch 12/20
844/844 - 5s - loss: 0.0053 - accuracy: 0.9991 - val_loss: 0.0449 -
val_accuracy: 0.9898 - 5s/epoch - 6ms/step
Epoch 13/20
844/844 - 5s - loss: 0.0039 - accuracy: 0.9995 - val_loss: 0.0478 -
val_accuracy: 0.9887 - 5s/epoch - 6ms/step
Epoch 14/20
844/844 - 5s - loss: 0.0031 - accuracy: 0.9996 - val_loss: 0.0482 -
val_accuracy: 0.9900 - 5s/epoch - 6ms/step
Epoch 15/20
844/844 - 5s - loss: 0.0026 - accuracy: 0.9998 - val_loss: 0.0470 -
val_accuracy: 0.9895 - 5s/epoch - 6ms/step
Epoch 16/20
844/844 - 5s - loss: 0.0019 - accuracy: 0.9999 - val_loss: 0.0472 -
val_accuracy: 0.9897 - 5s/epoch - 6ms/step
Epoch 17/20
844/844 - 5s - loss: 0.0017 - accuracy: 0.9999 - val_loss: 0.0480 -
val_accuracy: 0.9900 - 5s/epoch - 6ms/step
Epoch 18/20
844/844 - 5s - loss: 0.0014 - accuracy: 0.9999 - val_loss: 0.0482 -
val_accuracy: 0.9905 - 5s/epoch - 6ms/step
Epoch 19/20
844/844 - 5s - loss: 0.0013 - accuracy: 1.0000 - val_loss: 0.0497 -
val_accuracy: 0.9903 - 5s/epoch - 6ms/step
Epoch 20/20
844/844 - 5s - loss: 0.0011 - accuracy: 1.0000 - val_loss: 0.0498 -
val_accuracy: 0.9897 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for relu function, 64 batch size and 20 epochs...
Confusion Matrix
[[ 975   0   0   0   0   2   0   1   2   0]
 [  0 1127   1   2   1   0   1   1   2   0]
 [  0   1 1019   1   1   0   1   4   5   0]
 [  1   0   2  999   0   4   0   2   2   0]
 [  0   0   2   1  975   0   0   0   1   3]]
```

```
[ 1 0 0 3 0 883 4 0 0 1]
[ 4 3 1 1 1 3 944 0 1 0]
[ 1 1 5 3 0 0 0 1014 1 3]
[ 4 0 3 0 1 1 0 1 962 2]
[ 2 3 0 3 8 2 0 7 3 981]]
```

Precision: 0.9879

Recall: 0.9879

Confusion Matrix for relu Activation, 64 batch size, 20 epochs...



Training model with relu activation function, 128 batch sizes, 5 epochs_list

Epoch 1/5

422/422 - 4s - loss: 0.2679 - accuracy: 0.9200 - val_loss: 0.1184 - val_accuracy: 0.9688 - 4s/epoch - 10ms/step

Epoch 2/5

422/422 - 3s - loss: 0.1058 - accuracy: 0.9695 - val_loss: 0.0774 - val_accuracy: 0.9808 - 3s/epoch - 8ms/step

Epoch 3/5

422/422 - 3s - loss: 0.0722 - accuracy: 0.9788 - val_loss: 0.0668 - val_accuracy: 0.9818 - 3s/epoch - 8ms/step

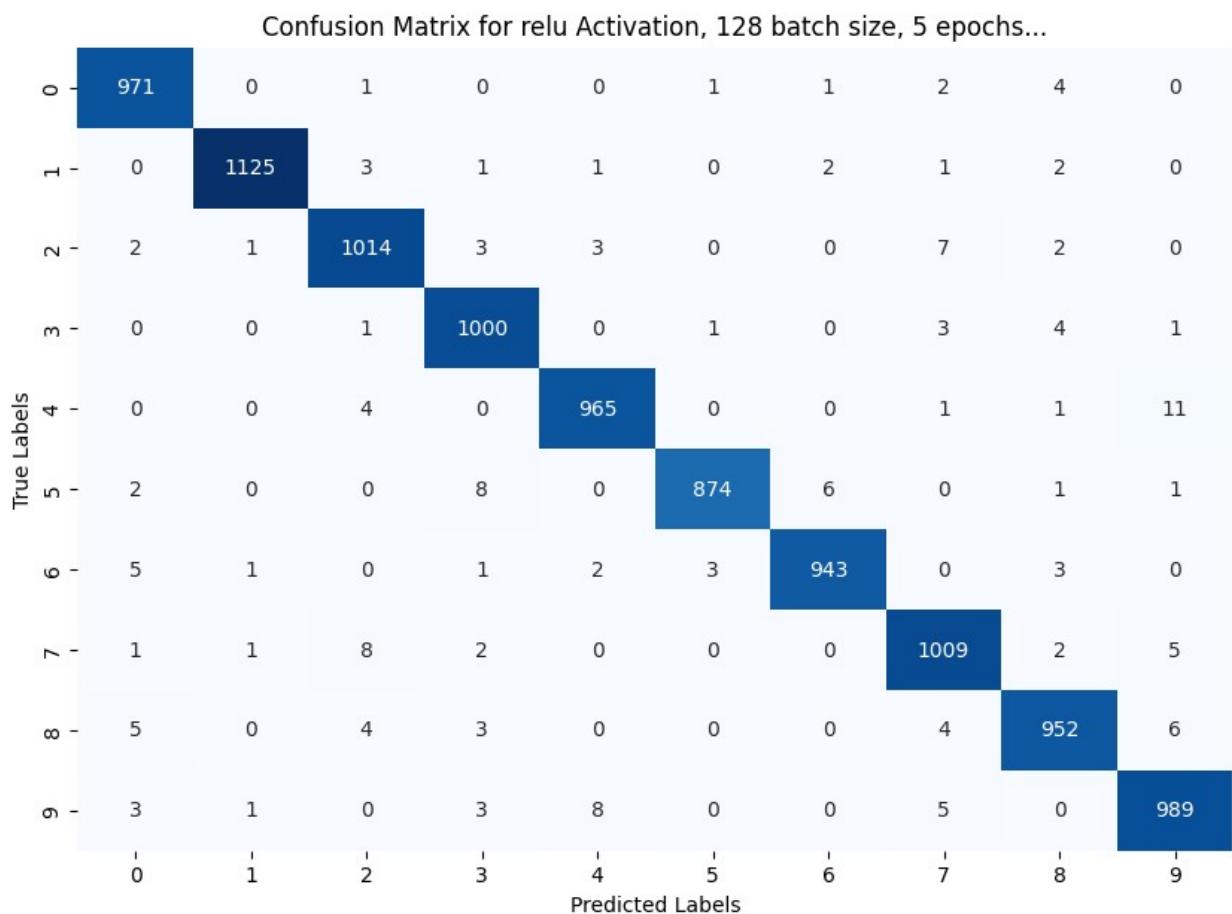
Epoch 4/5

422/422 - 3s - loss: 0.0554 - accuracy: 0.9836 - val_loss: 0.0524 -

```

val_accuracy: 0.9860 - 3s/epoch - 8ms/step
Epoch 5/5
422/422 - 3s - loss: 0.0455 - accuracy: 0.9866 - val_loss: 0.0502 -
val_accuracy: 0.9882 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for relu function, 128 batch size and 5 epochs...
Confusion Matrix
[[ 971   0   1   0   0   1   1   2   4   0]
 [  0 1125   3   1   1   0   2   1   2   0]
 [  2   1 1014   3   3   0   0   7   2   0]
 [  0   0   1 1000   0   1   0   3   4   1]
 [  0   0   4   0 965   0   0   1   1 11]
 [  2   0   0   8   0 874   6   0   1   1]
 [  5   1   0   1   2   3 943   0   3   0]
 [  1   1   8   2   0   0   0 1009   2   5]
 [  5   0   4   3   0   0   0   4 952   6]
 [  3   1   0   3   8   0   0   5   0 989]]
Precision: 0.9842
Recall: 0.9842

```



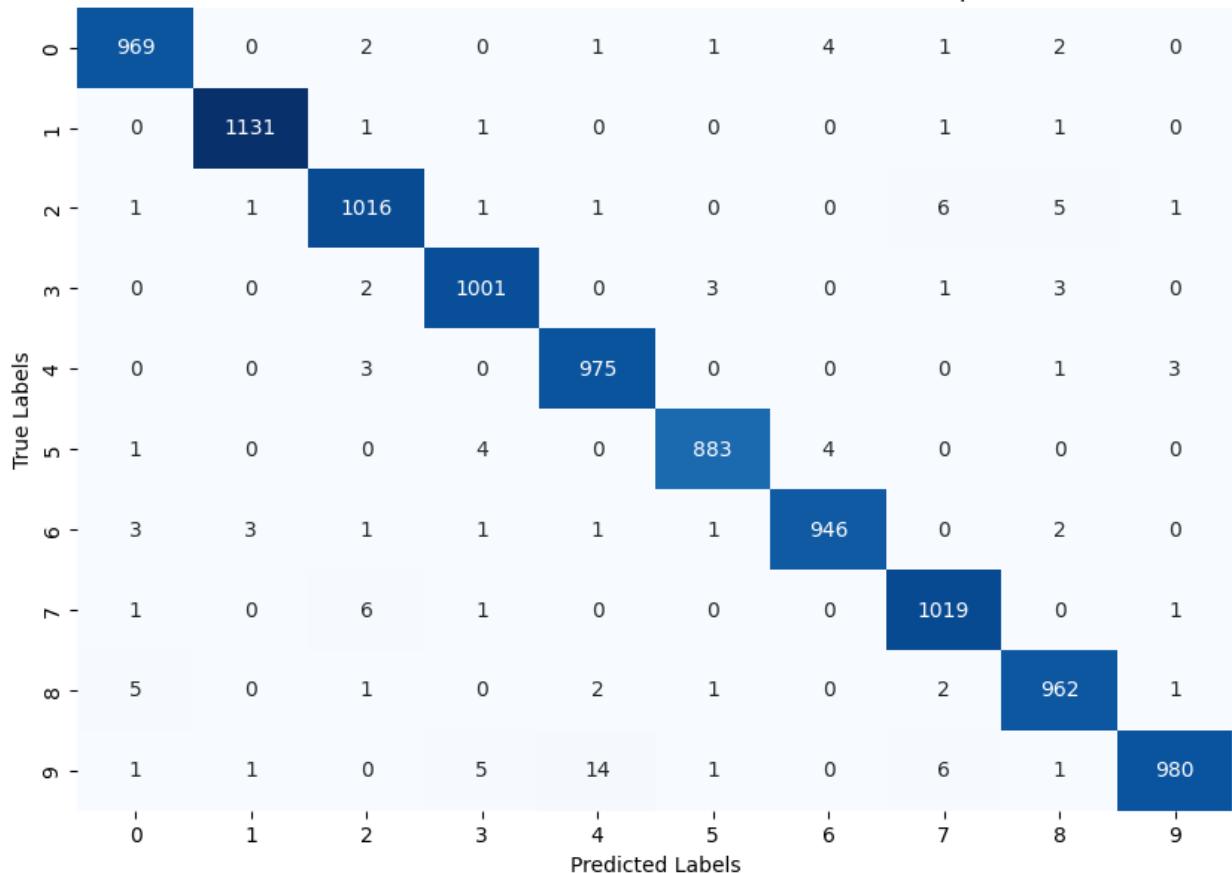
```
Training model with relu activation function, 128 batch sizes, 15 epochs_list
Epoch 1/15
422/422 - 4s - loss: 0.2678 - accuracy: 0.9187 - val_loss: 0.1104 -
val_accuracy: 0.9688 - 4s/epoch - 10ms/step
Epoch 2/15
422/422 - 3s - loss: 0.1060 - accuracy: 0.9698 - val_loss: 0.0766 -
val_accuracy: 0.9807 - 3s/epoch - 8ms/step
Epoch 3/15
422/422 - 3s - loss: 0.0724 - accuracy: 0.9794 - val_loss: 0.0721 -
val_accuracy: 0.9808 - 3s/epoch - 8ms/step
Epoch 4/15
422/422 - 3s - loss: 0.0561 - accuracy: 0.9839 - val_loss: 0.0567 -
val_accuracy: 0.9858 - 3s/epoch - 8ms/step
Epoch 5/15
422/422 - 3s - loss: 0.0441 - accuracy: 0.9878 - val_loss: 0.0541 -
val_accuracy: 0.9862 - 3s/epoch - 8ms/step
Epoch 6/15
422/422 - 3s - loss: 0.0374 - accuracy: 0.9894 - val_loss: 0.0459 -
val_accuracy: 0.9882 - 3s/epoch - 8ms/step
Epoch 7/15
422/422 - 3s - loss: 0.0312 - accuracy: 0.9913 - val_loss: 0.0485 -
val_accuracy: 0.9888 - 3s/epoch - 8ms/step
Epoch 8/15
422/422 - 3s - loss: 0.0262 - accuracy: 0.9926 - val_loss: 0.0468 -
val_accuracy: 0.9873 - 3s/epoch - 8ms/step
Epoch 9/15
422/422 - 3s - loss: 0.0224 - accuracy: 0.9942 - val_loss: 0.0437 -
val_accuracy: 0.9895 - 3s/epoch - 8ms/step
Epoch 10/15
422/422 - 3s - loss: 0.0197 - accuracy: 0.9948 - val_loss: 0.0435 -
val_accuracy: 0.9885 - 3s/epoch - 8ms/step
Epoch 11/15
422/422 - 3s - loss: 0.0168 - accuracy: 0.9956 - val_loss: 0.0479 -
val_accuracy: 0.9887 - 3s/epoch - 8ms/step
Epoch 12/15
422/422 - 3s - loss: 0.0144 - accuracy: 0.9965 - val_loss: 0.0427 -
val_accuracy: 0.9883 - 3s/epoch - 8ms/step
Epoch 13/15
422/422 - 3s - loss: 0.0119 - accuracy: 0.9976 - val_loss: 0.0488 -
val_accuracy: 0.9882 - 3s/epoch - 8ms/step
Epoch 14/15
422/422 - 3s - loss: 0.0103 - accuracy: 0.9979 - val_loss: 0.0437 -
val_accuracy: 0.9885 - 3s/epoch - 8ms/step
Epoch 15/15
422/422 - 3s - loss: 0.0093 - accuracy: 0.9982 - val_loss: 0.0455 -
val_accuracy: 0.9888 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for relu function, 128 batch size and 15 epochs...
Confusion Matrix
```

```
[[ 969  0  2  0  1  1  4  1  2  0]
 [ 0 1131  1  1  0  0  0  1  1  0]
 [ 1  1 1016  1  1  0  0  6  5  1]
 [ 0  0  2 1001  0  3  0  1  3  0]
 [ 0  0  3  0 975  0  0  0  1  3]
 [ 1  0  0  4  0 883  4  0  0  0]
 [ 3  3  1  1  1  1  946  0  2  0]
 [ 1  0  6  1  0  0  0 1019  0  1]
 [ 5  0  1  0  2  1  0  2  962  1]
 [ 1  1  0  5  14  1  0  6  1  980]]
```

Precision: 0.9882

Recall: 0.9882

Confusion Matrix for relu Activation, 128 batch size, 15 epochs...



Training model with relu activation function, 128 batch sizes, 20 epochs list

Epoch 1/20

422/422 - 5s - loss: 0.2641 - accuracy: 0.9202 - val_loss: 0.1091 - val_accuracy: 0.9727 - 5s/epoch - 11ms/step

Epoch 2/20

422/422 - 3s - loss: 0.1055 - accuracy: 0.9694 - val_loss: 0.0800 - val_accuracy: 0.9792 - 3s/epoch - 8ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.0726 - accuracy: 0.9791 - val_loss: 0.0613 -
val_accuracy: 0.9832 - 3s/epoch - 8ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0572 - accuracy: 0.9835 - val_loss: 0.0585 -
val_accuracy: 0.9857 - 3s/epoch - 8ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0454 - accuracy: 0.9866 - val_loss: 0.0546 -
val_accuracy: 0.9868 - 3s/epoch - 8ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0378 - accuracy: 0.9892 - val_loss: 0.0524 -
val_accuracy: 0.9860 - 3s/epoch - 8ms/step
Epoch 7/20
422/422 - 3s - loss: 0.0321 - accuracy: 0.9909 - val_loss: 0.0482 -
val_accuracy: 0.9883 - 3s/epoch - 8ms/step
Epoch 8/20
422/422 - 3s - loss: 0.0276 - accuracy: 0.9923 - val_loss: 0.0493 -
val_accuracy: 0.9883 - 3s/epoch - 8ms/step
Epoch 9/20
422/422 - 3s - loss: 0.0236 - accuracy: 0.9936 - val_loss: 0.0500 -
val_accuracy: 0.9877 - 3s/epoch - 8ms/step
Epoch 10/20
422/422 - 3s - loss: 0.0195 - accuracy: 0.9950 - val_loss: 0.0469 -
val_accuracy: 0.9878 - 3s/epoch - 8ms/step
Epoch 11/20
422/422 - 3s - loss: 0.0174 - accuracy: 0.9956 - val_loss: 0.0475 -
val_accuracy: 0.9878 - 3s/epoch - 8ms/step
Epoch 12/20
422/422 - 3s - loss: 0.0155 - accuracy: 0.9960 - val_loss: 0.0510 -
val_accuracy: 0.9880 - 3s/epoch - 8ms/step
Epoch 13/20
422/422 - 3s - loss: 0.0127 - accuracy: 0.9970 - val_loss: 0.0527 -
val_accuracy: 0.9872 - 3s/epoch - 8ms/step
Epoch 14/20
422/422 - 3s - loss: 0.0112 - accuracy: 0.9976 - val_loss: 0.0490 -
val_accuracy: 0.9883 - 3s/epoch - 8ms/step
Epoch 15/20
422/422 - 3s - loss: 0.0101 - accuracy: 0.9980 - val_loss: 0.0496 -
val_accuracy: 0.9885 - 3s/epoch - 8ms/step
Epoch 16/20
422/422 - 3s - loss: 0.0092 - accuracy: 0.9984 - val_loss: 0.0507 -
val_accuracy: 0.9883 - 3s/epoch - 8ms/step
Epoch 17/20
422/422 - 3s - loss: 0.0076 - accuracy: 0.9987 - val_loss: 0.0504 -
val_accuracy: 0.9883 - 3s/epoch - 8ms/step
Epoch 18/20
422/422 - 3s - loss: 0.0066 - accuracy: 0.9990 - val_loss: 0.0527 -
val_accuracy: 0.9882 - 3s/epoch - 8ms/step
Epoch 19/20
```

422/422 - 3s - loss: 0.0056 - accuracy: 0.9992 - val_loss: 0.0499 -
val_accuracy: 0.9892 - 3s/epoch - 8ms/step

Epoch 20/20

422/422 - 3s - loss: 0.0051 - accuracy: 0.9994 - val_loss: 0.0521 -
val_accuracy: 0.9887 - 3s/epoch - 8ms/step

313/313 [=====] - 1s 2ms/step

Results for relu function, 128 batch size and 20 epochs...

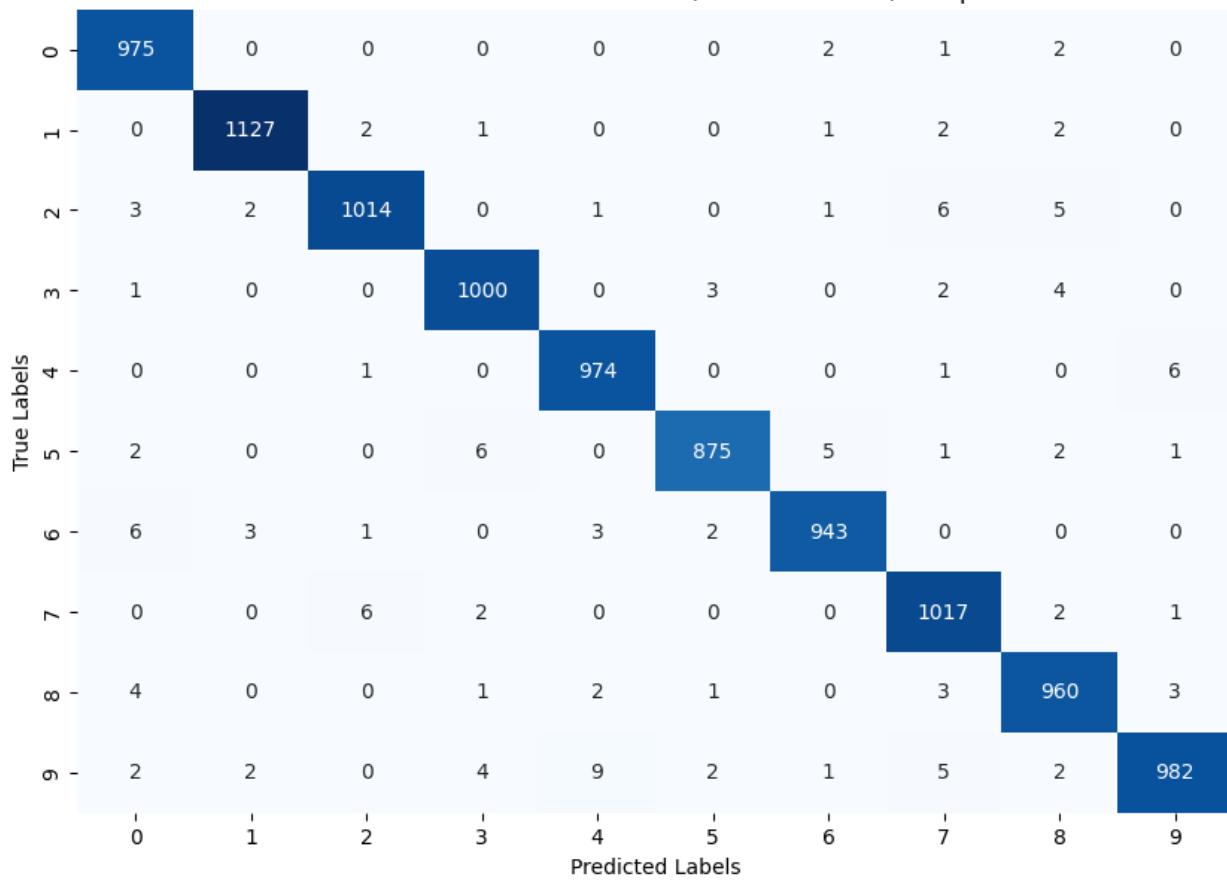
Confusion Matrix

```
[[ 975   0   0   0   0   0   2   1   2   0]
 [ 0 1127   2   1   0   0   1   2   2   0]
 [ 3  2 1014   0   1   0   1   6   5   0]
 [ 1   0   0 1000   0   3   0   2   4   0]
 [ 0   0   1   0 974   0   0   1   0   6]
 [ 2   0   0   6   0 875   5   1   2   1]
 [ 6   3   1   0   3   2 943   0   0   0]
 [ 0   0   6   2   0   0   0 1017   2   1]
 [ 4   0   0   1   2   1   0   3 960   3]
 [ 2   2   0   4   9   2   1   5   2 982]]
```

Precision: 0.9867

Recall: 0.9867

Confusion Matrix for relu Activation, 128 batch size, 20 epochs...



```
Training model with relu activation function, 256 batch sizes, 5 epochs_list
Epoch 1/5
211/211 - 3s - loss: 0.3441 - accuracy: 0.8954 - val_loss: 0.1619 -
val_accuracy: 0.9555 - 3s/epoch - 16ms/step
Epoch 2/5
211/211 - 3s - loss: 0.1550 - accuracy: 0.9534 - val_loss: 0.1057 -
val_accuracy: 0.9703 - 3s/epoch - 13ms/step
Epoch 3/5
211/211 - 3s - loss: 0.1143 - accuracy: 0.9666 - val_loss: 0.0880 -
val_accuracy: 0.9755 - 3s/epoch - 12ms/step
Epoch 4/5
211/211 - 3s - loss: 0.0895 - accuracy: 0.9739 - val_loss: 0.0712 -
val_accuracy: 0.9798 - 3s/epoch - 13ms/step
Epoch 5/5
211/211 - 3s - loss: 0.0757 - accuracy: 0.9781 - val_loss: 0.0713 -
val_accuracy: 0.9792 - 3s/epoch - 13ms/step
313/313 [=====] - 1s 2ms/step
Results for relu function, 256 batch size and 5 epochs...
Confusion Matrix
[[ 971    0    1    1    0    0    2    2    3    0]
 [  0 1126    2    1    0    1    1    1    3    0]
 [  4    2 1006    5    1    0    0    7    7    0]
 [  0    0    1  994    0    1    0    6    6    2]
 [  1    0    4    0  944    0    6    4    2   21]
 [  4    0    0   15    0  852    7    0   11    3]
 [  6    3    1    0    2    3  936    2    5    0]
 [  1    7    6    6    0    0    0  997    5    6]
 [  5    0    1    3    1    1    2    1  960    0]
 [  3    5    1    7    3    1    0    7    3  979]]
```

Precision: 0.9767
Recall: 0.9765

Confusion Matrix for relu Activation, 256 batch size, 5 epochs...

	0	1	2	3	4	5	6	7	8	9	
0	971	0	1	1	0	0	2	2	3	0	
1	0	1126	2	1	0	1	1	1	3	0	
2	4	2	1006	5	1	0	0	7	7	0	
3	0	0	1	994	0	1	0	6	6	2	
4	1	0	4	0	944	0	6	4	2	21	
5	4	0	0	15	0	852	7	0	11	3	
6	6	3	1	0	2	3	936	2	5	0	
7	1	7	6	6	0	0	0	997	5	6	
8	5	0	1	3	1	1	2	1	960	0	
9	3	5	1	7	3	1	0	7	3	979	
	0	1	2	3	4	5	6	7	8	9	
	True Labels										Predicted Labels

```
Training model with relu activation function, 256 batch sizes, 15
epochs_list
Epoch 1/15
211/211 - 3s - loss: 0.3630 - accuracy: 0.8926 - val_loss: 0.1682 -
val_accuracy: 0.9545 - 3s/epoch - 16ms/step
Epoch 2/15
211/211 - 3s - loss: 0.1632 - accuracy: 0.9529 - val_loss: 0.1142 -
val_accuracy: 0.9682 - 3s/epoch - 13ms/step
Epoch 3/15
211/211 - 3s - loss: 0.1179 - accuracy: 0.9660 - val_loss: 0.0892 -
val_accuracy: 0.9767 - 3s/epoch - 13ms/step
Epoch 4/15
211/211 - 3s - loss: 0.0933 - accuracy: 0.9734 - val_loss: 0.0764 -
val_accuracy: 0.9802 - 3s/epoch - 13ms/step
Epoch 5/15
211/211 - 3s - loss: 0.0758 - accuracy: 0.9785 - val_loss: 0.0688 -
val_accuracy: 0.9820 - 3s/epoch - 12ms/step
Epoch 6/15
211/211 - 3s - loss: 0.0651 - accuracy: 0.9816 - val_loss: 0.0659 -
val_accuracy: 0.9847 - 3s/epoch - 12ms/step
Epoch 7/15
```

```
211/211 - 3s - loss: 0.0575 - accuracy: 0.9838 - val_loss: 0.0641 -  
val_accuracy: 0.9848 - 3s/epoch - 13ms/step  
Epoch 8/15  
211/211 - 3s - loss: 0.0501 - accuracy: 0.9857 - val_loss: 0.0579 -  
val_accuracy: 0.9850 - 3s/epoch - 13ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.0436 - accuracy: 0.9876 - val_loss: 0.0569 -  
val_accuracy: 0.9865 - 3s/epoch - 12ms/step  
Epoch 10/15  
211/211 - 3s - loss: 0.0408 - accuracy: 0.9882 - val_loss: 0.0572 -  
val_accuracy: 0.9870 - 3s/epoch - 12ms/step  
Epoch 11/15  
211/211 - 3s - loss: 0.0380 - accuracy: 0.9891 - val_loss: 0.0544 -  
val_accuracy: 0.9863 - 3s/epoch - 12ms/step  
Epoch 12/15  
211/211 - 3s - loss: 0.0337 - accuracy: 0.9907 - val_loss: 0.0572 -  
val_accuracy: 0.9860 - 3s/epoch - 13ms/step  
Epoch 13/15  
211/211 - 3s - loss: 0.0308 - accuracy: 0.9916 - val_loss: 0.0528 -  
val_accuracy: 0.9882 - 3s/epoch - 12ms/step  
Epoch 14/15  
211/211 - 3s - loss: 0.0291 - accuracy: 0.9917 - val_loss: 0.0509 -  
val_accuracy: 0.9867 - 3s/epoch - 12ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.0260 - accuracy: 0.9929 - val_loss: 0.0540 -  
val_accuracy: 0.9862 - 3s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Results for relu function, 256 batch size and 15 epochs...  
Confusion Matrix  
[[ 968  0  2  1  1  1  4  1  1  1]  
[ 0 1129  2  1  0  0  2  1  0  0]  
[ 1  2 1016  3  2  0  2  5  1  0]  
[ 0  0  2 1000  0  2  0  3  2  1]  
[ 0  0  1  0  965  0  3  1  0  12]  
[ 1  0  0  9  0  875  5  0  0  2]  
[ 4  3  1  0  1  3  946  0  0  0]  
[ 0  2  7  2  1  0  0 1010  1  5]  
[ 5  0  6 10  2  3  1  3  934  10]  
[ 2  3  0  6  7  1  0  5  0  985]]  
Precision: 0.9829  
Recall: 0.9828
```

Confusion Matrix for relu Activation, 256 batch size, 15 epochs...

	0	1	2	3	4	5	6	7	8	9
0	968	0	2	1	1	1	4	1	1	1
1	0	1129	2	1	0	0	2	1	0	0
2	1	2	1016	3	2	0	2	5	1	0
3	0	0	2	1000	0	2	0	3	2	1
4	0	0	1	0	965	0	3	1	0	12
5	1	0	0	9	0	875	5	0	0	2
6	4	3	1	0	1	3	946	0	0	0
7	0	2	7	2	1	0	0	1010	1	5
8	5	0	6	10	2	3	1	3	934	10
9	2	3	0	6	7	1	0	5	0	985
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training model with relu activation function, 256 batch sizes, 20
epochs_list
Epoch 1/20
211/211 - 3s - loss: 0.3374 - accuracy: 0.8967 - val_loss: 0.1420 -
val_accuracy: 0.9622 - 3s/epoch - 16ms/step
Epoch 2/20
211/211 - 3s - loss: 0.1488 - accuracy: 0.9565 - val_loss: 0.1128 -
val_accuracy: 0.9685 - 3s/epoch - 13ms/step
Epoch 3/20
211/211 - 3s - loss: 0.1080 - accuracy: 0.9685 - val_loss: 0.0840 -
val_accuracy: 0.9757 - 3s/epoch - 13ms/step
Epoch 4/20
211/211 - 3s - loss: 0.0853 - accuracy: 0.9747 - val_loss: 0.0681 -
val_accuracy: 0.9832 - 3s/epoch - 13ms/step
Epoch 5/20
211/211 - 3s - loss: 0.0702 - accuracy: 0.9798 - val_loss: 0.0615 -
val_accuracy: 0.9853 - 3s/epoch - 14ms/step
Epoch 6/20
211/211 - 3s - loss: 0.0615 - accuracy: 0.9818 - val_loss: 0.0697 -
val_accuracy: 0.9823 - 3s/epoch - 13ms/step
Epoch 7/20
```

```

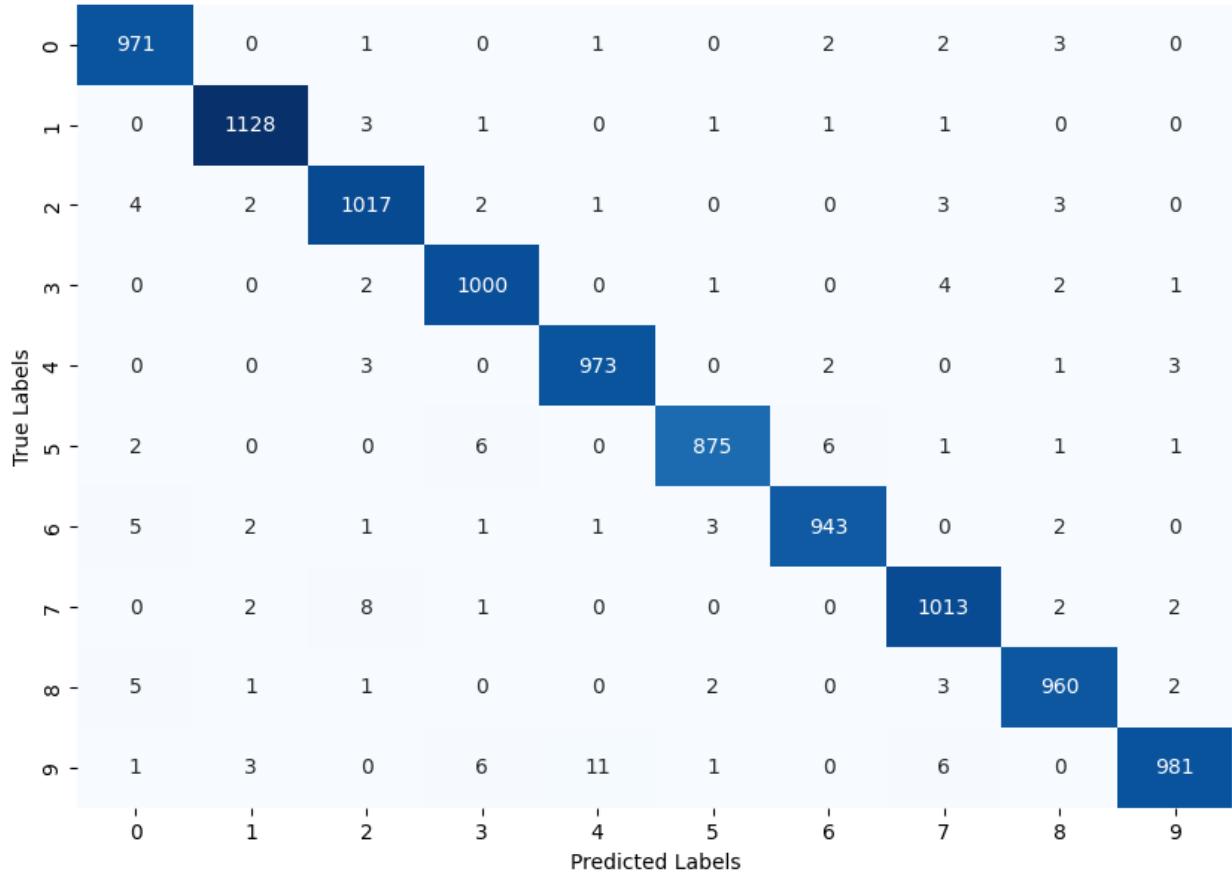
211/211 - 3s - loss: 0.0541 - accuracy: 0.9841 - val_loss: 0.0590 -
val_accuracy: 0.9865 - 3s/epoch - 13ms/step
Epoch 8/20
211/211 - 3s - loss: 0.0462 - accuracy: 0.9868 - val_loss: 0.0600 -
val_accuracy: 0.9852 - 3s/epoch - 13ms/step
Epoch 9/20
211/211 - 3s - loss: 0.0409 - accuracy: 0.9884 - val_loss: 0.0564 -
val_accuracy: 0.9843 - 3s/epoch - 13ms/step
Epoch 10/20
211/211 - 3s - loss: 0.0368 - accuracy: 0.9896 - val_loss: 0.0510 -
val_accuracy: 0.9860 - 3s/epoch - 13ms/step
Epoch 11/20
211/211 - 3s - loss: 0.0342 - accuracy: 0.9903 - val_loss: 0.0491 -
val_accuracy: 0.9872 - 3s/epoch - 13ms/step
Epoch 12/20
211/211 - 3s - loss: 0.0302 - accuracy: 0.9914 - val_loss: 0.0474 -
val_accuracy: 0.9885 - 3s/epoch - 13ms/step
Epoch 13/20
211/211 - 3s - loss: 0.0281 - accuracy: 0.9923 - val_loss: 0.0465 -
val_accuracy: 0.9867 - 3s/epoch - 13ms/step
Epoch 14/20
211/211 - 3s - loss: 0.0258 - accuracy: 0.9929 - val_loss: 0.0494 -
val_accuracy: 0.9875 - 3s/epoch - 13ms/step
Epoch 15/20
211/211 - 3s - loss: 0.0233 - accuracy: 0.9938 - val_loss: 0.0475 -
val_accuracy: 0.9875 - 3s/epoch - 13ms/step
Epoch 16/20
211/211 - 3s - loss: 0.0216 - accuracy: 0.9943 - val_loss: 0.0520 -
val_accuracy: 0.9857 - 3s/epoch - 13ms/step
Epoch 17/20
211/211 - 3s - loss: 0.0200 - accuracy: 0.9947 - val_loss: 0.0493 -
val_accuracy: 0.9865 - 3s/epoch - 13ms/step
Epoch 18/20
211/211 - 3s - loss: 0.0185 - accuracy: 0.9951 - val_loss: 0.0452 -
val_accuracy: 0.9877 - 3s/epoch - 13ms/step
Epoch 19/20
211/211 - 3s - loss: 0.0168 - accuracy: 0.9957 - val_loss: 0.0484 -
val_accuracy: 0.9868 - 3s/epoch - 13ms/step
Epoch 20/20
211/211 - 3s - loss: 0.0150 - accuracy: 0.9965 - val_loss: 0.0477 -
val_accuracy: 0.9877 - 3s/epoch - 13ms/step
313/313 [=====] - 1s 2ms/step
Results for relu function, 256 batch size and 20 epochs...
Confusion Matrix
[[ 971   0   1   0   1   0   2   2   3   0]
 [  0 1128   3   1   0   1   1   1   0   0]
 [  4   2 1017   2   1   0   0   3   3   0]
 [  0   0   2 1000   0   1   0   4   2   1]
 [  0   0   3   0  973   0   2   0   1   3]]
```

```
[ 2 0 0 6 0 875 6 1 1 1]
[ 5 2 1 1 1 3 943 0 2 0]
[ 0 2 8 1 0 0 0 1013 2 2]
[ 5 1 1 0 0 2 0 3 960 2]
[ 1 3 0 6 11 1 0 6 0 981]]
```

Precision: 0.9861

Recall: 0.9861

Confusion Matrix for relu Activation, 256 batch size, 20 epochs...



Training model with relu activation function, 16 batch sizes, 5 epochs_list

Epoch 1/5

3375/3375 - 15s - loss: 0.1460 - accuracy: 0.9542 - val_loss: 0.0668 - val_accuracy: 0.9813 - 15s/epoch - 5ms/step

Epoch 2/5

3375/3375 - 15s - loss: 0.0524 - accuracy: 0.9834 - val_loss: 0.0551 - val_accuracy: 0.9860 - 15s/epoch - 5ms/step

Epoch 3/5

3375/3375 - 15s - loss: 0.0313 - accuracy: 0.9902 - val_loss: 0.0513 - val_accuracy: 0.9858 - 15s/epoch - 4ms/step

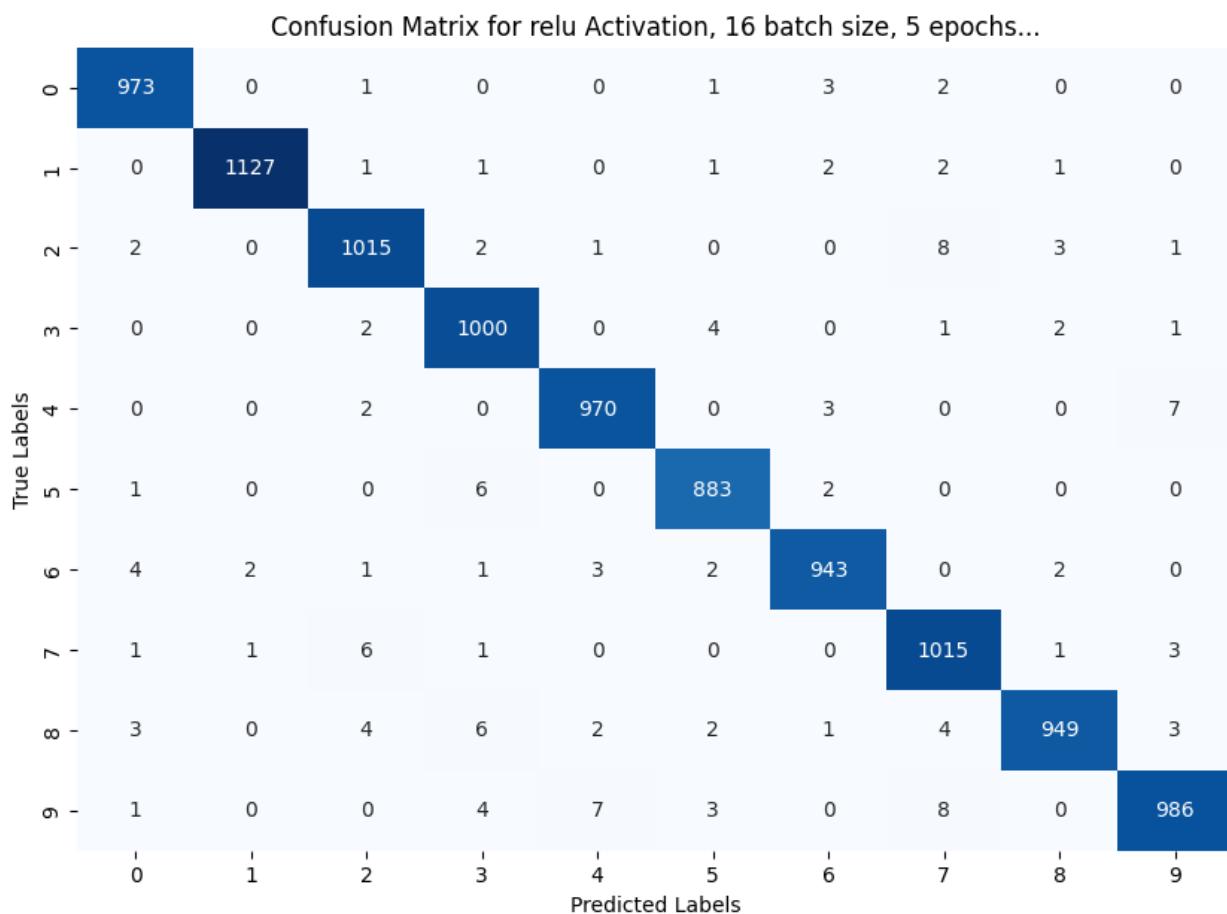
Epoch 4/5

3375/3375 - 15s - loss: 0.0187 - accuracy: 0.9944 - val_loss: 0.0495 -

```

val_accuracy: 0.9868 - 15s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 15s - loss: 0.0124 - accuracy: 0.9961 - val_loss: 0.0513 -
val_accuracy: 0.9882 - 15s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Results for relu function, 16 batch size and 5 epochs...
Confusion Matrix
[[ 973   0   1   0   0   1   3   2   0   0]
 [ 0 1127   1   1   0   1   2   2   1   0]
 [ 2   0 1015   2   1   0   0   8   3   1]
 [ 0   0   2 1000   0   4   0   1   2   1]
 [ 0   0   2   0 970   0   3   0   0   7]
 [ 1   0   0   6   0 883   2   0   0   0]
 [ 4   2   1   1   3   2 943   0   2   0]
 [ 1   1   6   1   0   0   0 1015   1   3]
 [ 3   0   4   6   2   2   1   4 949   3]
 [ 1   0   0   4   7   3   0   8   0 986]]
Precision: 0.9861
Recall: 0.9861

```



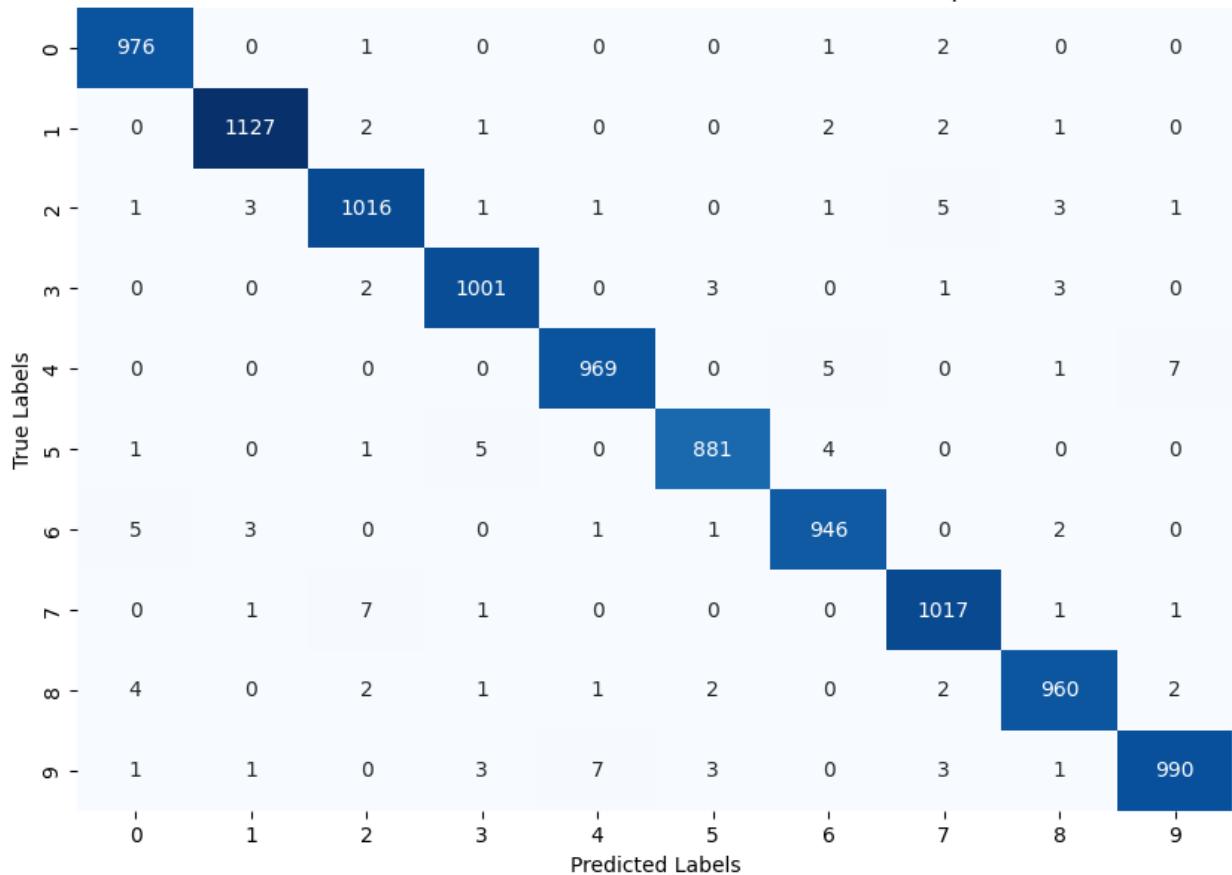
```
Training model with relu activation function, 16 batch sizes, 15 epochs_list
Epoch 1/15
3375/3375 - 15s - loss: 0.1591 - accuracy: 0.9509 - val_loss: 0.0770 -
val_accuracy: 0.9787 - 15s/epoch - 5ms/step
Epoch 2/15
3375/3375 - 15s - loss: 0.0512 - accuracy: 0.9846 - val_loss: 0.0505 -
val_accuracy: 0.9867 - 15s/epoch - 4ms/step
Epoch 3/15
3375/3375 - 15s - loss: 0.0300 - accuracy: 0.9908 - val_loss: 0.0514 -
val_accuracy: 0.9865 - 15s/epoch - 4ms/step
Epoch 4/15
3375/3375 - 15s - loss: 0.0207 - accuracy: 0.9933 - val_loss: 0.0498 -
val_accuracy: 0.9885 - 15s/epoch - 4ms/step
Epoch 5/15
3375/3375 - 15s - loss: 0.0122 - accuracy: 0.9963 - val_loss: 0.0471 -
val_accuracy: 0.9898 - 15s/epoch - 4ms/step
Epoch 6/15
3375/3375 - 15s - loss: 0.0080 - accuracy: 0.9977 - val_loss: 0.0482 -
val_accuracy: 0.9875 - 15s/epoch - 4ms/step
Epoch 7/15
3375/3375 - 15s - loss: 0.0052 - accuracy: 0.9986 - val_loss: 0.0535 -
val_accuracy: 0.9878 - 15s/epoch - 4ms/step
Epoch 8/15
3375/3375 - 16s - loss: 0.0021 - accuracy: 0.9997 - val_loss: 0.0544 -
val_accuracy: 0.9892 - 16s/epoch - 5ms/step
Epoch 9/15
3375/3375 - 15s - loss: 0.0011 - accuracy: 0.9999 - val_loss: 0.0571 -
val_accuracy: 0.9885 - 15s/epoch - 4ms/step
Epoch 10/15
3375/3375 - 15s - loss: 7.1547e-04 - accuracy: 0.9999 - val_loss:
0.0577 - val_accuracy: 0.9892 - 15s/epoch - 5ms/step
Epoch 11/15
3375/3375 - 14s - loss: 3.1510e-04 - accuracy: 1.0000 - val_loss:
0.0596 - val_accuracy: 0.9892 - 14s/epoch - 4ms/step
Epoch 12/15
3375/3375 - 15s - loss: 2.4116e-04 - accuracy: 1.0000 - val_loss:
0.0612 - val_accuracy: 0.9893 - 15s/epoch - 4ms/step
Epoch 13/15
3375/3375 - 15s - loss: 1.9874e-04 - accuracy: 1.0000 - val_loss:
0.0620 - val_accuracy: 0.9897 - 15s/epoch - 4ms/step
Epoch 14/15
3375/3375 - 15s - loss: 1.7230e-04 - accuracy: 1.0000 - val_loss:
0.0636 - val_accuracy: 0.9893 - 15s/epoch - 4ms/step
Epoch 15/15
3375/3375 - 14s - loss: 1.5422e-04 - accuracy: 1.0000 - val_loss:
0.0636 - val_accuracy: 0.9892 - 14s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Results for relu function, 16 batch size and 15 epochs...
Confusion Matrix
```

```
[[ 976  0   1   0   0   0   1   2   0   0]
 [ 0 1127  2   1   0   0   2   2   1   0]
 [ 1   3 1016  1   1   0   1   5   3   1]
 [ 0   0 2 1001  0   3   0   1   3   0]
 [ 0   0   0   0 969  0   5   0   1   7]
 [ 1   0   1   5   0 881  4   0   0   0]
 [ 5   3   0   0   1   1 946  0   2   0]
 [ 0   1   7   1   0   0   0 1017  1   1]
 [ 4   0   2   1   1   2   0   2 960  2]
 [ 1   1   0   3   7   3   0   3   1 990]]
```

Precision: 0.9883

Recall: 0.9883

Confusion Matrix for relu Activation, 16 batch size, 15 epochs...



Training model with relu activation function, 16 batch sizes, 20 epochs list

Epoch 1/20

3375/3375 - 15s - loss: 0.1475 - accuracy: 0.9551 - val_loss: 0.0625 - val_accuracy: 0.9830 - 15s/epoch - 4ms/step

Epoch 2/20

3375/3375 - 14s - loss: 0.0505 - accuracy: 0.9839 - val_loss: 0.0554 - val_accuracy: 0.9858 - 14s/epoch - 4ms/step

```
Epoch 3/20
3375/3375 - 15s - loss: 0.0296 - accuracy: 0.9905 - val_loss: 0.0471 -
val_accuracy: 0.9878 - 15s/epoch - 4ms/step
Epoch 4/20
3375/3375 - 15s - loss: 0.0186 - accuracy: 0.9939 - val_loss: 0.0469 -
val_accuracy: 0.9877 - 15s/epoch - 4ms/step
Epoch 5/20
3375/3375 - 15s - loss: 0.0110 - accuracy: 0.9967 - val_loss: 0.0538 -
val_accuracy: 0.9875 - 15s/epoch - 4ms/step
Epoch 6/20
3375/3375 - 15s - loss: 0.0067 - accuracy: 0.9981 - val_loss: 0.0526 -
val_accuracy: 0.9890 - 15s/epoch - 4ms/step
Epoch 7/20
3375/3375 - 14s - loss: 0.0041 - accuracy: 0.9989 - val_loss: 0.0547 -
val_accuracy: 0.9895 - 14s/epoch - 4ms/step
Epoch 8/20
3375/3375 - 15s - loss: 0.0022 - accuracy: 0.9995 - val_loss: 0.0543 -
val_accuracy: 0.9890 - 15s/epoch - 5ms/step
Epoch 9/20
3375/3375 - 15s - loss: 0.0011 - accuracy: 0.9999 - val_loss: 0.0531 -
val_accuracy: 0.9905 - 15s/epoch - 4ms/step
Epoch 10/20
3375/3375 - 14s - loss: 4.9402e-04 - accuracy: 1.0000 - val_loss:
0.0548 - val_accuracy: 0.9903 - 14s/epoch - 4ms/step
Epoch 11/20
3375/3375 - 15s - loss: 2.9745e-04 - accuracy: 1.0000 - val_loss:
0.0558 - val_accuracy: 0.9902 - 15s/epoch - 4ms/step
Epoch 12/20
3375/3375 - 15s - loss: 2.4681e-04 - accuracy: 1.0000 - val_loss:
0.0580 - val_accuracy: 0.9900 - 15s/epoch - 4ms/step
Epoch 13/20
3375/3375 - 15s - loss: 2.1885e-04 - accuracy: 1.0000 - val_loss:
0.0575 - val_accuracy: 0.9905 - 15s/epoch - 4ms/step
Epoch 14/20
3375/3375 - 15s - loss: 1.6675e-04 - accuracy: 1.0000 - val_loss:
0.0576 - val_accuracy: 0.9905 - 15s/epoch - 4ms/step
Epoch 15/20
3375/3375 - 15s - loss: 1.5352e-04 - accuracy: 1.0000 - val_loss:
0.0589 - val_accuracy: 0.9903 - 15s/epoch - 4ms/step
Epoch 16/20
3375/3375 - 15s - loss: 1.3586e-04 - accuracy: 1.0000 - val_loss:
0.0596 - val_accuracy: 0.9902 - 15s/epoch - 4ms/step
Epoch 17/20
3375/3375 - 15s - loss: 1.2451e-04 - accuracy: 1.0000 - val_loss:
0.0598 - val_accuracy: 0.9903 - 15s/epoch - 4ms/step
Epoch 18/20
3375/3375 - 14s - loss: 1.1506e-04 - accuracy: 1.0000 - val_loss:
0.0597 - val_accuracy: 0.9903 - 14s/epoch - 4ms/step
Epoch 19/20
```

```
3375/3375 - 15s - loss: 1.0648e-04 - accuracy: 1.0000 - val_loss:  
0.0604 - val_accuracy: 0.9902 - 15s/epoch - 4ms/step
```

Epoch 20/20

```
3375/3375 - 15s - loss: 9.8810e-05 - accuracy: 1.0000 - val_loss:  
0.0616 - val_accuracy: 0.9902 - 15s/epoch - 4ms/step
```

313/313 [=====] - 1s 2ms/step

Results for relu function, 16 batch size and 20 epochs...

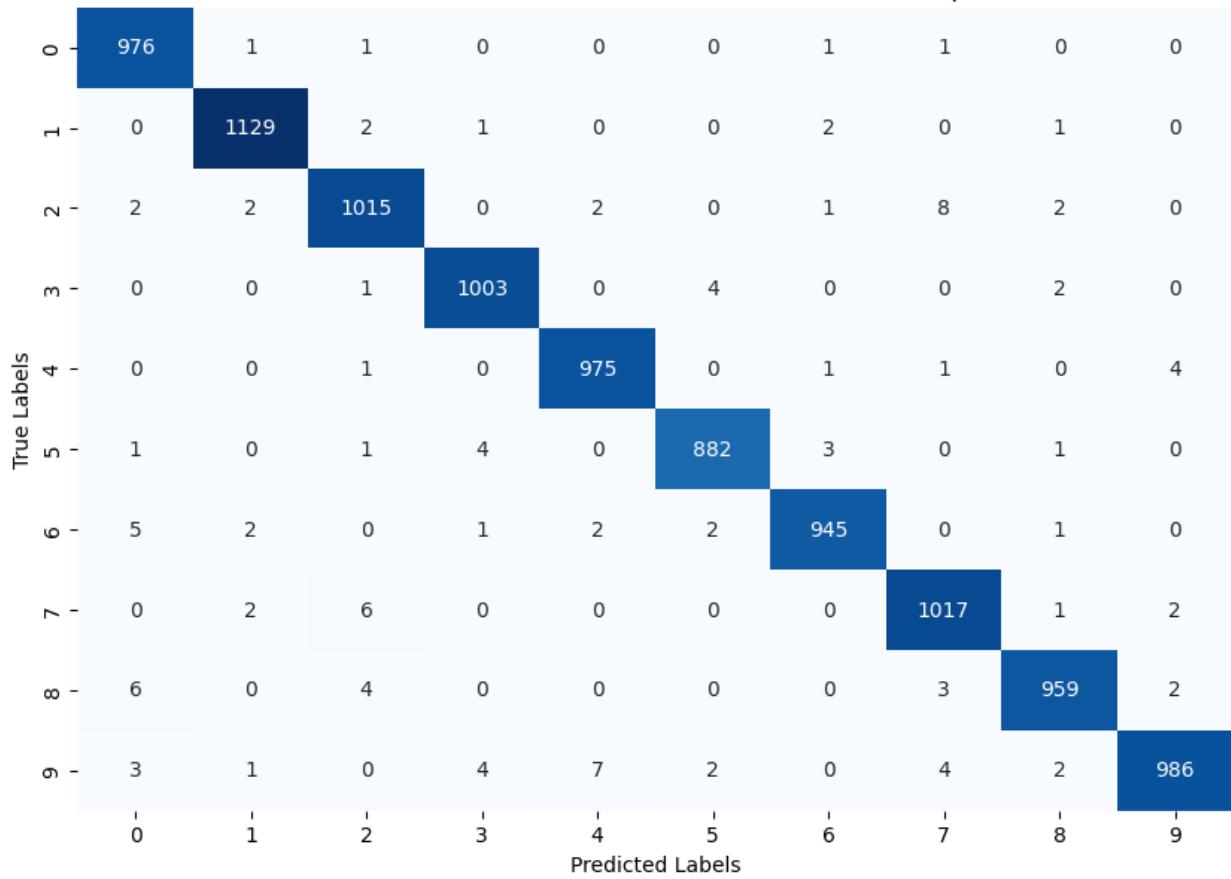
Confusion Matrix

```
[[ 976   1   1   0   0   0   1   1   0   0 ]  
[  0 1129   2   1   0   0   2   0   1   0 ]  
[  2   2 1015   0   2   0   1   8   2   0 ]  
[  0   0   1 1003   0   4   0   0   2   0 ]  
[  0   0   1   0 975   0   1   1   0   4 ]  
[  1   0   1   4   0 882   3   0   1   0 ]  
[  5   2   0   1   2   2 945   0   1   0 ]  
[  0   2   6   0   0   0   0 1017   1   2 ]  
[  6   0   4   0   0   0   0   3 959   2 ]  
[  3   1   0   4   7   2   0   4   2 986 ]]
```

Precision: 0.9887

Recall: 0.9887

Confusion Matrix for relu Activation, 16 batch size, 20 epochs...



```
Training model with sigmoid activation function, 64 batch sizes, 5 epochs_list
Epoch 1/5
844/844 - 6s - loss: 0.9081 - accuracy: 0.7415 - val_loss: 0.3190 -
val_accuracy: 0.9118 - 6s/epoch - 7ms/step
Epoch 2/5
844/844 - 6s - loss: 0.3418 - accuracy: 0.8997 - val_loss: 0.2565 -
val_accuracy: 0.9218 - 6s/epoch - 7ms/step
Epoch 3/5
844/844 - 6s - loss: 0.2753 - accuracy: 0.9169 - val_loss: 0.2048 -
val_accuracy: 0.9423 - 6s/epoch - 7ms/step
Epoch 4/5
844/844 - 5s - loss: 0.2332 - accuracy: 0.9302 - val_loss: 0.2308 -
val_accuracy: 0.9315 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.2056 - accuracy: 0.9379 - val_loss: 0.1622 -
val_accuracy: 0.9558 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for sigmoid function, 64 batch size and 5 epochs...
Confusion Matrix
[[ 960     0     0     4     0     4     4     1     7     0]
 [  0 1100     3     6     1     3     4     1    17     0]
 [  8     0 930     37    12     1     8    10    24     2]
 [  0     0     3 986     0     4     0     8     9     0]
 [  1     0     2     3 940     0     8     2     7    19]
 [  6     0     0   36     3 821     5     1    18     2]
 [  7     3     2     2   10    16 912     0     6     0]
 [  0     2    14    28     7     0     0 955     0    22]
 [  5     0     0   30     5     7     4     3 918     2]
 [  9     2     1   21    24     5     1    11    10 925]]
```

Precision: 0.9466
Recall: 0.9447

Confusion Matrix for sigmoid Activation, 64 batch size, 5 epochs...

	0	1	2	3	4	5	6	7	8	9	
True Labels	960	0	0	4	0	4	4	1	7	0	
0	960	0	0	4	0	4	4	1	7	0	
1	0	1100	3	6	1	3	4	1	17	0	
2	8	0	930	37	12	1	8	10	24	2	
3	0	0	3	986	0	4	0	8	9	0	
4	1	0	2	3	940	0	8	2	7	19	
5	6	0	0	36	3	821	5	1	18	2	
6	7	3	2	2	10	16	912	0	6	0	
7	0	2	14	28	7	0	0	955	0	22	
8	5	0	0	30	5	7	4	3	918	2	
9	9	2	1	21	24	5	1	11	10	925	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```
Training model with sigmoid activation function, 64 batch sizes, 15
epochs_list
Epoch 1/15
844/844 - 6s - loss: 0.8945 - accuracy: 0.7457 - val_loss: 0.3333 -
val_accuracy: 0.9025 - 6s/epoch - 7ms/step
Epoch 2/15
844/844 - 5s - loss: 0.3452 - accuracy: 0.8977 - val_loss: 0.2377 -
val_accuracy: 0.9313 - 5s/epoch - 6ms/step
Epoch 3/15
844/844 - 5s - loss: 0.2806 - accuracy: 0.9165 - val_loss: 0.1951 -
val_accuracy: 0.9452 - 5s/epoch - 6ms/step
Epoch 4/15
844/844 - 5s - loss: 0.2420 - accuracy: 0.9269 - val_loss: 0.1869 -
val_accuracy: 0.9455 - 5s/epoch - 6ms/step
Epoch 5/15
844/844 - 5s - loss: 0.2115 - accuracy: 0.9369 - val_loss: 0.1554 -
val_accuracy: 0.9550 - 5s/epoch - 6ms/step
Epoch 6/15
844/844 - 5s - loss: 0.1880 - accuracy: 0.9434 - val_loss: 0.1369 -
val_accuracy: 0.9625 - 5s/epoch - 6ms/step
Epoch 7/15
```

```
844/844 - 5s - loss: 0.1697 - accuracy: 0.9496 - val_loss: 0.1360 -  
val_accuracy: 0.9660 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.1523 - accuracy: 0.9552 - val_loss: 0.1184 -  
val_accuracy: 0.9680 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.1408 - accuracy: 0.9585 - val_loss: 0.1078 -  
val_accuracy: 0.9705 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.1290 - accuracy: 0.9622 - val_loss: 0.1033 -  
val_accuracy: 0.9743 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.1185 - accuracy: 0.9653 - val_loss: 0.0975 -  
val_accuracy: 0.9720 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.1109 - accuracy: 0.9672 - val_loss: 0.0913 -  
val_accuracy: 0.9750 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.1025 - accuracy: 0.9703 - val_loss: 0.0885 -  
val_accuracy: 0.9762 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.0952 - accuracy: 0.9726 - val_loss: 0.0872 -  
val_accuracy: 0.9762 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.0897 - accuracy: 0.9743 - val_loss: 0.0791 -  
val_accuracy: 0.9773 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Results for sigmoid function, 64 batch size and 15 epochs...  
Confusion Matrix  
[[ 969  0  0  0  0  2  5  1  2  1]  
[ 0 1124  2  1  0  1  4  1  2  0]  
[ 7  2 1000  2  4  0  6  6  5  0]  
[ 0  0  2 989  0  1  0  7  7  4]  
[ 2  0  0  0 956  0  6  8  2  8]  
[ 4  1  0  8  1 863  7  1  6  1]  
[ 7  2  0  0  2  5 938  0  4  0]  
[ 0  7  10  4  0  1  0 1004  0  2]  
[ 7  1  2  4  2  2  3  6 946  1]  
[ 7  5  1 11  15  4  0 14  3 949]]  
Precision: 0.9739  
Recall: 0.9738
```

Confusion Matrix for sigmoid Activation, 64 batch size, 15 epochs...

	0	1	2	3	4	5	6	7	8	9	
True Labels	969	0	0	0	0	2	5	1	2	1	Predicted Labels
0	969	0	0	0	0	2	5	1	2	1	
1	0	1124	2	1	0	1	4	1	2	0	
2	7	2	1000	2	4	0	6	6	5	0	
3	0	0	2	989	0	1	0	7	7	4	
4	2	0	0	0	956	0	6	8	2	8	
5	4	1	0	8	1	863	7	1	6	1	
6	7	2	0	0	2	5	938	0	4	0	
7	0	7	10	4	0	1	0	1004	0	2	
8	7	1	2	4	2	2	3	6	946	1	
9	7	5	1	11	15	4	0	14	3	949	
0	1	2	3	4	5	6	7	8	9	9	

```
Training model with sigmoid activation function, 64 batch sizes, 20
epochs_list
Epoch 1/20
844/844 - 6s - loss: 0.8647 - accuracy: 0.7485 - val_loss: 0.3167 -
val_accuracy: 0.9108 - 6s/epoch - 7ms/step
Epoch 2/20
844/844 - 5s - loss: 0.3478 - accuracy: 0.8971 - val_loss: 0.2406 -
val_accuracy: 0.9318 - 5s/epoch - 6ms/step
Epoch 3/20
844/844 - 5s - loss: 0.2880 - accuracy: 0.9135 - val_loss: 0.2331 -
val_accuracy: 0.9317 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: 0.2503 - accuracy: 0.9247 - val_loss: 0.1942 -
val_accuracy: 0.9413 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: 0.2198 - accuracy: 0.9334 - val_loss: 0.1747 -
val_accuracy: 0.9500 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 0.1973 - accuracy: 0.9399 - val_loss: 0.1616 -
val_accuracy: 0.9545 - 5s/epoch - 6ms/step
Epoch 7/20
```

```

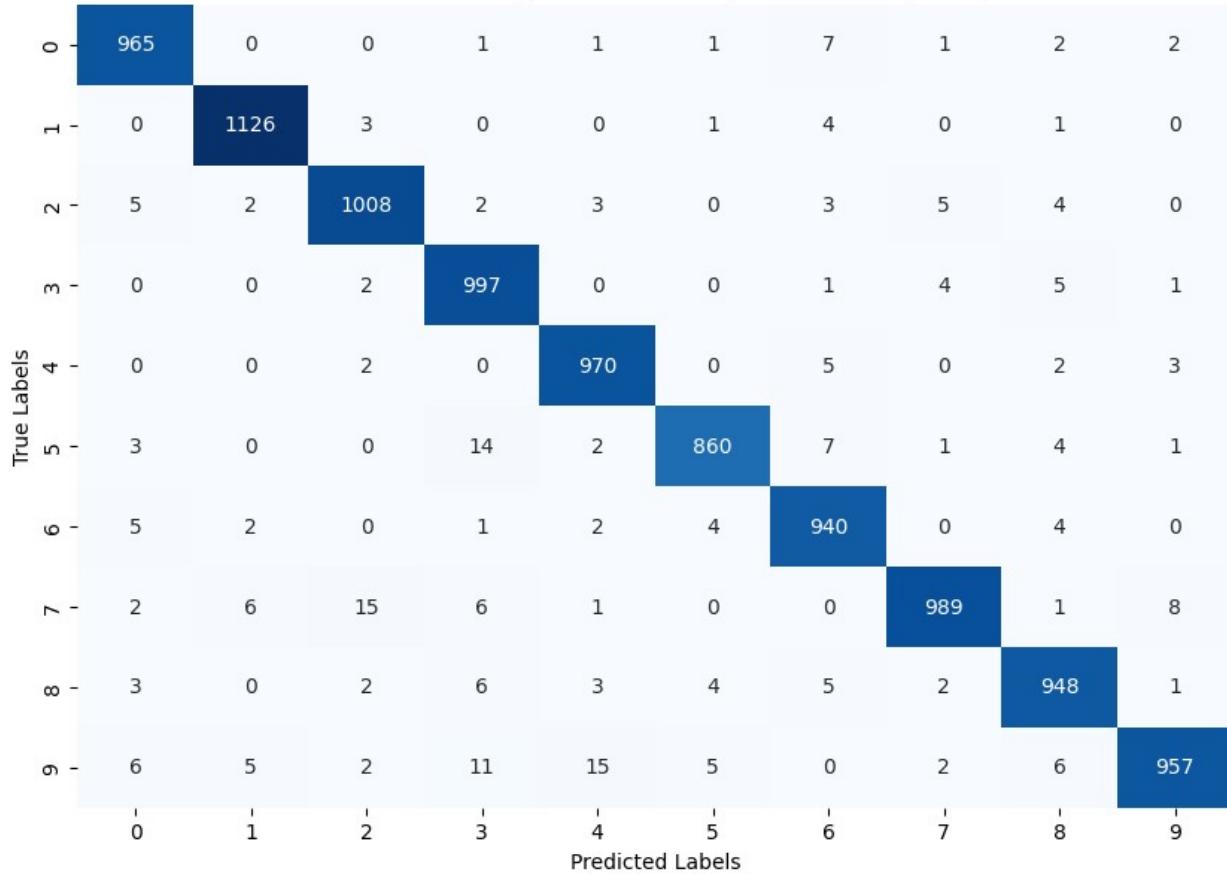
844/844 - 5s - loss: 0.1765 - accuracy: 0.9476 - val_loss: 0.1435 -
val_accuracy: 0.9607 - 5s/epoch - 7ms/step
Epoch 8/20
844/844 - 6s - loss: 0.1602 - accuracy: 0.9525 - val_loss: 0.1279 -
val_accuracy: 0.9643 - 6s/epoch - 7ms/step
Epoch 9/20
844/844 - 5s - loss: 0.1485 - accuracy: 0.9557 - val_loss: 0.1138 -
val_accuracy: 0.9695 - 5s/epoch - 6ms/step
Epoch 10/20
844/844 - 5s - loss: 0.1376 - accuracy: 0.9590 - val_loss: 0.1114 -
val_accuracy: 0.9695 - 5s/epoch - 6ms/step
Epoch 11/20
844/844 - 5s - loss: 0.1272 - accuracy: 0.9620 - val_loss: 0.1044 -
val_accuracy: 0.9715 - 5s/epoch - 6ms/step
Epoch 12/20
844/844 - 5s - loss: 0.1190 - accuracy: 0.9652 - val_loss: 0.0987 -
val_accuracy: 0.9735 - 5s/epoch - 6ms/step
Epoch 13/20
844/844 - 5s - loss: 0.1104 - accuracy: 0.9675 - val_loss: 0.0939 -
val_accuracy: 0.9747 - 5s/epoch - 6ms/step
Epoch 14/20
844/844 - 5s - loss: 0.1032 - accuracy: 0.9699 - val_loss: 0.0913 -
val_accuracy: 0.9753 - 5s/epoch - 6ms/step
Epoch 15/20
844/844 - 5s - loss: 0.0970 - accuracy: 0.9716 - val_loss: 0.0876 -
val_accuracy: 0.9765 - 5s/epoch - 6ms/step
Epoch 16/20
844/844 - 5s - loss: 0.0911 - accuracy: 0.9738 - val_loss: 0.0843 -
val_accuracy: 0.9763 - 5s/epoch - 6ms/step
Epoch 17/20
844/844 - 5s - loss: 0.0866 - accuracy: 0.9748 - val_loss: 0.0972 -
val_accuracy: 0.9730 - 5s/epoch - 6ms/step
Epoch 18/20
844/844 - 5s - loss: 0.0819 - accuracy: 0.9764 - val_loss: 0.0777 -
val_accuracy: 0.9798 - 5s/epoch - 6ms/step
Epoch 19/20
844/844 - 5s - loss: 0.0769 - accuracy: 0.9781 - val_loss: 0.0753 -
val_accuracy: 0.9798 - 5s/epoch - 6ms/step
Epoch 20/20
844/844 - 5s - loss: 0.0735 - accuracy: 0.9787 - val_loss: 0.0746 -
val_accuracy: 0.9795 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for sigmoid function, 64 batch size and 20 epochs...
Confusion Matrix
[[ 965   0   0   1   1   1   7   1   2   2]
 [  0 1126   3   0   0   1   4   0   1   0]
 [  5   2 1008   2   3   0   3   5   4   0]
 [  0   0   2  997   0   0   1   4   5   1]
 [  0   0   2   0  970   0   5   0   2   3]]
```

```
[ 3 0 0 14 2 860 7 1 4 1
[ 5 2 0 1 2 4 940 0 4 0
[ 2 6 15 6 1 0 0 989 1 8
[ 3 0 2 6 3 4 5 2 948 1
[ 6 5 2 11 15 5 0 2 6 957]
```

Precision: 0.9761

Recall: 0.9760

Confusion Matrix for sigmoid Activation, 64 batch size, 20 epochs...



Training model with sigmoid activation function, 128 batch sizes, 5 epochs_list

Epoch 1/5

422/422 - 4s - loss: 1.2286 - accuracy: 0.6730 - val_loss: 0.4844 - val_accuracy: 0.8872 - 4s/epoch - 10ms/step

Epoch 2/5

422/422 - 4s - loss: 0.4462 - accuracy: 0.8785 - val_loss: 0.3229 - val_accuracy: 0.9115 - 4s/epoch - 8ms/step

Epoch 3/5

422/422 - 3s - loss: 0.3447 - accuracy: 0.9014 - val_loss: 0.2570 - val_accuracy: 0.9288 - 3s/epoch - 8ms/step

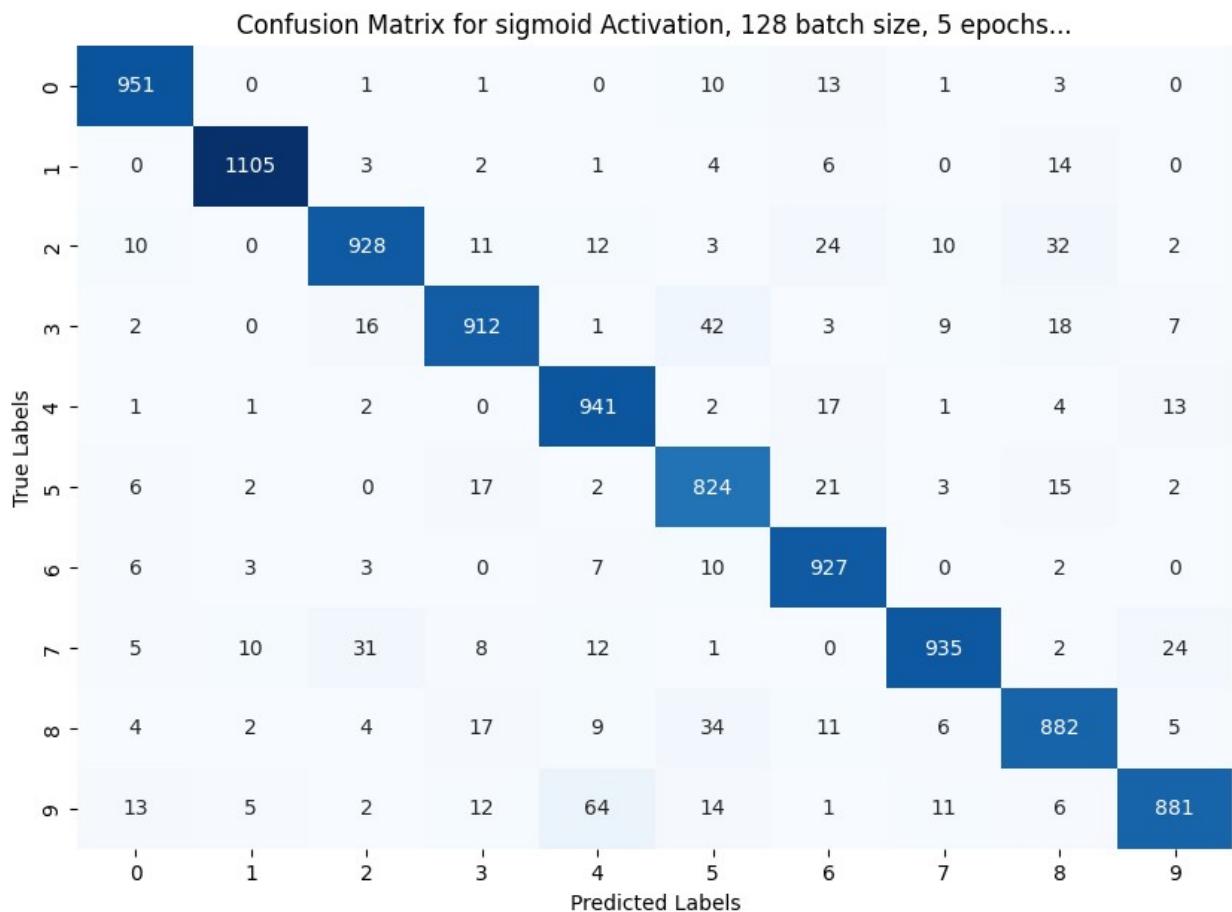
Epoch 4/5

422/422 - 4s - loss: 0.3022 - accuracy: 0.9119 - val_loss: 0.2325 -

```

val_accuracy: 0.9333 - 4s/epoch - 8ms/step
Epoch 5/5
422/422 - 4s - loss: 0.2729 - accuracy: 0.9192 - val_loss: 0.2145 -
val_accuracy: 0.9370 - 4s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for sigmoid function, 128 batch size and 5 epochs...
Confusion Matrix
[[ 951   0   1   1   0   10  13   1   3   0]
 [  0 1105   3   2   1   4   6   0  14   0]
 [ 10   0 928  11  12   3  24  10  32   2]
 [  2   0 16 912   1  42   3   9  18   7]
 [  1   1   2   0 941   2  17   1   4  13]
 [  6   2   0  17   2 824   21   3  15   2]
 [  6   3   3   0   7 10 927   0   2   0]
 [  5  10  31   8  12   1   0 935   2  24]
 [  4   2   4  17   9  34   11   6 882   5]
 [ 13   5   2  12  64  14   1  11   6 881]]
Precision: 0.9296
Recall: 0.9286

```



```
Training model with sigmoid activation function, 128 batch sizes, 15 epochs_list
Epoch 1/15
422/422 - 4s - loss: 1.1470 - accuracy: 0.6898 - val_loss: 0.4494 -
val_accuracy: 0.8920 - 4s/epoch - 10ms/step
Epoch 2/15
422/422 - 4s - loss: 0.4295 - accuracy: 0.8809 - val_loss: 0.3015 -
val_accuracy: 0.9183 - 4s/epoch - 9ms/step
Epoch 3/15
422/422 - 4s - loss: 0.3427 - accuracy: 0.9009 - val_loss: 0.2580 -
val_accuracy: 0.9278 - 4s/epoch - 9ms/step
Epoch 4/15
422/422 - 4s - loss: 0.3010 - accuracy: 0.9118 - val_loss: 0.2216 -
val_accuracy: 0.9370 - 4s/epoch - 9ms/step
Epoch 5/15
422/422 - 4s - loss: 0.2745 - accuracy: 0.9193 - val_loss: 0.2053 -
val_accuracy: 0.9418 - 4s/epoch - 9ms/step
Epoch 6/15
422/422 - 4s - loss: 0.2504 - accuracy: 0.9256 - val_loss: 0.2006 -
val_accuracy: 0.9422 - 4s/epoch - 9ms/step
Epoch 7/15
422/422 - 4s - loss: 0.2334 - accuracy: 0.9309 - val_loss: 0.1788 -
val_accuracy: 0.9480 - 4s/epoch - 9ms/step
Epoch 8/15
422/422 - 4s - loss: 0.2143 - accuracy: 0.9375 - val_loss: 0.1669 -
val_accuracy: 0.9520 - 4s/epoch - 9ms/step
Epoch 9/15
422/422 - 4s - loss: 0.2019 - accuracy: 0.9409 - val_loss: 0.1556 -
val_accuracy: 0.9558 - 4s/epoch - 9ms/step
Epoch 10/15
422/422 - 4s - loss: 0.1879 - accuracy: 0.9441 - val_loss: 0.1495 -
val_accuracy: 0.9585 - 4s/epoch - 9ms/step
Epoch 11/15
422/422 - 4s - loss: 0.1770 - accuracy: 0.9481 - val_loss: 0.1494 -
val_accuracy: 0.9615 - 4s/epoch - 9ms/step
Epoch 12/15
422/422 - 4s - loss: 0.1673 - accuracy: 0.9508 - val_loss: 0.1318 -
val_accuracy: 0.9627 - 4s/epoch - 9ms/step
Epoch 13/15
422/422 - 4s - loss: 0.1569 - accuracy: 0.9543 - val_loss: 0.1303 -
val_accuracy: 0.9630 - 4s/epoch - 9ms/step
Epoch 14/15
422/422 - 4s - loss: 0.1491 - accuracy: 0.9566 - val_loss: 0.1167 -
val_accuracy: 0.9680 - 4s/epoch - 9ms/step
Epoch 15/15
422/422 - 4s - loss: 0.1410 - accuracy: 0.9585 - val_loss: 0.1140 -
val_accuracy: 0.9710 - 4s/epoch - 9ms/step
313/313 [=====] - 1s 2ms/step
Results for sigmoid function, 128 batch size and 15 epochs...
Confusion Matrix
```

```
[[ 966  0  0  1  0  2  4  1  5  1
 [ 0 1119  2  1  1  1  4  0  7  0]
 [ 5  0 978 13  6  0  6  4 18  2]
 [ 0  0  2 982  0  4  0  3 17  2]
 [ 1  0  4  0 949  0  5  2  5 16]
 [ 5  1  0 19  1 842  6  1 16  1]
 [ 8  3  0  0  7  9 923  0  8  0]
 [ 2  5 20 15  5  0  0 953  3 25]
 [ 4  0  2 14  3  1  3  2 943  2]
 [ 8  5  0 16 17  4  1  6  7 945]]
```

Precision: 0.9606

Recall: 0.9600

Confusion Matrix for sigmoid Activation, 128 batch size, 15 epochs...



Training model with sigmoid activation function, 128 batch sizes, 20 epochs list

Epoch 1/20

422/422 - 4s - loss: 1.2321 - accuracy: 0.6731 - val_loss: 0.4545 - val_accuracy: 0.9007 - 4s/epoch - 10ms/step

Epoch 2/20

422/422 - 4s - loss: 0.4381 - accuracy: 0.8789 - val_loss: 0.3152 - val_accuracy: 0.9090 - 4s/epoch - 9ms/step

```
Epoch 3/20
422/422 - 4s - loss: 0.3480 - accuracy: 0.8992 - val_loss: 0.2610 -
val_accuracy: 0.9262 - 4s/epoch - 8ms/step
Epoch 4/20
422/422 - 4s - loss: 0.3082 - accuracy: 0.9085 - val_loss: 0.2271 -
val_accuracy: 0.9340 - 4s/epoch - 8ms/step
Epoch 5/20
422/422 - 4s - loss: 0.2785 - accuracy: 0.9169 - val_loss: 0.2079 -
val_accuracy: 0.9423 - 4s/epoch - 8ms/step
Epoch 6/20
422/422 - 4s - loss: 0.2550 - accuracy: 0.9240 - val_loss: 0.1987 -
val_accuracy: 0.9457 - 4s/epoch - 8ms/step
Epoch 7/20
422/422 - 4s - loss: 0.2354 - accuracy: 0.9291 - val_loss: 0.1806 -
val_accuracy: 0.9480 - 4s/epoch - 8ms/step
Epoch 8/20
422/422 - 4s - loss: 0.2179 - accuracy: 0.9352 - val_loss: 0.1692 -
val_accuracy: 0.9522 - 4s/epoch - 9ms/step
Epoch 9/20
422/422 - 4s - loss: 0.2031 - accuracy: 0.9394 - val_loss: 0.1593 -
val_accuracy: 0.9540 - 4s/epoch - 8ms/step
Epoch 10/20
422/422 - 4s - loss: 0.1909 - accuracy: 0.9438 - val_loss: 0.1434 -
val_accuracy: 0.9617 - 4s/epoch - 8ms/step
Epoch 11/20
422/422 - 4s - loss: 0.1784 - accuracy: 0.9475 - val_loss: 0.1346 -
val_accuracy: 0.9635 - 4s/epoch - 8ms/step
Epoch 12/20
422/422 - 4s - loss: 0.1690 - accuracy: 0.9497 - val_loss: 0.1302 -
val_accuracy: 0.9660 - 4s/epoch - 8ms/step
Epoch 13/20
422/422 - 4s - loss: 0.1590 - accuracy: 0.9533 - val_loss: 0.1245 -
val_accuracy: 0.9665 - 4s/epoch - 8ms/step
Epoch 14/20
422/422 - 4s - loss: 0.1493 - accuracy: 0.9566 - val_loss: 0.1162 -
val_accuracy: 0.9692 - 4s/epoch - 8ms/step
Epoch 15/20
422/422 - 4s - loss: 0.1422 - accuracy: 0.9581 - val_loss: 0.1171 -
val_accuracy: 0.9675 - 4s/epoch - 9ms/step
Epoch 16/20
422/422 - 4s - loss: 0.1351 - accuracy: 0.9598 - val_loss: 0.1128 -
val_accuracy: 0.9708 - 4s/epoch - 9ms/step
Epoch 17/20
422/422 - 4s - loss: 0.1286 - accuracy: 0.9626 - val_loss: 0.1035 -
val_accuracy: 0.9727 - 4s/epoch - 9ms/step
Epoch 18/20
422/422 - 4s - loss: 0.1227 - accuracy: 0.9648 - val_loss: 0.0983 -
val_accuracy: 0.9745 - 4s/epoch - 8ms/step
Epoch 19/20
```

```
422/422 - 4s - loss: 0.1170 - accuracy: 0.9659 - val_loss: 0.0983 -  
val_accuracy: 0.9747 - 4s/epoch - 8ms/step
```

```
Epoch 20/20
```

```
422/422 - 4s - loss: 0.1126 - accuracy: 0.9678 - val_loss: 0.0964 -  
val_accuracy: 0.9750 - 4s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for sigmoid function, 128 batch size and 20 epochs...
```

```
Confusion Matrix
```

```
[[ 965 0 0 1 0 2 8 1 2 1 ]  
[ 0 1127 2 1 0 1 3 0 1 0 ]  
[ 6 5 992 8 2 0 5 7 6 1 ]  
[ 0 0 2 994 0 1 1 8 3 1 ]  
[ 1 0 5 0 940 0 11 5 2 18 ]  
[ 8 1 2 16 0 850 8 1 4 2 ]  
[ 8 3 0 1 2 4 936 0 4 0 ]  
[ 3 8 14 8 0 0 988 0 7 ]  
[ 4 2 4 14 3 2 5 6 932 2 ]  
[ 5 6 2 15 12 6 1 10 3 949 ]]
```

```
Precision: 0.9675
```

```
Recall: 0.9673
```

Confusion Matrix for sigmoid Activation, 128 batch size, 20 epochs...



```
Training model with sigmoid activation function, 256 batch sizes, 5 epochs_list
Epoch 1/5
211/211 - 3s - loss: 1.6790 - accuracy: 0.5656 - val_loss: 0.8923 -
val_accuracy: 0.8363 - 3s/epoch - 16ms/step
Epoch 2/5
211/211 - 3s - loss: 0.6862 - accuracy: 0.8387 - val_loss: 0.4517 -
val_accuracy: 0.8992 - 3s/epoch - 13ms/step
Epoch 3/5
211/211 - 3s - loss: 0.4656 - accuracy: 0.8779 - val_loss: 0.3398 -
val_accuracy: 0.9115 - 3s/epoch - 13ms/step
Epoch 4/5
211/211 - 3s - loss: 0.3904 - accuracy: 0.8918 - val_loss: 0.2965 -
val_accuracy: 0.9203 - 3s/epoch - 13ms/step
Epoch 5/5
211/211 - 3s - loss: 0.3504 - accuracy: 0.9005 - val_loss: 0.2753 -
val_accuracy: 0.9235 - 3s/epoch - 13ms/step
313/313 [=====] - 1s 2ms/step
Results for sigmoid function, 256 batch size and 5 epochs...
Confusion Matrix
[[ 965    0    0    2    0    4    3    1    5    0]
 [  0 1102    2    2    0    3    4    0   22    0]
 [ 23    3  880   13   13    3   22   20   49    6]
 [  6    2   15  888    0   35    2   15   39    8]
 [  2    4    6    0  865    4   17    2   21   61]
 [ 17    3    2   34    4  773   12    6   37    4]
 [ 23    4    8    1    9   22  885    1    5    0]
 [  6   13   21    5    9    0    0  928    8   38]
 [ 10    2    5   13    5   20    7    6  894   12]
 [ 17    5    4   13   15    9    0   15   12  919]]
```

Precision: 0.9116
Recall: 0.9099

Confusion Matrix for sigmoid Activation, 256 batch size, 5 epochs...

	0	1	2	3	4	5	6	7	8	9	
0	965	0	0	2	0	4	3	1	5	0	
1	0	1102	2	2	0	3	4	0	22	0	
2	23	3	880	13	13	3	22	20	49	6	
3	6	2	15	888	0	35	2	15	39	8	
4	2	4	6	0	865	4	17	2	21	61	
5	17	3	2	34	4	773	12	6	37	4	
6	23	4	8	1	9	22	885	1	5	0	
7	6	13	21	5	9	0	0	928	8	38	
8	10	2	5	13	5	20	7	6	894	12	
9	17	5	4	13	15	9	0	15	12	919	
	0	1	2	3	4	5	6	7	8	9	
	True Labels										Predicted Labels

```
Training model with sigmoid activation function, 256 batch sizes, 15 epochs_list
Epoch 1/15
211/211 - 3s - loss: 1.6235 - accuracy: 0.5944 - val_loss: 0.8255 -
val_accuracy: 0.8607 - 3s/epoch - 16ms/step
Epoch 2/15
211/211 - 3s - loss: 0.6524 - accuracy: 0.8469 - val_loss: 0.4399 -
val_accuracy: 0.9002 - 3s/epoch - 13ms/step
Epoch 3/15
211/211 - 3s - loss: 0.4577 - accuracy: 0.8764 - val_loss: 0.3384 -
val_accuracy: 0.9125 - 3s/epoch - 13ms/step
Epoch 4/15
211/211 - 3s - loss: 0.3885 - accuracy: 0.8909 - val_loss: 0.2960 -
val_accuracy: 0.9173 - 3s/epoch - 13ms/step
Epoch 5/15
211/211 - 3s - loss: 0.3514 - accuracy: 0.8994 - val_loss: 0.2710 -
val_accuracy: 0.9267 - 3s/epoch - 13ms/step
Epoch 6/15
211/211 - 3s - loss: 0.3251 - accuracy: 0.9059 - val_loss: 0.2569 -
val_accuracy: 0.9287 - 3s/epoch - 13ms/step
Epoch 7/15
```

```
211/211 - 3s - loss: 0.3070 - accuracy: 0.9096 - val_loss: 0.2430 -  
val_accuracy: 0.9322 - 3s/epoch - 13ms/step  
Epoch 8/15  
211/211 - 3s - loss: 0.2920 - accuracy: 0.9145 - val_loss: 0.2268 -  
val_accuracy: 0.9370 - 3s/epoch - 13ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.2763 - accuracy: 0.9189 - val_loss: 0.2190 -  
val_accuracy: 0.9375 - 3s/epoch - 13ms/step  
Epoch 10/15  
211/211 - 3s - loss: 0.2658 - accuracy: 0.9207 - val_loss: 0.2105 -  
val_accuracy: 0.9413 - 3s/epoch - 13ms/step  
Epoch 11/15  
211/211 - 3s - loss: 0.2556 - accuracy: 0.9255 - val_loss: 0.2036 -  
val_accuracy: 0.9398 - 3s/epoch - 13ms/step  
Epoch 12/15  
211/211 - 3s - loss: 0.2457 - accuracy: 0.9274 - val_loss: 0.1898 -  
val_accuracy: 0.9477 - 3s/epoch - 13ms/step  
Epoch 13/15  
211/211 - 3s - loss: 0.2350 - accuracy: 0.9313 - val_loss: 0.1885 -  
val_accuracy: 0.9477 - 3s/epoch - 13ms/step  
Epoch 14/15  
211/211 - 3s - loss: 0.2264 - accuracy: 0.9335 - val_loss: 0.1756 -  
val_accuracy: 0.9495 - 3s/epoch - 13ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.2175 - accuracy: 0.9360 - val_loss: 0.1715 -  
val_accuracy: 0.9543 - 3s/epoch - 13ms/step  
313/313 [=====] - 1s 2ms/step  
Results for sigmoid function, 256 batch size and 15 epochs...  
Confusion Matrix  
[[ 961  0  2  2  0  4  7  1  3  0]  
[  0 1116  2  2  0  1  5  0  9  0]  
[  8  2 959  14  7  1  14  10  15  2]  
[  2  0 17 953  0  10  2  11  13  2]  
[  1  4  4  1  928  1  15  2  6  20]  
[  6  3  3  32  2  804  17  3  19  3]  
[  9  3  4  1  4  8  925  1  3  0]  
[  2  8  27  11  8  0  0  957  0  15]  
[  6  2  7  19  7  13  9  6  902  3]  
[ 10  8  2  14  32  5  1  15  6  916]]  
Precision: 0.9423  
Recall: 0.9421
```

Confusion Matrix for sigmoid Activation, 256 batch size, 15 epochs...

	0	1	2	3	4	5	6	7	8	9	10
True Labels	961	0	2	2	0	4	7	1	3	0	0
0	0	1116	2	2	0	1	5	0	9	0	0
1	8	2	959	14	7	1	14	10	15	2	0
2	2	0	17	953	0	10	2	11	13	2	0
3	1	4	4	1	928	1	15	2	6	20	0
4	6	3	3	32	2	804	17	3	19	3	0
5	9	3	4	1	4	8	925	1	3	0	0
6	2	8	27	11	8	0	0	957	0	15	0
7	6	2	7	19	7	13	9	6	902	3	0
8	10	8	2	14	32	5	1	15	6	916	0
9	0	1	2	3	4	5	6	7	8	9	0
Predicted Labels											

```
Training model with sigmoid activation function, 256 batch sizes, 20
epochs_list
Epoch 1/20
211/211 - 4s - loss: 1.6503 - accuracy: 0.5934 - val_loss: 0.8552 -
val_accuracy: 0.8390 - 4s/epoch - 17ms/step
Epoch 2/20
211/211 - 3s - loss: 0.6712 - accuracy: 0.8431 - val_loss: 0.4495 -
val_accuracy: 0.8952 - 3s/epoch - 14ms/step
Epoch 3/20
211/211 - 3s - loss: 0.4631 - accuracy: 0.8790 - val_loss: 0.3438 -
val_accuracy: 0.9125 - 3s/epoch - 13ms/step
Epoch 4/20
211/211 - 3s - loss: 0.3887 - accuracy: 0.8941 - val_loss: 0.3030 -
val_accuracy: 0.9188 - 3s/epoch - 13ms/step
Epoch 5/20
211/211 - 3s - loss: 0.3493 - accuracy: 0.9020 - val_loss: 0.2686 -
val_accuracy: 0.9260 - 3s/epoch - 13ms/step
Epoch 6/20
211/211 - 3s - loss: 0.3221 - accuracy: 0.9076 - val_loss: 0.2575 -
val_accuracy: 0.9267 - 3s/epoch - 13ms/step
Epoch 7/20
```

```

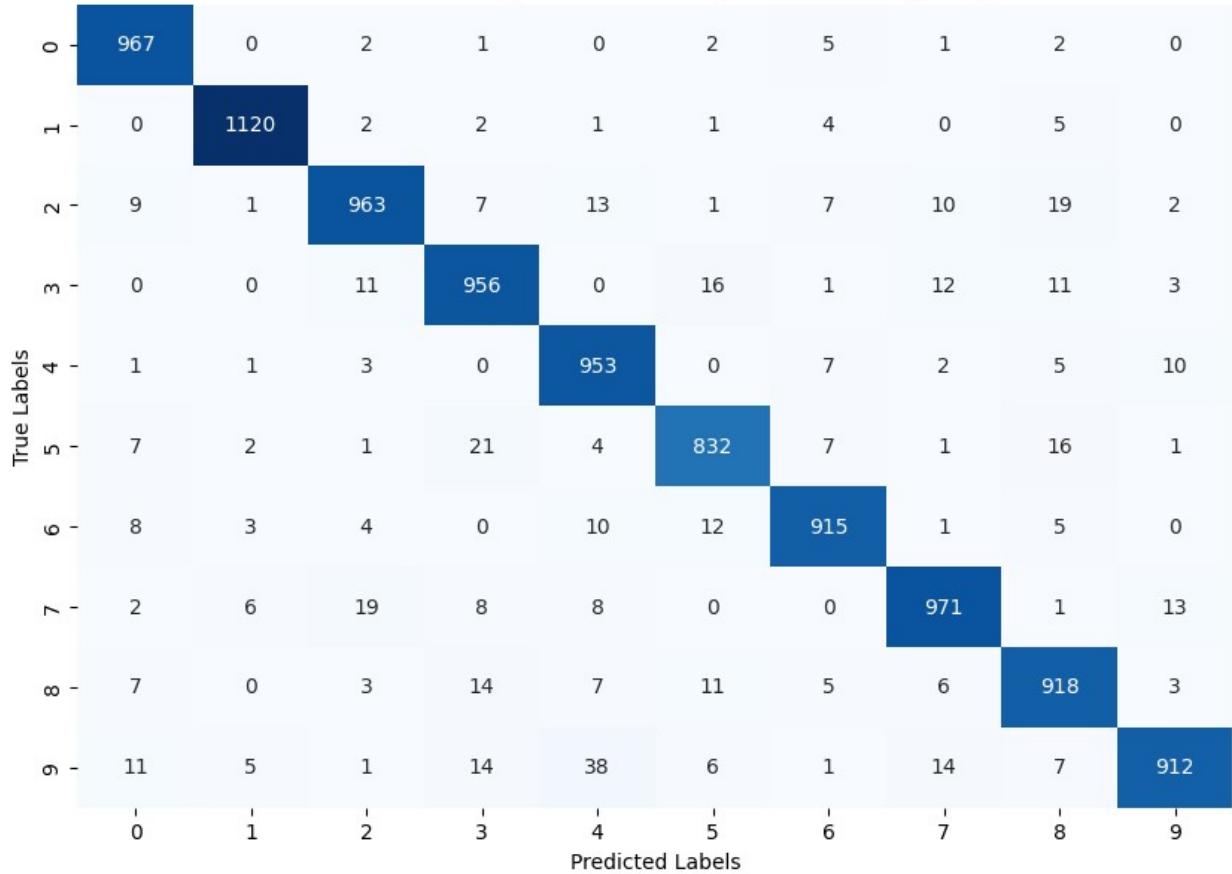
211/211 - 3s - loss: 0.3039 - accuracy: 0.9131 - val_loss: 0.2351 -
val_accuracy: 0.9355 - 3s/epoch - 13ms/step
Epoch 8/20
211/211 - 3s - loss: 0.2879 - accuracy: 0.9165 - val_loss: 0.2233 -
val_accuracy: 0.9390 - 3s/epoch - 13ms/step
Epoch 9/20
211/211 - 3s - loss: 0.2733 - accuracy: 0.9201 - val_loss: 0.2089 -
val_accuracy: 0.9437 - 3s/epoch - 13ms/step
Epoch 10/20
211/211 - 3s - loss: 0.2614 - accuracy: 0.9241 - val_loss: 0.2083 -
val_accuracy: 0.9418 - 3s/epoch - 13ms/step
Epoch 11/20
211/211 - 3s - loss: 0.2495 - accuracy: 0.9270 - val_loss: 0.2003 -
val_accuracy: 0.9443 - 3s/epoch - 13ms/step
Epoch 12/20
211/211 - 3s - loss: 0.2404 - accuracy: 0.9308 - val_loss: 0.1859 -
val_accuracy: 0.9493 - 3s/epoch - 13ms/step
Epoch 13/20
211/211 - 3s - loss: 0.2315 - accuracy: 0.9327 - val_loss: 0.1829 -
val_accuracy: 0.9493 - 3s/epoch - 13ms/step
Epoch 14/20
211/211 - 3s - loss: 0.2228 - accuracy: 0.9349 - val_loss: 0.1745 -
val_accuracy: 0.9517 - 3s/epoch - 13ms/step
Epoch 15/20
211/211 - 3s - loss: 0.2140 - accuracy: 0.9375 - val_loss: 0.1655 -
val_accuracy: 0.9538 - 3s/epoch - 13ms/step
Epoch 16/20
211/211 - 3s - loss: 0.2056 - accuracy: 0.9397 - val_loss: 0.1611 -
val_accuracy: 0.9560 - 3s/epoch - 13ms/step
Epoch 17/20
211/211 - 3s - loss: 0.1992 - accuracy: 0.9418 - val_loss: 0.1536 -
val_accuracy: 0.9573 - 3s/epoch - 13ms/step
Epoch 18/20
211/211 - 3s - loss: 0.1943 - accuracy: 0.9425 - val_loss: 0.1552 -
val_accuracy: 0.9588 - 3s/epoch - 14ms/step
Epoch 19/20
211/211 - 3s - loss: 0.1861 - accuracy: 0.9460 - val_loss: 0.1478 -
val_accuracy: 0.9598 - 3s/epoch - 14ms/step
Epoch 20/20
211/211 - 3s - loss: 0.1808 - accuracy: 0.9478 - val_loss: 0.1411 -
val_accuracy: 0.9627 - 3s/epoch - 14ms/step
313/313 [=====] - 1s 2ms/step
Results for sigmoid function, 256 batch size and 20 epochs...
Confusion Matrix
[[ 967   0   2   1   0   2   5   1   2   0]
 [  0 1120   2   2   1   1   4   0   5   0]
 [  9   1  963   7  13   1   7  10  19   2]
 [  0   0   11  956   0  16   1  12  11   3]
 [  1   1    3   0  953   0    7   2   5  10]]
```

```
[ 7  2  1 21  4 832  7  1 16  1
 [ 8  3  4  0 10 12 915  1  5  0]
 [ 2  6 19  8  8  0  0 971  1 13]
 [ 7  0  3 14  7 11  5  6 918  3]
 [11  5  1 14 38  6  1 14  7 912]]
```

Precision: 0.9510

Recall: 0.9507

Confusion Matrix for sigmoid Activation, 256 batch size, 20 epochs...



Training model with sigmoid activation function, 16 batch sizes, 5 epochs_list

Epoch 1/5

3375/3375 - 16s - loss: 0.5293 - accuracy: 0.8357 - val_loss: 0.2222 - val_accuracy: 0.9340 - 16s/epoch - 5ms/step

Epoch 2/5

3375/3375 - 15s - loss: 0.2311 - accuracy: 0.9301 - val_loss: 0.1662 - val_accuracy: 0.9515 - 15s/epoch - 4ms/step

Epoch 3/5

3375/3375 - 15s - loss: 0.1684 - accuracy: 0.9487 - val_loss: 0.1257 - val_accuracy: 0.9655 - 15s/epoch - 4ms/step

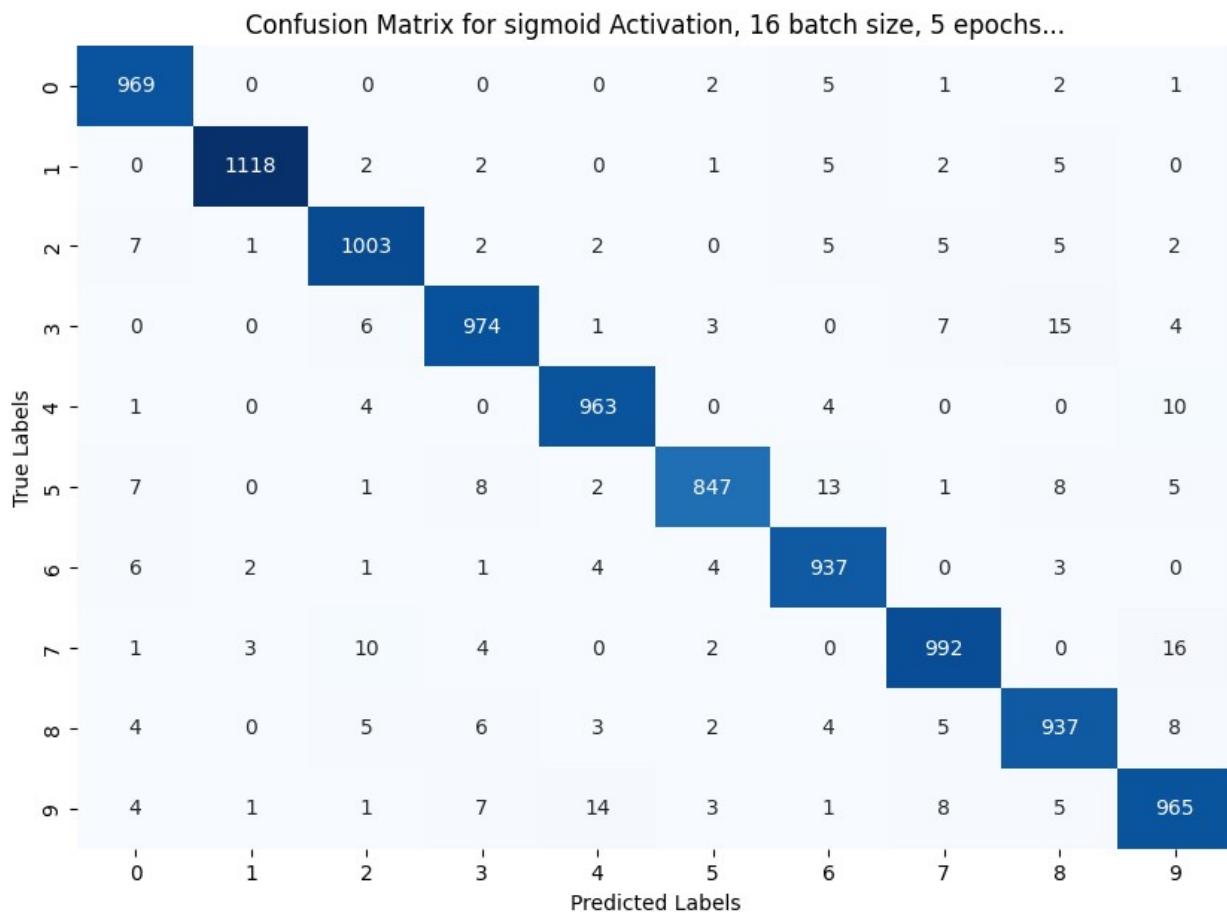
Epoch 4/5

3375/3375 - 15s - loss: 0.1315 - accuracy: 0.9608 - val_loss: 0.1090 -

```

val_accuracy: 0.9712 - 15s/epoch - 5ms/step
Epoch 5/5
3375/3375 - 16s - loss: 0.1085 - accuracy: 0.9676 - val_loss: 0.0882 -
val_accuracy: 0.9748 - 16s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Results for sigmoid function, 16 batch size and 5 epochs...
Confusion Matrix
[[ 969   0   0   0   0   2   5   1   2   1]
 [ 0 1118   2   2   0   1   5   2   5   0]
 [ 7   1 1003   2   2   0   5   5   5   2]
 [ 0   0   6 974   1   3   0   7 15   4]
 [ 1   0   4   0 963   0   4   0   0 10]
 [ 7   0   1   8   2 847   13   1   8   5]
 [ 6   2   1   1   4   4 937   0   3   0]
 [ 1   3 10   4   0   2   0 992   0 16]
 [ 4   0   5   6   3   2   4   5 937   8]
 [ 4   1   1   7 14   3   1   8   5 965]]
Precision: 0.9706
Recall: 0.9705

```



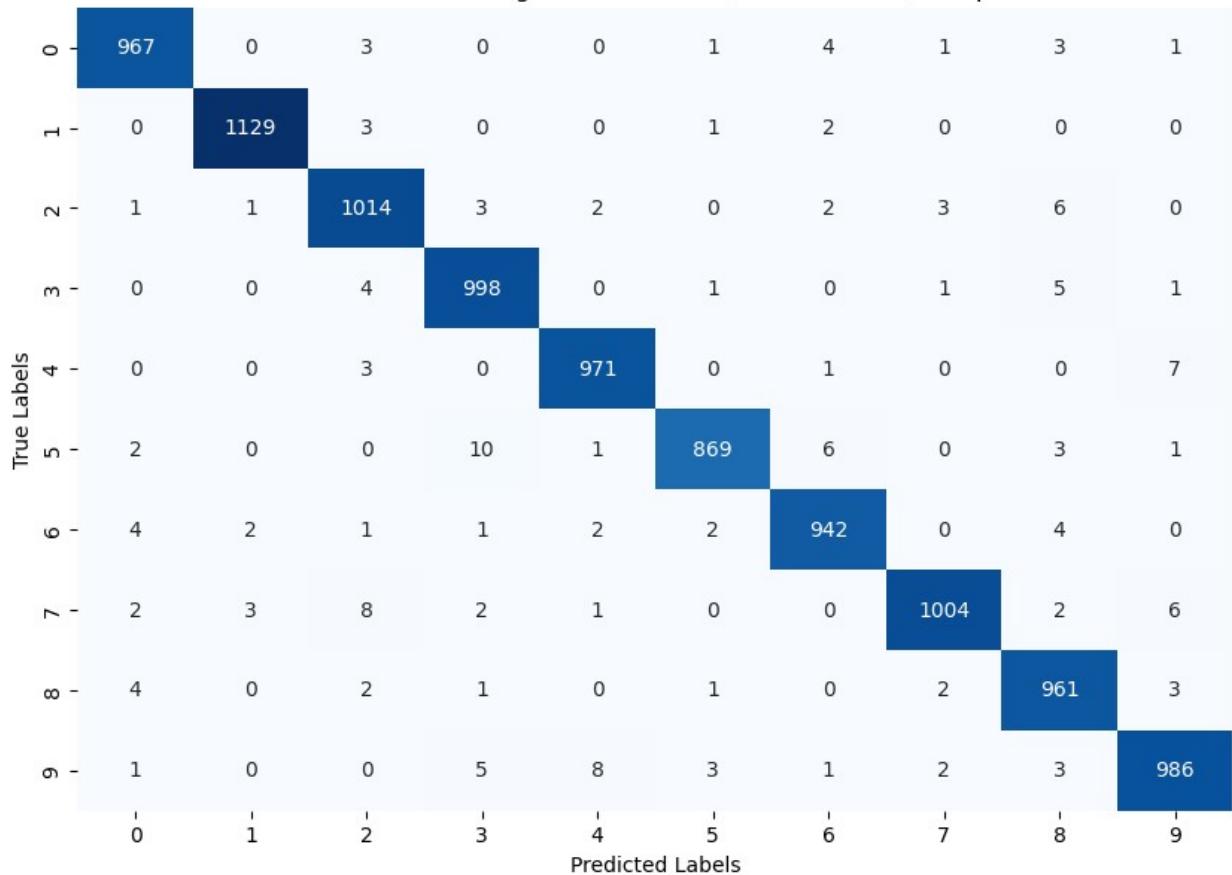
```
Training model with sigmoid activation function, 16 batch sizes, 15
epochs_list
Epoch 1/15
3375/3375 - 16s - loss: 0.5351 - accuracy: 0.8326 - val_loss: 0.1979 -
val_accuracy: 0.9425 - 16s/epoch - 5ms/step
Epoch 2/15
3375/3375 - 16s - loss: 0.2236 - accuracy: 0.9328 - val_loss: 0.1475 -
val_accuracy: 0.9585 - 16s/epoch - 5ms/step
Epoch 3/15
3375/3375 - 16s - loss: 0.1551 - accuracy: 0.9527 - val_loss: 0.1230 -
val_accuracy: 0.9638 - 16s/epoch - 5ms/step
Epoch 4/15
3375/3375 - 16s - loss: 0.1183 - accuracy: 0.9647 - val_loss: 0.0873 -
val_accuracy: 0.9752 - 16s/epoch - 5ms/step
Epoch 5/15
3375/3375 - 15s - loss: 0.0949 - accuracy: 0.9715 - val_loss: 0.0867 -
val_accuracy: 0.9757 - 15s/epoch - 5ms/step
Epoch 6/15
3375/3375 - 15s - loss: 0.0786 - accuracy: 0.9768 - val_loss: 0.0707 -
val_accuracy: 0.9783 - 15s/epoch - 4ms/step
Epoch 7/15
3375/3375 - 15s - loss: 0.0664 - accuracy: 0.9803 - val_loss: 0.0620 -
val_accuracy: 0.9828 - 15s/epoch - 5ms/step
Epoch 8/15
3375/3375 - 15s - loss: 0.0568 - accuracy: 0.9836 - val_loss: 0.0563 -
val_accuracy: 0.9862 - 15s/epoch - 4ms/step
Epoch 9/15
3375/3375 - 15s - loss: 0.0498 - accuracy: 0.9856 - val_loss: 0.0639 -
val_accuracy: 0.9815 - 15s/epoch - 5ms/step
Epoch 10/15
3375/3375 - 15s - loss: 0.0429 - accuracy: 0.9879 - val_loss: 0.0587 -
val_accuracy: 0.9848 - 15s/epoch - 5ms/step
Epoch 11/15
3375/3375 - 16s - loss: 0.0385 - accuracy: 0.9891 - val_loss: 0.0518 -
val_accuracy: 0.9873 - 16s/epoch - 5ms/step
Epoch 12/15
3375/3375 - 16s - loss: 0.0346 - accuracy: 0.9904 - val_loss: 0.0549 -
val_accuracy: 0.9852 - 16s/epoch - 5ms/step
Epoch 13/15
3375/3375 - 15s - loss: 0.0309 - accuracy: 0.9916 - val_loss: 0.0507 -
val_accuracy: 0.9875 - 15s/epoch - 5ms/step
Epoch 14/15
3375/3375 - 16s - loss: 0.0273 - accuracy: 0.9927 - val_loss: 0.0459 -
val_accuracy: 0.9873 - 16s/epoch - 5ms/step
Epoch 15/15
3375/3375 - 16s - loss: 0.0244 - accuracy: 0.9939 - val_loss: 0.0511 -
val_accuracy: 0.9875 - 16s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Results for sigmoid function, 16 batch size and 15 epochs...
Confusion Matrix
```

```
[[ 967  0  3  0  0  1  4  1  3  1]
 [ 0 1129  3  0  0  1  2  0  0  0]
 [ 1  1 1014  3  2  0  2  3  6  0]
 [ 0  0  4 998  0  1  0  1  5  1]
 [ 0  0  3  0 971  0  1  0  0  7]
 [ 2  0  0  10  1 869  6  0  3  1]
 [ 4  2  1  1  2  2 942  0  4  0]
 [ 2  3  8  2  1  0  0 1004  2  6]
 [ 4  0  2  1  0  1  0  2 961  3]
 [ 1  0  0  5  8  3  1  2  3 986]]
```

Precision: 0.9841

Recall: 0.9841

Confusion Matrix for sigmoid Activation, 16 batch size, 15 epochs...



Training model with sigmoid activation function, 16 batch sizes, 20 epochs list

Epoch 1/20

3375/3375 - 16s - loss: 0.5477 - accuracy: 0.8288 - val_loss: 0.2020 - val_accuracy: 0.9417 - 16s/epoch - 5ms/step

Epoch 2/20

3375/3375 - 15s - loss: 0.2319 - accuracy: 0.9296 - val_loss: 0.1538 - val_accuracy: 0.9575 - 15s/epoch - 4ms/step

```
Epoch 3/20
3375/3375 - 15s - loss: 0.1655 - accuracy: 0.9499 - val_loss: 0.1195 -
val_accuracy: 0.9655 - 15s/epoch - 4ms/step
Epoch 4/20
3375/3375 - 15s - loss: 0.1304 - accuracy: 0.9610 - val_loss: 0.1003 -
val_accuracy: 0.9712 - 15s/epoch - 4ms/step
Epoch 5/20
3375/3375 - 15s - loss: 0.1042 - accuracy: 0.9689 - val_loss: 0.0868 -
val_accuracy: 0.9753 - 15s/epoch - 5ms/step
Epoch 6/20
3375/3375 - 15s - loss: 0.0877 - accuracy: 0.9740 - val_loss: 0.0754 -
val_accuracy: 0.9792 - 15s/epoch - 5ms/step
Epoch 7/20
3375/3375 - 15s - loss: 0.0745 - accuracy: 0.9774 - val_loss: 0.0692 -
val_accuracy: 0.9815 - 15s/epoch - 4ms/step
Epoch 8/20
3375/3375 - 16s - loss: 0.0654 - accuracy: 0.9805 - val_loss: 0.0727 -
val_accuracy: 0.9780 - 16s/epoch - 5ms/step
Epoch 9/20
3375/3375 - 15s - loss: 0.0575 - accuracy: 0.9827 - val_loss: 0.0748 -
val_accuracy: 0.9795 - 15s/epoch - 4ms/step
Epoch 10/20
3375/3375 - 16s - loss: 0.0498 - accuracy: 0.9859 - val_loss: 0.0606 -
val_accuracy: 0.9817 - 16s/epoch - 5ms/step
Epoch 11/20
3375/3375 - 16s - loss: 0.0442 - accuracy: 0.9873 - val_loss: 0.0589 -
val_accuracy: 0.9838 - 16s/epoch - 5ms/step
Epoch 12/20
3375/3375 - 16s - loss: 0.0394 - accuracy: 0.9885 - val_loss: 0.0580 -
val_accuracy: 0.9850 - 16s/epoch - 5ms/step
Epoch 13/20
3375/3375 - 15s - loss: 0.0354 - accuracy: 0.9898 - val_loss: 0.0576 -
val_accuracy: 0.9840 - 15s/epoch - 5ms/step
Epoch 14/20
3375/3375 - 15s - loss: 0.0315 - accuracy: 0.9913 - val_loss: 0.0563 -
val_accuracy: 0.9852 - 15s/epoch - 5ms/step
Epoch 15/20
3375/3375 - 15s - loss: 0.0286 - accuracy: 0.9922 - val_loss: 0.0578 -
val_accuracy: 0.9847 - 15s/epoch - 5ms/step
Epoch 16/20
3375/3375 - 15s - loss: 0.0262 - accuracy: 0.9929 - val_loss: 0.0539 -
val_accuracy: 0.9855 - 15s/epoch - 5ms/step
Epoch 17/20
3375/3375 - 15s - loss: 0.0233 - accuracy: 0.9945 - val_loss: 0.0563 -
val_accuracy: 0.9850 - 15s/epoch - 5ms/step
Epoch 18/20
3375/3375 - 15s - loss: 0.0209 - accuracy: 0.9950 - val_loss: 0.0554 -
val_accuracy: 0.9848 - 15s/epoch - 5ms/step
Epoch 19/20
```

3375/3375 - 15s - loss: 0.0191 - accuracy: 0.9957 - val_loss: 0.0499 -
val_accuracy: 0.9860 - 15s/epoch - 4ms/step

Epoch 20/20

3375/3375 - 15s - loss: 0.0169 - accuracy: 0.9962 - val_loss: 0.0524 -
val_accuracy: 0.9862 - 15s/epoch - 4ms/step

313/313 [=====] - 1s 2ms/step

Results for sigmoid function, 16 batch size and 20 epochs...

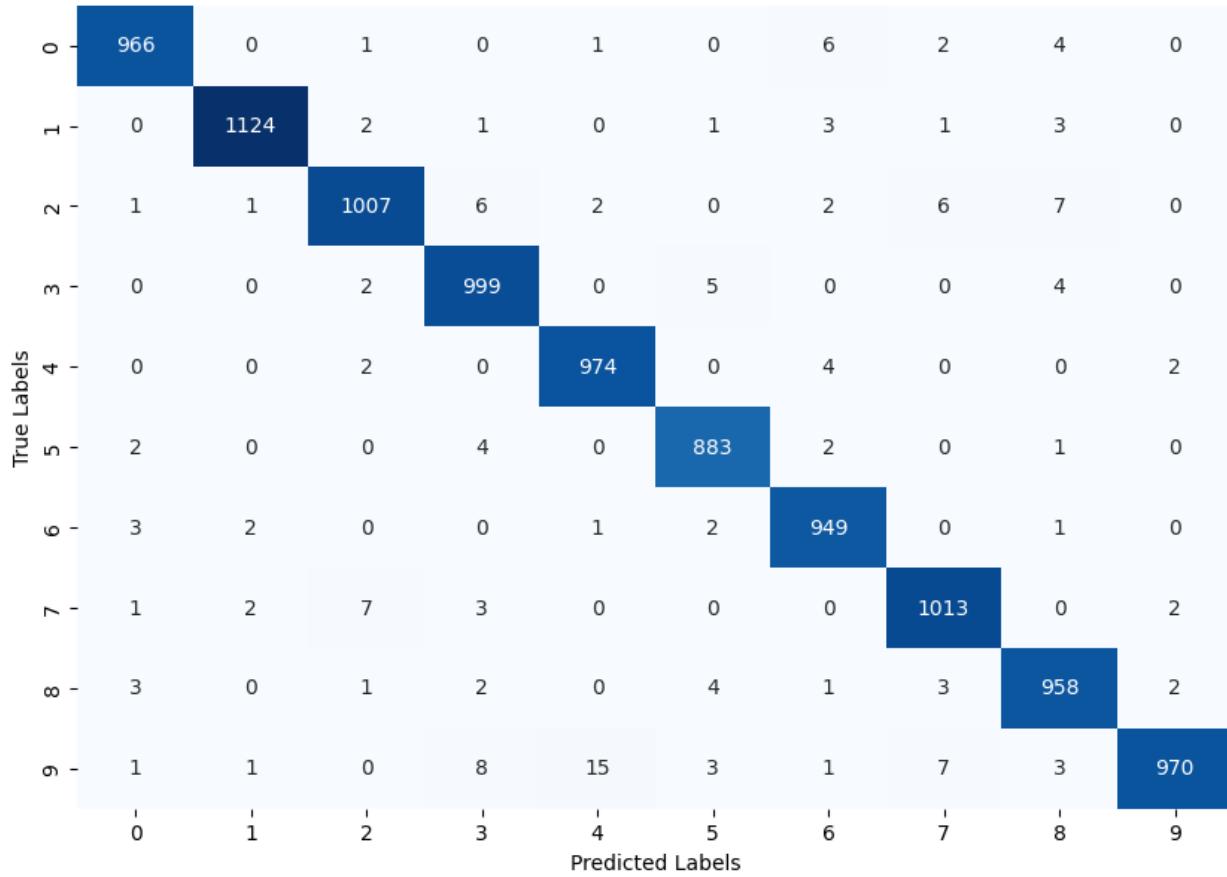
Confusion Matrix

```
[[ 966   0   1   0   1   0   6   2   4   0]
 [ 0 1124   2   1   0   1   3   1   3   0]
 [ 1   1 1007   6   2   0   2   6   7   0]
 [ 0   0   2 999   0   5   0   0   4   0]
 [ 0   0   2   0 974   0   4   0   0   2]
 [ 2   0   0   4   0 883   2   0   1   0]
 [ 3   2   0   0   1   2 949   0   1   0]
 [ 1   2   7   3   0   0   0 1013   0   2]
 [ 3   0   1   2   0   4   1   3 958   2]
 [ 1   1   0   8 15   3   1   7   3 970]]
```

Precision: 0.9844

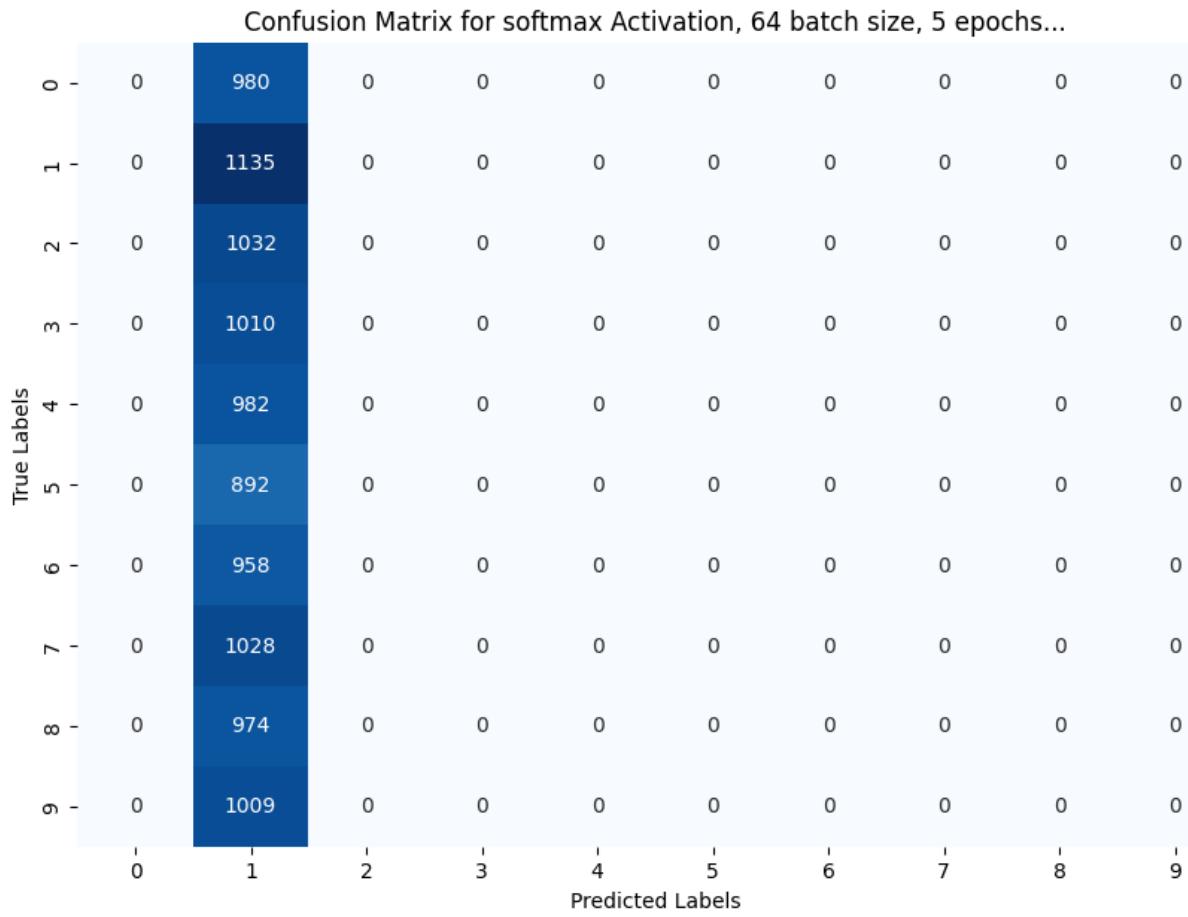
Recall: 0.9843

Confusion Matrix for sigmoid Activation, 16 batch size, 20 epochs...



```
Training model with softmax activation function, 64 batch sizes, 5
epochs_list
Epoch 1/5
844/844 - 7s - loss: 2.3013 - accuracy: 0.1124 - val_loss: 2.3009 -
val_accuracy: 0.1480 - 7s/epoch - 9ms/step
Epoch 2/5
844/844 - 7s - loss: 2.3006 - accuracy: 0.1133 - val_loss: 2.3011 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 3/5
844/844 - 7s - loss: 2.2999 - accuracy: 0.1132 - val_loss: 2.3001 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 4/5
844/844 - 7s - loss: 2.2986 - accuracy: 0.1132 - val_loss: 2.2983 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 5/5
844/844 - 7s - loss: 2.2955 - accuracy: 0.1140 - val_loss: 2.2934 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 64 batch size and 5 epochs...
Confusion Matrix
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with softmax activation function, 64 batch sizes, 15
epochs_list
Epoch 1/15
844/844 - 7s - loss: 2.3013 - accuracy: 0.1126 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 2/15
844/844 - 7s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 3/15
844/844 - 7s - loss: 2.3001 - accuracy: 0.1132 - val_loss: 2.3001 -
val_accuracy: 0.1050 - 7s/epoch - 9ms/step
Epoch 4/15
844/844 - 7s - loss: 2.2991 - accuracy: 0.1132 - val_loss: 2.2987 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 5/15
844/844 - 7s - loss: 2.2968 - accuracy: 0.1132 - val_loss: 2.2956 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 6/15
844/844 - 7s - loss: 2.2860 - accuracy: 0.1370 - val_loss: 2.2324 -
val_accuracy: 0.2275 - 7s/epoch - 8ms/step
Epoch 7/15
```

```
844/844 - 7s - loss: 2.0391 - accuracy: 0.2500 - val_loss: 1.8196 -  
val_accuracy: 0.3358 - 7s/epoch - 8ms/step  
Epoch 8/15  
844/844 - 7s - loss: 1.5272 - accuracy: 0.4257 - val_loss: 1.3195 -  
val_accuracy: 0.4523 - 7s/epoch - 8ms/step  
Epoch 9/15  
844/844 - 7s - loss: 1.2947 - accuracy: 0.4622 - val_loss: 1.2412 -  
val_accuracy: 0.4823 - 7s/epoch - 8ms/step  
Epoch 10/15  
844/844 - 7s - loss: 1.2276 - accuracy: 0.4923 - val_loss: 1.1780 -  
val_accuracy: 0.5003 - 7s/epoch - 8ms/step  
Epoch 11/15  
844/844 - 7s - loss: 1.1760 - accuracy: 0.5221 - val_loss: 1.1349 -  
val_accuracy: 0.5202 - 7s/epoch - 8ms/step  
Epoch 12/15  
844/844 - 7s - loss: 1.1408 - accuracy: 0.5464 - val_loss: 1.0951 -  
val_accuracy: 0.5850 - 7s/epoch - 8ms/step  
Epoch 13/15  
844/844 - 7s - loss: 1.1084 - accuracy: 0.5774 - val_loss: 1.0595 -  
val_accuracy: 0.5997 - 7s/epoch - 8ms/step  
Epoch 14/15  
844/844 - 7s - loss: 1.0811 - accuracy: 0.5948 - val_loss: 1.0304 -  
val_accuracy: 0.6178 - 7s/epoch - 8ms/step  
Epoch 15/15  
844/844 - 7s - loss: 1.0520 - accuracy: 0.6177 - val_loss: 0.9973 -  
val_accuracy: 0.6505 - 7s/epoch - 8ms/step  
313/313 [=====] - 1s 3ms/step  
Results for softmax function, 64 batch size and 15 epochs...  
Confusion Matrix  
[[ 893 1 16 26 2 31 1 0 7 3]  
[ 0 1120 6 2 0 1 0 0 6 0]  
[ 55 23 608 7 32 86 98 4 108 11]  
[ 677 2 16 187 0 32 3 6 66 21]  
[ 0 12 17 0 706 0 37 62 17 131]  
[ 445 7 60 184 5 81 6 0 91 13]  
[ 7 4 69 2 9 5 858 0 4 0]  
[ 1 12 4 7 9 3 1 884 23 84]  
[ 63 101 51 94 10 55 7 3 563 27]  
[ 8 9 3 19 48 2 2 412 22 484]]  
Precision: 0.6286  
Recall: 0.6384
```

Confusion Matrix for softmax Activation, 64 batch size, 15 epochs...

	0	1	2	3	4	5	6	7	8	9
True Labels	893	1	16	26	2	31	1	0	7	3
0	0	1120	6	2	0	1	0	0	6	0
1	55	23	608	7	32	86	98	4	108	11
2	677	2	16	187	0	32	3	6	66	21
3	0	12	17	0	706	0	37	62	17	131
4	445	7	60	184	5	81	6	0	91	13
5	7	4	69	2	9	5	858	0	4	0
6	1	12	4	7	9	3	1	884	23	84
7	63	101	51	94	10	55	7	3	563	27
8	8	9	3	19	48	2	2	412	22	484
9	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training model with softmax activation function, 64 batch sizes, 20
epochs_list
Epoch 1/20
844/844 - 7s - loss: 2.3011 - accuracy: 0.1120 - val_loss: 2.3008 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 2/20
844/844 - 7s - loss: 2.2996 - accuracy: 0.1139 - val_loss: 2.2990 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 3/20
844/844 - 7s - loss: 2.2968 - accuracy: 0.1132 - val_loss: 2.2949 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 4/20
844/844 - 7s - loss: 2.2896 - accuracy: 0.1200 - val_loss: 2.2760 -
val_accuracy: 0.1830 - 7s/epoch - 8ms/step
Epoch 5/20
844/844 - 7s - loss: 2.1320 - accuracy: 0.2283 - val_loss: 1.9887 -
val_accuracy: 0.2292 - 7s/epoch - 9ms/step
Epoch 6/20
844/844 - 7s - loss: 1.7326 - accuracy: 0.3575 - val_loss: 1.3734 -
val_accuracy: 0.4217 - 7s/epoch - 8ms/step
Epoch 7/20
```

```

844/844 - 7s - loss: 1.2888 - accuracy: 0.4354 - val_loss: 1.1966 -
val_accuracy: 0.4358 - 7s/epoch - 8ms/step
Epoch 8/20
844/844 - 7s - loss: 1.2070 - accuracy: 0.4512 - val_loss: 1.1607 -
val_accuracy: 0.4755 - 7s/epoch - 9ms/step
Epoch 9/20
844/844 - 7s - loss: 1.1767 - accuracy: 0.4674 - val_loss: 1.1333 -
val_accuracy: 0.4697 - 7s/epoch - 8ms/step
Epoch 10/20
844/844 - 7s - loss: 1.1554 - accuracy: 0.4885 - val_loss: 1.1171 -
val_accuracy: 0.5027 - 7s/epoch - 8ms/step
Epoch 11/20
844/844 - 7s - loss: 1.1337 - accuracy: 0.5143 - val_loss: 1.0911 -
val_accuracy: 0.5642 - 7s/epoch - 8ms/step
Epoch 12/20
844/844 - 7s - loss: 1.1013 - accuracy: 0.5518 - val_loss: 1.0512 -
val_accuracy: 0.6085 - 7s/epoch - 8ms/step
Epoch 13/20
844/844 - 7s - loss: 1.0482 - accuracy: 0.5974 - val_loss: 0.9959 -
val_accuracy: 0.6207 - 7s/epoch - 8ms/step
Epoch 14/20
844/844 - 7s - loss: 0.9881 - accuracy: 0.6249 - val_loss: 0.9306 -
val_accuracy: 0.6627 - 7s/epoch - 8ms/step
Epoch 15/20
844/844 - 7s - loss: 0.9412 - accuracy: 0.6460 - val_loss: 0.8880 -
val_accuracy: 0.7108 - 7s/epoch - 9ms/step
Epoch 16/20
844/844 - 7s - loss: 0.9042 - accuracy: 0.6588 - val_loss: 0.8567 -
val_accuracy: 0.7052 - 7s/epoch - 9ms/step
Epoch 17/20
844/844 - 8s - loss: 0.8748 - accuracy: 0.6681 - val_loss: 0.8247 -
val_accuracy: 0.7005 - 8s/epoch - 9ms/step
Epoch 18/20
844/844 - 7s - loss: 0.8496 - accuracy: 0.6772 - val_loss: 0.7993 -
val_accuracy: 0.7070 - 7s/epoch - 8ms/step
Epoch 19/20
844/844 - 7s - loss: 0.8287 - accuracy: 0.6852 - val_loss: 0.7749 -
val_accuracy: 0.7058 - 7s/epoch - 8ms/step
Epoch 20/20
844/844 - 7s - loss: 0.8055 - accuracy: 0.6981 - val_loss: 0.7626 -
val_accuracy: 0.7235 - 7s/epoch - 8ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 64 batch size and 20 epochs...
Confusion Matrix
[[ 925   0   28   0    9   0   17   1   0   0]
 [  0 1118   2   0    2   2   1   0   10   0]
 [ 52    7  674   5   28   41   19   2  190  14]
 [  0    0   3  913   0   55   0   1   20  18]
 [ 10    3   4   0  833   0   21   40   3  68]]
```

```
[ 0 0 25 748 1 74 0 0 37 7]
[ 732 1 67 1 5 0 149 0 3 0]
[ 1 4 1 0 7 2 1 938 9 651]
[ 11 21 187 78 4 71 5 2 561 34]
[ 2 7 5 33 30 11 1 20 7 893]]
```

Precision: 0.7079

Recall: 0.7078

Confusion Matrix for softmax Activation, 64 batch size, 20 epochs...

	0	1	2	3	4	5	6	7	8	9
0	925	0	28	0	9	0	17	1	0	0
1	0	1118	2	0	2	2	1	0	10	0
2	52	7	674	5	28	41	19	2	190	14
3	0	0	3	913	0	55	0	1	20	18
4	10	3	4	0	833	0	21	40	3	68
5	0	0	25	748	1	74	0	0	37	7
6	732	1	67	1	5	0	149	0	3	0
7	1	4	1	0	7	2	1	938	9	65
8	11	21	187	78	4	71	5	2	561	34
9	2	7	5	33	30	11	1	20	7	893
	0	1	2	3	4	5	6	7	8	9

Training model with softmax activation function, 128 batch sizes, 5 epochs list

Epoch 1/5

422/422 - 6s - loss: 2.3013 - accuracy: 0.1125 - val_loss: 2.3016 - val_accuracy: 0.1050 - 6s/epoch - 13ms/step

Epoch 2/5

422/422 - 5s - loss: 2.3006 - accuracy: 0.1132 - val_loss: 2.3011 - val_accuracy: 0.1050 - 5s/epoch - 12ms/step

Epoch 3/5

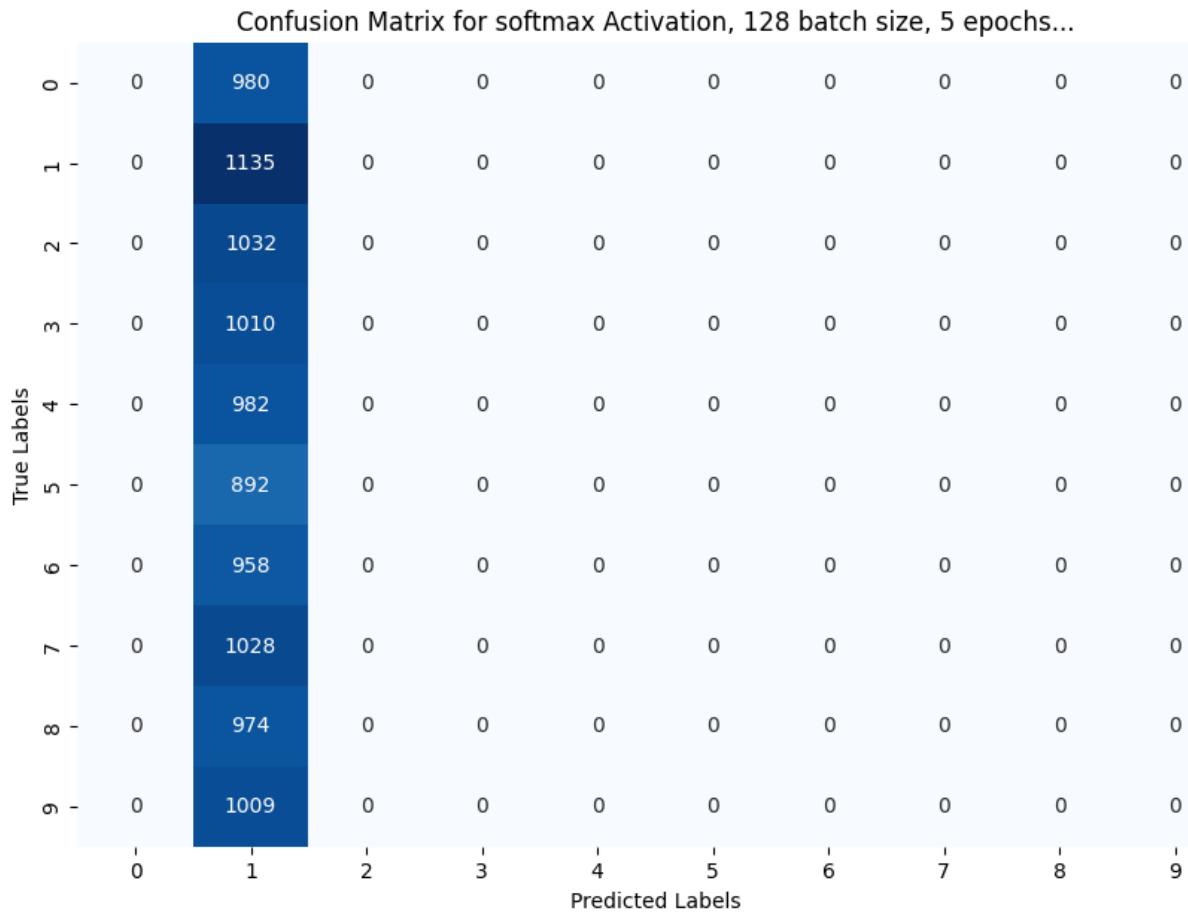
```
422/422 - 5s - loss: 2.3002 - accuracy: 0.1132 - val_loss: 2.3003 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
```

Epoch 4/5
422/422

422/422 - 5s - loss: 2.2997 - accuracy: 0.1132 - val_loss: 2.2998 -

```
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 5/5
422/422 - 5s - loss: 2.2989 - accuracy: 0.1132 - val_loss: 2.2992 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 128 batch size and 5 epochs...
Confusion Matrix
[[ 0 980 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with softmax activation function, 128 batch sizes, 15
epochs_list
Epoch 1/15
422/422 - 5s - loss: 2.3014 - accuracy: 0.1125 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 2/15
422/422 - 5s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3014 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 3/15
422/422 - 5s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3011 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 4/15
422/422 - 5s - loss: 2.3002 - accuracy: 0.1132 - val_loss: 2.3007 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 5/15
422/422 - 5s - loss: 2.2998 - accuracy: 0.1132 - val_loss: 2.3005 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 6/15
422/422 - 5s - loss: 2.2992 - accuracy: 0.1132 - val_loss: 2.2993 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 7/15
```

```

422/422 - 5s - loss: 2.2982 - accuracy: 0.1132 - val_loss: 2.2980 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 8/15
422/422 - 5s - loss: 2.2967 - accuracy: 0.1132 - val_loss: 2.2960 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 9/15
422/422 - 5s - loss: 2.2941 - accuracy: 0.1132 - val_loss: 2.2927 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 10/15
422/422 - 5s - loss: 2.2894 - accuracy: 0.1132 - val_loss: 2.2849 -
val_accuracy: 0.1067 - 5s/epoch - 11ms/step
Epoch 11/15
422/422 - 5s - loss: 2.2358 - accuracy: 0.2041 - val_loss: 2.1349 -
val_accuracy: 0.2133 - 5s/epoch - 12ms/step
Epoch 12/15
422/422 - 5s - loss: 2.0434 - accuracy: 0.2402 - val_loss: 1.9551 -
val_accuracy: 0.2562 - 5s/epoch - 11ms/step
Epoch 13/15
422/422 - 5s - loss: 1.8672 - accuracy: 0.3198 - val_loss: 1.7053 -
val_accuracy: 0.3370 - 5s/epoch - 12ms/step
Epoch 14/15
422/422 - 5s - loss: 1.6329 - accuracy: 0.3387 - val_loss: 1.5632 -
val_accuracy: 0.3448 - 5s/epoch - 11ms/step
Epoch 15/15
422/422 - 5s - loss: 1.5518 - accuracy: 0.3469 - val_loss: 1.5148 -
val_accuracy: 0.3515 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 128 batch size and 15 epochs...
Confusion Matrix
[[ 950   0   0   19   0   0   0   1   9   1]
 [  8 1112   2   4   0   0   2   0   7   0]
 [ 860   17   5   46   8   0   19   17   49   11]
 [ 796   9   1   72   17   0   8   20   78   9]
 [ 11   7   0   11   28   0   0   831   45   49]
 [ 549   22   3  125   26   0   12   25  117   13]
 [ 861   7  10   37   4   0   14   3   20   2]
 [  8   27   0   8   21   0   1  916   21   26]
 [ 281   46   3  157   73   0   5   74  287   48]
 [ 17   10   0   8  10   0   0  935   11   18]]
Precision: 0.2820
Recall: 0.3402

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result)))

```

Confusion Matrix for softmax Activation, 128 batch size, 15 epochs...

	0	1	2	3	4	5	6	7	8	9	10
True Labels	950	0	0	19	0	0	0	1	9	1	0
0	950	0	0	19	0	0	0	1	9	1	0
1	8	1112	2	4	0	0	2	0	7	0	0
2	860	17	5	46	8	0	19	17	49	11	0
3	796	9	1	72	17	0	8	20	78	9	0
4	11	7	0	11	28	0	0	831	45	49	0
5	549	22	3	125	26	0	12	25	117	13	0
6	861	7	10	37	4	0	14	3	20	2	0
7	8	27	0	8	21	0	1	916	21	26	0
8	281	46	3	157	73	0	5	74	287	48	0
9	17	10	0	8	10	0	0	935	11	18	0
10	0	1	2	3	4	5	6	7	8	9	0
Predicted Labels											

```
Training model with softmax activation function, 128 batch sizes, 20 epochs_list
Epoch 1/20
422/422 - 5s - loss: 2.3013 - accuracy: 0.1123 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 2/20
422/422 - 5s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 3/20
422/422 - 5s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 4/20
422/422 - 5s - loss: 2.3001 - accuracy: 0.1132 - val_loss: 2.3006 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 5/20
422/422 - 5s - loss: 2.2997 - accuracy: 0.1174 - val_loss: 2.3006 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 6/20
422/422 - 5s - loss: 2.2992 - accuracy: 0.1132 - val_loss: 2.2996 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 7/20
```

```

422/422 - 5s - loss: 2.2987 - accuracy: 0.1132 - val_loss: 2.2990 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 8/20
422/422 - 5s - loss: 2.2979 - accuracy: 0.1132 - val_loss: 2.2979 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 9/20
422/422 - 5s - loss: 2.2970 - accuracy: 0.1132 - val_loss: 2.2971 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 10/20
422/422 - 5s - loss: 2.2957 - accuracy: 0.1132 - val_loss: 2.2954 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 11/20
422/422 - 5s - loss: 2.2939 - accuracy: 0.1132 - val_loss: 2.2931 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 12/20
422/422 - 5s - loss: 2.2906 - accuracy: 0.1132 - val_loss: 2.2873 -
val_accuracy: 0.1053 - 5s/epoch - 12ms/step
Epoch 13/20
422/422 - 5s - loss: 2.2426 - accuracy: 0.1825 - val_loss: 2.1484 -
val_accuracy: 0.2147 - 5s/epoch - 12ms/step
Epoch 14/20
422/422 - 5s - loss: 2.0025 - accuracy: 0.2870 - val_loss: 1.8276 -
val_accuracy: 0.3333 - 5s/epoch - 11ms/step
Epoch 15/20
422/422 - 5s - loss: 1.7487 - accuracy: 0.3315 - val_loss: 1.6562 -
val_accuracy: 0.3293 - 5s/epoch - 11ms/step
Epoch 16/20
422/422 - 5s - loss: 1.6311 - accuracy: 0.3328 - val_loss: 1.5804 -
val_accuracy: 0.3333 - 5s/epoch - 11ms/step
Epoch 17/20
422/422 - 5s - loss: 1.5744 - accuracy: 0.3375 - val_loss: 1.5410 -
val_accuracy: 0.3420 - 5s/epoch - 11ms/step
Epoch 18/20
422/422 - 5s - loss: 1.5427 - accuracy: 0.3390 - val_loss: 1.5183 -
val_accuracy: 0.3482 - 5s/epoch - 11ms/step
Epoch 19/20
422/422 - 5s - loss: 1.5226 - accuracy: 0.3450 - val_loss: 1.5017 -
val_accuracy: 0.3620 - 5s/epoch - 11ms/step
Epoch 20/20
422/422 - 5s - loss: 1.5063 - accuracy: 0.3534 - val_loss: 1.4878 -
val_accuracy: 0.3517 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 128 batch size and 20 epochs...
Confusion Matrix
[[ 930   0   0   34   2   0   0   1   13   0]
 [  7 1115   0   3   1   0   4   1   4   0]
 [ 797  16   0   79  23   0   40  19  58   0]
 [ 811   3   0   92  13   0   12   8  71   0]
 [  5   6   0    7  74   0   0  858  32   0]]
```

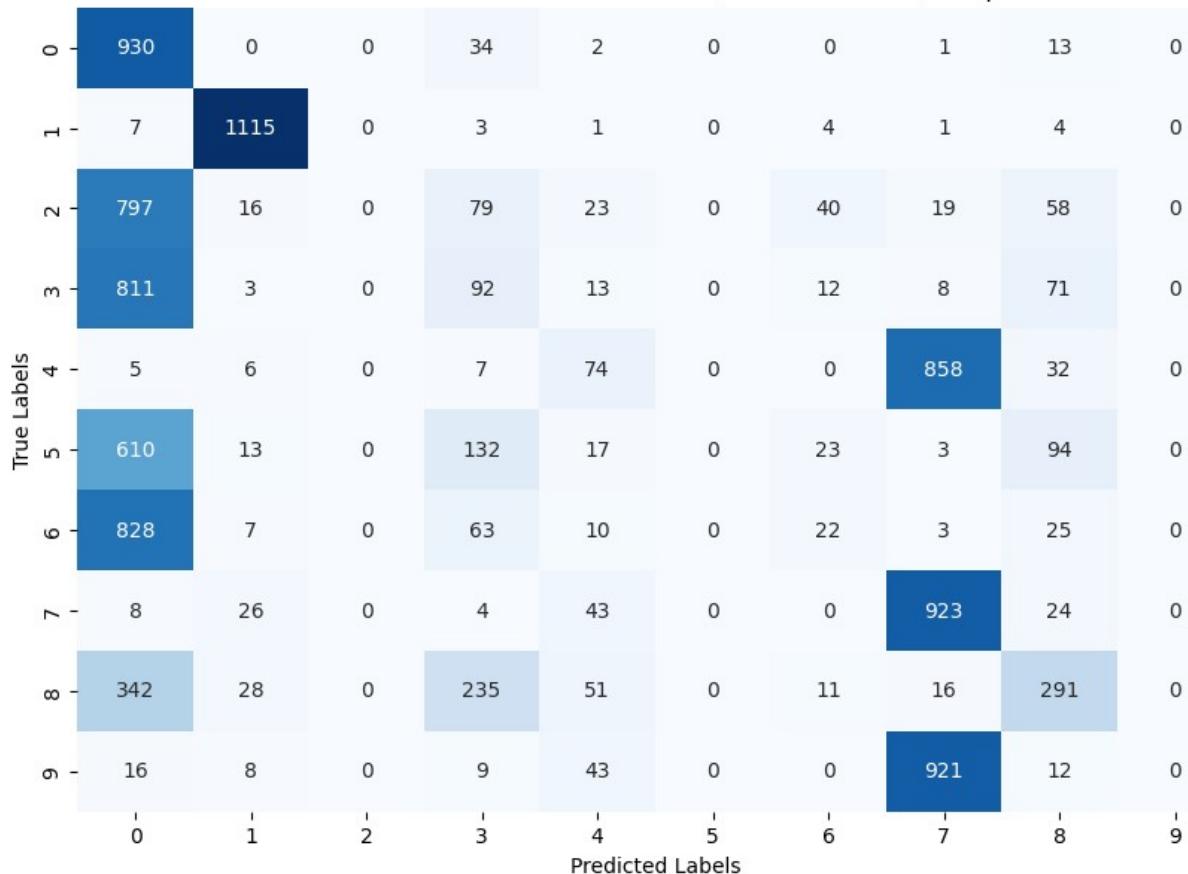
```
[ 610   13    0  132   17    0   23    3   94    0]
[ 828     7    0   63   10    0   22    3   25    0]
[  8   26    0    4   43    0    0  923   24    0]
[ 342   28    0  235   51    0   11   16  291    0]
[ 16     8    0    9   43    0    0  921   12    0]]
```

Precision: 0.2636

Recall: 0.3447

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation, 128 batch size, 20 epochs...



Training model with softmax activation function, 256 batch sizes, 5 epochs_list

Epoch 1/5

211/211 - 5s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3016 - val_accuracy: 0.1050 - 5s/epoch - 22ms/step

Epoch 2/5

211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -

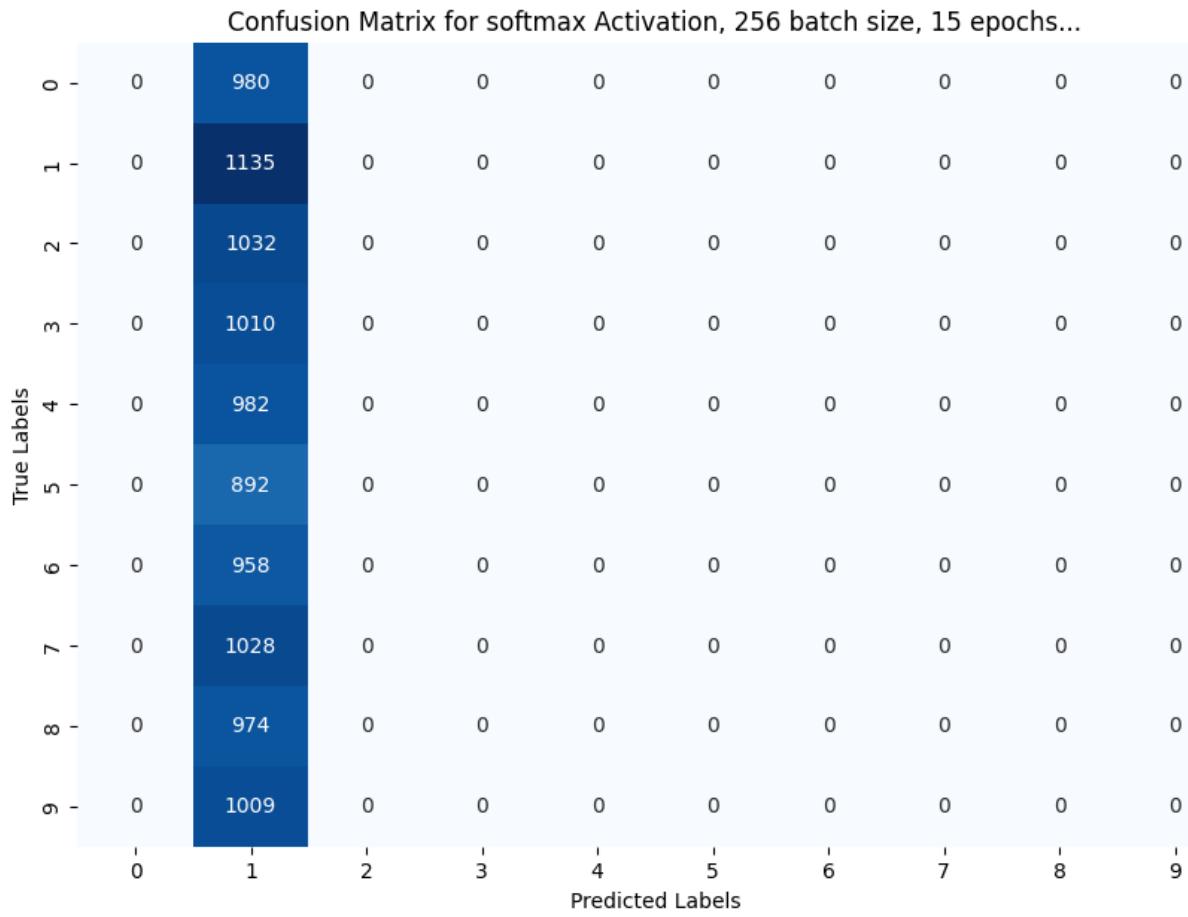
```
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 3/5
211/211 - 4s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3014 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 4/5
211/211 - 4s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 5/5
211/211 - 4s - loss: 2.3003 - accuracy: 0.1132 - val_loss: 2.3010 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 256 batch size and 5 epochs...
Confusion Matrix
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation, 256 batch size, 5 epochs...

```
Training model with softmax activation function, 256 batch sizes, 15 epochs_list
Epoch 1/15
211/211 - 4s - loss: 2.3017 - accuracy: 0.1099 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 2/15
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 3/15
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 4/15
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 5/15
211/211 - 4s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 6/15
211/211 - 4s - loss: 2.3006 - accuracy: 0.1132 - val_loss: 2.3014 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 7/15
```

```
211/211 - 4s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3012 -  
val_accuracy: 0.1050 - 4s/epoch - 17ms/step  
Epoch 8/15  
211/211 - 4s - loss: 2.3003 - accuracy: 0.1132 - val_loss: 2.3010 -  
val_accuracy: 0.1050 - 4s/epoch - 17ms/step  
Epoch 9/15  
211/211 - 4s - loss: 2.3002 - accuracy: 0.1132 - val_loss: 2.3008 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 10/15  
211/211 - 4s - loss: 2.3001 - accuracy: 0.1132 - val_loss: 2.3009 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 11/15  
211/211 - 4s - loss: 2.2999 - accuracy: 0.1132 - val_loss: 2.3006 -  
val_accuracy: 0.1050 - 4s/epoch - 17ms/step  
Epoch 12/15  
211/211 - 4s - loss: 2.2997 - accuracy: 0.1132 - val_loss: 2.3004 -  
val_accuracy: 0.1050 - 4s/epoch - 17ms/step  
Epoch 13/15  
211/211 - 4s - loss: 2.2996 - accuracy: 0.1132 - val_loss: 2.3001 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 14/15  
211/211 - 4s - loss: 2.2994 - accuracy: 0.1132 - val_loss: 2.3000 -  
val_accuracy: 0.1050 - 4s/epoch - 17ms/step  
Epoch 15/15  
211/211 - 4s - loss: 2.2992 - accuracy: 0.1132 - val_loss: 2.2998 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
313/313 [=====] - 1s 3ms/step  
Results for softmax function, 256 batch size and 15 epochs...  
Confusion Matrix  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with softmax activation function, 256 batch sizes, 20
epochs_list
Epoch 1/20
211/211 - 5s - loss: 2.3015 - accuracy: 0.1101 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 2/20
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 3/20
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 4/20
211/211 - 4s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3014 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 5/20
211/211 - 4s - loss: 2.3006 - accuracy: 0.1132 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 6/20
211/211 - 4s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 7/20
```

```

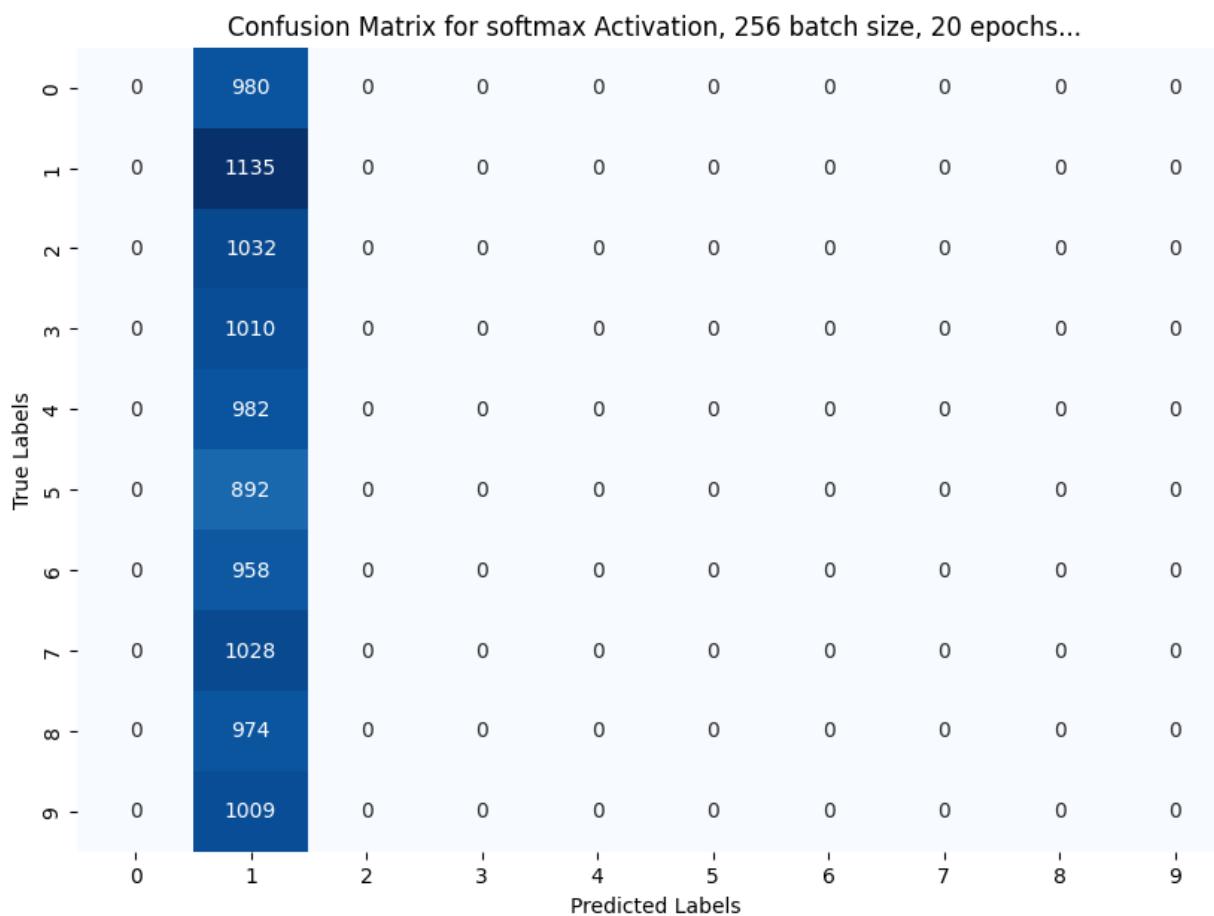
211/211 - 4s - loss: 2.3004 - accuracy: 0.1132 - val_loss: 2.3010 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 8/20
211/211 - 4s - loss: 2.3002 - accuracy: 0.1132 - val_loss: 2.3007 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 9/20
211/211 - 4s - loss: 2.3001 - accuracy: 0.1132 - val_loss: 2.3006 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 10/20
211/211 - 4s - loss: 2.2999 - accuracy: 0.1132 - val_loss: 2.3005 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 11/20
211/211 - 4s - loss: 2.2997 - accuracy: 0.1132 - val_loss: 2.3004 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 12/20
211/211 - 4s - loss: 2.2995 - accuracy: 0.1132 - val_loss: 2.3000 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 13/20
211/211 - 4s - loss: 2.2992 - accuracy: 0.1132 - val_loss: 2.2998 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 14/20
211/211 - 4s - loss: 2.2990 - accuracy: 0.1132 - val_loss: 2.2995 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 15/20
211/211 - 4s - loss: 2.2986 - accuracy: 0.1132 - val_loss: 2.2991 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 16/20
211/211 - 4s - loss: 2.2983 - accuracy: 0.1132 - val_loss: 2.2987 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 17/20
211/211 - 4s - loss: 2.2979 - accuracy: 0.1132 - val_loss: 2.2984 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 18/20
211/211 - 4s - loss: 2.2974 - accuracy: 0.1132 - val_loss: 2.2978 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 19/20
211/211 - 4s - loss: 2.2968 - accuracy: 0.1132 - val_loss: 2.2970 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 20/20
211/211 - 4s - loss: 2.2960 - accuracy: 0.1132 - val_loss: 2.2961 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 256 batch size and 20 epochs...
Confusion Matrix
[[ 0 980 0 0 0 0 0 0 0],
 [ 0 1135 0 0 0 0 0 0 0],
 [ 0 1032 0 0 0 0 0 0 0],
 [ 0 1010 0 0 0 0 0 0 0],
 [ 0 982 0 0 0 0 0 0 0]]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training model with softmax activation function, 16 batch sizes, 5 epochs_list

Epoch 1/5

3375/3375 - 19s - loss: 2.3005 - accuracy: 0.1134 - val_loss: 2.2966 - val_accuracy: 0.1050 - 19s/epoch - 6ms/step

Epoch 2/5

3375/3375 - 19s - loss: 1.9190 - accuracy: 0.2634 - val_loss: 1.5225 -

```
val_accuracy: 0.3487 - 19s/epoch - 6ms/step
Epoch 3/5
3375/3375 - 19s - loss: 1.4936 - accuracy: 0.3637 - val_loss: 1.4271 -
val_accuracy: 0.4055 - 19s/epoch - 6ms/step
Epoch 4/5
3375/3375 - 18s - loss: 1.3432 - accuracy: 0.4443 - val_loss: 1.1150 -
val_accuracy: 0.5502 - 18s/epoch - 5ms/step
Epoch 5/5
3375/3375 - 19s - loss: 1.0614 - accuracy: 0.5933 - val_loss: 0.9302 -
val_accuracy: 0.7028 - 19s/epoch - 6ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 16 batch size and 5 epochs...
Confusion Matrix
[[ 921    0   15   37    0    2    1    0    2    2]
 [  0 1114    6    1    0    1    0    0   13    0]
 [ 85    2  662   91   13   46   82    1   42    8]
 [ 310    1   20  575    1   49    0    4   39   11]
 [  1   15   20    0  466    0   30  111   15  324]
 [ 32   14   40  271    1  153    5    0  365   11]
 [  7    4   74    3    5    2  859    0    4    0]
 [  1    6    6   13   13    2    0  826   15  146]
 [  6  137   15   55    7   48    3    0  683   20]
 [  6    8    5   20   35    7    1  280   18  629]]
```

Precision: 0.6895
Recall: 0.6888

Confusion Matrix for softmax Activation, 16 batch size, 5 epochs...

	0	1	2	3	4	5	6	7	8	9
True Labels	921	0	15	37	0	2	1	0	2	2
0	921	0	15	37	0	2	1	0	2	2
1	0	1114	6	1	0	1	0	0	13	0
2	85	2	662	91	13	46	82	1	42	8
3	310	1	20	575	1	49	0	4	39	11
4	1	15	20	0	466	0	30	111	15	324
5	32	14	40	271	1	153	5	0	365	11
6	7	4	74	3	5	2	859	0	4	0
7	1	6	6	13	13	2	0	826	15	146
8	6	137	15	55	7	48	3	0	683	20
9	6	8	5	20	35	7	1	280	18	629
0	1	2	3	4	5	6	7	8	9	
Predicted Labels										

```
Training model with softmax activation function, 16 batch sizes, 15
epochs_list
Epoch 1/15
3375/3375 - 19s - loss: 2.3003 - accuracy: 0.1119 - val_loss: 2.2944 -
val_accuracy: 0.1050 - 19s/epoch - 6ms/step
Epoch 2/15
3375/3375 - 19s - loss: 1.7816 - accuracy: 0.2982 - val_loss: 1.4637 -
val_accuracy: 0.3748 - 19s/epoch - 6ms/step
Epoch 3/15
3375/3375 - 18s - loss: 1.3855 - accuracy: 0.3970 - val_loss: 1.2870 -
val_accuracy: 0.4160 - 18s/epoch - 5ms/step
Epoch 4/15
3375/3375 - 19s - loss: 1.2933 - accuracy: 0.4233 - val_loss: 1.2349 -
val_accuracy: 0.4283 - 19s/epoch - 6ms/step
Epoch 5/15
3375/3375 - 20s - loss: 1.2173 - accuracy: 0.4635 - val_loss: 1.1681 -
val_accuracy: 0.4903 - 20s/epoch - 6ms/step
Epoch 6/15
3375/3375 - 19s - loss: 1.1555 - accuracy: 0.5129 - val_loss: 1.0866 -
val_accuracy: 0.5208 - 19s/epoch - 6ms/step
Epoch 7/15
```

```
3375/3375 - 19s - loss: 1.0921 - accuracy: 0.5601 - val_loss: 1.0241 -  
val_accuracy: 0.6138 - 19s/epoch - 6ms/step  
Epoch 8/15  
3375/3375 - 19s - loss: 1.0365 - accuracy: 0.5914 - val_loss: 0.9587 -  
val_accuracy: 0.6430 - 19s/epoch - 6ms/step  
Epoch 9/15  
3375/3375 - 19s - loss: 0.9872 - accuracy: 0.6120 - val_loss: 0.9452 -  
val_accuracy: 0.6153 - 19s/epoch - 6ms/step  
Epoch 10/15  
3375/3375 - 19s - loss: 0.9512 - accuracy: 0.6211 - val_loss: 0.8738 -  
val_accuracy: 0.6590 - 19s/epoch - 6ms/step  
Epoch 11/15  
3375/3375 - 19s - loss: 0.9134 - accuracy: 0.6333 - val_loss: 0.8555 -  
val_accuracy: 0.6533 - 19s/epoch - 6ms/step  
Epoch 12/15  
3375/3375 - 18s - loss: 0.8784 - accuracy: 0.6514 - val_loss: 0.8506 -  
val_accuracy: 0.6785 - 18s/epoch - 5ms/step  
Epoch 13/15  
3375/3375 - 19s - loss: 0.8413 - accuracy: 0.6788 - val_loss: 0.7725 -  
val_accuracy: 0.6880 - 19s/epoch - 6ms/step  
Epoch 14/15  
3375/3375 - 19s - loss: 0.8017 - accuracy: 0.7001 - val_loss: 0.7260 -  
val_accuracy: 0.7445 - 19s/epoch - 6ms/step  
Epoch 15/15  
3375/3375 - 20s - loss: 0.7677 - accuracy: 0.7270 - val_loss: 0.7083 -  
val_accuracy: 0.7548 - 20s/epoch - 6ms/step  
313/313 [=====] - 1s 3ms/step  
Results for softmax function, 16 batch size and 15 epochs...  
Confusion Matrix  
[[ 944  0  1  0  10  0  15  0  9  1]  
[  0 1076 10  9  0   4  0   0  0  36]  
[ 16  37 482 24 33 186  2   2 136 114]  
[  0  23  3 943  0  36  0   1  2  2]  
[  8  0  7  0 838  0  1  23  2 103]  
[  2  3 16 240  0 567  5   0  58  1]  
[ 675  1  6  0  6  1 183  0  86  0]  
[  0 18  0  0 12  1  0 940  0  57]  
[ 16  1 139  3  7 77  93  0 625 13]  
[  1 52 43  2 60 13  0  25  9 804]]  
Precision: 0.7407  
Recall: 0.7402
```

Confusion Matrix for softmax Activation, 16 batch size, 15 epochs...

	0	1	2	3	4	5	6	7	8	9
0	944	0	1	0	10	0	15	0	9	1
1	0	1076	10	9	0	4	0	0	0	36
2	16	37	482	24	33	186	2	2	136	114
3	0	23	3	943	0	36	0	1	2	2
4	8	0	7	0	838	0	1	23	2	103
5	2	3	16	240	0	567	5	0	58	1
6	675	1	6	0	6	1	183	0	86	0
7	0	18	0	0	12	1	0	940	0	57
8	16	1	139	3	7	77	93	0	625	13
9	1	52	43	2	60	13	0	25	9	804
	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```
Training model with softmax activation function, 16 batch sizes, 20
epochs_list
Epoch 1/20
3375/3375 - 19s - loss: 2.3019 - accuracy: 0.1116 - val_loss: 2.3007 -
val_accuracy: 0.1050 - 19s/epoch - 6ms/step
Epoch 2/20
3375/3375 - 19s - loss: 2.2576 - accuracy: 0.1659 - val_loss: 1.8646 -
val_accuracy: 0.4437 - 19s/epoch - 6ms/step
Epoch 3/20
3375/3375 - 18s - loss: 1.2806 - accuracy: 0.4983 - val_loss: 1.0986 -
val_accuracy: 0.5578 - 18s/epoch - 5ms/step
Epoch 4/20
3375/3375 - 19s - loss: 1.0790 - accuracy: 0.5783 - val_loss: 0.9746 -
val_accuracy: 0.6352 - 19s/epoch - 6ms/step
Epoch 5/20
3375/3375 - 18s - loss: 0.9933 - accuracy: 0.6333 - val_loss: 0.9042 -
val_accuracy: 0.6657 - 18s/epoch - 5ms/step
Epoch 6/20
3375/3375 - 19s - loss: 0.9238 - accuracy: 0.6740 - val_loss: 0.8162 -
val_accuracy: 0.7395 - 19s/epoch - 6ms/step
Epoch 7/20
```

```

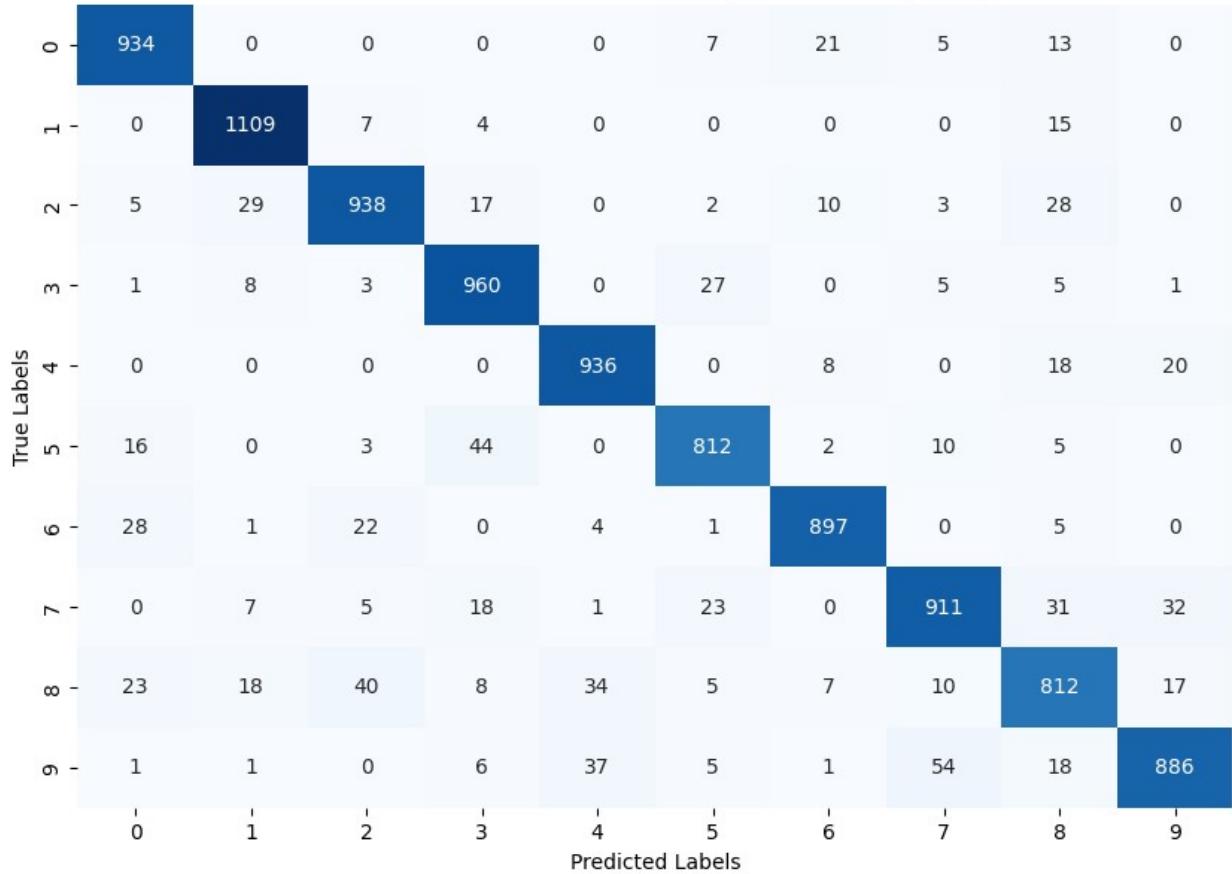
3375/3375 - 19s - loss: 0.8277 - accuracy: 0.7352 - val_loss: 0.7077 -
val_accuracy: 0.7993 - 19s/epoch - 6ms/step
Epoch 8/20
3375/3375 - 19s - loss: 0.7386 - accuracy: 0.7795 - val_loss: 0.6344 -
val_accuracy: 0.8415 - 19s/epoch - 6ms/step
Epoch 9/20
3375/3375 - 19s - loss: 0.6740 - accuracy: 0.8070 - val_loss: 0.6429 -
val_accuracy: 0.7795 - 19s/epoch - 6ms/step
Epoch 10/20
3375/3375 - 19s - loss: 0.6076 - accuracy: 0.8376 - val_loss: 0.5292 -
val_accuracy: 0.8665 - 19s/epoch - 6ms/step
Epoch 11/20
3375/3375 - 19s - loss: 0.5471 - accuracy: 0.8600 - val_loss: 0.4836 -
val_accuracy: 0.8868 - 19s/epoch - 6ms/step
Epoch 12/20
3375/3375 - 19s - loss: 0.5008 - accuracy: 0.8754 - val_loss: 0.4482 -
val_accuracy: 0.8997 - 19s/epoch - 6ms/step
Epoch 13/20
3375/3375 - 19s - loss: 0.4712 - accuracy: 0.8820 - val_loss: 0.4474 -
val_accuracy: 0.8860 - 19s/epoch - 6ms/step
Epoch 14/20
3375/3375 - 18s - loss: 0.4418 - accuracy: 0.8890 - val_loss: 0.3848 -
val_accuracy: 0.9100 - 18s/epoch - 5ms/step
Epoch 15/20
3375/3375 - 19s - loss: 0.4218 - accuracy: 0.8936 - val_loss: 0.3964 -
val_accuracy: 0.9105 - 19s/epoch - 6ms/step
Epoch 16/20
3375/3375 - 19s - loss: 0.4070 - accuracy: 0.8958 - val_loss: 0.3768 -
val_accuracy: 0.9078 - 19s/epoch - 6ms/step
Epoch 17/20
3375/3375 - 19s - loss: 0.3951 - accuracy: 0.8966 - val_loss: 0.3817 -
val_accuracy: 0.9018 - 19s/epoch - 6ms/step
Epoch 18/20
3375/3375 - 19s - loss: 0.3818 - accuracy: 0.8988 - val_loss: 0.3542 -
val_accuracy: 0.9092 - 19s/epoch - 6ms/step
Epoch 19/20
3375/3375 - 19s - loss: 0.3667 - accuracy: 0.9041 - val_loss: 0.4361 -
val_accuracy: 0.8775 - 19s/epoch - 6ms/step
Epoch 20/20
3375/3375 - 19s - loss: 0.3498 - accuracy: 0.9094 - val_loss: 0.3222 -
val_accuracy: 0.9260 - 19s/epoch - 6ms/step
313/313 [=====] - 1s 3ms/step
Results for softmax function, 16 batch size and 20 epochs...
Confusion Matrix
[[ 934   0   0   0   0    7   21    5   13   0]
 [  0 1109   7   4   0    0    0   15   0]
 [  5   29  938  17   0    2   10    3   28   0]
 [  1   8   3  960   0   27    0    5    5   1]
 [  0   0   0   0  936   0    8    0   18   20]]
```

```
[ 16   0   3   44   0  812   2   10   5   0
[ 28   1  22   0   4   1  897   0   5   0
[  0   7   5  18   1  23   0  911  31  32
[ 23  18  40   8  34   5   7  10  812  17
[  1   1   0   6  37   5   1  54  18  886]]
```

Precision: 0.9193

Recall: 0.9195

Confusion Matrix for softmax Activation, 16 batch size, 20 epochs...



Training model with exponential activation function, 64 batch sizes, 5 epochs_list

Epoch 1/5

844/844 - 6s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 6s/epoch - 8ms/step

Epoch 2/5

844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 5s/epoch - 6ms/step

Epoch 3/5

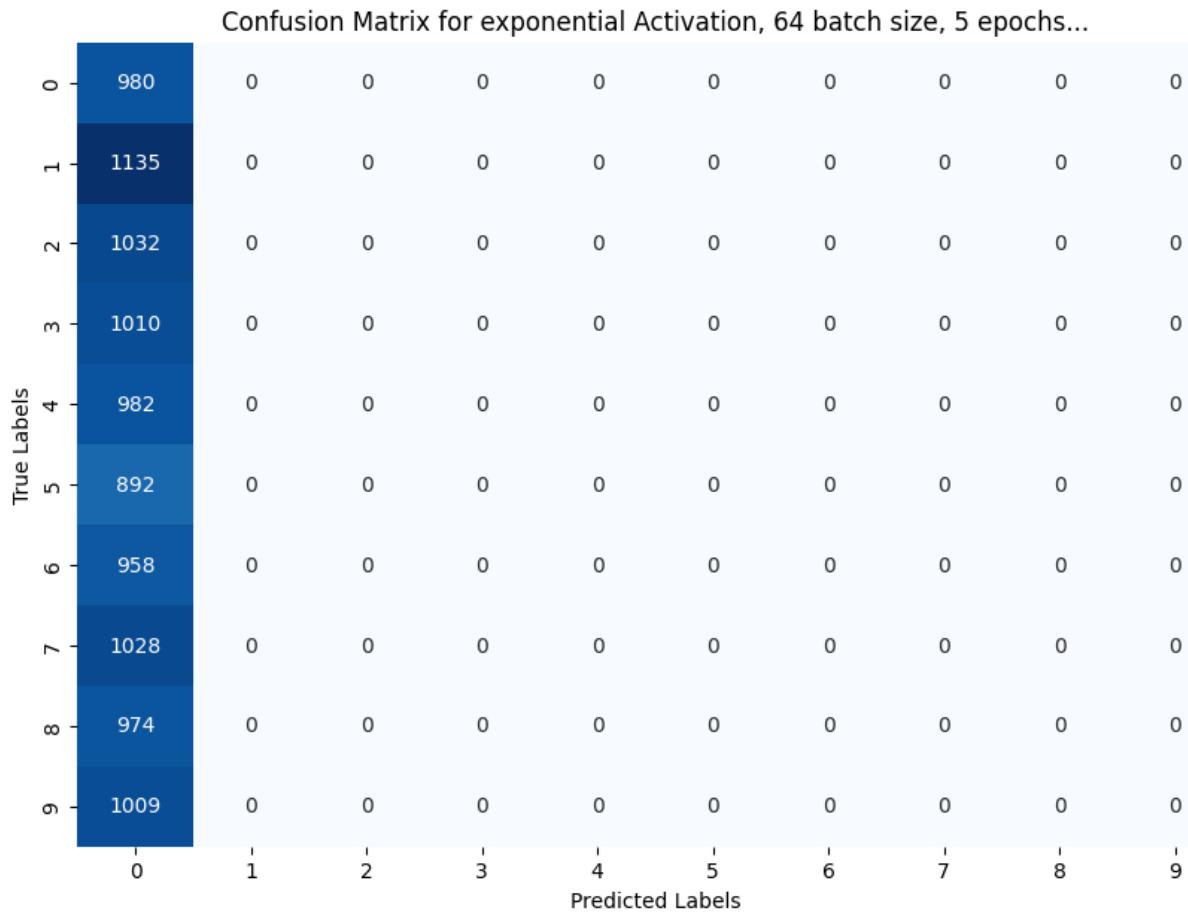
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan - val_accuracy: 0.0978 - 5s/epoch - 6ms/step

Epoch 4/5

844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -

```
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for exponential function, 64 batch size and 5 epochs...
Confusion Matrix
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
Precision: 0.0096
Recall: 0.0980

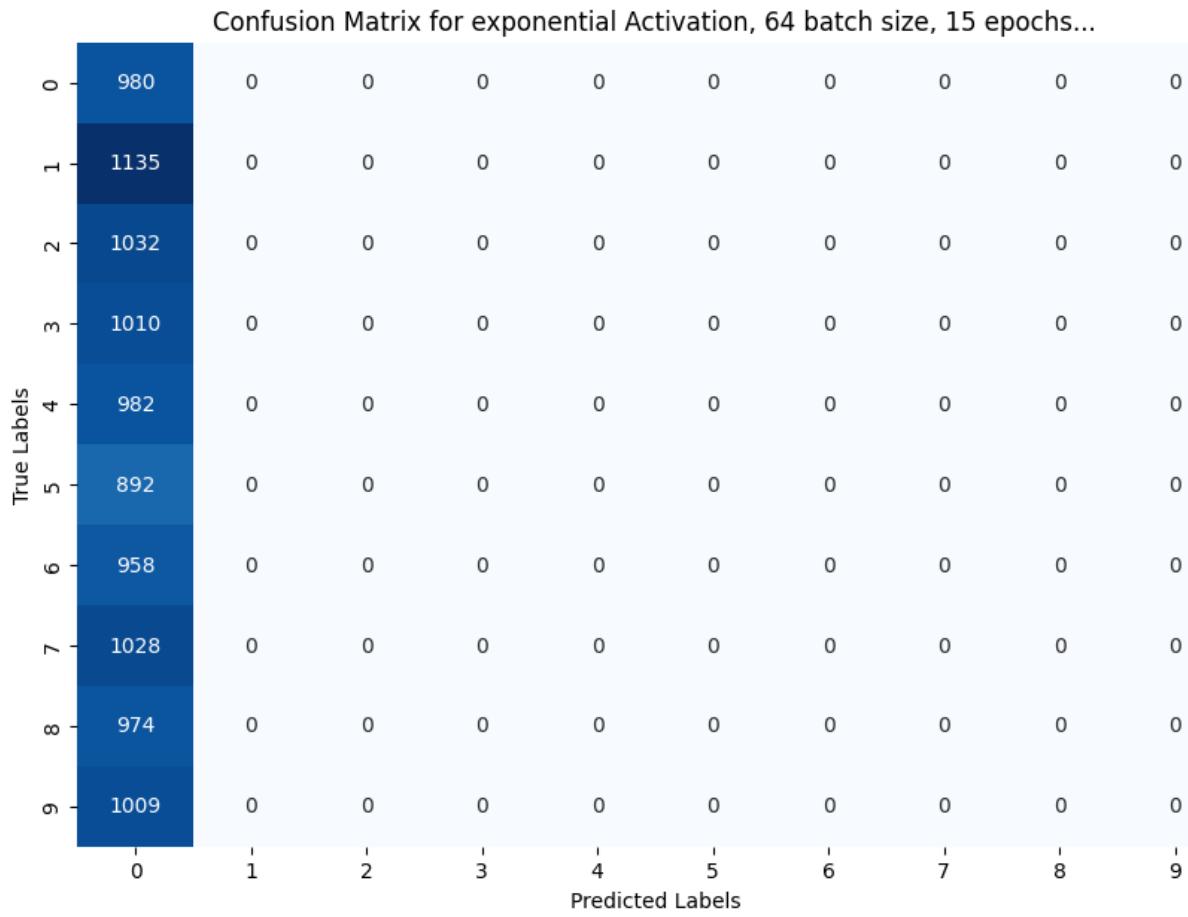
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with exponential activation function, 64 batch sizes,
15 epochs_list
Epoch 1/15
844/844 - 6s - loss: nan - accuracy: 0.0989 - val_loss: nan -
val_accuracy: 0.0978 - 6s/epoch - 7ms/step
Epoch 2/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 3/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 4/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 5/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 6/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 7/15
```

```
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 8/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 9/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 10/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 11/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 12/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 13/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 14/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 15/15
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for exponential function, 64 batch size and 15 epochs...
Confusion Matrix
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
Precision: 0.0096
Recall: 0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with exponential activation function, 64 batch sizes,
20 epochs_list
Epoch 1/20
844/844 - 6s - loss: nan - accuracy: 0.0989 - val_loss: nan -
val_accuracy: 0.0978 - 6s/epoch - 7ms/step
Epoch 2/20
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 3/20
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 7/20
```

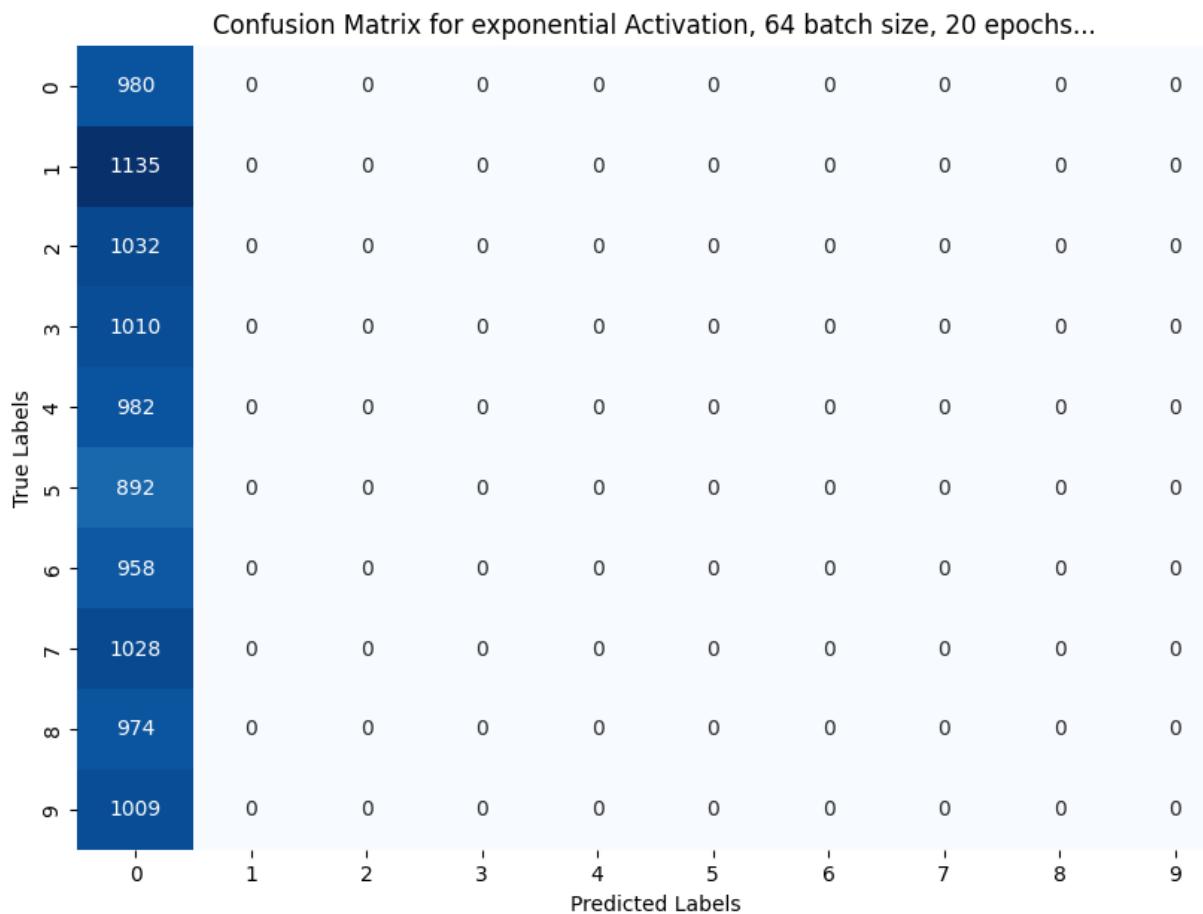
```
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 8/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 9/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 13/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 14/20  
844/844 - 6s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 6s/epoch - 7ms/step  
Epoch 15/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Results for exponential function, 64 batch size and 20 epochs...  
Confusion Matrix  
[[ 980  0  0  0  0  0  0  0  0 ]  
[ 1135  0  0  0  0  0  0  0  0 ]  
[ 1032  0  0  0  0  0  0  0  0 ]  
[ 1010  0  0  0  0  0  0  0  0 ]  
[  982  0  0  0  0  0  0  0  0 ]
```

```

[ 892  0  0  0  0  0  0  0  0  0]
[ 958  0  0  0  0  0  0  0  0  0]
[1028  0  0  0  0  0  0  0  0  0]
[ 974  0  0  0  0  0  0  0  0  0]
[1009  0  0  0  0  0  0  0  0  0]]
Precision:  0.0096
Recall:   0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))

```



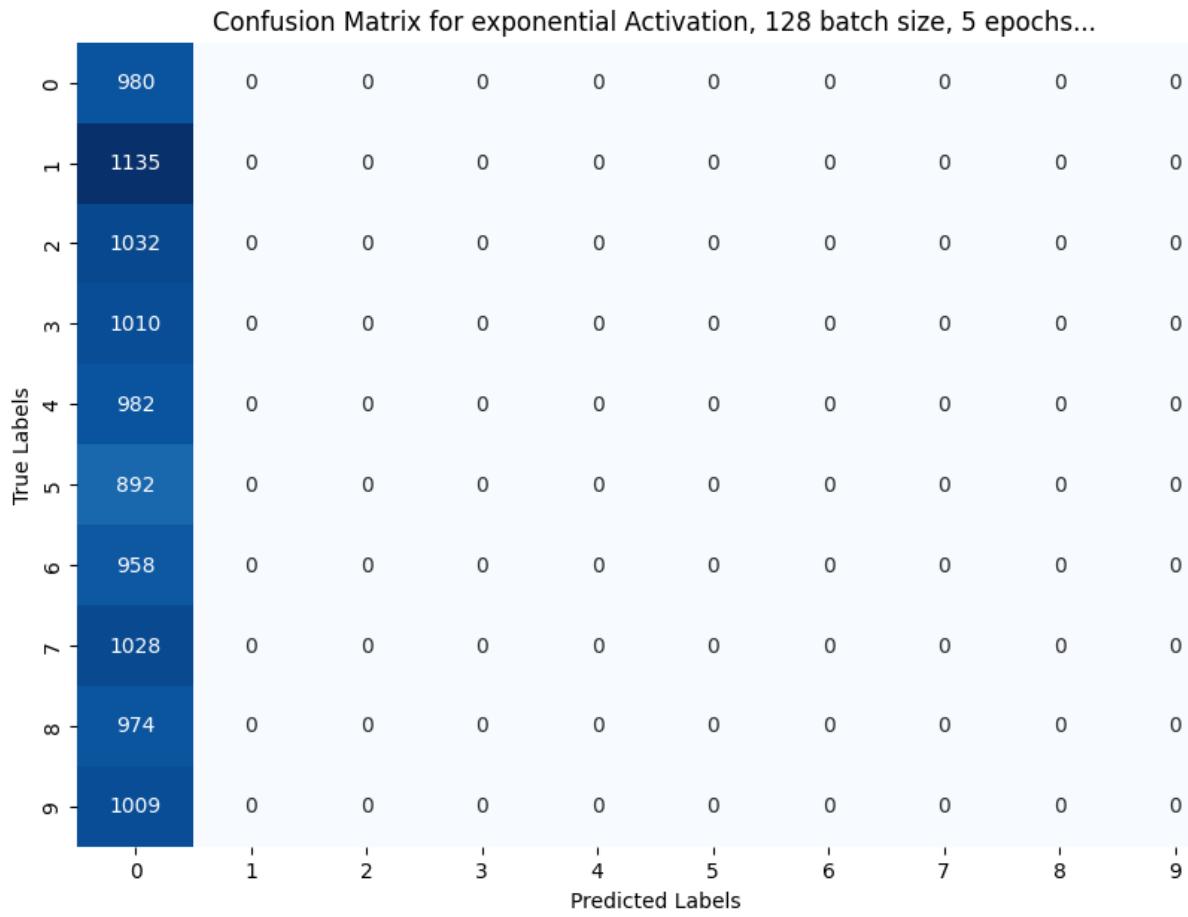
```

Training model with exponential activation function, 128 batch sizes,
5 epochs_list
Epoch 1/5
422/422 - 4s - loss: nan - accuracy: 0.0990 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 10ms/step
Epoch 2/5
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -

```

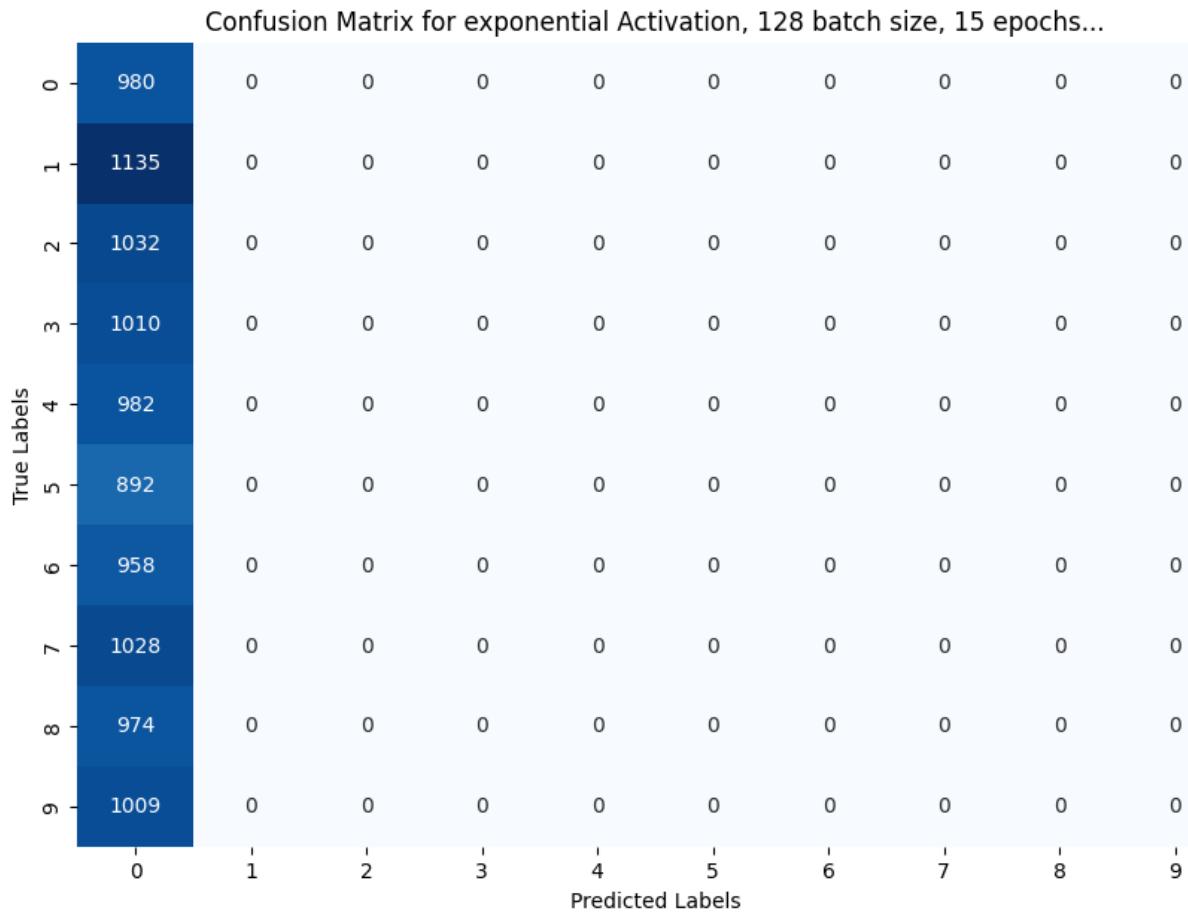
```
val_accuracy: 0.0978 - 4s/epoch - 8ms/step
Epoch 3/5
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 8ms/step
Epoch 4/5
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 8ms/step
Epoch 5/5
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for exponential function, 128 batch size and 5 epochs...
Confusion Matrix
[[ 980  0  0  0  0  0  0  0  0  0]
 [1135  0  0  0  0  0  0  0  0  0]
 [1032  0  0  0  0  0  0  0  0  0]
 [1010  0  0  0  0  0  0  0  0  0]
 [ 982  0  0  0  0  0  0  0  0  0]
 [ 892  0  0  0  0  0  0  0  0  0]
 [ 958  0  0  0  0  0  0  0  0  0]
 [1028  0  0  0  0  0  0  0  0  0]
 [ 974  0  0  0  0  0  0  0  0  0]
 [1009  0  0  0  0  0  0  0  0  0]]
Precision: 0.0096
Recall: 0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with exponential activation function, 128 batch sizes,
15 epochs_list
Epoch 1/15
422/422 - 4s - loss: nan - accuracy: 0.0989 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 10ms/step
Epoch 2/15
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 9ms/step
Epoch 3/15
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 8ms/step
Epoch 4/15
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 8ms/step
Epoch 5/15
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 9ms/step
Epoch 6/15
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 9ms/step
Epoch 7/15
```

```
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 8/15  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 9/15  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 10/15  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 11/15  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 8ms/step  
Epoch 12/15  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 13/15  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 14/15  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 15/15  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
313/313 [=====] - 1s 3ms/step  
Results for exponential function, 128 batch size and 15 epochs...  
Confusion Matrix  
[[ 980  0  0  0  0  0  0  0  0  0]  
[1135  0  0  0  0  0  0  0  0  0]  
[1032  0  0  0  0  0  0  0  0  0]  
[1010  0  0  0  0  0  0  0  0  0]  
[ 982  0  0  0  0  0  0  0  0  0]  
[ 892  0  0  0  0  0  0  0  0  0]  
[ 958  0  0  0  0  0  0  0  0  0]  
[1028  0  0  0  0  0  0  0  0  0]  
[ 974  0  0  0  0  0  0  0  0  0]  
[1009  0  0  0  0  0  0  0  0  0]]  
Precision: 0.0096  
Recall: 0.0980  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with exponential activation function, 128 batch sizes,
20 epochs_list
Epoch 1/20
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 10ms/step
Epoch 2/20
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 9ms/step
Epoch 3/20
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 9ms/step
Epoch 4/20
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 9ms/step
Epoch 5/20
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 10ms/step
Epoch 6/20
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 9ms/step
Epoch 7/20
```

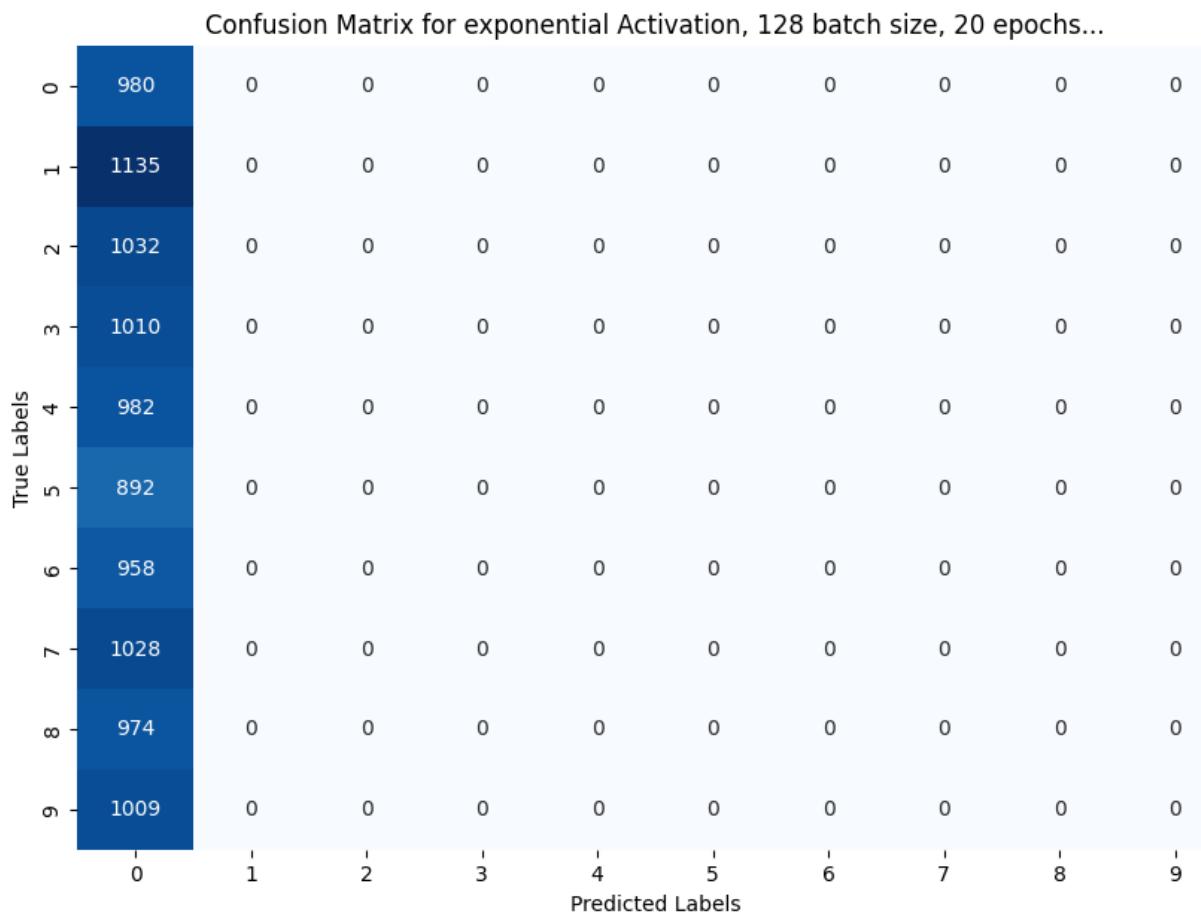
```
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 8/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 9/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 10ms/step  
Epoch 10/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 10ms/step  
Epoch 11/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 10ms/step  
Epoch 12/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 13/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 9ms/step  
Epoch 14/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 8ms/step  
Epoch 15/20  
422/422 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 8ms/step  
Epoch 16/20  
422/422 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 8ms/step  
Epoch 17/20  
422/422 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 8ms/step  
Epoch 18/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 8ms/step  
Epoch 19/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 10ms/step  
Epoch 20/20  
422/422 - 4s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 4s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for exponential function, 128 batch size and 20 epochs...  
Confusion Matrix  
[[ 980  0  0  0  0  0  0  0  0 ]  
[ 1135  0  0  0  0  0  0  0  0 ]  
[ 1032  0  0  0  0  0  0  0  0 ]  
[ 1010  0  0  0  0  0  0  0  0 ]  
[  982  0  0  0  0  0  0  0  0 ]
```

```

[ 892  0  0  0  0  0  0  0  0  0]
[ 958  0  0  0  0  0  0  0  0  0]
[1028  0  0  0  0  0  0  0  0  0]
[ 974  0  0  0  0  0  0  0  0  0]
[1009  0  0  0  0  0  0  0  0  0]]
Precision:  0.0096
Recall:   0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))

```



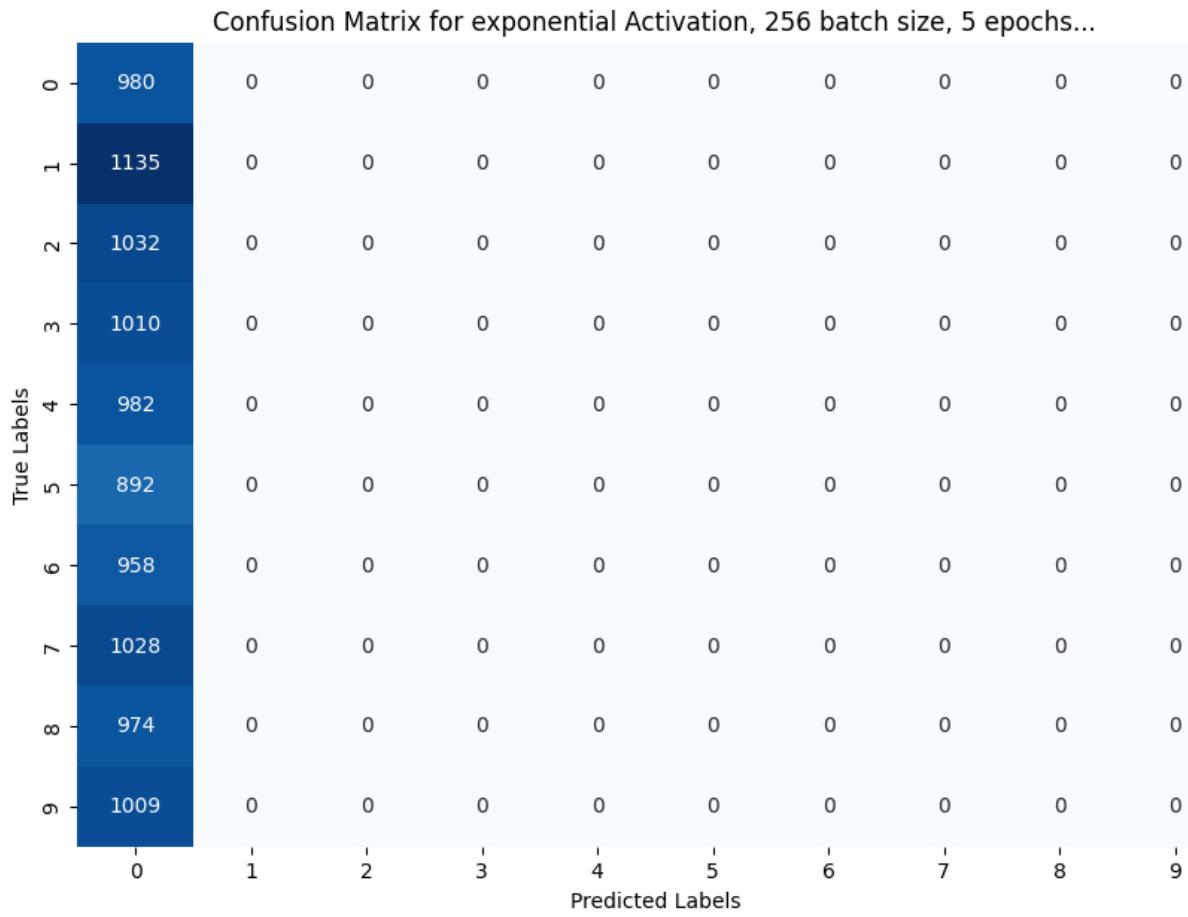
```

Training model with exponential activation function, 256 batch sizes,
5 epochs_list
Epoch 1/5
211/211 - 4s - loss: nan - accuracy: 0.0987 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 19ms/step
Epoch 2/5
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -

```

```
val_accuracy: 0.0978 - 3s/epoch - 15ms/step
Epoch 3/5
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 14ms/step
Epoch 4/5
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 14ms/step
Epoch 5/5
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 15ms/step
313/313 [=====] - 1s 3ms/step
Results for exponential function, 256 batch size and 5 epochs...
Confusion Matrix
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
Precision: 0.0096
Recall: 0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

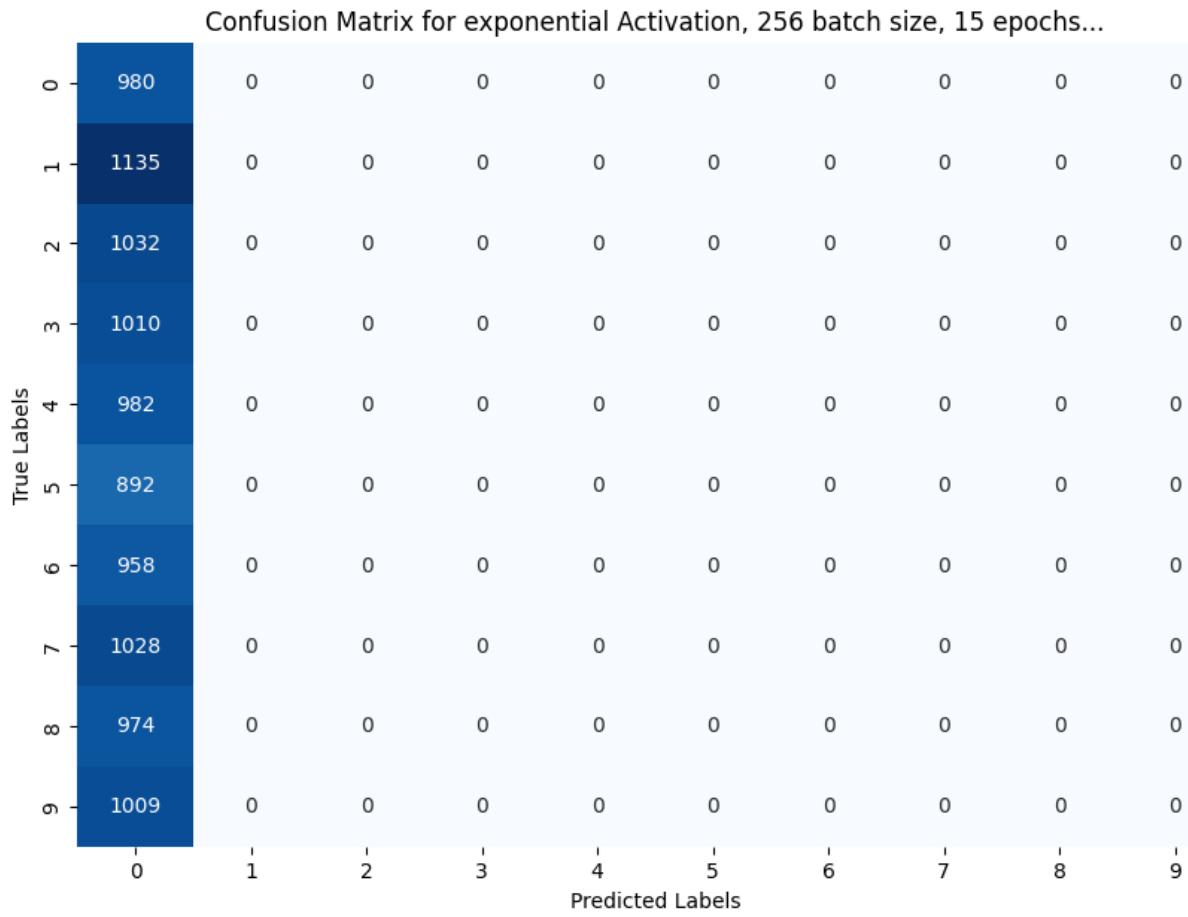


```
Training model with exponential activation function, 256 batch sizes,
15 epochs_list
Epoch 1/15
211/211 - 4s - loss: nan - accuracy: 0.0991 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 17ms/step
Epoch 2/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 14ms/step
Epoch 3/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 4/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 5/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 6/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 7/15
```

```
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 8/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 9/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 10/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 11/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 12/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 13/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 14/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 13ms/step
Epoch 15/15
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 14ms/step
313/313 [=====] - 1s 3ms/step
Results for exponential function, 256 batch size and 15 epochs...
Confusion Matrix
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
```

Precision: 0.0096
Recall: 0.0980

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with exponential activation function, 256 batch sizes,
20 epochs_list
Epoch 1/20
211/211 - 4s - loss: nan - accuracy: 0.0986 - val_loss: nan -
val_accuracy: 0.0978 - 4s/epoch - 18ms/step
Epoch 2/20
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 15ms/step
Epoch 3/20
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 15ms/step
Epoch 4/20
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 15ms/step
Epoch 5/20
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 15ms/step
Epoch 6/20
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 3s/epoch - 15ms/step
Epoch 7/20
```

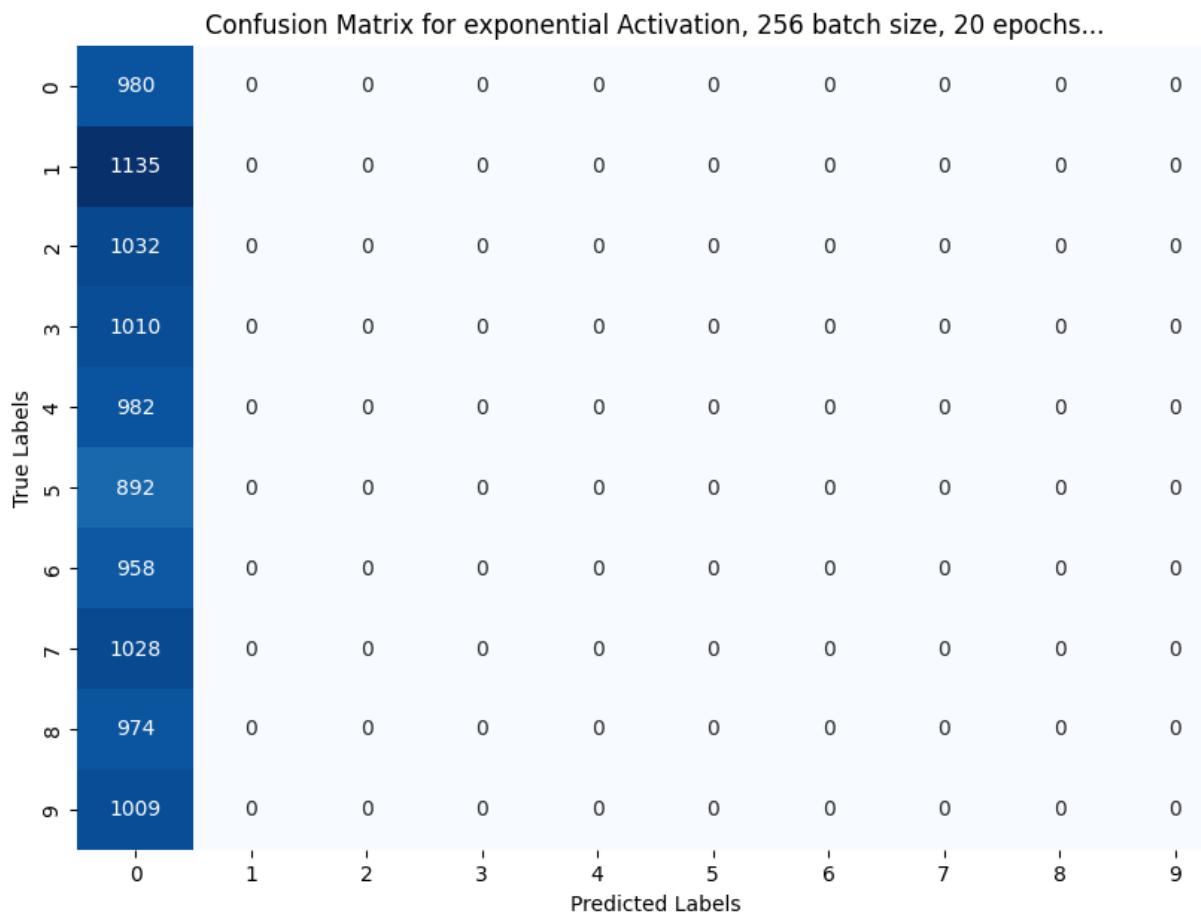
```
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 15ms/step  
Epoch 8/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 15ms/step  
Epoch 9/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 15ms/step  
Epoch 10/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 15ms/step  
Epoch 11/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 15ms/step  
Epoch 12/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 15ms/step  
Epoch 13/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 14ms/step  
Epoch 14/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 14ms/step  
Epoch 15/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 14ms/step  
Epoch 16/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 13ms/step  
Epoch 17/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 13ms/step  
Epoch 18/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 13ms/step  
Epoch 19/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 13ms/step  
Epoch 20/20  
211/211 - 3s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 3s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Results for exponential function, 256 batch size and 20 epochs...  
Confusion Matrix  
[[ 980  0  0  0  0  0  0  0  0 ]  
 [1135  0  0  0  0  0  0  0  0 ]  
 [1032  0  0  0  0  0  0  0  0 ]  
 [1010  0  0  0  0  0  0  0  0 ]  
 [ 982  0  0  0  0  0  0  0  0 ]
```

```

[ 892  0  0  0  0  0  0  0  0  0]
[ 958  0  0  0  0  0  0  0  0  0]
[1028  0  0  0  0  0  0  0  0  0]
[ 974  0  0  0  0  0  0  0  0  0]
[1009  0  0  0  0  0  0  0  0  0]]
Precision:  0.0096
Recall:   0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))

```



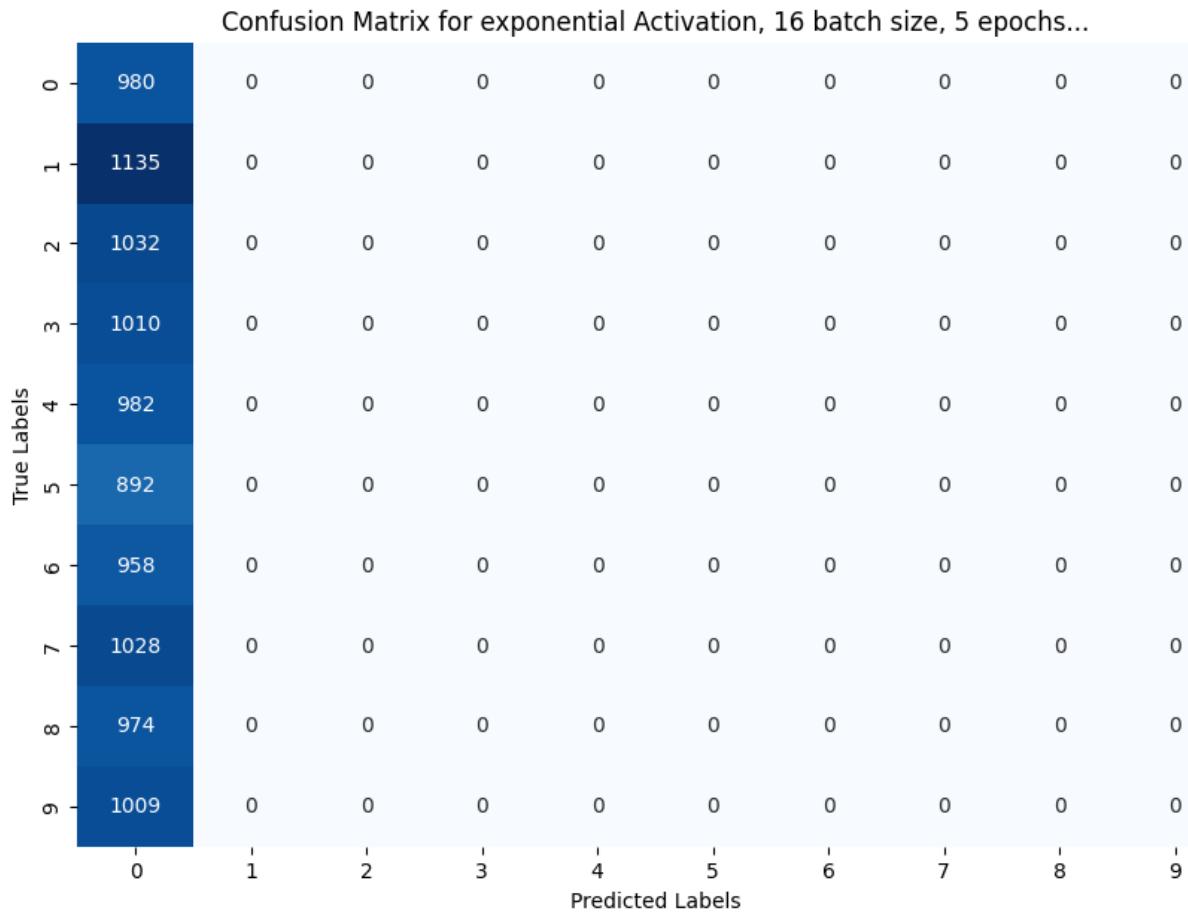
```

Training model with exponential activation function, 16 batch sizes, 5
epochs_list
Epoch 1/5
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 2/5
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -

```

```
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 3/5
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 4/5
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 5/5
3375/3375 - 17s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 17s/epoch - 5ms/step
313/313 [=====] - 1s 3ms/step
Results for exponential function, 16 batch size and 5 epochs...
Confusion Matrix
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
Precision: 0.0096
Recall: 0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

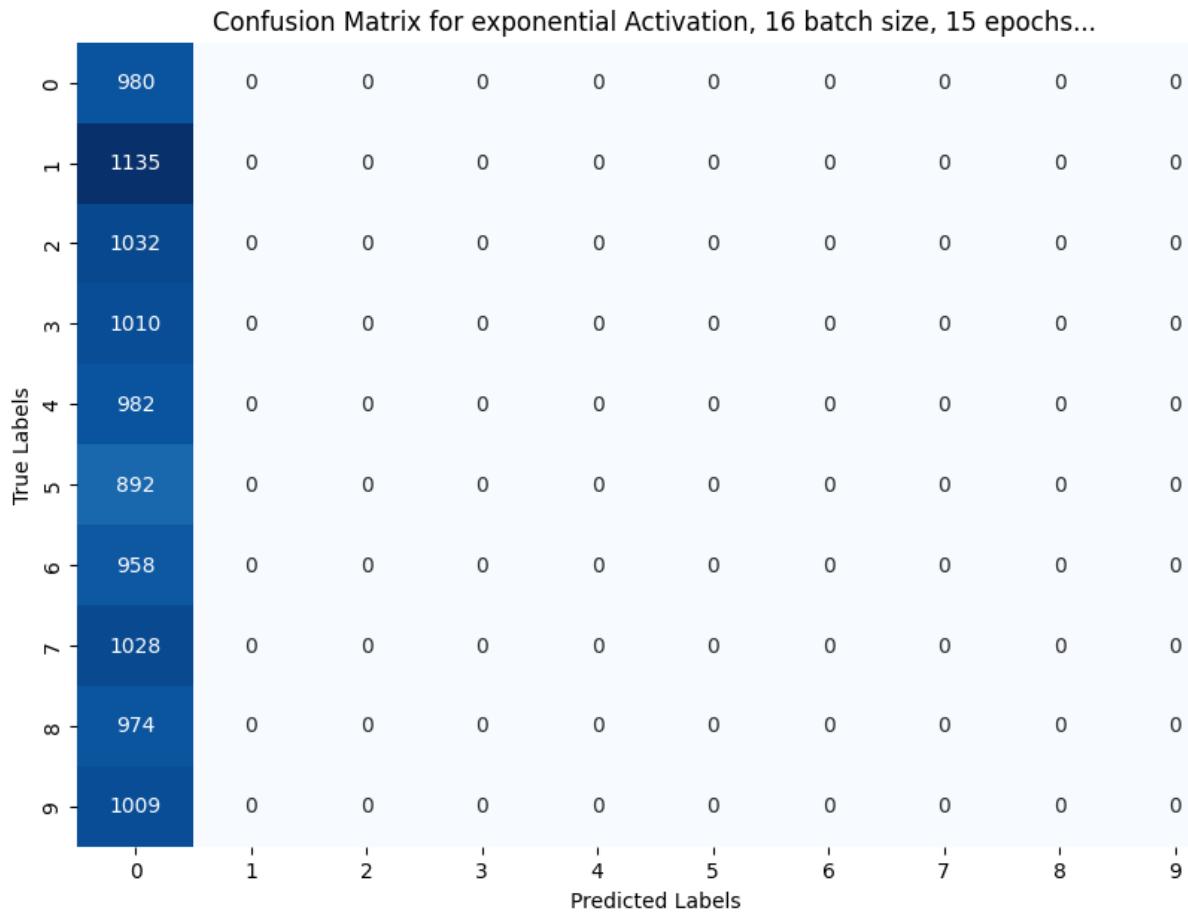


```
Training model with exponential activation function, 16 batch sizes,
15 epochs_list
Epoch 1/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 2/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 3/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 4/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 5/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 6/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 7/15
```

```
3375/3375 - 15s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 15s/epoch - 5ms/step
Epoch 8/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 9/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 10/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 11/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 12/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 13/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 14/15
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Results for exponential function, 16 batch size and 15 epochs...
Confusion Matrix
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
```

Precision: 0.0096
Recall: 0.0980

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training model with exponential activation function, 16 batch sizes,
20 epochs_list
Epoch 1/20
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 2/20
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 3/20
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 4/20
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 5/20
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 6/20
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -
val_accuracy: 0.0978 - 16s/epoch - 5ms/step
Epoch 7/20
```

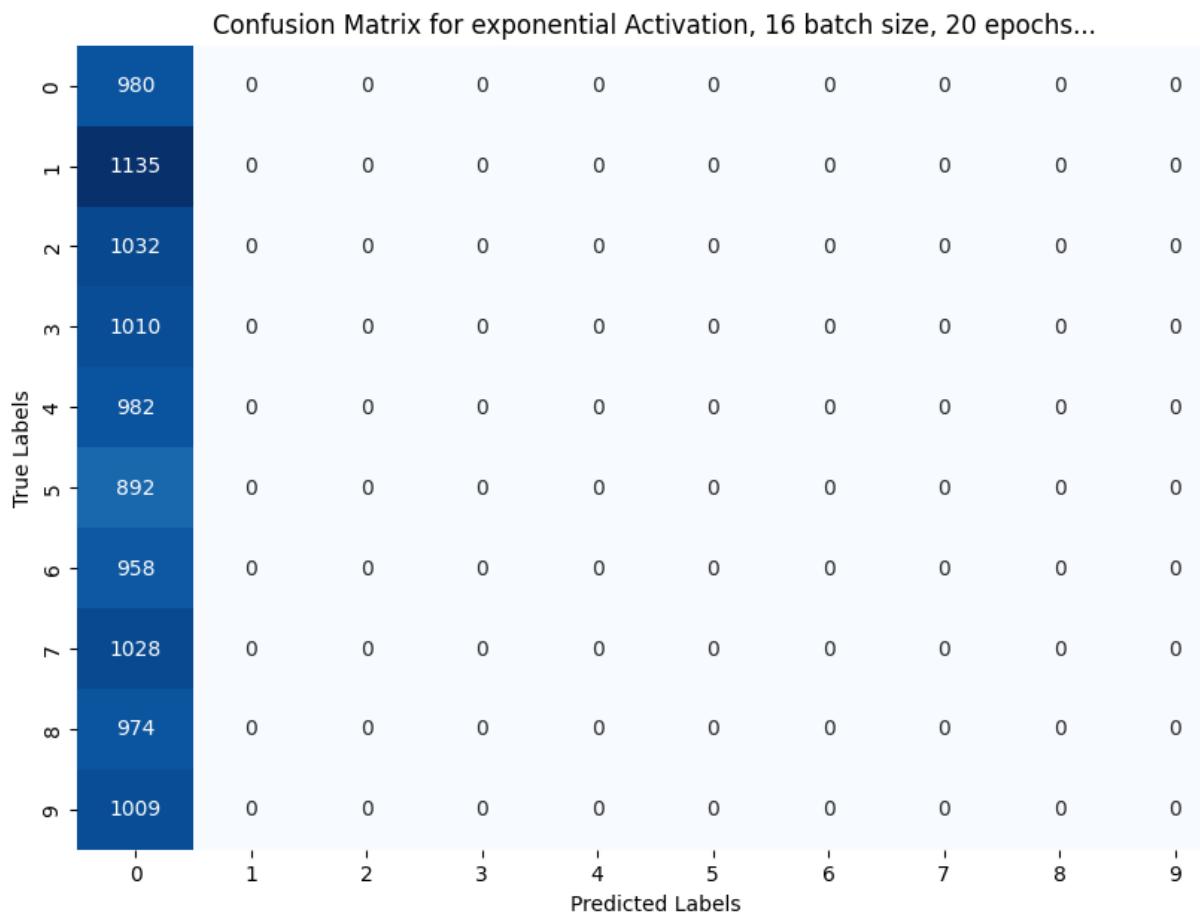
```
3375/3375 - 15s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 15s/epoch - 5ms/step  
Epoch 8/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 9/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 10/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 11/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 12/20  
3375/3375 - 15s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 15s/epoch - 4ms/step  
Epoch 13/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 14/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 15/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 16/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 17/20  
3375/3375 - 15s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 15s/epoch - 5ms/step  
Epoch 18/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 19/20  
3375/3375 - 16s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 16s/epoch - 5ms/step  
Epoch 20/20  
3375/3375 - 17s - loss: nan - accuracy: 0.0988 - val_loss: nan -  
val_accuracy: 0.0978 - 17s/epoch - 5ms/step  
313/313 [=====] - 1s 3ms/step  
Results for exponential function, 16 batch size and 20 epochs...  
Confusion Matrix  
[[ 980  0  0  0  0  0  0  0  0 ]  
 [1135  0  0  0  0  0  0  0  0 ]  
 [1032  0  0  0  0  0  0  0  0 ]  
 [1010  0  0  0  0  0  0  0  0 ]  
 [ 982  0  0  0  0  0  0  0  0 ]
```

```

[ 892  0  0  0  0  0  0  0  0  0]
[ 958  0  0  0  0  0  0  0  0  0]
[1028  0  0  0  0  0  0  0  0  0]
[ 974  0  0  0  0  0  0  0  0  0]
[1009  0  0  0  0  0  0  0  0  0]]
Precision:  0.0096
Recall:   0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))

```



```

Training model with hard_sigmoid activation function, 64 batch sizes,
5 epochs_list
Epoch 1/5
844/844 - 9s - loss: 0.9359 - accuracy: 0.7304 - val_loss: 0.3803 -
val_accuracy: 0.8917 - 9s/epoch - 11ms/step
Epoch 2/5
844/844 - 8s - loss: 0.3600 - accuracy: 0.8932 - val_loss: 0.2485 -

```

```
val_accuracy: 0.9300 - 8s/epoch - 10ms/step
Epoch 3/5
844/844 - 8s - loss: 0.2914 - accuracy: 0.9133 - val_loss: 0.2126 -
val_accuracy: 0.9390 - 8s/epoch - 9ms/step
Epoch 4/5
844/844 - 8s - loss: 0.2523 - accuracy: 0.9247 - val_loss: 0.1876 -
val_accuracy: 0.9443 - 8s/epoch - 9ms/step
Epoch 5/5
844/844 - 8s - loss: 0.2181 - accuracy: 0.9349 - val_loss: 0.1663 -
val_accuracy: 0.9530 - 8s/epoch - 9ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 64 batch size and 5 epochs...
Confusion Matrix
[[ 949   0   1   2   0   8  14   1   5   0]
 [  0 1124   2   2   0   1   3   0   3   0]
 [  6   4  956   17   2   1  11   8  24   3]
 [  0   2  14  955   0   8   1   7  20   3]
 [  0   8   6   1  904   2  17   2   7  35]
 [  3   3   0  34   2  813   8   2  25   2]
 [  5   3   3   0   2  10  929   1   5   0]
 [  1  16  24  10   6   1   0  940   1  29]
 [  3   3   2  17   3  10   7   3  922   4]
 [  5  10   1  14  16  10   1   8  10  934]]
```

Precision: 0.9432
Recall: 0.9426

Confusion Matrix for hard_sigmoid Activation, 64 batch size, 5 epochs...											
	0	1	2	3	4	5	6	7	8	9	
0	949	0	1	2	0	8	14	1	5	0	
1	0	1124	2	2	0	1	3	0	3	0	
2	6	4	956	17	2	1	11	8	24	3	
3	0	2	14	955	0	8	1	7	20	3	
4	0	8	6	1	904	2	17	2	7	35	
5	3	3	0	34	2	813	8	2	25	2	
6	5	3	3	0	2	10	929	1	5	0	
7	1	16	24	10	6	1	0	940	1	29	
8	3	3	2	17	3	10	7	3	922	4	
9	5	10	1	14	16	10	1	8	10	934	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```

Training model with hard_sigmoid activation function, 64 batch sizes,
15 epochs_list
Epoch 1/15
844/844 - 8s - loss: 0.9142 - accuracy: 0.7427 - val_loss: 0.3232 -
val_accuracy: 0.9107 - 8s/epoch - 10ms/step
Epoch 2/15
844/844 - 8s - loss: 0.3550 - accuracy: 0.8940 - val_loss: 0.2493 -
val_accuracy: 0.9257 - 8s/epoch - 9ms/step
Epoch 3/15
844/844 - 8s - loss: 0.2907 - accuracy: 0.9118 - val_loss: 0.2090 -
val_accuracy: 0.9400 - 8s/epoch - 9ms/step
Epoch 4/15
844/844 - 8s - loss: 0.2486 - accuracy: 0.9262 - val_loss: 0.1883 -
val_accuracy: 0.9448 - 8s/epoch - 9ms/step
Epoch 5/15
844/844 - 7s - loss: 0.2188 - accuracy: 0.9347 - val_loss: 0.1646 -
val_accuracy: 0.9552 - 7s/epoch - 9ms/step
Epoch 6/15
844/844 - 7s - loss: 0.1960 - accuracy: 0.9408 - val_loss: 0.1451 -
val_accuracy: 0.9590 - 7s/epoch - 9ms/step
Epoch 7/15

```

```
844/844 - 8s - loss: 0.1746 - accuracy: 0.9479 - val_loss: 0.1351 -  
val_accuracy: 0.9632 - 8s/epoch - 9ms/step  
Epoch 8/15  
844/844 - 7s - loss: 0.1577 - accuracy: 0.9535 - val_loss: 0.1240 -  
val_accuracy: 0.9660 - 7s/epoch - 9ms/step  
Epoch 9/15  
844/844 - 7s - loss: 0.1413 - accuracy: 0.9587 - val_loss: 0.1103 -  
val_accuracy: 0.9710 - 7s/epoch - 9ms/step  
Epoch 10/15  
844/844 - 8s - loss: 0.1307 - accuracy: 0.9615 - val_loss: 0.1027 -  
val_accuracy: 0.9722 - 8s/epoch - 9ms/step  
Epoch 11/15  
844/844 - 8s - loss: 0.1199 - accuracy: 0.9659 - val_loss: 0.0929 -  
val_accuracy: 0.9753 - 8s/epoch - 9ms/step  
Epoch 12/15  
844/844 - 8s - loss: 0.1114 - accuracy: 0.9671 - val_loss: 0.0961 -  
val_accuracy: 0.9750 - 8s/epoch - 9ms/step  
Epoch 13/15  
844/844 - 8s - loss: 0.1037 - accuracy: 0.9695 - val_loss: 0.0918 -  
val_accuracy: 0.9763 - 8s/epoch - 9ms/step  
Epoch 14/15  
844/844 - 8s - loss: 0.0975 - accuracy: 0.9710 - val_loss: 0.0949 -  
val_accuracy: 0.9747 - 8s/epoch - 9ms/step  
Epoch 15/15  
844/844 - 8s - loss: 0.0913 - accuracy: 0.9733 - val_loss: 0.0838 -  
val_accuracy: 0.9787 - 8s/epoch - 9ms/step  
313/313 [=====] - 1s 3ms/step  
Results for hard_sigmoid function, 64 batch size and 15 epochs...  
Confusion Matrix  
[[ 964 0 0 2 0 4 6 2 2 0]  
[ 0 1127 2 1 0 1 2 0 2 0]  
[ 4 4 994 12 4 0 3 7 3 1]  
[ 0 0 1 999 0 2 0 5 2 1]  
[ 1 0 5 0 941 0 8 2 2 23]  
[ 3 1 0 20 1 859 5 0 1 2]  
[ 6 2 1 0 4 8 934 0 3 0]  
[ 0 7 11 10 0 1 0 990 0 9]  
[ 4 0 2 22 2 6 3 4 924 7]  
[ 5 5 0 13 6 4 1 7 0 968]]  
Precision: 0.9704  
Recall: 0.9700
```

Confusion Matrix for hard_sigmoid Activation, 64 batch size, 15 epochs...											
	0	1	2	3	4	5	6	7	8	9	
0	964	0	0	2	0	4	6	2	2	0	
1	0	1127	2	1	0	1	2	0	2	0	
2	4	4	994	12	4	0	3	7	3	1	
3	0	0	1	999	0	2	0	5	2	1	
4	1	0	5	0	941	0	8	2	2	23	
5	3	1	0	20	1	859	5	0	1	2	
6	6	2	1	0	4	8	934	0	3	0	
7	0	7	11	10	0	1	0	990	0	9	
8	4	0	2	22	2	6	3	4	924	7	
9	5	5	0	13	6	4	1	7	0	968	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels

```
Training model with hard_sigmoid activation function, 64 batch sizes,
20 epochs_list
Epoch 1/20
844/844 - 8s - loss: 0.8596 - accuracy: 0.7551 - val_loss: 0.3343 -
val_accuracy: 0.9098 - 8s/epoch - 10ms/step
Epoch 2/20
844/844 - 8s - loss: 0.3560 - accuracy: 0.8945 - val_loss: 0.2512 -
val_accuracy: 0.9272 - 8s/epoch - 9ms/step
Epoch 3/20
844/844 - 8s - loss: 0.2905 - accuracy: 0.9129 - val_loss: 0.2199 -
val_accuracy: 0.9323 - 8s/epoch - 9ms/step
Epoch 4/20
844/844 - 8s - loss: 0.2515 - accuracy: 0.9251 - val_loss: 0.1809 -
val_accuracy: 0.9503 - 8s/epoch - 9ms/step
Epoch 5/20
844/844 - 8s - loss: 0.2221 - accuracy: 0.9330 - val_loss: 0.1671 -
val_accuracy: 0.9527 - 8s/epoch - 9ms/step
Epoch 6/20
844/844 - 8s - loss: 0.1983 - accuracy: 0.9404 - val_loss: 0.1548 -
val_accuracy: 0.9575 - 8s/epoch - 10ms/step
Epoch 7/20
```

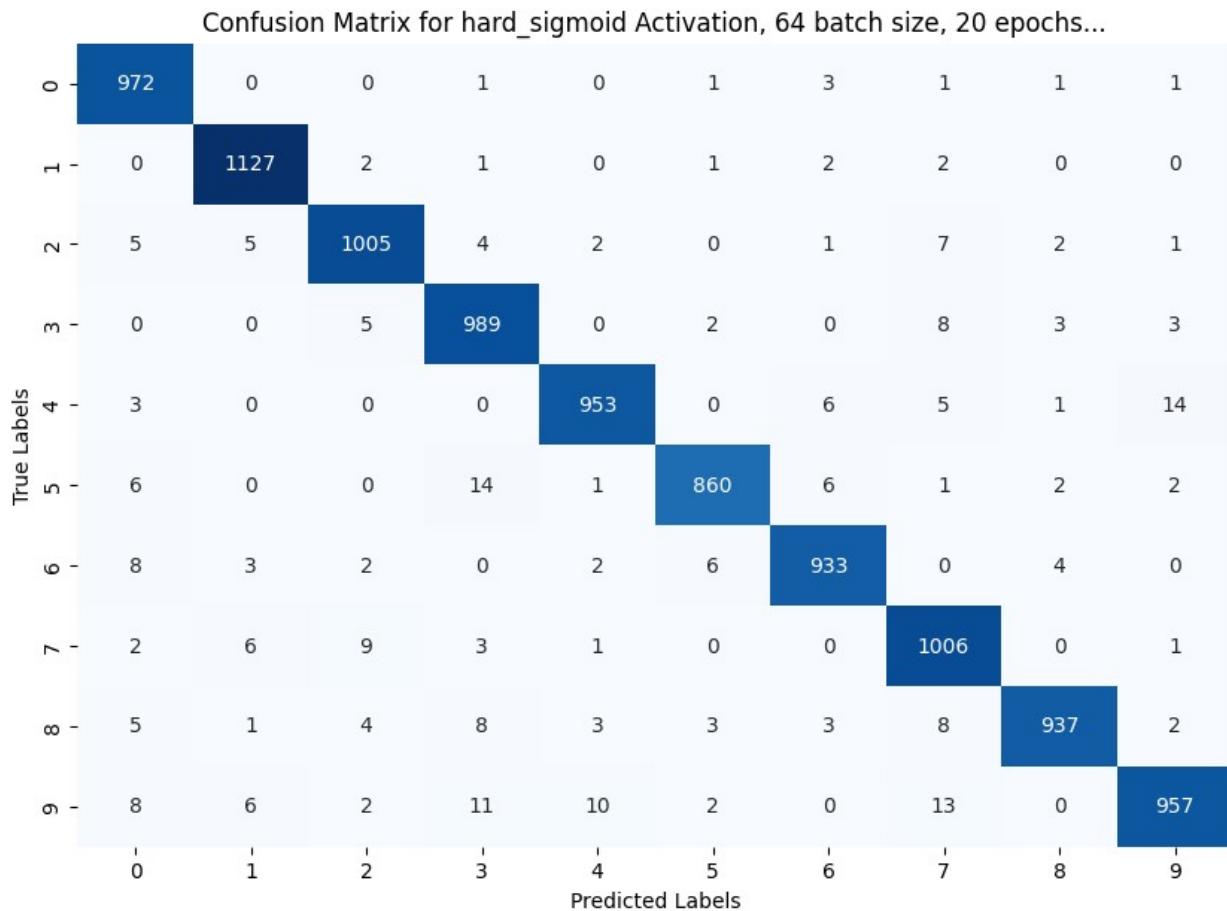
```

844/844 - 8s - loss: 0.1790 - accuracy: 0.9459 - val_loss: 0.1472 -
val_accuracy: 0.9578 - 8s/epoch - 10ms/step
Epoch 8/20
844/844 - 8s - loss: 0.1641 - accuracy: 0.9512 - val_loss: 0.1278 -
val_accuracy: 0.9637 - 8s/epoch - 9ms/step
Epoch 9/20
844/844 - 8s - loss: 0.1512 - accuracy: 0.9552 - val_loss: 0.1160 -
val_accuracy: 0.9682 - 8s/epoch - 9ms/step
Epoch 10/20
844/844 - 8s - loss: 0.1390 - accuracy: 0.9596 - val_loss: 0.1142 -
val_accuracy: 0.9680 - 8s/epoch - 9ms/step
Epoch 11/20
844/844 - 8s - loss: 0.1297 - accuracy: 0.9618 - val_loss: 0.1083 -
val_accuracy: 0.9698 - 8s/epoch - 9ms/step
Epoch 12/20
844/844 - 8s - loss: 0.1219 - accuracy: 0.9646 - val_loss: 0.1032 -
val_accuracy: 0.9732 - 8s/epoch - 9ms/step
Epoch 13/20
844/844 - 8s - loss: 0.1129 - accuracy: 0.9673 - val_loss: 0.0947 -
val_accuracy: 0.9742 - 8s/epoch - 9ms/step
Epoch 14/20
844/844 - 7s - loss: 0.1066 - accuracy: 0.9684 - val_loss: 0.0982 -
val_accuracy: 0.9738 - 7s/epoch - 9ms/step
Epoch 15/20
844/844 - 8s - loss: 0.1002 - accuracy: 0.9707 - val_loss: 0.0963 -
val_accuracy: 0.9730 - 8s/epoch - 9ms/step
Epoch 16/20
844/844 - 8s - loss: 0.0943 - accuracy: 0.9728 - val_loss: 0.0853 -
val_accuracy: 0.9760 - 8s/epoch - 9ms/step
Epoch 17/20
844/844 - 8s - loss: 0.0890 - accuracy: 0.9746 - val_loss: 0.0881 -
val_accuracy: 0.9742 - 8s/epoch - 9ms/step
Epoch 18/20
844/844 - 8s - loss: 0.0839 - accuracy: 0.9756 - val_loss: 0.0786 -
val_accuracy: 0.9778 - 8s/epoch - 9ms/step
Epoch 19/20
844/844 - 8s - loss: 0.0804 - accuracy: 0.9764 - val_loss: 0.0778 -
val_accuracy: 0.9778 - 8s/epoch - 9ms/step
Epoch 20/20
844/844 - 8s - loss: 0.0755 - accuracy: 0.9783 - val_loss: 0.0775 -
val_accuracy: 0.9778 - 8s/epoch - 9ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 64 batch size and 20 epochs...
Confusion Matrix
[[ 972   0   0   1   0   1   3   1   1   1]
 [  0 1127   2   1   0   1   2   2   0   0]
 [  5   5 1005   4   2   0   1   7   2   1]
 [  0   0   5  989   0   2   0   8   3   3]
 [  3   0   0   0  953   0   6   5   1  14]]
```

```
[ 6 0 0 14 1 860 6 1 2 2]
[ 8 3 2 0 2 6 933 0 4 0]
[ 2 6 9 3 1 0 0 1006 0 1]
[ 5 1 4 8 3 3 3 8 937 2]
[ 8 6 2 11 10 2 0 13 0 957]]
```

Precision: 0.9740

Recall: 0.9739



Training model with hard_sigmoid activation function, 128 batch sizes, 5 epochs_list

Epoch 1/5

422/422 - 6s - loss: 1.1875 - accuracy: 0.6818 - val_loss: 0.4729 - val_accuracy: 0.8858 - 6s/epoch - 15ms/step

Epoch 2/5

422/422 - 6s - loss: 0.4468 - accuracy: 0.8759 - val_loss: 0.3330 - val_accuracy: 0.9037 - 6s/epoch - 13ms/step

Epoch 3/5

422/422 - 5s - loss: 0.3642 - accuracy: 0.8934 - val_loss: 0.2706 - val_accuracy: 0.9235 - 5s/epoch - 13ms/step

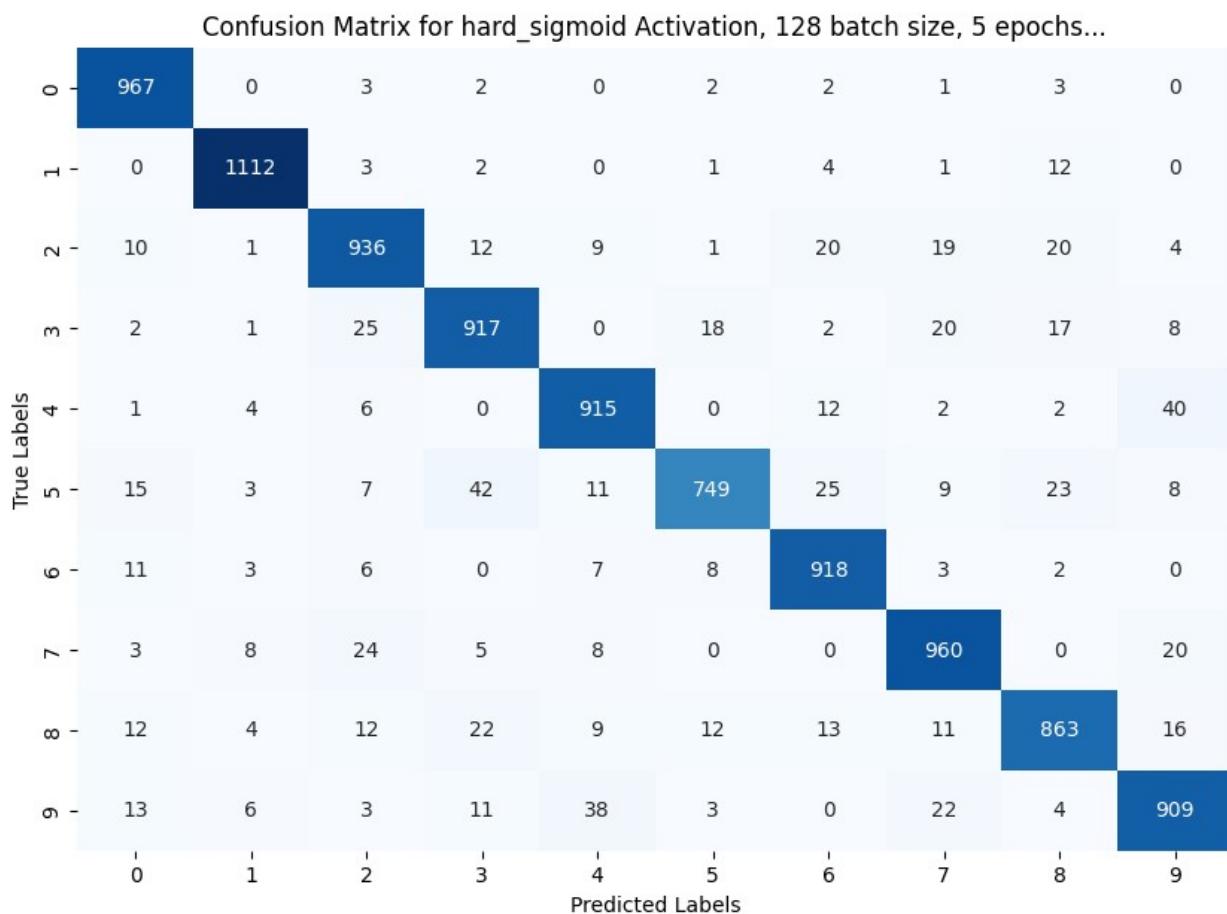
Epoch 4/5

422/422 - 5s - loss: 0.3245 - accuracy: 0.9036 - val_loss: 0.2560 -

```

val_accuracy: 0.9252 - 5s/epoch - 13ms/step
Epoch 5/5
422/422 - 5s - loss: 0.2954 - accuracy: 0.9125 - val_loss: 0.2204 -
val_accuracy: 0.9372 - 5s/epoch - 13ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 128 batch size and 5 epochs...
Confusion Matrix
[[ 967 0 3 2 0 2 2 1 3 0]
 [ 0 1112 3 2 0 1 4 1 12 0]
 [ 10 1 936 12 9 1 20 19 20 4]
 [ 2 1 25 917 0 18 2 20 17 8]
 [ 1 4 6 0 915 0 12 2 2 40]
 [ 15 3 7 42 11 749 25 9 23 8]
 [ 11 3 6 0 7 8 918 3 2 0]
 [ 3 8 24 5 8 0 0 960 0 20]
 [ 12 4 12 22 9 12 13 11 863 16]
 [ 13 6 3 11 38 3 0 22 4 909]]
Precision: 0.9247
Recall: 0.9246

```

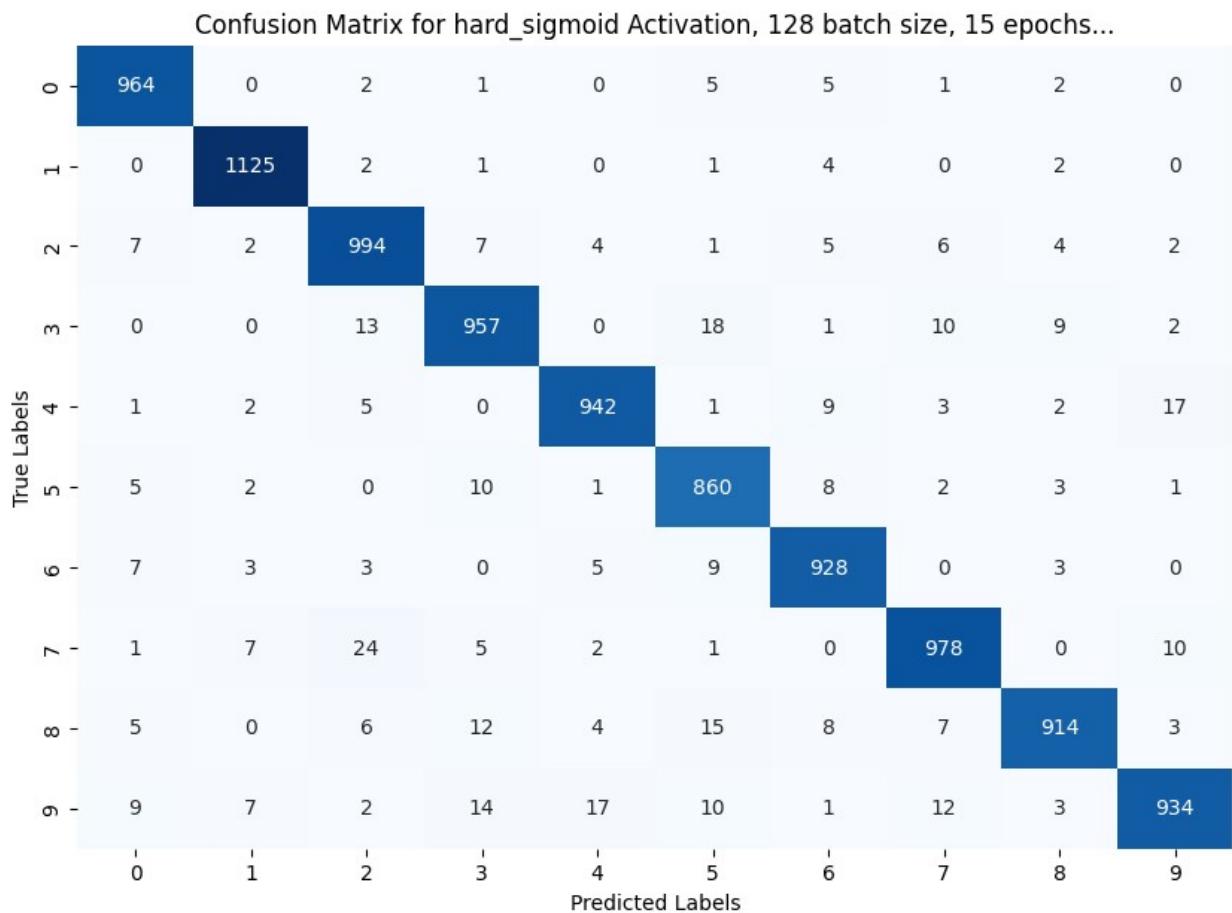


```
Training model with hard_sigmoid activation function, 128 batch sizes,
15 epochs_list
Epoch 1/15
422/422 - 6s - loss: 1.2687 - accuracy: 0.6686 - val_loss: 0.4853 -
val_accuracy: 0.8923 - 6s/epoch - 15ms/step
Epoch 2/15
422/422 - 6s - loss: 0.4571 - accuracy: 0.8734 - val_loss: 0.3256 -
val_accuracy: 0.9140 - 6s/epoch - 13ms/step
Epoch 3/15
422/422 - 6s - loss: 0.3632 - accuracy: 0.8943 - val_loss: 0.2726 -
val_accuracy: 0.9233 - 6s/epoch - 13ms/step
Epoch 4/15
422/422 - 6s - loss: 0.3196 - accuracy: 0.9056 - val_loss: 0.2430 -
val_accuracy: 0.9287 - 6s/epoch - 13ms/step
Epoch 5/15
422/422 - 6s - loss: 0.2946 - accuracy: 0.9128 - val_loss: 0.2249 -
val_accuracy: 0.9338 - 6s/epoch - 13ms/step
Epoch 6/15
422/422 - 5s - loss: 0.2736 - accuracy: 0.9199 - val_loss: 0.2241 -
val_accuracy: 0.9363 - 5s/epoch - 13ms/step
Epoch 7/15
422/422 - 5s - loss: 0.2521 - accuracy: 0.9257 - val_loss: 0.2117 -
val_accuracy: 0.9378 - 5s/epoch - 13ms/step
Epoch 8/15
422/422 - 5s - loss: 0.2342 - accuracy: 0.9308 - val_loss: 0.1776 -
val_accuracy: 0.9497 - 5s/epoch - 12ms/step
Epoch 9/15
422/422 - 6s - loss: 0.2187 - accuracy: 0.9353 - val_loss: 0.1776 -
val_accuracy: 0.9513 - 6s/epoch - 13ms/step
Epoch 10/15
422/422 - 6s - loss: 0.2061 - accuracy: 0.9396 - val_loss: 0.1563 -
val_accuracy: 0.9578 - 6s/epoch - 13ms/step
Epoch 11/15
422/422 - 6s - loss: 0.1941 - accuracy: 0.9443 - val_loss: 0.1504 -
val_accuracy: 0.9598 - 6s/epoch - 13ms/step
Epoch 12/15
422/422 - 5s - loss: 0.1821 - accuracy: 0.9459 - val_loss: 0.1416 -
val_accuracy: 0.9612 - 5s/epoch - 13ms/step
Epoch 13/15
422/422 - 5s - loss: 0.1719 - accuracy: 0.9498 - val_loss: 0.1349 -
val_accuracy: 0.9647 - 5s/epoch - 13ms/step
Epoch 14/15
422/422 - 5s - loss: 0.1641 - accuracy: 0.9521 - val_loss: 0.1291 -
val_accuracy: 0.9643 - 5s/epoch - 13ms/step
Epoch 15/15
422/422 - 5s - loss: 0.1556 - accuracy: 0.9549 - val_loss: 0.1265 -
val_accuracy: 0.9673 - 5s/epoch - 12ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 128 batch size and 15 epochs...
Confusion Matrix
```

```
[[ 964  0  2  1  0  5  5  1  2  0]
 [ 0 1125  2  1  0  1  4  0  2  0]
 [ 7  2 994  7  4  1  5  6  4  2]
 [ 0  0 13 957  0 18  1 10  9  2]
 [ 1  2  5  0 942  1  9  3  2 17]
 [ 5  2  0 10  1 860  8  2  3  1]
 [ 7  3  3  0  5  9 928  0  3  0]
 [ 1  7 24  5  2  1  0 978  0 10]
 [ 5  0  6 12  4 15  8  7 914  3]
 [ 9  7  2 14  17 10  1 12  3 934]]
```

Precision: 0.9597

Recall: 0.9596



Training model with hard_sigmoid activation function, 128 batch sizes, 20 epochs list

Epoch 1/20

422/422 - 6s - loss: 1.2454 - accuracy: 0.6732 - val_loss: 0.4837 - val_accuracy: 0.8883 - 6s/epoch - 14ms/step

Epoch 2/20

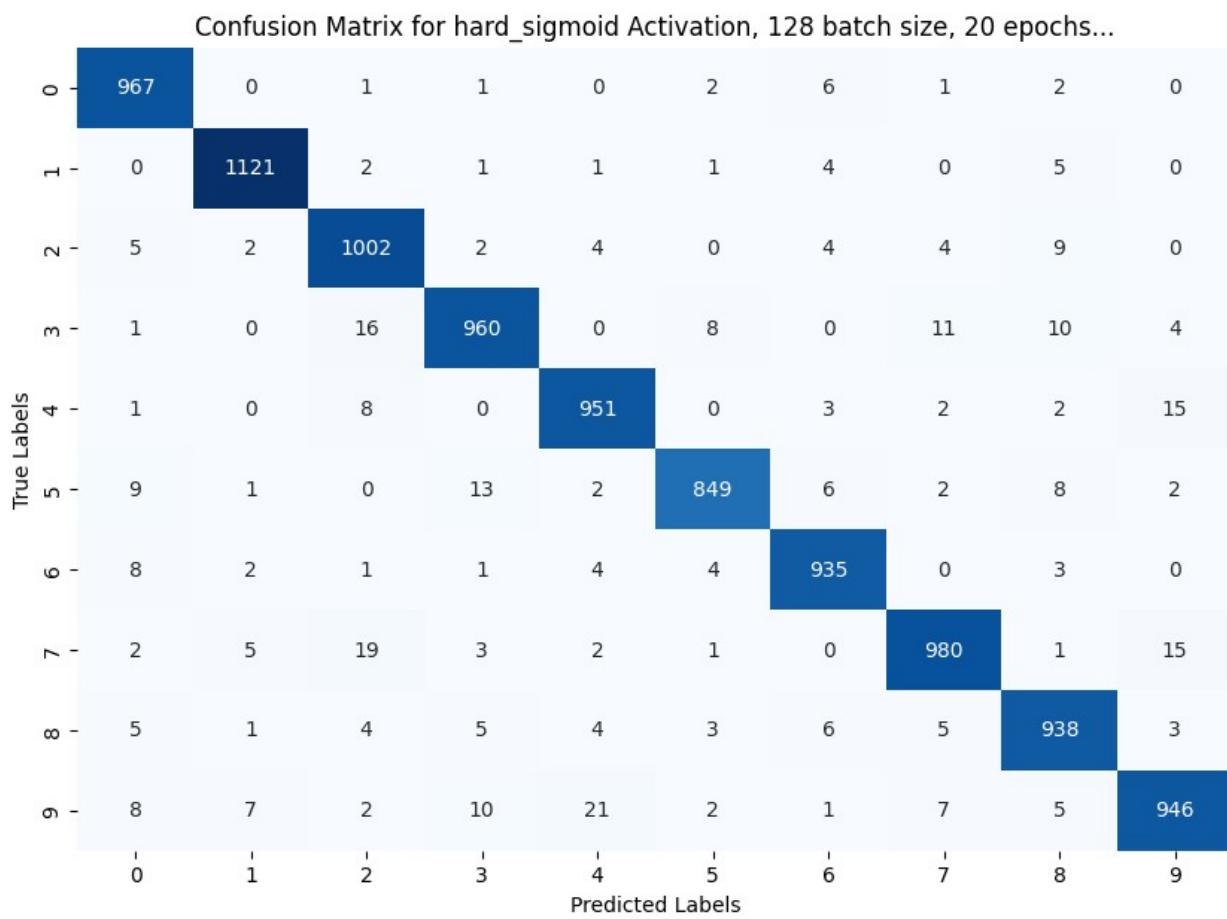
422/422 - 5s - loss: 0.4543 - accuracy: 0.8751 - val_loss: 0.3337 - val_accuracy: 0.9062 - 5s/epoch - 13ms/step

```
Epoch 3/20
422/422 - 5s - loss: 0.3637 - accuracy: 0.8942 - val_loss: 0.2755 -
val_accuracy: 0.9217 - 5s/epoch - 12ms/step
Epoch 4/20
422/422 - 5s - loss: 0.3210 - accuracy: 0.9064 - val_loss: 0.2472 -
val_accuracy: 0.9290 - 5s/epoch - 12ms/step
Epoch 5/20
422/422 - 5s - loss: 0.2921 - accuracy: 0.9127 - val_loss: 0.2168 -
val_accuracy: 0.9363 - 5s/epoch - 13ms/step
Epoch 6/20
422/422 - 5s - loss: 0.2697 - accuracy: 0.9200 - val_loss: 0.2020 -
val_accuracy: 0.9425 - 5s/epoch - 13ms/step
Epoch 7/20
422/422 - 5s - loss: 0.2504 - accuracy: 0.9261 - val_loss: 0.1926 -
val_accuracy: 0.9455 - 5s/epoch - 12ms/step
Epoch 8/20
422/422 - 5s - loss: 0.2340 - accuracy: 0.9305 - val_loss: 0.1763 -
val_accuracy: 0.9497 - 5s/epoch - 12ms/step
Epoch 9/20
422/422 - 5s - loss: 0.2194 - accuracy: 0.9350 - val_loss: 0.1714 -
val_accuracy: 0.9517 - 5s/epoch - 13ms/step
Epoch 10/20
422/422 - 5s - loss: 0.2072 - accuracy: 0.9377 - val_loss: 0.1768 -
val_accuracy: 0.9483 - 5s/epoch - 13ms/step
Epoch 11/20
422/422 - 5s - loss: 0.1950 - accuracy: 0.9424 - val_loss: 0.1544 -
val_accuracy: 0.9583 - 5s/epoch - 13ms/step
Epoch 12/20
422/422 - 5s - loss: 0.1858 - accuracy: 0.9448 - val_loss: 0.1447 -
val_accuracy: 0.9603 - 5s/epoch - 13ms/step
Epoch 13/20
422/422 - 6s - loss: 0.1756 - accuracy: 0.9482 - val_loss: 0.1339 -
val_accuracy: 0.9637 - 6s/epoch - 13ms/step
Epoch 14/20
422/422 - 5s - loss: 0.1691 - accuracy: 0.9504 - val_loss: 0.1297 -
val_accuracy: 0.9652 - 5s/epoch - 13ms/step
Epoch 15/20
422/422 - 5s - loss: 0.1606 - accuracy: 0.9528 - val_loss: 0.1367 -
val_accuracy: 0.9610 - 5s/epoch - 13ms/step
Epoch 16/20
422/422 - 6s - loss: 0.1541 - accuracy: 0.9539 - val_loss: 0.1219 -
val_accuracy: 0.9675 - 6s/epoch - 13ms/step
Epoch 17/20
422/422 - 5s - loss: 0.1472 - accuracy: 0.9567 - val_loss: 0.1181 -
val_accuracy: 0.9683 - 5s/epoch - 13ms/step
Epoch 18/20
422/422 - 5s - loss: 0.1398 - accuracy: 0.9586 - val_loss: 0.1138 -
val_accuracy: 0.9700 - 5s/epoch - 13ms/step
Epoch 19/20
```

```

422/422 - 5s - loss: 0.1345 - accuracy: 0.9603 - val_loss: 0.1065 -
val_accuracy: 0.9712 - 5s/epoch - 13ms/step
Epoch 20/20
422/422 - 5s - loss: 0.1300 - accuracy: 0.9619 - val_loss: 0.1060 -
val_accuracy: 0.9707 - 5s/epoch - 13ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 128 batch size and 20 epochs...
Confusion Matrix
[[ 967   0   1   1   0   2   6   1   2   0]
 [ 0 1121   2   1   1   1   4   0   5   0]
 [ 5  2 1002   2   4   0   4   4   9   0]
 [ 1   0  16 960   0   8   0  11 10   4]
 [ 1   0   8   0 951   0   3   2   2 15]
 [ 9   1   0  13   2 849   6   2   8   2]
 [ 8   2   1   1   4   4 935   0   3   0]
 [ 2   5  19   3   2   1   0 980   1 15]
 [ 5   1   4   5   4   3   6   5 938   3]
 [ 8   7   2  10  21   2   1   7   5 946]]
Precision: 0.9650
Recall: 0.9649

```



```
Training model with hard_sigmoid activation function, 256 batch sizes,
5 epochs_list
Epoch 1/5
211/211 - 5s - loss: 1.7704 - accuracy: 0.5429 - val_loss: 1.0087 -
val_accuracy: 0.8358 - 5s/epoch - 26ms/step
Epoch 2/5
211/211 - 5s - loss: 0.7371 - accuracy: 0.8243 - val_loss: 0.4778 -
val_accuracy: 0.8877 - 5s/epoch - 22ms/step
Epoch 3/5
211/211 - 5s - loss: 0.4886 - accuracy: 0.8695 - val_loss: 0.3685 -
val_accuracy: 0.9038 - 5s/epoch - 22ms/step
Epoch 4/5
211/211 - 5s - loss: 0.4083 - accuracy: 0.8871 - val_loss: 0.3128 -
val_accuracy: 0.9150 - 5s/epoch - 22ms/step
Epoch 5/5
211/211 - 5s - loss: 0.3667 - accuracy: 0.8951 - val_loss: 0.2844 -
val_accuracy: 0.9213 - 5s/epoch - 22ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 256 batch size and 5 epochs...
Confusion Matrix
[[ 959    0    3    1    0    8    4    1    4    0]
 [  0 1103    1    4    1    3    4    0   19    0]
 [ 15    4  890   13   23    3   19   22   38    5]
 [  3    2   22  885    0   46    2   20   26    4]
 [  3    3    4    1  925    2   10    1    7   26]
 [ 13    3    4   34   14  778   13    5   24    4]
 [ 22    3    9    0   20   24  877    0    3    0]
 [  4   13   27    5   12    0    0  931    5   31]
 [ 11    3    9   17   11   31    8   10  863   11]
 [ 16    5    7    9   59   15    0   21    8  869]]]
Precision: 0.9086
Recall: 0.9080
```

Confusion Matrix for hard_sigmoid Activation, 256 batch size, 5 epochs...											
	0	1	2	3	4	5	6	7	8	9	
0	959	0	3	1	0	8	4	1	4	0	0
1	0	1103	1	4	1	3	4	0	19	0	0
2	15	4	890	13	23	3	19	22	38	5	0
3	3	2	22	885	0	46	2	20	26	4	0
4	3	3	4	1	925	2	10	1	7	26	0
5	13	3	4	34	14	778	13	5	24	4	0
6	22	3	9	0	20	24	877	0	3	0	0
7	4	13	27	5	12	0	0	931	5	31	0
8	11	3	9	17	11	31	8	10	863	11	0
9	16	5	7	9	59	15	0	21	8	869	0
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

```

Training model with hard_sigmoid activation function, 256 batch sizes,
15 epochs_list
Epoch 1/15
211/211 - 5s - loss: 1.7532 - accuracy: 0.5558 - val_loss: 0.9777 -
val_accuracy: 0.8082 - 5s/epoch - 25ms/step
Epoch 2/15
211/211 - 5s - loss: 0.7198 - accuracy: 0.8282 - val_loss: 0.4751 -
val_accuracy: 0.8908 - 5s/epoch - 22ms/step
Epoch 3/15
211/211 - 4s - loss: 0.4844 - accuracy: 0.8702 - val_loss: 0.3627 -
val_accuracy: 0.9065 - 4s/epoch - 21ms/step
Epoch 4/15
211/211 - 4s - loss: 0.4082 - accuracy: 0.8853 - val_loss: 0.3079 -
val_accuracy: 0.9160 - 4s/epoch - 21ms/step
Epoch 5/15
211/211 - 4s - loss: 0.3660 - accuracy: 0.8948 - val_loss: 0.2841 -
val_accuracy: 0.9215 - 4s/epoch - 21ms/step
Epoch 6/15
211/211 - 4s - loss: 0.3402 - accuracy: 0.9011 - val_loss: 0.2668 -
val_accuracy: 0.9245 - 4s/epoch - 21ms/step
Epoch 7/15

```

```
211/211 - 4s - loss: 0.3192 - accuracy: 0.9061 - val_loss: 0.2562 -  
val_accuracy: 0.9270 - 4s/epoch - 21ms/step  
Epoch 8/15  
211/211 - 4s - loss: 0.3045 - accuracy: 0.9096 - val_loss: 0.2368 -  
val_accuracy: 0.9317 - 4s/epoch - 21ms/step  
Epoch 9/15  
211/211 - 4s - loss: 0.2887 - accuracy: 0.9149 - val_loss: 0.2290 -  
val_accuracy: 0.9328 - 4s/epoch - 21ms/step  
Epoch 10/15  
211/211 - 4s - loss: 0.2791 - accuracy: 0.9177 - val_loss: 0.2168 -  
val_accuracy: 0.9370 - 4s/epoch - 21ms/step  
Epoch 11/15  
211/211 - 4s - loss: 0.2676 - accuracy: 0.9216 - val_loss: 0.2077 -  
val_accuracy: 0.9432 - 4s/epoch - 21ms/step  
Epoch 12/15  
211/211 - 4s - loss: 0.2571 - accuracy: 0.9241 - val_loss: 0.2057 -  
val_accuracy: 0.9385 - 4s/epoch - 21ms/step  
Epoch 13/15  
211/211 - 4s - loss: 0.2468 - accuracy: 0.9272 - val_loss: 0.1926 -  
val_accuracy: 0.9428 - 4s/epoch - 21ms/step  
Epoch 14/15  
211/211 - 5s - loss: 0.2376 - accuracy: 0.9294 - val_loss: 0.1854 -  
val_accuracy: 0.9483 - 5s/epoch - 21ms/step  
Epoch 15/15  
211/211 - 5s - loss: 0.2308 - accuracy: 0.9317 - val_loss: 0.1824 -  
val_accuracy: 0.9473 - 5s/epoch - 22ms/step  
313/313 [=====] - 1s 3ms/step  
Results for hard_sigmoid function, 256 batch size and 15 epochs...  
Confusion Matrix  
[[ 966  0  1  1  0  4  4  1  3  0]  
[  0 1113  3  2  0  1  4  1  11  0]  
[ 11  0 946 13 13  1 17 12 18  1]  
[  2  1 10 943  0 17  2 13 18  4]  
[  1  3  2  1 941  1 14  2  3 14]  
[  8  3  1 35  4 802 15  4 16  4]  
[ 10  3  3  1  4  8 926  1  2  0]  
[  5  8 24  9  7  0  0 965  0 10]  
[ 12  2  4 16  7 17 12  9 892  3]  
[ 14  7  1 14 47  7  1 27  4 887]]  
Precision: 0.9385  
Recall: 0.9381
```

Confusion Matrix for hard_sigmoid Activation, 256 batch size, 15 epochs...											
	0	1	2	3	4	5	6	7	8	9	
0	966	0	1	1	0	4	4	1	3	0	
1	0	1113	3	2	0	1	4	1	11	0	
2	11	0	946	13	13	1	17	12	18	1	
3	2	1	10	943	0	17	2	13	18	4	
4	1	3	2	1	941	1	14	2	3	14	
5	8	3	1	35	4	802	15	4	16	4	
6	10	3	3	1	4	8	926	1	2	0	
7	5	8	24	9	7	0	0	965	0	10	
8	12	2	4	16	7	17	12	9	892	3	
9	14	7	1	14	47	7	1	27	4	887	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```

Training model with hard_sigmoid activation function, 256 batch sizes,
20 epochs_list
Epoch 1/20
211/211 - 5s - loss: 1.6930 - accuracy: 0.5679 - val_loss: 0.9186 -
val_accuracy: 0.8115 - 5s/epoch - 24ms/step
Epoch 2/20
211/211 - 4s - loss: 0.6984 - accuracy: 0.8324 - val_loss: 0.4664 -
val_accuracy: 0.8915 - 4s/epoch - 21ms/step
Epoch 3/20
211/211 - 4s - loss: 0.4809 - accuracy: 0.8717 - val_loss: 0.3586 -
val_accuracy: 0.9092 - 4s/epoch - 21ms/step
Epoch 4/20
211/211 - 4s - loss: 0.4067 - accuracy: 0.8857 - val_loss: 0.3100 -
val_accuracy: 0.9183 - 4s/epoch - 21ms/step
Epoch 5/20
211/211 - 4s - loss: 0.3678 - accuracy: 0.8950 - val_loss: 0.2835 -
val_accuracy: 0.9225 - 4s/epoch - 21ms/step
Epoch 6/20
211/211 - 4s - loss: 0.3421 - accuracy: 0.9005 - val_loss: 0.2633 -
val_accuracy: 0.9250 - 4s/epoch - 21ms/step
Epoch 7/20

```

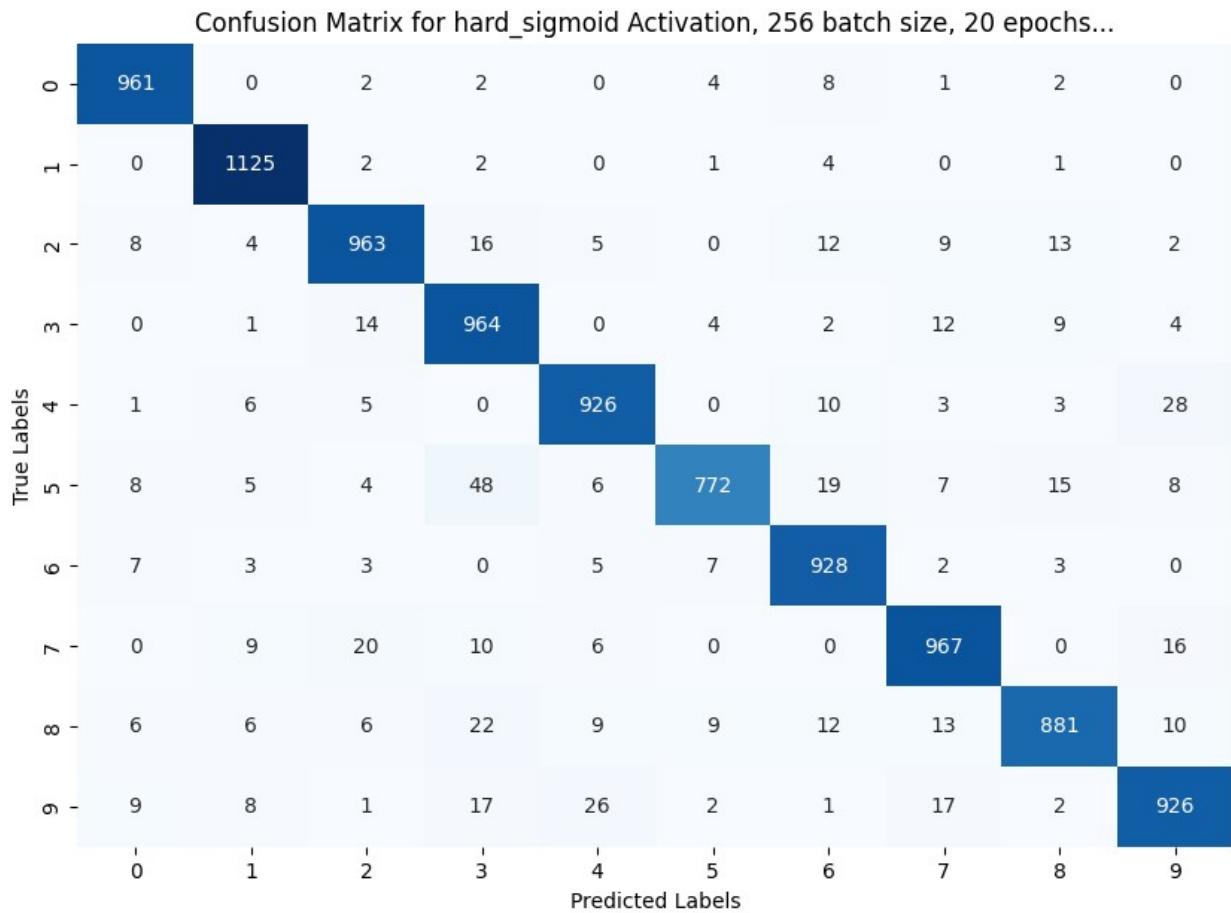
```

211/211 - 4s - loss: 0.3232 - accuracy: 0.9064 - val_loss: 0.2486 -
val_accuracy: 0.9293 - 4s/epoch - 21ms/step
Epoch 8/20
211/211 - 5s - loss: 0.3073 - accuracy: 0.9095 - val_loss: 0.2398 -
val_accuracy: 0.9312 - 5s/epoch - 21ms/step
Epoch 9/20
211/211 - 4s - loss: 0.2931 - accuracy: 0.9141 - val_loss: 0.2259 -
val_accuracy: 0.9358 - 4s/epoch - 21ms/step
Epoch 10/20
211/211 - 5s - loss: 0.2804 - accuracy: 0.9176 - val_loss: 0.2213 -
val_accuracy: 0.9317 - 5s/epoch - 21ms/step
Epoch 11/20
211/211 - 4s - loss: 0.2696 - accuracy: 0.9207 - val_loss: 0.2077 -
val_accuracy: 0.9412 - 4s/epoch - 21ms/step
Epoch 12/20
211/211 - 4s - loss: 0.2596 - accuracy: 0.9241 - val_loss: 0.2001 -
val_accuracy: 0.9420 - 4s/epoch - 20ms/step
Epoch 13/20
211/211 - 4s - loss: 0.2525 - accuracy: 0.9254 - val_loss: 0.1953 -
val_accuracy: 0.9450 - 4s/epoch - 21ms/step
Epoch 14/20
211/211 - 4s - loss: 0.2425 - accuracy: 0.9284 - val_loss: 0.1879 -
val_accuracy: 0.9462 - 4s/epoch - 21ms/step
Epoch 15/20
211/211 - 4s - loss: 0.2350 - accuracy: 0.9303 - val_loss: 0.1804 -
val_accuracy: 0.9487 - 4s/epoch - 21ms/step
Epoch 16/20
211/211 - 4s - loss: 0.2288 - accuracy: 0.9325 - val_loss: 0.1765 -
val_accuracy: 0.9488 - 4s/epoch - 21ms/step
Epoch 17/20
211/211 - 4s - loss: 0.2202 - accuracy: 0.9346 - val_loss: 0.1790 -
val_accuracy: 0.9465 - 4s/epoch - 21ms/step
Epoch 18/20
211/211 - 4s - loss: 0.2159 - accuracy: 0.9354 - val_loss: 0.1668 -
val_accuracy: 0.9525 - 4s/epoch - 21ms/step
Epoch 19/20
211/211 - 4s - loss: 0.2077 - accuracy: 0.9385 - val_loss: 0.1741 -
val_accuracy: 0.9497 - 4s/epoch - 21ms/step
Epoch 20/20
211/211 - 4s - loss: 0.2022 - accuracy: 0.9397 - val_loss: 0.1664 -
val_accuracy: 0.9510 - 4s/epoch - 21ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 256 batch size and 20 epochs...
Confusion Matrix
[[ 961   0   2   2   0   4   8   1   2   0]
 [  0 1125   2   2   0   1   4   0   1   0]
 [  8   4  963  16   5   0  12   9  13   2]
 [  0   1  14  964   0   4   2  12   9   4]
 [  1   6   5   0  926   0  10   3   3  28]]
```

```
[ 8  5  4  48  6 772  19   7  15   8]
[ 7  3  3  0   5   7 928   2   3   0]
[ 0  9  20 10   6   0   0 967   0 16]
[ 6  6  6 22   9   9 12 13 881 10]
[ 9  8  1 17  26   2   1 17   2 926]]
```

Precision: 0.9418

Recall: 0.9413



Training model with hard_sigmoid activation function, 16 batch sizes, 5 epochs_list

Epoch 1/5

3375/3375 - 20s - loss: 0.5464 - accuracy: 0.8297 - val_loss: 0.2336 - val_accuracy: 0.9293 - 20s/epoch - 6ms/step

Epoch 2/5

3375/3375 - 19s - loss: 0.2377 - accuracy: 0.9274 - val_loss: 0.1765 - val_accuracy: 0.9470 - 19s/epoch - 6ms/step

Epoch 3/5

3375/3375 - 20s - loss: 0.1676 - accuracy: 0.9498 - val_loss: 0.1382 - val_accuracy: 0.9588 - 20s/epoch - 6ms/step

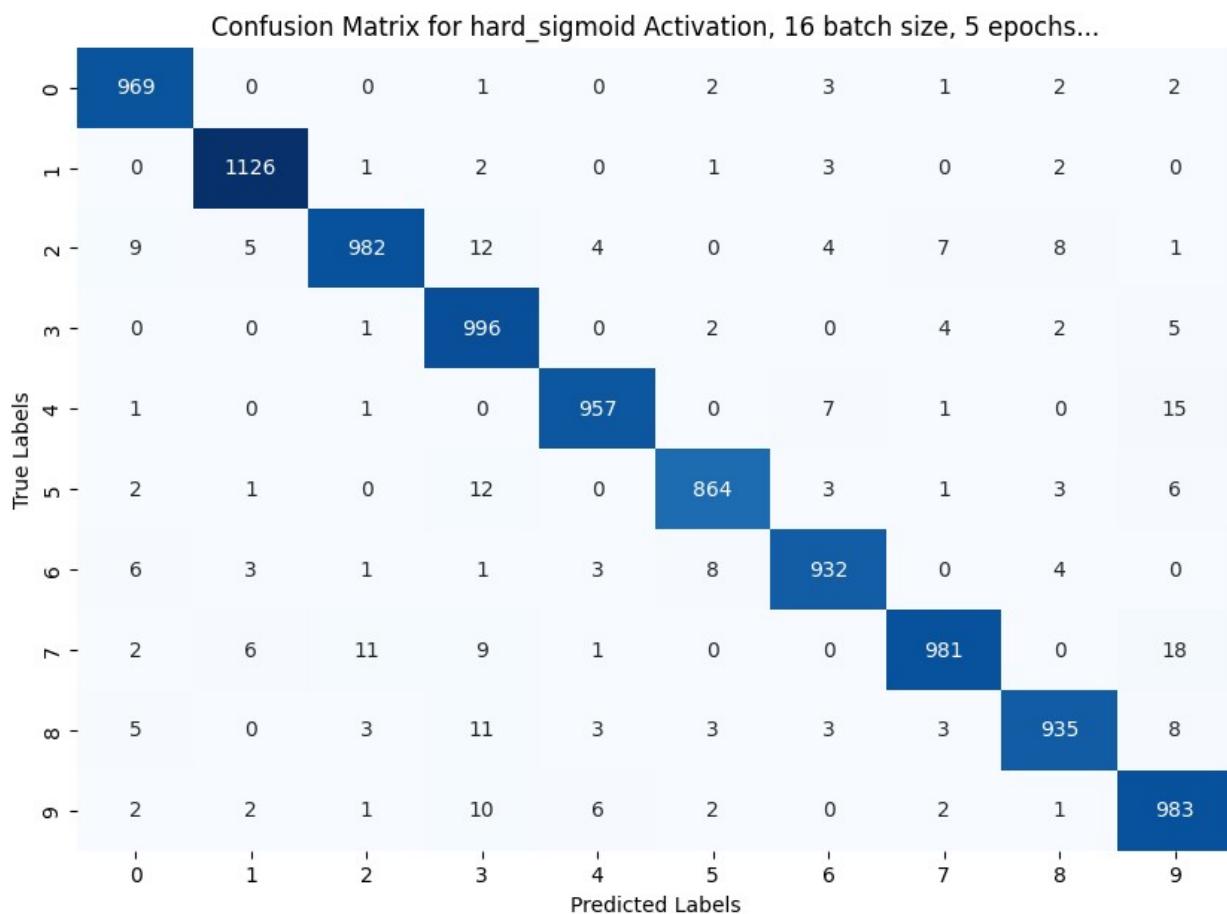
Epoch 4/5

3375/3375 - 21s - loss: 0.1271 - accuracy: 0.9615 - val_loss: 0.0935 -

```

val_accuracy: 0.9735 - 21s/epoch - 6ms/step
Epoch 5/5
3375/3375 - 20s - loss: 0.1015 - accuracy: 0.9699 - val_loss: 0.0799 -
val_accuracy: 0.9782 - 20s/epoch - 6ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 16 batch size and 5 epochs...
Confusion Matrix
[[ 969 0 0 1 0 2 3 1 2 2]
 [ 0 1126 1 2 0 1 3 0 2 0]
 [ 9 5 982 12 4 0 4 7 8 1]
 [ 0 0 1 996 0 2 0 4 2 5]
 [ 1 0 1 0 957 0 7 1 0 15]
 [ 2 1 0 12 0 864 3 1 3 6]
 [ 6 3 1 1 3 8 932 0 4 0]
 [ 2 6 11 9 1 0 0 981 0 18]
 [ 5 0 3 11 3 3 3 3 935 8]
 [ 2 2 1 10 6 2 0 2 1 983]]
Precision: 0.9728
Recall: 0.9725

```



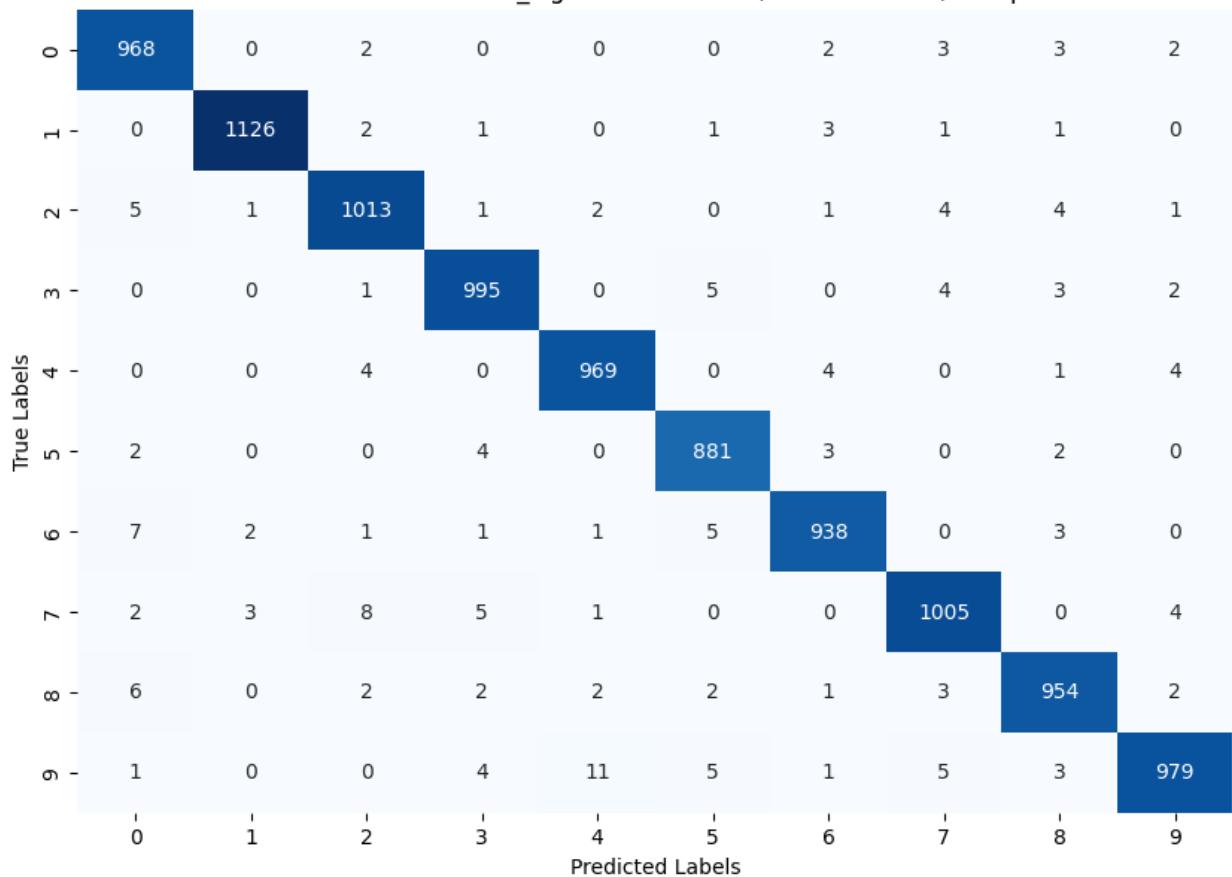
```
Training model with hard_sigmoid activation function, 16 batch sizes,
15 epochs_list
Epoch 1/15
3375/3375 - 20s - loss: 0.5610 - accuracy: 0.8256 - val_loss: 0.2041 -
val_accuracy: 0.9392 - 20s/epoch - 6ms/step
Epoch 2/15
3375/3375 - 19s - loss: 0.2416 - accuracy: 0.9257 - val_loss: 0.1598 -
val_accuracy: 0.9508 - 19s/epoch - 6ms/step
Epoch 3/15
3375/3375 - 20s - loss: 0.1739 - accuracy: 0.9469 - val_loss: 0.1105 -
val_accuracy: 0.9693 - 20s/epoch - 6ms/step
Epoch 4/15
3375/3375 - 20s - loss: 0.1344 - accuracy: 0.9590 - val_loss: 0.0984 -
val_accuracy: 0.9727 - 20s/epoch - 6ms/step
Epoch 5/15
3375/3375 - 20s - loss: 0.1088 - accuracy: 0.9669 - val_loss: 0.0904 -
val_accuracy: 0.9740 - 20s/epoch - 6ms/step
Epoch 6/15
3375/3375 - 20s - loss: 0.0902 - accuracy: 0.9730 - val_loss: 0.0800 -
val_accuracy: 0.9773 - 20s/epoch - 6ms/step
Epoch 7/15
3375/3375 - 20s - loss: 0.0765 - accuracy: 0.9771 - val_loss: 0.0727 -
val_accuracy: 0.9802 - 20s/epoch - 6ms/step
Epoch 8/15
3375/3375 - 22s - loss: 0.0651 - accuracy: 0.9807 - val_loss: 0.0653 -
val_accuracy: 0.9803 - 22s/epoch - 6ms/step
Epoch 9/15
3375/3375 - 20s - loss: 0.0566 - accuracy: 0.9833 - val_loss: 0.0623 -
val_accuracy: 0.9822 - 20s/epoch - 6ms/step
Epoch 10/15
3375/3375 - 20s - loss: 0.0496 - accuracy: 0.9862 - val_loss: 0.0616 -
val_accuracy: 0.9817 - 20s/epoch - 6ms/step
Epoch 11/15
3375/3375 - 20s - loss: 0.0432 - accuracy: 0.9883 - val_loss: 0.0532 -
val_accuracy: 0.9863 - 20s/epoch - 6ms/step
Epoch 12/15
3375/3375 - 20s - loss: 0.0386 - accuracy: 0.9891 - val_loss: 0.0535 -
val_accuracy: 0.9847 - 20s/epoch - 6ms/step
Epoch 13/15
3375/3375 - 19s - loss: 0.0340 - accuracy: 0.9903 - val_loss: 0.0598 -
val_accuracy: 0.9830 - 19s/epoch - 6ms/step
Epoch 14/15
3375/3375 - 20s - loss: 0.0311 - accuracy: 0.9915 - val_loss: 0.0490 -
val_accuracy: 0.9875 - 20s/epoch - 6ms/step
Epoch 15/15
3375/3375 - 20s - loss: 0.0274 - accuracy: 0.9927 - val_loss: 0.0490 -
val_accuracy: 0.9872 - 20s/epoch - 6ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 16 batch size and 15 epochs...
Confusion Matrix
```

```
[[ 968  0  2  0  0  0  2  3  3  2]
 [ 0 1126  2  1  0  1  3  1  1  0]
 [ 5  1 1013  1  2  0  1  4  4  1]
 [ 0  0  1  995  0  5  0  4  3  2]
 [ 0  0  4  0  969  0  4  0  1  4]
 [ 2  0  0  4  0  881  3  0  2  0]
 [ 7  2  1  1  1  5  938  0  3  0]
 [ 2  3  8  5  1  0  0 1005  0  4]
 [ 6  0  2  2  2  2  1  3  954  2]
 [ 1  0  0  4  11  5  1  5  3  979]]
```

Precision: 0.9828

Recall: 0.9828

Confusion Matrix for hard_sigmoid Activation, 16 batch size, 15 epochs...



Training model with hard_sigmoid activation function, 16 batch sizes, 20 epochs list

Epoch 1/20

3375/3375 - 21s - loss: 0.6191 - accuracy: 0.8095 - val_loss: 0.2414 - val_accuracy: 0.9275 - 21s/epoch - 6ms/step

Epoch 2/20

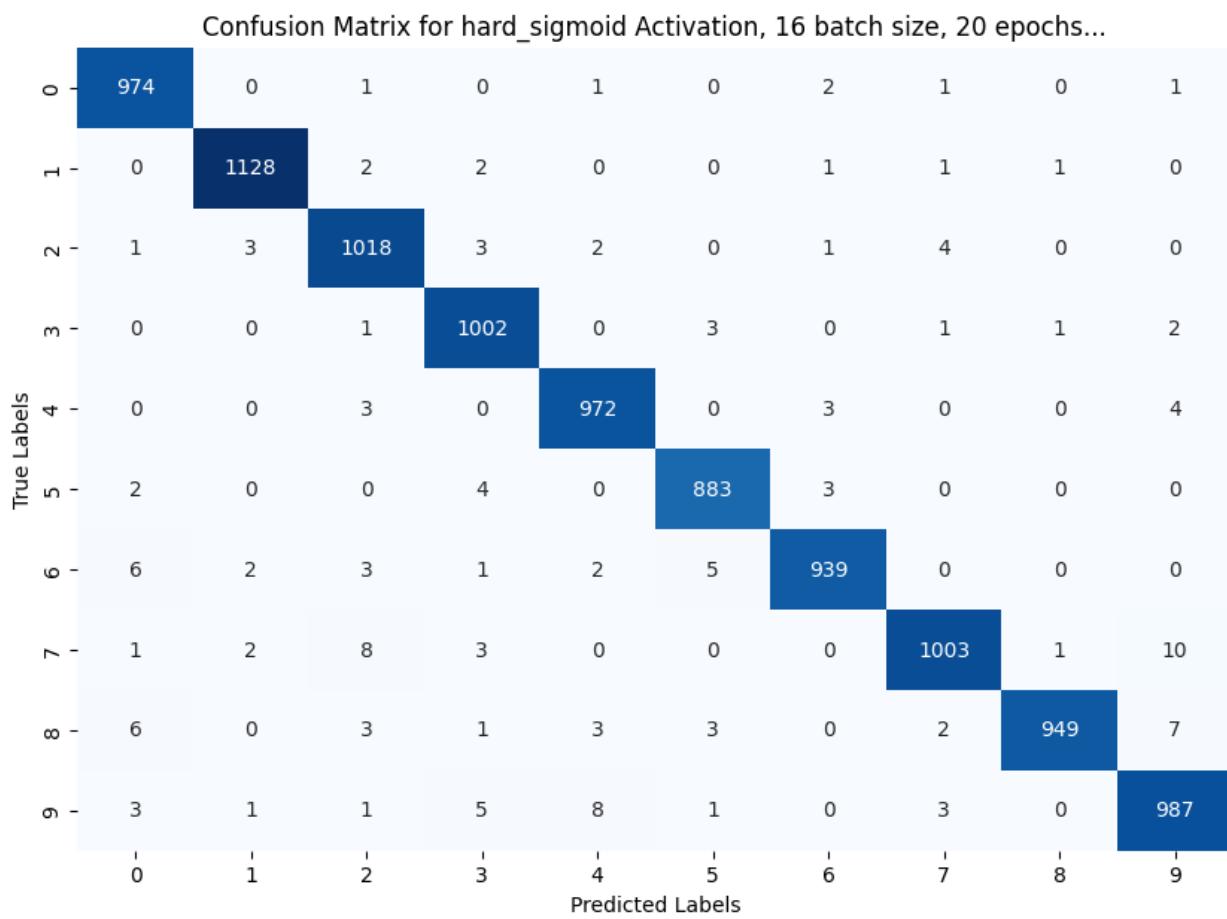
3375/3375 - 20s - loss: 0.2640 - accuracy: 0.9218 - val_loss: 0.1965 - val_accuracy: 0.9425 - 20s/epoch - 6ms/step

```
Epoch 3/20
3375/3375 - 20s - loss: 0.2005 - accuracy: 0.9403 - val_loss: 0.1579 -
val_accuracy: 0.9553 - 20s/epoch - 6ms/step
Epoch 4/20
3375/3375 - 20s - loss: 0.1546 - accuracy: 0.9538 - val_loss: 0.1101 -
val_accuracy: 0.9710 - 20s/epoch - 6ms/step
Epoch 5/20
3375/3375 - 20s - loss: 0.1253 - accuracy: 0.9628 - val_loss: 0.0955 -
val_accuracy: 0.9728 - 20s/epoch - 6ms/step
Epoch 6/20
3375/3375 - 20s - loss: 0.1023 - accuracy: 0.9695 - val_loss: 0.0958 -
val_accuracy: 0.9717 - 20s/epoch - 6ms/step
Epoch 7/20
3375/3375 - 21s - loss: 0.0852 - accuracy: 0.9756 - val_loss: 0.0775 -
val_accuracy: 0.9785 - 21s/epoch - 6ms/step
Epoch 8/20
3375/3375 - 21s - loss: 0.0728 - accuracy: 0.9788 - val_loss: 0.0827 -
val_accuracy: 0.9760 - 21s/epoch - 6ms/step
Epoch 9/20
3375/3375 - 21s - loss: 0.0626 - accuracy: 0.9816 - val_loss: 0.0699 -
val_accuracy: 0.9795 - 21s/epoch - 6ms/step
Epoch 10/20
3375/3375 - 20s - loss: 0.0547 - accuracy: 0.9839 - val_loss: 0.0662 -
val_accuracy: 0.9822 - 20s/epoch - 6ms/step
Epoch 11/20
3375/3375 - 20s - loss: 0.0491 - accuracy: 0.9858 - val_loss: 0.0612 -
val_accuracy: 0.9835 - 20s/epoch - 6ms/step
Epoch 12/20
3375/3375 - 21s - loss: 0.0429 - accuracy: 0.9876 - val_loss: 0.0586 -
val_accuracy: 0.9847 - 21s/epoch - 6ms/step
Epoch 13/20
3375/3375 - 20s - loss: 0.0378 - accuracy: 0.9897 - val_loss: 0.0588 -
val_accuracy: 0.9843 - 20s/epoch - 6ms/step
Epoch 14/20
3375/3375 - 20s - loss: 0.0347 - accuracy: 0.9907 - val_loss: 0.0553 -
val_accuracy: 0.9868 - 20s/epoch - 6ms/step
Epoch 15/20
3375/3375 - 20s - loss: 0.0311 - accuracy: 0.9917 - val_loss: 0.0526 -
val_accuracy: 0.9855 - 20s/epoch - 6ms/step
Epoch 16/20
3375/3375 - 21s - loss: 0.0280 - accuracy: 0.9929 - val_loss: 0.0530 -
val_accuracy: 0.9865 - 21s/epoch - 6ms/step
Epoch 17/20
3375/3375 - 20s - loss: 0.0252 - accuracy: 0.9936 - val_loss: 0.0503 -
val_accuracy: 0.9867 - 20s/epoch - 6ms/step
Epoch 18/20
3375/3375 - 20s - loss: 0.0222 - accuracy: 0.9945 - val_loss: 0.0513 -
val_accuracy: 0.9867 - 20s/epoch - 6ms/step
Epoch 19/20
```

```

3375/3375 - 21s - loss: 0.0205 - accuracy: 0.9951 - val_loss: 0.0508 -
val_accuracy: 0.9863 - 21s/epoch - 6ms/step
Epoch 20/20
3375/3375 - 20s - loss: 0.0183 - accuracy: 0.9958 - val_loss: 0.0504 -
val_accuracy: 0.9873 - 20s/epoch - 6ms/step
313/313 [=====] - 1s 3ms/step
Results for hard_sigmoid function, 16 batch size and 20 epochs...
Confusion Matrix
[[ 974   0   1   0   1   0   2   1   0   1]
 [ 0 1128   2   2   0   0   1   1   1   0]
 [ 1   3 1018   3   2   0   1   4   0   0]
 [ 0   0   1 1002   0   3   0   1   1   2]
 [ 0   0   3   0  972   0   3   0   0   4]
 [ 2   0   0   4   0  883   3   0   0   0]
 [ 6   2   3   1   2   5  939   0   0   0]
 [ 1   2   8   3   0   0   0 1003   1   10]
 [ 6   0   3   1   3   3   0   2  949   7]
 [ 3   1   1   5   8   1   0   3   0  987]]
Precision: 0.9855
Recall: 0.9855

```



Observations:

ReLU Activation: Exhibits strong and consistent performance across a range of batch sizes and epochs, with precision and recall metrics generally above 0.98, signifying excellent model accuracy. The highest performance is noted when the batch size is small (16) and epochs are high (20), indicating an optimal combination for this activation function.

Sigmoid Activation: Shows good performance, but with a slight decrease compared to ReLU. Precision and recall are generally in the high 0.90s but drop notably when batch sizes increase to 256. This activation function seems less robust against changes in batch size and epochs, suggesting a potential sensitivity to training regimen.

Softmax Activation: Demonstrates significantly lower precision and recall compared to ReLU and Sigmoid, particularly when used in internal layers. This is expected as Softmax is typically used in the output layer for classification tasks due to its probabilistic output. The decrease in performance is drastic with larger batch sizes, highlighting its inadequacy for internal layer activation in the tested configurations.

Exponential Activation: Yields the poorest performance, with precision and recall metrics significantly lower than other activation functions, often near 0.01, indicating nearly no classification capability. This result underscores the challenges with the exponential function, likely due to its contribution to exploding gradients.

Hard Sigmoid Activation: Presents varied results, with some configurations achieving precision and recall metrics in the high 0.90s, yet performance deteriorates in specific settings (notably with certain batch sizes and epochs). This suggests that while Hard Sigmoid can perform well, it may require careful tuning of batch sizes and epochs for optimal performance.

###Experiment with Various Activation Function Choices with Different Network Size, Depth, Epoch and Batch Sizes:

```
act_func = ["relu", "sigmoid", "softmax"]
layers_configs = [(1,1),(2,2),(3,3)] ## Tuple format:
(num_conv_layers, num_dense_layers)
neuron_configs = [64, 128, 256]
batch_sizes = [64, 128, 256]
epochs_list = [5, 15, 20]

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten,
Dense, Activation

def create_flexible_model(conv_activation = "relu", dense_activation =
"relu", num_conv_layers = 1, num_dense_layers = 1, num_neurons = 100):
    model = Sequential()
    ### add specified number of convolutional layers

    for _ in range(num_conv_layers):
        model.add(Conv2D(32,(3,3), activation = conv_activation,
kernel_initializer="he_uniform", input_shape=(28, 28, 1)))
        model.add(MaxPooling2D((2,2)))

    model.add(Flatten())
```



```

        print(f"Recall: {recall:.4f}")

    # Plotting the confusion matrix
    plt.figure(figsize=(10, 7))
    sns.heatmap(cm, annot=True, fmt="g", cmap="Blues",
cbar=False)
    plt.xlabel("Predicted Labels")
    plt.ylabel("True Labels")
    plt.title(f"Confusion Matrix for {act} Activation")
    plt.show()

Training Model with relu activation, 1 conv_layers, 1 dense layers, 64
batch size, 5 epochs..
Epoch 1/5
844/844 - 5s - loss: 0.2159 - accuracy: 0.9336 - val_loss: 0.0770 -
val_accuracy: 0.9815 - 5s/epoch - 6ms/step
Epoch 2/5
844/844 - 4s - loss: 0.0798 - accuracy: 0.9763 - val_loss: 0.0652 -
val_accuracy: 0.9820 - 4s/epoch - 5ms/step
Epoch 3/5
844/844 - 4s - loss: 0.0541 - accuracy: 0.9844 - val_loss: 0.0538 -
val_accuracy: 0.9855 - 4s/epoch - 5ms/step
Epoch 4/5
844/844 - 4s - loss: 0.0409 - accuracy: 0.9879 - val_loss: 0.0528 -
val_accuracy: 0.9870 - 4s/epoch - 5ms/step
Epoch 5/5
844/844 - 4s - loss: 0.0316 - accuracy: 0.9907 - val_loss: 0.0514 -
val_accuracy: 0.9863 - 4s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 976   1   0   0   1   0   1   1   0   0]
 [  0 1132   1   1   0   0   1   0   0   0]
 [  8   8 1003   1   2   0   1   5   4   0]
 [  1   0   2  989   0   7   0   2   1   8]
 [  0   0   0   0  976   0   0   0   0   6]
 [  2   0   0   3   0  880   2   0   1   4]
 [ 11   3   0   0   3   8  932   0   1   0]
 [  1   5   6   2   3   0   0  996   0  15]
 [ 11   0   2   0   4   2   0   1  943  11]
 [  1   3   0   1   9   2   0   4   3  986]]
Precision: 0.9815
Recall: 0.9813

```

Confusion Matrix for relu Activation										
	0	1	2	3	4	5	6	7	8	9
True Labels	976	1	0	0	1	0	1	1	0	0
0	976	1	0	0	1	0	0	1	0	0
1	0	1132	1	1	0	0	1	0	0	0
2	8	8	1003	1	2	0	1	5	4	0
3	1	0	2	989	0	7	0	2	1	8
4	0	0	0	0	976	0	0	0	0	6
5	2	0	0	3	0	880	2	0	1	4
6	11	3	0	0	3	8	932	0	1	0
7	1	5	6	2	3	0	0	996	0	15
8	11	0	2	0	4	2	0	1	943	11
9	1	3	0	1	9	2	0	4	3	986
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training Model with relu activation, 1 conv_layers, 1 dense layers, 64
batch size, 15 epochs..
Epoch 1/15
844/844 - 5s - loss: 0.2476 - accuracy: 0.9239 - val_loss: 0.0953 -
val_accuracy: 0.9753 - 5s/epoch - 6ms/step
Epoch 2/15
844/844 - 4s - loss: 0.0994 - accuracy: 0.9703 - val_loss: 0.0670 -
val_accuracy: 0.9808 - 4s/epoch - 5ms/step
Epoch 3/15
844/844 - 4s - loss: 0.0635 - accuracy: 0.9812 - val_loss: 0.0623 -
val_accuracy: 0.9823 - 4s/epoch - 5ms/step
Epoch 4/15
844/844 - 4s - loss: 0.0460 - accuracy: 0.9857 - val_loss: 0.0487 -
val_accuracy: 0.9880 - 4s/epoch - 5ms/step
Epoch 5/15
844/844 - 4s - loss: 0.0349 - accuracy: 0.9897 - val_loss: 0.0499 -
val_accuracy: 0.9863 - 4s/epoch - 5ms/step
Epoch 6/15
844/844 - 4s - loss: 0.0277 - accuracy: 0.9919 - val_loss: 0.0428 -
val_accuracy: 0.9882 - 4s/epoch - 5ms/step
Epoch 7/15
```

```
844/844 - 4s - loss: 0.0222 - accuracy: 0.9938 - val_loss: 0.0467 -  
val_accuracy: 0.9865 - 4s/epoch - 5ms/step  
Epoch 8/15  
844/844 - 4s - loss: 0.0176 - accuracy: 0.9953 - val_loss: 0.0411 -  
val_accuracy: 0.9897 - 4s/epoch - 5ms/step  
Epoch 9/15  
844/844 - 4s - loss: 0.0141 - accuracy: 0.9962 - val_loss: 0.0455 -  
val_accuracy: 0.9880 - 4s/epoch - 5ms/step  
Epoch 10/15  
844/844 - 4s - loss: 0.0117 - accuracy: 0.9968 - val_loss: 0.0451 -  
val_accuracy: 0.9877 - 4s/epoch - 5ms/step  
Epoch 11/15  
844/844 - 4s - loss: 0.0089 - accuracy: 0.9981 - val_loss: 0.0450 -  
val_accuracy: 0.9887 - 4s/epoch - 5ms/step  
Epoch 12/15  
844/844 - 4s - loss: 0.0075 - accuracy: 0.9983 - val_loss: 0.0404 -  
val_accuracy: 0.9895 - 4s/epoch - 5ms/step  
Epoch 13/15  
844/844 - 4s - loss: 0.0056 - accuracy: 0.9991 - val_loss: 0.0443 -  
val_accuracy: 0.9882 - 4s/epoch - 5ms/step  
Epoch 14/15  
844/844 - 4s - loss: 0.0051 - accuracy: 0.9990 - val_loss: 0.0461 -  
val_accuracy: 0.9893 - 4s/epoch - 5ms/step  
Epoch 15/15  
844/844 - 4s - loss: 0.0034 - accuracy: 0.9997 - val_loss: 0.0460 -  
val_accuracy: 0.9888 - 4s/epoch - 5ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 974  0  0  0  0  0  2  1  2  1]  
[  0 1130  2  1  0  0  1  1  0  0]  
[  1   3 1017  1  1  0  0  5  4  0]  
[  0   0   0 999  0  5  0  2  4  0]  
[  0   0   1  0 977  0  1  0  0  3]  
[  2   0   1   8  0 873  6  0  1  1]  
[  3   2   0   0  2   2 949  0  0  0]  
[  0   0   7   2  0   0   0 1016  1  2]  
[  6   0   1   0  2   2   0   3 957  3]  
[  1   0   0   4  6   2   1   6   3 986]]  
Precision: 0.9878  
Recall: 0.9878
```

Confusion Matrix for relu Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	974	0	0	0	0	0	2	1	2	1	Predicted Labels
0	974	0	0	0	0	0	2	1	2	1	0
1	0	1130	2	1	0	0	1	1	0	0	1
2	1	3	1017	1	1	0	0	5	4	0	2
3	0	0	0	999	0	5	0	2	4	0	3
4	0	0	1	0	977	0	1	0	0	3	1
5	2	0	1	8	0	873	6	0	1	1	1
6	3	2	0	0	2	2	949	0	0	0	0
7	0	0	7	2	0	0	0	0	1016	1	2
8	6	0	1	0	2	2	0	3	957	3	3
9	1	0	0	4	6	2	1	6	3	986	9

```
Training Model with relu activation, 1 conv_layers, 1 dense layers, 64
batch size, 20 epochs..
Epoch 1/20
844/844 - 5s - loss: 0.2192 - accuracy: 0.9337 - val_loss: 0.0877 -
val_accuracy: 0.9762 - 5s/epoch - 6ms/step
Epoch 2/20
844/844 - 4s - loss: 0.0809 - accuracy: 0.9760 - val_loss: 0.0633 -
val_accuracy: 0.9835 - 4s/epoch - 5ms/step
Epoch 3/20
844/844 - 4s - loss: 0.0549 - accuracy: 0.9841 - val_loss: 0.0546 -
val_accuracy: 0.9845 - 4s/epoch - 5ms/step
Epoch 4/20
844/844 - 4s - loss: 0.0410 - accuracy: 0.9878 - val_loss: 0.0605 -
val_accuracy: 0.9827 - 4s/epoch - 5ms/step
Epoch 5/20
844/844 - 4s - loss: 0.0322 - accuracy: 0.9907 - val_loss: 0.0496 -
val_accuracy: 0.9872 - 4s/epoch - 5ms/step
Epoch 6/20
844/844 - 4s - loss: 0.0262 - accuracy: 0.9922 - val_loss: 0.0445 -
val_accuracy: 0.9882 - 4s/epoch - 5ms/step
Epoch 7/20
```

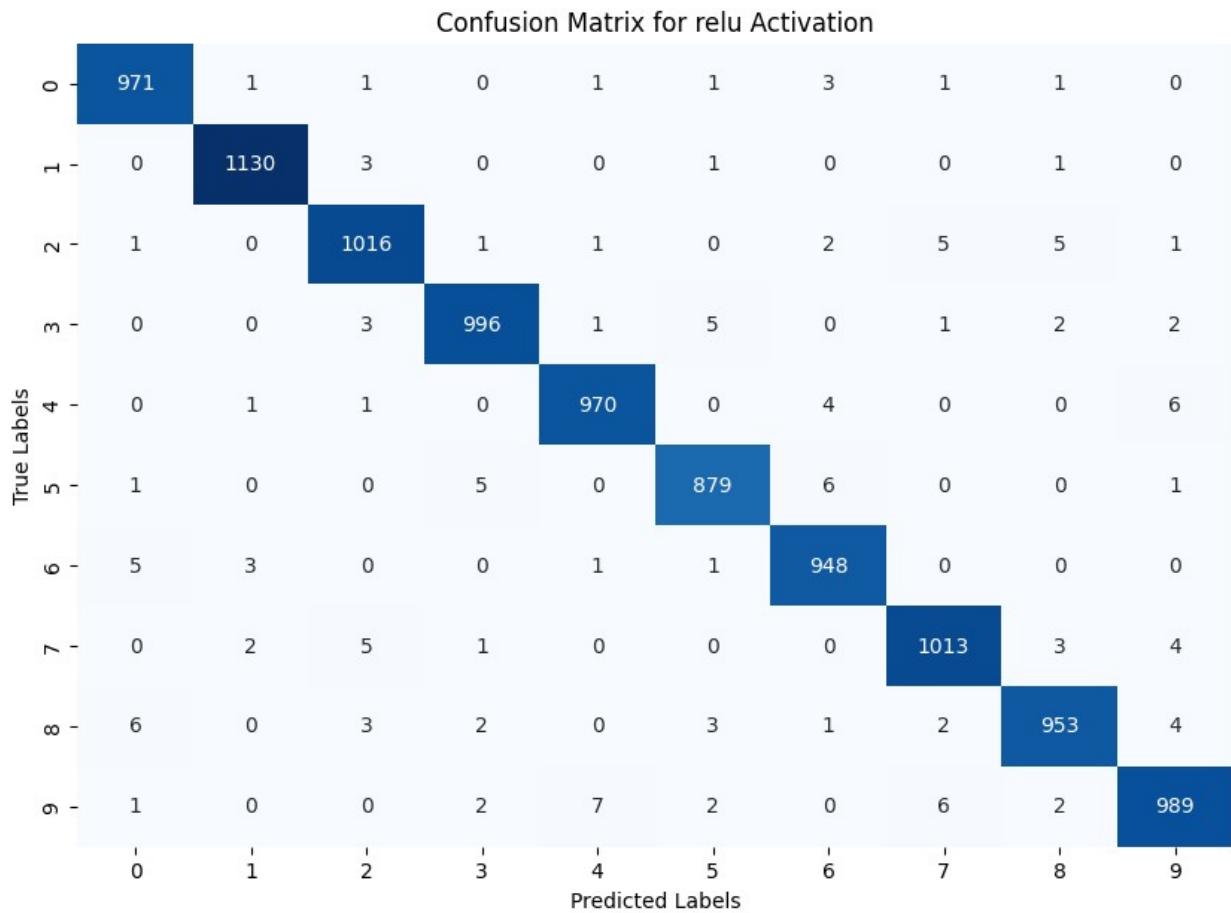
```

844/844 - 4s - loss: 0.0208 - accuracy: 0.9941 - val_loss: 0.0503 -
val_accuracy: 0.9872 - 4s/epoch - 5ms/step
Epoch 8/20
844/844 - 4s - loss: 0.0161 - accuracy: 0.9957 - val_loss: 0.0456 -
val_accuracy: 0.9885 - 4s/epoch - 5ms/step
Epoch 9/20
844/844 - 4s - loss: 0.0135 - accuracy: 0.9962 - val_loss: 0.0440 -
val_accuracy: 0.9892 - 4s/epoch - 5ms/step
Epoch 10/20
844/844 - 4s - loss: 0.0107 - accuracy: 0.9973 - val_loss: 0.0474 -
val_accuracy: 0.9890 - 4s/epoch - 5ms/step
Epoch 11/20
844/844 - 4s - loss: 0.0077 - accuracy: 0.9985 - val_loss: 0.0478 -
val_accuracy: 0.9893 - 4s/epoch - 5ms/step
Epoch 12/20
844/844 - 4s - loss: 0.0064 - accuracy: 0.9988 - val_loss: 0.0511 -
val_accuracy: 0.9885 - 4s/epoch - 5ms/step
Epoch 13/20
844/844 - 4s - loss: 0.0051 - accuracy: 0.9991 - val_loss: 0.0454 -
val_accuracy: 0.9898 - 4s/epoch - 5ms/step
Epoch 14/20
844/844 - 4s - loss: 0.0044 - accuracy: 0.9994 - val_loss: 0.0474 -
val_accuracy: 0.9903 - 4s/epoch - 5ms/step
Epoch 15/20
844/844 - 4s - loss: 0.0032 - accuracy: 0.9998 - val_loss: 0.0474 -
val_accuracy: 0.9893 - 4s/epoch - 5ms/step
Epoch 16/20
844/844 - 4s - loss: 0.0025 - accuracy: 0.9999 - val_loss: 0.0467 -
val_accuracy: 0.9908 - 4s/epoch - 5ms/step
Epoch 17/20
844/844 - 4s - loss: 0.0022 - accuracy: 0.9999 - val_loss: 0.0493 -
val_accuracy: 0.9900 - 4s/epoch - 5ms/step
Epoch 18/20
844/844 - 4s - loss: 0.0017 - accuracy: 0.9999 - val_loss: 0.0499 -
val_accuracy: 0.9898 - 4s/epoch - 5ms/step
Epoch 19/20
844/844 - 4s - loss: 0.0014 - accuracy: 1.0000 - val_loss: 0.0504 -
val_accuracy: 0.9897 - 4s/epoch - 5ms/step
Epoch 20/20
844/844 - 4s - loss: 0.0013 - accuracy: 1.0000 - val_loss: 0.0511 -
val_accuracy: 0.9905 - 4s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 971   1   1   0   1   1   3   1   1   0]
 [  0 1130   3   0   0   1   0   0   1   0]
 [  1   0 1016   1   1   0   2   5   5   1]
 [  0   0   3  996   1   5   0   1   2   2]
 [  0   1   1   0  970   0   4   0   0   6]]
```

```
[ 1 0 0 5 0 879 6 0 0 1]
[ 5 3 0 0 1 1 948 0 0 0]
[ 0 2 5 1 0 0 0 1013 3 4]
[ 6 0 3 2 0 3 1 2 953 4]
[ 1 0 0 2 7 2 0 6 2 989]]
```

Precision: 0.9865

Recall: 0.9865



Training Model with relu activation, 1 conv_layers, 1 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 3s - loss: 0.2815 - accuracy: 0.9157 - val_loss: 0.1124 - val_accuracy: 0.9717 - 3s/epoch - 8ms/step

Epoch 2/5

422/422 - 3s - loss: 0.1193 - accuracy: 0.9649 - val_loss: 0.0878 - val_accuracy: 0.9763 - 3s/epoch - 7ms/step

Epoch 3/5

422/422 - 3s - loss: 0.0807 - accuracy: 0.9759 - val_loss: 0.0738 - val_accuracy: 0.9787 - 3s/epoch - 7ms/step

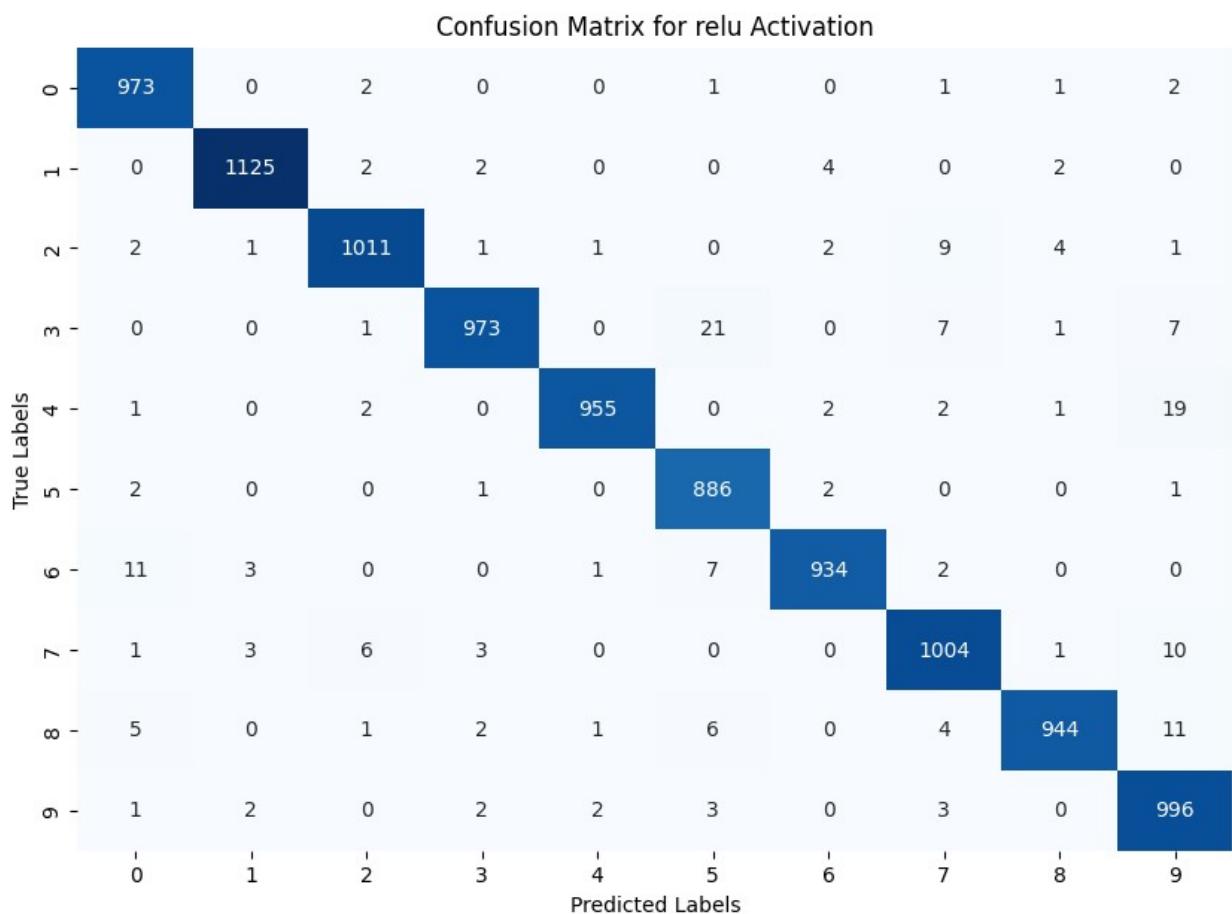
Epoch 4/5

422/422 - 3s - loss: 0.0640 - accuracy: 0.9808 - val_loss: 0.0654 -

```

val_accuracy: 0.9822 - 3s/epoch - 7ms/step
Epoch 5/5
422/422 - 3s - loss: 0.0515 - accuracy: 0.9846 - val_loss: 0.0615 -
val_accuracy: 0.9827 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 973   0   2   0   0   1   0   1   1   2]
 [ 0 1125   2   2   0   0   4   0   2   0]
 [ 2   1 1011   1   1   0   2   9   4   1]
 [ 0   0   1 973   0  21   0   7   1   7]
 [ 1   0   2   0 955   0   2   2   1 19]
 [ 2   0   0   1   0 886   2   0   0   1]
 [ 11   3   0   0   1   7 934   2   0   0]
 [ 1   3   6   3   0   0   0 1004   1 10]
 [ 5   0   1   2   1   6   0   4 944   11]
 [ 1   2   0   2   2   3   0   3   0 996]]
Precision: 0.9804
Recall: 0.9801

```

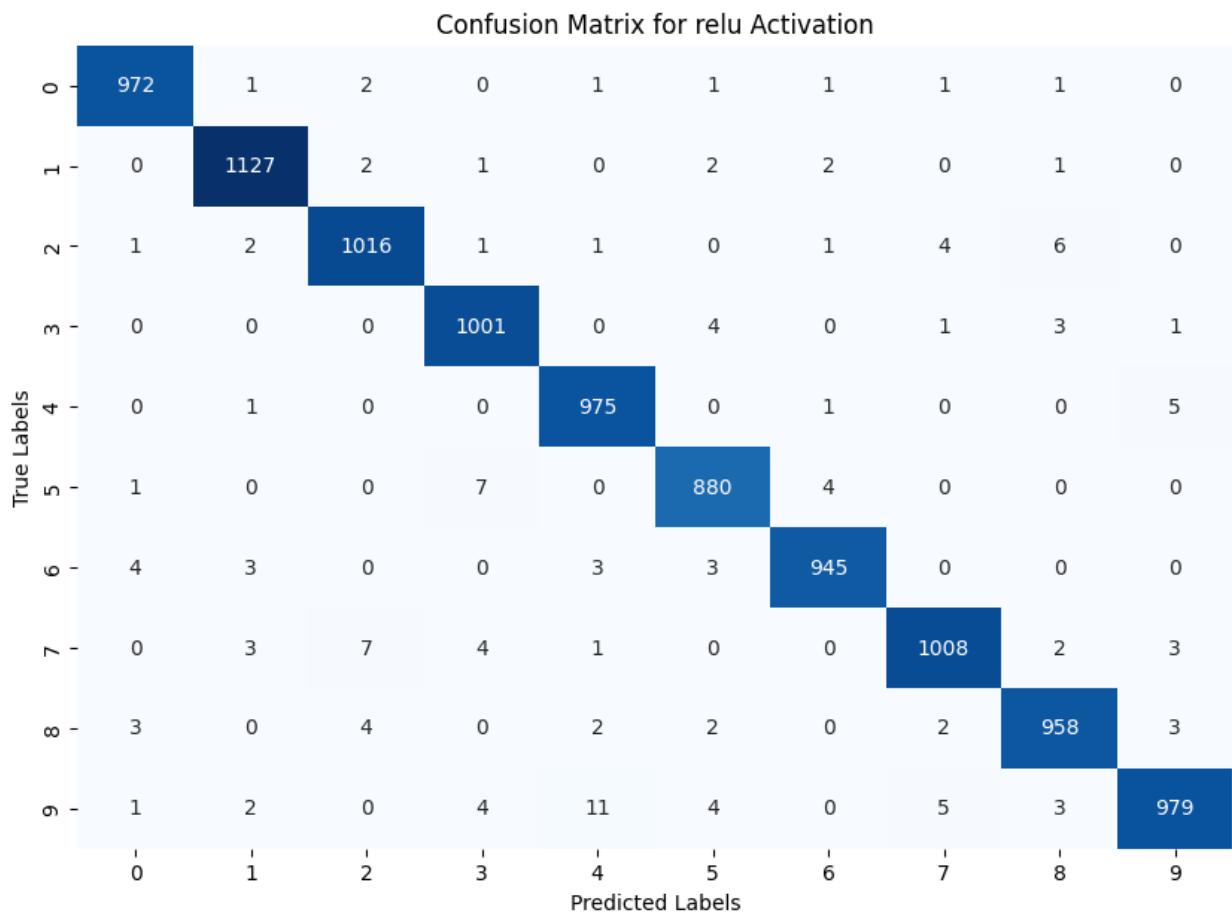


```
Training Model with relu activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 3s - loss: 0.2752 - accuracy: 0.9166 - val_loss: 0.1030 -
val_accuracy: 0.9717 - 3s/epoch - 8ms/step
Epoch 2/15
422/422 - 3s - loss: 0.1116 - accuracy: 0.9677 - val_loss: 0.0797 -
val_accuracy: 0.9770 - 3s/epoch - 7ms/step
Epoch 3/15
422/422 - 3s - loss: 0.0762 - accuracy: 0.9778 - val_loss: 0.0618 -
val_accuracy: 0.9833 - 3s/epoch - 7ms/step
Epoch 4/15
422/422 - 3s - loss: 0.0591 - accuracy: 0.9829 - val_loss: 0.0567 -
val_accuracy: 0.9842 - 3s/epoch - 7ms/step
Epoch 5/15
422/422 - 3s - loss: 0.0470 - accuracy: 0.9864 - val_loss: 0.0495 -
val_accuracy: 0.9863 - 3s/epoch - 7ms/step
Epoch 6/15
422/422 - 3s - loss: 0.0402 - accuracy: 0.9881 - val_loss: 0.0485 -
val_accuracy: 0.9855 - 3s/epoch - 7ms/step
Epoch 7/15
422/422 - 3s - loss: 0.0341 - accuracy: 0.9901 - val_loss: 0.0503 -
val_accuracy: 0.9875 - 3s/epoch - 7ms/step
Epoch 8/15
422/422 - 3s - loss: 0.0305 - accuracy: 0.9912 - val_loss: 0.0495 -
val_accuracy: 0.9873 - 3s/epoch - 7ms/step
Epoch 9/15
422/422 - 3s - loss: 0.0261 - accuracy: 0.9926 - val_loss: 0.0479 -
val_accuracy: 0.9875 - 3s/epoch - 7ms/step
Epoch 10/15
422/422 - 3s - loss: 0.0223 - accuracy: 0.9940 - val_loss: 0.0437 -
val_accuracy: 0.9890 - 3s/epoch - 7ms/step
Epoch 11/15
422/422 - 3s - loss: 0.0192 - accuracy: 0.9948 - val_loss: 0.0454 -
val_accuracy: 0.9870 - 3s/epoch - 7ms/step
Epoch 12/15
422/422 - 3s - loss: 0.0172 - accuracy: 0.9956 - val_loss: 0.0425 -
val_accuracy: 0.9892 - 3s/epoch - 7ms/step
Epoch 13/15
422/422 - 3s - loss: 0.0149 - accuracy: 0.9961 - val_loss: 0.0464 -
val_accuracy: 0.9872 - 3s/epoch - 7ms/step
Epoch 14/15
422/422 - 3s - loss: 0.0124 - accuracy: 0.9973 - val_loss: 0.0464 -
val_accuracy: 0.9873 - 3s/epoch - 7ms/step
Epoch 15/15
422/422 - 3s - loss: 0.0108 - accuracy: 0.9976 - val_loss: 0.0466 -
val_accuracy: 0.9887 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
```

```
[[ 972 1 2 0 1 1 1 1 1 0]
 [ 0 1127 2 1 0 2 2 0 1 0]
 [ 1 2 1016 1 1 0 1 4 6 0]
 [ 0 0 0 1001 0 4 0 1 3 1]
 [ 0 1 0 0 975 0 1 0 0 5]
 [ 1 0 0 7 0 880 4 0 0 0]
 [ 4 3 0 0 3 3 945 0 0 0]
 [ 0 3 7 4 1 0 0 1008 2 3]
 [ 3 0 4 0 2 2 0 2 958 3]
 [ 1 2 0 4 11 4 0 5 3 979]]
```

Precision: 0.9861

Recall: 0.9861



Training Model with relu activation, 1 conv_layers, 1 dense layers, 128 batch size, 20 epochs..

Epoch 1/20

422/422 - 3s - loss: 0.2979 - accuracy: 0.9086 - val_loss: 0.1311 - val_accuracy: 0.9625 - 3s/epoch - 8ms/step

Epoch 2/20

422/422 - 3s - loss: 0.1360 - accuracy: 0.9583 - val_loss: 0.0866 - val_accuracy: 0.9765 - 3s/epoch - 7ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.0933 - accuracy: 0.9720 - val_loss: 0.0696 -
val_accuracy: 0.9805 - 3s/epoch - 7ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0701 - accuracy: 0.9786 - val_loss: 0.0629 -
val_accuracy: 0.9818 - 3s/epoch - 7ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0561 - accuracy: 0.9833 - val_loss: 0.0628 -
val_accuracy: 0.9827 - 3s/epoch - 7ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0471 - accuracy: 0.9863 - val_loss: 0.0616 -
val_accuracy: 0.9822 - 3s/epoch - 7ms/step
Epoch 7/20
422/422 - 3s - loss: 0.0390 - accuracy: 0.9887 - val_loss: 0.0521 -
val_accuracy: 0.9853 - 3s/epoch - 7ms/step
Epoch 8/20
422/422 - 3s - loss: 0.0331 - accuracy: 0.9905 - val_loss: 0.0526 -
val_accuracy: 0.9852 - 3s/epoch - 7ms/step
Epoch 9/20
422/422 - 3s - loss: 0.0294 - accuracy: 0.9915 - val_loss: 0.0493 -
val_accuracy: 0.9853 - 3s/epoch - 7ms/step
Epoch 10/20
422/422 - 3s - loss: 0.0259 - accuracy: 0.9929 - val_loss: 0.0465 -
val_accuracy: 0.9858 - 3s/epoch - 7ms/step
Epoch 11/20
422/422 - 3s - loss: 0.0230 - accuracy: 0.9934 - val_loss: 0.0475 -
val_accuracy: 0.9863 - 3s/epoch - 7ms/step
Epoch 12/20
422/422 - 3s - loss: 0.0194 - accuracy: 0.9948 - val_loss: 0.0501 -
val_accuracy: 0.9862 - 3s/epoch - 7ms/step
Epoch 13/20
422/422 - 3s - loss: 0.0170 - accuracy: 0.9956 - val_loss: 0.0461 -
val_accuracy: 0.9867 - 3s/epoch - 7ms/step
Epoch 14/20
422/422 - 3s - loss: 0.0154 - accuracy: 0.9959 - val_loss: 0.0482 -
val_accuracy: 0.9867 - 3s/epoch - 7ms/step
Epoch 15/20
422/422 - 3s - loss: 0.0134 - accuracy: 0.9969 - val_loss: 0.0497 -
val_accuracy: 0.9867 - 3s/epoch - 7ms/step
Epoch 16/20
422/422 - 3s - loss: 0.0116 - accuracy: 0.9975 - val_loss: 0.0449 -
val_accuracy: 0.9887 - 3s/epoch - 7ms/step
Epoch 17/20
422/422 - 3s - loss: 0.0106 - accuracy: 0.9976 - val_loss: 0.0488 -
val_accuracy: 0.9877 - 3s/epoch - 7ms/step
Epoch 18/20
422/422 - 3s - loss: 0.0090 - accuracy: 0.9983 - val_loss: 0.0537 -
val_accuracy: 0.9862 - 3s/epoch - 7ms/step
Epoch 19/20
```

```
422/422 - 3s - loss: 0.0076 - accuracy: 0.9987 - val_loss: 0.0456 -  
val_accuracy: 0.9882 - 3s/epoch - 7ms/step
```

```
Epoch 20/20
```

```
422/422 - 3s - loss: 0.0067 - accuracy: 0.9990 - val_loss: 0.0480 -  
val_accuracy: 0.9865 - 3s/epoch - 7ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

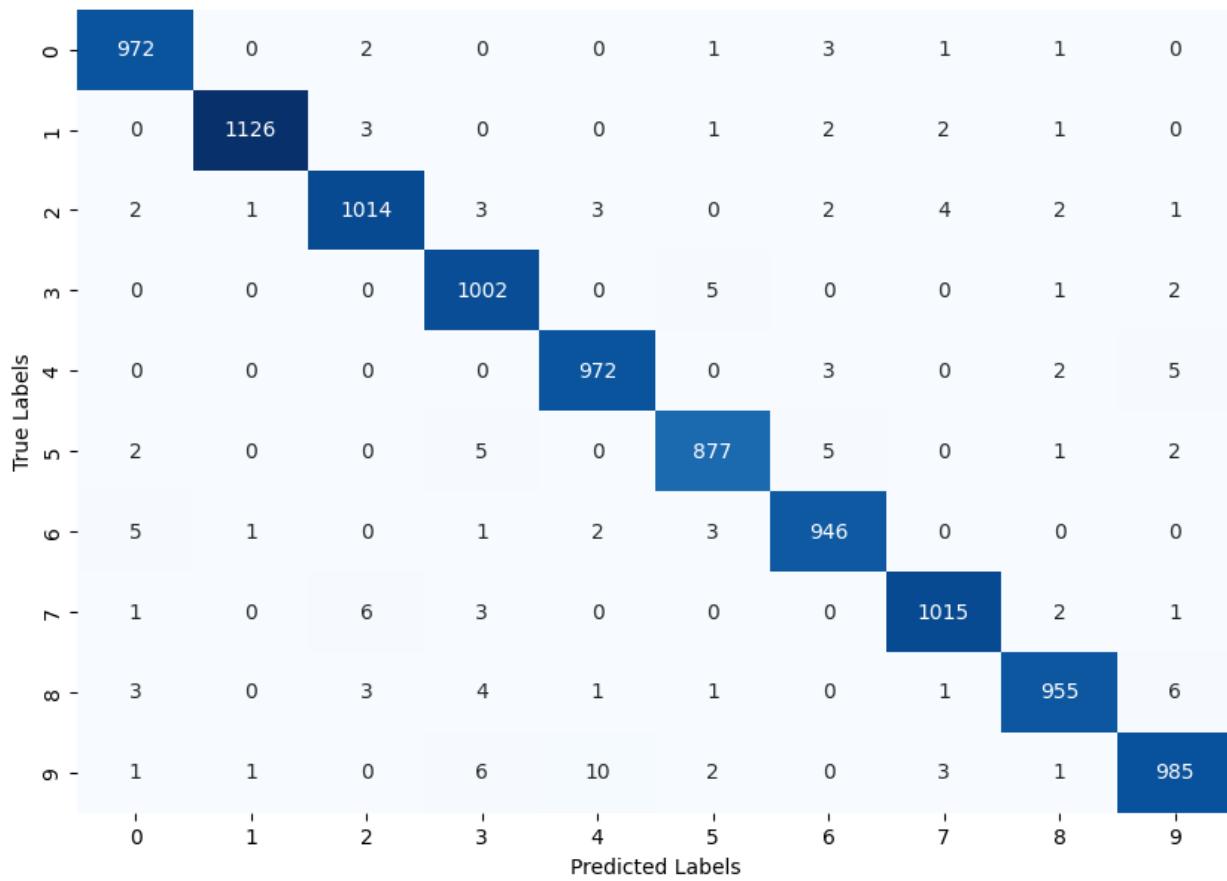
```
Confusion Matrix:
```

```
[[ 972    0    2    0    0    1    3    1    1    0]
 [ 0 1126    3    0    0    1    2    2    1    0]
 [ 2    1 1014    3    3    0    2    4    2    1]
 [ 0    0    0 1002    0    5    0    0    1    2]
 [ 0    0    0    0 972    0    3    0    2    5]
 [ 2    0    0    5    0 877    5    0    1    2]
 [ 5    1    0    1    2    3 946    0    0    0]
 [ 1    0    6    3    0    0    0 1015    2    1]
 [ 3    0    3    4    1    1    0    1    955    6]
 [ 1    1    0    6   10    2    0    3    1    985]]
```

```
Precision: 0.9864
```

```
Recall: 0.9864
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 1 conv_layers, 1 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 3s - loss: 0.3928 - accuracy: 0.8793 - val_loss: 0.1635 -
val_accuracy: 0.9567 - 3s/epoch - 13ms/step
Epoch 2/5
211/211 - 2s - loss: 0.1691 - accuracy: 0.9501 - val_loss: 0.1144 -
val_accuracy: 0.9687 - 2s/epoch - 10ms/step
Epoch 3/5
211/211 - 2s - loss: 0.1225 - accuracy: 0.9642 - val_loss: 0.0957 -
val_accuracy: 0.9745 - 2s/epoch - 10ms/step
Epoch 4/5
211/211 - 2s - loss: 0.0966 - accuracy: 0.9718 - val_loss: 0.0805 -
val_accuracy: 0.9790 - 2s/epoch - 10ms/step
Epoch 5/5
211/211 - 2s - loss: 0.0807 - accuracy: 0.9767 - val_loss: 0.0714 -
val_accuracy: 0.9797 - 2s/epoch - 10ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 972     0     1     1     0     2     0     2     2     0]
 [  0 1124     3     3     0     0     2     0     3     0]
 [  7     1 999     7     4     0     1     4     9     0]
 [  0     0     1 996     0     4     0     3     5     1]
 [  1     0     3     0 970     0     0     2     2     4]
 [  4     0     0    11     0 869     4     1     3     0]
 [ 11     3     2     1     3     8 925     0     5     0]
 [  0     2    16     6     0     1     0 997     2     4]
 [  6     0     2     7     4     1     3     2 949     0]
 [  6     4     1     9    10     2     0     5     4 968]]
```

Precision: 0.9771
Recall: 0.9769

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9	
0	972	0	1	1	0	2	0	2	2	0	
1	0	1124	3	3	0	0	2	0	3	0	
2	7	1	999	7	4	0	1	4	9	0	
3	0	0	1	996	0	4	0	3	5	1	
4	1	0	3	0	970	0	0	2	2	4	
5	4	0	0	11	0	869	4	1	3	0	
6	11	3	2	1	3	8	925	0	5	0	
7	0	2	16	6	0	1	0	997	2	4	
8	6	0	2	7	4	1	3	2	949	0	
9	6	4	1	9	10	2	0	5	4	968	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```
Training Model with relu activation, 1 conv_layers, 1 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 3s - loss: 0.3663 - accuracy: 0.8896 - val_loss: 0.1599 -
val_accuracy: 0.9567 - 3s/epoch - 13ms/step
Epoch 2/15
211/211 - 2s - loss: 0.1601 - accuracy: 0.9533 - val_loss: 0.1045 -
val_accuracy: 0.9707 - 2s/epoch - 11ms/step
Epoch 3/15
211/211 - 2s - loss: 0.1153 - accuracy: 0.9666 - val_loss: 0.0850 -
val_accuracy: 0.9760 - 2s/epoch - 11ms/step
Epoch 4/15
211/211 - 2s - loss: 0.0918 - accuracy: 0.9739 - val_loss: 0.0782 -
val_accuracy: 0.9787 - 2s/epoch - 11ms/step
Epoch 5/15
211/211 - 2s - loss: 0.0748 - accuracy: 0.9787 - val_loss: 0.0707 -
val_accuracy: 0.9810 - 2s/epoch - 11ms/step
Epoch 6/15
211/211 - 2s - loss: 0.0643 - accuracy: 0.9812 - val_loss: 0.0629 -
val_accuracy: 0.9833 - 2s/epoch - 11ms/step
Epoch 7/15
```

```
211/211 - 2s - loss: 0.0548 - accuracy: 0.9844 - val_loss: 0.0615 -  
val_accuracy: 0.9828 - 2s/epoch - 11ms/step  
Epoch 8/15  
211/211 - 2s - loss: 0.0491 - accuracy: 0.9860 - val_loss: 0.0586 -  
val_accuracy: 0.9848 - 2s/epoch - 11ms/step  
Epoch 9/15  
211/211 - 2s - loss: 0.0443 - accuracy: 0.9873 - val_loss: 0.0620 -  
val_accuracy: 0.9830 - 2s/epoch - 11ms/step  
Epoch 10/15  
211/211 - 2s - loss: 0.0398 - accuracy: 0.9887 - val_loss: 0.0553 -  
val_accuracy: 0.9853 - 2s/epoch - 11ms/step  
Epoch 11/15  
211/211 - 2s - loss: 0.0358 - accuracy: 0.9896 - val_loss: 0.0552 -  
val_accuracy: 0.9843 - 2s/epoch - 11ms/step  
Epoch 12/15  
211/211 - 2s - loss: 0.0325 - accuracy: 0.9908 - val_loss: 0.0568 -  
val_accuracy: 0.9857 - 2s/epoch - 11ms/step  
Epoch 13/15  
211/211 - 2s - loss: 0.0301 - accuracy: 0.9920 - val_loss: 0.0497 -  
val_accuracy: 0.9870 - 2s/epoch - 11ms/step  
Epoch 14/15  
211/211 - 2s - loss: 0.0277 - accuracy: 0.9924 - val_loss: 0.0528 -  
val_accuracy: 0.9862 - 2s/epoch - 11ms/step  
Epoch 15/15  
211/211 - 2s - loss: 0.0257 - accuracy: 0.9930 - val_loss: 0.0536 -  
val_accuracy: 0.9847 - 2s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 966  0  1  1  2  1  4  3  1  1]  
[ 0 1126  2  3  0  0  2  0  2  0]  
[ 1  0 1013  5  2  0  1  3  7  0]  
[ 0  0  1 1002  0  2  0  2  1  2]  
[ 0  0  2  0  973  0  0  0  0  7]  
[ 1  0  0  7  0  876  5  1  2  0]  
[ 7  2  0  0  2  3  942  0  2  0]  
[ 0  2  6  4  0  0  0 1005  3  8]  
[ 3  0  1  5  1  1  0  1  959  3]  
[ 2  1  1  4  9  1  0  5  1  985]]  
Precision: 0.9847  
Recall: 0.9847
```

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
0	966	0	1	1	2	1	4	3	1	1
1	0	1126	2	3	0	0	2	0	2	0
2	1	0	1013	5	2	0	1	3	7	0
3	0	0	1	1002	0	2	0	2	1	2
4	0	0	2	0	973	0	0	0	0	7
5	1	0	0	7	0	876	5	1	2	0
6	7	2	0	0	2	3	942	0	2	0
7	0	2	6	4	0	0	0	1005	3	8
8	3	0	1	5	1	1	0	1	959	3
9	2	1	1	4	9	1	0	5	1	985
	0	1	2	3	4	5	6	7	8	9
	True Labels									Predicted Labels

```
Training Model with relu activation, 1 conv_layers, 1 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 3s - loss: 0.3771 - accuracy: 0.8854 - val_loss: 0.1770 -
val_accuracy: 0.9508 - 3s/epoch - 13ms/step
Epoch 2/20
211/211 - 2s - loss: 0.1840 - accuracy: 0.9464 - val_loss: 0.1244 -
val_accuracy: 0.9678 - 2s/epoch - 10ms/step
Epoch 3/20
211/211 - 2s - loss: 0.1327 - accuracy: 0.9621 - val_loss: 0.0971 -
val_accuracy: 0.9743 - 2s/epoch - 10ms/step
Epoch 4/20
211/211 - 2s - loss: 0.1054 - accuracy: 0.9697 - val_loss: 0.0867 -
val_accuracy: 0.9763 - 2s/epoch - 11ms/step
Epoch 5/20
211/211 - 2s - loss: 0.0851 - accuracy: 0.9756 - val_loss: 0.0826 -
val_accuracy: 0.9763 - 2s/epoch - 10ms/step
Epoch 6/20
211/211 - 2s - loss: 0.0734 - accuracy: 0.9787 - val_loss: 0.0675 -
val_accuracy: 0.9818 - 2s/epoch - 11ms/step
Epoch 7/20
```

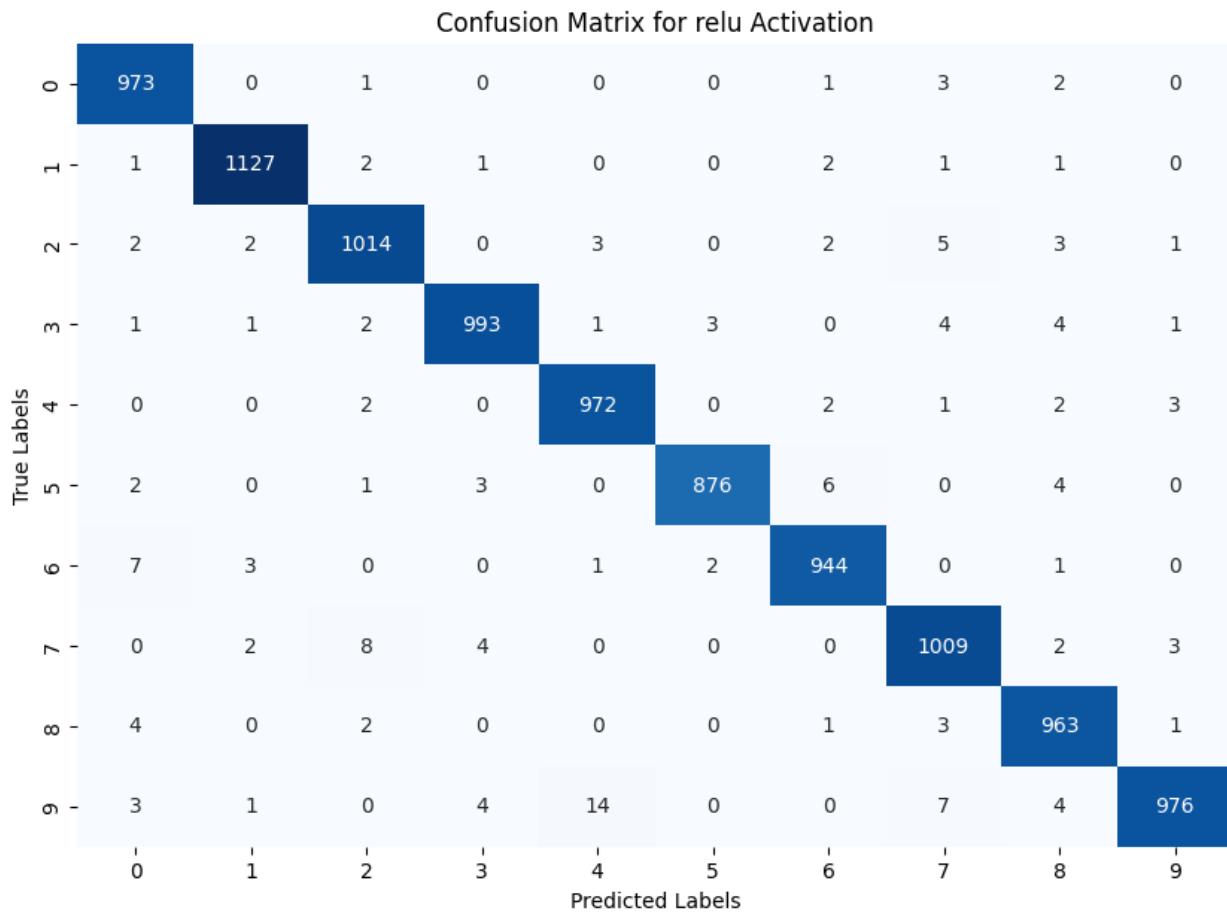
```

211/211 - 2s - loss: 0.0628 - accuracy: 0.9822 - val_loss: 0.0624 -
val_accuracy: 0.9838 - 2s/epoch - 10ms/step
Epoch 8/20
211/211 - 2s - loss: 0.0569 - accuracy: 0.9837 - val_loss: 0.0607 -
val_accuracy: 0.9835 - 2s/epoch - 11ms/step
Epoch 9/20
211/211 - 2s - loss: 0.0504 - accuracy: 0.9854 - val_loss: 0.0577 -
val_accuracy: 0.9835 - 2s/epoch - 10ms/step
Epoch 10/20
211/211 - 2s - loss: 0.0451 - accuracy: 0.9874 - val_loss: 0.0573 -
val_accuracy: 0.9852 - 2s/epoch - 10ms/step
Epoch 11/20
211/211 - 2s - loss: 0.0411 - accuracy: 0.9885 - val_loss: 0.0568 -
val_accuracy: 0.9847 - 2s/epoch - 11ms/step
Epoch 12/20
211/211 - 2s - loss: 0.0381 - accuracy: 0.9893 - val_loss: 0.0542 -
val_accuracy: 0.9870 - 2s/epoch - 11ms/step
Epoch 13/20
211/211 - 2s - loss: 0.0355 - accuracy: 0.9902 - val_loss: 0.0534 -
val_accuracy: 0.9858 - 2s/epoch - 10ms/step
Epoch 14/20
211/211 - 2s - loss: 0.0318 - accuracy: 0.9907 - val_loss: 0.0534 -
val_accuracy: 0.9852 - 2s/epoch - 10ms/step
Epoch 15/20
211/211 - 2s - loss: 0.0300 - accuracy: 0.9916 - val_loss: 0.0521 -
val_accuracy: 0.9852 - 2s/epoch - 11ms/step
Epoch 16/20
211/211 - 2s - loss: 0.0268 - accuracy: 0.9930 - val_loss: 0.0549 -
val_accuracy: 0.9852 - 2s/epoch - 11ms/step
Epoch 17/20
211/211 - 2s - loss: 0.0260 - accuracy: 0.9928 - val_loss: 0.0511 -
val_accuracy: 0.9858 - 2s/epoch - 11ms/step
Epoch 18/20
211/211 - 2s - loss: 0.0236 - accuracy: 0.9934 - val_loss: 0.0472 -
val_accuracy: 0.9872 - 2s/epoch - 11ms/step
Epoch 19/20
211/211 - 2s - loss: 0.0217 - accuracy: 0.9944 - val_loss: 0.0495 -
val_accuracy: 0.9875 - 2s/epoch - 10ms/step
Epoch 20/20
211/211 - 2s - loss: 0.0207 - accuracy: 0.9947 - val_loss: 0.0499 -
val_accuracy: 0.9870 - 2s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 973   0   1   0   0   0   1   3   2   0]
 [  1 1127   2   1   0   0   2   1   1   0]
 [  2   2 1014   0   3   0   2   5   3   1]
 [  1   1   2  993   1   3   0   4   4   1]
 [  0   0   2   0  972   0   2   1   2   3]]
```

```
[ 2 0 1 3 0 876 6 0 4 0]
[ 7 3 0 0 1 2 944 0 1 0]
[ 0 2 8 4 0 0 0 1009 2 3]
[ 4 0 2 0 0 0 1 3 963 1]
[ 3 1 0 4 14 0 0 7 4 976]]
```

Precision: 0.9847

Recall: 0.9847



Training Model with relu activation, 1 conv_layers, 1 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 5s - loss: 0.2086 - accuracy: 0.9364 - val_loss: 0.0845 - val_accuracy: 0.9768 - 5s/epoch - 6ms/step

Epoch 2/5

844/844 - 4s - loss: 0.0783 - accuracy: 0.9766 - val_loss: 0.0729 - val_accuracy: 0.9807 - 4s/epoch - 5ms/step

Epoch 3/5

844/844 - 5s - loss: 0.0521 - accuracy: 0.9839 - val_loss: 0.0657 - val_accuracy: 0.9813 - 5s/epoch - 6ms/step

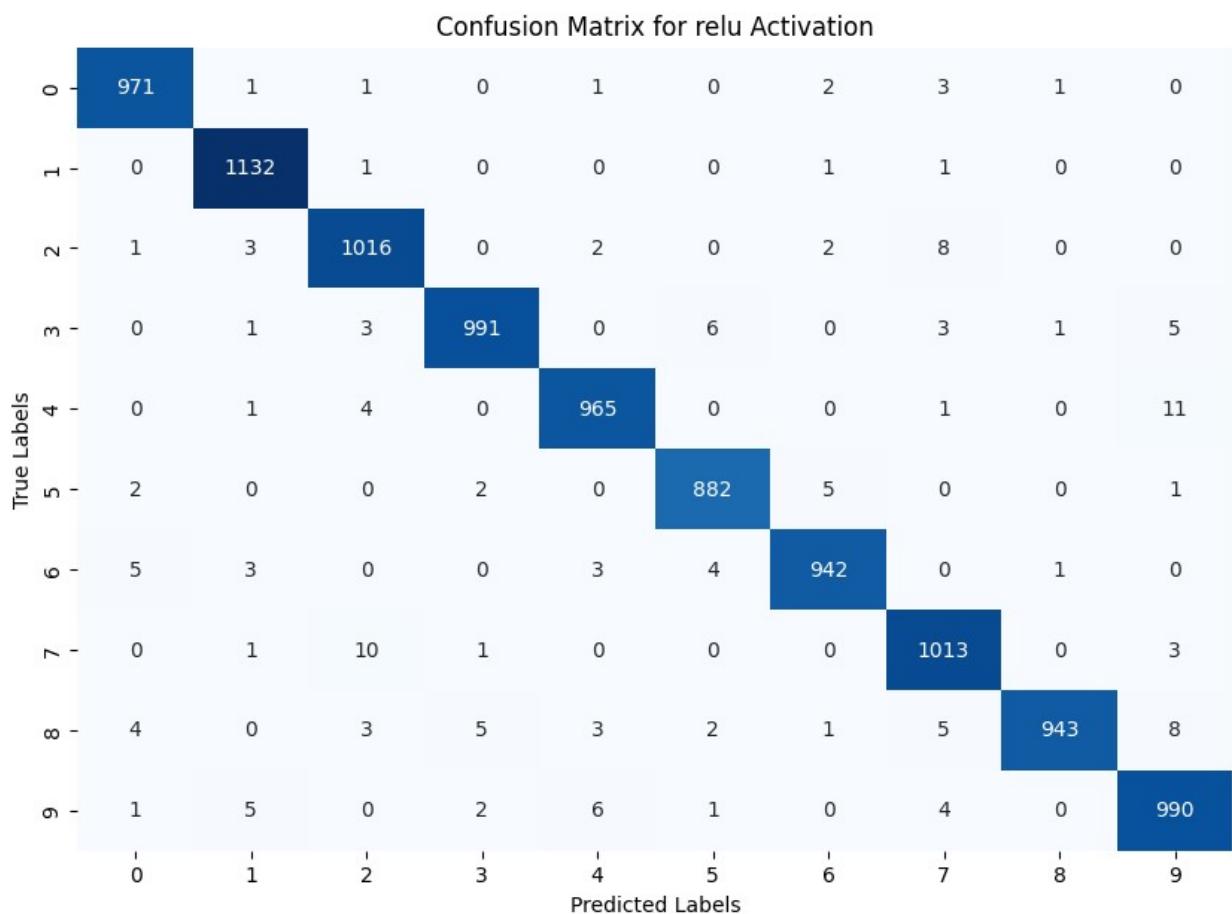
Epoch 4/5

844/844 - 5s - loss: 0.0381 - accuracy: 0.9885 - val_loss: 0.0556 -

```

val_accuracy: 0.9843 - 5s/epoch - 5ms/step
Epoch 5/5
844/844 - 5s - loss: 0.0288 - accuracy: 0.9909 - val_loss: 0.0506 -
val_accuracy: 0.9860 - 5s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 971   1   1   0   1   0   2   3   1   0]
 [  0 1132   1   0   0   0   1   1   0   0]
 [  1   3 1016   0   2   0   2   8   0   0]
 [  0   1   3 991   0   6   0   3   1   5]
 [  0   1   4   0 965   0   0   1   0 11]
 [  2   0   0   2   0 882   5   0   0   1]
 [  5   3   0   0   3   4 942   0   1   0]
 [  0   1  10   1   0   0   0 1013   0   3]
 [  4   0   3   5   3   2   1   5 943   8]
 [  1   5   0   2   6   1   0   4   0 990]]
Precision: 0.9846
Recall: 0.9845

```

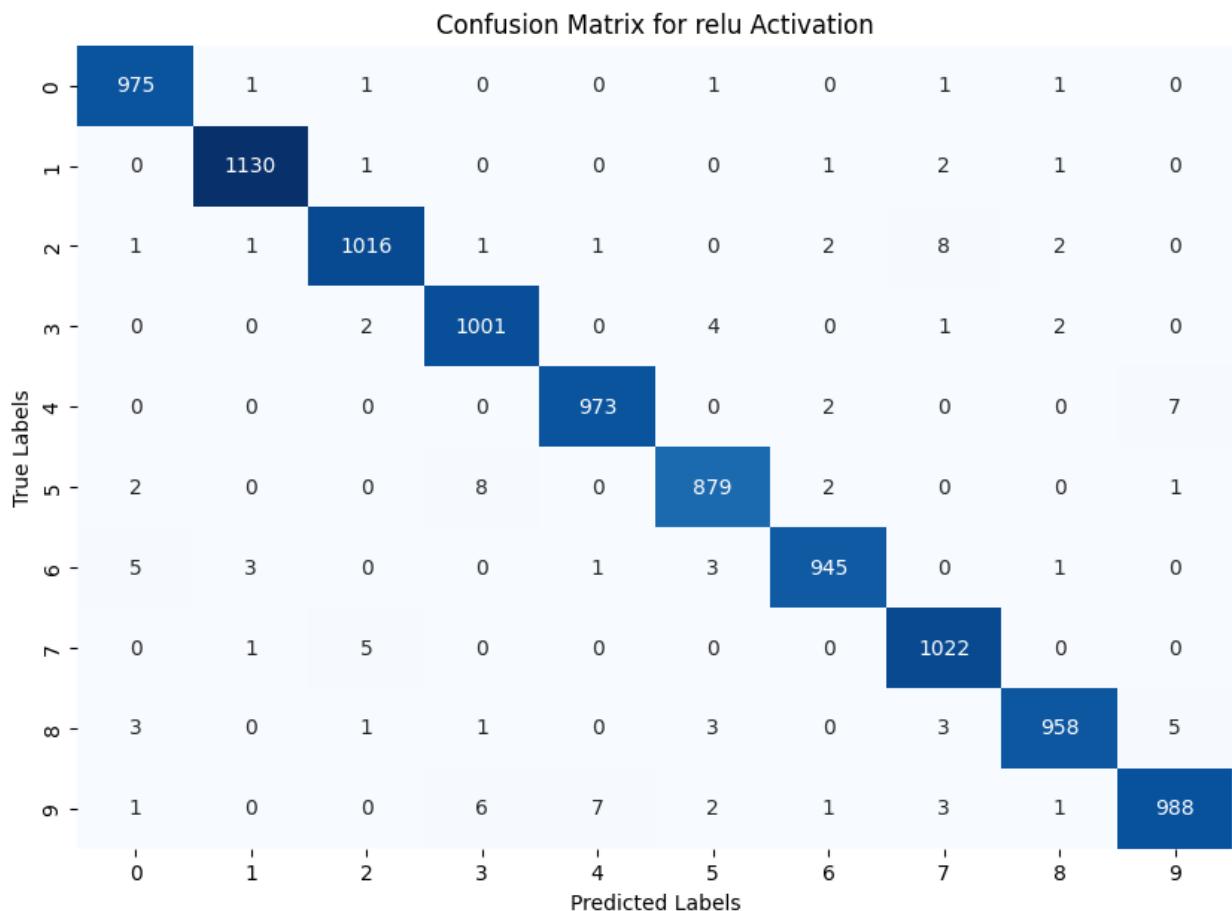


```
Training Model with relu activation, 1 conv_layers, 1 dense layers, 64
batch size, 15 epochs..
Epoch 1/15
844/844 - 5s - loss: 0.2112 - accuracy: 0.9352 - val_loss: 0.1016 -
val_accuracy: 0.9738 - 5s/epoch - 6ms/step
Epoch 2/15
844/844 - 5s - loss: 0.0782 - accuracy: 0.9768 - val_loss: 0.0602 -
val_accuracy: 0.9833 - 5s/epoch - 5ms/step
Epoch 3/15
844/844 - 4s - loss: 0.0505 - accuracy: 0.9851 - val_loss: 0.0556 -
val_accuracy: 0.9843 - 4s/epoch - 5ms/step
Epoch 4/15
844/844 - 5s - loss: 0.0372 - accuracy: 0.9892 - val_loss: 0.0463 -
val_accuracy: 0.9883 - 5s/epoch - 5ms/step
Epoch 5/15
844/844 - 5s - loss: 0.0288 - accuracy: 0.9915 - val_loss: 0.0470 -
val_accuracy: 0.9867 - 5s/epoch - 5ms/step
Epoch 6/15
844/844 - 4s - loss: 0.0222 - accuracy: 0.9938 - val_loss: 0.0462 -
val_accuracy: 0.9880 - 4s/epoch - 5ms/step
Epoch 7/15
844/844 - 4s - loss: 0.0173 - accuracy: 0.9951 - val_loss: 0.0455 -
val_accuracy: 0.9875 - 4s/epoch - 5ms/step
Epoch 8/15
844/844 - 4s - loss: 0.0127 - accuracy: 0.9968 - val_loss: 0.0438 -
val_accuracy: 0.9890 - 4s/epoch - 5ms/step
Epoch 9/15
844/844 - 5s - loss: 0.0103 - accuracy: 0.9978 - val_loss: 0.0472 -
val_accuracy: 0.9887 - 5s/epoch - 5ms/step
Epoch 10/15
844/844 - 5s - loss: 0.0074 - accuracy: 0.9987 - val_loss: 0.0451 -
val_accuracy: 0.9880 - 5s/epoch - 5ms/step
Epoch 11/15
844/844 - 4s - loss: 0.0055 - accuracy: 0.9992 - val_loss: 0.0469 -
val_accuracy: 0.9888 - 4s/epoch - 5ms/step
Epoch 12/15
844/844 - 5s - loss: 0.0048 - accuracy: 0.9991 - val_loss: 0.0471 -
val_accuracy: 0.9882 - 5s/epoch - 5ms/step
Epoch 13/15
844/844 - 5s - loss: 0.0036 - accuracy: 0.9996 - val_loss: 0.0454 -
val_accuracy: 0.9893 - 5s/epoch - 5ms/step
Epoch 14/15
844/844 - 5s - loss: 0.0028 - accuracy: 0.9998 - val_loss: 0.0464 -
val_accuracy: 0.9898 - 5s/epoch - 5ms/step
Epoch 15/15
844/844 - 5s - loss: 0.0023 - accuracy: 0.9999 - val_loss: 0.0462 -
val_accuracy: 0.9898 - 5s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
```

```
[[ 975  1  1  0  0  1  0  1  1  0]
 [ 0 1130  1  0  0  0  1  2  1  0]
 [ 1  1 1016  1  1  0  2  8  2  0]
 [ 0  0  2 1001  0  4  0  1  2  0]
 [ 0  0  0  0  973  0  2  0  0  7]
 [ 2  0  0  8  0  879  2  0  0  1]
 [ 5  3  0  0  1  3  945  0  1  0]
 [ 0  1  5  0  0  0  0 1022  0  0]
 [ 3  0  1  1  0  3  0  3  958  5]
 [ 1  0  0  6  7  2  1  3  1  988]]
```

Precision: 0.9887

Recall: 0.9887



Training Model with relu activation, 1 conv_layers, 1 dense layers, 64 batch size, 20 epochs..

Epoch 1/20

844/844 - 5s - loss: 0.2091 - accuracy: 0.9371 - val_loss: 0.0819 - val_accuracy: 0.9787 - 5s/epoch - 6ms/step

Epoch 2/20

844/844 - 4s - loss: 0.0768 - accuracy: 0.9780 - val_loss: 0.0750 - val_accuracy: 0.9797 - 4s/epoch - 5ms/step

```
Epoch 3/20
844/844 - 4s - loss: 0.0518 - accuracy: 0.9843 - val_loss: 0.0545 -
val_accuracy: 0.9853 - 4s/epoch - 5ms/step
Epoch 4/20
844/844 - 4s - loss: 0.0371 - accuracy: 0.9885 - val_loss: 0.0530 -
val_accuracy: 0.9860 - 4s/epoch - 5ms/step
Epoch 5/20
844/844 - 5s - loss: 0.0285 - accuracy: 0.9918 - val_loss: 0.0547 -
val_accuracy: 0.9852 - 5s/epoch - 5ms/step
Epoch 6/20
844/844 - 5s - loss: 0.0228 - accuracy: 0.9934 - val_loss: 0.0441 -
val_accuracy: 0.9885 - 5s/epoch - 6ms/step
Epoch 7/20
844/844 - 5s - loss: 0.0174 - accuracy: 0.9952 - val_loss: 0.0512 -
val_accuracy: 0.9868 - 5s/epoch - 5ms/step
Epoch 8/20
844/844 - 5s - loss: 0.0129 - accuracy: 0.9966 - val_loss: 0.0479 -
val_accuracy: 0.9885 - 5s/epoch - 5ms/step
Epoch 9/20
844/844 - 4s - loss: 0.0108 - accuracy: 0.9971 - val_loss: 0.0495 -
val_accuracy: 0.9895 - 4s/epoch - 5ms/step
Epoch 10/20
844/844 - 4s - loss: 0.0079 - accuracy: 0.9983 - val_loss: 0.0487 -
val_accuracy: 0.9885 - 4s/epoch - 5ms/step
Epoch 11/20
844/844 - 5s - loss: 0.0058 - accuracy: 0.9991 - val_loss: 0.0510 -
val_accuracy: 0.9882 - 5s/epoch - 5ms/step
Epoch 12/20
844/844 - 5s - loss: 0.0048 - accuracy: 0.9990 - val_loss: 0.0490 -
val_accuracy: 0.9892 - 5s/epoch - 5ms/step
Epoch 13/20
844/844 - 5s - loss: 0.0036 - accuracy: 0.9996 - val_loss: 0.0511 -
val_accuracy: 0.9883 - 5s/epoch - 5ms/step
Epoch 14/20
844/844 - 5s - loss: 0.0029 - accuracy: 0.9997 - val_loss: 0.0530 -
val_accuracy: 0.9887 - 5s/epoch - 5ms/step
Epoch 15/20
844/844 - 5s - loss: 0.0026 - accuracy: 0.9997 - val_loss: 0.0523 -
val_accuracy: 0.9885 - 5s/epoch - 5ms/step
Epoch 16/20
844/844 - 5s - loss: 0.0020 - accuracy: 0.9998 - val_loss: 0.0530 -
val_accuracy: 0.9885 - 5s/epoch - 5ms/step
Epoch 17/20
844/844 - 5s - loss: 0.0017 - accuracy: 0.9999 - val_loss: 0.0536 -
val_accuracy: 0.9888 - 5s/epoch - 5ms/step
Epoch 18/20
844/844 - 5s - loss: 0.0015 - accuracy: 0.9999 - val_loss: 0.0534 -
val_accuracy: 0.9890 - 5s/epoch - 5ms/step
Epoch 19/20
```

```
844/844 - 5s - loss: 0.0012 - accuracy: 0.9999 - val_loss: 0.0541 -  
val_accuracy: 0.9885 - 5s/epoch - 5ms/step
```

```
Epoch 20/20
```

```
844/844 - 5s - loss: 0.0011 - accuracy: 0.9999 - val_loss: 0.0557 -  
val_accuracy: 0.9890 - 5s/epoch - 5ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

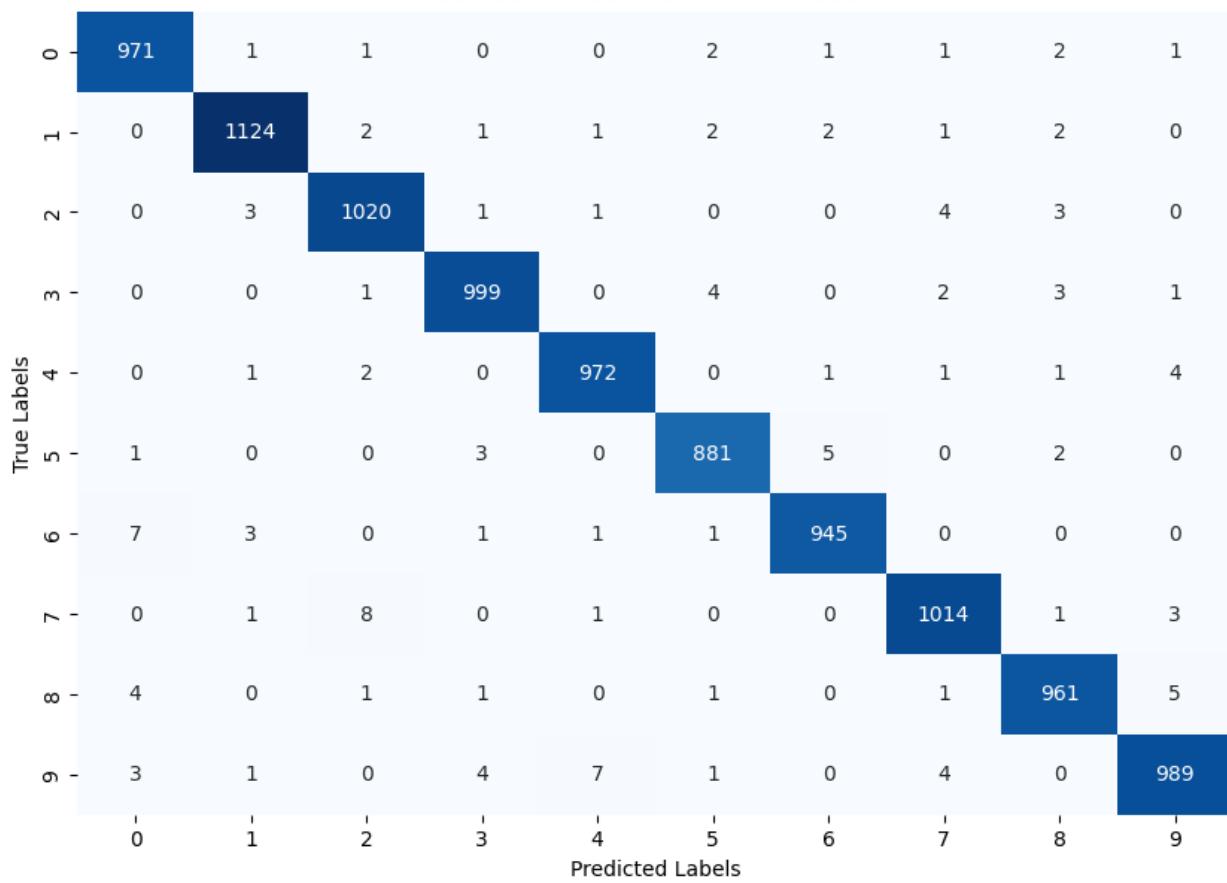
```
Confusion Matrix:
```

```
[[ 971   1   1   0   0   2   1   1   2   1]
 [ 0 1124   2   1   1   2   2   1   2   0]
 [ 0   3 1020   1   1   0   0   4   3   0]
 [ 0   0   1 999   0   4   0   2   3   1]
 [ 0   1   2   0 972   0   1   1   1   4]
 [ 1   0   0   3   0 881   5   0   2   0]
 [ 7   3   0   1   1   1 945   0   0   0]
 [ 0   1   8   0   1   0   0 1014   1   3]
 [ 4   0   1   1   0   1   0   1 961   5]
 [ 3   1   0   4   7   1   0   4   0 989]]
```

```
Precision: 0.9876
```

```
Recall: 0.9876
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 1 conv_layers, 1 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 4s - loss: 0.2814 - accuracy: 0.9170 - val_loss: 0.1265 -
val_accuracy: 0.9655 - 4s/epoch - 8ms/step
Epoch 2/5
422/422 - 3s - loss: 0.1192 - accuracy: 0.9647 - val_loss: 0.0788 -
val_accuracy: 0.9783 - 3s/epoch - 7ms/step
Epoch 3/5
422/422 - 3s - loss: 0.0802 - accuracy: 0.9768 - val_loss: 0.0698 -
val_accuracy: 0.9800 - 3s/epoch - 7ms/step
Epoch 4/5
422/422 - 3s - loss: 0.0612 - accuracy: 0.9818 - val_loss: 0.0549 -
val_accuracy: 0.9862 - 3s/epoch - 7ms/step
Epoch 5/5
422/422 - 3s - loss: 0.0487 - accuracy: 0.9856 - val_loss: 0.0543 -
val_accuracy: 0.9855 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 968    0    1    0    2    1    2    2    4    0]
 [  0 1123    3    1    1    2    3    0    2    0]
 [  5   3 1000    6    2    0    2    6    7    1]
 [  1    0    0  997    0    6    0    0    2    4]
 [  0    0    0    0  971    0    4    1    1    5]
 [  2    0    0    1    0  886    3    0    0    0]
 [  5    1    0    0    2    7  941    0    2    0]
 [  0    2    6   11    2    0    0  994    4    9]
 [  4    0    1    4    2    2    0    1  955    5]
 [  1    1    0    4    8    3    1    2    1  988]]
```

Precision: 0.9824
Recall: 0.9823

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
0	968	0	1	0	2	1	2	2	4	0
1	0	1123	3	1	1	2	3	0	2	0
2	5	3	1000	6	2	0	2	6	7	1
3	1	0	0	997	0	6	0	0	2	4
4	0	0	0	0	971	0	4	1	1	5
5	2	0	0	1	0	886	3	0	0	0
6	5	1	0	0	2	7	941	0	2	0
7	0	2	6	11	2	0	0	994	4	9
8	4	0	1	4	2	2	0	1	955	5
9	1	1	0	4	8	3	1	2	1	988
	0	1	2	3	4	5	6	7	8	9
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```
Training Model with relu activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 4s - loss: 0.2761 - accuracy: 0.9153 - val_loss: 0.1131 -
val_accuracy: 0.9670 - 4s/epoch - 9ms/step
Epoch 2/15
422/422 - 3s - loss: 0.1194 - accuracy: 0.9642 - val_loss: 0.0849 -
val_accuracy: 0.9780 - 3s/epoch - 7ms/step
Epoch 3/15
422/422 - 3s - loss: 0.0825 - accuracy: 0.9745 - val_loss: 0.0826 -
val_accuracy: 0.9753 - 3s/epoch - 7ms/step
Epoch 4/15
422/422 - 3s - loss: 0.0631 - accuracy: 0.9819 - val_loss: 0.0529 -
val_accuracy: 0.9845 - 3s/epoch - 7ms/step
Epoch 5/15
422/422 - 3s - loss: 0.0512 - accuracy: 0.9844 - val_loss: 0.0563 -
val_accuracy: 0.9847 - 3s/epoch - 7ms/step
Epoch 6/15
422/422 - 3s - loss: 0.0404 - accuracy: 0.9885 - val_loss: 0.0544 -
val_accuracy: 0.9840 - 3s/epoch - 7ms/step
Epoch 7/15
```

```
422/422 - 3s - loss: 0.0344 - accuracy: 0.9902 - val_loss: 0.0466 -  
val_accuracy: 0.9870 - 3s/epoch - 7ms/step  
Epoch 8/15  
422/422 - 3s - loss: 0.0293 - accuracy: 0.9917 - val_loss: 0.0444 -  
val_accuracy: 0.9863 - 3s/epoch - 7ms/step  
Epoch 9/15  
422/422 - 3s - loss: 0.0252 - accuracy: 0.9930 - val_loss: 0.0501 -  
val_accuracy: 0.9862 - 3s/epoch - 7ms/step  
Epoch 10/15  
422/422 - 3s - loss: 0.0213 - accuracy: 0.9940 - val_loss: 0.0435 -  
val_accuracy: 0.9877 - 3s/epoch - 7ms/step  
Epoch 11/15  
422/422 - 3s - loss: 0.0182 - accuracy: 0.9949 - val_loss: 0.0457 -  
val_accuracy: 0.9870 - 3s/epoch - 7ms/step  
Epoch 12/15  
422/422 - 3s - loss: 0.0158 - accuracy: 0.9959 - val_loss: 0.0505 -  
val_accuracy: 0.9865 - 3s/epoch - 7ms/step  
Epoch 13/15  
422/422 - 3s - loss: 0.0142 - accuracy: 0.9966 - val_loss: 0.0498 -  
val_accuracy: 0.9870 - 3s/epoch - 7ms/step  
Epoch 14/15  
422/422 - 3s - loss: 0.0119 - accuracy: 0.9972 - val_loss: 0.0463 -  
val_accuracy: 0.9880 - 3s/epoch - 7ms/step  
Epoch 15/15  
422/422 - 3s - loss: 0.0100 - accuracy: 0.9978 - val_loss: 0.0456 -  
val_accuracy: 0.9885 - 3s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 974  0  0  0  0  1  1  1  2  1]  
[  0 1130  1  1  0  1  1  0  1  0]  
[  3  2 1009  1  2  0  2  6  7  0]  
[  0  0  0 999  0  2  0  1  3  5]  
[  0  0  3  0 972  0  1  0  1  5]  
[  2  0  0  9  0 874  2  1  3  1]  
[  5  3  1  1  1  3 942  0  2  0]  
[  1  4  5  1  1  0  0 1009  3  4]  
[  4  0  0  2  0  1  0  1 962  4]  
[  1  1  0  4 11  1  0  3  0 988]]  
Precision: 0.9859  
Recall: 0.9859
```

Confusion Matrix for relu Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	974	0	0	0	0	1	1	1	2	1	1
0	974	0	0	0	0	1	1	1	2	1	1
1	0	1130	1	1	0	1	1	0	1	0	0
2	3	2	1009	1	2	0	2	6	7	0	0
3	0	0	0	999	0	2	0	1	3	5	0
4	0	0	3	0	972	0	1	0	1	5	0
5	2	0	0	9	0	874	2	1	3	1	0
6	5	3	1	1	1	3	942	0	2	0	0
7	1	4	5	1	1	0	0	0	1009	3	4
8	4	0	0	2	0	1	0	0	1	962	4
9	1	1	0	4	11	1	0	3	0	988	9

```

Training Model with relu activation, 1 conv_layers, 1 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 4s - loss: 0.2554 - accuracy: 0.9234 - val_loss: 0.0986 -
val_accuracy: 0.9757 - 4s/epoch - 9ms/step
Epoch 2/20
422/422 - 3s - loss: 0.0990 - accuracy: 0.9715 - val_loss: 0.0681 -
val_accuracy: 0.9837 - 3s/epoch - 7ms/step
Epoch 3/20
422/422 - 3s - loss: 0.0679 - accuracy: 0.9804 - val_loss: 0.0721 -
val_accuracy: 0.9830 - 3s/epoch - 7ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0514 - accuracy: 0.9852 - val_loss: 0.0624 -
val_accuracy: 0.9832 - 3s/epoch - 7ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0415 - accuracy: 0.9879 - val_loss: 0.0553 -
val_accuracy: 0.9855 - 3s/epoch - 7ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0347 - accuracy: 0.9902 - val_loss: 0.0503 -
val_accuracy: 0.9873 - 3s/epoch - 7ms/step
Epoch 7/20

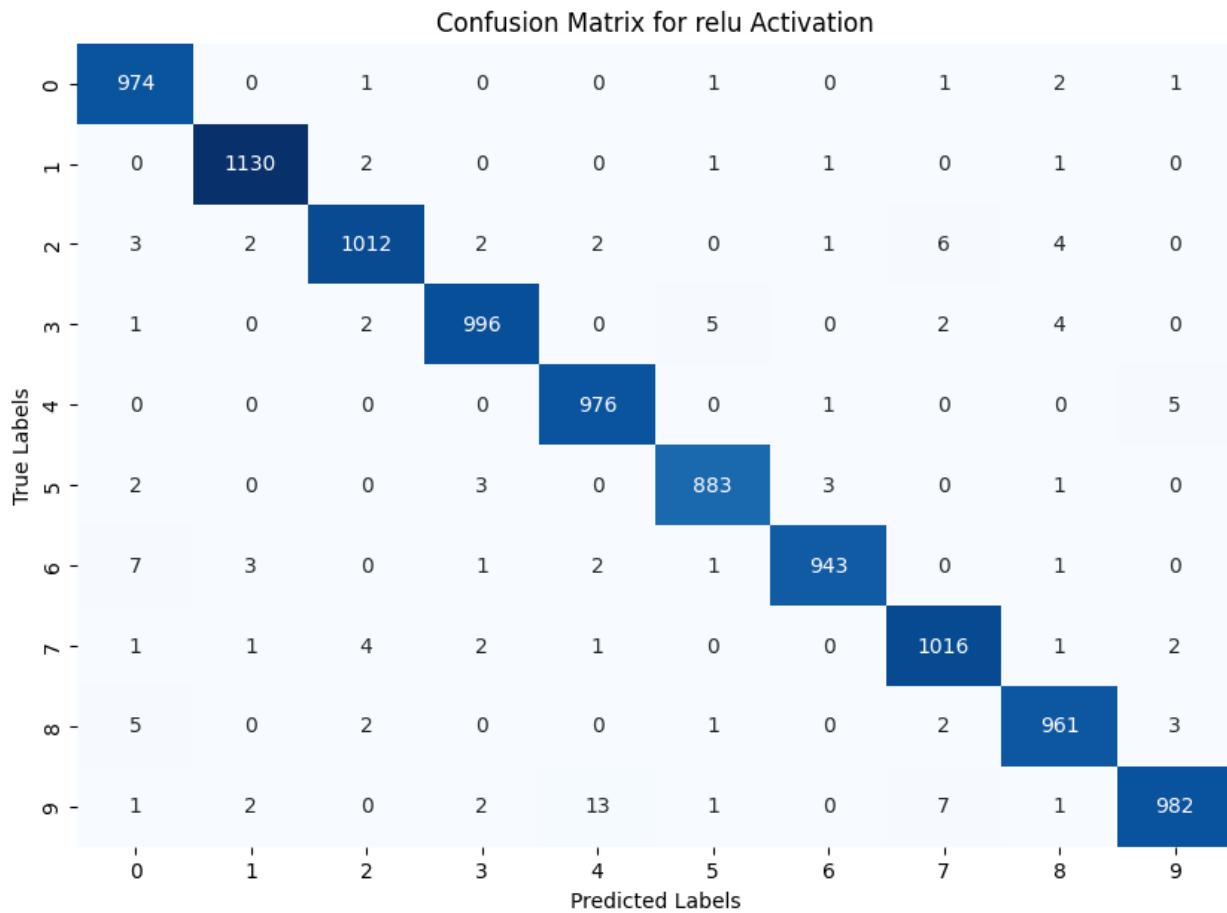
```

```
422/422 - 3s - loss: 0.0297 - accuracy: 0.9916 - val_loss: 0.0513 -  
val_accuracy: 0.9872 - 3s/epoch - 7ms/step  
Epoch 8/20  
422/422 - 3s - loss: 0.0249 - accuracy: 0.9929 - val_loss: 0.0490 -  
val_accuracy: 0.9880 - 3s/epoch - 7ms/step  
Epoch 9/20  
422/422 - 3s - loss: 0.0217 - accuracy: 0.9944 - val_loss: 0.0470 -  
val_accuracy: 0.9897 - 3s/epoch - 7ms/step  
Epoch 10/20  
422/422 - 3s - loss: 0.0182 - accuracy: 0.9954 - val_loss: 0.0474 -  
val_accuracy: 0.9892 - 3s/epoch - 7ms/step  
Epoch 11/20  
422/422 - 3s - loss: 0.0158 - accuracy: 0.9959 - val_loss: 0.0469 -  
val_accuracy: 0.9888 - 3s/epoch - 7ms/step  
Epoch 12/20  
422/422 - 3s - loss: 0.0137 - accuracy: 0.9966 - val_loss: 0.0472 -  
val_accuracy: 0.9887 - 3s/epoch - 7ms/step  
Epoch 13/20  
422/422 - 3s - loss: 0.0114 - accuracy: 0.9976 - val_loss: 0.0507 -  
val_accuracy: 0.9882 - 3s/epoch - 7ms/step  
Epoch 14/20  
422/422 - 3s - loss: 0.0103 - accuracy: 0.9978 - val_loss: 0.0471 -  
val_accuracy: 0.9887 - 3s/epoch - 7ms/step  
Epoch 15/20  
422/422 - 3s - loss: 0.0085 - accuracy: 0.9985 - val_loss: 0.0476 -  
val_accuracy: 0.9887 - 3s/epoch - 7ms/step  
Epoch 16/20  
422/422 - 3s - loss: 0.0077 - accuracy: 0.9986 - val_loss: 0.0511 -  
val_accuracy: 0.9873 - 3s/epoch - 7ms/step  
Epoch 17/20  
422/422 - 3s - loss: 0.0066 - accuracy: 0.9990 - val_loss: 0.0470 -  
val_accuracy: 0.9905 - 3s/epoch - 7ms/step  
Epoch 18/20  
422/422 - 3s - loss: 0.0057 - accuracy: 0.9992 - val_loss: 0.0499 -  
val_accuracy: 0.9890 - 3s/epoch - 7ms/step  
Epoch 19/20  
422/422 - 3s - loss: 0.0052 - accuracy: 0.9993 - val_loss: 0.0495 -  
val_accuracy: 0.9895 - 3s/epoch - 7ms/step  
Epoch 20/20  
422/422 - 3s - loss: 0.0044 - accuracy: 0.9996 - val_loss: 0.0488 -  
val_accuracy: 0.9893 - 3s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 974   0   1   0   0   1   0   1   2   1]  
 [  0 1130   2   0   0   1   1   0   1   0]  
 [  3   2 1012   2   2   0   1   6   4   0]  
 [  1   0   2  996   0   5   0   2   4   0]  
 [  0   0   0   0  976   0   1   0   0   5]]
```

```
[ 2 0 0 3 0 883 3 0 1 0]
[ 7 3 0 1 2 1 943 0 1 0]
[ 1 1 4 2 1 0 0 1016 1 2]
[ 5 0 2 0 0 1 0 2 961 3]
[ 1 2 0 2 13 1 0 7 1 982]]
```

Precision: 0.9873

Recall: 0.9873



Training Model with relu activation, 1 conv_layers, 1 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 3s - loss: 0.3498 - accuracy: 0.8941 - val_loss: 0.1492 - val_accuracy: 0.9607 - 3s/epoch - 14ms/step

Epoch 2/5

211/211 - 2s - loss: 0.1486 - accuracy: 0.9572 - val_loss: 0.1037 - val_accuracy: 0.9732 - 2s/epoch - 11ms/step

Epoch 3/5

211/211 - 2s - loss: 0.1064 - accuracy: 0.9695 - val_loss: 0.0829 - val_accuracy: 0.9777 - 2s/epoch - 11ms/step

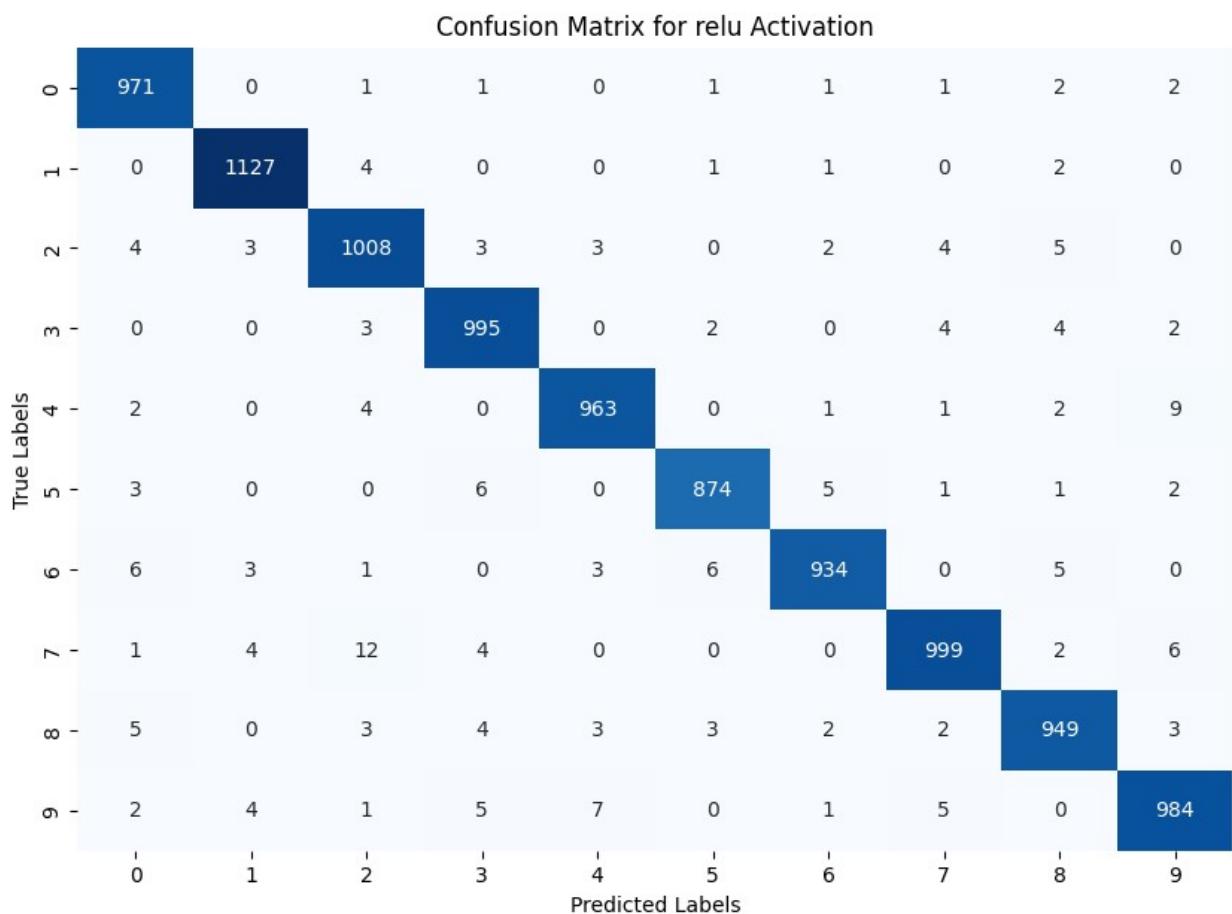
Epoch 4/5

211/211 - 2s - loss: 0.0832 - accuracy: 0.9761 - val_loss: 0.0765 -

```

val_accuracy: 0.9797 - 2s/epoch - 11ms/step
Epoch 5/5
211/211 - 2s - loss: 0.0688 - accuracy: 0.9806 - val_loss: 0.0636 -
val_accuracy: 0.9847 - 2s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 971   0   1   1   0   1   1   1   2   2]
 [  0 1127   4   0   0   1   1   0   2   0]
 [  4   3 1008   3   3   0   2   4   5   0]
 [  0   0   3 995   0   2   0   4   4   2]
 [  2   0   4   0 963   0   1   1   2   9]
 [  3   0   0   6   0 874   5   1   1   2]
 [  6   3   1   0   3   6 934   0   5   0]
 [  1   4  12   4   0   0   0 999   2   6]
 [  5   0   3   4   3   3   2   2 949   3]
 [  2   4   1   5   7   0   1   5   0 984]]
Precision: 0.9804
Recall: 0.9804

```

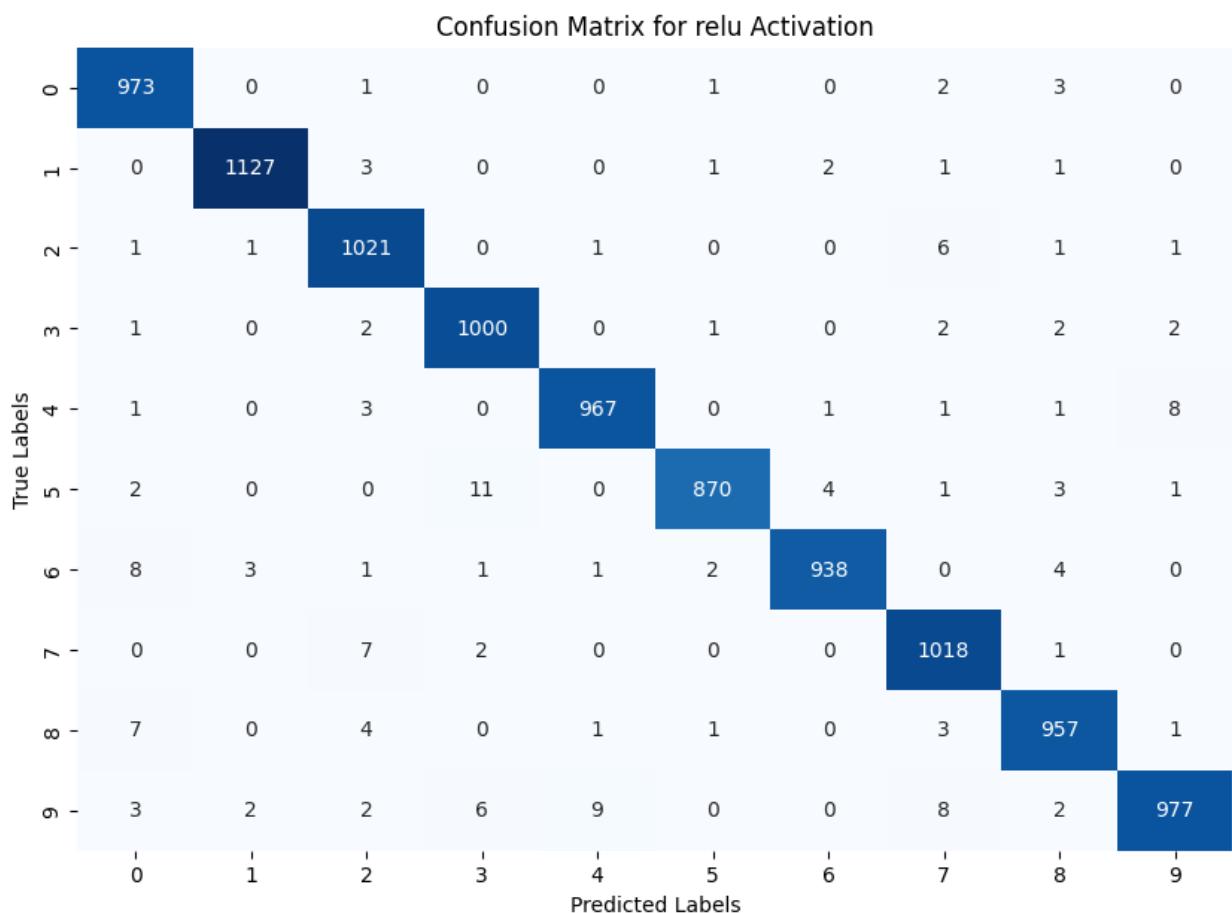


```
Training Model with relu activation, 1 conv_layers, 1 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 3s - loss: 0.3474 - accuracy: 0.8949 - val_loss: 0.1562 -
val_accuracy: 0.9580 - 3s/epoch - 14ms/step
Epoch 2/15
211/211 - 2s - loss: 0.1607 - accuracy: 0.9529 - val_loss: 0.1102 -
val_accuracy: 0.9703 - 2s/epoch - 11ms/step
Epoch 3/15
211/211 - 2s - loss: 0.1177 - accuracy: 0.9653 - val_loss: 0.0883 -
val_accuracy: 0.9772 - 2s/epoch - 11ms/step
Epoch 4/15
211/211 - 2s - loss: 0.0938 - accuracy: 0.9729 - val_loss: 0.0763 -
val_accuracy: 0.9805 - 2s/epoch - 11ms/step
Epoch 5/15
211/211 - 2s - loss: 0.0757 - accuracy: 0.9785 - val_loss: 0.0774 -
val_accuracy: 0.9788 - 2s/epoch - 11ms/step
Epoch 6/15
211/211 - 2s - loss: 0.0647 - accuracy: 0.9815 - val_loss: 0.0672 -
val_accuracy: 0.9823 - 2s/epoch - 11ms/step
Epoch 7/15
211/211 - 2s - loss: 0.0545 - accuracy: 0.9846 - val_loss: 0.0590 -
val_accuracy: 0.9845 - 2s/epoch - 11ms/step
Epoch 8/15
211/211 - 2s - loss: 0.0493 - accuracy: 0.9859 - val_loss: 0.0530 -
val_accuracy: 0.9877 - 2s/epoch - 11ms/step
Epoch 9/15
211/211 - 2s - loss: 0.0437 - accuracy: 0.9877 - val_loss: 0.0553 -
val_accuracy: 0.9857 - 2s/epoch - 11ms/step
Epoch 10/15
211/211 - 2s - loss: 0.0385 - accuracy: 0.9891 - val_loss: 0.0545 -
val_accuracy: 0.9858 - 2s/epoch - 11ms/step
Epoch 11/15
211/211 - 2s - loss: 0.0348 - accuracy: 0.9903 - val_loss: 0.0526 -
val_accuracy: 0.9873 - 2s/epoch - 11ms/step
Epoch 12/15
211/211 - 2s - loss: 0.0315 - accuracy: 0.9912 - val_loss: 0.0516 -
val_accuracy: 0.9867 - 2s/epoch - 11ms/step
Epoch 13/15
211/211 - 2s - loss: 0.0297 - accuracy: 0.9917 - val_loss: 0.0482 -
val_accuracy: 0.9887 - 2s/epoch - 11ms/step
Epoch 14/15
211/211 - 2s - loss: 0.0258 - accuracy: 0.9930 - val_loss: 0.0491 -
val_accuracy: 0.9878 - 2s/epoch - 11ms/step
Epoch 15/15
211/211 - 2s - loss: 0.0244 - accuracy: 0.9936 - val_loss: 0.0487 -
val_accuracy: 0.9887 - 2s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
```

```
[[ 973  0   1   0   0   1   0   2   3   0]
 [ 0 1127  3   0   0   1   2   1   1   0]
 [ 1   1 1021  0   1   0   0   6   1   1]
 [ 1   0   2 1000  0   1   0   2   2   2]
 [ 1   0   3   0 967  0   1   1   1   8]
 [ 2   0   0   11  0 870  4   1   3   1]
 [ 8   3   1   1   1   2 938  0   4   0]
 [ 0   0   7   2   0   0   0 1018  1   0]
 [ 7   0   4   0   1   1   0   3 957  1]
 [ 3   2   2   6   9   0   0   8   2 977]]
```

Precision: 0.9849

Recall: 0.9848



Training Model with relu activation, 1 conv_layers, 1 dense layers,
256 batch size, 20 epochs..

Epoch 1/20

211/211 - 3s - loss: 0.3394 - accuracy: 0.8984 - val_loss: 0.1428 -
val_accuracy: 0.9620 - 3s/epoch - 14ms/step

Epoch 2/20

211/211 - 2s - loss: 0.1455 - accuracy: 0.9577 - val_loss: 0.1010 -
val_accuracy: 0.9733 - 2s/epoch - 11ms/step

```
Epoch 3/20
211/211 - 2s - loss: 0.1045 - accuracy: 0.9704 - val_loss: 0.0864 -
val_accuracy: 0.9772 - 2s/epoch - 11ms/step
Epoch 4/20
211/211 - 2s - loss: 0.0817 - accuracy: 0.9770 - val_loss: 0.0742 -
val_accuracy: 0.9803 - 2s/epoch - 11ms/step
Epoch 5/20
211/211 - 2s - loss: 0.0677 - accuracy: 0.9805 - val_loss: 0.0675 -
val_accuracy: 0.9802 - 2s/epoch - 11ms/step
Epoch 6/20
211/211 - 2s - loss: 0.0570 - accuracy: 0.9836 - val_loss: 0.0618 -
val_accuracy: 0.9835 - 2s/epoch - 11ms/step
Epoch 7/20
211/211 - 2s - loss: 0.0496 - accuracy: 0.9858 - val_loss: 0.0573 -
val_accuracy: 0.9848 - 2s/epoch - 11ms/step
Epoch 8/20
211/211 - 2s - loss: 0.0440 - accuracy: 0.9875 - val_loss: 0.0540 -
val_accuracy: 0.9858 - 2s/epoch - 11ms/step
Epoch 9/20
211/211 - 2s - loss: 0.0392 - accuracy: 0.9890 - val_loss: 0.0535 -
val_accuracy: 0.9865 - 2s/epoch - 11ms/step
Epoch 10/20
211/211 - 2s - loss: 0.0353 - accuracy: 0.9903 - val_loss: 0.0534 -
val_accuracy: 0.9863 - 2s/epoch - 11ms/step
Epoch 11/20
211/211 - 2s - loss: 0.0312 - accuracy: 0.9916 - val_loss: 0.0494 -
val_accuracy: 0.9865 - 2s/epoch - 11ms/step
Epoch 12/20
211/211 - 2s - loss: 0.0284 - accuracy: 0.9923 - val_loss: 0.0495 -
val_accuracy: 0.9873 - 2s/epoch - 11ms/step
Epoch 13/20
211/211 - 2s - loss: 0.0263 - accuracy: 0.9927 - val_loss: 0.0494 -
val_accuracy: 0.9873 - 2s/epoch - 11ms/step
Epoch 14/20
211/211 - 2s - loss: 0.0239 - accuracy: 0.9936 - val_loss: 0.0480 -
val_accuracy: 0.9885 - 2s/epoch - 11ms/step
Epoch 15/20
211/211 - 2s - loss: 0.0219 - accuracy: 0.9942 - val_loss: 0.0504 -
val_accuracy: 0.9860 - 2s/epoch - 11ms/step
Epoch 16/20
211/211 - 2s - loss: 0.0197 - accuracy: 0.9949 - val_loss: 0.0491 -
val_accuracy: 0.9875 - 2s/epoch - 11ms/step
Epoch 17/20
211/211 - 2s - loss: 0.0181 - accuracy: 0.9957 - val_loss: 0.0493 -
val_accuracy: 0.9863 - 2s/epoch - 11ms/step
Epoch 18/20
211/211 - 2s - loss: 0.0169 - accuracy: 0.9957 - val_loss: 0.0494 -
val_accuracy: 0.9873 - 2s/epoch - 11ms/step
Epoch 19/20
```

```
211/211 - 2s - loss: 0.0156 - accuracy: 0.9961 - val_loss: 0.0470 -  
val_accuracy: 0.9875 - 2s/epoch - 11ms/step
```

```
Epoch 20/20
```

```
211/211 - 2s - loss: 0.0141 - accuracy: 0.9969 - val_loss: 0.0508 -  
val_accuracy: 0.9880 - 2s/epoch - 11ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

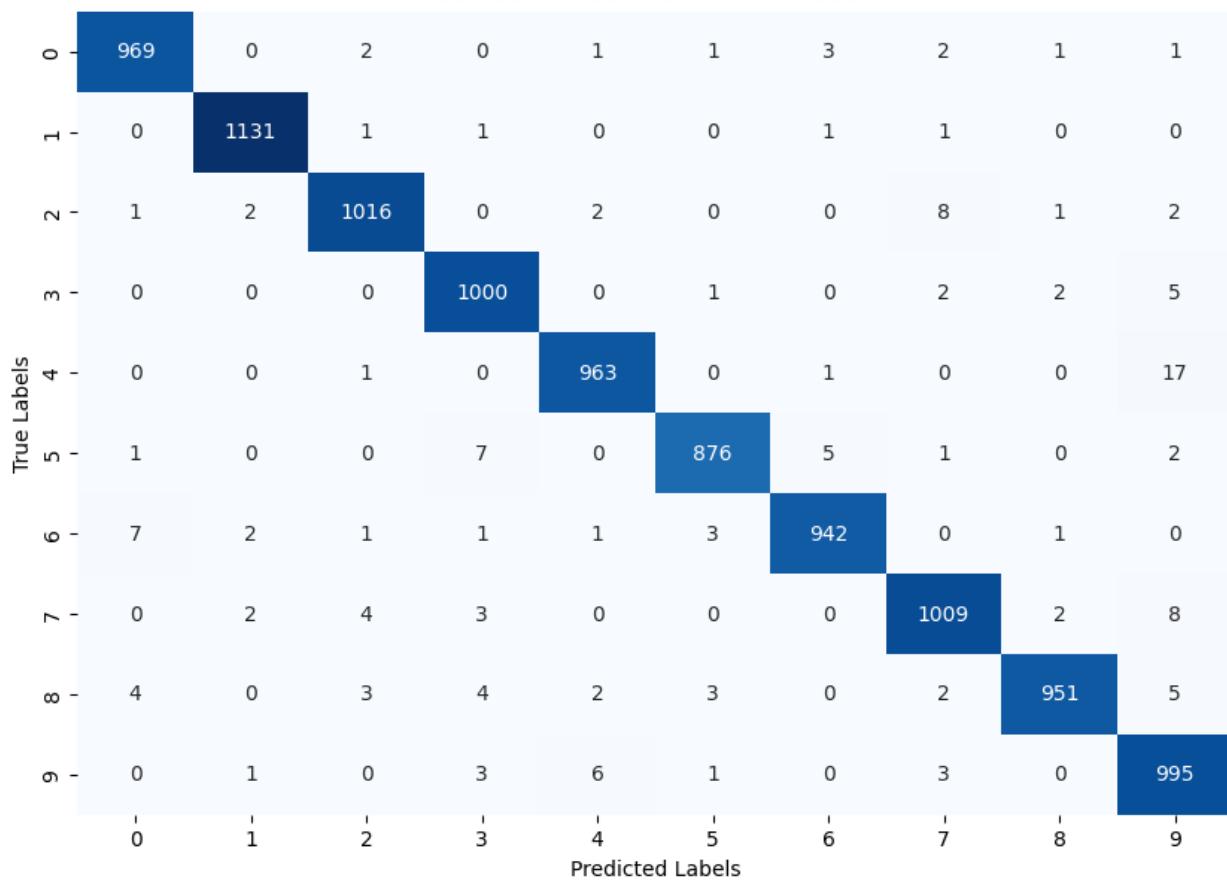
```
Confusion Matrix:
```

```
[[ 969   0   2   0   1   1   3   2   1   1]
 [ 0 1131   1   1   0   0   1   1   0   0]
 [ 1   2 1016   0   2   0   0   8   1   2]
 [ 0   0   0 1000   0   1   0   2   2   5]
 [ 0   0   1   0 963   0   1   0   0 17]
 [ 1   0   0   7   0 876   5   1   0   2]
 [ 7   2   1   1   1   3 942   0   1   0]
 [ 0   2   4   3   0   0   0 1009   2   8]
 [ 4   0   3   4   2   3   0   2 951   5]
 [ 0   1   0   3   6   1   0   3   0 995]]
```

```
Precision: 0.9853
```

```
Recall: 0.9852
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 1 conv_layers, 1 dense layers, 64
batch size, 5 epochs..
Epoch 1/5
844/844 - 6s - loss: 0.2013 - accuracy: 0.9368 - val_loss: 0.0873 -
val_accuracy: 0.9747 - 6s/epoch - 7ms/step
Epoch 2/5
844/844 - 5s - loss: 0.0784 - accuracy: 0.9755 - val_loss: 0.0611 -
val_accuracy: 0.9823 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 0.0514 - accuracy: 0.9844 - val_loss: 0.0565 -
val_accuracy: 0.9830 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 0.0377 - accuracy: 0.9883 - val_loss: 0.0573 -
val_accuracy: 0.9842 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.0278 - accuracy: 0.9919 - val_loss: 0.0494 -
val_accuracy: 0.9865 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 970    0    0    0    0    2    2    1    2    3]
 [  0 1124    1    5    0    1    3    0    1    0]
 [  4    1 1002    4    1    0    1    6   13    0]
 [  0    0    1  996    0    5    0    1    3    4]
 [  0    0    1    0  967    0    0    0    0   14]
 [  1    0    0    3    0  884    3    0    1    0]
 [  7    2    0    0    2    3  943    0    1    0]
 [  0    1    4    2    0    0    0 1012    3    6]
 [  6    1    1    1    0    2    0    1  956    6]
 [  3    2    0    3    3    1    0    2    0  995]]
```

Precision: 0.9850
Recall: 0.9849

Confusion Matrix for relu Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	970	0	0	0	0	2	2	1	2	3	970
0	970	0	0	0	0	2	2	1	2	3	970
1	0	1124	1	5	0	1	3	0	1	0	1124
2	4	1	1002	4	1	0	1	6	13	0	1002
3	0	0	1	996	0	5	0	1	3	4	996
4	0	0	1	0	967	0	0	0	0	14	967
5	1	0	0	3	0	884	3	0	1	0	884
6	7	2	0	0	2	3	943	0	1	0	943
7	0	1	4	2	0	0	0	0	3	6	1012
8	6	1	1	1	0	2	0	1	956	6	956
9	3	2	0	3	3	1	0	2	0	995	995
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```
Training Model with relu activation, 1 conv_layers, 1 dense layers, 64
batch size, 15 epochs..
Epoch 1/15
844/844 - 6s - loss: 0.2169 - accuracy: 0.9334 - val_loss: 0.0892 -
val_accuracy: 0.9747 - 6s/epoch - 7ms/step
Epoch 2/15
844/844 - 5s - loss: 0.0834 - accuracy: 0.9746 - val_loss: 0.0742 -
val_accuracy: 0.9787 - 5s/epoch - 6ms/step
Epoch 3/15
844/844 - 5s - loss: 0.0544 - accuracy: 0.9836 - val_loss: 0.0521 -
val_accuracy: 0.9857 - 5s/epoch - 6ms/step
Epoch 4/15
844/844 - 5s - loss: 0.0386 - accuracy: 0.9881 - val_loss: 0.0520 -
val_accuracy: 0.9852 - 5s/epoch - 6ms/step
Epoch 5/15
844/844 - 5s - loss: 0.0297 - accuracy: 0.9914 - val_loss: 0.0526 -
val_accuracy: 0.9862 - 5s/epoch - 6ms/step
Epoch 6/15
844/844 - 5s - loss: 0.0226 - accuracy: 0.9933 - val_loss: 0.0451 -
val_accuracy: 0.9878 - 5s/epoch - 6ms/step
Epoch 7/15
```

```
844/844 - 5s - loss: 0.0162 - accuracy: 0.9956 - val_loss: 0.0444 -  
val_accuracy: 0.9880 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.0133 - accuracy: 0.9965 - val_loss: 0.0427 -  
val_accuracy: 0.9888 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.0098 - accuracy: 0.9976 - val_loss: 0.0450 -  
val_accuracy: 0.9878 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.0076 - accuracy: 0.9985 - val_loss: 0.0481 -  
val_accuracy: 0.9882 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.0056 - accuracy: 0.9991 - val_loss: 0.0438 -  
val_accuracy: 0.9893 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.0042 - accuracy: 0.9994 - val_loss: 0.0519 -  
val_accuracy: 0.9863 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.0035 - accuracy: 0.9996 - val_loss: 0.0433 -  
val_accuracy: 0.9888 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.0025 - accuracy: 0.9999 - val_loss: 0.0473 -  
val_accuracy: 0.9888 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.0020 - accuracy: 0.9998 - val_loss: 0.0489 -  
val_accuracy: 0.9897 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 972 0 1 0 0 1 2 1 2 1 ]  
[ 0 1127 1 1 0 1 2 2 1 0 ]  
[ 2 1 1019 0 1 0 1 4 4 0 ]  
[ 0 0 2 1002 0 2 0 1 1 2 ]  
[ 0 0 1 0 970 0 2 1 1 7 ]  
[ 2 0 0 8 0 875 4 0 1 2 ]  
[ 7 2 0 1 1 2 945 0 0 0 ]  
[ 1 1 6 1 0 0 0 1017 2 0 ]  
[ 3 0 1 2 0 2 0 2 961 3 ]  
[ 2 1 0 4 5 1 0 7 0 989 ]]
```

Precision: 0.9877
Recall: 0.9877

Confusion Matrix for relu Activation											
	0	1	2	3	4	5	6	7	8	9	
0	972	0	1	0	0	1	2	1	2	1	0
1	0	1127	1	1	0	1	2	2	1	0	
2	2	1	1019	0	1	0	1	4	4	0	
3	0	0	2	1002	0	2	0	1	1	2	
4	0	0	1	0	970	0	2	1	1	7	
5	2	0	0	8	0	875	4	0	1	2	
6	7	2	0	1	1	2	945	0	0	0	
7	1	1	6	1	0	0	0	1017	2	0	
8	3	0	1	2	0	2	0	2	961	3	
9	2	1	0	4	5	1	0	7	0	989	
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

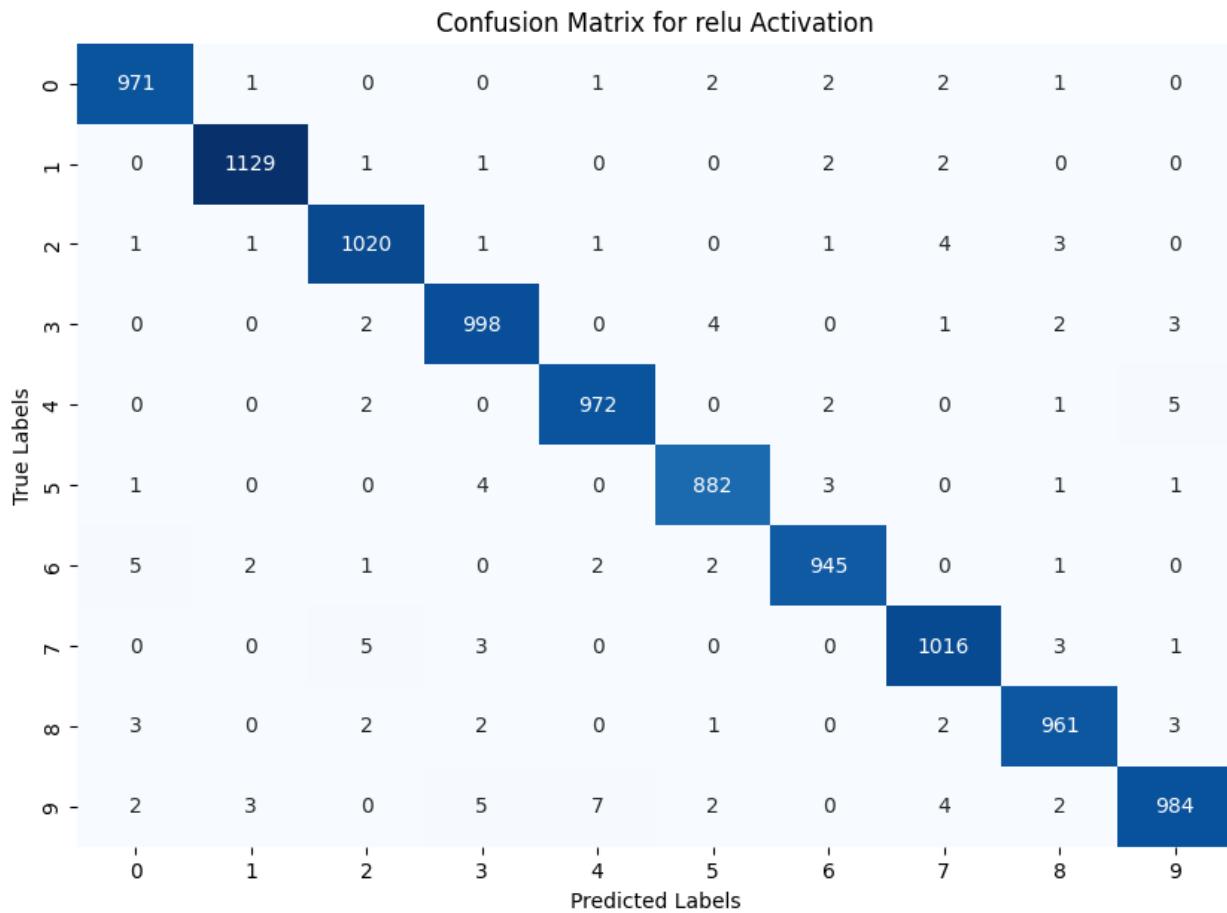
```
Training Model with relu activation, 1 conv_layers, 1 dense layers, 64
batch size, 20 epochs..
Epoch 1/20
844/844 - 6s - loss: 0.1951 - accuracy: 0.9407 - val_loss: 0.0799 -
val_accuracy: 0.9775 - 6s/epoch - 7ms/step
Epoch 2/20
844/844 - 5s - loss: 0.0686 - accuracy: 0.9801 - val_loss: 0.0608 -
val_accuracy: 0.9835 - 5s/epoch - 6ms/step
Epoch 3/20
844/844 - 5s - loss: 0.0457 - accuracy: 0.9868 - val_loss: 0.0556 -
val_accuracy: 0.9848 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: 0.0322 - accuracy: 0.9904 - val_loss: 0.0567 -
val_accuracy: 0.9837 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: 0.0234 - accuracy: 0.9933 - val_loss: 0.0436 -
val_accuracy: 0.9883 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 0.0171 - accuracy: 0.9952 - val_loss: 0.0476 -
val_accuracy: 0.9880 - 5s/epoch - 6ms/step
Epoch 7/20
```

```
844/844 - 5s - loss: 0.0132 - accuracy: 0.9964 - val_loss: 0.0459 -  
val_accuracy: 0.9898 - 5s/epoch - 6ms/step  
Epoch 8/20  
844/844 - 5s - loss: 0.0097 - accuracy: 0.9978 - val_loss: 0.0535 -  
val_accuracy: 0.9883 - 5s/epoch - 6ms/step  
Epoch 9/20  
844/844 - 5s - loss: 0.0078 - accuracy: 0.9984 - val_loss: 0.0448 -  
val_accuracy: 0.9900 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: 0.0056 - accuracy: 0.9991 - val_loss: 0.0461 -  
val_accuracy: 0.9903 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: 0.0039 - accuracy: 0.9996 - val_loss: 0.0475 -  
val_accuracy: 0.9893 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 5s - loss: 0.0033 - accuracy: 0.9996 - val_loss: 0.0475 -  
val_accuracy: 0.9897 - 5s/epoch - 6ms/step  
Epoch 13/20  
844/844 - 5s - loss: 0.0023 - accuracy: 0.9999 - val_loss: 0.0454 -  
val_accuracy: 0.9900 - 5s/epoch - 6ms/step  
Epoch 14/20  
844/844 - 5s - loss: 0.0019 - accuracy: 0.9999 - val_loss: 0.0501 -  
val_accuracy: 0.9890 - 5s/epoch - 6ms/step  
Epoch 15/20  
844/844 - 5s - loss: 0.0016 - accuracy: 0.9999 - val_loss: 0.0487 -  
val_accuracy: 0.9903 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: 0.0013 - accuracy: 0.9999 - val_loss: 0.0497 -  
val_accuracy: 0.9898 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: 0.0011 - accuracy: 1.0000 - val_loss: 0.0495 -  
val_accuracy: 0.9902 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: 0.0010 - accuracy: 1.0000 - val_loss: 0.0512 -  
val_accuracy: 0.9900 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: 8.8756e-04 - accuracy: 1.0000 - val_loss: 0.0511  
- val_accuracy: 0.9900 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: 7.8007e-04 - accuracy: 1.0000 - val_loss: 0.0502  
- val_accuracy: 0.9898 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 971   1   0   0   1   2   2   2   1   0]  
 [  0 1129   1   1   0   0   2   2   0   0]  
 [  1   1 1020   1   1   0   1   4   3   0]  
 [  0   0   2  998   0   4   0   1   2   3]  
 [  0   0   2   0  972   0   2   0   1   5]]
```

```
[ 1 0 0 4 0 882 3 0 1 1]
[ 5 2 1 0 2 2 945 0 1 0]
[ 0 0 5 3 0 0 0 1016 3 1]
[ 3 0 2 2 0 1 0 2 961 3]
[ 2 3 0 5 7 2 0 4 2 984]]
```

Precision: 0.9878

Recall: 0.9878



Training Model with relu activation, 1 conv_layers, 1 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 4s - loss: 0.2655 - accuracy: 0.9198 - val_loss: 0.1122 - val_accuracy: 0.9680 - 4s/epoch - 10ms/step

Epoch 2/5

422/422 - 4s - loss: 0.1091 - accuracy: 0.9685 - val_loss: 0.0804 - val_accuracy: 0.9782 - 4s/epoch - 8ms/step

Epoch 3/5

422/422 - 4s - loss: 0.0754 - accuracy: 0.9781 - val_loss: 0.0612 - val_accuracy: 0.9835 - 4s/epoch - 8ms/step

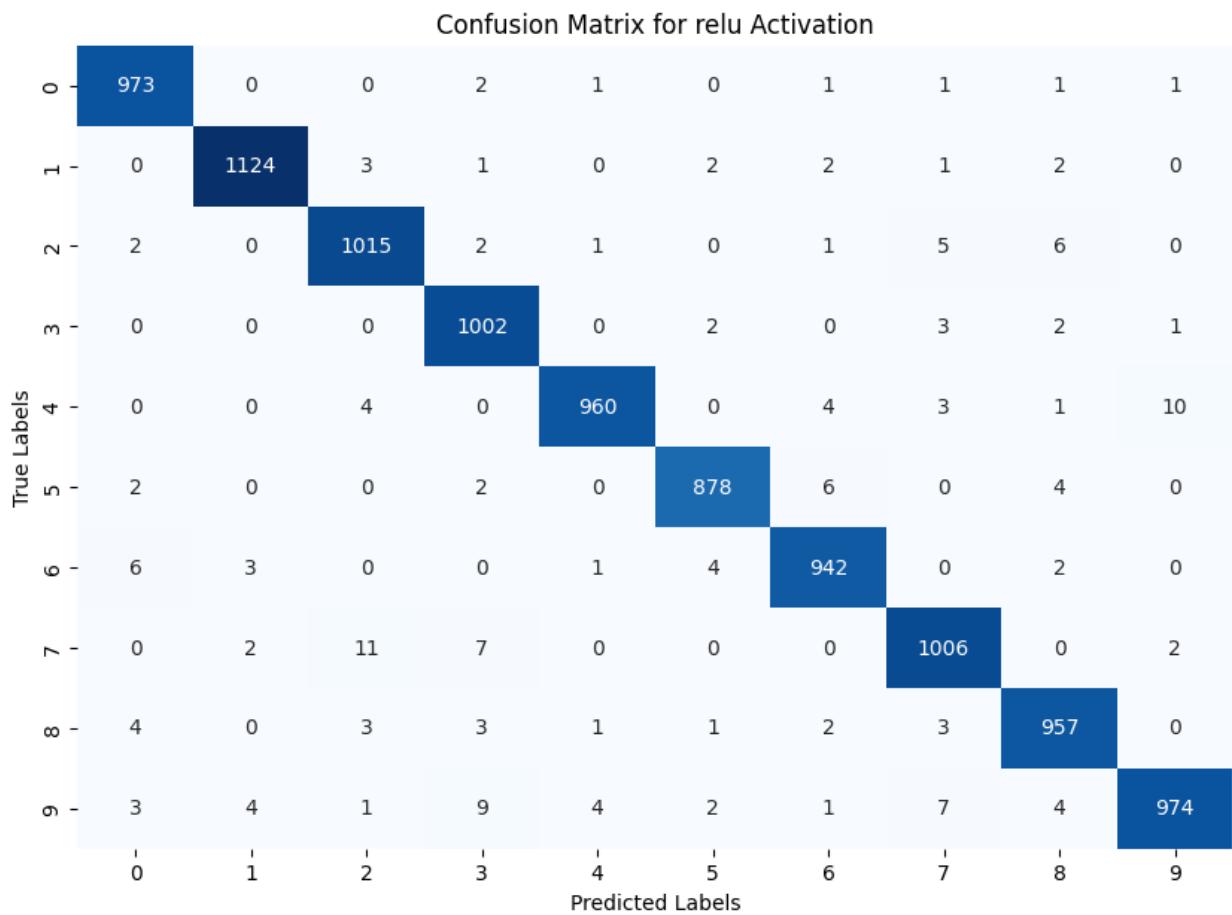
Epoch 4/5

422/422 - 4s - loss: 0.0555 - accuracy: 0.9842 - val_loss: 0.0585 -

```

val_accuracy: 0.9838 - 4s/epoch - 8ms/step
Epoch 5/5
422/422 - 4s - loss: 0.0436 - accuracy: 0.9874 - val_loss: 0.0543 -
val_accuracy: 0.9853 - 4s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 973   0   0   2   1   0   1   1   1   1]
 [ 0 1124   3   1   0   2   2   1   2   0]
 [ 2   0 1015   2   1   0   1   5   6   0]
 [ 0   0   0 1002   0   2   0   3   2   1]
 [ 0   0   4   0 960   0   4   3   1 10]
 [ 2   0   0   2   0 878   6   0   4   0]
 [ 6   3   0   0   1   4 942   0   2   0]
 [ 0   2   11   7   0   0   0 1006   0   2]
 [ 4   0   3   3   1   1   2   3 957   0]
 [ 3   4   1   9   4   2   1   7   4 974]]
Precision: 0.9831
Recall: 0.9831

```



Training Model with relu activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epochs..

Epoch 1/15
422/422 - 4s - loss: 0.2646 - accuracy: 0.9199 - val_loss: 0.1121 -
val_accuracy: 0.9693 - 4s/epoch - 10ms/step

Epoch 2/15
422/422 - 4s - loss: 0.1117 - accuracy: 0.9671 - val_loss: 0.0910 -
val_accuracy: 0.9738 - 4s/epoch - 8ms/step

Epoch 3/15
422/422 - 4s - loss: 0.0783 - accuracy: 0.9766 - val_loss: 0.0683 -
val_accuracy: 0.9798 - 4s/epoch - 8ms/step

Epoch 4/15
422/422 - 4s - loss: 0.0601 - accuracy: 0.9827 - val_loss: 0.0653 -
val_accuracy: 0.9832 - 4s/epoch - 8ms/step

Epoch 5/15
422/422 - 4s - loss: 0.0482 - accuracy: 0.9861 - val_loss: 0.0540 -
val_accuracy: 0.9845 - 4s/epoch - 8ms/step

Epoch 6/15
422/422 - 3s - loss: 0.0394 - accuracy: 0.9886 - val_loss: 0.0566 -
val_accuracy: 0.9842 - 3s/epoch - 8ms/step

Epoch 7/15
422/422 - 4s - loss: 0.0330 - accuracy: 0.9904 - val_loss: 0.0536 -
val_accuracy: 0.9850 - 4s/epoch - 8ms/step

Epoch 8/15
422/422 - 3s - loss: 0.0272 - accuracy: 0.9924 - val_loss: 0.0556 -
val_accuracy: 0.9857 - 3s/epoch - 8ms/step

Epoch 9/15
422/422 - 3s - loss: 0.0233 - accuracy: 0.9934 - val_loss: 0.0549 -
val_accuracy: 0.9847 - 3s/epoch - 8ms/step

Epoch 10/15
422/422 - 4s - loss: 0.0194 - accuracy: 0.9949 - val_loss: 0.0485 -
val_accuracy: 0.9882 - 4s/epoch - 8ms/step

Epoch 11/15
422/422 - 4s - loss: 0.0164 - accuracy: 0.9961 - val_loss: 0.0495 -
val_accuracy: 0.9878 - 4s/epoch - 8ms/step

Epoch 12/15
422/422 - 3s - loss: 0.0132 - accuracy: 0.9969 - val_loss: 0.0517 -
val_accuracy: 0.9868 - 3s/epoch - 8ms/step

Epoch 13/15
422/422 - 3s - loss: 0.0112 - accuracy: 0.9976 - val_loss: 0.0452 -
val_accuracy: 0.9885 - 3s/epoch - 8ms/step

Epoch 14/15
422/422 - 4s - loss: 0.0094 - accuracy: 0.9981 - val_loss: 0.0434 -
val_accuracy: 0.9895 - 4s/epoch - 8ms/step

Epoch 15/15
422/422 - 3s - loss: 0.0079 - accuracy: 0.9986 - val_loss: 0.0483 -
val_accuracy: 0.9880 - 3s/epoch - 8ms/step

313/313 [=====] - 1s 2ms/step

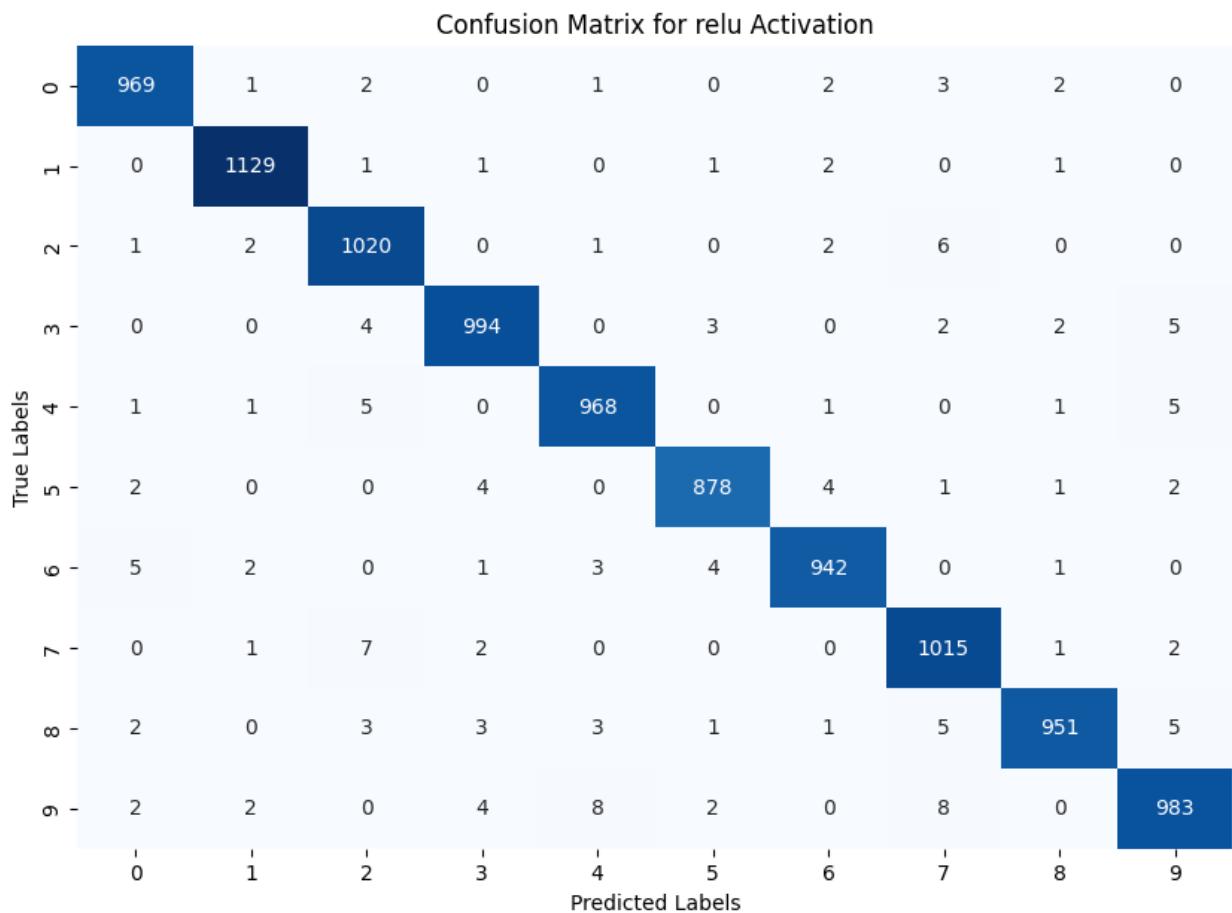
Results for activation function: relu

Confusion Matrix:

```
[[ 969  1  2  0  1  0  2  3  2  0]
 [ 0 1129  1  1  0  1  2  0  1  0]
 [ 1 2 1020  0  1  0  2  6  0  0]
 [ 0 0 4 994  0  3  0  2  2  5]
 [ 1 1 5 0 968  0  1  0  1  5]
 [ 2 0 0 4 0 878  4  1  1  2]
 [ 5 2 0 1 3 4 942  0  1  0]
 [ 0 1 7 2 0 0 0 1015  1  2]
 [ 2 0 3 3 3 1 1 5 951  5]
 [ 2 2 0 4 8 2 0 8 0 983]]
```

Precision: 0.9849

Recall: 0.9849



Training Model with relu activation, 1 conv_layers, 1 dense layers, 128 batch size, 20 epochs..

Epoch 1/20

422/422 - 4s - loss: 0.2728 - accuracy: 0.9171 - val_loss: 0.1264 - val_accuracy: 0.9628 - 4s/epoch - 9ms/step

Epoch 2/20

422/422 - 3s - loss: 0.1181 - accuracy: 0.9640 - val_loss: 0.0865 - val_accuracy: 0.9770 - 3s/epoch - 8ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.0835 - accuracy: 0.9750 - val_loss: 0.0688 -
val_accuracy: 0.9798 - 3s/epoch - 8ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0634 - accuracy: 0.9806 - val_loss: 0.0612 -
val_accuracy: 0.9828 - 3s/epoch - 8ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0508 - accuracy: 0.9843 - val_loss: 0.0548 -
val_accuracy: 0.9863 - 3s/epoch - 8ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0426 - accuracy: 0.9869 - val_loss: 0.0570 -
val_accuracy: 0.9848 - 3s/epoch - 8ms/step
Epoch 7/20
422/422 - 3s - loss: 0.0355 - accuracy: 0.9897 - val_loss: 0.0541 -
val_accuracy: 0.9852 - 3s/epoch - 8ms/step
Epoch 8/20
422/422 - 3s - loss: 0.0306 - accuracy: 0.9909 - val_loss: 0.0511 -
val_accuracy: 0.9855 - 3s/epoch - 8ms/step
Epoch 9/20
422/422 - 4s - loss: 0.0252 - accuracy: 0.9927 - val_loss: 0.0468 -
val_accuracy: 0.9865 - 4s/epoch - 8ms/step
Epoch 10/20
422/422 - 3s - loss: 0.0213 - accuracy: 0.9938 - val_loss: 0.0467 -
val_accuracy: 0.9883 - 3s/epoch - 8ms/step
Epoch 11/20
422/422 - 4s - loss: 0.0181 - accuracy: 0.9952 - val_loss: 0.0492 -
val_accuracy: 0.9858 - 4s/epoch - 9ms/step
Epoch 12/20
422/422 - 4s - loss: 0.0145 - accuracy: 0.9963 - val_loss: 0.0505 -
val_accuracy: 0.9863 - 4s/epoch - 8ms/step
Epoch 13/20
422/422 - 4s - loss: 0.0130 - accuracy: 0.9969 - val_loss: 0.0463 -
val_accuracy: 0.9882 - 4s/epoch - 8ms/step
Epoch 14/20
422/422 - 4s - loss: 0.0110 - accuracy: 0.9976 - val_loss: 0.0503 -
val_accuracy: 0.9878 - 4s/epoch - 8ms/step
Epoch 15/20
422/422 - 4s - loss: 0.0094 - accuracy: 0.9979 - val_loss: 0.0463 -
val_accuracy: 0.9880 - 4s/epoch - 8ms/step
Epoch 16/20
422/422 - 4s - loss: 0.0082 - accuracy: 0.9984 - val_loss: 0.0445 -
val_accuracy: 0.9880 - 4s/epoch - 8ms/step
Epoch 17/20
422/422 - 3s - loss: 0.0066 - accuracy: 0.9991 - val_loss: 0.0462 -
val_accuracy: 0.9878 - 3s/epoch - 8ms/step
Epoch 18/20
422/422 - 3s - loss: 0.0062 - accuracy: 0.9989 - val_loss: 0.0482 -
val_accuracy: 0.9880 - 3s/epoch - 8ms/step
Epoch 19/20
```

```
422/422 - 3s - loss: 0.0051 - accuracy: 0.9993 - val_loss: 0.0501 -  
val_accuracy: 0.9880 - 3s/epoch - 8ms/step
```

```
Epoch 20/20
```

```
422/422 - 3s - loss: 0.0041 - accuracy: 0.9996 - val_loss: 0.0502 -  
val_accuracy: 0.9873 - 3s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

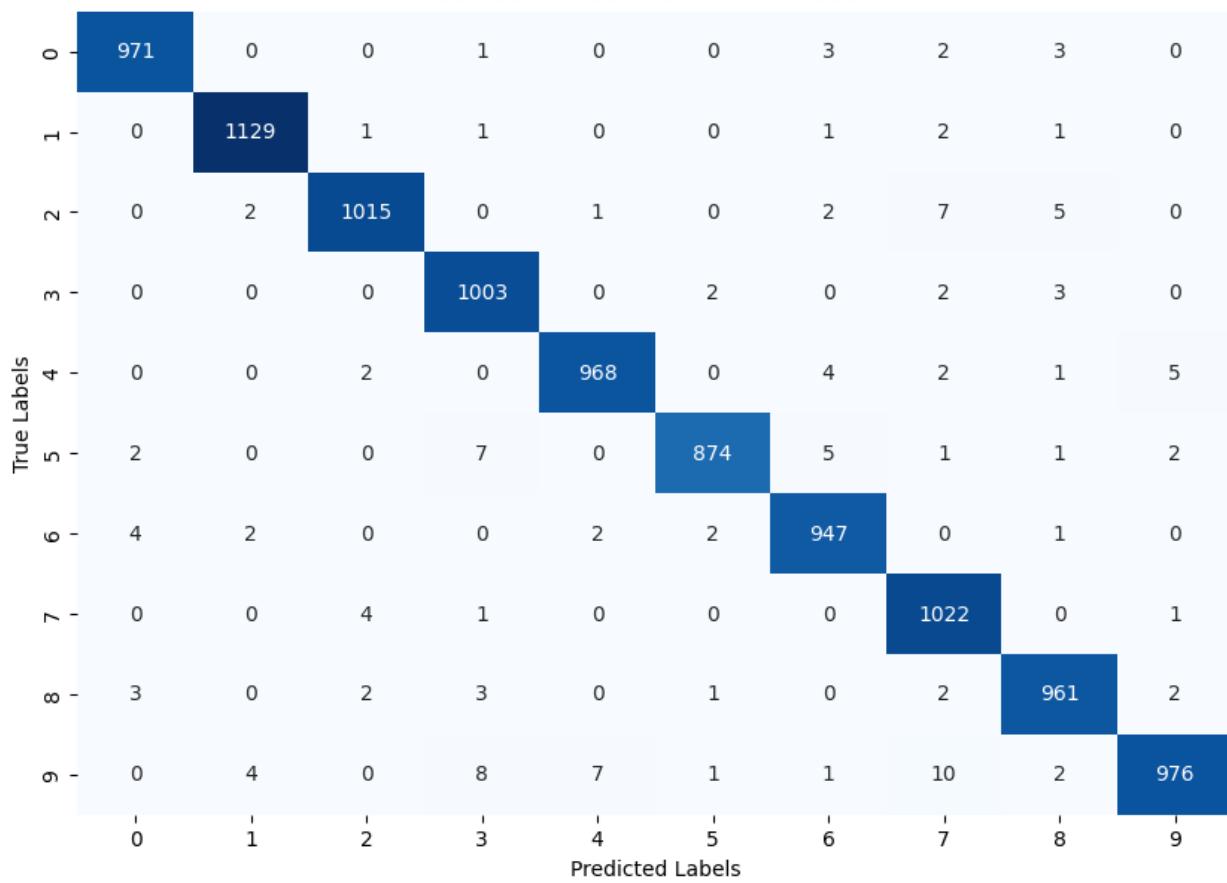
```
Confusion Matrix:
```

```
[[ 971   0   0   1   0   0   3   2   3   0]
 [ 0 1129   1   1   0   0   1   2   1   0]
 [ 0   2 1015   0   1   0   2   7   5   0]
 [ 0   0   0 1003   0   2   0   2   3   0]
 [ 0   0   2   0 968   0   4   2   1   5]
 [ 2   0   0   7   0 874   5   1   1   2]
 [ 4   2   0   0   2   2 947   0   1   0]
 [ 0   0   4   1   0   0   0 1022   0   1]
 [ 3   0   2   3   0   1   0   2 961   2]
 [ 0   4   0   8   7   1   1 10   2 976]]
```

```
Precision: 0.9867
```

```
Recall: 0.9866
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 1 conv_layers, 1 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 3s - loss: 0.3088 - accuracy: 0.9067 - val_loss: 0.1291 -
val_accuracy: 0.9672 - 3s/epoch - 16ms/step
Epoch 2/5
211/211 - 3s - loss: 0.1335 - accuracy: 0.9614 - val_loss: 0.0919 -
val_accuracy: 0.9763 - 3s/epoch - 13ms/step
Epoch 3/5
211/211 - 3s - loss: 0.0961 - accuracy: 0.9726 - val_loss: 0.0794 -
val_accuracy: 0.9785 - 3s/epoch - 12ms/step
Epoch 4/5
211/211 - 3s - loss: 0.0756 - accuracy: 0.9790 - val_loss: 0.0694 -
val_accuracy: 0.9817 - 3s/epoch - 13ms/step
Epoch 5/5
211/211 - 3s - loss: 0.0635 - accuracy: 0.9821 - val_loss: 0.0635 -
val_accuracy: 0.9817 - 3s/epoch - 13ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 973    0    0    0    0    1    1    1    3    1]
 [  0 1126    2    1    0    1    2    0    3    0]
 [  6    3 1004    3    2    0    1    6    6    1]
 [  0    0    0 1001    0    2    0    2    1    4]
 [  1    0    3    0  959    0    3    0    2   14]
 [  4    0    0   10    0  868    5    0    2    3]
 [  9    2    0    1    3    5  936    0    2    0]
 [  1    4    7    6    1    0    0  993    3   13]
 [  8    0    1   12    3    3    1    2  939    5]
 [  6    4    1    6    5    1    0    1    0  985]]
```

Precision: 0.9786
Recall: 0.9784

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9	
0	973	0	0	0	0	1	1	1	3	1	
1	0	1126	2	1	0	1	2	0	3	0	
2	6	3	1004	3	2	0	1	6	6	1	
3	0	0	0	1001	0	2	0	2	1	4	
4	1	0	3	0	959	0	3	0	2	14	
5	4	0	0	10	0	868	5	0	2	3	
6	9	2	0	1	3	5	936	0	2	0	
7	1	4	7	6	1	0	0	993	3	13	
8	8	0	1	12	3	3	1	2	939	5	
9	6	4	1	6	5	1	0	1	0	985	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

Training Model with relu activation, 1 conv_layers, 1 dense layers, 256 batch size, 15 epochs..

Epoch 1/15

211/211 - 3s - loss: 0.3247 - accuracy: 0.9018 - val_loss: 0.1382 - val_accuracy: 0.9617 - 3s/epoch - 15ms/step

Epoch 2/15

211/211 - 3s - loss: 0.1446 - accuracy: 0.9574 - val_loss: 0.0985 - val_accuracy: 0.9728 - 3s/epoch - 13ms/step

Epoch 3/15

211/211 - 3s - loss: 0.1037 - accuracy: 0.9694 - val_loss: 0.0841 - val_accuracy: 0.9773 - 3s/epoch - 13ms/step

Epoch 4/15

211/211 - 3s - loss: 0.0793 - accuracy: 0.9771 - val_loss: 0.0727 - val_accuracy: 0.9795 - 3s/epoch - 13ms/step

Epoch 5/15

211/211 - 3s - loss: 0.0677 - accuracy: 0.9799 - val_loss: 0.0653 - val_accuracy: 0.9833 - 3s/epoch - 13ms/step

Epoch 6/15

211/211 - 3s - loss: 0.0568 - accuracy: 0.9832 - val_loss: 0.0717 - val_accuracy: 0.9798 - 3s/epoch - 13ms/step

Epoch 7/15

```
211/211 - 3s - loss: 0.0500 - accuracy: 0.9856 - val_loss: 0.0584 -  
val_accuracy: 0.9845 - 3s/epoch - 13ms/step  
Epoch 8/15  
211/211 - 3s - loss: 0.0426 - accuracy: 0.9886 - val_loss: 0.0587 -  
val_accuracy: 0.9848 - 3s/epoch - 13ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.0376 - accuracy: 0.9896 - val_loss: 0.0523 -  
val_accuracy: 0.9858 - 3s/epoch - 13ms/step  
Epoch 10/15  
211/211 - 3s - loss: 0.0327 - accuracy: 0.9910 - val_loss: 0.0552 -  
val_accuracy: 0.9870 - 3s/epoch - 13ms/step  
Epoch 11/15  
211/211 - 3s - loss: 0.0298 - accuracy: 0.9914 - val_loss: 0.0518 -  
val_accuracy: 0.9862 - 3s/epoch - 13ms/step  
Epoch 12/15  
211/211 - 3s - loss: 0.0272 - accuracy: 0.9926 - val_loss: 0.0508 -  
val_accuracy: 0.9867 - 3s/epoch - 13ms/step  
Epoch 13/15  
211/211 - 3s - loss: 0.0235 - accuracy: 0.9942 - val_loss: 0.0496 -  
val_accuracy: 0.9867 - 3s/epoch - 13ms/step  
Epoch 14/15  
211/211 - 3s - loss: 0.0217 - accuracy: 0.9947 - val_loss: 0.0509 -  
val_accuracy: 0.9875 - 3s/epoch - 13ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.0202 - accuracy: 0.9948 - val_loss: 0.0495 -  
val_accuracy: 0.9865 - 3s/epoch - 13ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 969  1  0  0  1  1  3  2  2  1]  
[  0 1128  2  1  0  0  1  1  2  0]  
[  3   1 1016  2  1  0  0  4  5  0]  
[  0   0   1 998  0  2  0  2  4  3]  
[  0   1   2  0 973  0  1  0  1  4]  
[  2   0   0   7  0 874  5  0  2  2]  
[  4   3   1   1  2   2 943  0  2  0]  
[  0   3   8   4  0   0   0 1005  3  5]  
[  1   0   1   3  1   2   0   1 963  2]  
[  1   4   0   6 10   2   0   4   4 978]]  
Precision: 0.9847  
Recall: 0.9847
```

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	969	1	0	0	1	1	3	2	2	1
0	0	1128	2	1	0	0	1	1	2	0
1	3	1	1016	2	1	0	0	4	5	0
2	0	0	1	998	0	2	0	2	4	3
3	0	1	2	0	973	0	1	0	1	4
4	2	0	0	7	0	874	5	0	2	2
5	4	3	1	1	2	2	943	0	2	0
6	0	3	8	4	0	0	0	1005	3	5
7	1	0	1	3	1	2	0	1	963	2
8	1	4	0	6	10	2	0	4	4	978
9	1	2	3	4	5	6	7	8	9	9
Predicted Labels										

Training Model with relu activation, 1 conv_layers, 1 dense layers,
 256 batch size, 20 epochs..

Epoch 1/20
 211/211 - 3s - loss: 0.3529 - accuracy: 0.8913 - val_loss: 0.1450 -
 val_accuracy: 0.9603 - 3s/epoch - 15ms/step

Epoch 2/20
 211/211 - 3s - loss: 0.1492 - accuracy: 0.9561 - val_loss: 0.0969 -
 val_accuracy: 0.9730 - 3s/epoch - 13ms/step

Epoch 3/20
 211/211 - 3s - loss: 0.1082 - accuracy: 0.9684 - val_loss: 0.1019 -
 val_accuracy: 0.9712 - 3s/epoch - 13ms/step

Epoch 4/20
 211/211 - 3s - loss: 0.0882 - accuracy: 0.9743 - val_loss: 0.0760 -
 val_accuracy: 0.9790 - 3s/epoch - 13ms/step

Epoch 5/20
 211/211 - 3s - loss: 0.0719 - accuracy: 0.9789 - val_loss: 0.0673 -
 val_accuracy: 0.9815 - 3s/epoch - 13ms/step

Epoch 6/20
 211/211 - 3s - loss: 0.0634 - accuracy: 0.9812 - val_loss: 0.0661 -
 val_accuracy: 0.9832 - 3s/epoch - 13ms/step

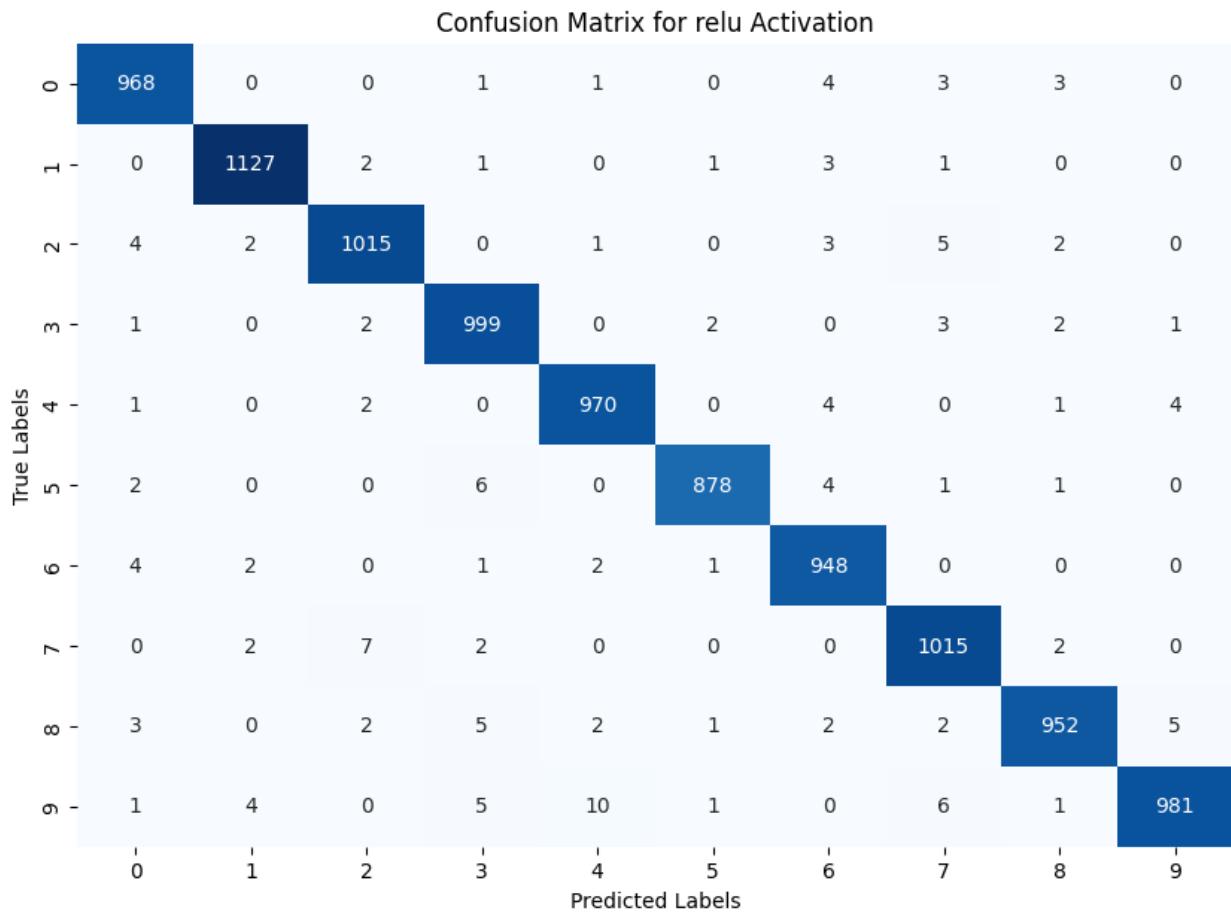
Epoch 7/20

```
211/211 - 3s - loss: 0.0554 - accuracy: 0.9830 - val_loss: 0.0616 -  
val_accuracy: 0.9837 - 3s/epoch - 13ms/step  
Epoch 8/20  
211/211 - 3s - loss: 0.0467 - accuracy: 0.9868 - val_loss: 0.0543 -  
val_accuracy: 0.9842 - 3s/epoch - 13ms/step  
Epoch 9/20  
211/211 - 3s - loss: 0.0428 - accuracy: 0.9877 - val_loss: 0.0532 -  
val_accuracy: 0.9855 - 3s/epoch - 13ms/step  
Epoch 10/20  
211/211 - 3s - loss: 0.0379 - accuracy: 0.9891 - val_loss: 0.0516 -  
val_accuracy: 0.9858 - 3s/epoch - 13ms/step  
Epoch 11/20  
211/211 - 3s - loss: 0.0335 - accuracy: 0.9907 - val_loss: 0.0548 -  
val_accuracy: 0.9853 - 3s/epoch - 12ms/step  
Epoch 12/20  
211/211 - 3s - loss: 0.0308 - accuracy: 0.9914 - val_loss: 0.0489 -  
val_accuracy: 0.9863 - 3s/epoch - 13ms/step  
Epoch 13/20  
211/211 - 3s - loss: 0.0270 - accuracy: 0.9926 - val_loss: 0.0490 -  
val_accuracy: 0.9862 - 3s/epoch - 13ms/step  
Epoch 14/20  
211/211 - 3s - loss: 0.0256 - accuracy: 0.9928 - val_loss: 0.0485 -  
val_accuracy: 0.9877 - 3s/epoch - 13ms/step  
Epoch 15/20  
211/211 - 3s - loss: 0.0226 - accuracy: 0.9938 - val_loss: 0.0454 -  
val_accuracy: 0.9880 - 3s/epoch - 13ms/step  
Epoch 16/20  
211/211 - 3s - loss: 0.0195 - accuracy: 0.9953 - val_loss: 0.0459 -  
val_accuracy: 0.9865 - 3s/epoch - 13ms/step  
Epoch 17/20  
211/211 - 3s - loss: 0.0188 - accuracy: 0.9953 - val_loss: 0.0462 -  
val_accuracy: 0.9870 - 3s/epoch - 13ms/step  
Epoch 18/20  
211/211 - 3s - loss: 0.0164 - accuracy: 0.9961 - val_loss: 0.0459 -  
val_accuracy: 0.9882 - 3s/epoch - 13ms/step  
Epoch 19/20  
211/211 - 3s - loss: 0.0148 - accuracy: 0.9965 - val_loss: 0.0443 -  
val_accuracy: 0.9860 - 3s/epoch - 13ms/step  
Epoch 20/20  
211/211 - 3s - loss: 0.0144 - accuracy: 0.9968 - val_loss: 0.0458 -  
val_accuracy: 0.9885 - 3s/epoch - 13ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 968 0 0 1 1 0 4 3 3 0]  
 [ 0 1127 2 1 0 1 3 1 0 0]  
 [ 4 2 1015 0 1 0 3 5 2 0]  
 [ 1 0 2 999 0 2 0 3 2 1]  
 [ 1 0 2 0 970 0 4 0 1 4]]
```

```
[ 2 0 0 6 0 878 4 1 1 0]
[ 4 2 0 1 2 1 948 0 0 0]
[ 0 2 7 2 0 0 0 1015 2 0]
[ 3 0 2 5 2 1 2 2 952 5]
[ 1 4 0 5 10 1 0 6 1 981]]
```

Precision: 0.9853

Recall: 0.9853



Training Model with relu activation, 2 conv_layers, 2 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 7s - loss: 0.1909 - accuracy: 0.9404 - val_loss: 0.0651 - val_accuracy: 0.9800 - 7s/epoch - 8ms/step

Epoch 2/5

844/844 - 6s - loss: 0.0676 - accuracy: 0.9789 - val_loss: 0.0638 - val_accuracy: 0.9818 - 6s/epoch - 7ms/step

Epoch 3/5

844/844 - 6s - loss: 0.0497 - accuracy: 0.9840 - val_loss: 0.0617 - val_accuracy: 0.9818 - 6s/epoch - 7ms/step

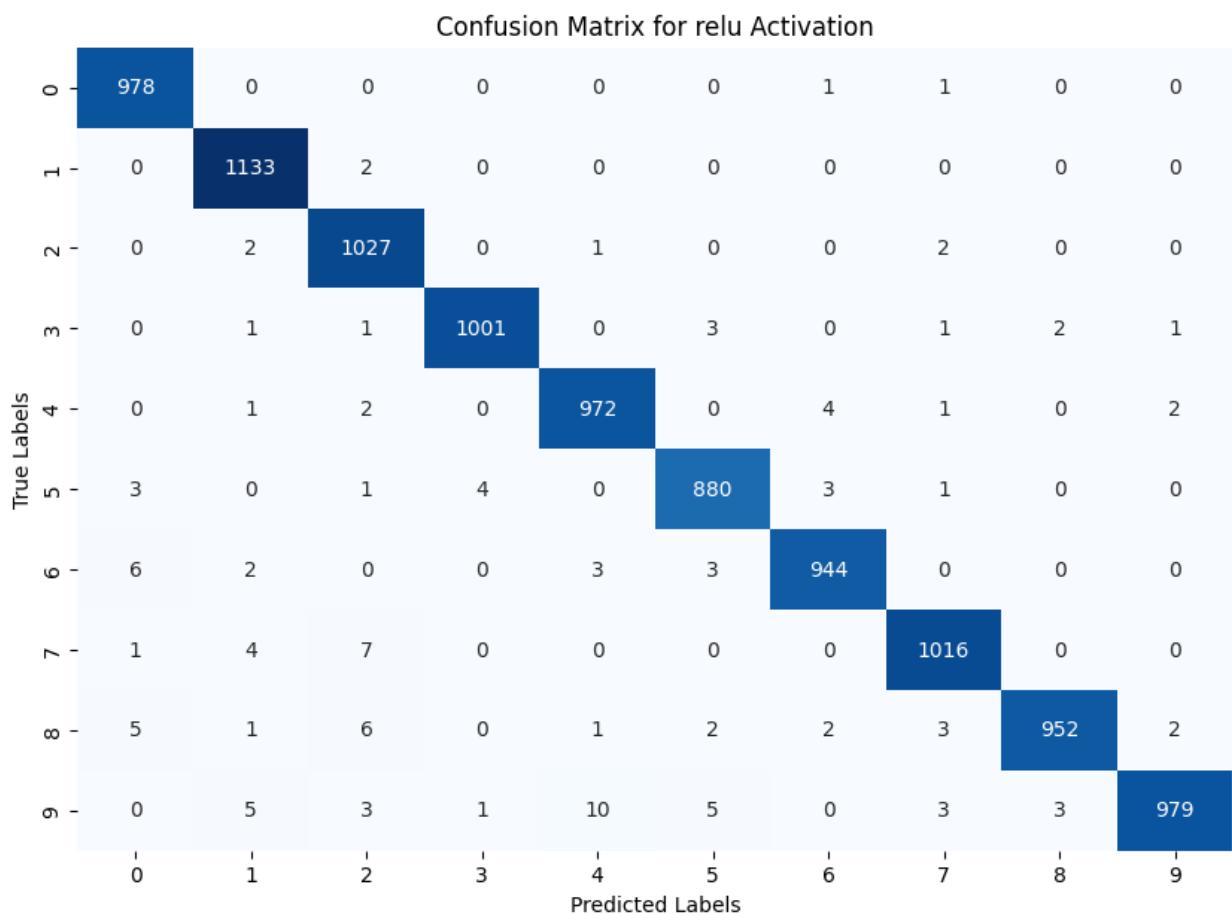
Epoch 4/5

844/844 - 6s - loss: 0.0377 - accuracy: 0.9879 - val_loss: 0.0533 -

```

val_accuracy: 0.9855 - 6s/epoch - 7ms/step
Epoch 5/5
844/844 - 6s - loss: 0.0314 - accuracy: 0.9902 - val_loss: 0.0459 -
val_accuracy: 0.9872 - 6s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 978   0   0   0   0   0   1   1   0   0]
 [ 0 1133   2   0   0   0   0   0   0   0]
 [ 0   2 1027   0   1   0   0   2   0   0]
 [ 0   1   1 1001   0   3   0   1   2   1]
 [ 0   1   2   0 972   0   4   1   0   2]
 [ 3   0   1   4   0 880   3   1   0   0]
 [ 6   2   0   0   3   3 944   0   0   0]
 [ 1   4   7   0   0   0   0 1016   0   0]
 [ 5   1   6   0   1   2   2   3 952   2]
 [ 0   5   3   1 10   5   0   3   3 979]]
Precision: 0.9883
Recall: 0.9882

```



Training Model with relu activation, 2 conv_layers, 2 dense layers, 64 batch size, 15 epochs..

Epoch 1/15
844/844 - 7s - loss: 0.1902 - accuracy: 0.9402 - val_loss: 0.0744 - val_accuracy: 0.9792 - 7s/epoch - 8ms/step

Epoch 2/15
844/844 - 6s - loss: 0.0668 - accuracy: 0.9797 - val_loss: 0.0595 - val_accuracy: 0.9818 - 6s/epoch - 7ms/step

Epoch 3/15
844/844 - 6s - loss: 0.0481 - accuracy: 0.9848 - val_loss: 0.0649 - val_accuracy: 0.9815 - 6s/epoch - 7ms/step

Epoch 4/15
844/844 - 6s - loss: 0.0364 - accuracy: 0.9881 - val_loss: 0.0447 - val_accuracy: 0.9872 - 6s/epoch - 7ms/step

Epoch 5/15
844/844 - 6s - loss: 0.0294 - accuracy: 0.9911 - val_loss: 0.0383 - val_accuracy: 0.9895 - 6s/epoch - 7ms/step

Epoch 6/15
844/844 - 6s - loss: 0.0233 - accuracy: 0.9927 - val_loss: 0.0365 - val_accuracy: 0.9902 - 6s/epoch - 7ms/step

Epoch 7/15
844/844 - 6s - loss: 0.0193 - accuracy: 0.9940 - val_loss: 0.0415 - val_accuracy: 0.9907 - 6s/epoch - 7ms/step

Epoch 8/15
844/844 - 6s - loss: 0.0151 - accuracy: 0.9953 - val_loss: 0.0430 - val_accuracy: 0.9890 - 6s/epoch - 7ms/step

Epoch 9/15
844/844 - 6s - loss: 0.0122 - accuracy: 0.9960 - val_loss: 0.0468 - val_accuracy: 0.9887 - 6s/epoch - 7ms/step

Epoch 10/15
844/844 - 6s - loss: 0.0098 - accuracy: 0.9971 - val_loss: 0.0421 - val_accuracy: 0.9897 - 6s/epoch - 7ms/step

Epoch 11/15
844/844 - 6s - loss: 0.0092 - accuracy: 0.9967 - val_loss: 0.0474 - val_accuracy: 0.9898 - 6s/epoch - 7ms/step

Epoch 12/15
844/844 - 6s - loss: 0.0080 - accuracy: 0.9973 - val_loss: 0.0463 - val_accuracy: 0.9898 - 6s/epoch - 7ms/step

Epoch 13/15
844/844 - 6s - loss: 0.0066 - accuracy: 0.9978 - val_loss: 0.0448 - val_accuracy: 0.9908 - 6s/epoch - 7ms/step

Epoch 14/15
844/844 - 6s - loss: 0.0046 - accuracy: 0.9987 - val_loss: 0.0462 - val_accuracy: 0.9895 - 6s/epoch - 7ms/step

Epoch 15/15
844/844 - 6s - loss: 0.0046 - accuracy: 0.9984 - val_loss: 0.0554 - val_accuracy: 0.9878 - 6s/epoch - 7ms/step

313/313 [=====] - 1s 2ms/step

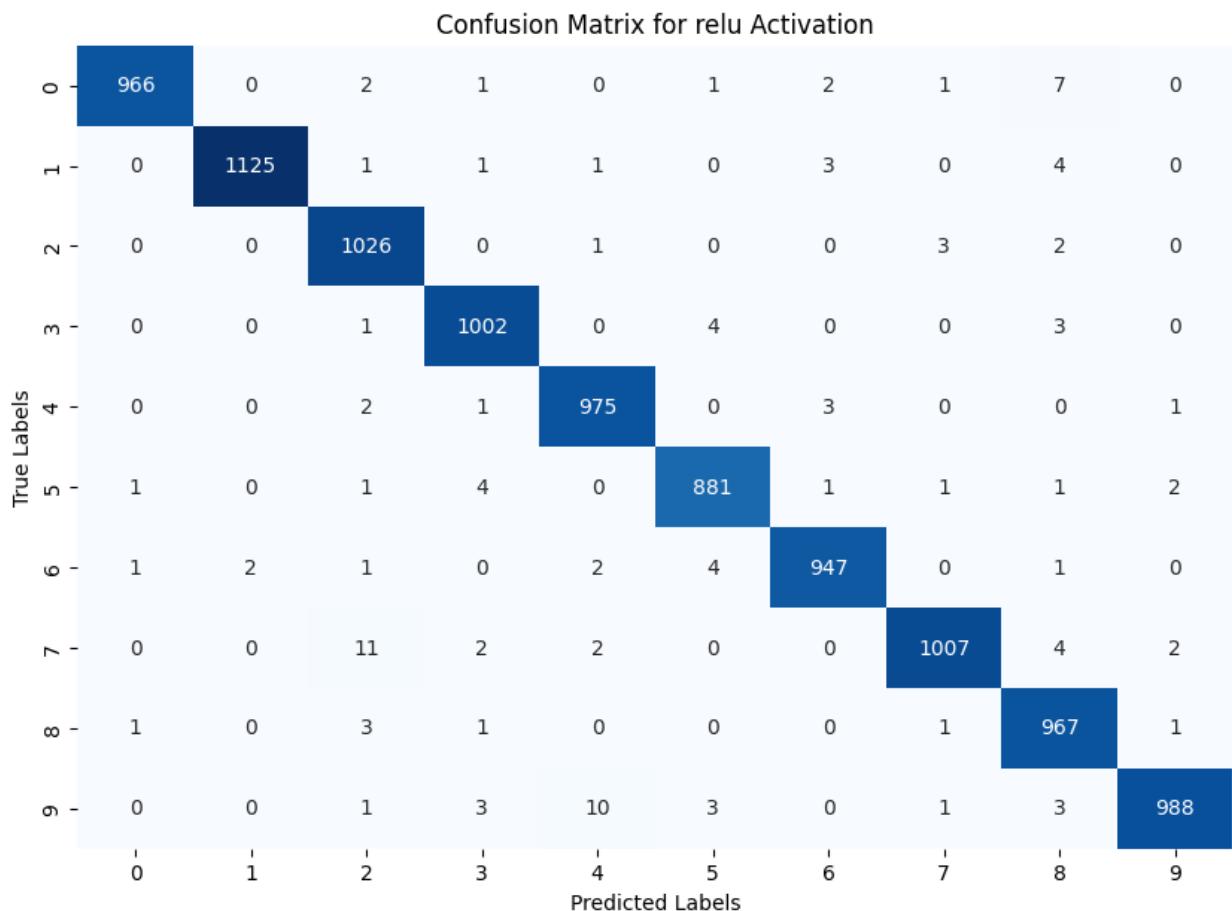
Results for activation function: relu

Confusion Matrix:

```
[[ 966  0   2   1   0   1   2   1   7   0]
 [ 0 1125  1   1   1   0   3   0   4   0]
 [ 0   0 1026  0   1   0   0   3   2   0]
 [ 0   0   1 1002  0   4   0   0   3   0]
 [ 0   0   2   1 975  0   3   0   0   1]
 [ 1   0   1   4   0 881  1   1   1   2]
 [ 1   2   1   0   2   4 947  0   1   0]
 [ 0   0   11  2   2   0   0 1007  4   2]
 [ 1   0   3   1   0   0   0   1 967  1]
 [ 0   0   1   3   10  3   0   1   3 988]]
```

Precision: 0.9885

Recall: 0.9884



Training Model with relu activation, 2 conv_layers, 2 dense layers, 64 batch size, 20 epochs..

Epoch 1/20

844/844 - 7s - loss: 0.2038 - accuracy: 0.9358 - val_loss: 0.0731 - val_accuracy: 0.9792 - 7s/epoch - 8ms/step

Epoch 2/20

844/844 - 6s - loss: 0.0637 - accuracy: 0.9800 - val_loss: 0.0549 - val_accuracy: 0.9852 - 6s/epoch - 7ms/step

```
Epoch 3/20
844/844 - 6s - loss: 0.0442 - accuracy: 0.9856 - val_loss: 0.0570 -
val_accuracy: 0.9853 - 6s/epoch - 7ms/step
Epoch 4/20
844/844 - 6s - loss: 0.0347 - accuracy: 0.9887 - val_loss: 0.0498 -
val_accuracy: 0.9865 - 6s/epoch - 7ms/step
Epoch 5/20
844/844 - 6s - loss: 0.0279 - accuracy: 0.9911 - val_loss: 0.0458 -
val_accuracy: 0.9885 - 6s/epoch - 7ms/step
Epoch 6/20
844/844 - 6s - loss: 0.0236 - accuracy: 0.9922 - val_loss: 0.0635 -
val_accuracy: 0.9828 - 6s/epoch - 7ms/step
Epoch 7/20
844/844 - 6s - loss: 0.0178 - accuracy: 0.9941 - val_loss: 0.0440 -
val_accuracy: 0.9890 - 6s/epoch - 7ms/step
Epoch 8/20
844/844 - 6s - loss: 0.0159 - accuracy: 0.9948 - val_loss: 0.0442 -
val_accuracy: 0.9880 - 6s/epoch - 7ms/step
Epoch 9/20
844/844 - 6s - loss: 0.0145 - accuracy: 0.9951 - val_loss: 0.0518 -
val_accuracy: 0.9865 - 6s/epoch - 7ms/step
Epoch 10/20
844/844 - 6s - loss: 0.0112 - accuracy: 0.9966 - val_loss: 0.0558 -
val_accuracy: 0.9877 - 6s/epoch - 7ms/step
Epoch 11/20
844/844 - 6s - loss: 0.0099 - accuracy: 0.9967 - val_loss: 0.0485 -
val_accuracy: 0.9890 - 6s/epoch - 7ms/step
Epoch 12/20
844/844 - 6s - loss: 0.0078 - accuracy: 0.9973 - val_loss: 0.0494 -
val_accuracy: 0.9890 - 6s/epoch - 7ms/step
Epoch 13/20
844/844 - 6s - loss: 0.0064 - accuracy: 0.9979 - val_loss: 0.0512 -
val_accuracy: 0.9888 - 6s/epoch - 7ms/step
Epoch 14/20
844/844 - 6s - loss: 0.0050 - accuracy: 0.9984 - val_loss: 0.0566 -
val_accuracy: 0.9887 - 6s/epoch - 7ms/step
Epoch 15/20
844/844 - 6s - loss: 0.0031 - accuracy: 0.9992 - val_loss: 0.0560 -
val_accuracy: 0.9897 - 6s/epoch - 7ms/step
Epoch 16/20
844/844 - 6s - loss: 0.0022 - accuracy: 0.9993 - val_loss: 0.0541 -
val_accuracy: 0.9890 - 6s/epoch - 7ms/step
Epoch 17/20
844/844 - 6s - loss: 0.0015 - accuracy: 0.9997 - val_loss: 0.0555 -
val_accuracy: 0.9905 - 6s/epoch - 7ms/step
Epoch 18/20
844/844 - 6s - loss: 0.0013 - accuracy: 0.9998 - val_loss: 0.0573 -
val_accuracy: 0.9895 - 6s/epoch - 7ms/step
Epoch 19/20
```

```
844/844 - 6s - loss: 0.0012 - accuracy: 0.9998 - val_loss: 0.0685 -  
val_accuracy: 0.9885 - 6s/epoch - 7ms/step
```

```
Epoch 20/20
```

```
844/844 - 6s - loss: 0.0022 - accuracy: 0.9995 - val_loss: 0.0601 -  
val_accuracy: 0.9897 - 6s/epoch - 7ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

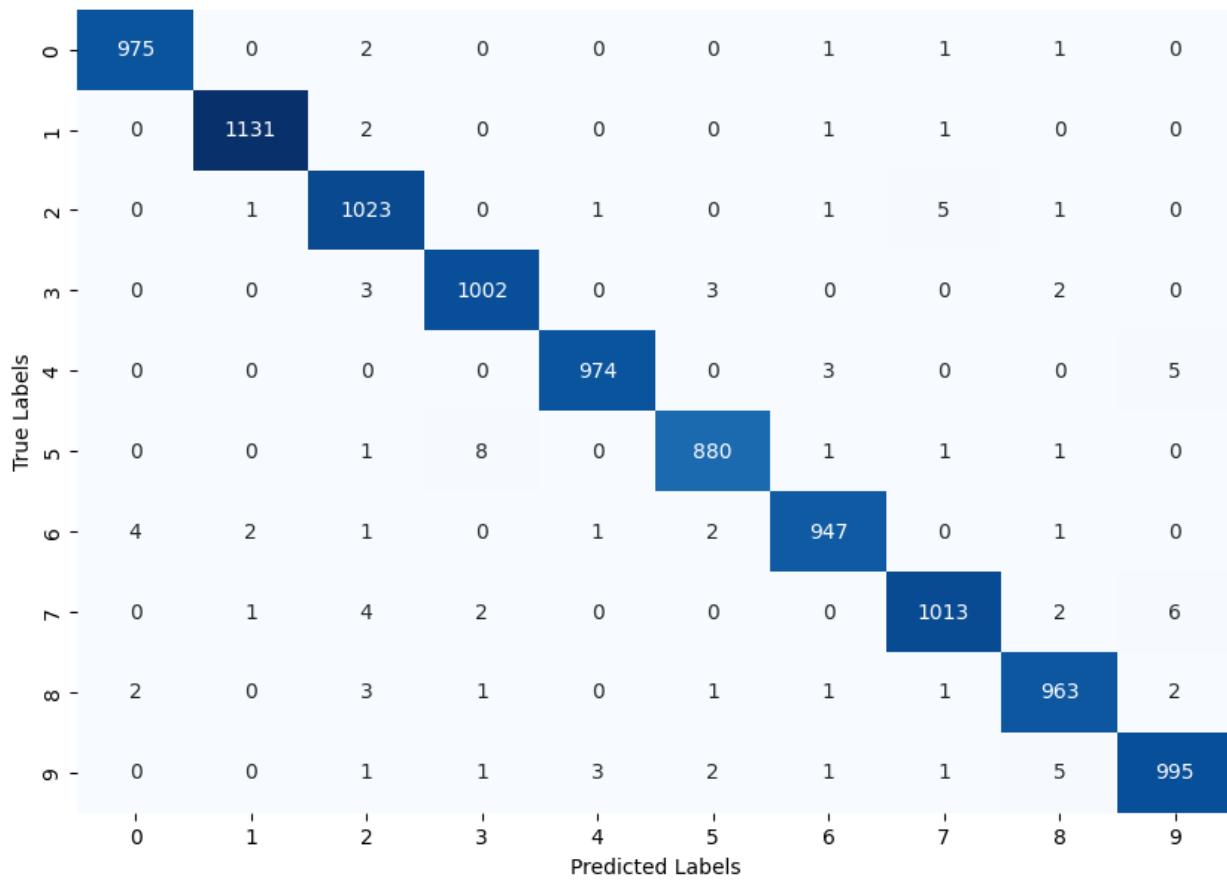
```
Confusion Matrix:
```

```
[[ 975   0   2   0   0   0   1   1   1   0]
 [ 0 1131   2   0   0   0   1   1   0   0]
 [ 0   1 1023   0   1   0   1   5   1   0]
 [ 0   0   3 1002   0   3   0   0   2   0]
 [ 0   0   0   0  974   0   3   0   0   5]
 [ 0   0   1   8   0  880   1   1   1   0]
 [ 4   2   1   0   1   2  947   0   1   0]
 [ 0   1   4   2   0   0   0 1013   2   6]
 [ 2   0   3   1   0   1   1   1   963   2]
 [ 0   0   1   1   3   2   1   1   5  995]]
```

```
Precision: 0.9903
```

```
Recall: 0.9903
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 2 conv_layers, 2 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 5s - loss: 0.2797 - accuracy: 0.9137 - val_loss: 0.0847 -
val_accuracy: 0.9745 - 5s/epoch - 11ms/step
Epoch 2/5
422/422 - 4s - loss: 0.0825 - accuracy: 0.9746 - val_loss: 0.0691 -
val_accuracy: 0.9798 - 4s/epoch - 10ms/step
Epoch 3/5
422/422 - 4s - loss: 0.0614 - accuracy: 0.9808 - val_loss: 0.0522 -
val_accuracy: 0.9847 - 4s/epoch - 10ms/step
Epoch 4/5
422/422 - 4s - loss: 0.0495 - accuracy: 0.9847 - val_loss: 0.0559 -
val_accuracy: 0.9847 - 4s/epoch - 10ms/step
Epoch 5/5
422/422 - 4s - loss: 0.0420 - accuracy: 0.9867 - val_loss: 0.0455 -
val_accuracy: 0.9860 - 4s/epoch - 10ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 967    0    3    0    2    0    4    2    2    0]
 [  0 1132    1    1    0    0    0    1    0    0]
 [  1    0 1025    0    2    0    0    3    1    0]
 [  0    0    2 998    0    5    0    3    2    0]
 [  0    0    0    0 978    0    0    1    0    3]
 [  1    0    0    5    0 881    3    1    1    0]
 [  3    2    0    0    4    2 946    0    1    0]
 [  0    2    5    1    0    0    0 1014    1    5]
 [  4    0    3    1    1    1    0    2 961    1]
 [  2    1    1    3    4    5    0    5    0 988]]
```

Precision: 0.9890
Recall: 0.9890

Confusion Matrix for relu Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	967	0	3	0	2	0	4	2	2	0	
0	967	0	3	0	2	0	4	2	2	0	
1	0	1132	1	1	0	0	0	1	0	0	
2	1	0	1025	0	2	0	0	3	1	0	
3	0	0	2	998	0	5	0	3	2	0	
4	0	0	0	0	978	0	0	1	0	3	
5	1	0	0	5	0	881	3	1	1	0	
6	3	2	0	0	4	2	946	0	1	0	
7	0	2	5	1	0	0	0	1014	1	5	
8	4	0	3	1	1	1	0	2	961	1	
9	2	1	1	3	4	5	0	5	0	988	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```

Training Model with relu activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 5s - loss: 0.3019 - accuracy: 0.9034 - val_loss: 0.0895 -
val_accuracy: 0.9738 - 5s/epoch - 11ms/step
Epoch 2/15
422/422 - 4s - loss: 0.0831 - accuracy: 0.9743 - val_loss: 0.0586 -
val_accuracy: 0.9843 - 4s/epoch - 10ms/step
Epoch 3/15
422/422 - 4s - loss: 0.0593 - accuracy: 0.9816 - val_loss: 0.0489 -
val_accuracy: 0.9870 - 4s/epoch - 10ms/step
Epoch 4/15
422/422 - 4s - loss: 0.0481 - accuracy: 0.9846 - val_loss: 0.0476 -
val_accuracy: 0.9883 - 4s/epoch - 10ms/step
Epoch 5/15
422/422 - 4s - loss: 0.0391 - accuracy: 0.9878 - val_loss: 0.0620 -
val_accuracy: 0.9830 - 4s/epoch - 10ms/step
Epoch 6/15
422/422 - 4s - loss: 0.0348 - accuracy: 0.9892 - val_loss: 0.0491 -
val_accuracy: 0.9865 - 4s/epoch - 10ms/step
Epoch 7/15

```

```
422/422 - 4s - loss: 0.0300 - accuracy: 0.9906 - val_loss: 0.0462 -  
val_accuracy: 0.9873 - 4s/epoch - 10ms/step  
Epoch 8/15  
422/422 - 4s - loss: 0.0255 - accuracy: 0.9918 - val_loss: 0.0477 -  
val_accuracy: 0.9870 - 4s/epoch - 10ms/step  
Epoch 9/15  
422/422 - 4s - loss: 0.0216 - accuracy: 0.9932 - val_loss: 0.0429 -  
val_accuracy: 0.9890 - 4s/epoch - 10ms/step  
Epoch 10/15  
422/422 - 4s - loss: 0.0208 - accuracy: 0.9934 - val_loss: 0.0509 -  
val_accuracy: 0.9880 - 4s/epoch - 9ms/step  
Epoch 11/15  
422/422 - 4s - loss: 0.0183 - accuracy: 0.9942 - val_loss: 0.0392 -  
val_accuracy: 0.9910 - 4s/epoch - 10ms/step  
Epoch 12/15  
422/422 - 4s - loss: 0.0155 - accuracy: 0.9951 - val_loss: 0.0419 -  
val_accuracy: 0.9882 - 4s/epoch - 10ms/step  
Epoch 13/15  
422/422 - 4s - loss: 0.0140 - accuracy: 0.9956 - val_loss: 0.0466 -  
val_accuracy: 0.9892 - 4s/epoch - 10ms/step  
Epoch 14/15  
422/422 - 4s - loss: 0.0118 - accuracy: 0.9963 - val_loss: 0.0440 -  
val_accuracy: 0.9888 - 4s/epoch - 10ms/step  
Epoch 15/15  
422/422 - 4s - loss: 0.0096 - accuracy: 0.9972 - val_loss: 0.0432 -  
val_accuracy: 0.9897 - 4s/epoch - 10ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 976   0   1   0   0   0   1   1   1   0]  
[  0 1130   2   1   1   0   0   0   1   0]  
[  1   3 1021   1   2   0   0   4   0   0]  
[  0   0   3 999   0   6   0   0   2   0]  
[  0   0   0   0 976   0   0   1   1   4]  
[  1   0   0   4   0 885   1   1   0   0]  
[  4   2   0   0   1   4 946   0   1   0]  
[  0   1   3   0   0   0   0 1022   2   0]  
[  4   0   2   0   0   1   1   2 962   2]  
[  0   3   0   1   6   4   1   2   2 990]]  
Precision: 0.9907  
Recall: 0.9907
```

Confusion Matrix for relu Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	976	0	1	0	0	0	1	1	1	0	
0	976	0	1	0	0	0	0	0	0	0	
1	0	1130	2	1	1	0	0	0	1	0	
2	1	3	1021	1	2	0	0	4	0	0	
3	0	0	3	999	0	6	0	0	2	0	
4	0	0	0	0	976	0	0	1	1	4	
5	1	0	0	4	0	885	1	1	0	0	
6	4	2	0	0	1	4	946	0	1	0	
7	0	1	3	0	0	0	0	1022	2	0	
8	4	0	2	0	0	1	1	2	962	2	
9	0	3	0	1	6	4	1	2	2	990	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```

Training Model with relu activation, 2 conv_layers, 2 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 5s - loss: 0.2774 - accuracy: 0.9137 - val_loss: 0.0867 -
val_accuracy: 0.9752 - 5s/epoch - 11ms/step
Epoch 2/20
422/422 - 4s - loss: 0.0785 - accuracy: 0.9758 - val_loss: 0.0556 -
val_accuracy: 0.9838 - 4s/epoch - 9ms/step
Epoch 3/20
422/422 - 4s - loss: 0.0562 - accuracy: 0.9829 - val_loss: 0.0451 -
val_accuracy: 0.9867 - 4s/epoch - 10ms/step
Epoch 4/20
422/422 - 4s - loss: 0.0451 - accuracy: 0.9858 - val_loss: 0.0440 -
val_accuracy: 0.9882 - 4s/epoch - 9ms/step
Epoch 5/20
422/422 - 4s - loss: 0.0374 - accuracy: 0.9883 - val_loss: 0.0379 -
val_accuracy: 0.9883 - 4s/epoch - 9ms/step
Epoch 6/20
422/422 - 4s - loss: 0.0316 - accuracy: 0.9903 - val_loss: 0.0422 -
val_accuracy: 0.9882 - 4s/epoch - 10ms/step
Epoch 7/20

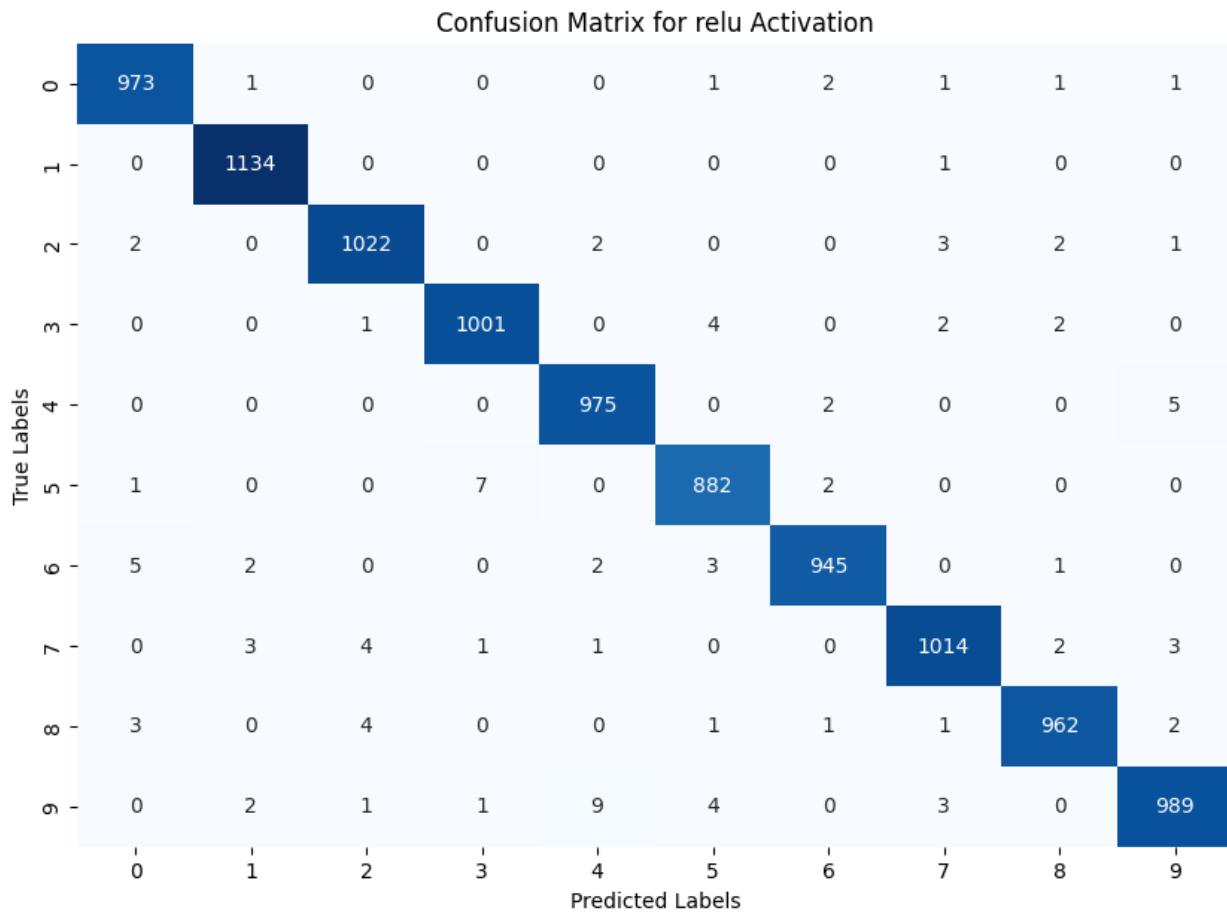
```

```
422/422 - 4s - loss: 0.0261 - accuracy: 0.9918 - val_loss: 0.0450 -  
val_accuracy: 0.9892 - 4s/epoch - 9ms/step  
Epoch 8/20  
422/422 - 4s - loss: 0.0232 - accuracy: 0.9926 - val_loss: 0.0373 -  
val_accuracy: 0.9890 - 4s/epoch - 9ms/step  
Epoch 9/20  
422/422 - 4s - loss: 0.0208 - accuracy: 0.9932 - val_loss: 0.0376 -  
val_accuracy: 0.9885 - 4s/epoch - 10ms/step  
Epoch 10/20  
422/422 - 4s - loss: 0.0170 - accuracy: 0.9946 - val_loss: 0.0416 -  
val_accuracy: 0.9885 - 4s/epoch - 9ms/step  
Epoch 11/20  
422/422 - 4s - loss: 0.0145 - accuracy: 0.9955 - val_loss: 0.0421 -  
val_accuracy: 0.9893 - 4s/epoch - 9ms/step  
Epoch 12/20  
422/422 - 4s - loss: 0.0128 - accuracy: 0.9960 - val_loss: 0.0375 -  
val_accuracy: 0.9892 - 4s/epoch - 9ms/step  
Epoch 13/20  
422/422 - 4s - loss: 0.0109 - accuracy: 0.9966 - val_loss: 0.0461 -  
val_accuracy: 0.9880 - 4s/epoch - 9ms/step  
Epoch 14/20  
422/422 - 4s - loss: 0.0103 - accuracy: 0.9969 - val_loss: 0.0412 -  
val_accuracy: 0.9897 - 4s/epoch - 9ms/step  
Epoch 15/20  
422/422 - 4s - loss: 0.0089 - accuracy: 0.9973 - val_loss: 0.0366 -  
val_accuracy: 0.9908 - 4s/epoch - 9ms/step  
Epoch 16/20  
422/422 - 4s - loss: 0.0079 - accuracy: 0.9977 - val_loss: 0.0473 -  
val_accuracy: 0.9887 - 4s/epoch - 10ms/step  
Epoch 17/20  
422/422 - 4s - loss: 0.0066 - accuracy: 0.9981 - val_loss: 0.0383 -  
val_accuracy: 0.9902 - 4s/epoch - 10ms/step  
Epoch 18/20  
422/422 - 4s - loss: 0.0053 - accuracy: 0.9985 - val_loss: 0.0369 -  
val_accuracy: 0.9903 - 4s/epoch - 10ms/step  
Epoch 19/20  
422/422 - 4s - loss: 0.0039 - accuracy: 0.9991 - val_loss: 0.0408 -  
val_accuracy: 0.9895 - 4s/epoch - 10ms/step  
Epoch 20/20  
422/422 - 4s - loss: 0.0033 - accuracy: 0.9993 - val_loss: 0.0395 -  
val_accuracy: 0.9910 - 4s/epoch - 10ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 973   1   0   0   0   1   2   1   1   1]  
 [  0 1134   0   0   0   0   0   1   0   0]  
 [  2   0 1022   0   2   0   0   3   2   1]  
 [  0   0   1 1001   0   4   0   2   2   0]  
 [  0   0   0   0  975   0   2   0   0   5]]
```

```
[ 1 0 0 7 0 882 2 0 0 0]
[ 5 2 0 0 2 3 945 0 1 0]
[ 0 3 4 1 1 0 0 1014 2 3]
[ 3 0 4 0 0 1 1 1 962 2]
[ 0 2 1 1 9 4 0 3 0 989]]
```

Precision: 0.9897

Recall: 0.9897



Training Model with relu activation, 2 conv_layers, 2 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 4s - loss: 0.3943 - accuracy: 0.8736 - val_loss: 0.1056 - val_accuracy: 0.9693 - 4s/epoch - 21ms/step

Epoch 2/5

211/211 - 3s - loss: 0.1060 - accuracy: 0.9672 - val_loss: 0.0636 - val_accuracy: 0.9823 - 3s/epoch - 15ms/step

Epoch 3/5

211/211 - 3s - loss: 0.0753 - accuracy: 0.9776 - val_loss: 0.0542 - val_accuracy: 0.9848 - 3s/epoch - 15ms/step

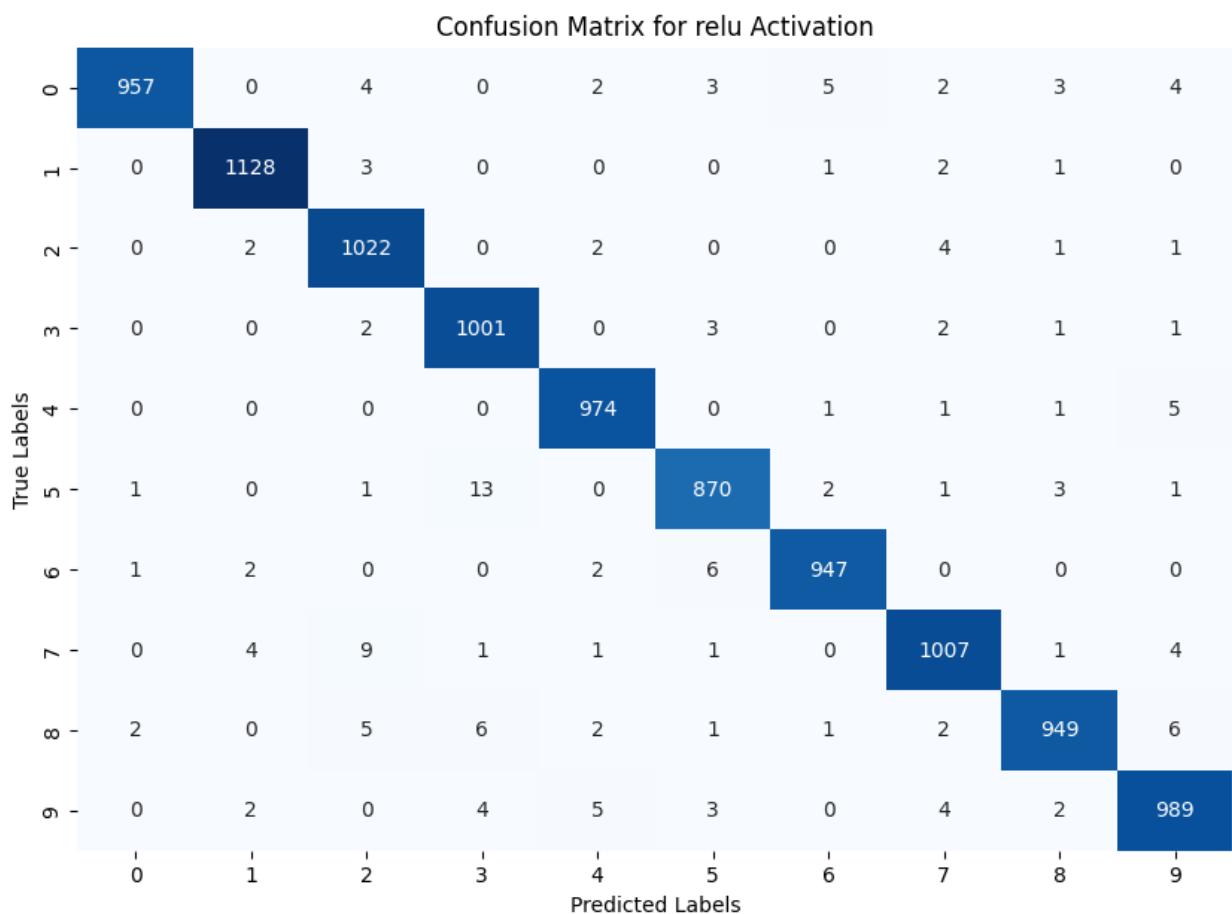
Epoch 4/5

211/211 - 3s - loss: 0.0612 - accuracy: 0.9812 - val_loss: 0.0507 -

```

val_accuracy: 0.9875 - 3s/epoch - 15ms/step
Epoch 5/5
211/211 - 3s - loss: 0.0517 - accuracy: 0.9841 - val_loss: 0.0497 -
val_accuracy: 0.9865 - 3s/epoch - 15ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 957   0   4   0   2   3   5   2   3   4]
 [  0 1128   3   0   0   0   1   2   1   0]
 [  0   2 1022   0   2   0   0   4   1   1]
 [  0   0   2 1001   0   3   0   2   1   1]
 [  0   0   0   0  974   0   1   1   1   5]
 [  1   0   1   13   0  870   2   1   3   1]
 [  1   2   0   0   2   6  947   0   0   0]
 [  0   4   9   1   1   1   0 1007   1   4]
 [  2   0   5   6   2   1   1   2  949   6]
 [  0   2   0   4   5   3   0   4   2  989]]
Precision: 0.9844
Recall: 0.9844

```

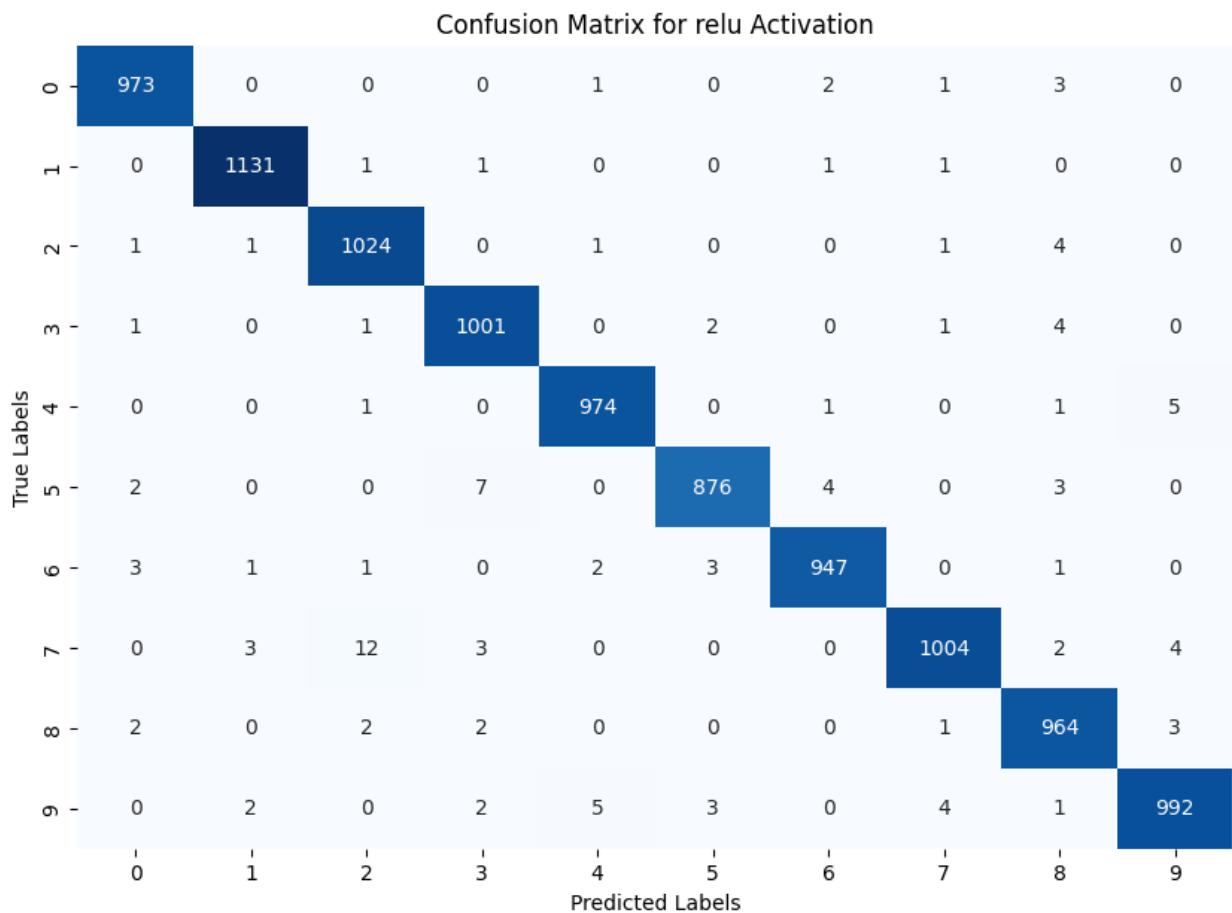


```
Training Model with relu activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 4s - loss: 0.4487 - accuracy: 0.8615 - val_loss: 0.1291 -
val_accuracy: 0.9577 - 4s/epoch - 19ms/step
Epoch 2/15
211/211 - 3s - loss: 0.1056 - accuracy: 0.9684 - val_loss: 0.0736 -
val_accuracy: 0.9785 - 3s/epoch - 15ms/step
Epoch 3/15
211/211 - 3s - loss: 0.0763 - accuracy: 0.9766 - val_loss: 0.0583 -
val_accuracy: 0.9828 - 3s/epoch - 15ms/step
Epoch 4/15
211/211 - 3s - loss: 0.0621 - accuracy: 0.9808 - val_loss: 0.0559 -
val_accuracy: 0.9837 - 3s/epoch - 15ms/step
Epoch 5/15
211/211 - 3s - loss: 0.0527 - accuracy: 0.9838 - val_loss: 0.0513 -
val_accuracy: 0.9845 - 3s/epoch - 15ms/step
Epoch 6/15
211/211 - 3s - loss: 0.0464 - accuracy: 0.9855 - val_loss: 0.0478 -
val_accuracy: 0.9865 - 3s/epoch - 15ms/step
Epoch 7/15
211/211 - 3s - loss: 0.0403 - accuracy: 0.9879 - val_loss: 0.0510 -
val_accuracy: 0.9867 - 3s/epoch - 15ms/step
Epoch 8/15
211/211 - 3s - loss: 0.0377 - accuracy: 0.9885 - val_loss: 0.0520 -
val_accuracy: 0.9848 - 3s/epoch - 15ms/step
Epoch 9/15
211/211 - 3s - loss: 0.0335 - accuracy: 0.9895 - val_loss: 0.0410 -
val_accuracy: 0.9892 - 3s/epoch - 15ms/step
Epoch 10/15
211/211 - 3s - loss: 0.0282 - accuracy: 0.9915 - val_loss: 0.0435 -
val_accuracy: 0.9882 - 3s/epoch - 15ms/step
Epoch 11/15
211/211 - 3s - loss: 0.0272 - accuracy: 0.9918 - val_loss: 0.0418 -
val_accuracy: 0.9877 - 3s/epoch - 15ms/step
Epoch 12/15
211/211 - 3s - loss: 0.0257 - accuracy: 0.9919 - val_loss: 0.0466 -
val_accuracy: 0.9880 - 3s/epoch - 15ms/step
Epoch 13/15
211/211 - 3s - loss: 0.0230 - accuracy: 0.9928 - val_loss: 0.0442 -
val_accuracy: 0.9890 - 3s/epoch - 15ms/step
Epoch 14/15
211/211 - 3s - loss: 0.0214 - accuracy: 0.9938 - val_loss: 0.0434 -
val_accuracy: 0.9885 - 3s/epoch - 15ms/step
Epoch 15/15
211/211 - 3s - loss: 0.0203 - accuracy: 0.9937 - val_loss: 0.0432 -
val_accuracy: 0.9903 - 3s/epoch - 15ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
```

```
[[ 973  0  0  0  1  0  2  1  3  0]
 [ 0 1131  1  1  0  0  1  1  0  0]
 [ 1  1 1024  0  1  0  0  1  4  0]
 [ 1  0  1 1001  0  2  0  1  4  0]
 [ 0  0  1  0  974  0  1  0  1  5]
 [ 2  0  0  7  0  876  4  0  3  0]
 [ 3  1  1  0  2  3  947  0  1  0]
 [ 0  3  12  3  0  0  0 1004  2  4]
 [ 2  0  2  2  0  0  0  1  964  3]
 [ 0  2  0  2  5  3  0  4  1  992]]
```

Precision: 0.9886

Recall: 0.9886



Training Model with relu activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..

Epoch 1/20

211/211 - 4s - loss: 0.3911 - accuracy: 0.8746 - val_loss: 0.1126 -
val_accuracy: 0.9675 - 4s/epoch - 19ms/step

Epoch 2/20

211/211 - 3s - loss: 0.1075 - accuracy: 0.9674 - val_loss: 0.0778 -
val_accuracy: 0.9762 - 3s/epoch - 15ms/step

```
Epoch 3/20
211/211 - 3s - loss: 0.0798 - accuracy: 0.9753 - val_loss: 0.0612 -
val_accuracy: 0.9822 - 3s/epoch - 15ms/step
Epoch 4/20
211/211 - 3s - loss: 0.0651 - accuracy: 0.9800 - val_loss: 0.0593 -
val_accuracy: 0.9828 - 3s/epoch - 15ms/step
Epoch 5/20
211/211 - 3s - loss: 0.0535 - accuracy: 0.9838 - val_loss: 0.0560 -
val_accuracy: 0.9850 - 3s/epoch - 16ms/step
Epoch 6/20
211/211 - 3s - loss: 0.0474 - accuracy: 0.9850 - val_loss: 0.0605 -
val_accuracy: 0.9828 - 3s/epoch - 15ms/step
Epoch 7/20
211/211 - 3s - loss: 0.0416 - accuracy: 0.9865 - val_loss: 0.0495 -
val_accuracy: 0.9858 - 3s/epoch - 15ms/step
Epoch 8/20
211/211 - 3s - loss: 0.0372 - accuracy: 0.9883 - val_loss: 0.0474 -
val_accuracy: 0.9875 - 3s/epoch - 15ms/step
Epoch 9/20
211/211 - 3s - loss: 0.0347 - accuracy: 0.9891 - val_loss: 0.0449 -
val_accuracy: 0.9875 - 3s/epoch - 15ms/step
Epoch 10/20
211/211 - 3s - loss: 0.0297 - accuracy: 0.9906 - val_loss: 0.0443 -
val_accuracy: 0.9880 - 3s/epoch - 15ms/step
Epoch 11/20
211/211 - 3s - loss: 0.0278 - accuracy: 0.9912 - val_loss: 0.0438 -
val_accuracy: 0.9882 - 3s/epoch - 15ms/step
Epoch 12/20
211/211 - 3s - loss: 0.0256 - accuracy: 0.9915 - val_loss: 0.0503 -
val_accuracy: 0.9853 - 3s/epoch - 15ms/step
Epoch 13/20
211/211 - 3s - loss: 0.0231 - accuracy: 0.9926 - val_loss: 0.0497 -
val_accuracy: 0.9862 - 3s/epoch - 15ms/step
Epoch 14/20
211/211 - 3s - loss: 0.0223 - accuracy: 0.9926 - val_loss: 0.0484 -
val_accuracy: 0.9877 - 3s/epoch - 15ms/step
Epoch 15/20
211/211 - 3s - loss: 0.0193 - accuracy: 0.9941 - val_loss: 0.0467 -
val_accuracy: 0.9893 - 3s/epoch - 15ms/step
Epoch 16/20
211/211 - 3s - loss: 0.0184 - accuracy: 0.9940 - val_loss: 0.0493 -
val_accuracy: 0.9863 - 3s/epoch - 15ms/step
Epoch 17/20
211/211 - 3s - loss: 0.0169 - accuracy: 0.9943 - val_loss: 0.0467 -
val_accuracy: 0.9895 - 3s/epoch - 15ms/step
Epoch 18/20
211/211 - 3s - loss: 0.0144 - accuracy: 0.9958 - val_loss: 0.0503 -
val_accuracy: 0.9878 - 3s/epoch - 15ms/step
Epoch 19/20
```

```
211/211 - 3s - loss: 0.0138 - accuracy: 0.9955 - val_loss: 0.0466 -  
val_accuracy: 0.9878 - 3s/epoch - 15ms/step
```

```
Epoch 20/20
```

```
211/211 - 3s - loss: 0.0130 - accuracy: 0.9958 - val_loss: 0.0486 -  
val_accuracy: 0.9895 - 3s/epoch - 15ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

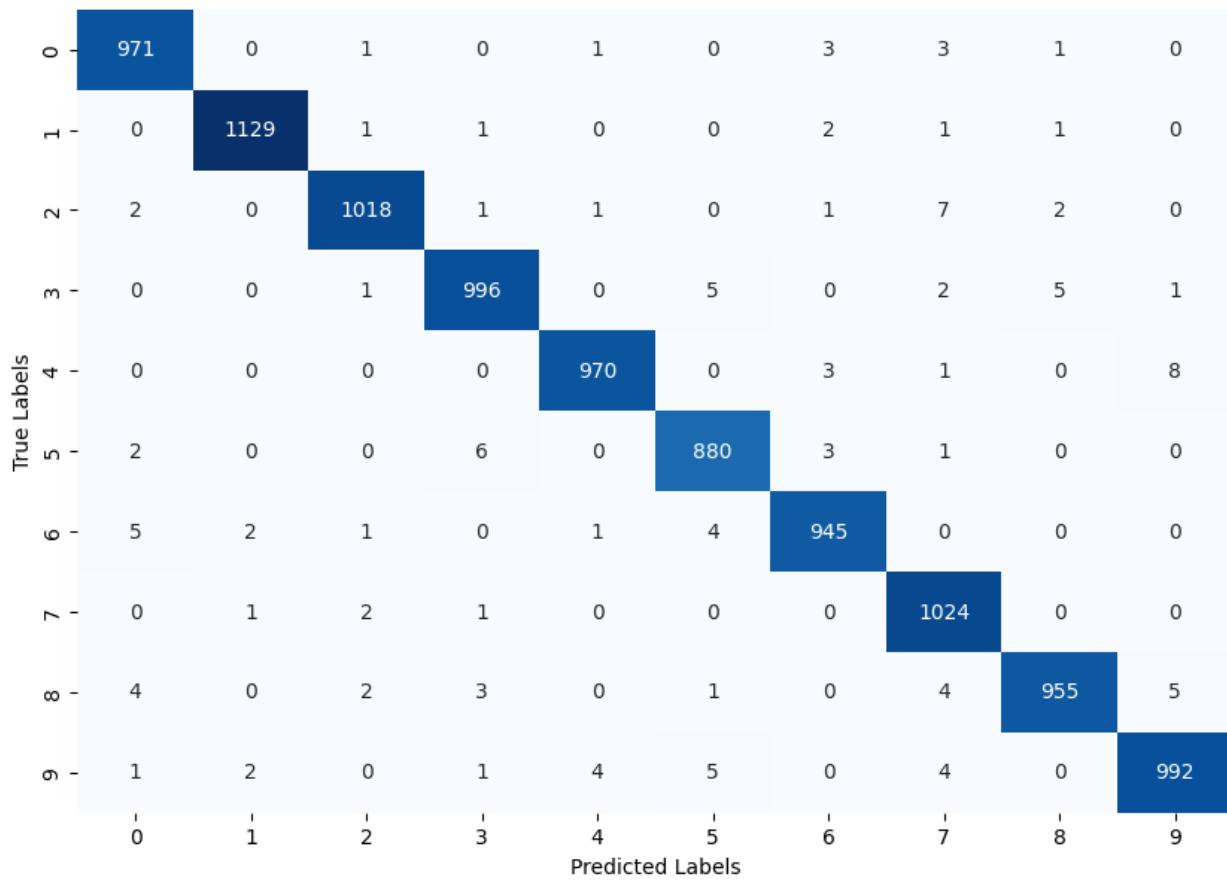
```
Confusion Matrix:
```

```
[[ 971   0   1   0   1   0   3   3   1   0]
 [ 0 1129   1   1   0   0   2   1   1   0]
 [ 2   0 1018   1   1   0   1   7   2   0]
 [ 0   0   1 996   0   5   0   2   5   1]
 [ 0   0   0   0 970   0   3   1   0   8]
 [ 2   0   0   6   0 880   3   1   0   0]
 [ 5   2   1   0   1   4 945   0   0   0]
 [ 0   1   2   1   0   0   0 1024   0   0]
 [ 4   0   2   3   0   1   0   4 955   5]
 [ 1   2   0   1   4   5   0   4   0 992]]
```

```
Precision: 0.9880
```

```
Recall: 0.9880
```

Confusion Matrix for relu Activation



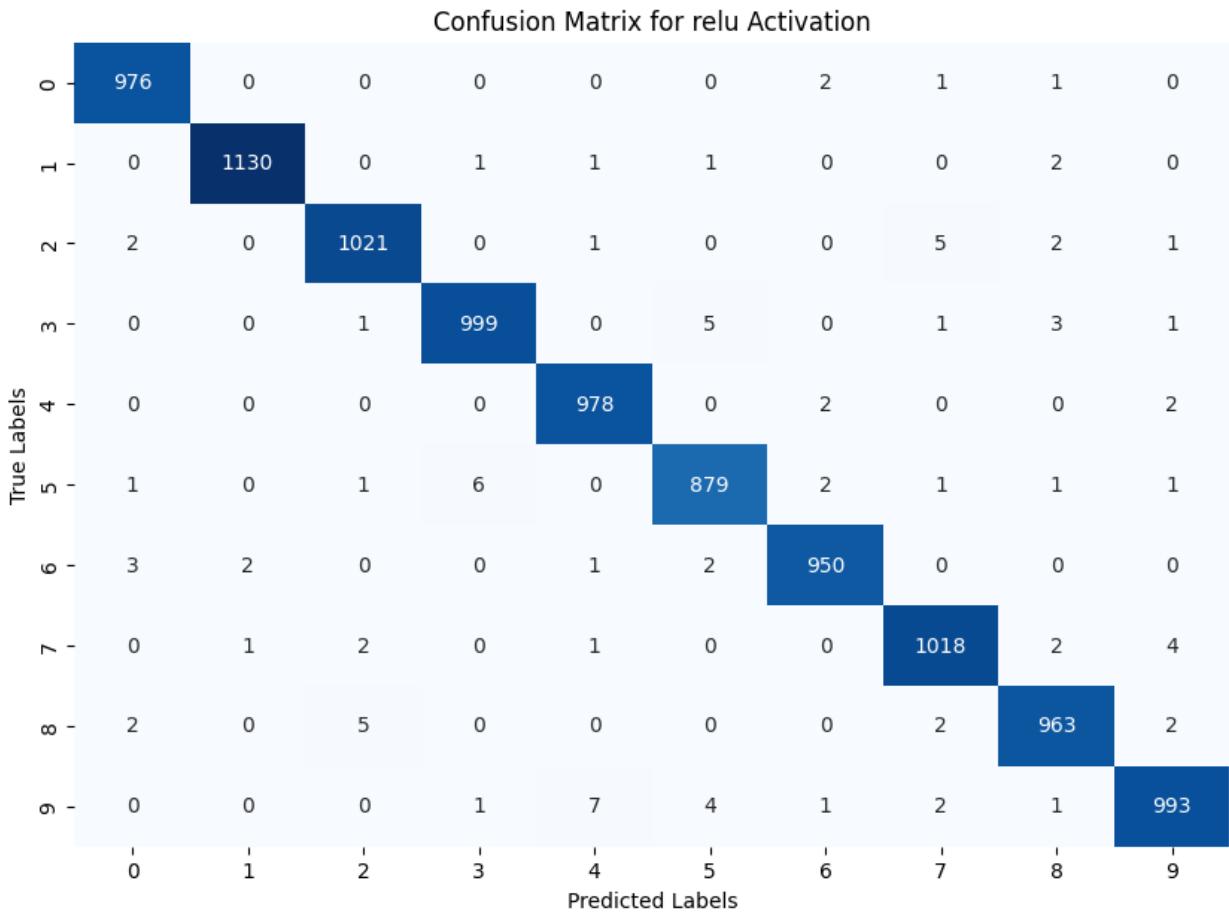
```
Training Model with relu activation, 2 conv_layers, 2 dense layers, 64
batch size, 5 epochs..
Epoch 1/5
844/844 - 7s - loss: 0.1804 - accuracy: 0.9438 - val_loss: 0.0790 -
val_accuracy: 0.9770 - 7s/epoch - 8ms/step
Epoch 2/5
844/844 - 6s - loss: 0.0593 - accuracy: 0.9820 - val_loss: 0.0437 -
val_accuracy: 0.9872 - 6s/epoch - 7ms/step
Epoch 3/5
844/844 - 6s - loss: 0.0398 - accuracy: 0.9874 - val_loss: 0.0416 -
val_accuracy: 0.9882 - 6s/epoch - 7ms/step
Epoch 4/5
844/844 - 6s - loss: 0.0297 - accuracy: 0.9910 - val_loss: 0.0381 -
val_accuracy: 0.9895 - 6s/epoch - 7ms/step
Epoch 5/5
844/844 - 6s - loss: 0.0232 - accuracy: 0.9926 - val_loss: 0.0400 -
val_accuracy: 0.9897 - 6s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 969    0    1    0    0    0    7    2    1    0]
 [  0 1130    0    0    1    0    3    1    0    0]
 [  1    1 1020    0    4    0    0    5    1    0]
 [  0    0    0 1004    0    3    0    1    2    0]
 [  0    0    0    0  980    0    1    0    0    1]
 [  1    0    0    6    0  880    4    1    0    0]
 [  1    2    0    0    2    1  952    0    0    0]
 [  0    1    3    0    0    0    0 1023    0    1]
 [  2    0    2    5    0    1    3    5  955    1]
 [  1    1    0    1   18    5    1   13    3 966]]
```

Precision: 0.9880
Recall: 0.9879

Confusion Matrix for relu Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	969	0	1	0	0	0	7	2	1	0	0
0	969	0	1	0	0	0	7	2	1	0	0
1	0	1130	0	0	1	0	3	1	0	0	0
2	1	1	1020	0	4	0	0	5	1	0	0
3	0	0	0	1004	0	3	0	1	2	0	0
4	0	0	0	0	980	0	1	0	0	1	0
5	1	0	0	6	0	880	4	1	0	0	0
6	1	2	0	0	2	1	952	0	0	0	0
7	0	1	3	0	0	0	0	1023	0	0	1
8	2	0	2	5	0	1	3	5	955	1	0
9	1	1	0	1	18	5	1	13	3	966	0
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```
Training Model with relu activation, 2 conv_layers, 2 dense layers, 64
batch size, 15 epochs..
Epoch 1/15
844/844 - 7s - loss: 0.1935 - accuracy: 0.9387 - val_loss: 0.0602 -
val_accuracy: 0.9820 - 7s/epoch - 8ms/step
Epoch 2/15
844/844 - 6s - loss: 0.0559 - accuracy: 0.9823 - val_loss: 0.0513 -
val_accuracy: 0.9865 - 6s/epoch - 7ms/step
Epoch 3/15
844/844 - 6s - loss: 0.0403 - accuracy: 0.9871 - val_loss: 0.0421 -
val_accuracy: 0.9890 - 6s/epoch - 7ms/step
Epoch 4/15
844/844 - 6s - loss: 0.0297 - accuracy: 0.9907 - val_loss: 0.0460 -
val_accuracy: 0.9868 - 6s/epoch - 7ms/step
Epoch 5/15
844/844 - 6s - loss: 0.0243 - accuracy: 0.9924 - val_loss: 0.0444 -
val_accuracy: 0.9888 - 6s/epoch - 7ms/step
Epoch 6/15
844/844 - 6s - loss: 0.0184 - accuracy: 0.9943 - val_loss: 0.0367 -
val_accuracy: 0.9908 - 6s/epoch - 7ms/step
Epoch 7/15
```

```
844/844 - 6s - loss: 0.0159 - accuracy: 0.9950 - val_loss: 0.0454 -  
val_accuracy: 0.9880 - 6s/epoch - 7ms/step  
Epoch 8/15  
844/844 - 6s - loss: 0.0122 - accuracy: 0.9961 - val_loss: 0.0470 -  
val_accuracy: 0.9875 - 6s/epoch - 7ms/step  
Epoch 9/15  
844/844 - 6s - loss: 0.0102 - accuracy: 0.9967 - val_loss: 0.0424 -  
val_accuracy: 0.9910 - 6s/epoch - 7ms/step  
Epoch 10/15  
844/844 - 6s - loss: 0.0074 - accuracy: 0.9977 - val_loss: 0.0523 -  
val_accuracy: 0.9885 - 6s/epoch - 7ms/step  
Epoch 11/15  
844/844 - 6s - loss: 0.0060 - accuracy: 0.9982 - val_loss: 0.0510 -  
val_accuracy: 0.9892 - 6s/epoch - 7ms/step  
Epoch 12/15  
844/844 - 6s - loss: 0.0055 - accuracy: 0.9982 - val_loss: 0.0530 -  
val_accuracy: 0.9887 - 6s/epoch - 7ms/step  
Epoch 13/15  
844/844 - 6s - loss: 0.0041 - accuracy: 0.9989 - val_loss: 0.0400 -  
val_accuracy: 0.9910 - 6s/epoch - 7ms/step  
Epoch 14/15  
844/844 - 6s - loss: 0.0024 - accuracy: 0.9994 - val_loss: 0.0422 -  
val_accuracy: 0.9920 - 6s/epoch - 7ms/step  
Epoch 15/15  
844/844 - 6s - loss: 0.0028 - accuracy: 0.9991 - val_loss: 0.0423 -  
val_accuracy: 0.9912 - 6s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 976  0  0  0  0  0  2  1  1  0]  
[  0 1130  0  1  1  1  0  0  2  0]  
[  2  0 1021  0  1  0  0  5  2  1]  
[  0  0  1 999  0  5  0  1  3  1]  
[  0  0  0  0 978  0  2  0  0  2]  
[  1  0  1  6  0 879  2  1  1  1]  
[  3  2  0  0  1  2 950  0  0  0]  
[  0  1  2  0  1  0  0 1018  2  4]  
[  2  0  5  0  0  0  0  2 963  2]  
[  0  0  0  1  7  4  1  2  1 993]]  
Precision: 0.9907  
Recall: 0.9907
```



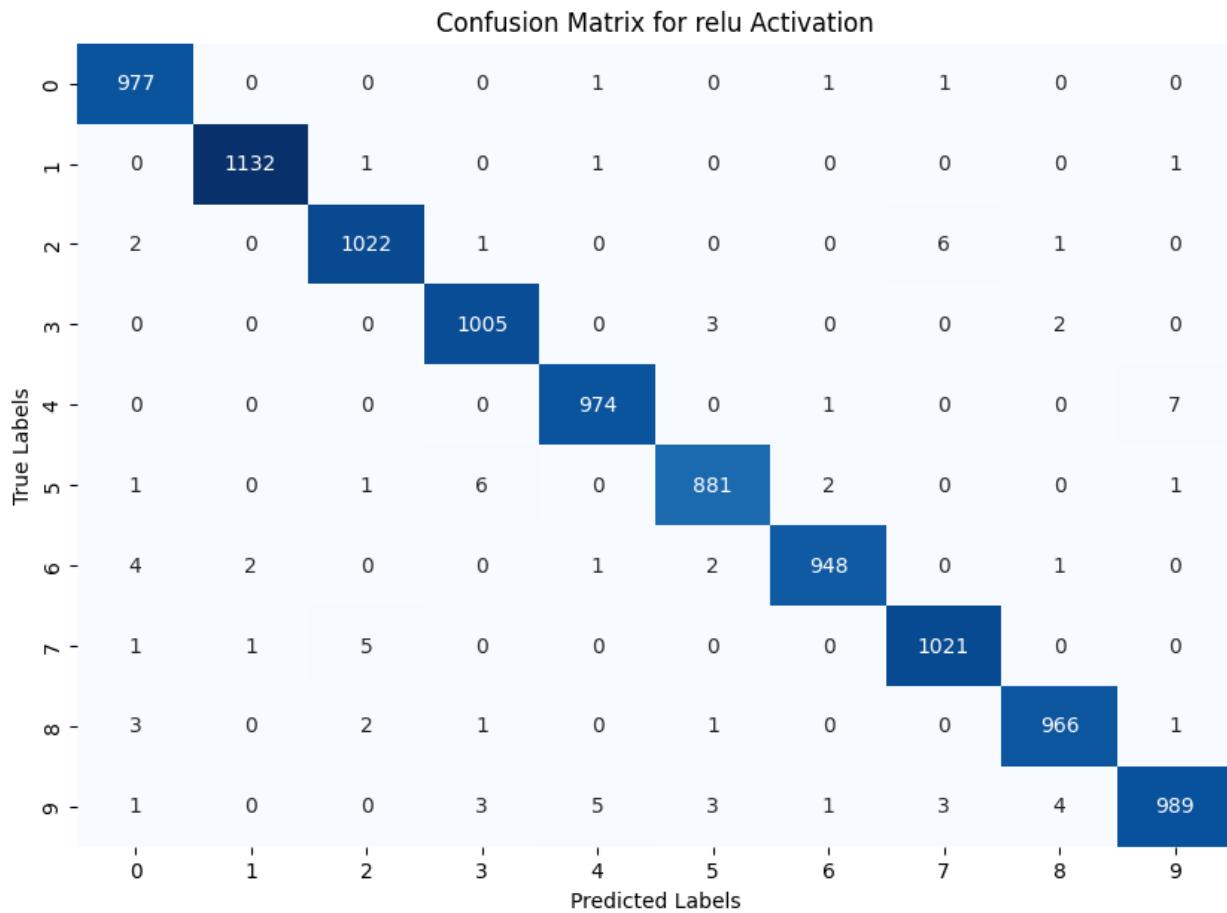
```
Training Model with relu activation, 2 conv_layers, 2 dense layers, 64
batch size, 20 epochs..
Epoch 1/20
844/844 - 7s - loss: 0.1817 - accuracy: 0.9431 - val_loss: 0.0715 -
val_accuracy: 0.9792 - 7s/epoch - 8ms/step
Epoch 2/20
844/844 - 6s - loss: 0.0537 - accuracy: 0.9839 - val_loss: 0.0501 -
val_accuracy: 0.9870 - 6s/epoch - 7ms/step
Epoch 3/20
844/844 - 6s - loss: 0.0378 - accuracy: 0.9882 - val_loss: 0.0410 -
val_accuracy: 0.9890 - 6s/epoch - 7ms/step
Epoch 4/20
844/844 - 6s - loss: 0.0297 - accuracy: 0.9907 - val_loss: 0.0398 -
val_accuracy: 0.9898 - 6s/epoch - 7ms/step
Epoch 5/20
844/844 - 6s - loss: 0.0223 - accuracy: 0.9928 - val_loss: 0.0419 -
val_accuracy: 0.9888 - 6s/epoch - 7ms/step
Epoch 6/20
844/844 - 6s - loss: 0.0175 - accuracy: 0.9945 - val_loss: 0.0417 -
val_accuracy: 0.9897 - 6s/epoch - 7ms/step
Epoch 7/20
```

```
844/844 - 6s - loss: 0.0140 - accuracy: 0.9955 - val_loss: 0.0378 -  
val_accuracy: 0.9912 - 6s/epoch - 7ms/step  
Epoch 8/20  
844/844 - 6s - loss: 0.0106 - accuracy: 0.9965 - val_loss: 0.0501 -  
val_accuracy: 0.9870 - 6s/epoch - 7ms/step  
Epoch 9/20  
844/844 - 6s - loss: 0.0078 - accuracy: 0.9975 - val_loss: 0.0404 -  
val_accuracy: 0.9913 - 6s/epoch - 7ms/step  
Epoch 10/20  
844/844 - 6s - loss: 0.0062 - accuracy: 0.9982 - val_loss: 0.0439 -  
val_accuracy: 0.9907 - 6s/epoch - 7ms/step  
Epoch 11/20  
844/844 - 6s - loss: 0.0055 - accuracy: 0.9982 - val_loss: 0.0466 -  
val_accuracy: 0.9900 - 6s/epoch - 7ms/step  
Epoch 12/20  
844/844 - 6s - loss: 0.0039 - accuracy: 0.9988 - val_loss: 0.0487 -  
val_accuracy: 0.9908 - 6s/epoch - 7ms/step  
Epoch 13/20  
844/844 - 6s - loss: 0.0021 - accuracy: 0.9995 - val_loss: 0.0466 -  
val_accuracy: 0.9912 - 6s/epoch - 7ms/step  
Epoch 14/20  
844/844 - 6s - loss: 0.0019 - accuracy: 0.9996 - val_loss: 0.0449 -  
val_accuracy: 0.9917 - 6s/epoch - 7ms/step  
Epoch 15/20  
844/844 - 6s - loss: 0.0012 - accuracy: 0.9998 - val_loss: 0.0467 -  
val_accuracy: 0.9917 - 6s/epoch - 7ms/step  
Epoch 16/20  
844/844 - 6s - loss: 6.3395e-04 - accuracy: 0.9999 - val_loss: 0.0467  
- val_accuracy: 0.9925 - 6s/epoch - 7ms/step  
Epoch 17/20  
844/844 - 6s - loss: 8.3572e-04 - accuracy: 0.9997 - val_loss: 0.0514  
- val_accuracy: 0.9907 - 6s/epoch - 7ms/step  
Epoch 18/20  
844/844 - 6s - loss: 0.0013 - accuracy: 0.9997 - val_loss: 0.0521 -  
val_accuracy: 0.9912 - 6s/epoch - 7ms/step  
Epoch 19/20  
844/844 - 6s - loss: 6.2735e-04 - accuracy: 0.9999 - val_loss: 0.0507  
- val_accuracy: 0.9917 - 6s/epoch - 7ms/step  
Epoch 20/20  
844/844 - 6s - loss: 2.7058e-04 - accuracy: 1.0000 - val_loss: 0.0513  
- val_accuracy: 0.9922 - 6s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 977  0  0  0  1  0  1  1  0  0 ]  
 [ 0 1132  1  0  1  0  0  0  0  1 ]  
 [ 2  0 1022  1  0  0  0  6  1  0 ]  
 [ 0  0  0 1005  0  3  0  0  2  0 ]  
 [ 0  0  0  0  974  0  1  0  0  7 ]
```

```
[ 1 0 1 6 0 881 2 0 0 1]
[ 4 2 0 0 1 2 948 0 1 0]
[ 1 1 5 0 0 0 0 1021 0 0]
[ 3 0 2 1 0 1 0 0 966 1]
[ 1 0 0 3 5 3 1 3 4 989]]
```

Precision: 0.9915

Recall: 0.9915



Training Model with relu activation, 2 conv_layers, 2 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 5s - loss: 0.2186 - accuracy: 0.9334 - val_loss: 0.0749 - val_accuracy: 0.9790 - 5s/epoch - 12ms/step

Epoch 2/5

422/422 - 4s - loss: 0.0705 - accuracy: 0.9786 - val_loss: 0.0580 - val_accuracy: 0.9837 - 4s/epoch - 10ms/step

Epoch 3/5

422/422 - 4s - loss: 0.0524 - accuracy: 0.9838 - val_loss: 0.0642 - val_accuracy: 0.9805 - 4s/epoch - 10ms/step

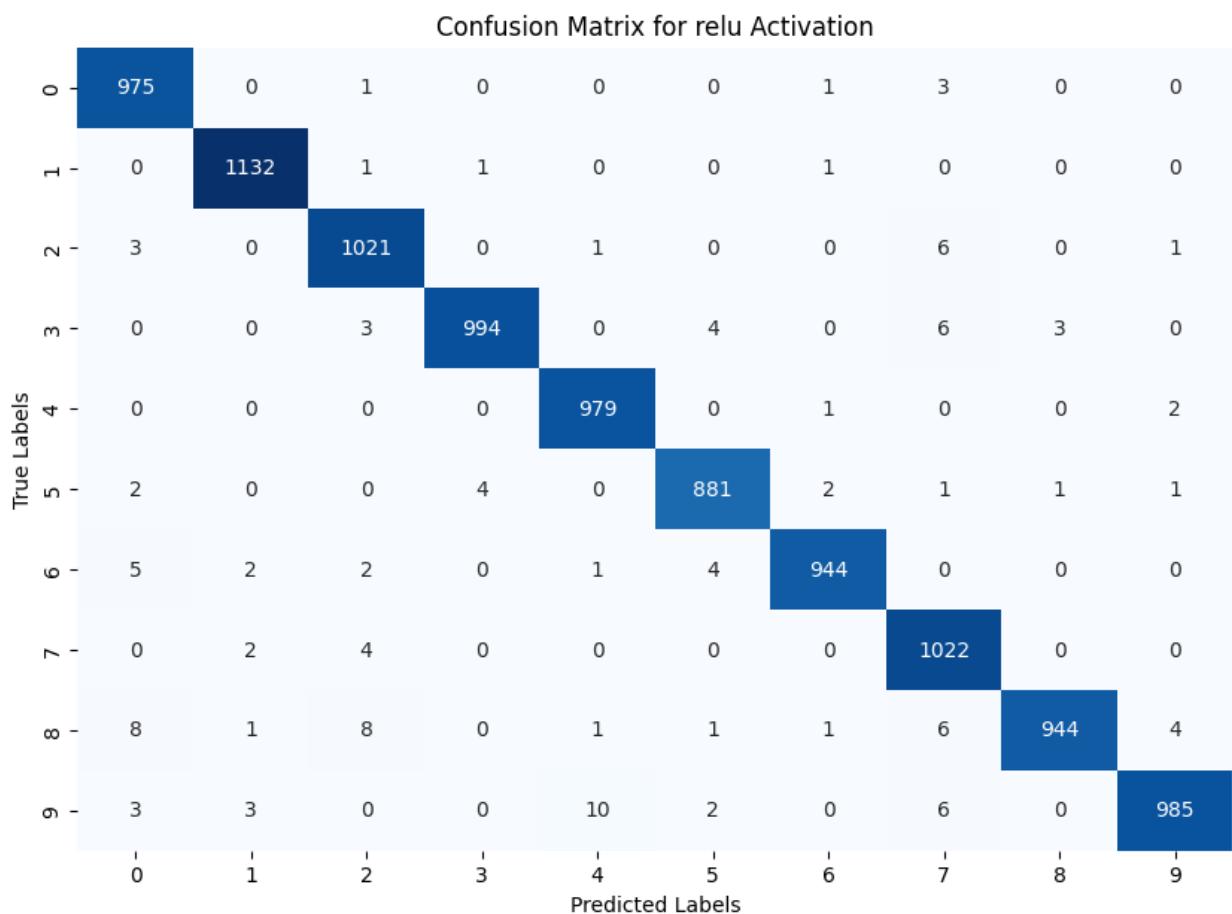
Epoch 4/5

422/422 - 4s - loss: 0.0408 - accuracy: 0.9870 - val_loss: 0.0491 -

```

val_accuracy: 0.9862 - 4s/epoch - 10ms/step
Epoch 5/5
422/422 - 4s - loss: 0.0327 - accuracy: 0.9901 - val_loss: 0.0437 -
val_accuracy: 0.9880 - 4s/epoch - 10ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 975   0   1   0   0   0   1   3   0   0]
 [ 0 1132   1   1   0   0   1   0   0   0]
 [ 3   0 1021   0   1   0   0   6   0   1]
 [ 0   0   3 994   0   4   0   6   3   0]
 [ 0   0   0   0 979   0   1   0   0   2]
 [ 2   0   0   4   0 881   2   1   1   1]
 [ 5   2   2   0   1   4 944   0   0   0]
 [ 0   2   4   0   0   0   0 1022   0   0]
 [ 8   1   8   0   1   1   1   6 944   4]
 [ 3   3   0   0 10   2   0   6   0 985]]
Precision: 0.9878
Recall: 0.9877

```



Training Model with relu activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..

Epoch 1/15
422/422 - 5s - loss: 0.2502 - accuracy: 0.9217 - val_loss: 0.0963 -
val_accuracy: 0.9710 - 5s/epoch - 12ms/step

Epoch 2/15
422/422 - 4s - loss: 0.0738 - accuracy: 0.9779 - val_loss: 0.0574 -
val_accuracy: 0.9828 - 4s/epoch - 10ms/step

Epoch 3/15
422/422 - 4s - loss: 0.0517 - accuracy: 0.9837 - val_loss: 0.0642 -
val_accuracy: 0.9817 - 4s/epoch - 10ms/step

Epoch 4/15
422/422 - 4s - loss: 0.0417 - accuracy: 0.9875 - val_loss: 0.0460 -
val_accuracy: 0.9873 - 4s/epoch - 10ms/step

Epoch 5/15
422/422 - 4s - loss: 0.0343 - accuracy: 0.9884 - val_loss: 0.0389 -
val_accuracy: 0.9880 - 4s/epoch - 10ms/step

Epoch 6/15
422/422 - 4s - loss: 0.0274 - accuracy: 0.9914 - val_loss: 0.0386 -
val_accuracy: 0.9887 - 4s/epoch - 10ms/step

Epoch 7/15
422/422 - 4s - loss: 0.0228 - accuracy: 0.9926 - val_loss: 0.0380 -
val_accuracy: 0.9902 - 4s/epoch - 10ms/step

Epoch 8/15
422/422 - 4s - loss: 0.0196 - accuracy: 0.9935 - val_loss: 0.0392 -
val_accuracy: 0.9890 - 4s/epoch - 10ms/step

Epoch 9/15
422/422 - 4s - loss: 0.0163 - accuracy: 0.9945 - val_loss: 0.0386 -
val_accuracy: 0.9893 - 4s/epoch - 10ms/step

Epoch 10/15
422/422 - 4s - loss: 0.0142 - accuracy: 0.9954 - val_loss: 0.0414 -
val_accuracy: 0.9883 - 4s/epoch - 10ms/step

Epoch 11/15
422/422 - 4s - loss: 0.0119 - accuracy: 0.9963 - val_loss: 0.0339 -
val_accuracy: 0.9912 - 4s/epoch - 10ms/step

Epoch 12/15
422/422 - 4s - loss: 0.0101 - accuracy: 0.9972 - val_loss: 0.0382 -
val_accuracy: 0.9907 - 4s/epoch - 10ms/step

Epoch 13/15
422/422 - 4s - loss: 0.0084 - accuracy: 0.9977 - val_loss: 0.0391 -
val_accuracy: 0.9905 - 4s/epoch - 10ms/step

Epoch 14/15
422/422 - 4s - loss: 0.0070 - accuracy: 0.9980 - val_loss: 0.0398 -
val_accuracy: 0.9893 - 4s/epoch - 10ms/step

Epoch 15/15
422/422 - 4s - loss: 0.0060 - accuracy: 0.9982 - val_loss: 0.0383 -
val_accuracy: 0.9898 - 4s/epoch - 10ms/step

313/313 [=====] - 1s 2ms/step

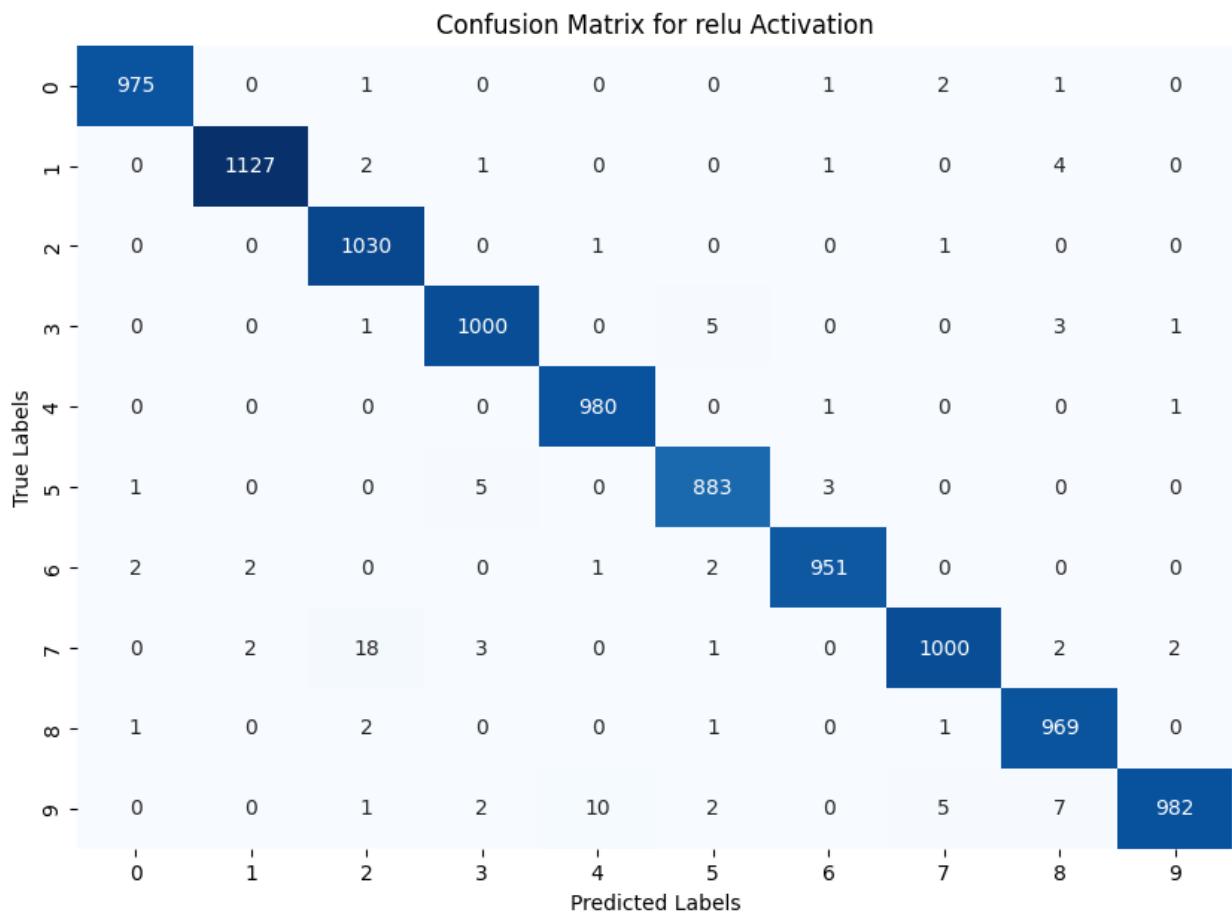
Results for activation function: relu

Confusion Matrix:

```
[[ 975  0   1   0   0   0   1   2   1   0]
 [ 0 1127  2   1   0   0   1   0   4   0]
 [ 0   0 1030  0   1   0   0   1   0   0]
 [ 0   0 1000  0   0   5   0   0   3   1]
 [ 0   0   0 980  0   0   1   0   0   1]
 [ 1   0   0   5  0   883  3   0   0   0]
 [ 2   2   0   0   1   2   951  0   0   0]
 [ 0   2   18  3   0   1   0   1000  2   2]
 [ 1   0   2   0   0   1   0   1   969  0]
 [ 0   0   1   2   10  2   0   5   7   982]]
```

Precision: 0.9898

Recall: 0.9897



Training Model with relu activation, 2 conv_layers, 2 dense layers,
128 batch size, 20 epochs..

Epoch 1/20

422/422 - 5s - loss: 0.2573 - accuracy: 0.9201 - val_loss: 0.0730 -
val_accuracy: 0.9803 - 5s/epoch - 12ms/step

Epoch 2/20

422/422 - 4s - loss: 0.0750 - accuracy: 0.9776 - val_loss: 0.0596 -
val_accuracy: 0.9823 - 4s/epoch - 10ms/step

```
Epoch 3/20
422/422 - 4s - loss: 0.0531 - accuracy: 0.9839 - val_loss: 0.0576 -
val_accuracy: 0.9838 - 4s/epoch - 10ms/step
Epoch 4/20
422/422 - 4s - loss: 0.0426 - accuracy: 0.9867 - val_loss: 0.0465 -
val_accuracy: 0.9858 - 4s/epoch - 10ms/step
Epoch 5/20
422/422 - 4s - loss: 0.0346 - accuracy: 0.9891 - val_loss: 0.0416 -
val_accuracy: 0.9882 - 4s/epoch - 10ms/step
Epoch 6/20
422/422 - 4s - loss: 0.0291 - accuracy: 0.9911 - val_loss: 0.0522 -
val_accuracy: 0.9838 - 4s/epoch - 10ms/step
Epoch 7/20
422/422 - 4s - loss: 0.0240 - accuracy: 0.9924 - val_loss: 0.0384 -
val_accuracy: 0.9890 - 4s/epoch - 10ms/step
Epoch 8/20
422/422 - 4s - loss: 0.0197 - accuracy: 0.9941 - val_loss: 0.0370 -
val_accuracy: 0.9902 - 4s/epoch - 10ms/step
Epoch 9/20
422/422 - 4s - loss: 0.0167 - accuracy: 0.9948 - val_loss: 0.0350 -
val_accuracy: 0.9903 - 4s/epoch - 10ms/step
Epoch 10/20
422/422 - 4s - loss: 0.0147 - accuracy: 0.9952 - val_loss: 0.0394 -
val_accuracy: 0.9893 - 4s/epoch - 10ms/step
Epoch 11/20
422/422 - 4s - loss: 0.0124 - accuracy: 0.9962 - val_loss: 0.0368 -
val_accuracy: 0.9903 - 4s/epoch - 10ms/step
Epoch 12/20
422/422 - 4s - loss: 0.0110 - accuracy: 0.9966 - val_loss: 0.0331 -
val_accuracy: 0.9912 - 4s/epoch - 10ms/step
Epoch 13/20
422/422 - 4s - loss: 0.0090 - accuracy: 0.9976 - val_loss: 0.0389 -
val_accuracy: 0.9913 - 4s/epoch - 10ms/step
Epoch 14/20
422/422 - 4s - loss: 0.0077 - accuracy: 0.9976 - val_loss: 0.0399 -
val_accuracy: 0.9903 - 4s/epoch - 10ms/step
Epoch 15/20
422/422 - 4s - loss: 0.0064 - accuracy: 0.9981 - val_loss: 0.0409 -
val_accuracy: 0.9907 - 4s/epoch - 10ms/step
Epoch 16/20
422/422 - 4s - loss: 0.0051 - accuracy: 0.9984 - val_loss: 0.0441 -
val_accuracy: 0.9910 - 4s/epoch - 10ms/step
Epoch 17/20
422/422 - 4s - loss: 0.0041 - accuracy: 0.9989 - val_loss: 0.0420 -
val_accuracy: 0.9917 - 4s/epoch - 10ms/step
Epoch 18/20
422/422 - 4s - loss: 0.0033 - accuracy: 0.9991 - val_loss: 0.0447 -
val_accuracy: 0.9912 - 4s/epoch - 10ms/step
Epoch 19/20
```

```
422/422 - 4s - loss: 0.0024 - accuracy: 0.9995 - val_loss: 0.0417 -  
val_accuracy: 0.9913 - 4s/epoch - 10ms/step
```

```
Epoch 20/20
```

```
422/422 - 4s - loss: 0.0018 - accuracy: 0.9997 - val_loss: 0.0420 -  
val_accuracy: 0.9912 - 4s/epoch - 10ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

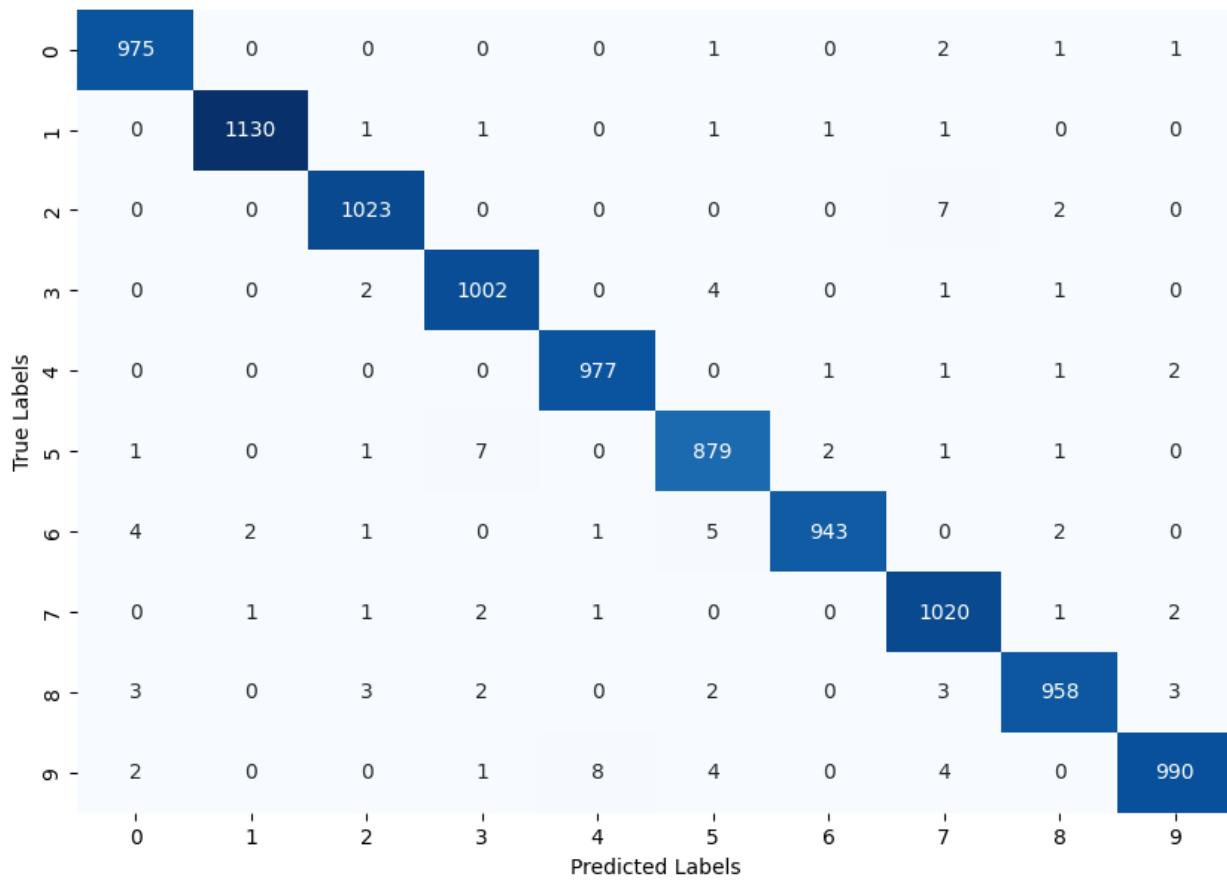
```
Confusion Matrix:
```

```
[[ 975  0  0  0  0  1  0  2  1  1]  
 [ 0 1130  1  1  0  1  1  1  0  0]  
 [ 0  0 1023  0  0  0  0  7  2  0]  
 [ 0  0  2 1002  0  4  0  1  1  0]  
 [ 0  0  0  0 977  0  1  1  1  2]  
 [ 1  0  1  7  0 879  2  1  1  0]  
 [ 4  2  1  0  1  5 943  0  2  0]  
 [ 0  1  1  2  1  0  0 1020  1  2]  
 [ 3  0  3  2  0  2  0  3 958  3]  
 [ 2  0  0  1  8  4  0  4  0 990]]
```

```
Precision: 0.9897
```

```
Recall: 0.9897
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 2 conv_layers, 2 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 4s - loss: 0.3651 - accuracy: 0.8871 - val_loss: 0.1029 -
val_accuracy: 0.9697 - 4s/epoch - 20ms/step
Epoch 2/5
211/211 - 3s - loss: 0.0981 - accuracy: 0.9703 - val_loss: 0.0804 -
val_accuracy: 0.9770 - 3s/epoch - 16ms/step
Epoch 3/5
211/211 - 4s - loss: 0.0707 - accuracy: 0.9779 - val_loss: 0.0638 -
val_accuracy: 0.9822 - 4s/epoch - 17ms/step
Epoch 4/5
211/211 - 3s - loss: 0.0565 - accuracy: 0.9827 - val_loss: 0.0605 -
val_accuracy: 0.9817 - 3s/epoch - 16ms/step
Epoch 5/5
211/211 - 3s - loss: 0.0460 - accuracy: 0.9861 - val_loss: 0.0525 -
val_accuracy: 0.9855 - 3s/epoch - 16ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 969    0    0    0    0    1    1    2    3    4]
 [  0 1131    1    1    0    0    1    1    0    0]
 [  4    4 1003    5    1    0    0    8    6    1]
 [  0    0    1 993    0    6    0    4    5    1]
 [  0    0    0    0 969    0    0    1    0   12]
 [  1    0    0    5    0 880    4    1    0    1]
 [  4    2    0    0    6    5 938    0    3    0]
 [  1    1    3    2    0    0    0 1019    1    1]
 [  3    0    0    1    1    0    2    5 956    6]
 [  0    2    0    1    2    3    0    6    1 994]]
```

Precision: 0.9853
Recall: 0.9852

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
0	969	0	0	0	0	1	1	2	3	4
1	0	1131	1	1	0	0	1	1	0	0
2	4	4	1003	5	1	0	0	8	6	1
3	0	0	1	993	0	6	0	4	5	1
4	0	0	0	0	969	0	0	1	0	12
5	1	0	0	5	0	880	4	1	0	1
6	4	2	0	0	6	5	938	0	3	0
7	1	1	3	2	0	0	0	1019	1	1
8	3	0	0	1	1	0	2	5	956	6
9	0	2	0	1	2	3	0	6	1	994
	0	1	2	3	4	5	6	7	8	9
	True Labels									Predicted Labels

Training Model with relu activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..

Epoch 1/15

211/211 - 4s - loss: 0.3684 - accuracy: 0.8846 - val_loss: 0.1087 -
val_accuracy: 0.9678 - 4s/epoch - 20ms/step

Epoch 2/15

211/211 - 3s - loss: 0.1012 - accuracy: 0.9694 - val_loss: 0.0787 -
val_accuracy: 0.9803 - 3s/epoch - 16ms/step

Epoch 3/15

211/211 - 3s - loss: 0.0739 - accuracy: 0.9775 - val_loss: 0.0752 -
val_accuracy: 0.9773 - 3s/epoch - 16ms/step

Epoch 4/15

211/211 - 3s - loss: 0.0574 - accuracy: 0.9824 - val_loss: 0.0566 -
val_accuracy: 0.9845 - 3s/epoch - 16ms/step

Epoch 5/15

211/211 - 3s - loss: 0.0461 - accuracy: 0.9861 - val_loss: 0.0592 -
val_accuracy: 0.9845 - 3s/epoch - 16ms/step

Epoch 6/15

211/211 - 3s - loss: 0.0415 - accuracy: 0.9870 - val_loss: 0.0526 -
val_accuracy: 0.9875 - 3s/epoch - 16ms/step

Epoch 7/15

```
211/211 - 3s - loss: 0.0347 - accuracy: 0.9894 - val_loss: 0.0510 -  
val_accuracy: 0.9860 - 3s/epoch - 16ms/step  
Epoch 8/15  
211/211 - 3s - loss: 0.0317 - accuracy: 0.9902 - val_loss: 0.0456 -  
val_accuracy: 0.9897 - 3s/epoch - 16ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.0261 - accuracy: 0.9919 - val_loss: 0.0507 -  
val_accuracy: 0.9872 - 3s/epoch - 16ms/step  
Epoch 10/15  
211/211 - 3s - loss: 0.0256 - accuracy: 0.9919 - val_loss: 0.0446 -  
val_accuracy: 0.9885 - 3s/epoch - 16ms/step  
Epoch 11/15  
211/211 - 3s - loss: 0.0218 - accuracy: 0.9934 - val_loss: 0.0444 -  
val_accuracy: 0.9887 - 3s/epoch - 16ms/step  
Epoch 12/15  
211/211 - 3s - loss: 0.0198 - accuracy: 0.9939 - val_loss: 0.0436 -  
val_accuracy: 0.9892 - 3s/epoch - 16ms/step  
Epoch 13/15  
211/211 - 3s - loss: 0.0174 - accuracy: 0.9947 - val_loss: 0.0467 -  
val_accuracy: 0.9882 - 3s/epoch - 16ms/step  
Epoch 14/15  
211/211 - 3s - loss: 0.0153 - accuracy: 0.9951 - val_loss: 0.0427 -  
val_accuracy: 0.9890 - 3s/epoch - 16ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.0144 - accuracy: 0.9958 - val_loss: 0.0489 -  
val_accuracy: 0.9880 - 3s/epoch - 16ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 975  0  1  0  0  0  2  1  1  0]  
[  0 1131  1  1  0  0  1  1  0  0]  
[  1  1 1022  0  1  0  1  5  1  0]  
[  1  0  1 1002  0  3  0  0  2  1]  
[  0  0  1  0  968  0  2  2  0  9]  
[  2  0  0  7  0  878  2  1  0  2]  
[  4  2  0  0  0  3  946  0  3  0]  
[  0  2  3  5  0  0  0 1010  3  5]  
[  2  0  2  5  0  1  0  1  960  3]  
[  1  1  0  4  4  5  1  0  0  993]]  
Precision: 0.9885  
Recall: 0.9885
```

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	975	0	1	0	0	0	2	1	1	0
0	975	0	1	0	0	0	2	1	1	0
1	0	1131	1	1	0	0	1	1	0	0
2	1	1	1022	0	1	0	1	5	1	0
3	1	0	1	1002	0	3	0	0	2	1
4	0	0	1	0	968	0	2	2	0	9
5	2	0	0	7	0	878	2	1	0	2
6	4	2	0	0	0	3	946	0	3	0
7	0	2	3	5	0	0	0	1010	3	5
8	2	0	2	5	0	1	0	1	960	3
9	1	1	0	4	4	5	1	0	0	993
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

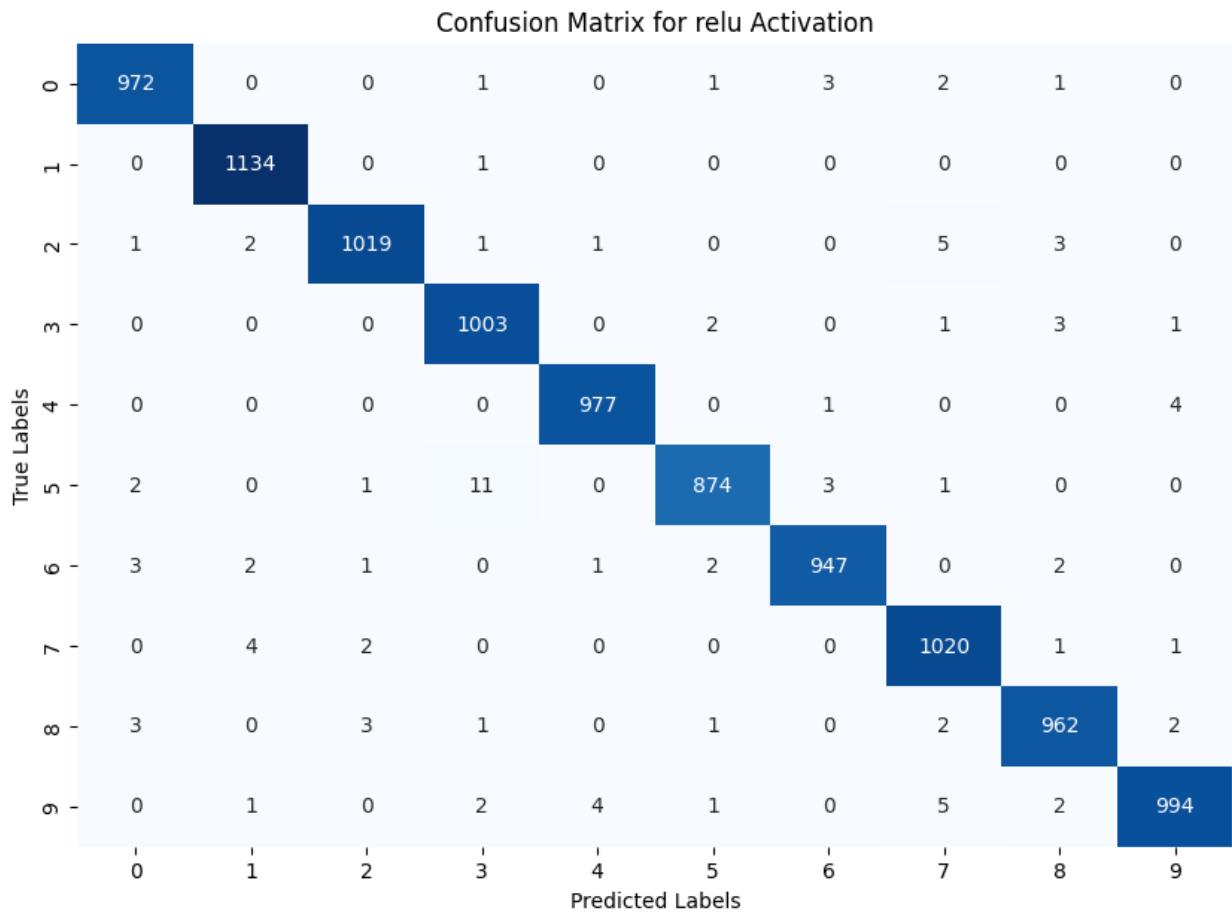
```
Training Model with relu activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 4s - loss: 0.3501 - accuracy: 0.8949 - val_loss: 0.1004 -
val_accuracy: 0.9708 - 4s/epoch - 19ms/step
Epoch 2/20
211/211 - 3s - loss: 0.0984 - accuracy: 0.9694 - val_loss: 0.0732 -
val_accuracy: 0.9812 - 3s/epoch - 16ms/step
Epoch 3/20
211/211 - 3s - loss: 0.0707 - accuracy: 0.9787 - val_loss: 0.0728 -
val_accuracy: 0.9790 - 3s/epoch - 16ms/step
Epoch 4/20
211/211 - 3s - loss: 0.0596 - accuracy: 0.9816 - val_loss: 0.0538 -
val_accuracy: 0.9853 - 3s/epoch - 16ms/step
Epoch 5/20
211/211 - 3s - loss: 0.0498 - accuracy: 0.9846 - val_loss: 0.0587 -
val_accuracy: 0.9842 - 3s/epoch - 16ms/step
Epoch 6/20
211/211 - 3s - loss: 0.0415 - accuracy: 0.9874 - val_loss: 0.0615 -
val_accuracy: 0.9820 - 3s/epoch - 16ms/step
Epoch 7/20
```

```
211/211 - 3s - loss: 0.0377 - accuracy: 0.9885 - val_loss: 0.0441 -  
val_accuracy: 0.9868 - 3s/epoch - 16ms/step  
Epoch 8/20  
211/211 - 3s - loss: 0.0330 - accuracy: 0.9903 - val_loss: 0.0549 -  
val_accuracy: 0.9860 - 3s/epoch - 16ms/step  
Epoch 9/20  
211/211 - 3s - loss: 0.0287 - accuracy: 0.9913 - val_loss: 0.0459 -  
val_accuracy: 0.9880 - 3s/epoch - 16ms/step  
Epoch 10/20  
211/211 - 3s - loss: 0.0266 - accuracy: 0.9917 - val_loss: 0.0489 -  
val_accuracy: 0.9858 - 3s/epoch - 16ms/step  
Epoch 11/20  
211/211 - 3s - loss: 0.0246 - accuracy: 0.9923 - val_loss: 0.0429 -  
val_accuracy: 0.9875 - 3s/epoch - 16ms/step  
Epoch 12/20  
211/211 - 3s - loss: 0.0210 - accuracy: 0.9934 - val_loss: 0.0478 -  
val_accuracy: 0.9875 - 3s/epoch - 16ms/step  
Epoch 13/20  
211/211 - 3s - loss: 0.0186 - accuracy: 0.9941 - val_loss: 0.0432 -  
val_accuracy: 0.9880 - 3s/epoch - 16ms/step  
Epoch 14/20  
211/211 - 3s - loss: 0.0169 - accuracy: 0.9946 - val_loss: 0.0470 -  
val_accuracy: 0.9877 - 3s/epoch - 16ms/step  
Epoch 15/20  
211/211 - 3s - loss: 0.0151 - accuracy: 0.9955 - val_loss: 0.0502 -  
val_accuracy: 0.9872 - 3s/epoch - 16ms/step  
Epoch 16/20  
211/211 - 4s - loss: 0.0136 - accuracy: 0.9959 - val_loss: 0.0438 -  
val_accuracy: 0.9877 - 4s/epoch - 17ms/step  
Epoch 17/20  
211/211 - 4s - loss: 0.0119 - accuracy: 0.9964 - val_loss: 0.0472 -  
val_accuracy: 0.9878 - 4s/epoch - 17ms/step  
Epoch 18/20  
211/211 - 3s - loss: 0.0099 - accuracy: 0.9973 - val_loss: 0.0429 -  
val_accuracy: 0.9897 - 3s/epoch - 16ms/step  
Epoch 19/20  
211/211 - 3s - loss: 0.0093 - accuracy: 0.9976 - val_loss: 0.0490 -  
val_accuracy: 0.9880 - 3s/epoch - 16ms/step  
Epoch 20/20  
211/211 - 3s - loss: 0.0078 - accuracy: 0.9979 - val_loss: 0.0470 -  
val_accuracy: 0.9890 - 3s/epoch - 16ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 972 0 0 1 0 1 3 2 1 0]  
[ 0 1134 0 1 0 0 0 0 0 0]  
[ 1 2 1019 1 1 0 0 5 3 0]  
[ 0 0 0 1003 0 2 0 1 3 1]  
[ 0 0 0 0 977 0 1 0 0 4]]
```

```
[ 2 0 1 11 0 874 3 1 0 0]
[ 3 2 1 0 1 2 947 0 2 0]
[ 0 4 2 0 0 0 0 1020 1 1]
[ 3 0 3 1 0 1 0 2 962 2]
[ 0 1 0 2 4 1 0 5 2 994]]
```

Precision: 0.9902

Recall: 0.9902



Training Model with relu activation, 2 conv_layers, 2 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 7s - loss: 0.1766 - accuracy: 0.9445 - val_loss: 0.0622 - val_accuracy: 0.9822 - 7s/epoch - 8ms/step

Epoch 2/5

844/844 - 6s - loss: 0.0538 - accuracy: 0.9830 - val_loss: 0.0566 - val_accuracy: 0.9825 - 6s/epoch - 8ms/step

Epoch 3/5

844/844 - 6s - loss: 0.0350 - accuracy: 0.9891 - val_loss: 0.0399 - val_accuracy: 0.9878 - 6s/epoch - 8ms/step

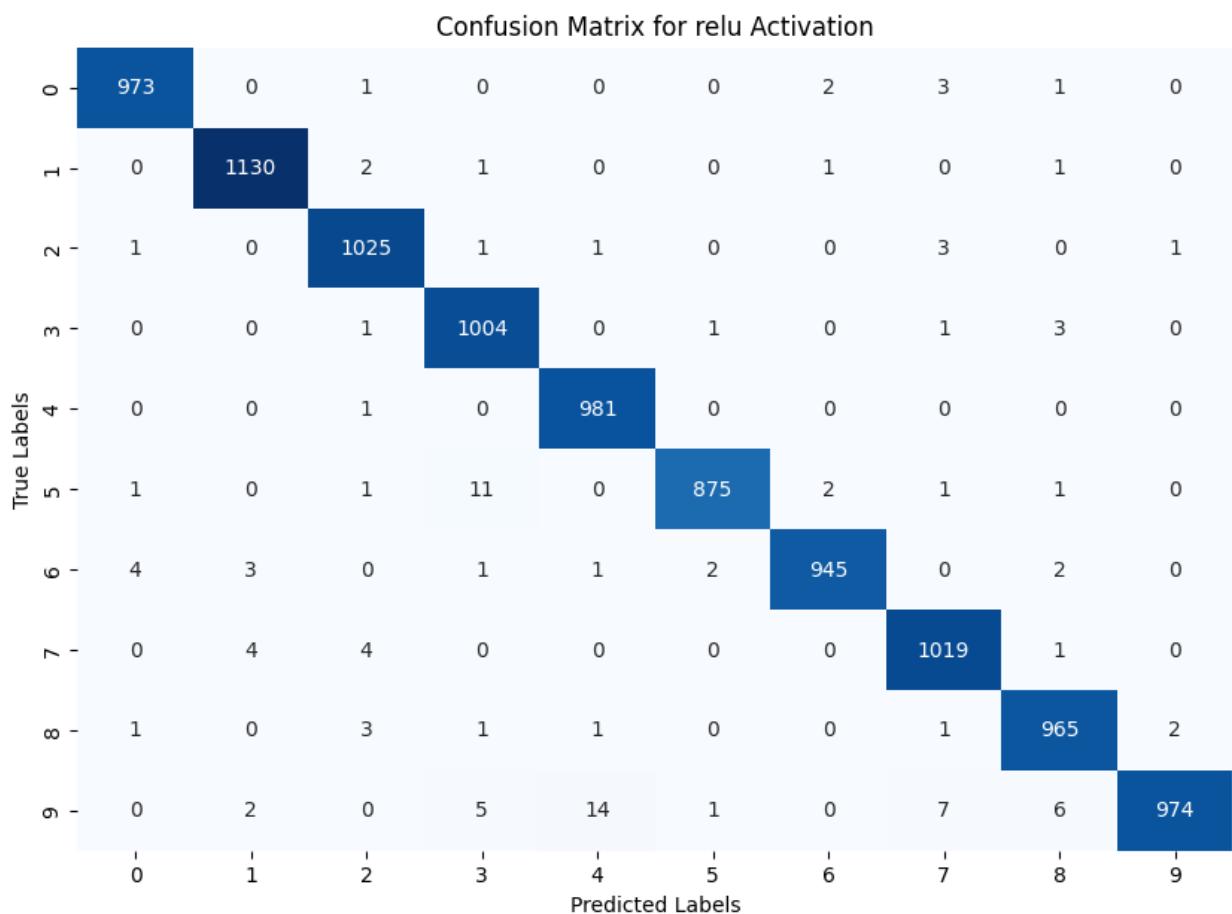
Epoch 4/5

844/844 - 6s - loss: 0.0246 - accuracy: 0.9919 - val_loss: 0.0467 -

```

val_accuracy: 0.9872 - 6s/epoch - 8ms/step
Epoch 5/5
844/844 - 6s - loss: 0.0199 - accuracy: 0.9936 - val_loss: 0.0391 -
val_accuracy: 0.9888 - 6s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 973   0   1   0   0   0   2   3   1   0]
 [ 0 1130   2   1   0   0   1   0   1   0]
 [ 1   0 1025   1   1   0   0   3   0   1]
 [ 0   0   1 1004   0   1   0   1   3   0]
 [ 0   0   1   0 981   0   0   0   0   0]
 [ 1   0   1   11   0 875   2   1   1   0]
 [ 4   3   0   1   1   2 945   0   2   0]
 [ 0   4   4   0   0   0   0 1019   1   0]
 [ 1   0   3   1   1   0   0   1 965   2]
 [ 0   2   0   5 14   1   0   7   6 974]]
Precision: 0.9892
Recall: 0.9891

```

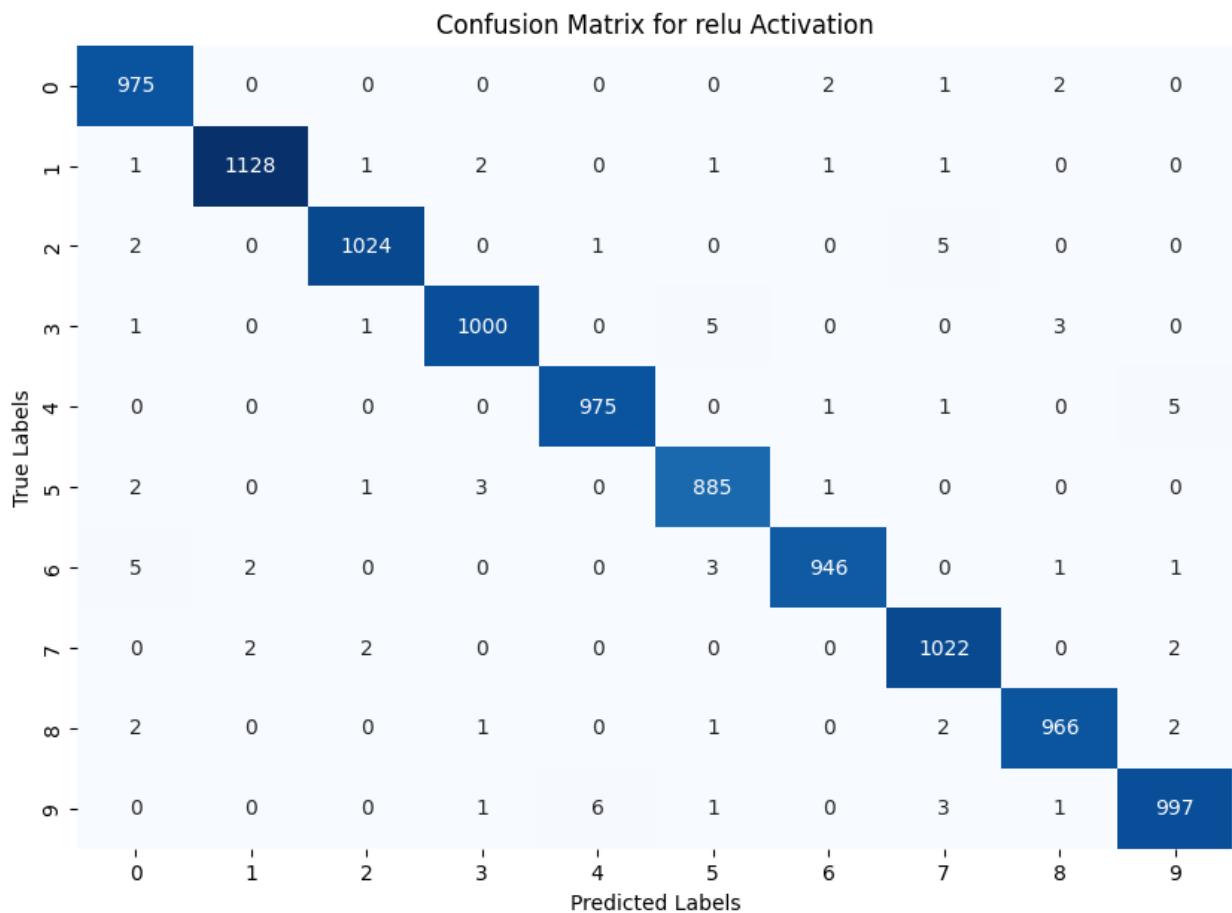


```
Training Model with relu activation, 2 conv_layers, 2 dense layers, 64
batch size, 15 epochs..
Epoch 1/15
844/844 - 7s - loss: 0.1691 - accuracy: 0.9471 - val_loss: 0.0618 -
val_accuracy: 0.9827 - 7s/epoch - 8ms/step
Epoch 2/15
844/844 - 6s - loss: 0.0516 - accuracy: 0.9833 - val_loss: 0.0472 -
val_accuracy: 0.9865 - 6s/epoch - 8ms/step
Epoch 3/15
844/844 - 6s - loss: 0.0361 - accuracy: 0.9884 - val_loss: 0.0488 -
val_accuracy: 0.9858 - 6s/epoch - 8ms/step
Epoch 4/15
844/844 - 6s - loss: 0.0246 - accuracy: 0.9921 - val_loss: 0.0394 -
val_accuracy: 0.9878 - 6s/epoch - 8ms/step
Epoch 5/15
844/844 - 6s - loss: 0.0192 - accuracy: 0.9940 - val_loss: 0.0385 -
val_accuracy: 0.9903 - 6s/epoch - 8ms/step
Epoch 6/15
844/844 - 6s - loss: 0.0148 - accuracy: 0.9957 - val_loss: 0.0331 -
val_accuracy: 0.9907 - 6s/epoch - 7ms/step
Epoch 7/15
844/844 - 6s - loss: 0.0109 - accuracy: 0.9967 - val_loss: 0.0341 -
val_accuracy: 0.9913 - 6s/epoch - 7ms/step
Epoch 8/15
844/844 - 6s - loss: 0.0079 - accuracy: 0.9978 - val_loss: 0.0437 -
val_accuracy: 0.9900 - 6s/epoch - 7ms/step
Epoch 9/15
844/844 - 6s - loss: 0.0054 - accuracy: 0.9988 - val_loss: 0.0496 -
val_accuracy: 0.9900 - 6s/epoch - 7ms/step
Epoch 10/15
844/844 - 6s - loss: 0.0049 - accuracy: 0.9985 - val_loss: 0.0369 -
val_accuracy: 0.9927 - 6s/epoch - 8ms/step
Epoch 11/15
844/844 - 6s - loss: 0.0031 - accuracy: 0.9991 - val_loss: 0.0420 -
val_accuracy: 0.9910 - 6s/epoch - 7ms/step
Epoch 12/15
844/844 - 6s - loss: 0.0017 - accuracy: 0.9996 - val_loss: 0.0424 -
val_accuracy: 0.9912 - 6s/epoch - 8ms/step
Epoch 13/15
844/844 - 6s - loss: 0.0015 - accuracy: 0.9997 - val_loss: 0.0406 -
val_accuracy: 0.9912 - 6s/epoch - 8ms/step
Epoch 14/15
844/844 - 6s - loss: 7.8269e-04 - accuracy: 0.9999 - val_loss: 0.0408 -
val_accuracy: 0.9912 - 6s/epoch - 8ms/step
Epoch 15/15
844/844 - 6s - loss: 5.7551e-04 - accuracy: 0.9999 - val_loss: 0.0414 -
val_accuracy: 0.9915 - 6s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
```

```
[[ 975  0  0  0  0  0  2  1  2  0]
 [ 1 1128  1  2  0  1  1  1  0  0]
 [ 2  0 1024  0  1  0  0  5  0  0]
 [ 1  0  1 1000  0  5  0  0  3  0]
 [ 0  0  0  0  975  0  1  1  0  5]
 [ 2  0  1  3  0  885  1  0  0  0]
 [ 5  2  0  0  0  3  946  0  1  1]
 [ 0  2  2  0  0  0  0 1022  0  2]
 [ 2  0  0  1  0  1  0  2  966  2]
 [ 0  0  0  1  6  1  0  3  1  997]]
```

Precision: 0.9918

Recall: 0.9918



Training Model with relu activation, 2 conv_layers, 2 dense layers, 64 batch size, 20 epochs..

Epoch 1/20

844/844 - 7s - loss: 0.1706 - accuracy: 0.9468 - val_loss: 0.0634 - val_accuracy: 0.9792 - 7s/epoch - 9ms/step

Epoch 2/20

844/844 - 6s - loss: 0.0569 - accuracy: 0.9824 - val_loss: 0.0469 - val_accuracy: 0.9870 - 6s/epoch - 8ms/step

```
Epoch 3/20
844/844 - 6s - loss: 0.0393 - accuracy: 0.9876 - val_loss: 0.0413 -
val_accuracy: 0.9882 - 6s/epoch - 8ms/step
Epoch 4/20
844/844 - 6s - loss: 0.0289 - accuracy: 0.9911 - val_loss: 0.0371 -
val_accuracy: 0.9898 - 6s/epoch - 7ms/step
Epoch 5/20
844/844 - 6s - loss: 0.0229 - accuracy: 0.9929 - val_loss: 0.0465 -
val_accuracy: 0.9868 - 6s/epoch - 7ms/step
Epoch 6/20
844/844 - 6s - loss: 0.0174 - accuracy: 0.9945 - val_loss: 0.0324 -
val_accuracy: 0.9915 - 6s/epoch - 8ms/step
Epoch 7/20
844/844 - 6s - loss: 0.0124 - accuracy: 0.9959 - val_loss: 0.0396 -
val_accuracy: 0.9892 - 6s/epoch - 8ms/step
Epoch 8/20
844/844 - 6s - loss: 0.0106 - accuracy: 0.9965 - val_loss: 0.0402 -
val_accuracy: 0.9892 - 6s/epoch - 8ms/step
Epoch 9/20
844/844 - 7s - loss: 0.0090 - accuracy: 0.9969 - val_loss: 0.0397 -
val_accuracy: 0.9903 - 7s/epoch - 8ms/step
Epoch 10/20
844/844 - 6s - loss: 0.0052 - accuracy: 0.9984 - val_loss: 0.0355 -
val_accuracy: 0.9920 - 6s/epoch - 8ms/step
Epoch 11/20
844/844 - 6s - loss: 0.0051 - accuracy: 0.9982 - val_loss: 0.0398 -
val_accuracy: 0.9910 - 6s/epoch - 8ms/step
Epoch 12/20
844/844 - 6s - loss: 0.0037 - accuracy: 0.9989 - val_loss: 0.0344 -
val_accuracy: 0.9930 - 6s/epoch - 8ms/step
Epoch 13/20
844/844 - 6s - loss: 0.0023 - accuracy: 0.9995 - val_loss: 0.0370 -
val_accuracy: 0.9930 - 6s/epoch - 8ms/step
Epoch 14/20
844/844 - 6s - loss: 0.0021 - accuracy: 0.9993 - val_loss: 0.0416 -
val_accuracy: 0.9908 - 6s/epoch - 8ms/step
Epoch 15/20
844/844 - 6s - loss: 0.0010 - accuracy: 0.9998 - val_loss: 0.0405 -
val_accuracy: 0.9927 - 6s/epoch - 8ms/step
Epoch 16/20
844/844 - 6s - loss: 4.0548e-04 - accuracy: 1.0000 - val_loss: 0.0407 -
val_accuracy: 0.9918 - 6s/epoch - 8ms/step
Epoch 17/20
844/844 - 6s - loss: 5.4724e-04 - accuracy: 0.9999 - val_loss: 0.0415 -
val_accuracy: 0.9928 - 6s/epoch - 7ms/step
Epoch 18/20
844/844 - 6s - loss: 2.0622e-04 - accuracy: 1.0000 - val_loss: 0.0413 -
val_accuracy: 0.9920 - 6s/epoch - 8ms/step
Epoch 19/20
```

```
844/844 - 7s - loss: 1.6659e-04 - accuracy: 1.0000 - val_loss: 0.0422  
- val_accuracy: 0.9925 - 7s/epoch - 8ms/step
```

Epoch 20/20

```
844/844 - 6s - loss: 1.2947e-04 - accuracy: 1.0000 - val_loss: 0.0427  
- val_accuracy: 0.9925 - 6s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

Results for activation function: relu

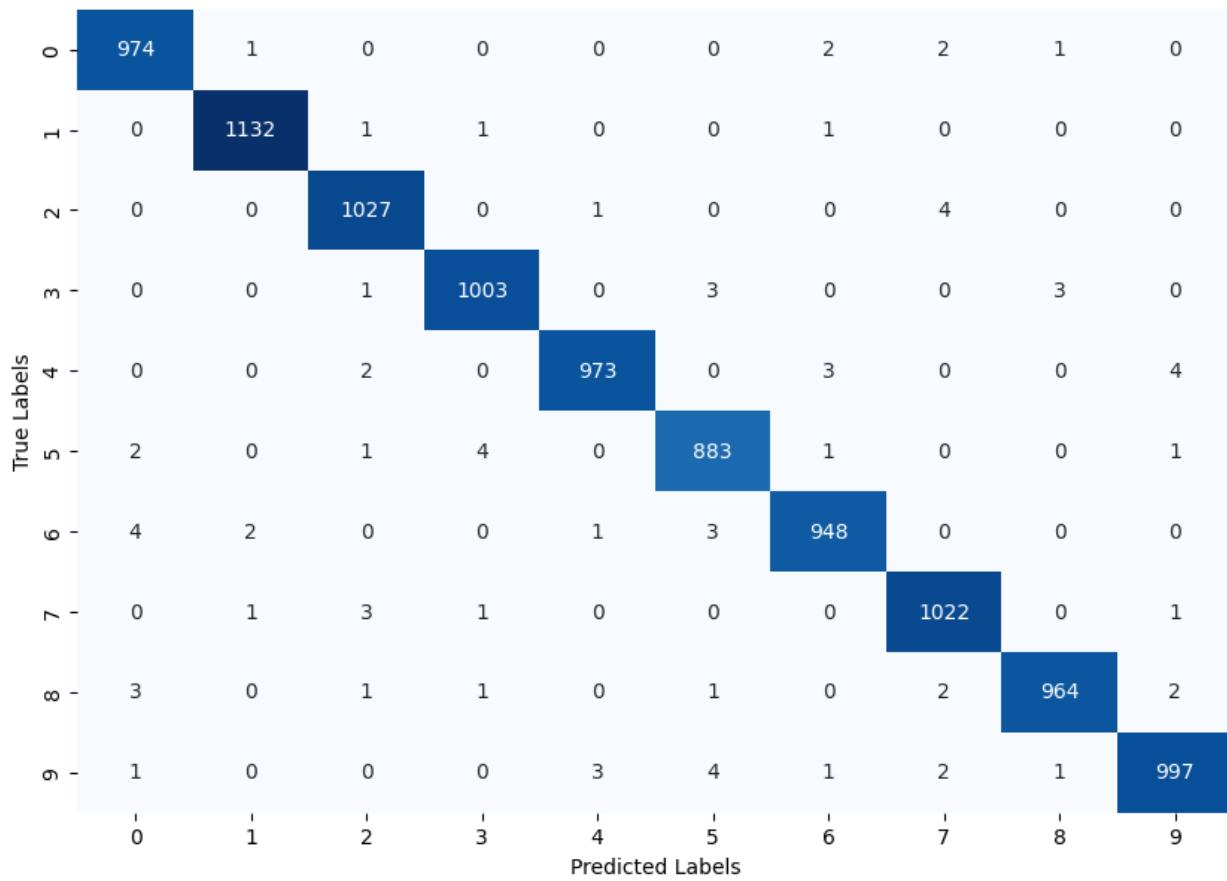
Confusion Matrix:

```
[[ 974   1   0   0   0   0   2   2   1   0 ]  
[  0 1132   1   1   0   0   1   0   0   0 ]  
[  0   0 1027   0   1   0   0   4   0   0 ]  
[  0   0   1 1003   0   3   0   0   3   0 ]  
[  0   0   2   0 973   0   3   0   0   4 ]  
[  2   0   1   4   0 883   1   0   0   1 ]  
[  4   2   0   0   1   3 948   0   0   0 ]  
[  0   1   3   1   0   0   0 1022   0   1 ]  
[  3   0   1   1   0   1   0   2 964   2 ]  
[  1   0   0   0   3   4   1   2   1 997 ]]
```

Precision: 0.9923

Recall: 0.9923

Confusion Matrix for relu Activation



```
Training Model with relu activation, 2 conv_layers, 2 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 5s - loss: 0.2182 - accuracy: 0.9328 - val_loss: 0.0786 -
val_accuracy: 0.9747 - 5s/epoch - 12ms/step
Epoch 2/5
422/422 - 5s - loss: 0.0666 - accuracy: 0.9794 - val_loss: 0.0515 -
val_accuracy: 0.9852 - 5s/epoch - 11ms/step
Epoch 3/5
422/422 - 5s - loss: 0.0485 - accuracy: 0.9851 - val_loss: 0.0463 -
val_accuracy: 0.9877 - 5s/epoch - 11ms/step
Epoch 4/5
422/422 - 5s - loss: 0.0364 - accuracy: 0.9886 - val_loss: 0.0426 -
val_accuracy: 0.9883 - 5s/epoch - 11ms/step
Epoch 5/5
422/422 - 5s - loss: 0.0293 - accuracy: 0.9907 - val_loss: 0.0413 -
val_accuracy: 0.9892 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 971    0    0    0    1    3    2    2    1    0]
 [  0 1130    1    0    0    1    1    2    0    0]
 [  2    1 1021    0    1    0    0    7    0    0]
 [  0    0    2  999    0    6    0    2    1    0]
 [  0    0    0    0  976    0    2    2    0    2]
 [  1    0    0    5    0  882    2    1    1    0]
 [  1    1    0    0    3    5  948    0    0    0]
 [  0    0    3    1    0    0    0 1024    0    0]
 [  6    1    4    3    1    3    2    3  950    1]
 [  0    1    1    2    5    3    0    8    0  989]]
```

Precision: 0.9891
Recall: 0.9890

Confusion Matrix for relu Activation										
	0	1	2	3	4	5	6	7	8	9
0	971	0	0	0	1	3	2	2	1	0
1	0	1130	1	0	0	1	1	2	0	0
2	2	1	1021	0	1	0	0	7	0	0
3	0	0	2	999	0	6	0	2	1	0
4	0	0	0	0	976	0	2	2	0	2
5	1	0	0	5	0	882	2	1	1	0
6	1	1	0	0	3	5	948	0	0	0
7	0	0	3	1	0	0	0	1024	0	0
8	6	1	4	3	1	3	2	3	950	1
9	0	1	1	2	5	3	0	8	0	989
	0	1	2	3	4	5	6	7	8	9

```

Training Model with relu activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 5s - loss: 0.2273 - accuracy: 0.9287 - val_loss: 0.0831 -
val_accuracy: 0.9743 - 5s/epoch - 12ms/step
Epoch 2/15
422/422 - 4s - loss: 0.0674 - accuracy: 0.9786 - val_loss: 0.0534 -
val_accuracy: 0.9850 - 4s/epoch - 11ms/step
Epoch 3/15
422/422 - 4s - loss: 0.0483 - accuracy: 0.9849 - val_loss: 0.0432 -
val_accuracy: 0.9870 - 4s/epoch - 11ms/step
Epoch 4/15
422/422 - 5s - loss: 0.0368 - accuracy: 0.9881 - val_loss: 0.0413 -
val_accuracy: 0.9863 - 5s/epoch - 11ms/step
Epoch 5/15
422/422 - 5s - loss: 0.0274 - accuracy: 0.9913 - val_loss: 0.0414 -
val_accuracy: 0.9882 - 5s/epoch - 11ms/step
Epoch 6/15
422/422 - 5s - loss: 0.0227 - accuracy: 0.9929 - val_loss: 0.0487 -
val_accuracy: 0.9863 - 5s/epoch - 11ms/step
Epoch 7/15

```

```
422/422 - 4s - loss: 0.0191 - accuracy: 0.9940 - val_loss: 0.0436 -  
val_accuracy: 0.9877 - 4s/epoch - 10ms/step  
Epoch 8/15  
422/422 - 4s - loss: 0.0149 - accuracy: 0.9956 - val_loss: 0.0338 -  
val_accuracy: 0.9915 - 4s/epoch - 10ms/step  
Epoch 9/15  
422/422 - 5s - loss: 0.0120 - accuracy: 0.9966 - val_loss: 0.0349 -  
val_accuracy: 0.9910 - 5s/epoch - 11ms/step  
Epoch 10/15  
422/422 - 5s - loss: 0.0093 - accuracy: 0.9969 - val_loss: 0.0389 -  
val_accuracy: 0.9887 - 5s/epoch - 11ms/step  
Epoch 11/15  
422/422 - 5s - loss: 0.0070 - accuracy: 0.9984 - val_loss: 0.0344 -  
val_accuracy: 0.9913 - 5s/epoch - 11ms/step  
Epoch 12/15  
422/422 - 5s - loss: 0.0058 - accuracy: 0.9984 - val_loss: 0.0365 -  
val_accuracy: 0.9922 - 5s/epoch - 11ms/step  
Epoch 13/15  
422/422 - 4s - loss: 0.0044 - accuracy: 0.9988 - val_loss: 0.0423 -  
val_accuracy: 0.9893 - 4s/epoch - 11ms/step  
Epoch 14/15  
422/422 - 5s - loss: 0.0035 - accuracy: 0.9993 - val_loss: 0.0341 -  
val_accuracy: 0.9918 - 5s/epoch - 11ms/step  
Epoch 15/15  
422/422 - 5s - loss: 0.0030 - accuracy: 0.9993 - val_loss: 0.0355 -  
val_accuracy: 0.9918 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 973  0  1  0  1  1  1  1  2  0]  
[ 0 1127  0  2  0  0  2  1  3  0]  
[ 0  2 1021  0  1  0  0  5  3  0]  
[ 0  0  0 1000  0  5  0  0  3  2]  
[ 0  0  1  0 975  0  1  0  0  5]  
[ 0  0  0  5  0 886  1  0  0  0]  
[ 3  1  0  0  1  3 947  0  3  0]  
[ 0  1  3  0  0  0  0 1018  1  5]  
[ 2  0  2  2  0  1  0  1 963  3]  
[ 0  0  0  3  7  5  0  0  0 994]]  
Precision: 0.9904  
Recall: 0.9904
```

Confusion Matrix for relu Activation										
	0	1	2	3	4	5	6	7	8	9
0	973	0	1	0	1	1	1	1	2	0
1	0	1127	0	2	0	0	2	1	3	0
2	0	2	1021	0	1	0	0	5	3	0
3	0	0	0	1000	0	5	0	0	3	2
4	0	0	1	0	975	0	1	0	0	5
5	0	0	0	5	0	886	1	0	0	0
6	3	1	0	0	1	3	947	0	3	0
7	0	1	3	0	0	0	0	1018	1	5
8	2	0	2	2	0	1	0	1	963	3
9	0	0	0	3	7	5	0	0	0	994
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```

Training Model with relu activation, 2 conv_layers, 2 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 5s - loss: 0.2181 - accuracy: 0.9314 - val_loss: 0.0643 -
val_accuracy: 0.9812 - 5s/epoch - 13ms/step
Epoch 2/20
422/422 - 5s - loss: 0.0683 - accuracy: 0.9792 - val_loss: 0.0722 -
val_accuracy: 0.9795 - 5s/epoch - 11ms/step
Epoch 3/20
422/422 - 5s - loss: 0.0491 - accuracy: 0.9842 - val_loss: 0.0486 -
val_accuracy: 0.9858 - 5s/epoch - 11ms/step
Epoch 4/20
422/422 - 5s - loss: 0.0372 - accuracy: 0.9882 - val_loss: 0.0408 -
val_accuracy: 0.9892 - 5s/epoch - 11ms/step
Epoch 5/20
422/422 - 5s - loss: 0.0290 - accuracy: 0.9908 - val_loss: 0.0468 -
val_accuracy: 0.9870 - 5s/epoch - 11ms/step
Epoch 6/20
422/422 - 4s - loss: 0.0244 - accuracy: 0.9922 - val_loss: 0.0426 -
val_accuracy: 0.9895 - 4s/epoch - 11ms/step
Epoch 7/20

```

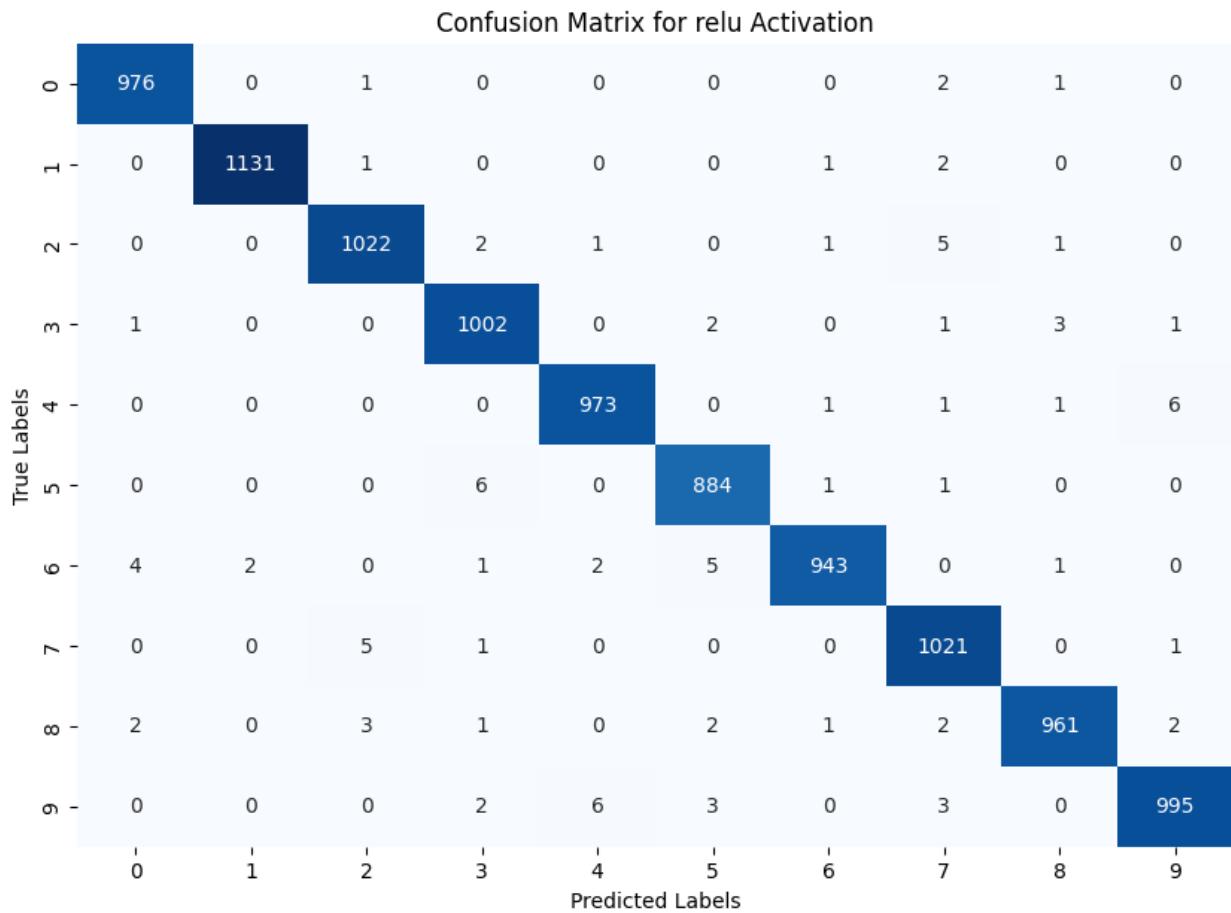
```

422/422 - 4s - loss: 0.0197 - accuracy: 0.9933 - val_loss: 0.0436 -
val_accuracy: 0.9897 - 4s/epoch - 11ms/step
Epoch 8/20
422/422 - 5s - loss: 0.0161 - accuracy: 0.9951 - val_loss: 0.0483 -
val_accuracy: 0.9882 - 5s/epoch - 11ms/step
Epoch 9/20
422/422 - 4s - loss: 0.0130 - accuracy: 0.9963 - val_loss: 0.0521 -
val_accuracy: 0.9867 - 4s/epoch - 10ms/step
Epoch 10/20
422/422 - 4s - loss: 0.0112 - accuracy: 0.9967 - val_loss: 0.0416 -
val_accuracy: 0.9893 - 4s/epoch - 10ms/step
Epoch 11/20
422/422 - 5s - loss: 0.0075 - accuracy: 0.9979 - val_loss: 0.0437 -
val_accuracy: 0.9907 - 5s/epoch - 11ms/step
Epoch 12/20
422/422 - 5s - loss: 0.0067 - accuracy: 0.9981 - val_loss: 0.0462 -
val_accuracy: 0.9885 - 5s/epoch - 11ms/step
Epoch 13/20
422/422 - 5s - loss: 0.0053 - accuracy: 0.9985 - val_loss: 0.0389 -
val_accuracy: 0.9913 - 5s/epoch - 11ms/step
Epoch 14/20
422/422 - 5s - loss: 0.0047 - accuracy: 0.9986 - val_loss: 0.0454 -
val_accuracy: 0.9898 - 5s/epoch - 11ms/step
Epoch 15/20
422/422 - 4s - loss: 0.0053 - accuracy: 0.9985 - val_loss: 0.0453 -
val_accuracy: 0.9893 - 4s/epoch - 11ms/step
Epoch 16/20
422/422 - 5s - loss: 0.0027 - accuracy: 0.9993 - val_loss: 0.0436 -
val_accuracy: 0.9902 - 5s/epoch - 11ms/step
Epoch 17/20
422/422 - 5s - loss: 0.0021 - accuracy: 0.9995 - val_loss: 0.0437 -
val_accuracy: 0.9903 - 5s/epoch - 11ms/step
Epoch 18/20
422/422 - 5s - loss: 0.0016 - accuracy: 0.9998 - val_loss: 0.0516 -
val_accuracy: 0.9897 - 5s/epoch - 11ms/step
Epoch 19/20
422/422 - 5s - loss: 0.0014 - accuracy: 0.9998 - val_loss: 0.0444 -
val_accuracy: 0.9920 - 5s/epoch - 11ms/step
Epoch 20/20
422/422 - 5s - loss: 8.6730e-04 - accuracy: 0.9999 - val_loss: 0.0480
- val_accuracy: 0.9913 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 976   0   1   0   0   0   0   2   1   0]
 [  0 1131   1   0   0   0   1   2   0   0]
 [  0   0 1022   2   1   0   1   5   1   0]
 [  1   0   0 1002   0   2   0   1   3   1]
 [  0   0   0   0  973   0   1   1   1   6]]
```

```
[ 0  0  0  6  0 884  1  1  0  0]
[ 4  2  0  1  2  5 943  0  1  0]
[ 0  0  5  1  0  0  0 1021  0  1]
[ 2  0  3  1  0  2  1  2 961  2]
[ 0  0  0  2  6  3  0  3  0 995]]
```

Precision: 0.9908

Recall: 0.9908



Training Model with relu activation, 2 conv_layers, 2 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 4s - loss: 0.3304 - accuracy: 0.8962 - val_loss: 0.0891 - val_accuracy: 0.9743 - 4s/epoch - 20ms/step

Epoch 2/5

211/211 - 4s - loss: 0.0929 - accuracy: 0.9714 - val_loss: 0.0738 - val_accuracy: 0.9798 - 4s/epoch - 17ms/step

Epoch 3/5

211/211 - 4s - loss: 0.0635 - accuracy: 0.9799 - val_loss: 0.0621 - val_accuracy: 0.9825 - 4s/epoch - 17ms/step

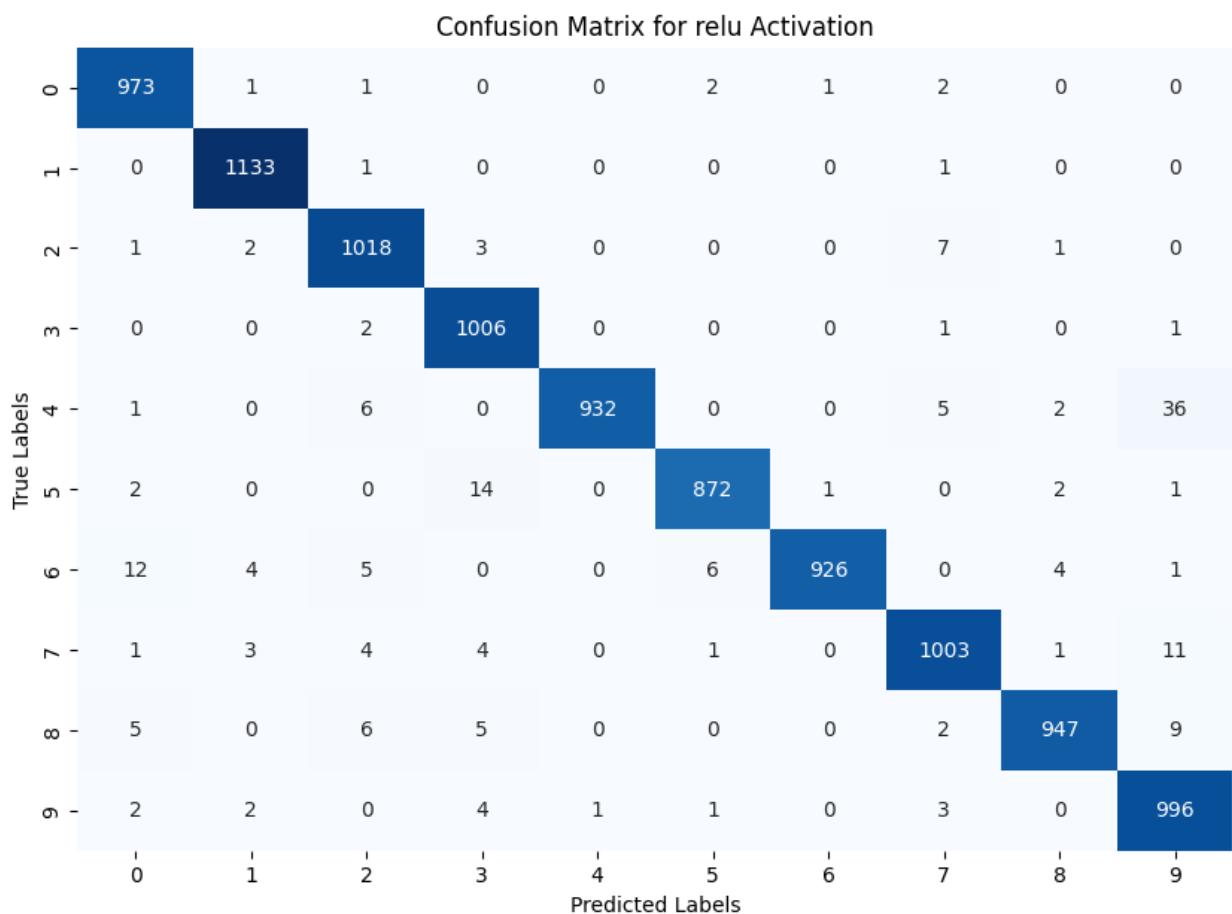
Epoch 4/5

211/211 - 3s - loss: 0.0512 - accuracy: 0.9840 - val_loss: 0.0525 -

```

val_accuracy: 0.9867 - 3s/epoch - 16ms/step
Epoch 5/5
211/211 - 4s - loss: 0.0438 - accuracy: 0.9864 - val_loss: 0.0611 -
val_accuracy: 0.9835 - 4s/epoch - 17ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 973   1   1   0   0   2   1   2   0   0]
 [  0 1133   1   0   0   0   0   1   0   0]
 [  1   2 1018   3   0   0   0   7   1   0]
 [  0   0   2 1006   0   0   0   1   0   1]
 [  1   0   6   0 932   0   0   5   2 36]
 [  2   0   0  14   0 872   1   0   2   1]
 [ 12   4   5   0   0   6 926   0   4   1]
 [  1   3   4   4   0   1   0 1003   1 11]
 [  5   0   6   5   0   0   0   2 947   9]
 [  2   2   0   4   1   1   0   3   0 996]]
Precision: 0.9810
Recall: 0.9806

```



Training Model with relu activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..

Epoch 1/15
211/211 - 4s - loss: 0.3060 - accuracy: 0.9029 - val_loss: 0.0972 -
val_accuracy: 0.9693 - 4s/epoch - 20ms/step

Epoch 2/15
211/211 - 3s - loss: 0.0886 - accuracy: 0.9726 - val_loss: 0.0648 -
val_accuracy: 0.9815 - 3s/epoch - 16ms/step

Epoch 3/15
211/211 - 3s - loss: 0.0636 - accuracy: 0.9799 - val_loss: 0.0616 -
val_accuracy: 0.9822 - 3s/epoch - 17ms/step

Epoch 4/15
211/211 - 3s - loss: 0.0506 - accuracy: 0.9839 - val_loss: 0.0492 -
val_accuracy: 0.9860 - 3s/epoch - 16ms/step

Epoch 5/15
211/211 - 4s - loss: 0.0417 - accuracy: 0.9871 - val_loss: 0.0485 -
val_accuracy: 0.9868 - 4s/epoch - 17ms/step

Epoch 6/15
211/211 - 4s - loss: 0.0362 - accuracy: 0.9885 - val_loss: 0.0512 -
val_accuracy: 0.9847 - 4s/epoch - 17ms/step

Epoch 7/15
211/211 - 4s - loss: 0.0307 - accuracy: 0.9908 - val_loss: 0.0466 -
val_accuracy: 0.9867 - 4s/epoch - 17ms/step

Epoch 8/15
211/211 - 4s - loss: 0.0275 - accuracy: 0.9916 - val_loss: 0.0500 -
val_accuracy: 0.9862 - 4s/epoch - 17ms/step

Epoch 9/15
211/211 - 4s - loss: 0.0239 - accuracy: 0.9925 - val_loss: 0.0433 -
val_accuracy: 0.9873 - 4s/epoch - 17ms/step

Epoch 10/15
211/211 - 4s - loss: 0.0205 - accuracy: 0.9936 - val_loss: 0.0390 -
val_accuracy: 0.9892 - 4s/epoch - 17ms/step

Epoch 11/15
211/211 - 4s - loss: 0.0177 - accuracy: 0.9947 - val_loss: 0.0421 -
val_accuracy: 0.9890 - 4s/epoch - 17ms/step

Epoch 12/15
211/211 - 4s - loss: 0.0161 - accuracy: 0.9951 - val_loss: 0.0364 -
val_accuracy: 0.9893 - 4s/epoch - 17ms/step

Epoch 13/15
211/211 - 4s - loss: 0.0130 - accuracy: 0.9963 - val_loss: 0.0400 -
val_accuracy: 0.9893 - 4s/epoch - 17ms/step

Epoch 14/15
211/211 - 4s - loss: 0.0128 - accuracy: 0.9962 - val_loss: 0.0391 -
val_accuracy: 0.9895 - 4s/epoch - 17ms/step

Epoch 15/15
211/211 - 4s - loss: 0.0099 - accuracy: 0.9973 - val_loss: 0.0412 -
val_accuracy: 0.9888 - 4s/epoch - 17ms/step

313/313 [=====] - 1s 2ms/step

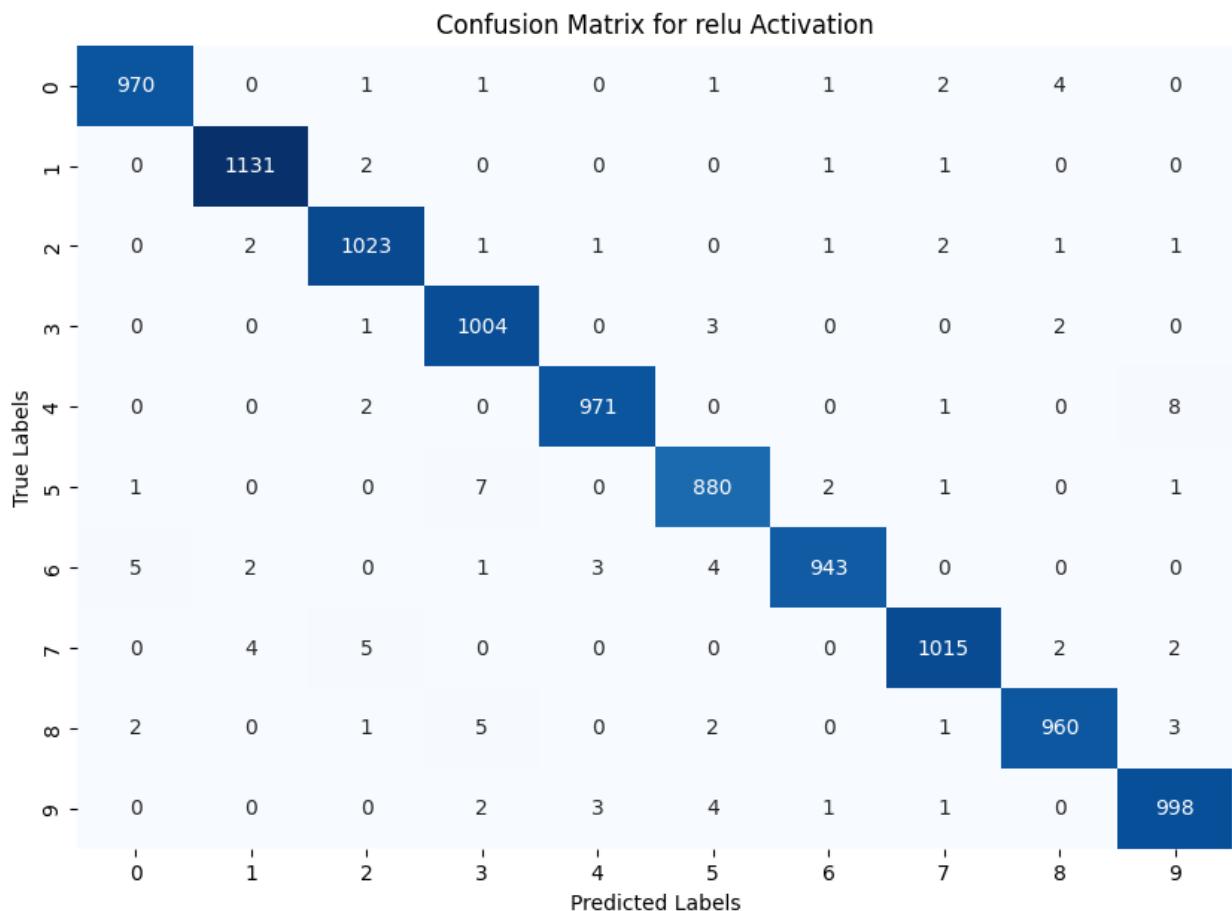
Results for activation function: relu

Confusion Matrix:

```
[[ 970  0   1   1   0   1   1   2   4   0]
 [ 0 1131  2   0   0   0   1   1   0   0]
 [ 0 2 1023  1   1   0   1   2   1   1]
 [ 0 0 1 1004  0   3   0   0   2   0]
 [ 0 0 2 0 971  0   0   1   0   8]
 [ 1 0 0 7 0 880  2   1   0   1]
 [ 5 2 0 1 3 4 943  0   0   0]
 [ 0 4 5 0 0 0 0 1015  2   2]
 [ 2 0 1 5 0 2 0 1 960  3]
 [ 0 0 0 2 3 4 1 1 0 998]]
```

Precision: 0.9895

Recall: 0.9895



Training Model with relu activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..

Epoch 1/20

211/211 - 5s - loss: 0.2920 - accuracy: 0.9085 - val_loss: 0.0979 -
val_accuracy: 0.9710 - 5s/epoch - 23ms/step

Epoch 2/20

211/211 - 4s - loss: 0.0897 - accuracy: 0.9726 - val_loss: 0.0715 -
val_accuracy: 0.9795 - 4s/epoch - 17ms/step

```
Epoch 3/20
211/211 - 4s - loss: 0.0654 - accuracy: 0.9796 - val_loss: 0.0583 -
val_accuracy: 0.9843 - 4s/epoch - 18ms/step
Epoch 4/20
211/211 - 4s - loss: 0.0514 - accuracy: 0.9834 - val_loss: 0.0536 -
val_accuracy: 0.9837 - 4s/epoch - 17ms/step
Epoch 5/20
211/211 - 4s - loss: 0.0436 - accuracy: 0.9863 - val_loss: 0.0477 -
val_accuracy: 0.9878 - 4s/epoch - 17ms/step
Epoch 6/20
211/211 - 4s - loss: 0.0371 - accuracy: 0.9885 - val_loss: 0.0503 -
val_accuracy: 0.9867 - 4s/epoch - 18ms/step
Epoch 7/20
211/211 - 4s - loss: 0.0307 - accuracy: 0.9898 - val_loss: 0.0437 -
val_accuracy: 0.9882 - 4s/epoch - 17ms/step
Epoch 8/20
211/211 - 4s - loss: 0.0265 - accuracy: 0.9918 - val_loss: 0.0533 -
val_accuracy: 0.9858 - 4s/epoch - 17ms/step
Epoch 9/20
211/211 - 3s - loss: 0.0235 - accuracy: 0.9926 - val_loss: 0.0463 -
val_accuracy: 0.9883 - 3s/epoch - 16ms/step
Epoch 10/20
211/211 - 4s - loss: 0.0221 - accuracy: 0.9928 - val_loss: 0.0427 -
val_accuracy: 0.9892 - 4s/epoch - 17ms/step
Epoch 11/20
211/211 - 4s - loss: 0.0177 - accuracy: 0.9945 - val_loss: 0.0402 -
val_accuracy: 0.9885 - 4s/epoch - 17ms/step
Epoch 12/20
211/211 - 4s - loss: 0.0148 - accuracy: 0.9954 - val_loss: 0.0401 -
val_accuracy: 0.9900 - 4s/epoch - 17ms/step
Epoch 13/20
211/211 - 4s - loss: 0.0130 - accuracy: 0.9959 - val_loss: 0.0454 -
val_accuracy: 0.9882 - 4s/epoch - 18ms/step
Epoch 14/20
211/211 - 4s - loss: 0.0106 - accuracy: 0.9968 - val_loss: 0.0407 -
val_accuracy: 0.9903 - 4s/epoch - 17ms/step
Epoch 15/20
211/211 - 4s - loss: 0.0095 - accuracy: 0.9973 - val_loss: 0.0455 -
val_accuracy: 0.9885 - 4s/epoch - 18ms/step
Epoch 16/20
211/211 - 4s - loss: 0.0085 - accuracy: 0.9979 - val_loss: 0.0398 -
val_accuracy: 0.9895 - 4s/epoch - 18ms/step
Epoch 17/20
211/211 - 4s - loss: 0.0078 - accuracy: 0.9977 - val_loss: 0.0437 -
val_accuracy: 0.9893 - 4s/epoch - 17ms/step
Epoch 18/20
211/211 - 4s - loss: 0.0069 - accuracy: 0.9980 - val_loss: 0.0402 -
val_accuracy: 0.9903 - 4s/epoch - 18ms/step
Epoch 19/20
```

```
211/211 - 4s - loss: 0.0054 - accuracy: 0.9986 - val_loss: 0.0419 -  
val_accuracy: 0.9893 - 4s/epoch - 17ms/step
```

```
Epoch 20/20
```

```
211/211 - 4s - loss: 0.0039 - accuracy: 0.9992 - val_loss: 0.0500 -  
val_accuracy: 0.9892 - 4s/epoch - 17ms/step
```

```
313/313 [=====] - 1s 3ms/step
```

```
Results for activation function: relu
```

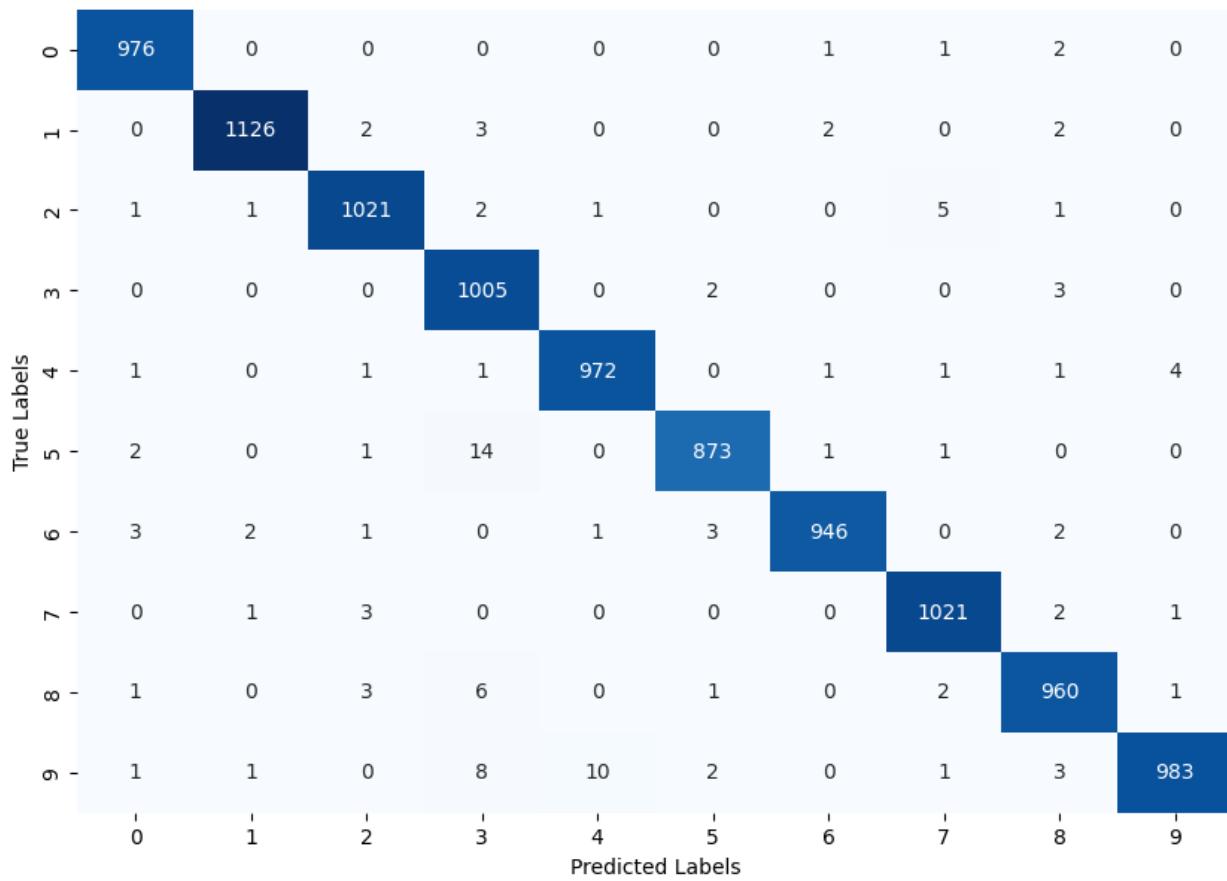
```
Confusion Matrix:
```

```
[[ 976   0   0   0   0   0   1   1   2   0]
 [ 0 1126   2   3   0   0   2   0   2   0]
 [ 1   1 1021   2   1   0   0   5   1   0]
 [ 0   0   0 1005   0   2   0   0   3   0]
 [ 1   0   1   1 972   0   1   1   1   4]
 [ 2   0   1   14   0 873   1   1   0   0]
 [ 3   2   1   0   1   3 946   0   2   0]
 [ 0   1   3   0   0   0   0 1021   2   1]
 [ 1   0   3   6   0   1   0   2 960   1]
 [ 1   1   0   8 10  2   0   1   3 983]]
```

```
Precision: 0.9884
```

```
Recall: 0.9883
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 3 conv_layers, 3 dense layers, 64
batch size, 5 epochs..
Epoch 1/5
844/844 - 7s - loss: 0.3824 - accuracy: 0.8753 - val_loss: 0.1601 -
val_accuracy: 0.9577 - 7s/epoch - 9ms/step
Epoch 2/5
844/844 - 6s - loss: 0.1189 - accuracy: 0.9647 - val_loss: 0.0988 -
val_accuracy: 0.9707 - 6s/epoch - 8ms/step
Epoch 3/5
844/844 - 6s - loss: 0.0901 - accuracy: 0.9726 - val_loss: 0.0797 -
val_accuracy: 0.9777 - 6s/epoch - 7ms/step
Epoch 4/5
844/844 - 6s - loss: 0.0749 - accuracy: 0.9764 - val_loss: 0.0647 -
val_accuracy: 0.9803 - 6s/epoch - 7ms/step
Epoch 5/5
844/844 - 6s - loss: 0.0614 - accuracy: 0.9814 - val_loss: 0.0592 -
val_accuracy: 0.9825 - 6s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 970    0    1    0    0    0    3    3    1    2]
 [  0 1125    2    2    0    1    0    4    1    0]
 [  6    0 999    3    1    0    2   15    5    1]
 [  0    0    3 999    0    2    0    4    1    1]
 [  1    1    0    0 965    0    4    1    1    9]
 [  7    0    0    5    0 868    5    1    5    1]
 [  2    2    1    0    1    3 946    0    3    0]
 [  1    1    8    5    0    1    0 1008    2    2]
 [  1    0    4    2    4    1    2    1  949   10]
 [  1    0    0    0    2    7    1    6    6 986]]
```

Precision: 0.9815
Recall: 0.9815

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9	
0	970	0	1	0	0	0	3	3	1	2	
1	0	1125	2	2	0	1	0	4	1	0	
2	6	0	999	3	1	0	2	15	5	1	
3	0	0	3	999	0	2	0	4	1	1	
4	1	1	0	0	965	0	4	1	1	9	
5	7	0	0	5	0	868	5	1	5	1	
6	2	2	1	0	1	3	946	0	3	0	
7	1	1	8	5	0	1	0	1008	2	2	
8	1	0	4	2	4	1	2	1	949	10	
9	1	0	0	0	2	7	1	6	6	986	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

Training Model with relu activation, 3 conv_layers, 3 dense layers, 64 batch size, 15 epochs..

Epoch 1/15

844/844 - 7s - loss: 0.3544 - accuracy: 0.8859 - val_loss: 0.1111 - val_accuracy: 0.9673 - 7s/epoch - 8ms/step

Epoch 2/15

844/844 - 6s - loss: 0.1236 - accuracy: 0.9617 - val_loss: 0.0991 - val_accuracy: 0.9707 - 6s/epoch - 7ms/step

Epoch 3/15

844/844 - 6s - loss: 0.0909 - accuracy: 0.9731 - val_loss: 0.0910 - val_accuracy: 0.9697 - 6s/epoch - 7ms/step

Epoch 4/15

844/844 - 6s - loss: 0.0747 - accuracy: 0.9775 - val_loss: 0.0530 - val_accuracy: 0.9840 - 6s/epoch - 7ms/step

Epoch 5/15

844/844 - 6s - loss: 0.0641 - accuracy: 0.9800 - val_loss: 0.0659 - val_accuracy: 0.9793 - 6s/epoch - 7ms/step

Epoch 6/15

844/844 - 6s - loss: 0.0590 - accuracy: 0.9819 - val_loss: 0.0700 - val_accuracy: 0.9795 - 6s/epoch - 7ms/step

Epoch 7/15

```
844/844 - 6s - loss: 0.0493 - accuracy: 0.9849 - val_loss: 0.0537 -  
val_accuracy: 0.9845 - 6s/epoch - 7ms/step  
Epoch 8/15  
844/844 - 6s - loss: 0.0453 - accuracy: 0.9855 - val_loss: 0.0783 -  
val_accuracy: 0.9770 - 6s/epoch - 7ms/step  
Epoch 9/15  
844/844 - 6s - loss: 0.0400 - accuracy: 0.9874 - val_loss: 0.0705 -  
val_accuracy: 0.9792 - 6s/epoch - 7ms/step  
Epoch 10/15  
844/844 - 6s - loss: 0.0367 - accuracy: 0.9885 - val_loss: 0.0651 -  
val_accuracy: 0.9815 - 6s/epoch - 7ms/step  
Epoch 11/15  
844/844 - 6s - loss: 0.0331 - accuracy: 0.9894 - val_loss: 0.0537 -  
val_accuracy: 0.9850 - 6s/epoch - 7ms/step  
Epoch 12/15  
844/844 - 6s - loss: 0.0308 - accuracy: 0.9902 - val_loss: 0.0564 -  
val_accuracy: 0.9848 - 6s/epoch - 7ms/step  
Epoch 13/15  
844/844 - 6s - loss: 0.0287 - accuracy: 0.9907 - val_loss: 0.0662 -  
val_accuracy: 0.9845 - 6s/epoch - 7ms/step  
Epoch 14/15  
844/844 - 6s - loss: 0.0276 - accuracy: 0.9905 - val_loss: 0.0669 -  
val_accuracy: 0.9818 - 6s/epoch - 7ms/step  
Epoch 15/15  
844/844 - 6s - loss: 0.0272 - accuracy: 0.9914 - val_loss: 0.0565 -  
val_accuracy: 0.9845 - 6s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 960    1    4    0    2    3    4    4    1    1]  
[  0  1128    1    2    0    0    1    2    1    0]  
[  2    2  1014    2    1    0    2    8    1    0]  
[  0    0    3 1000    0    7    0    0    0    0]  
[  0    0    1    0   975    0    0    1    0    5]  
[  1    1    1    3    0   882    1    1    2    0]  
[  2    3    1    0    2    8   941    0    1    0]  
[  0    4   14    2    0    1    0 1003    2    2]  
[  0    0    1    2    3    3    0    2   960    3]  
[  2    0    0    0   14    8    0    5    4  976]]  
Precision: 0.9840  
Recall: 0.9839
```

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	960	1	4	0	2	3	4	4	1	1
0	960	1	4	0	2	3	4	4	1	1
1	0	1128	1	2	0	0	1	2	1	0
2	2	2	1014	2	1	0	2	8	1	0
3	0	0	3	1000	0	7	0	0	0	0
4	0	0	1	0	975	0	0	1	0	5
5	1	1	1	3	0	882	1	1	2	0
6	2	3	1	0	2	8	941	0	1	0
7	0	4	14	2	0	1	0	1003	2	2
8	0	0	1	2	3	3	0	2	960	3
9	2	0	0	0	14	8	0	5	4	976
Predicted Labels	0	1	2	3	4	5	6	7	8	9

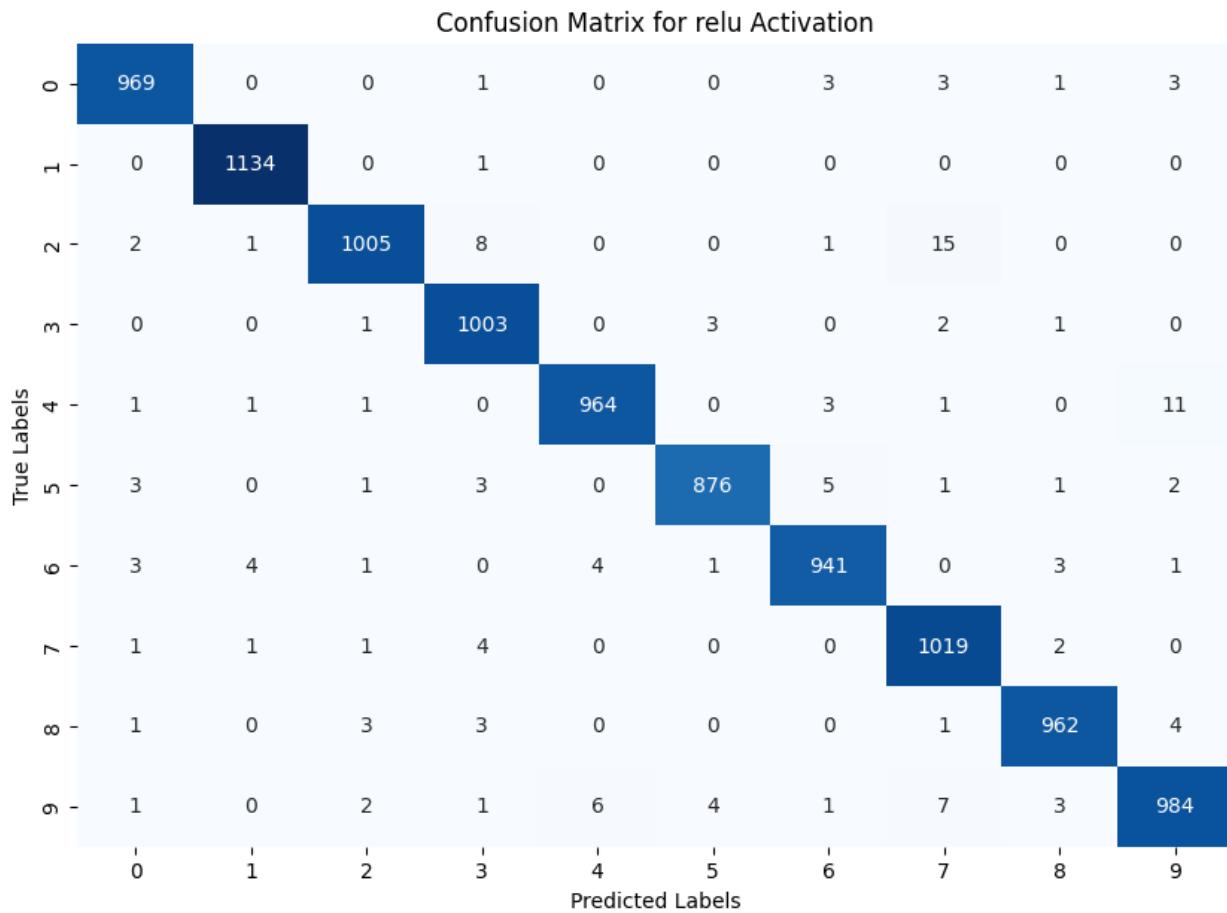
```
Training Model with relu activation, 3 conv_layers, 3 dense layers, 64
batch size, 20 epochs..
Epoch 1/20
844/844 - 7s - loss: 0.3270 - accuracy: 0.8961 - val_loss: 0.1517 -
val_accuracy: 0.9513 - 7s/epoch - 8ms/step
Epoch 2/20
844/844 - 6s - loss: 0.1176 - accuracy: 0.9645 - val_loss: 0.1289 -
val_accuracy: 0.9615 - 6s/epoch - 7ms/step
Epoch 3/20
844/844 - 6s - loss: 0.0875 - accuracy: 0.9735 - val_loss: 0.0729 -
val_accuracy: 0.9768 - 6s/epoch - 7ms/step
Epoch 4/20
844/844 - 6s - loss: 0.0690 - accuracy: 0.9794 - val_loss: 0.0683 -
val_accuracy: 0.9780 - 6s/epoch - 7ms/step
Epoch 5/20
844/844 - 6s - loss: 0.0592 - accuracy: 0.9818 - val_loss: 0.0684 -
val_accuracy: 0.9792 - 6s/epoch - 7ms/step
Epoch 6/20
844/844 - 6s - loss: 0.0508 - accuracy: 0.9843 - val_loss: 0.0499 -
val_accuracy: 0.9852 - 6s/epoch - 7ms/step
Epoch 7/20
```

```
844/844 - 6s - loss: 0.0456 - accuracy: 0.9853 - val_loss: 0.0484 -  
val_accuracy: 0.9857 - 6s/epoch - 7ms/step  
Epoch 8/20  
844/844 - 6s - loss: 0.0412 - accuracy: 0.9866 - val_loss: 0.0534 -  
val_accuracy: 0.9860 - 6s/epoch - 7ms/step  
Epoch 9/20  
844/844 - 6s - loss: 0.0361 - accuracy: 0.9882 - val_loss: 0.0494 -  
val_accuracy: 0.9847 - 6s/epoch - 7ms/step  
Epoch 10/20  
844/844 - 6s - loss: 0.0315 - accuracy: 0.9897 - val_loss: 0.0459 -  
val_accuracy: 0.9867 - 6s/epoch - 7ms/step  
Epoch 11/20  
844/844 - 6s - loss: 0.0289 - accuracy: 0.9909 - val_loss: 0.0480 -  
val_accuracy: 0.9870 - 6s/epoch - 7ms/step  
Epoch 12/20  
844/844 - 6s - loss: 0.0255 - accuracy: 0.9919 - val_loss: 0.0507 -  
val_accuracy: 0.9873 - 6s/epoch - 7ms/step  
Epoch 13/20  
844/844 - 6s - loss: 0.0217 - accuracy: 0.9931 - val_loss: 0.0547 -  
val_accuracy: 0.9863 - 6s/epoch - 7ms/step  
Epoch 14/20  
844/844 - 6s - loss: 0.0217 - accuracy: 0.9928 - val_loss: 0.0499 -  
val_accuracy: 0.9865 - 6s/epoch - 7ms/step  
Epoch 15/20  
844/844 - 6s - loss: 0.0191 - accuracy: 0.9938 - val_loss: 0.0553 -  
val_accuracy: 0.9870 - 6s/epoch - 7ms/step  
Epoch 16/20  
844/844 - 6s - loss: 0.0191 - accuracy: 0.9938 - val_loss: 0.0543 -  
val_accuracy: 0.9867 - 6s/epoch - 7ms/step  
Epoch 17/20  
844/844 - 6s - loss: 0.0153 - accuracy: 0.9949 - val_loss: 0.0609 -  
val_accuracy: 0.9852 - 6s/epoch - 7ms/step  
Epoch 18/20  
844/844 - 6s - loss: 0.0147 - accuracy: 0.9950 - val_loss: 0.0544 -  
val_accuracy: 0.9878 - 6s/epoch - 7ms/step  
Epoch 19/20  
844/844 - 6s - loss: 0.0152 - accuracy: 0.9951 - val_loss: 0.0577 -  
val_accuracy: 0.9865 - 6s/epoch - 7ms/step  
Epoch 20/20  
844/844 - 6s - loss: 0.0158 - accuracy: 0.9946 - val_loss: 0.0482 -  
val_accuracy: 0.9898 - 6s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 969 0 0 1 0 0 3 3 1 3]  
[ 0 1134 0 1 0 0 0 0 0 0]  
[ 2 1 1005 8 0 0 1 15 0 0]  
[ 0 0 1 1003 0 3 0 2 1 0]  
[ 1 1 1 0 964 0 3 1 0 11]]
```

```
[ 3 0 1 3 0 876 5 1 1 2]
[ 3 4 1 0 4 1 941 0 3 1]
[ 1 1 1 4 0 0 0 1019 2 0]
[ 1 0 3 3 0 0 0 1 962 4]
[ 1 0 2 1 6 4 1 7 3 984]]
```

Precision: 0.9857

Recall: 0.9857



Training Model with relu activation, 3 conv_layers, 3 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 5s - loss: 0.4211 - accuracy: 0.8637 - val_loss: 0.2061 - val_accuracy: 0.9410 - 5s/epoch - 13ms/step

Epoch 2/5

422/422 - 4s - loss: 0.1280 - accuracy: 0.9605 - val_loss: 0.1132 - val_accuracy: 0.9662 - 4s/epoch - 11ms/step

Epoch 3/5

422/422 - 4s - loss: 0.0904 - accuracy: 0.9723 - val_loss: 0.0824 - val_accuracy: 0.9765 - 4s/epoch - 10ms/step

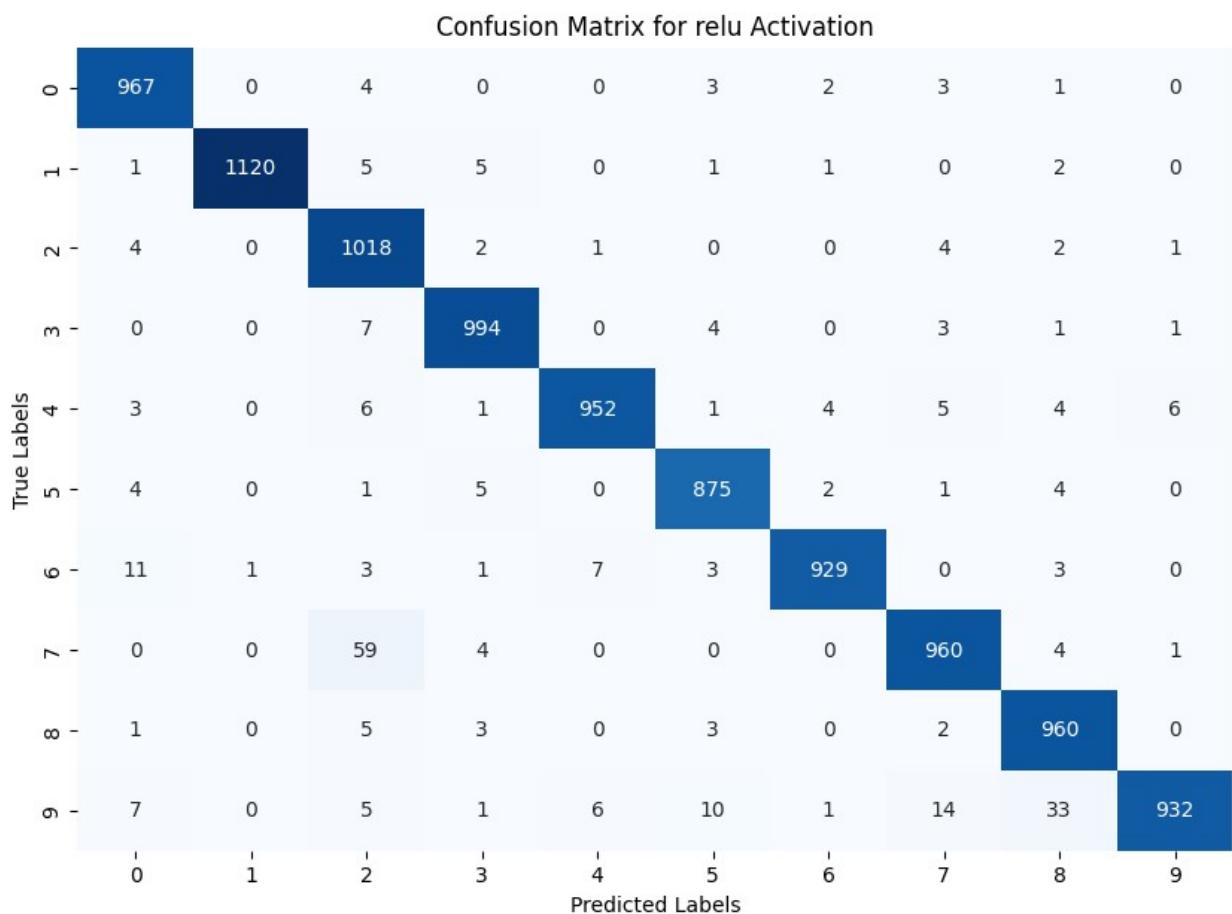
Epoch 4/5

422/422 - 4s - loss: 0.0759 - accuracy: 0.9762 - val_loss: 0.0724 -

```

val_accuracy: 0.9800 - 4s/epoch - 10ms/step
Epoch 5/5
422/422 - 4s - loss: 0.0644 - accuracy: 0.9799 - val_loss: 0.0769 -
val_accuracy: 0.9772 - 4s/epoch - 10ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 967   0   4   0   0   3   2   3   1   0]
 [ 1 1120   5   5   0   1   1   0   2   0]
 [ 4   0 1018   2   1   0   0   4   2   1]
 [ 0   0   7 994   0   4   0   3   1   1]
 [ 3   0   6   1 952   1   4   5   4   6]
 [ 4   0   1   5   0 875   2   1   4   0]
 [ 11   1   3   1   7   3 929   0   3   0]
 [ 0   0   59   4   0   0   0 960   4   1]
 [ 1   0   5   3   0   3   0   2 960   0]
 [ 7   0   5   1   6 10   1 14   33 932]]
Precision: 0.9715
Recall: 0.9707

```



Training Model with relu activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..

Epoch 1/15
422/422 - 5s - loss: 0.4980 - accuracy: 0.8375 - val_loss: 0.1649 -
val_accuracy: 0.9515 - 5s/epoch - 12ms/step

Epoch 2/15
422/422 - 4s - loss: 0.1428 - accuracy: 0.9559 - val_loss: 0.1175 -
val_accuracy: 0.9653 - 4s/epoch - 10ms/step

Epoch 3/15
422/422 - 4s - loss: 0.1041 - accuracy: 0.9675 - val_loss: 0.0926 -
val_accuracy: 0.9728 - 4s/epoch - 10ms/step

Epoch 4/15
422/422 - 4s - loss: 0.0859 - accuracy: 0.9737 - val_loss: 0.0792 -
val_accuracy: 0.9730 - 4s/epoch - 10ms/step

Epoch 5/15
422/422 - 4s - loss: 0.0718 - accuracy: 0.9777 - val_loss: 0.0727 -
val_accuracy: 0.9788 - 4s/epoch - 10ms/step

Epoch 6/15
422/422 - 4s - loss: 0.0640 - accuracy: 0.9792 - val_loss: 0.0617 -
val_accuracy: 0.9820 - 4s/epoch - 10ms/step

Epoch 7/15
422/422 - 4s - loss: 0.0566 - accuracy: 0.9820 - val_loss: 0.0641 -
val_accuracy: 0.9805 - 4s/epoch - 10ms/step

Epoch 8/15
422/422 - 4s - loss: 0.0503 - accuracy: 0.9834 - val_loss: 0.0662 -
val_accuracy: 0.9823 - 4s/epoch - 10ms/step

Epoch 9/15
422/422 - 4s - loss: 0.0465 - accuracy: 0.9853 - val_loss: 0.0536 -
val_accuracy: 0.9852 - 4s/epoch - 11ms/step

Epoch 10/15
422/422 - 4s - loss: 0.0446 - accuracy: 0.9855 - val_loss: 0.0732 -
val_accuracy: 0.9790 - 4s/epoch - 10ms/step

Epoch 11/15
422/422 - 4s - loss: 0.0398 - accuracy: 0.9876 - val_loss: 0.0540 -
val_accuracy: 0.9848 - 4s/epoch - 10ms/step

Epoch 12/15
422/422 - 4s - loss: 0.0343 - accuracy: 0.9889 - val_loss: 0.0572 -
val_accuracy: 0.9842 - 4s/epoch - 10ms/step

Epoch 13/15
422/422 - 4s - loss: 0.0332 - accuracy: 0.9892 - val_loss: 0.0699 -
val_accuracy: 0.9803 - 4s/epoch - 10ms/step

Epoch 14/15
422/422 - 4s - loss: 0.0299 - accuracy: 0.9903 - val_loss: 0.0558 -
val_accuracy: 0.9853 - 4s/epoch - 10ms/step

Epoch 15/15
422/422 - 4s - loss: 0.0285 - accuracy: 0.9908 - val_loss: 0.0686 -
val_accuracy: 0.9823 - 4s/epoch - 10ms/step

313/313 [=====] - 1s 2ms/step

Results for activation function: relu

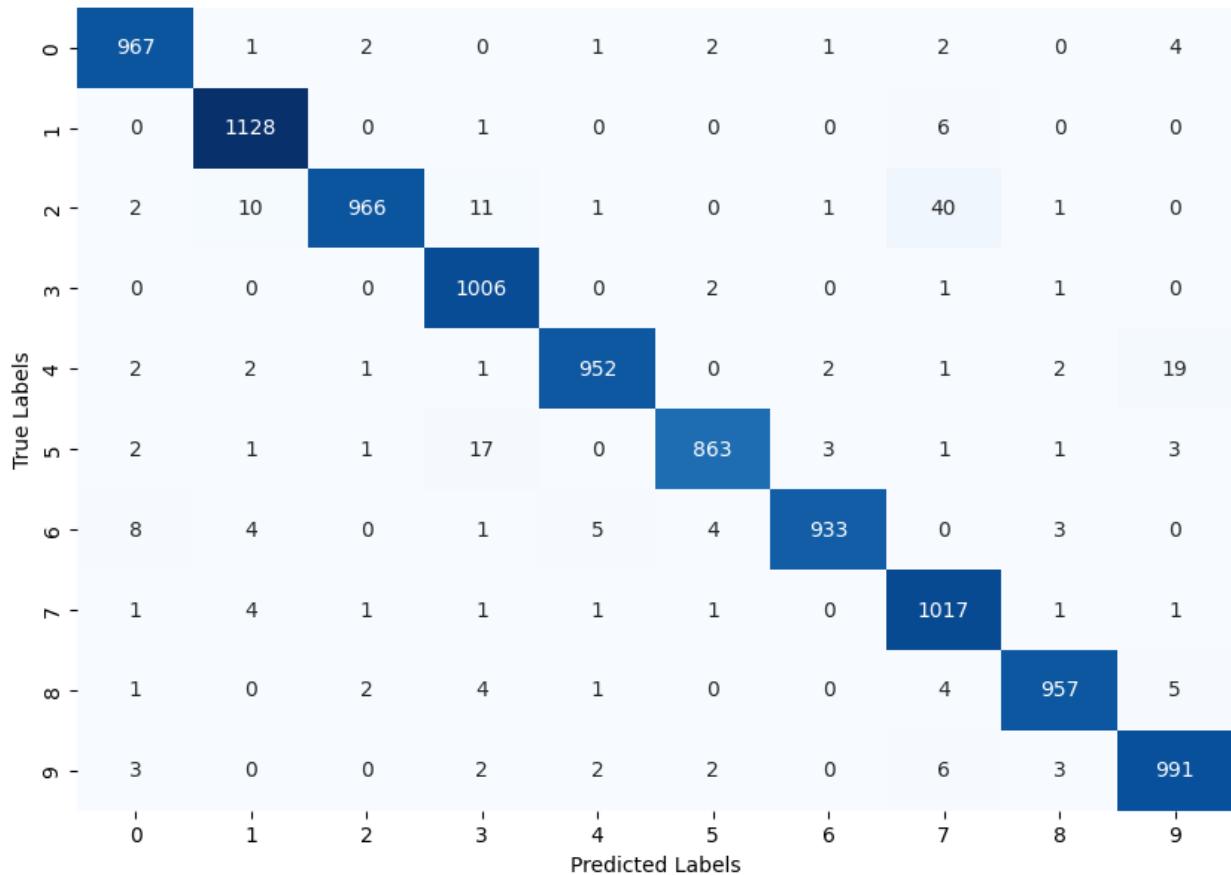
Confusion Matrix:

```
[[ 967  1  2  0  1  2  1  2  0  4]
 [ 0 1128  0  1  0  0  0  6  0  0]
 [ 2 10  966 11  1  0  1  40  1  0]
 [ 0 0  0 1006  0  2  0  1  1  0]
 [ 2 2  1  1  952  0  2  1  2  19]
 [ 2 1  1 17  0  863  3  1  1  3]
 [ 8 4  0  1  5  4  933  0  3  0]
 [ 1 4  1  1  1  1  0 1017  1  1]
 [ 1 0  2  4  1  0  0  4  957  5]
 [ 3 0  0  2  2  2  0  6  3  991]]
```

Precision: 0.9784

Recall: 0.9780

Confusion Matrix for relu Activation



Training Model with relu activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..

Epoch 1/20

422/422 - 6s - loss: 0.4238 - accuracy: 0.8635 - val_loss: 0.1222 -
val_accuracy: 0.9658 - 6s/epoch - 14ms/step

Epoch 2/20

422/422 - 5s - loss: 0.1247 - accuracy: 0.9615 - val_loss: 0.1091 -
val_accuracy: 0.9667 - 5s/epoch - 11ms/step

```
Epoch 3/20
422/422 - 4s - loss: 0.0931 - accuracy: 0.9714 - val_loss: 0.0731 -
val_accuracy: 0.9778 - 4s/epoch - 11ms/step
Epoch 4/20
422/422 - 4s - loss: 0.0753 - accuracy: 0.9772 - val_loss: 0.0605 -
val_accuracy: 0.9810 - 4s/epoch - 10ms/step
Epoch 5/20
422/422 - 4s - loss: 0.0657 - accuracy: 0.9796 - val_loss: 0.0752 -
val_accuracy: 0.9775 - 4s/epoch - 10ms/step
Epoch 6/20
422/422 - 4s - loss: 0.0565 - accuracy: 0.9828 - val_loss: 0.0595 -
val_accuracy: 0.9822 - 4s/epoch - 11ms/step
Epoch 7/20
422/422 - 5s - loss: 0.0492 - accuracy: 0.9848 - val_loss: 0.0685 -
val_accuracy: 0.9808 - 5s/epoch - 11ms/step
Epoch 8/20
422/422 - 5s - loss: 0.0455 - accuracy: 0.9859 - val_loss: 0.0582 -
val_accuracy: 0.9830 - 5s/epoch - 11ms/step
Epoch 9/20
422/422 - 5s - loss: 0.0394 - accuracy: 0.9877 - val_loss: 0.0560 -
val_accuracy: 0.9827 - 5s/epoch - 11ms/step
Epoch 10/20
422/422 - 5s - loss: 0.0367 - accuracy: 0.9880 - val_loss: 0.0730 -
val_accuracy: 0.9775 - 5s/epoch - 11ms/step
Epoch 11/20
422/422 - 5s - loss: 0.0342 - accuracy: 0.9888 - val_loss: 0.0542 -
val_accuracy: 0.9828 - 5s/epoch - 12ms/step
Epoch 12/20
422/422 - 5s - loss: 0.0292 - accuracy: 0.9910 - val_loss: 0.0486 -
val_accuracy: 0.9850 - 5s/epoch - 11ms/step
Epoch 13/20
422/422 - 5s - loss: 0.0263 - accuracy: 0.9915 - val_loss: 0.0474 -
val_accuracy: 0.9860 - 5s/epoch - 11ms/step
Epoch 14/20
422/422 - 5s - loss: 0.0239 - accuracy: 0.9924 - val_loss: 0.0635 -
val_accuracy: 0.9828 - 5s/epoch - 11ms/step
Epoch 15/20
422/422 - 4s - loss: 0.0218 - accuracy: 0.9931 - val_loss: 0.0532 -
val_accuracy: 0.9837 - 4s/epoch - 11ms/step
Epoch 16/20
422/422 - 5s - loss: 0.0214 - accuracy: 0.9928 - val_loss: 0.0555 -
val_accuracy: 0.9837 - 5s/epoch - 11ms/step
Epoch 17/20
422/422 - 5s - loss: 0.0197 - accuracy: 0.9937 - val_loss: 0.0485 -
val_accuracy: 0.9855 - 5s/epoch - 11ms/step
Epoch 18/20
422/422 - 5s - loss: 0.0160 - accuracy: 0.9948 - val_loss: 0.0526 -
val_accuracy: 0.9865 - 5s/epoch - 11ms/step
Epoch 19/20
```

```
422/422 - 5s - loss: 0.0134 - accuracy: 0.9956 - val_loss: 0.0574 -  
val_accuracy: 0.9838 - 5s/epoch - 11ms/step
```

```
Epoch 20/20
```

```
422/422 - 5s - loss: 0.0150 - accuracy: 0.9950 - val_loss: 0.0561 -  
val_accuracy: 0.9852 - 5s/epoch - 11ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

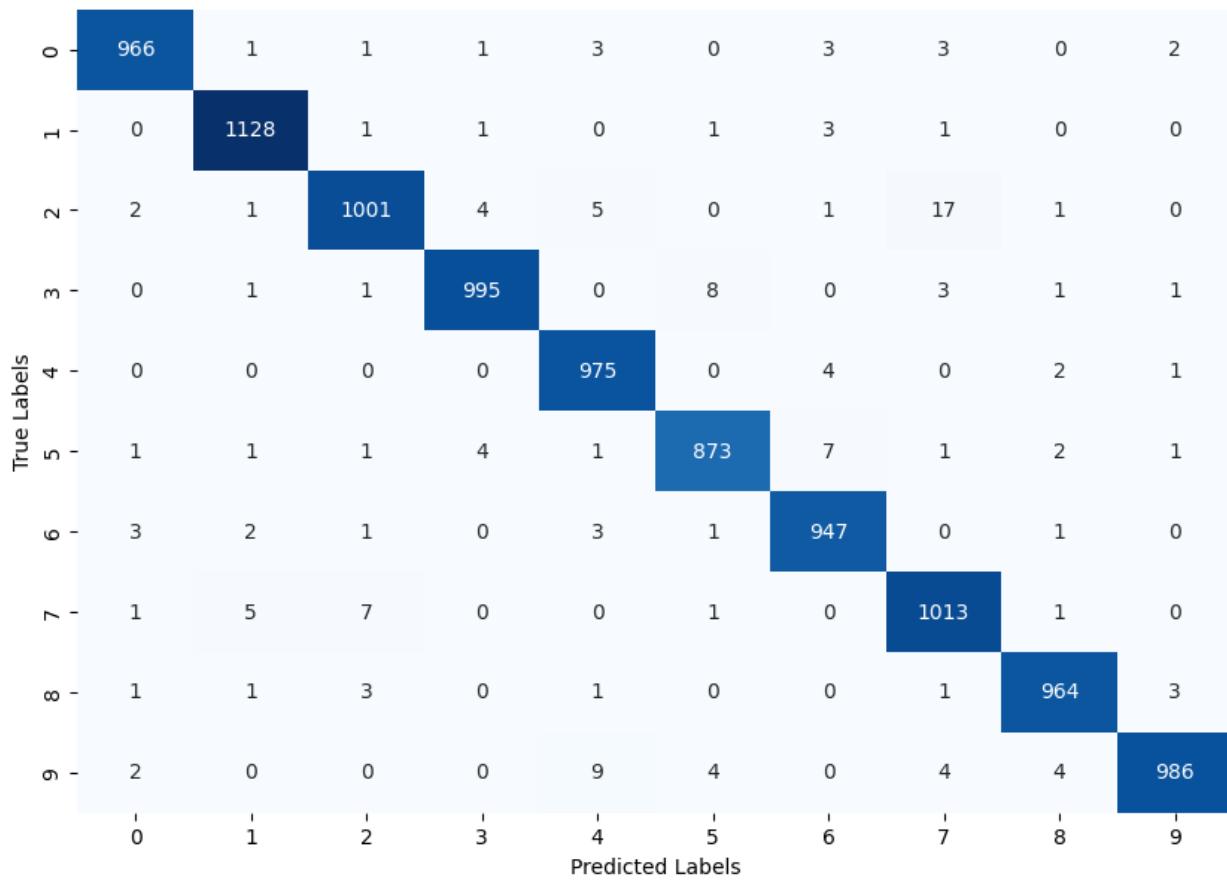
```
Confusion Matrix:
```

```
[[ 966   1   1   1   3   0   3   3   0   2]
 [ 0 1128   1   1   0   1   3   1   0   0]
 [ 2   1 1001   4   5   0   1  17   1   0]
 [ 0   1   1 995   0   8   0   3   1   1]
 [ 0   0   0   0 975   0   4   0   2   1]
 [ 1   1   1   4   1 873   7   1   2   1]
 [ 3   2   1   0   3   1 947   0   1   0]
 [ 1   5   7   0   0   1   0 1013   1   0]
 [ 1   1   3   0   1   0   0   1 964   3]
 [ 2   0   0   0   9   4   0   4   4 986]]
```

```
Precision: 0.9848
```

```
Recall: 0.9848
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 3 conv_layers, 3 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 4s - loss: 0.7162 - accuracy: 0.7566 - val_loss: 0.1773 -
val_accuracy: 0.9485 - 4s/epoch - 21ms/step
Epoch 2/5
211/211 - 3s - loss: 0.1799 - accuracy: 0.9448 - val_loss: 0.1222 -
val_accuracy: 0.9638 - 3s/epoch - 16ms/step
Epoch 3/5
211/211 - 3s - loss: 0.1361 - accuracy: 0.9575 - val_loss: 0.1141 -
val_accuracy: 0.9682 - 3s/epoch - 16ms/step
Epoch 4/5
211/211 - 3s - loss: 0.1089 - accuracy: 0.9659 - val_loss: 0.0916 -
val_accuracy: 0.9715 - 3s/epoch - 16ms/step
Epoch 5/5
211/211 - 3s - loss: 0.0947 - accuracy: 0.9708 - val_loss: 0.0819 -
val_accuracy: 0.9752 - 3s/epoch - 16ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 966    1    3    0    4    1    2    1    2    0]
 [  0 1131    1    1    1    0    0    1    0    0]
 [  3    4  992    2    3    0    0    24    4    0]
 [  0    0    4  983    0    6    0    8    5    4]
 [  1    2    0    0  972    0    0    0    2    5]
 [  7    3    0    4    2  865    8    1    2    0]
 [  4    3    0    1   13    3  930    0    3    1]
 [  0    5   16    4    0    1    0  995    1    6]
 [  1    0    2    0   10    1    2    4  945    9]
 [  5    3    0    1   14    8    0    7    4  967]]
```

Precision: 0.9747
Recall: 0.9746

Confusion Matrix for relu Activation

	0	1	3	0	4	1	2	1	2	0
True Labels	966	1	3	0	4	1	2	1	2	0
0	966	1	3	0	4	1	2	1	2	0
1	0	1131	1	1	1	0	0	1	0	0
2	3	4	992	2	3	0	0	24	4	0
3	0	0	4	983	0	6	0	8	5	4
4	1	2	0	0	972	0	0	0	2	5
5	7	3	0	4	2	865	8	1	2	0
6	4	3	0	1	13	3	930	0	3	1
7	0	5	16	4	0	1	0	995	1	6
8	1	0	2	0	10	1	2	4	945	9
9	5	3	0	1	14	8	0	7	4	967
0	1	2	3	4	5	6	7	8	9	9
Predicted Labels										

Training Model with relu activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..

Epoch 1/15

211/211 - 4s - loss: 0.8107 - accuracy: 0.7346 - val_loss: 0.2334 -
val_accuracy: 0.9265 - 4s/epoch - 21ms/step

Epoch 2/15

211/211 - 3s - loss: 0.1840 - accuracy: 0.9435 - val_loss: 0.1205 -
val_accuracy: 0.9617 - 3s/epoch - 16ms/step

Epoch 3/15

211/211 - 3s - loss: 0.1352 - accuracy: 0.9581 - val_loss: 0.0874 -
val_accuracy: 0.9735 - 3s/epoch - 17ms/step

Epoch 4/15

211/211 - 4s - loss: 0.1088 - accuracy: 0.9667 - val_loss: 0.0880 -
val_accuracy: 0.9732 - 4s/epoch - 17ms/step

Epoch 5/15

211/211 - 4s - loss: 0.0925 - accuracy: 0.9715 - val_loss: 0.0678 -
val_accuracy: 0.9783 - 4s/epoch - 18ms/step

Epoch 6/15

211/211 - 4s - loss: 0.0824 - accuracy: 0.9744 - val_loss: 0.0668 -
val_accuracy: 0.9792 - 4s/epoch - 17ms/step

Epoch 7/15

```
211/211 - 4s - loss: 0.0759 - accuracy: 0.9763 - val_loss: 0.0633 -  
val_accuracy: 0.9815 - 4s/epoch - 17ms/step  
Epoch 8/15  
211/211 - 3s - loss: 0.0690 - accuracy: 0.9781 - val_loss: 0.0644 -  
val_accuracy: 0.9817 - 3s/epoch - 17ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.0616 - accuracy: 0.9806 - val_loss: 0.0580 -  
val_accuracy: 0.9828 - 3s/epoch - 17ms/step  
Epoch 10/15  
211/211 - 4s - loss: 0.0552 - accuracy: 0.9831 - val_loss: 0.0540 -  
val_accuracy: 0.9830 - 4s/epoch - 17ms/step  
Epoch 11/15  
211/211 - 4s - loss: 0.0519 - accuracy: 0.9836 - val_loss: 0.0698 -  
val_accuracy: 0.9778 - 4s/epoch - 18ms/step  
Epoch 12/15  
211/211 - 4s - loss: 0.0506 - accuracy: 0.9839 - val_loss: 0.0483 -  
val_accuracy: 0.9850 - 4s/epoch - 17ms/step  
Epoch 13/15  
211/211 - 4s - loss: 0.0465 - accuracy: 0.9851 - val_loss: 0.0552 -  
val_accuracy: 0.9827 - 4s/epoch - 17ms/step  
Epoch 14/15  
211/211 - 4s - loss: 0.0442 - accuracy: 0.9861 - val_loss: 0.0572 -  
val_accuracy: 0.9835 - 4s/epoch - 17ms/step  
Epoch 15/15  
211/211 - 4s - loss: 0.0400 - accuracy: 0.9874 - val_loss: 0.0559 -  
val_accuracy: 0.9830 - 4s/epoch - 17ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 970 1 2 1 1 0 2 3 0 0 ]  
[ 0 1126 3 1 0 1 4 0 0 0 ]  
[ 5 1 1003 7 1 0 2 7 5 1 ]  
[ 0 1 2 997 0 7 0 1 2 0 ]  
[ 1 0 0 0 972 0 2 0 1 6 ]  
[ 4 0 0 5 0 869 8 1 4 1 ]  
[ 7 1 1 0 1 3 943 0 2 0 ]  
[ 0 7 35 14 0 1 0 968 2 1 ]  
[ 5 0 2 2 3 0 5 0 955 2 ]  
[ 8 1 0 0 7 9 0 1 11 972 ]]  
Precision: 0.9776  
Recall: 0.9775
```

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	970	1	2	1	1	0	2	3	0	0	0
0	0	1126	3	1	0	1	4	0	0	0	0
1	5	1	1003	7	1	0	2	7	5	1	0
2	0	1	2	997	0	7	0	1	2	0	0
3	1	0	0	0	972	0	2	0	1	6	0
4	4	0	0	5	0	869	8	1	4	1	0
5	7	1	1	0	1	3	943	0	2	0	0
6	0	7	35	14	0	1	0	968	2	1	0
7	5	0	2	2	3	0	5	0	955	2	0
8	8	1	0	0	7	9	0	1	11	972	0
9	1	2	3	4	5	6	7	8	9	9	0
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

Training Model with relu activation, 3 conv_layers, 3 dense layers, 256 batch size, 20 epochs..

Epoch 1/20

211/211 - 4s - loss: 0.6562 - accuracy: 0.7872 - val_loss: 0.1644 - val_accuracy: 0.9510 - 4s/epoch - 21ms/step

Epoch 2/20

211/211 - 3s - loss: 0.1633 - accuracy: 0.9489 - val_loss: 0.1222 - val_accuracy: 0.9652 - 3s/epoch - 17ms/step

Epoch 3/20

211/211 - 4s - loss: 0.1190 - accuracy: 0.9630 - val_loss: 0.0892 - val_accuracy: 0.9732 - 4s/epoch - 17ms/step

Epoch 4/20

211/211 - 3s - loss: 0.0988 - accuracy: 0.9689 - val_loss: 0.0871 - val_accuracy: 0.9730 - 3s/epoch - 16ms/step

Epoch 5/20

211/211 - 3s - loss: 0.0831 - accuracy: 0.9739 - val_loss: 0.0741 - val_accuracy: 0.9758 - 3s/epoch - 16ms/step

Epoch 6/20

211/211 - 3s - loss: 0.0733 - accuracy: 0.9767 - val_loss: 0.0832 - val_accuracy: 0.9740 - 3s/epoch - 16ms/step

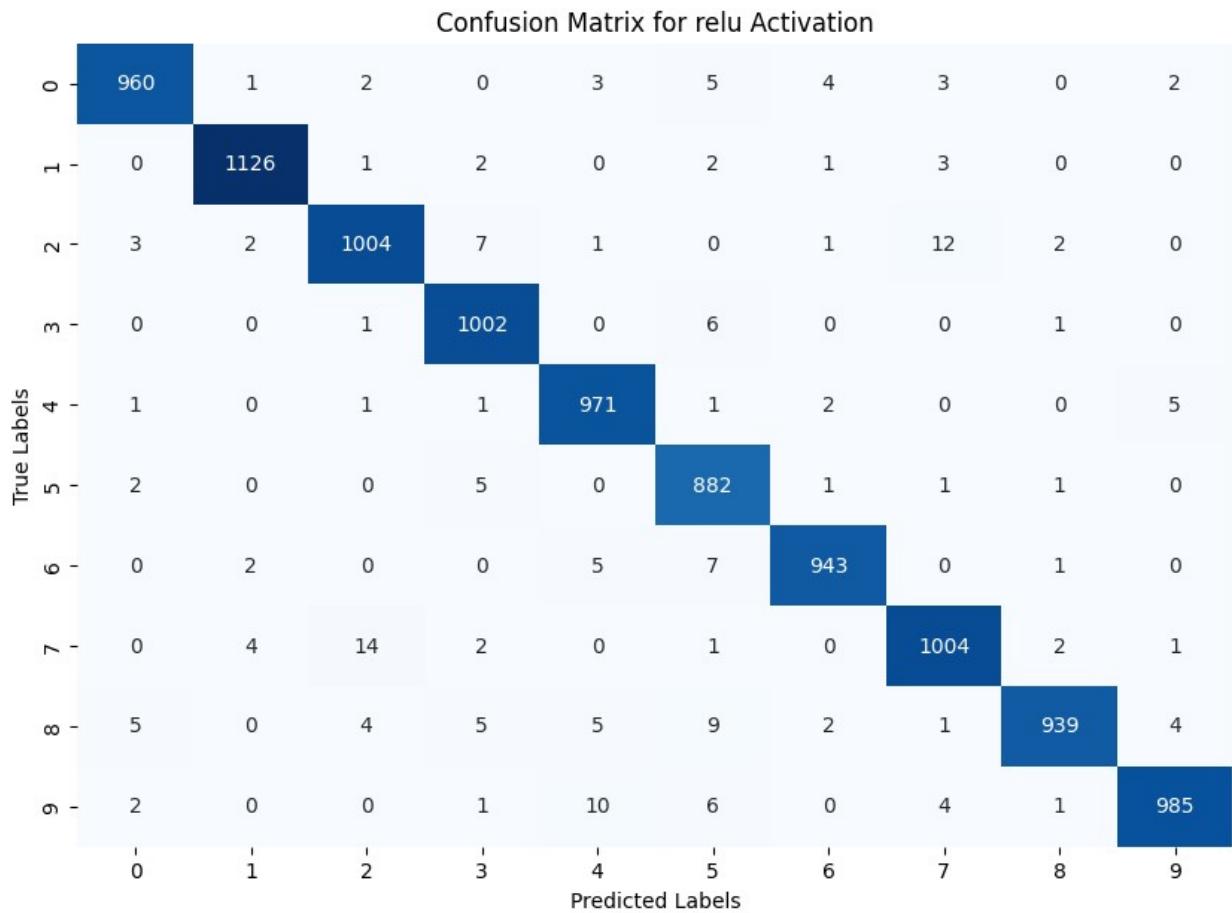
Epoch 7/20

```
211/211 - 3s - loss: 0.0683 - accuracy: 0.9783 - val_loss: 0.0641 -  
val_accuracy: 0.9795 - 3s/epoch - 16ms/step  
Epoch 8/20  
211/211 - 3s - loss: 0.0627 - accuracy: 0.9800 - val_loss: 0.0666 -  
val_accuracy: 0.9805 - 3s/epoch - 16ms/step  
Epoch 9/20  
211/211 - 3s - loss: 0.0561 - accuracy: 0.9824 - val_loss: 0.0639 -  
val_accuracy: 0.9813 - 3s/epoch - 16ms/step  
Epoch 10/20  
211/211 - 3s - loss: 0.0515 - accuracy: 0.9834 - val_loss: 0.0607 -  
val_accuracy: 0.9815 - 3s/epoch - 16ms/step  
Epoch 11/20  
211/211 - 3s - loss: 0.0487 - accuracy: 0.9848 - val_loss: 0.0620 -  
val_accuracy: 0.9808 - 3s/epoch - 16ms/step  
Epoch 12/20  
211/211 - 3s - loss: 0.0450 - accuracy: 0.9854 - val_loss: 0.0633 -  
val_accuracy: 0.9800 - 3s/epoch - 16ms/step  
Epoch 13/20  
211/211 - 3s - loss: 0.0432 - accuracy: 0.9867 - val_loss: 0.0548 -  
val_accuracy: 0.9853 - 3s/epoch - 16ms/step  
Epoch 14/20  
211/211 - 3s - loss: 0.0398 - accuracy: 0.9867 - val_loss: 0.0648 -  
val_accuracy: 0.9823 - 3s/epoch - 16ms/step  
Epoch 15/20  
211/211 - 3s - loss: 0.0380 - accuracy: 0.9877 - val_loss: 0.0540 -  
val_accuracy: 0.9833 - 3s/epoch - 16ms/step  
Epoch 16/20  
211/211 - 3s - loss: 0.0347 - accuracy: 0.9888 - val_loss: 0.0476 -  
val_accuracy: 0.9872 - 3s/epoch - 16ms/step  
Epoch 17/20  
211/211 - 3s - loss: 0.0325 - accuracy: 0.9902 - val_loss: 0.0473 -  
val_accuracy: 0.9855 - 3s/epoch - 16ms/step  
Epoch 18/20  
211/211 - 3s - loss: 0.0330 - accuracy: 0.9893 - val_loss: 0.0499 -  
val_accuracy: 0.9853 - 3s/epoch - 16ms/step  
Epoch 19/20  
211/211 - 3s - loss: 0.0264 - accuracy: 0.9914 - val_loss: 0.0538 -  
val_accuracy: 0.9847 - 3s/epoch - 16ms/step  
Epoch 20/20  
211/211 - 3s - loss: 0.0279 - accuracy: 0.9909 - val_loss: 0.0537 -  
val_accuracy: 0.9852 - 3s/epoch - 16ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 960 1 2 0 3 5 4 3 0 2]  
 [ 0 1126 1 2 0 2 1 3 0 0]  
 [ 3 2 1004 7 1 0 1 12 2 0]  
 [ 0 0 1 1002 0 6 0 0 1 0]  
 [ 1 0 1 1 971 1 2 0 0 5]]
```

```
[ 2 0 0 5 0 882 1 1 1 0]
[ 0 2 0 0 5 7 943 0 1 0]
[ 0 4 14 2 0 1 0 1004 2 1]
[ 5 0 4 5 5 9 2 1 939 4]
[ 2 0 0 1 10 6 0 4 1 985]]
```

Precision: 0.9817

Recall: 0.9816



Training Model with relu activation, 3 conv_layers, 3 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 7s - loss: 0.2926 - accuracy: 0.9064 - val_loss: 0.1064 - val_accuracy: 0.9650 - 7s/epoch - 9ms/step

Epoch 2/5

844/844 - 6s - loss: 0.0991 - accuracy: 0.9687 - val_loss: 0.0715 - val_accuracy: 0.9775 - 6s/epoch - 7ms/step

Epoch 3/5

844/844 - 6s - loss: 0.0706 - accuracy: 0.9781 - val_loss: 0.0628 - val_accuracy: 0.9800 - 6s/epoch - 7ms/step

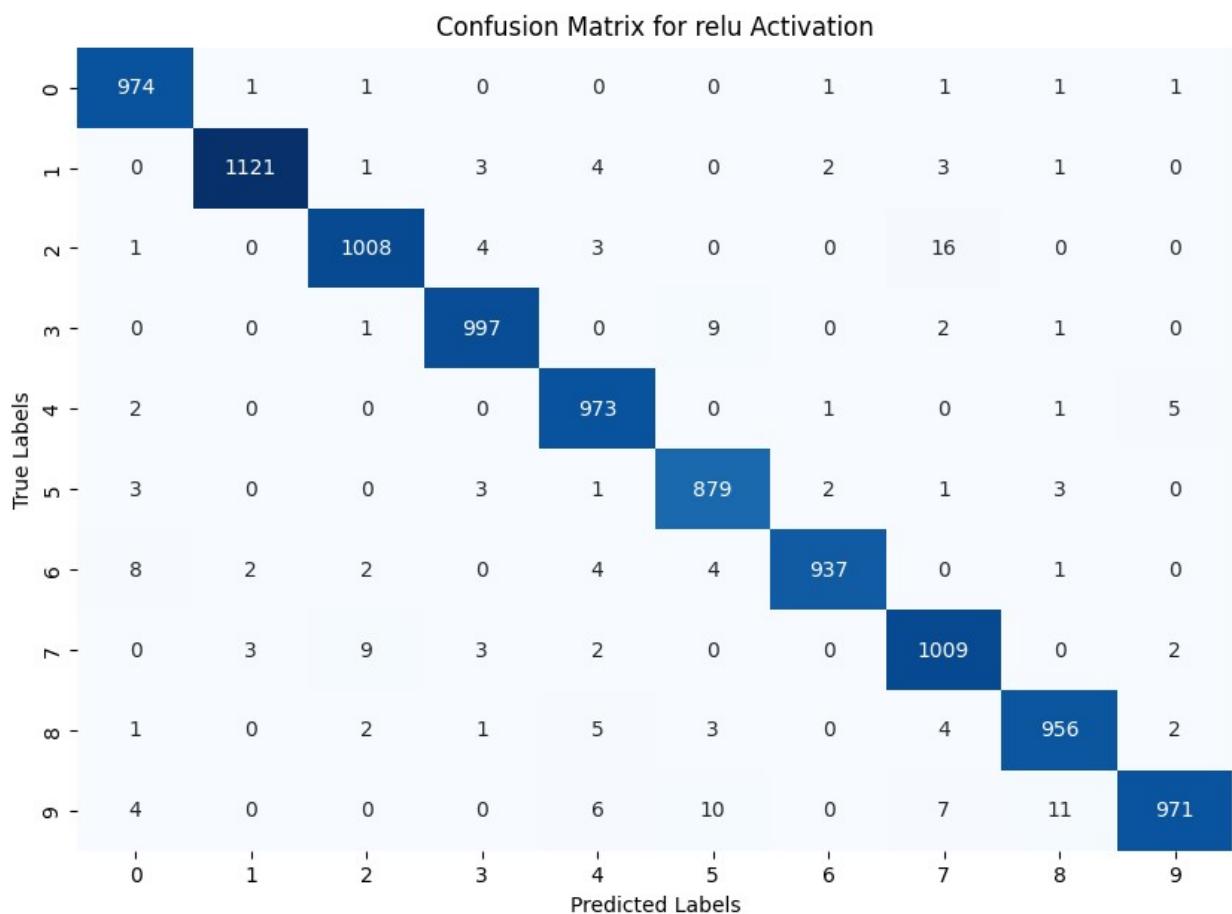
Epoch 4/5

844/844 - 6s - loss: 0.0605 - accuracy: 0.9814 - val_loss: 0.0608 -

```

val_accuracy: 0.9822 - 6s/epoch - 7ms/step
Epoch 5/5
844/844 - 6s - loss: 0.0500 - accuracy: 0.9847 - val_loss: 0.0522 -
val_accuracy: 0.9843 - 6s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 974   1   1   0   0   0   1   1   1   1]
 [  0 1121   1   3   4   0   2   3   1   0]
 [  1   0 1008   4   3   0   0   16   0   0]
 [  0   0   1 997   0   9   0   2   1   0]
 [  2   0   0   0 973   0   1   0   1   5]
 [  3   0   0   3   1 879   2   1   3   0]
 [  8   2   2   0   4   4 937   0   1   0]
 [  0   3   9   3   2   0   0 1009   0   2]
 [  1   0   2   1   5   3   0   4 956   2]
 [  4   0   0   0   6  10   0   7 11 971]]
Precision: 0.9826
Recall: 0.9825

```

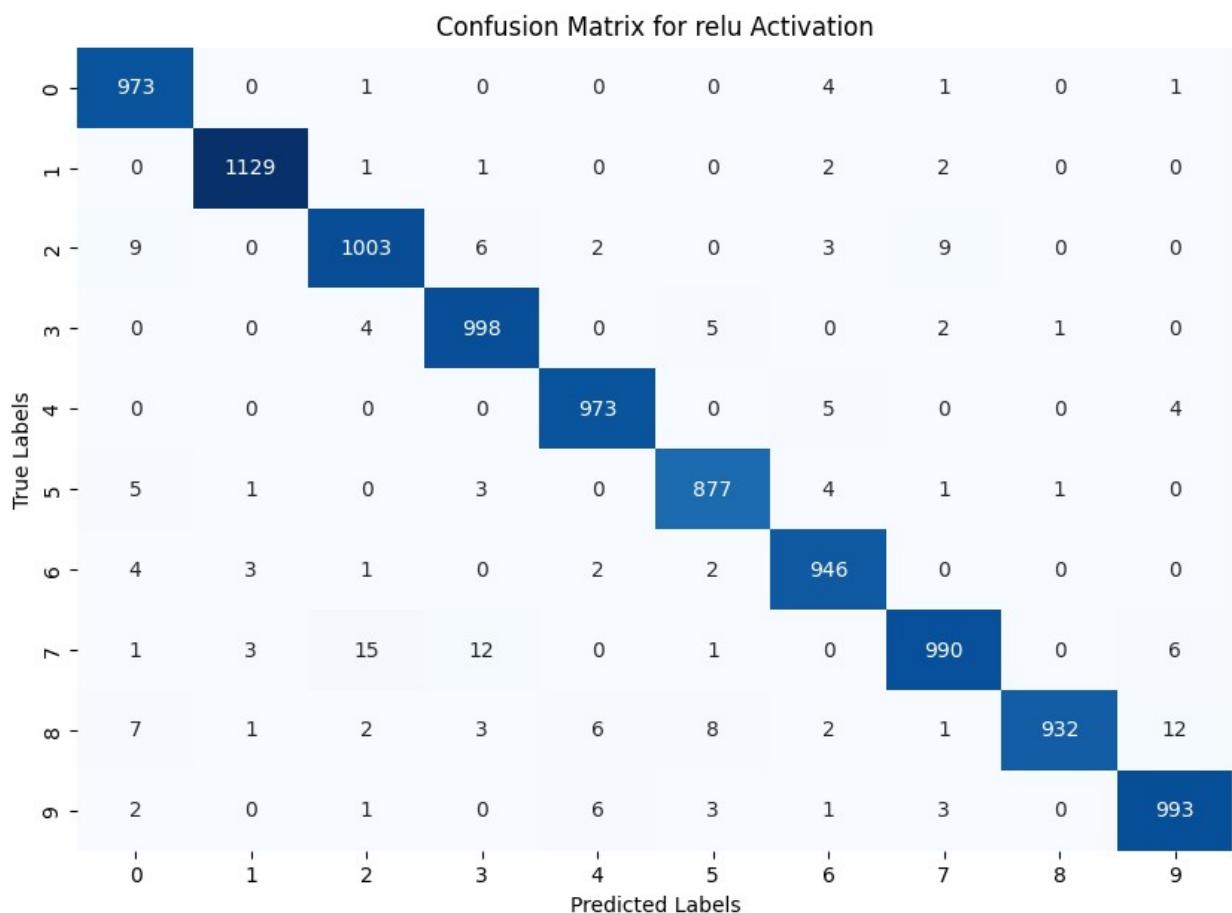


```
Training Model with relu activation, 3 conv_layers, 3 dense layers, 64
batch size, 15 epochs..
Epoch 1/15
844/844 - 7s - loss: 0.3155 - accuracy: 0.8977 - val_loss: 0.1054 -
val_accuracy: 0.9688 - 7s/epoch - 9ms/step
Epoch 2/15
844/844 - 7s - loss: 0.1051 - accuracy: 0.9672 - val_loss: 0.0778 -
val_accuracy: 0.9777 - 7s/epoch - 8ms/step
Epoch 3/15
844/844 - 6s - loss: 0.0763 - accuracy: 0.9769 - val_loss: 0.0734 -
val_accuracy: 0.9763 - 6s/epoch - 8ms/step
Epoch 4/15
844/844 - 6s - loss: 0.0638 - accuracy: 0.9801 - val_loss: 0.0732 -
val_accuracy: 0.9790 - 6s/epoch - 8ms/step
Epoch 5/15
844/844 - 6s - loss: 0.0520 - accuracy: 0.9839 - val_loss: 0.0584 -
val_accuracy: 0.9823 - 6s/epoch - 7ms/step
Epoch 6/15
844/844 - 6s - loss: 0.0465 - accuracy: 0.9852 - val_loss: 0.0578 -
val_accuracy: 0.9815 - 6s/epoch - 8ms/step
Epoch 7/15
844/844 - 6s - loss: 0.0401 - accuracy: 0.9876 - val_loss: 0.0537 -
val_accuracy: 0.9852 - 6s/epoch - 8ms/step
Epoch 8/15
844/844 - 6s - loss: 0.0360 - accuracy: 0.9888 - val_loss: 0.0616 -
val_accuracy: 0.9830 - 6s/epoch - 7ms/step
Epoch 9/15
844/844 - 6s - loss: 0.0301 - accuracy: 0.9905 - val_loss: 0.0486 -
val_accuracy: 0.9858 - 6s/epoch - 8ms/step
Epoch 10/15
844/844 - 6s - loss: 0.0286 - accuracy: 0.9906 - val_loss: 0.0697 -
val_accuracy: 0.9810 - 6s/epoch - 7ms/step
Epoch 11/15
844/844 - 6s - loss: 0.0256 - accuracy: 0.9917 - val_loss: 0.0642 -
val_accuracy: 0.9830 - 6s/epoch - 7ms/step
Epoch 12/15
844/844 - 6s - loss: 0.0252 - accuracy: 0.9914 - val_loss: 0.0494 -
val_accuracy: 0.9858 - 6s/epoch - 7ms/step
Epoch 13/15
844/844 - 6s - loss: 0.0200 - accuracy: 0.9935 - val_loss: 0.0659 -
val_accuracy: 0.9833 - 6s/epoch - 8ms/step
Epoch 14/15
844/844 - 6s - loss: 0.0187 - accuracy: 0.9935 - val_loss: 0.0535 -
val_accuracy: 0.9868 - 6s/epoch - 8ms/step
Epoch 15/15
844/844 - 6s - loss: 0.0174 - accuracy: 0.9939 - val_loss: 0.0671 -
val_accuracy: 0.9843 - 6s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
```

```
[[ 973 0 1 0 0 0 4 1 0 1]
 [ 0 1129 1 1 0 0 2 2 0 0]
 [ 9 0 1003 6 2 0 3 9 0 0]
 [ 0 0 4 998 0 5 0 2 1 0]
 [ 0 0 0 0 973 0 5 0 0 4]
 [ 5 1 0 3 0 877 4 1 1 0]
 [ 4 3 1 0 2 2 946 0 0 0]
 [ 1 3 15 12 0 1 0 990 0 6]
 [ 7 1 2 3 6 8 2 1 932 12]
 [ 2 0 1 0 6 3 1 3 0 993]]
```

Precision: 0.9815

Recall: 0.9814



Training Model with relu activation, 3 conv_layers, 3 dense layers, 64 batch size, 20 epochs..

Epoch 1/20

844/844 - 7s - loss: 0.3532 - accuracy: 0.8869 - val_loss: 0.1033 - val_accuracy: 0.9678 - 7s/epoch - 8ms/step

Epoch 2/20

844/844 - 6s - loss: 0.1131 - accuracy: 0.9652 - val_loss: 0.0787 - val_accuracy: 0.9773 - 6s/epoch - 7ms/step

```
Epoch 3/20
844/844 - 6s - loss: 0.0791 - accuracy: 0.9755 - val_loss: 0.0631 -
val_accuracy: 0.9812 - 6s/epoch - 7ms/step
Epoch 4/20
844/844 - 6s - loss: 0.0647 - accuracy: 0.9796 - val_loss: 0.0614 -
val_accuracy: 0.9822 - 6s/epoch - 7ms/step
Epoch 5/20
844/844 - 6s - loss: 0.0558 - accuracy: 0.9823 - val_loss: 0.0638 -
val_accuracy: 0.9807 - 6s/epoch - 7ms/step
Epoch 6/20
844/844 - 6s - loss: 0.0480 - accuracy: 0.9839 - val_loss: 0.0621 -
val_accuracy: 0.9830 - 6s/epoch - 7ms/step
Epoch 7/20
844/844 - 6s - loss: 0.0406 - accuracy: 0.9871 - val_loss: 0.0574 -
val_accuracy: 0.9830 - 6s/epoch - 7ms/step
Epoch 8/20
844/844 - 6s - loss: 0.0380 - accuracy: 0.9878 - val_loss: 0.0587 -
val_accuracy: 0.9842 - 6s/epoch - 7ms/step
Epoch 9/20
844/844 - 6s - loss: 0.0319 - accuracy: 0.9895 - val_loss: 0.0570 -
val_accuracy: 0.9847 - 6s/epoch - 7ms/step
Epoch 10/20
844/844 - 6s - loss: 0.0293 - accuracy: 0.9903 - val_loss: 0.0543 -
val_accuracy: 0.9855 - 6s/epoch - 7ms/step
Epoch 11/20
844/844 - 6s - loss: 0.0262 - accuracy: 0.9908 - val_loss: 0.0558 -
val_accuracy: 0.9855 - 6s/epoch - 7ms/step
Epoch 12/20
844/844 - 6s - loss: 0.0248 - accuracy: 0.9919 - val_loss: 0.0631 -
val_accuracy: 0.9840 - 6s/epoch - 7ms/step
Epoch 13/20
844/844 - 6s - loss: 0.0212 - accuracy: 0.9930 - val_loss: 0.0556 -
val_accuracy: 0.9858 - 6s/epoch - 7ms/step
Epoch 14/20
844/844 - 6s - loss: 0.0206 - accuracy: 0.9932 - val_loss: 0.0590 -
val_accuracy: 0.9868 - 6s/epoch - 7ms/step
Epoch 15/20
844/844 - 6s - loss: 0.0176 - accuracy: 0.9942 - val_loss: 0.0584 -
val_accuracy: 0.9845 - 6s/epoch - 7ms/step
Epoch 16/20
844/844 - 6s - loss: 0.0159 - accuracy: 0.9947 - val_loss: 0.0572 -
val_accuracy: 0.9863 - 6s/epoch - 7ms/step
Epoch 17/20
844/844 - 6s - loss: 0.0153 - accuracy: 0.9946 - val_loss: 0.0624 -
val_accuracy: 0.9857 - 6s/epoch - 7ms/step
Epoch 18/20
844/844 - 6s - loss: 0.0128 - accuracy: 0.9956 - val_loss: 0.0571 -
val_accuracy: 0.9877 - 6s/epoch - 7ms/step
Epoch 19/20
```

```
844/844 - 6s - loss: 0.0122 - accuracy: 0.9960 - val_loss: 0.0588 -  
val_accuracy: 0.9877 - 6s/epoch - 7ms/step
```

```
Epoch 20/20
```

```
844/844 - 6s - loss: 0.0127 - accuracy: 0.9957 - val_loss: 0.0599 -  
val_accuracy: 0.9857 - 6s/epoch - 7ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

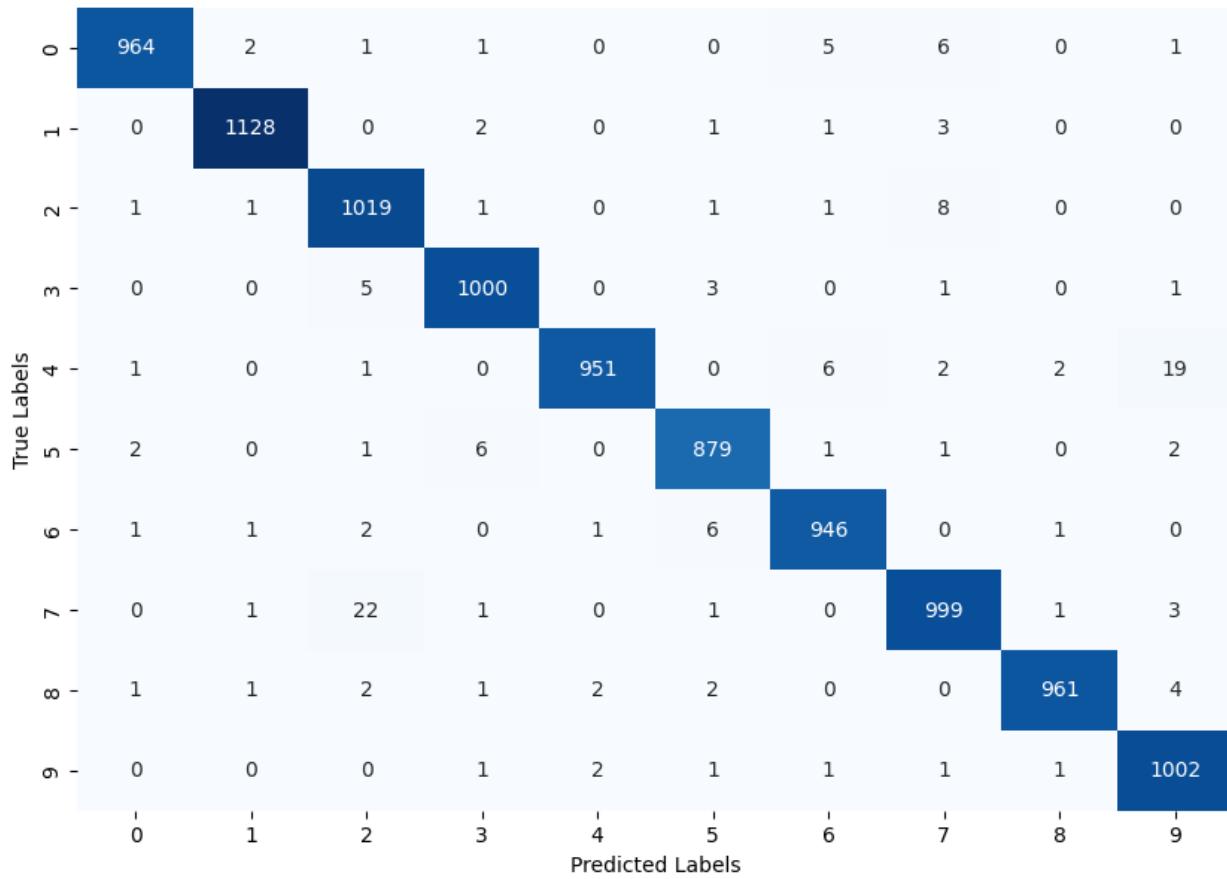
```
Confusion Matrix:
```

```
[[ 964   2   1   1   0   0   5   6   0   1]
 [ 0 1128   0   2   0   1   1   3   0   0]
 [ 1   1 1019   1   0   1   1   8   0   0]
 [ 0   0   5 1000   0   3   0   1   0   1]
 [ 1   0   1   0 951   0   6   2   2   19]
 [ 2   0   1   6   0 879   1   1   0   2]
 [ 1   1   2   0   1   6 946   0   1   0]
 [ 0   1   22  1   0   1   0 999   1   3]
 [ 1   1   2   1   2   2   0   0 961   4]
 [ 0   0   0   1   2   1   1   1   1 1002]]
```

```
Precision: 0.9850
```

```
Recall: 0.9849
```

Confusion Matrix for relu Activation



```
Training Model with relu activation, 3 conv_layers, 3 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 5s - loss: 0.3810 - accuracy: 0.8763 - val_loss: 0.1267 -
val_accuracy: 0.9623 - 5s/epoch - 13ms/step
Epoch 2/5
422/422 - 4s - loss: 0.1235 - accuracy: 0.9625 - val_loss: 0.0936 -
val_accuracy: 0.9715 - 4s/epoch - 11ms/step
Epoch 3/5
422/422 - 5s - loss: 0.0914 - accuracy: 0.9715 - val_loss: 0.1074 -
val_accuracy: 0.9665 - 5s/epoch - 11ms/step
Epoch 4/5
422/422 - 5s - loss: 0.0756 - accuracy: 0.9766 - val_loss: 0.0740 -
val_accuracy: 0.9788 - 5s/epoch - 11ms/step
Epoch 5/5
422/422 - 5s - loss: 0.0635 - accuracy: 0.9798 - val_loss: 0.0665 -
val_accuracy: 0.9808 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 955    0    6    1    2    1    3    7    1    4]
 [  0 1129    0    4    0    0    0    1    1    0]
 [  1    1 1002    5    1    0    3   19    0    0]
 [  0    0    3 1000    1    0    0    6    0    0]
 [  1    1    0    0   968    0    5    1    2    4]
 [  5    0    1   12    0   868    1    2    3    0]
 [  5    2    2    0    9   10  927    0    2    1]
 [  0    1   13    3    1    0    0 1006    2    2]
 [  1    1    4    4    3    3    0    3   950    5]
 [  0    0    0    3    4    1    0    4    5  992]]
```

Precision: 0.9798
Recall: 0.9797

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9
0	955	0	6	1	2	1	3	7	1	4
1	0	1129	0	4	0	0	0	1	1	0
2	1	1	1002	5	1	0	3	19	0	0
3	0	0	3	1000	1	0	0	6	0	0
4	1	1	0	0	968	0	5	1	2	4
5	5	0	1	12	0	868	1	2	3	0
6	5	2	2	0	9	10	927	0	2	1
7	0	1	13	3	1	0	0	1006	2	2
8	1	1	4	4	3	3	0	3	950	5
9	0	0	0	3	4	1	0	4	5	992
	0	1	2	3	4	5	6	7	8	9
	True Labels									Predicted Labels

Training Model with relu activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..

Epoch 1/15

422/422 - 5s - loss: 0.4745 - accuracy: 0.8445 - val_loss: 0.1328 -
val_accuracy: 0.9598 - 5s/epoch - 13ms/step

Epoch 2/15

422/422 - 5s - loss: 0.1280 - accuracy: 0.9609 - val_loss: 0.0872 -
val_accuracy: 0.9748 - 5s/epoch - 11ms/step

Epoch 3/15

422/422 - 5s - loss: 0.0899 - accuracy: 0.9721 - val_loss: 0.0813 -
val_accuracy: 0.9752 - 5s/epoch - 11ms/step

Epoch 4/15

422/422 - 5s - loss: 0.0743 - accuracy: 0.9764 - val_loss: 0.0666 -
val_accuracy: 0.9792 - 5s/epoch - 11ms/step

Epoch 5/15

422/422 - 5s - loss: 0.0621 - accuracy: 0.9806 - val_loss: 0.0676 -
val_accuracy: 0.9777 - 5s/epoch - 11ms/step

Epoch 6/15

422/422 - 5s - loss: 0.0549 - accuracy: 0.9830 - val_loss: 0.0505 -
val_accuracy: 0.9862 - 5s/epoch - 11ms/step

Epoch 7/15

```
422/422 - 5s - loss: 0.0478 - accuracy: 0.9845 - val_loss: 0.0573 -  
val_accuracy: 0.9842 - 5s/epoch - 11ms/step  
Epoch 8/15  
422/422 - 5s - loss: 0.0457 - accuracy: 0.9854 - val_loss: 0.0466 -  
val_accuracy: 0.9872 - 5s/epoch - 11ms/step  
Epoch 9/15  
422/422 - 5s - loss: 0.0405 - accuracy: 0.9869 - val_loss: 0.0499 -  
val_accuracy: 0.9850 - 5s/epoch - 11ms/step  
Epoch 10/15  
422/422 - 5s - loss: 0.0358 - accuracy: 0.9888 - val_loss: 0.0495 -  
val_accuracy: 0.9857 - 5s/epoch - 11ms/step  
Epoch 11/15  
422/422 - 5s - loss: 0.0314 - accuracy: 0.9897 - val_loss: 0.0554 -  
val_accuracy: 0.9858 - 5s/epoch - 11ms/step  
Epoch 12/15  
422/422 - 5s - loss: 0.0310 - accuracy: 0.9900 - val_loss: 0.0597 -  
val_accuracy: 0.9838 - 5s/epoch - 11ms/step  
Epoch 13/15  
422/422 - 4s - loss: 0.0261 - accuracy: 0.9919 - val_loss: 0.0478 -  
val_accuracy: 0.9872 - 4s/epoch - 10ms/step  
Epoch 14/15  
422/422 - 5s - loss: 0.0242 - accuracy: 0.9920 - val_loss: 0.0493 -  
val_accuracy: 0.9875 - 5s/epoch - 11ms/step  
Epoch 15/15  
422/422 - 5s - loss: 0.0225 - accuracy: 0.9930 - val_loss: 0.0496 -  
val_accuracy: 0.9865 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 972    1    0    0    1    0    1    4    1    0]  
[  0  1130    1    1    0    0    0    2    1    0]  
[  4    1  989    0    0    0    1   35    2    0]  
[  0    1    7  982    1    3    0   14    1    1]  
[  1    1    0    0  967    0    5    1    2    5]  
[  3    1    1    5    0  874    1    3    3    1]  
[  7    5    0    0    4    2  935    0    5    0]  
[  0    2    1    1    1    0    0 1022    1    0]  
[  1    0    1    0    0    0    1    6  962    3]  
[  5    1    0    3    7    5    0    9    7 972]]  
Precision: 0.9809  
Recall: 0.9805
```

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	972	1	0	0	1	0	1	4	1	0	
0	972	1	0	0	1	0	0	2	1	0	
1	0	1130	1	1	0	0	0				
2	4	1	989	0	0	0	1	35	2	0	
3	0	1	7	982	1	3	0	14	1	1	
4	1	1	0	0	967	0	5	1	2	5	
5	3	1	1	5	0	874	1	3	3	1	
6	7	5	0	0	4	2	935	0	5	0	
7	0	2	1	1	1	0	0	1022	1	0	
8	1	0	1	0	0	0	1	6	962	3	
9	5	1	0	3	7	5	0	9	7	972	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

Training Model with relu activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..

Epoch 1/20

422/422 - 6s - loss: 0.5025 - accuracy: 0.8330 - val_loss: 0.1425 -
val_accuracy: 0.9593 - 6s/epoch - 13ms/step

Epoch 2/20

422/422 - 5s - loss: 0.1423 - accuracy: 0.9557 - val_loss: 0.1007 -
val_accuracy: 0.9702 - 5s/epoch - 11ms/step

Epoch 3/20

422/422 - 5s - loss: 0.1054 - accuracy: 0.9675 - val_loss: 0.0814 -
val_accuracy: 0.9772 - 5s/epoch - 11ms/step

Epoch 4/20

422/422 - 5s - loss: 0.0877 - accuracy: 0.9722 - val_loss: 0.0759 -
val_accuracy: 0.9770 - 5s/epoch - 11ms/step

Epoch 5/20

422/422 - 5s - loss: 0.0738 - accuracy: 0.9774 - val_loss: 0.0622 -
val_accuracy: 0.9820 - 5s/epoch - 11ms/step

Epoch 6/20

422/422 - 5s - loss: 0.0661 - accuracy: 0.9794 - val_loss: 0.0623 -
val_accuracy: 0.9813 - 5s/epoch - 11ms/step

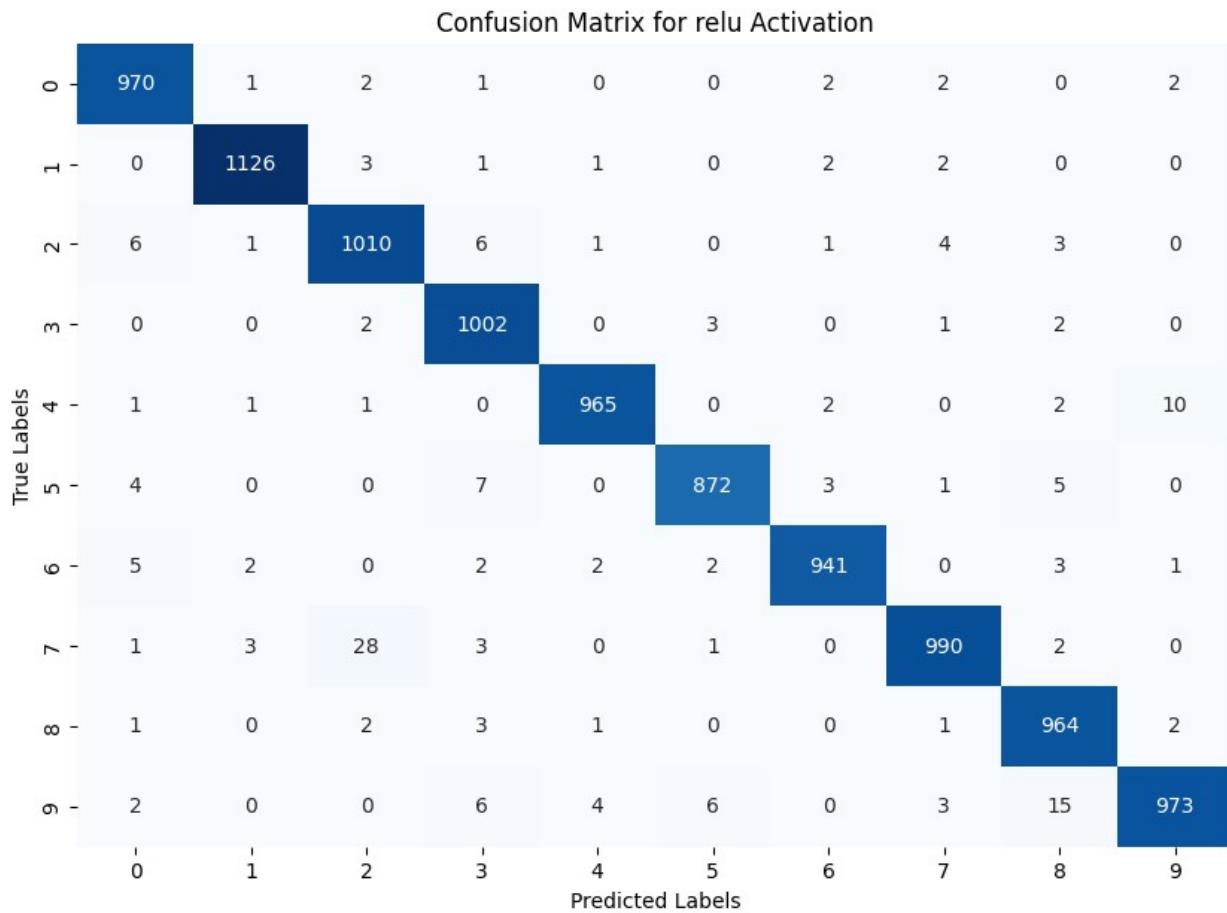
Epoch 7/20

```
422/422 - 5s - loss: 0.0569 - accuracy: 0.9821 - val_loss: 0.0502 -  
val_accuracy: 0.9850 - 5s/epoch - 11ms/step  
Epoch 8/20  
422/422 - 5s - loss: 0.0507 - accuracy: 0.9844 - val_loss: 0.0541 -  
val_accuracy: 0.9830 - 5s/epoch - 11ms/step  
Epoch 9/20  
422/422 - 5s - loss: 0.0496 - accuracy: 0.9841 - val_loss: 0.0515 -  
val_accuracy: 0.9843 - 5s/epoch - 11ms/step  
Epoch 10/20  
422/422 - 4s - loss: 0.0448 - accuracy: 0.9856 - val_loss: 0.0502 -  
val_accuracy: 0.9853 - 4s/epoch - 11ms/step  
Epoch 11/20  
422/422 - 5s - loss: 0.0388 - accuracy: 0.9874 - val_loss: 0.0457 -  
val_accuracy: 0.9868 - 5s/epoch - 11ms/step  
Epoch 12/20  
422/422 - 4s - loss: 0.0377 - accuracy: 0.9877 - val_loss: 0.0650 -  
val_accuracy: 0.9810 - 4s/epoch - 11ms/step  
Epoch 13/20  
422/422 - 4s - loss: 0.0358 - accuracy: 0.9884 - val_loss: 0.0532 -  
val_accuracy: 0.9837 - 4s/epoch - 11ms/step  
Epoch 14/20  
422/422 - 5s - loss: 0.0314 - accuracy: 0.9900 - val_loss: 0.0500 -  
val_accuracy: 0.9850 - 5s/epoch - 11ms/step  
Epoch 15/20  
422/422 - 4s - loss: 0.0302 - accuracy: 0.9902 - val_loss: 0.0548 -  
val_accuracy: 0.9865 - 4s/epoch - 10ms/step  
Epoch 16/20  
422/422 - 4s - loss: 0.0296 - accuracy: 0.9900 - val_loss: 0.0476 -  
val_accuracy: 0.9863 - 4s/epoch - 10ms/step  
Epoch 17/20  
422/422 - 4s - loss: 0.0264 - accuracy: 0.9911 - val_loss: 0.0473 -  
val_accuracy: 0.9882 - 4s/epoch - 11ms/step  
Epoch 18/20  
422/422 - 5s - loss: 0.0249 - accuracy: 0.9919 - val_loss: 0.0526 -  
val_accuracy: 0.9853 - 5s/epoch - 11ms/step  
Epoch 19/20  
422/422 - 4s - loss: 0.0236 - accuracy: 0.9919 - val_loss: 0.0530 -  
val_accuracy: 0.9858 - 4s/epoch - 11ms/step  
Epoch 20/20  
422/422 - 4s - loss: 0.0191 - accuracy: 0.9937 - val_loss: 0.0556 -  
val_accuracy: 0.9845 - 4s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 970   1   2   1   0   0   2   2   0   2]  
 [  0 1126   3   1   1   0   2   2   0   0]  
 [  6   1 1010   6   1   0   1   4   3   0]  
 [  0   0   2 1002   0   3   0   1   2   0]  
 [  1   1   1   0  965   0   2   0   2  10]]
```

```
[ 4 0 0 7 0 872 3 1 5 0]
[ 5 2 0 2 2 2 941 0 3 1]
[ 1 3 28 3 0 1 0 990 2 0]
[ 1 0 2 3 1 0 0 1 964 2]
[ 2 0 0 6 4 6 0 3 15 973]]
```

Precision: 0.9814

Recall: 0.9813



Training Model with relu activation, 3 conv_layers, 3 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 5s - loss: 0.5642 - accuracy: 0.8166 - val_loss: 0.1955 - val_accuracy: 0.9402 - 5s/epoch - 22ms/step

Epoch 2/5

211/211 - 4s - loss: 0.1690 - accuracy: 0.9482 - val_loss: 0.1166 - val_accuracy: 0.9643 - 4s/epoch - 17ms/step

Epoch 3/5

211/211 - 4s - loss: 0.1216 - accuracy: 0.9629 - val_loss: 0.0940 - val_accuracy: 0.9723 - 4s/epoch - 18ms/step

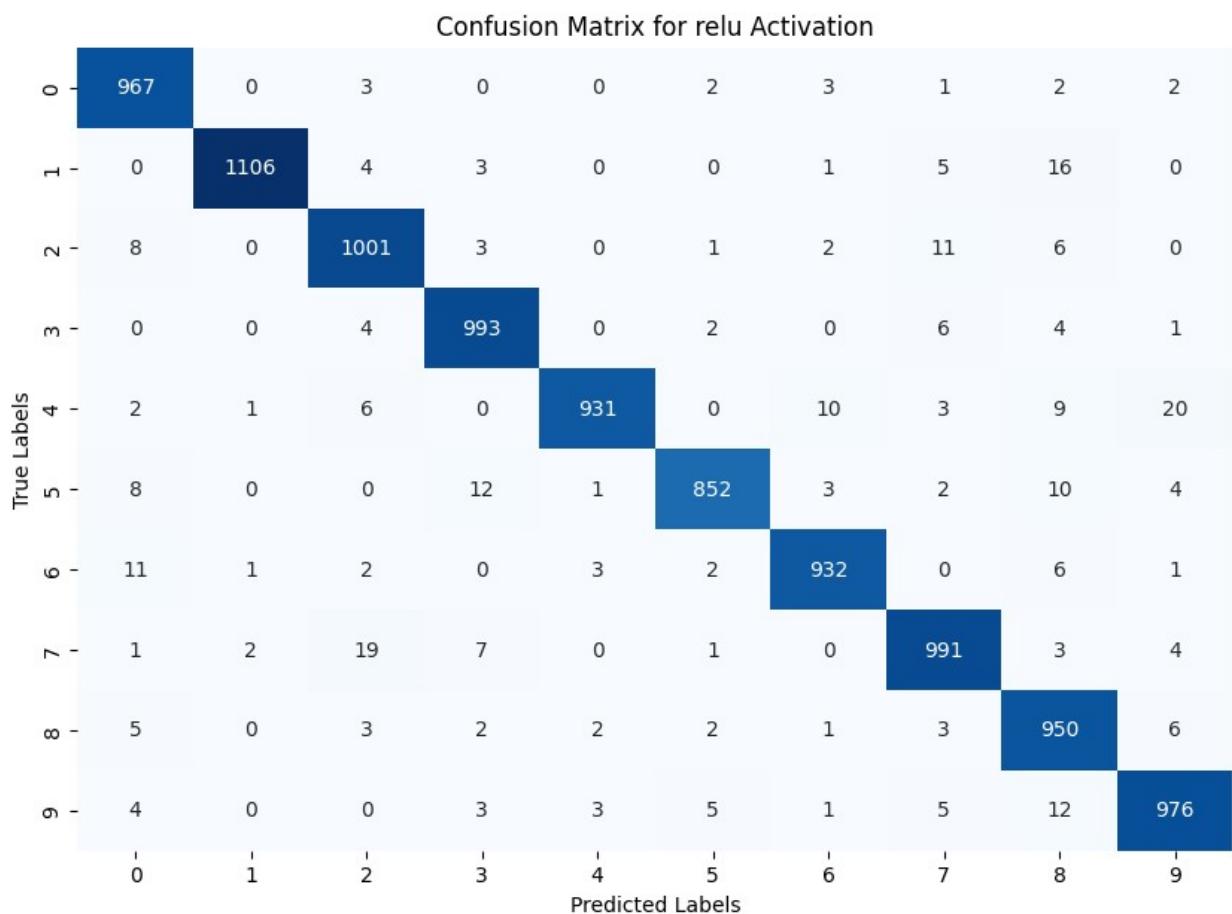
Epoch 4/5

211/211 - 4s - loss: 0.1006 - accuracy: 0.9688 - val_loss: 0.0924 -

```

val_accuracy: 0.9723 - 4s/epoch - 18ms/step
Epoch 5/5
211/211 - 4s - loss: 0.0866 - accuracy: 0.9732 - val_loss: 0.0816 -
val_accuracy: 0.9757 - 4s/epoch - 17ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 967   0   3   0   0   2   3   1   2   2]
 [  0 1106   4   3   0   0   1   5   16   0]
 [  8   0 1001   3   0   1   2   11   6   0]
 [  0   0   4 993   0   2   0   6   4   1]
 [  2   1   6   0 931   0   10   3   9   20]
 [  8   0   0  12   1 852   3   2   10   4]
 [ 11   1   2   0   3   2 932   0   6   1]
 [  1   2  19   7   0   1   0 991   3   4]
 [  5   0   3   2   2   2   1   3 950   6]
 [  4   0   0   3   3   5   1   5 12 976]]
Precision: 0.9703
Recall: 0.9699

```

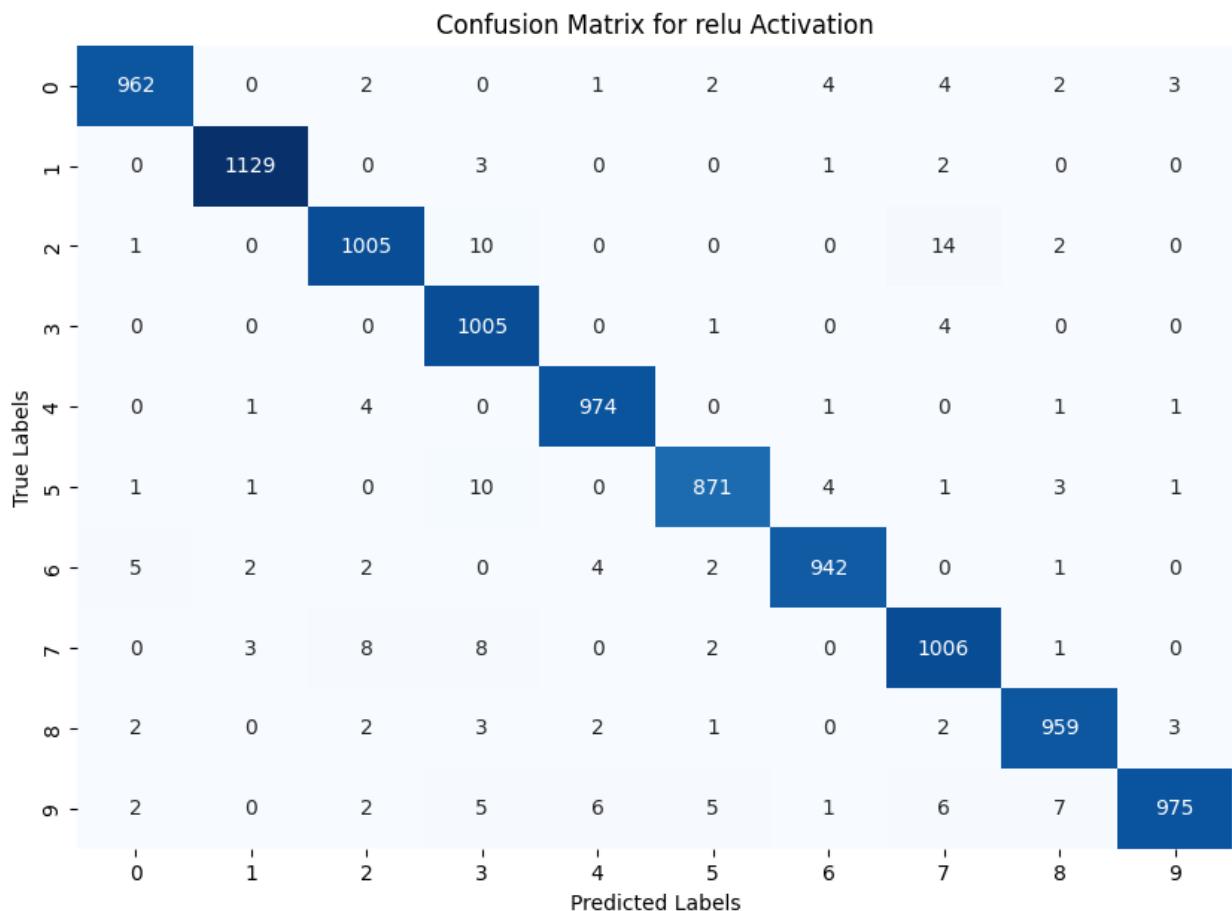


```
Training Model with relu activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 4s - loss: 0.8342 - accuracy: 0.7427 - val_loss: 0.2161 -
val_accuracy: 0.9343 - 4s/epoch - 21ms/step
Epoch 2/15
211/211 - 4s - loss: 0.1946 - accuracy: 0.9407 - val_loss: 0.1394 -
val_accuracy: 0.9582 - 4s/epoch - 18ms/step
Epoch 3/15
211/211 - 4s - loss: 0.1381 - accuracy: 0.9572 - val_loss: 0.1133 -
val_accuracy: 0.9657 - 4s/epoch - 17ms/step
Epoch 4/15
211/211 - 4s - loss: 0.1103 - accuracy: 0.9657 - val_loss: 0.0881 -
val_accuracy: 0.9735 - 4s/epoch - 17ms/step
Epoch 5/15
211/211 - 4s - loss: 0.0941 - accuracy: 0.9716 - val_loss: 0.0855 -
val_accuracy: 0.9763 - 4s/epoch - 18ms/step
Epoch 6/15
211/211 - 4s - loss: 0.0795 - accuracy: 0.9753 - val_loss: 0.0775 -
val_accuracy: 0.9775 - 4s/epoch - 18ms/step
Epoch 7/15
211/211 - 4s - loss: 0.0714 - accuracy: 0.9779 - val_loss: 0.0740 -
val_accuracy: 0.9785 - 4s/epoch - 17ms/step
Epoch 8/15
211/211 - 4s - loss: 0.0652 - accuracy: 0.9797 - val_loss: 0.0639 -
val_accuracy: 0.9810 - 4s/epoch - 17ms/step
Epoch 9/15
211/211 - 4s - loss: 0.0594 - accuracy: 0.9817 - val_loss: 0.0606 -
val_accuracy: 0.9830 - 4s/epoch - 17ms/step
Epoch 10/15
211/211 - 4s - loss: 0.0519 - accuracy: 0.9835 - val_loss: 0.0567 -
val_accuracy: 0.9845 - 4s/epoch - 17ms/step
Epoch 11/15
211/211 - 4s - loss: 0.0517 - accuracy: 0.9835 - val_loss: 0.0739 -
val_accuracy: 0.9775 - 4s/epoch - 17ms/step
Epoch 12/15
211/211 - 4s - loss: 0.0461 - accuracy: 0.9853 - val_loss: 0.0553 -
val_accuracy: 0.9840 - 4s/epoch - 18ms/step
Epoch 13/15
211/211 - 4s - loss: 0.0429 - accuracy: 0.9869 - val_loss: 0.0679 -
val_accuracy: 0.9800 - 4s/epoch - 18ms/step
Epoch 14/15
211/211 - 4s - loss: 0.0410 - accuracy: 0.9865 - val_loss: 0.0544 -
val_accuracy: 0.9845 - 4s/epoch - 18ms/step
Epoch 15/15
211/211 - 4s - loss: 0.0372 - accuracy: 0.9880 - val_loss: 0.0537 -
val_accuracy: 0.9842 - 4s/epoch - 17ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
```

```
[[ 962  0  2  0  1  2  4  4  2  3]
 [ 0 1129  0  3  0  0  1  2  0  0]
 [ 1  0 1005 10  0  0  0 14  2  0]
 [ 0  0  0 1005  0  1  0  4  0  0]
 [ 0  1  4  0 974  0  1  0  1  1]
 [ 1  1  0 10  0 871  4  1  3  1]
 [ 5  2  2  0  4  2 942  0  1  0]
 [ 0  3  8  8  0  2  0 1006  1  0]
 [ 2  0  2  3  2  1  0  2 959  3]
 [ 2  0  2  5  6  5  1  6  7 975]]
```

Precision: 0.9829

Recall: 0.9828



Training Model with relu activation, 3 conv_layers, 3 dense layers,
256 batch size, 20 epochs..

Epoch 1/20

211/211 - 5s - loss: 0.4350 - accuracy: 0.8586 - val_loss: 0.1468 -
val_accuracy: 0.9567 - 5s/epoch - 21ms/step

Epoch 2/20

211/211 - 4s - loss: 0.1377 - accuracy: 0.9569 - val_loss: 0.0887 -
val_accuracy: 0.9740 - 4s/epoch - 17ms/step

```
Epoch 3/20
211/211 - 4s - loss: 0.1047 - accuracy: 0.9683 - val_loss: 0.0684 -
val_accuracy: 0.9808 - 4s/epoch - 18ms/step
Epoch 4/20
211/211 - 4s - loss: 0.0806 - accuracy: 0.9756 - val_loss: 0.0669 -
val_accuracy: 0.9780 - 4s/epoch - 17ms/step
Epoch 5/20
211/211 - 4s - loss: 0.0708 - accuracy: 0.9777 - val_loss: 0.0731 -
val_accuracy: 0.9778 - 4s/epoch - 18ms/step
Epoch 6/20
211/211 - 4s - loss: 0.0582 - accuracy: 0.9818 - val_loss: 0.0635 -
val_accuracy: 0.9797 - 4s/epoch - 18ms/step
Epoch 7/20
211/211 - 4s - loss: 0.0529 - accuracy: 0.9830 - val_loss: 0.0558 -
val_accuracy: 0.9830 - 4s/epoch - 18ms/step
Epoch 8/20
211/211 - 4s - loss: 0.0488 - accuracy: 0.9842 - val_loss: 0.0541 -
val_accuracy: 0.9832 - 4s/epoch - 17ms/step
Epoch 9/20
211/211 - 4s - loss: 0.0414 - accuracy: 0.9869 - val_loss: 0.0514 -
val_accuracy: 0.9840 - 4s/epoch - 18ms/step
Epoch 10/20
211/211 - 4s - loss: 0.0407 - accuracy: 0.9869 - val_loss: 0.0581 -
val_accuracy: 0.9837 - 4s/epoch - 18ms/step
Epoch 11/20
211/211 - 4s - loss: 0.0373 - accuracy: 0.9878 - val_loss: 0.0487 -
val_accuracy: 0.9850 - 4s/epoch - 17ms/step
Epoch 12/20
211/211 - 4s - loss: 0.0374 - accuracy: 0.9884 - val_loss: 0.0486 -
val_accuracy: 0.9847 - 4s/epoch - 17ms/step
Epoch 13/20
211/211 - 4s - loss: 0.0311 - accuracy: 0.9905 - val_loss: 0.0465 -
val_accuracy: 0.9868 - 4s/epoch - 18ms/step
Epoch 14/20
211/211 - 4s - loss: 0.0296 - accuracy: 0.9903 - val_loss: 0.0501 -
val_accuracy: 0.9867 - 4s/epoch - 18ms/step
Epoch 15/20
211/211 - 4s - loss: 0.0262 - accuracy: 0.9917 - val_loss: 0.0557 -
val_accuracy: 0.9837 - 4s/epoch - 18ms/step
Epoch 16/20
211/211 - 4s - loss: 0.0265 - accuracy: 0.9911 - val_loss: 0.0453 -
val_accuracy: 0.9878 - 4s/epoch - 17ms/step
Epoch 17/20
211/211 - 4s - loss: 0.0225 - accuracy: 0.9927 - val_loss: 0.0540 -
val_accuracy: 0.9867 - 4s/epoch - 18ms/step
Epoch 18/20
211/211 - 4s - loss: 0.0215 - accuracy: 0.9933 - val_loss: 0.0584 -
val_accuracy: 0.9823 - 4s/epoch - 18ms/step
Epoch 19/20
```

```
211/211 - 4s - loss: 0.0203 - accuracy: 0.9936 - val_loss: 0.0491 -  
val_accuracy: 0.9858 - 4s/epoch - 18ms/step
```

```
Epoch 20/20
```

```
211/211 - 4s - loss: 0.0168 - accuracy: 0.9946 - val_loss: 0.0519 -  
val_accuracy: 0.9852 - 4s/epoch - 18ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: relu
```

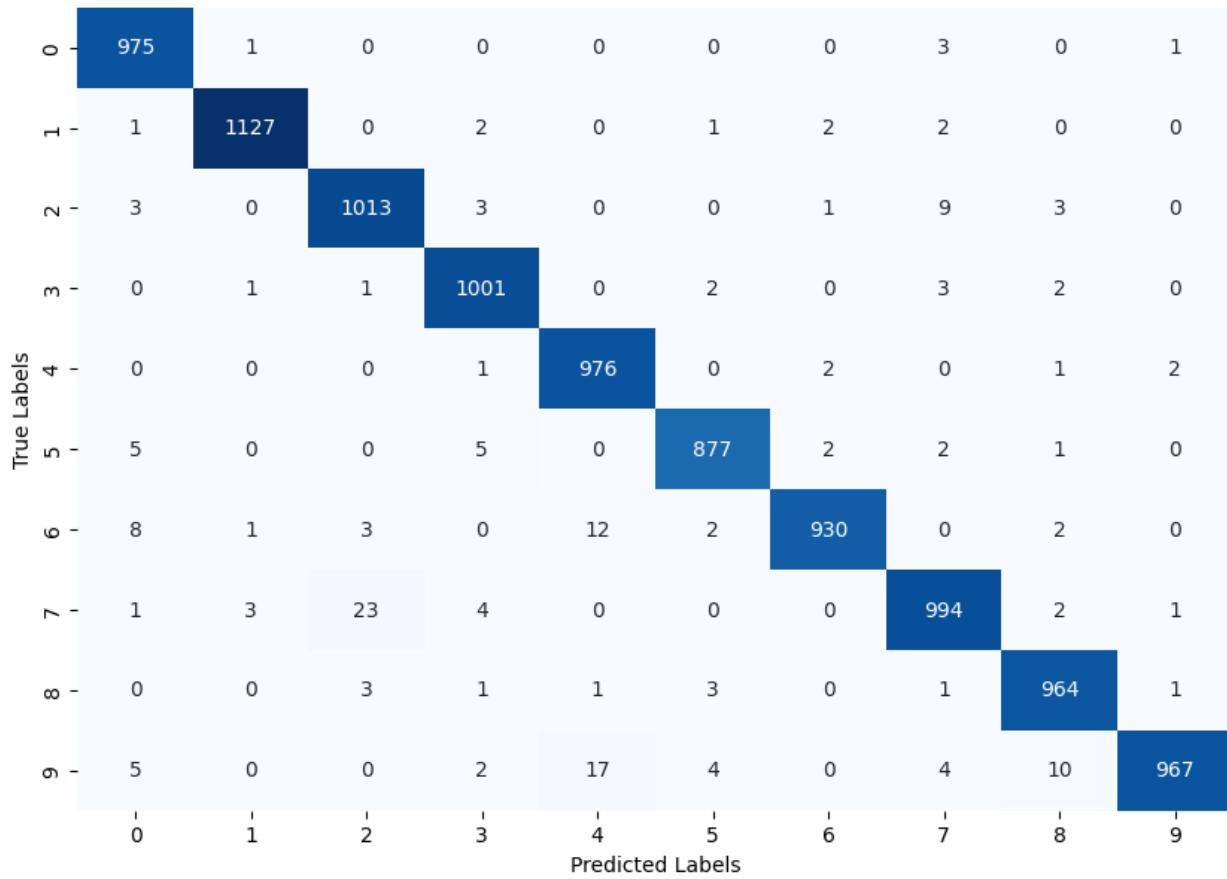
```
Confusion Matrix:
```

```
[[ 975   1   0   0   0   0   0   3   0   1]
 [ 1 1127   0   2   0   1   2   2   0   0]
 [ 3   0 1013   3   0   0   1   9   3   0]
 [ 0   1   1 1001   0   2   0   3   2   0]
 [ 0   0   0   1 976   0   2   0   1   2]
 [ 5   0   0   5   0 877   2   2   1   0]
 [ 8   1   3   0 12   2 930   0   2   0]
 [ 1   3 23   4   0   0   0 994   2   1]
 [ 0   0   3   1   1   3   0   1 964   1]
 [ 5   0   0   2 17   4   0   4 10 967]]
```

```
Precision: 0.9825
```

```
Recall: 0.9824
```

Confusion Matrix for relu Activation



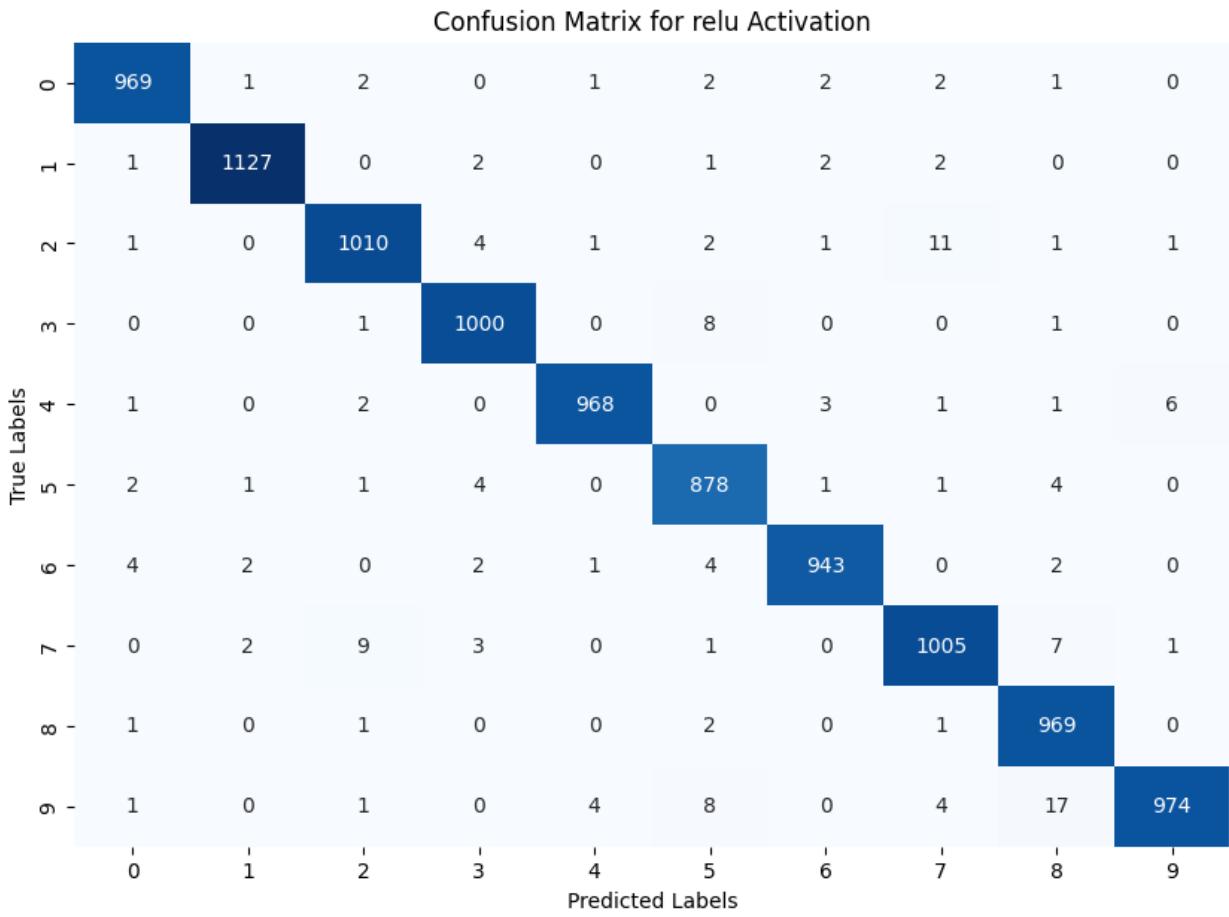
```
Training Model with relu activation, 3 conv_layers, 3 dense layers, 64
batch size, 5 epochs..
Epoch 1/5
844/844 - 8s - loss: 0.2503 - accuracy: 0.9211 - val_loss: 0.0885 -
val_accuracy: 0.9723 - 8s/epoch - 9ms/step
Epoch 2/5
844/844 - 7s - loss: 0.0926 - accuracy: 0.9705 - val_loss: 0.0728 -
val_accuracy: 0.9778 - 7s/epoch - 8ms/step
Epoch 3/5
844/844 - 7s - loss: 0.0701 - accuracy: 0.9778 - val_loss: 0.0457 -
val_accuracy: 0.9868 - 7s/epoch - 8ms/step
Epoch 4/5
844/844 - 7s - loss: 0.0541 - accuracy: 0.9828 - val_loss: 0.0458 -
val_accuracy: 0.9875 - 7s/epoch - 8ms/step
Epoch 5/5
844/844 - 7s - loss: 0.0449 - accuracy: 0.9859 - val_loss: 0.0476 -
val_accuracy: 0.9850 - 7s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: relu
Confusion Matrix:
[[ 955    1    3    1    4    2    9    3    0    2]
 [  0 1130    0    2    0    0    1    2    0    0]
 [  1    0 1010    0    0    1    2   17    0    1]
 [  0    1    6 993    0    5    0    4    0    1]
 [  0    0    0    0 977    0    2    0    0    3]
 [  1    0    1    7    0 872    4    1    4    2]
 [  2    1    1    0    1    1 951    0    1    0]
 [  0    1    4    2    0    1    0 1016    1    3]
 [  1    1    6    4    5    2    3    0  947    5]
 [  2    0    0    1    7    2    1    7    2 987]]
```

Precision: 0.9839
Recall: 0.9838

Confusion Matrix for relu Activation											
	0	1	3	1	4	2	9	3	0	2	
True Labels	955	1	3	1	4	2	9	3	0	2	
0	955	1	3	1	4	2	9	3	0	2	
1	0	1130	0	2	0	0	1	2	0	0	
2	1	0	1010	0	0	1	2	17	0	1	
3	0	1	6	993	0	5	0	4	0	1	
4	0	0	0	0	977	0	2	0	0	3	
5	1	0	1	7	0	872	4	1	4	2	
6	2	1	1	0	1	1	951	0	1	0	
7	0	1	4	2	0	1	0	1016	1	3	
8	1	1	6	4	5	2	3	0	947	5	
9	2	0	0	1	7	2	1	7	2	987	
0	1	2	3	4	5	6	7	8	9	9	
Predicted Labels											

```
Training Model with relu activation, 3 conv_layers, 3 dense layers, 64
batch size, 15 epochs..
Epoch 1/15
844/844 - 8s - loss: 0.2868 - accuracy: 0.9088 - val_loss: 0.1074 -
val_accuracy: 0.9675 - 8s/epoch - 9ms/step
Epoch 2/15
844/844 - 7s - loss: 0.1012 - accuracy: 0.9681 - val_loss: 0.0727 -
val_accuracy: 0.9775 - 7s/epoch - 8ms/step
Epoch 3/15
844/844 - 7s - loss: 0.0742 - accuracy: 0.9765 - val_loss: 0.0633 -
val_accuracy: 0.9807 - 7s/epoch - 9ms/step
Epoch 4/15
844/844 - 7s - loss: 0.0605 - accuracy: 0.9810 - val_loss: 0.0504 -
val_accuracy: 0.9835 - 7s/epoch - 9ms/step
Epoch 5/15
844/844 - 7s - loss: 0.0487 - accuracy: 0.9842 - val_loss: 0.0496 -
val_accuracy: 0.9847 - 7s/epoch - 8ms/step
Epoch 6/15
844/844 - 7s - loss: 0.0420 - accuracy: 0.9867 - val_loss: 0.0569 -
val_accuracy: 0.9848 - 7s/epoch - 8ms/step
Epoch 7/15
```

```
844/844 - 7s - loss: 0.0360 - accuracy: 0.9886 - val_loss: 0.0495 -  
val_accuracy: 0.9853 - 7s/epoch - 8ms/step  
Epoch 8/15  
844/844 - 7s - loss: 0.0328 - accuracy: 0.9894 - val_loss: 0.0452 -  
val_accuracy: 0.9860 - 7s/epoch - 8ms/step  
Epoch 9/15  
844/844 - 7s - loss: 0.0273 - accuracy: 0.9911 - val_loss: 0.0505 -  
val_accuracy: 0.9862 - 7s/epoch - 8ms/step  
Epoch 10/15  
844/844 - 7s - loss: 0.0236 - accuracy: 0.9922 - val_loss: 0.0452 -  
val_accuracy: 0.9868 - 7s/epoch - 8ms/step  
Epoch 11/15  
844/844 - 7s - loss: 0.0204 - accuracy: 0.9931 - val_loss: 0.0576 -  
val_accuracy: 0.9837 - 7s/epoch - 8ms/step  
Epoch 12/15  
844/844 - 7s - loss: 0.0200 - accuracy: 0.9934 - val_loss: 0.0487 -  
val_accuracy: 0.9870 - 7s/epoch - 8ms/step  
Epoch 13/15  
844/844 - 7s - loss: 0.0177 - accuracy: 0.9939 - val_loss: 0.0595 -  
val_accuracy: 0.9847 - 7s/epoch - 8ms/step  
Epoch 14/15  
844/844 - 7s - loss: 0.0164 - accuracy: 0.9944 - val_loss: 0.0561 -  
val_accuracy: 0.9860 - 7s/epoch - 8ms/step  
Epoch 15/15  
844/844 - 7s - loss: 0.0144 - accuracy: 0.9950 - val_loss: 0.0465 -  
val_accuracy: 0.9878 - 7s/epoch - 8ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 969 1 2 0 1 2 2 2 1 0]  
[ 1 1127 0 2 0 1 2 2 0 0]  
[ 1 0 1010 4 1 2 1 11 1 1]  
[ 0 0 1 1000 0 8 0 0 1 0]  
[ 1 0 2 0 968 0 3 1 1 6]  
[ 2 1 1 4 0 878 1 1 4 0]  
[ 4 2 0 2 1 4 943 0 2 0]  
[ 0 2 9 3 0 1 0 1005 7 1]  
[ 1 0 1 0 0 2 0 1 969 0]  
[ 1 0 1 0 4 8 0 4 17 974]]  
Precision: 0.9844  
Recall: 0.9843
```



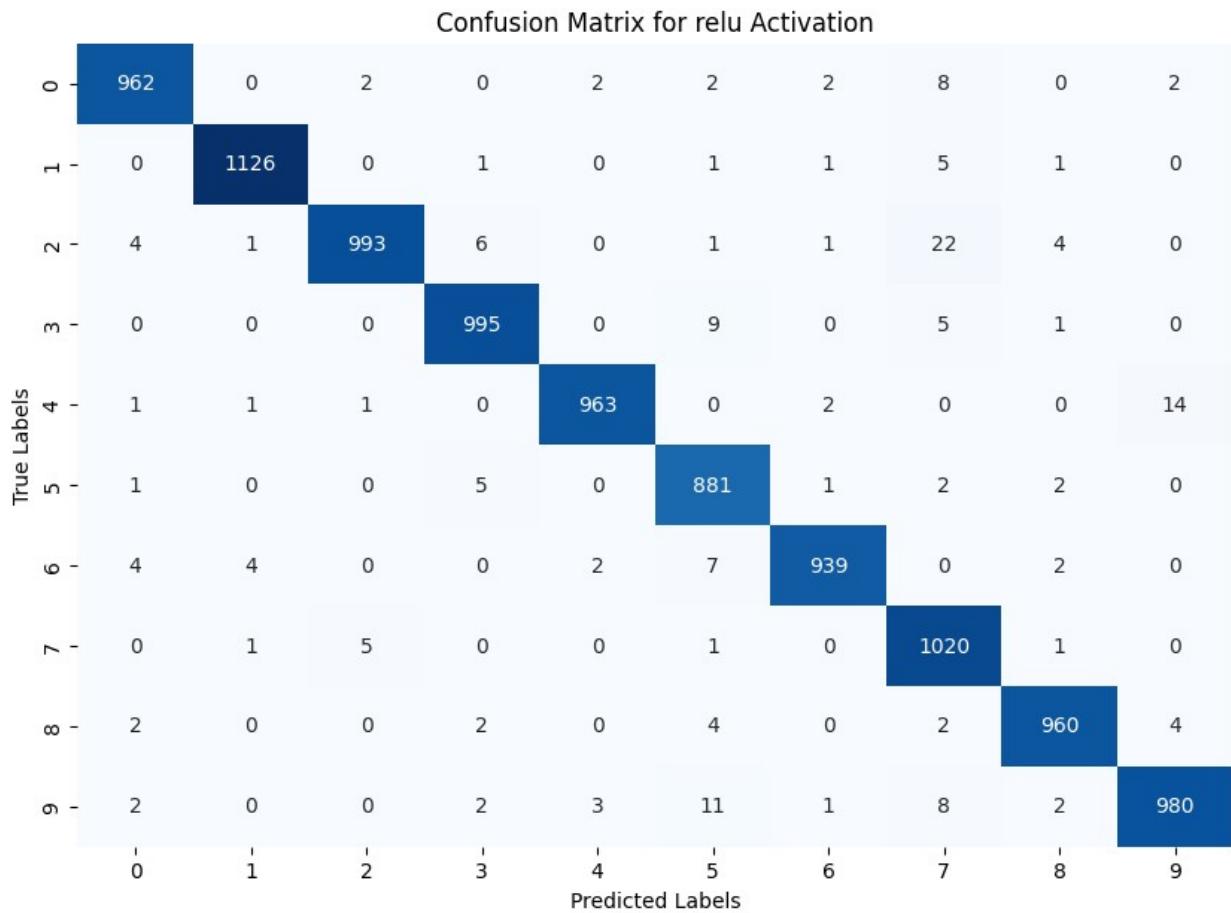
```
Training Model with relu activation, 3 conv_layers, 3 dense layers, 64
batch size, 20 epochs..
Epoch 1/20
844/844 - 8s - loss: 0.3397 - accuracy: 0.8918 - val_loss: 0.1149 -
val_accuracy: 0.9647 - 8s/epoch - 9ms/step
Epoch 2/20
844/844 - 7s - loss: 0.1020 - accuracy: 0.9681 - val_loss: 0.0649 -
val_accuracy: 0.9793 - 7s/epoch - 8ms/step
Epoch 3/20
844/844 - 7s - loss: 0.0766 - accuracy: 0.9768 - val_loss: 0.0700 -
val_accuracy: 0.9788 - 7s/epoch - 8ms/step
Epoch 4/20
844/844 - 7s - loss: 0.0620 - accuracy: 0.9803 - val_loss: 0.0553 -
val_accuracy: 0.9832 - 7s/epoch - 8ms/step
Epoch 5/20
844/844 - 7s - loss: 0.0518 - accuracy: 0.9836 - val_loss: 0.0625 -
val_accuracy: 0.9820 - 7s/epoch - 8ms/step
Epoch 6/20
844/844 - 7s - loss: 0.0449 - accuracy: 0.9861 - val_loss: 0.0579 -
val_accuracy: 0.9842 - 7s/epoch - 8ms/step
Epoch 7/20
```

```
844/844 - 7s - loss: 0.0389 - accuracy: 0.9874 - val_loss: 0.0613 -  
val_accuracy: 0.9833 - 7s/epoch - 8ms/step  
Epoch 8/20  
844/844 - 7s - loss: 0.0354 - accuracy: 0.9888 - val_loss: 0.0569 -  
val_accuracy: 0.9842 - 7s/epoch - 8ms/step  
Epoch 9/20  
844/844 - 7s - loss: 0.0313 - accuracy: 0.9901 - val_loss: 0.0574 -  
val_accuracy: 0.9842 - 7s/epoch - 8ms/step  
Epoch 10/20  
844/844 - 7s - loss: 0.0281 - accuracy: 0.9907 - val_loss: 0.0716 -  
val_accuracy: 0.9800 - 7s/epoch - 8ms/step  
Epoch 11/20  
844/844 - 7s - loss: 0.0245 - accuracy: 0.9920 - val_loss: 0.0527 -  
val_accuracy: 0.9857 - 7s/epoch - 8ms/step  
Epoch 12/20  
844/844 - 7s - loss: 0.0230 - accuracy: 0.9922 - val_loss: 0.0553 -  
val_accuracy: 0.9857 - 7s/epoch - 8ms/step  
Epoch 13/20  
844/844 - 7s - loss: 0.0208 - accuracy: 0.9927 - val_loss: 0.0594 -  
val_accuracy: 0.9843 - 7s/epoch - 8ms/step  
Epoch 14/20  
844/844 - 7s - loss: 0.0212 - accuracy: 0.9929 - val_loss: 0.0609 -  
val_accuracy: 0.9838 - 7s/epoch - 8ms/step  
Epoch 15/20  
844/844 - 7s - loss: 0.0169 - accuracy: 0.9944 - val_loss: 0.0534 -  
val_accuracy: 0.9865 - 7s/epoch - 8ms/step  
Epoch 16/20  
844/844 - 7s - loss: 0.0138 - accuracy: 0.9955 - val_loss: 0.0614 -  
val_accuracy: 0.9845 - 7s/epoch - 8ms/step  
Epoch 17/20  
844/844 - 7s - loss: 0.0143 - accuracy: 0.9950 - val_loss: 0.0580 -  
val_accuracy: 0.9872 - 7s/epoch - 8ms/step  
Epoch 18/20  
844/844 - 7s - loss: 0.0139 - accuracy: 0.9951 - val_loss: 0.0542 -  
val_accuracy: 0.9872 - 7s/epoch - 8ms/step  
Epoch 19/20  
844/844 - 7s - loss: 0.0139 - accuracy: 0.9951 - val_loss: 0.0714 -  
val_accuracy: 0.9840 - 7s/epoch - 8ms/step  
Epoch 20/20  
844/844 - 7s - loss: 0.0114 - accuracy: 0.9964 - val_loss: 0.0619 -  
val_accuracy: 0.9872 - 7s/epoch - 8ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 962 0 2 0 2 2 2 8 0 2]  
[ 0 1126 0 1 0 1 1 5 1 0]  
[ 4 1 993 6 0 1 1 22 4 0]  
[ 0 0 0 995 0 9 0 5 1 0]  
[ 1 1 1 0 963 0 2 0 0 14]]
```

```
[ 1 0 0 5 0 881 1 2 2 0]
[ 4 4 0 0 2 7 939 0 2 0]
[ 0 1 5 0 0 1 0 1020 1 0]
[ 2 0 0 2 0 4 0 2 960 4]
[ 2 0 0 2 3 11 1 8 2 980]]
```

Precision: 0.9821

Recall: 0.9819



Training Model with relu activation, 3 conv_layers, 3 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 6s - loss: 0.4083 - accuracy: 0.8688 - val_loss: 0.1017 - val_accuracy: 0.9700 - 6s/epoch - 14ms/step

Epoch 2/5

422/422 - 5s - loss: 0.1254 - accuracy: 0.9609 - val_loss: 0.0831 - val_accuracy: 0.9722 - 5s/epoch - 12ms/step

Epoch 3/5

422/422 - 5s - loss: 0.0886 - accuracy: 0.9718 - val_loss: 0.0898 - val_accuracy: 0.9720 - 5s/epoch - 12ms/step

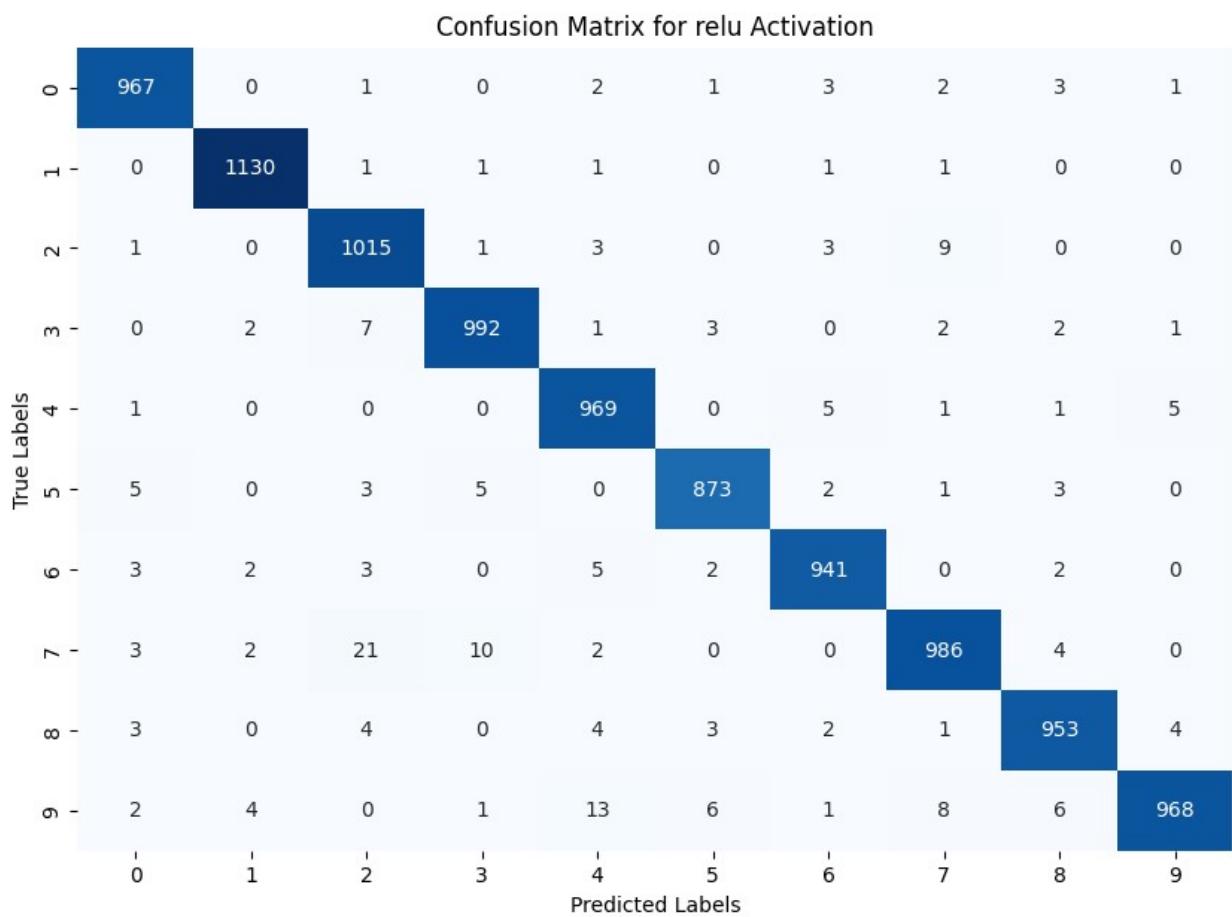
Epoch 4/5

422/422 - 5s - loss: 0.0738 - accuracy: 0.9769 - val_loss: 0.0609 -

```

val_accuracy: 0.9807 - 5s/epoch - 11ms/step
Epoch 5/5
422/422 - 5s - loss: 0.0639 - accuracy: 0.9801 - val_loss: 0.0587 -
val_accuracy: 0.9803 - 5s/epoch - 12ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: relu
Confusion Matrix:
[[ 967   0   1   0   2   1   3   2   3   1]
 [ 0 1130   1   1   1   0   1   1   0   0]
 [ 1   0 1015   1   3   0   3   9   0   0]
 [ 0   2   7 992   1   3   0   2   2   1]
 [ 1   0   0   0 969   0   5   1   1   5]
 [ 5   0   3   5   0 873   2   1   3   0]
 [ 3   2   3   0   5   2 941   0   2   0]
 [ 3   2   21  10   2   0   0 986   4   0]
 [ 3   0   4   0   4   3   2   1 953   4]
 [ 2   4   0   1 13   6   1   8   6 968]]
Precision: 0.9795
Recall: 0.9794

```



Training Model with relu activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..

Epoch 1/15
422/422 - 6s - loss: 0.3776 - accuracy: 0.8779 - val_loss: 0.1114 -
val_accuracy: 0.9630 - 6s/epoch - 14ms/step

Epoch 2/15
422/422 - 5s - loss: 0.1164 - accuracy: 0.9642 - val_loss: 0.1020 -
val_accuracy: 0.9685 - 5s/epoch - 12ms/step

Epoch 3/15
422/422 - 5s - loss: 0.0847 - accuracy: 0.9737 - val_loss: 0.0801 -
val_accuracy: 0.9740 - 5s/epoch - 12ms/step

Epoch 4/15
422/422 - 5s - loss: 0.0684 - accuracy: 0.9790 - val_loss: 0.0573 -
val_accuracy: 0.9810 - 5s/epoch - 12ms/step

Epoch 5/15
422/422 - 5s - loss: 0.0589 - accuracy: 0.9817 - val_loss: 0.0627 -
val_accuracy: 0.9822 - 5s/epoch - 12ms/step

Epoch 6/15
422/422 - 5s - loss: 0.0488 - accuracy: 0.9849 - val_loss: 0.0566 -
val_accuracy: 0.9818 - 5s/epoch - 12ms/step

Epoch 7/15
422/422 - 5s - loss: 0.0426 - accuracy: 0.9867 - val_loss: 0.0523 -
val_accuracy: 0.9848 - 5s/epoch - 12ms/step

Epoch 8/15
422/422 - 5s - loss: 0.0382 - accuracy: 0.9884 - val_loss: 0.0457 -
val_accuracy: 0.9858 - 5s/epoch - 12ms/step

Epoch 9/15
422/422 - 5s - loss: 0.0317 - accuracy: 0.9902 - val_loss: 0.0484 -
val_accuracy: 0.9860 - 5s/epoch - 12ms/step

Epoch 10/15
422/422 - 5s - loss: 0.0282 - accuracy: 0.9912 - val_loss: 0.0497 -
val_accuracy: 0.9853 - 5s/epoch - 12ms/step

Epoch 11/15
422/422 - 5s - loss: 0.0264 - accuracy: 0.9910 - val_loss: 0.0492 -
val_accuracy: 0.9870 - 5s/epoch - 12ms/step

Epoch 12/15
422/422 - 5s - loss: 0.0235 - accuracy: 0.9922 - val_loss: 0.0440 -
val_accuracy: 0.9890 - 5s/epoch - 12ms/step

Epoch 13/15
422/422 - 5s - loss: 0.0214 - accuracy: 0.9932 - val_loss: 0.0505 -
val_accuracy: 0.9872 - 5s/epoch - 12ms/step

Epoch 14/15
422/422 - 5s - loss: 0.0193 - accuracy: 0.9939 - val_loss: 0.0513 -
val_accuracy: 0.9873 - 5s/epoch - 11ms/step

Epoch 15/15
422/422 - 5s - loss: 0.0162 - accuracy: 0.9947 - val_loss: 0.0573 -
val_accuracy: 0.9850 - 5s/epoch - 11ms/step

313/313 [=====] - 1s 2ms/step

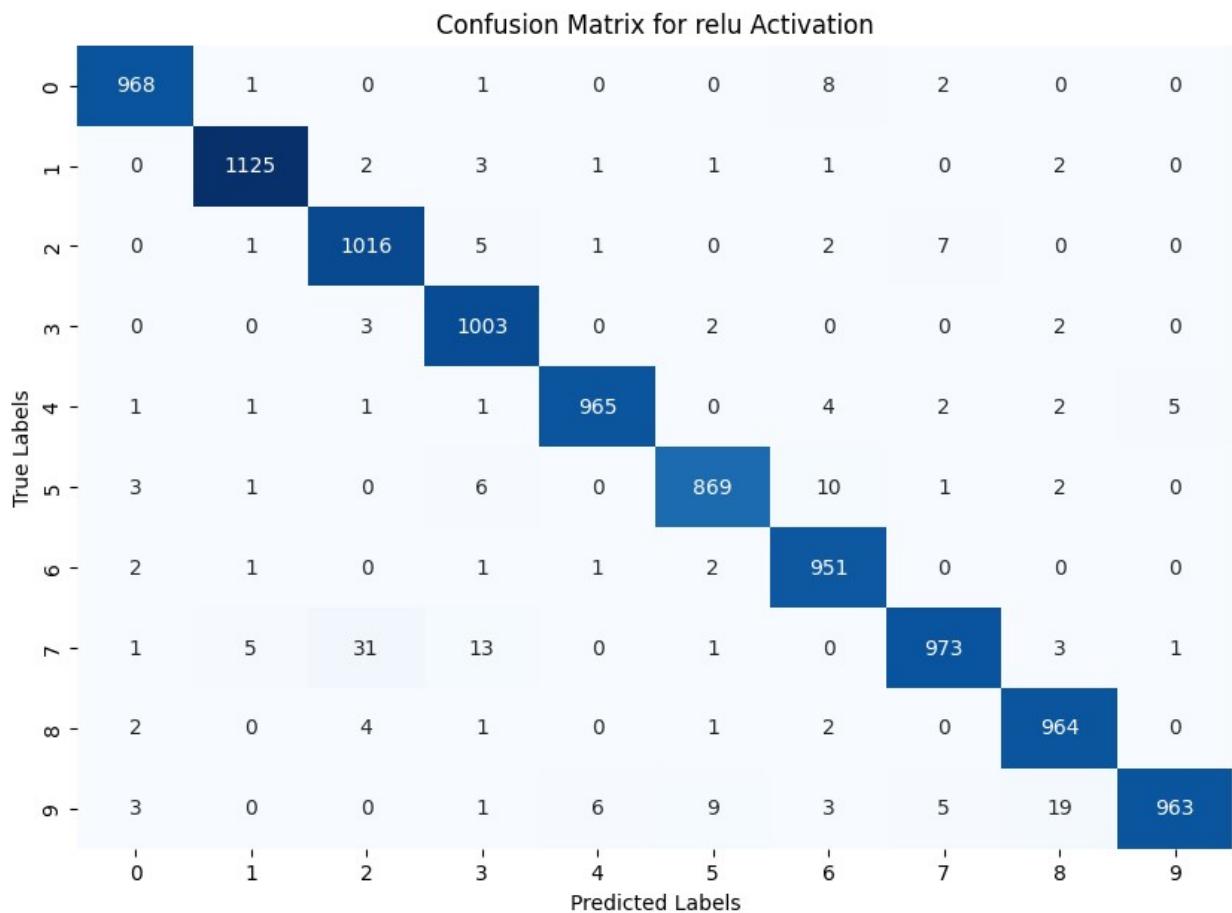
Results for activation function: relu

Confusion Matrix:

```
[[ 968  1  0  1  0  0  8  2  0  0]
 [ 0 1125  2  3  1  1  1  0  2  0]
 [ 0 1 1016  5  1  0  2  7  0  0]
 [ 0 0 3 1003  0  2  0  0  2  0]
 [ 1 1 1 1 965  0  4  2  2  5]
 [ 3 1 0 6 0 869 10 1 2 0]
 [ 2 1 0 1 1 2 951 0 0 0]
 [ 1 5 31 13 0 1 0 973 3 1]
 [ 2 0 4 1 0 1 2 0 964 0]
 [ 3 0 0 1 6 9 3 5 19 963]]
```

Precision: 0.9799

Recall: 0.9797



Training Model with relu activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..

Epoch 1/20

422/422 - 6s - loss: 0.3534 - accuracy: 0.8882 - val_loss: 0.1131 -
val_accuracy: 0.9653 - 6s/epoch - 14ms/step

Epoch 2/20

422/422 - 5s - loss: 0.1160 - accuracy: 0.9644 - val_loss: 0.0752 -
val_accuracy: 0.9788 - 5s/epoch - 12ms/step

```
Epoch 3/20
422/422 - 5s - loss: 0.0883 - accuracy: 0.9724 - val_loss: 0.0779 -
val_accuracy: 0.9762 - 5s/epoch - 12ms/step
Epoch 4/20
422/422 - 5s - loss: 0.0679 - accuracy: 0.9785 - val_loss: 0.0583 -
val_accuracy: 0.9827 - 5s/epoch - 12ms/step
Epoch 5/20
422/422 - 5s - loss: 0.0558 - accuracy: 0.9822 - val_loss: 0.0567 -
val_accuracy: 0.9840 - 5s/epoch - 12ms/step
Epoch 6/20
422/422 - 5s - loss: 0.0479 - accuracy: 0.9853 - val_loss: 0.0506 -
val_accuracy: 0.9853 - 5s/epoch - 12ms/step
Epoch 7/20
422/422 - 5s - loss: 0.0437 - accuracy: 0.9862 - val_loss: 0.0587 -
val_accuracy: 0.9820 - 5s/epoch - 12ms/step
Epoch 8/20
422/422 - 5s - loss: 0.0363 - accuracy: 0.9881 - val_loss: 0.0586 -
val_accuracy: 0.9833 - 5s/epoch - 12ms/step
Epoch 9/20
422/422 - 5s - loss: 0.0324 - accuracy: 0.9890 - val_loss: 0.0548 -
val_accuracy: 0.9843 - 5s/epoch - 12ms/step
Epoch 10/20
422/422 - 5s - loss: 0.0309 - accuracy: 0.9897 - val_loss: 0.0500 -
val_accuracy: 0.9863 - 5s/epoch - 12ms/step
Epoch 11/20
422/422 - 5s - loss: 0.0267 - accuracy: 0.9915 - val_loss: 0.0631 -
val_accuracy: 0.9823 - 5s/epoch - 12ms/step
Epoch 12/20
422/422 - 5s - loss: 0.0242 - accuracy: 0.9918 - val_loss: 0.0610 -
val_accuracy: 0.9838 - 5s/epoch - 12ms/step
Epoch 13/20
422/422 - 5s - loss: 0.0208 - accuracy: 0.9931 - val_loss: 0.0579 -
val_accuracy: 0.9855 - 5s/epoch - 12ms/step
Epoch 14/20
422/422 - 5s - loss: 0.0201 - accuracy: 0.9929 - val_loss: 0.0514 -
val_accuracy: 0.9863 - 5s/epoch - 12ms/step
Epoch 15/20
422/422 - 5s - loss: 0.0184 - accuracy: 0.9938 - val_loss: 0.0640 -
val_accuracy: 0.9853 - 5s/epoch - 11ms/step
Epoch 16/20
422/422 - 5s - loss: 0.0161 - accuracy: 0.9944 - val_loss: 0.0544 -
val_accuracy: 0.9872 - 5s/epoch - 11ms/step
Epoch 17/20
422/422 - 5s - loss: 0.0133 - accuracy: 0.9957 - val_loss: 0.0519 -
val_accuracy: 0.9868 - 5s/epoch - 12ms/step
Epoch 18/20
422/422 - 5s - loss: 0.0127 - accuracy: 0.9956 - val_loss: 0.0522 -
val_accuracy: 0.9868 - 5s/epoch - 12ms/step
Epoch 19/20
```

422/422 - 5s - loss: 0.0105 - accuracy: 0.9966 - val_loss: 0.0524 -
val_accuracy: 0.9870 - 5s/epoch - 12ms/step

Epoch 20/20

422/422 - 5s - loss: 0.0100 - accuracy: 0.9968 - val_loss: 0.0592 -
val_accuracy: 0.9848 - 5s/epoch - 12ms/step

313/313 [=====] - 1s 3ms/step

Results for activation function: relu

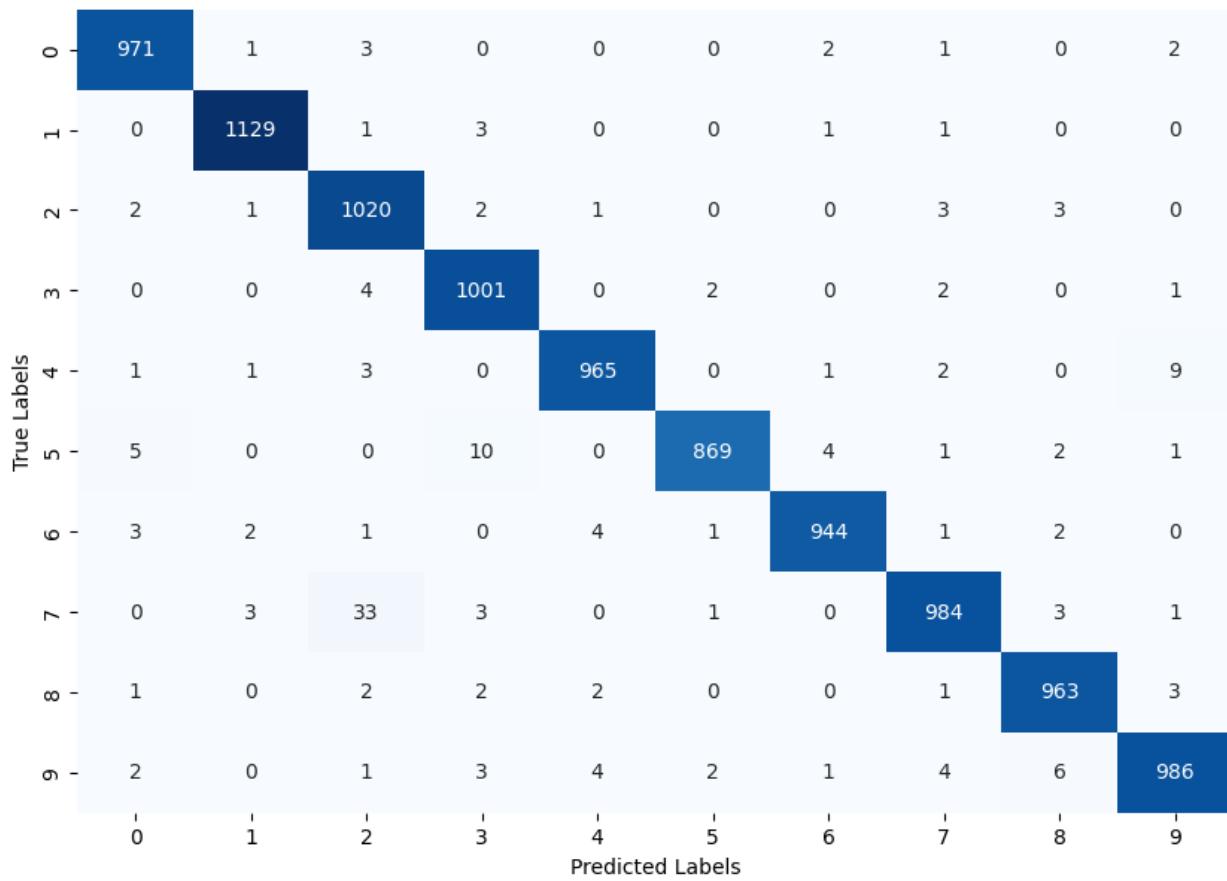
Confusion Matrix:

```
[[ 971   1   3   0   0   0   2   1   0   2]
 [ 0 1129   1   3   0   0   1   1   0   0]
 [ 2   1 1020   2   1   0   0   3   3   0]
 [ 0   0   4 1001   0   2   0   2   0   1]
 [ 1   1   3   0 965   0   1   2   0   9]
 [ 5   0   0   10   0 869   4   1   2   1]
 [ 3   2   1   0   4   1 944   1   2   0]
 [ 0   3   33   3   0   1   0 984   3   1]
 [ 1   0   2   2   2   0   0   1 963   3]
 [ 2   0   1   3   4   2   1   4   6 986]]
```

Precision: 0.9833

Recall: 0.9832

Confusion Matrix for relu Activation



```
Training Model with relu activation, 3 conv_layers, 3 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 5s - loss: 0.5220 - accuracy: 0.8317 - val_loss: 0.1493 -
val_accuracy: 0.9567 - 5s/epoch - 26ms/step
Epoch 2/5
211/211 - 4s - loss: 0.1384 - accuracy: 0.9570 - val_loss: 0.0973 -
val_accuracy: 0.9698 - 4s/epoch - 19ms/step
Epoch 3/5
211/211 - 4s - loss: 0.1049 - accuracy: 0.9672 - val_loss: 0.0904 -
val_accuracy: 0.9735 - 4s/epoch - 19ms/step
Epoch 4/5
211/211 - 4s - loss: 0.0842 - accuracy: 0.9742 - val_loss: 0.0775 -
val_accuracy: 0.9760 - 4s/epoch - 19ms/step
Epoch 5/5
211/211 - 4s - loss: 0.0729 - accuracy: 0.9769 - val_loss: 0.0767 -
val_accuracy: 0.9770 - 4s/epoch - 20ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: relu
Confusion Matrix:
[[ 958    1    3    1    5    3    7    1    1    0]
 [  0 1130    1    1    0    0    1    1    1    0]
 [  2    3 1000    4    2    1    3   13    4    0]
 [  0    0    4 1000    0    3    0    3    0    0]
 [  1    1    1    0  972    0    4    0    2    1]
 [  5    2    2    3    0  873    4    1    2    0]
 [  1    4    0    0    6    4  941    0    2    0]
 [  0    8   11   10    2    0    0  994    1    2]
 [  1    0    2    5    5    4    1    5  951    0]
 [  4    3    0    5   28   12    1    2   13  941]]
```

Precision: 0.9762
Recall: 0.9760

Confusion Matrix for relu Activation

	0	1	3	5	7	1	1	0
0	958	1	3	1	5	3	1	0
1	0	1130	1	1	0	0	1	1
2	2	3	1000	4	2	1	3	13
3	0	0	4	1000	0	3	0	3
4	1	1	1	0	972	0	4	0
5	5	2	2	3	0	873	4	1
6	1	4	0	0	6	4	941	0
7	0	8	11	10	2	0	0	994
8	1	0	2	5	5	4	1	5
9	4	3	0	5	28	12	1	2
0	1	2	3	4	5	6	7	8
								9

Training Model with relu activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..

Epoch 1/15

211/211 - 5s - loss: 0.4708 - accuracy: 0.8476 - val_loss: 0.1407 -
val_accuracy: 0.9620 - 5s/epoch - 23ms/step

Epoch 2/15

211/211 - 4s - loss: 0.1516 - accuracy: 0.9525 - val_loss: 0.1227 -
val_accuracy: 0.9637 - 4s/epoch - 19ms/step

Epoch 3/15

211/211 - 4s - loss: 0.1035 - accuracy: 0.9671 - val_loss: 0.0850 -
val_accuracy: 0.9747 - 4s/epoch - 19ms/step

Epoch 4/15

211/211 - 4s - loss: 0.0872 - accuracy: 0.9729 - val_loss: 0.0701 -
val_accuracy: 0.9798 - 4s/epoch - 19ms/step

Epoch 5/15

211/211 - 4s - loss: 0.0735 - accuracy: 0.9773 - val_loss: 0.0703 -
val_accuracy: 0.9792 - 4s/epoch - 19ms/step

Epoch 6/15

211/211 - 4s - loss: 0.0618 - accuracy: 0.9807 - val_loss: 0.0710 -
val_accuracy: 0.9770 - 4s/epoch - 19ms/step

Epoch 7/15

```
211/211 - 4s - loss: 0.0546 - accuracy: 0.9817 - val_loss: 0.0611 -  
val_accuracy: 0.9805 - 4s/epoch - 18ms/step  
Epoch 8/15  
211/211 - 4s - loss: 0.0481 - accuracy: 0.9846 - val_loss: 0.0583 -  
val_accuracy: 0.9832 - 4s/epoch - 18ms/step  
Epoch 9/15  
211/211 - 4s - loss: 0.0407 - accuracy: 0.9870 - val_loss: 0.0607 -  
val_accuracy: 0.9812 - 4s/epoch - 18ms/step  
Epoch 10/15  
211/211 - 4s - loss: 0.0390 - accuracy: 0.9872 - val_loss: 0.0643 -  
val_accuracy: 0.9808 - 4s/epoch - 19ms/step  
Epoch 11/15  
211/211 - 4s - loss: 0.0328 - accuracy: 0.9891 - val_loss: 0.0538 -  
val_accuracy: 0.9838 - 4s/epoch - 19ms/step  
Epoch 12/15  
211/211 - 4s - loss: 0.0314 - accuracy: 0.9897 - val_loss: 0.0576 -  
val_accuracy: 0.9832 - 4s/epoch - 19ms/step  
Epoch 13/15  
211/211 - 4s - loss: 0.0273 - accuracy: 0.9912 - val_loss: 0.0568 -  
val_accuracy: 0.9832 - 4s/epoch - 19ms/step  
Epoch 14/15  
211/211 - 4s - loss: 0.0272 - accuracy: 0.9911 - val_loss: 0.0582 -  
val_accuracy: 0.9832 - 4s/epoch - 18ms/step  
Epoch 15/15  
211/211 - 4s - loss: 0.0232 - accuracy: 0.9929 - val_loss: 0.0551 -  
val_accuracy: 0.9837 - 4s/epoch - 18ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 963 1 2 0 2 3 4 3 1 1 ]  
[ 0 1127 2 2 0 2 0 0 2 0 ]  
[ 2 3 1009 4 1 0 1 11 1 0 ]  
[ 0 0 3 996 0 3 0 5 2 1 ]  
[ 1 1 1 0 971 0 1 0 0 7 ]  
[ 3 0 1 6 0 873 4 2 3 0 ]  
[ 4 4 0 0 3 4 940 0 3 0 ]  
[ 0 3 9 4 0 1 0 1008 2 1 ]  
[ 1 1 4 2 0 2 0 0 963 1 ]  
[ 4 0 0 6 5 7 0 6 10 971 ]]
```

Precision: 0.9821
Recall: 0.9821

Confusion Matrix for relu Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	963	1	2	0	2	3	4	3	1	1	1
0	963	1	2	0	2	3	4	3	1	1	1
1	0	1127	2	2	0	2	0	0	2	0	0
2	2	3	1009	4	1	0	1	11	1	0	0
3	0	0	3	996	0	3	0	5	2	1	1
4	1	1	1	0	971	0	1	0	0	7	
5	3	0	1	6	0	873	4	2	3	0	
6	4	4	0	0	3	4	940	0	3	0	
7	0	3	9	4	0	1	0	1008	2	1	
8	1	1	4	2	0	2	0	0	963	1	
9	4	0	0	6	5	7	0	6	10	971	
0	1	2	3	4	5	6	7	8	9	9	

Training Model with relu activation, 3 conv_layers, 3 dense layers,
256 batch size, 20 epochs..

Epoch 1/20

211/211 - 5s - loss: 0.5636 - accuracy: 0.8177 - val_loss: 0.1329 - val_accuracy: 0.9608 - 5s/epoch - 23ms/step

Epoch 2/20

211/211 - 4s - loss: 0.1344 - accuracy: 0.9583 - val_loss: 0.1023 - val_accuracy: 0.9692 - 4s/epoch - 18ms/step

Epoch 3/20

211/211 - 4s - loss: 0.1046 - accuracy: 0.9672 - val_loss: 0.0756 - val_accuracy: 0.9773 - 4s/epoch - 18ms/step

Epoch 4/20

211/211 - 4s - loss: 0.0860 - accuracy: 0.9731 - val_loss: 0.0737 - val_accuracy: 0.9777 - 4s/epoch - 18ms/step

Epoch 5/20

211/211 - 4s - loss: 0.0753 - accuracy: 0.9769 - val_loss: 0.0703 - val_accuracy: 0.9792 - 4s/epoch - 18ms/step

Epoch 6/20

211/211 - 4s - loss: 0.0642 - accuracy: 0.9796 - val_loss: 0.0571 - val_accuracy: 0.9820 - 4s/epoch - 18ms/step

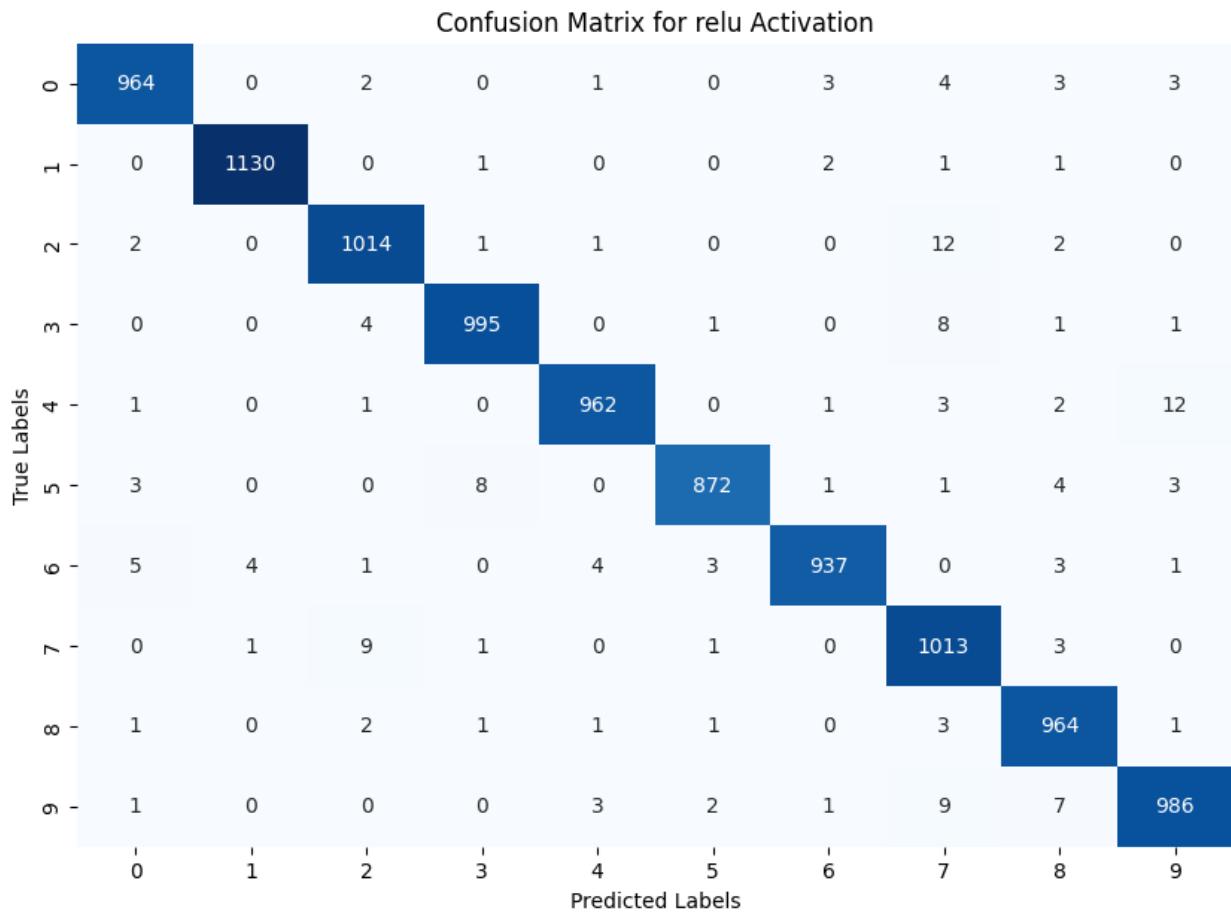
Epoch 7/20

```
211/211 - 4s - loss: 0.0575 - accuracy: 0.9818 - val_loss: 0.0641 -  
val_accuracy: 0.9827 - 4s/epoch - 18ms/step  
Epoch 8/20  
211/211 - 4s - loss: 0.0536 - accuracy: 0.9828 - val_loss: 0.0517 -  
val_accuracy: 0.9850 - 4s/epoch - 19ms/step  
Epoch 9/20  
211/211 - 4s - loss: 0.0492 - accuracy: 0.9843 - val_loss: 0.0597 -  
val_accuracy: 0.9835 - 4s/epoch - 19ms/step  
Epoch 10/20  
211/211 - 4s - loss: 0.0453 - accuracy: 0.9860 - val_loss: 0.0526 -  
val_accuracy: 0.9837 - 4s/epoch - 18ms/step  
Epoch 11/20  
211/211 - 4s - loss: 0.0406 - accuracy: 0.9876 - val_loss: 0.0601 -  
val_accuracy: 0.9810 - 4s/epoch - 18ms/step  
Epoch 12/20  
211/211 - 4s - loss: 0.0388 - accuracy: 0.9878 - val_loss: 0.0543 -  
val_accuracy: 0.9833 - 4s/epoch - 18ms/step  
Epoch 13/20  
211/211 - 4s - loss: 0.0336 - accuracy: 0.9893 - val_loss: 0.0504 -  
val_accuracy: 0.9837 - 4s/epoch - 18ms/step  
Epoch 14/20  
211/211 - 4s - loss: 0.0329 - accuracy: 0.9892 - val_loss: 0.0557 -  
val_accuracy: 0.9835 - 4s/epoch - 18ms/step  
Epoch 15/20  
211/211 - 4s - loss: 0.0294 - accuracy: 0.9900 - val_loss: 0.0594 -  
val_accuracy: 0.9818 - 4s/epoch - 19ms/step  
Epoch 16/20  
211/211 - 4s - loss: 0.0289 - accuracy: 0.9905 - val_loss: 0.0482 -  
val_accuracy: 0.9867 - 4s/epoch - 19ms/step  
Epoch 17/20  
211/211 - 4s - loss: 0.0271 - accuracy: 0.9913 - val_loss: 0.0486 -  
val_accuracy: 0.9857 - 4s/epoch - 19ms/step  
Epoch 18/20  
211/211 - 4s - loss: 0.0235 - accuracy: 0.9924 - val_loss: 0.0507 -  
val_accuracy: 0.9855 - 4s/epoch - 18ms/step  
Epoch 19/20  
211/211 - 4s - loss: 0.0224 - accuracy: 0.9927 - val_loss: 0.0502 -  
val_accuracy: 0.9858 - 4s/epoch - 18ms/step  
Epoch 20/20  
211/211 - 4s - loss: 0.0184 - accuracy: 0.9941 - val_loss: 0.0504 -  
val_accuracy: 0.9873 - 4s/epoch - 18ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: relu  
Confusion Matrix:  
[[ 964   0   2   0   1   0   3   4   3   3]  
 [  0 1130   0   1   0   0   2   1   1   0]  
 [  2   0 1014   1   1   0   0  12   2   0]  
 [  0   0   4  995   0   1   0   8   1   1]  
 [  1   0   1   0  962   0   1   3   2  12]]
```

```
[ 3 0 0 8 0 872 1 1 4 3]
[ 5 4 1 0 4 3 937 0 3 1]
[ 0 1 9 1 0 1 0 1013 3 0]
[ 1 0 2 1 1 1 0 3 964 1]
[ 1 0 0 0 3 2 1 9 7 986]]
```

Precision: 0.9838

Recall: 0.9837



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 5s - loss: 0.8715 - accuracy: 0.7657 - val_loss: 0.3311 - val_accuracy: 0.9113 - 5s/epoch - 6ms/step

Epoch 2/5

844/844 - 4s - loss: 0.3436 - accuracy: 0.9004 - val_loss: 0.2328 - val_accuracy: 0.9322 - 4s/epoch - 5ms/step

Epoch 3/5

844/844 - 5s - loss: 0.2755 - accuracy: 0.9188 - val_loss: 0.1997 - val_accuracy: 0.9455 - 5s/epoch - 5ms/step

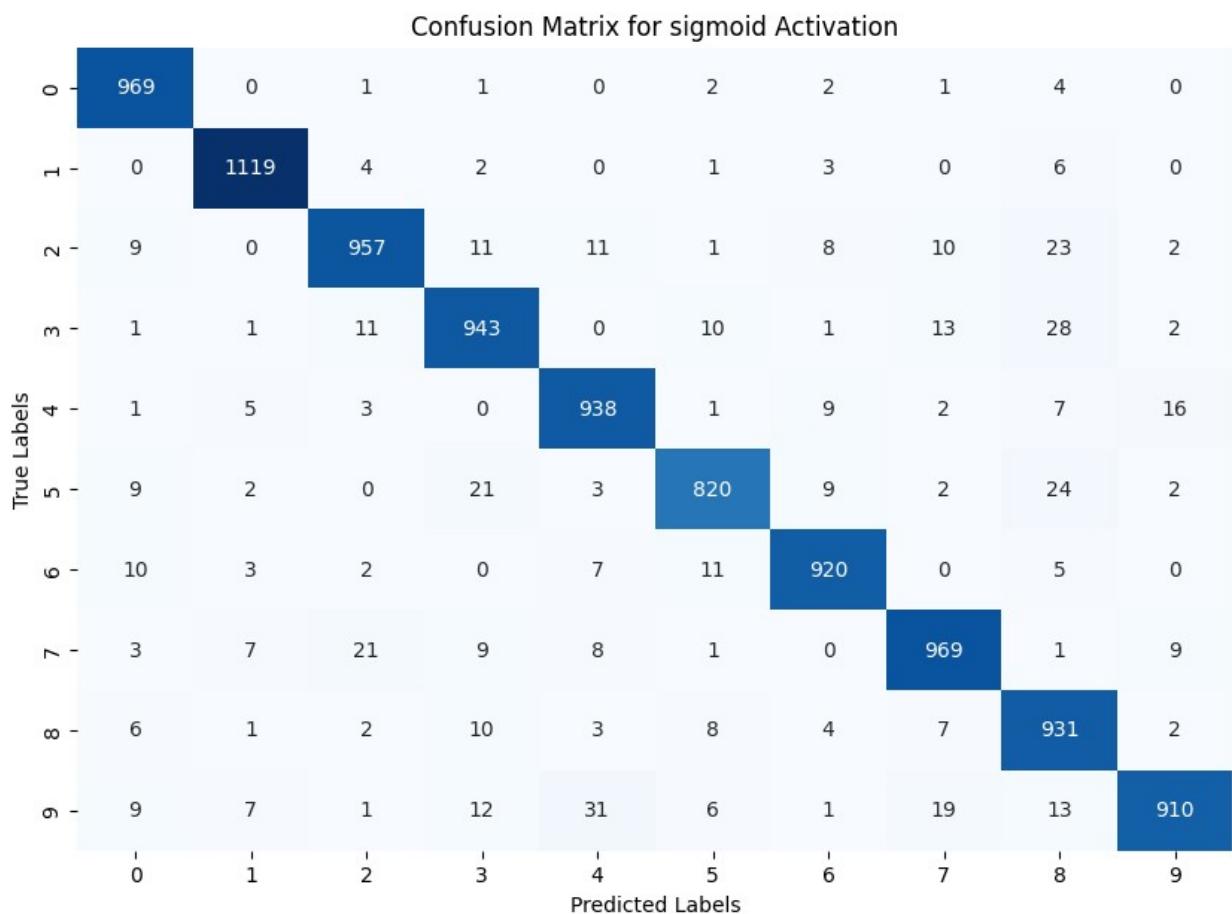
Epoch 4/5

844/844 - 5s - loss: 0.2354 - accuracy: 0.9299 - val_loss: 0.1683 -

```

val_accuracy: 0.9535 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.2062 - accuracy: 0.9395 - val_loss: 0.1478 -
val_accuracy: 0.9595 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 969   0   1   1   0   2   2   1   4   0]
 [ 0 1119   4   2   0   1   3   0   6   0]
 [ 9   0 957  11  11   1   8  10  23   2]
 [ 1   1  11 943   0  10   1  13  28   2]
 [ 1   5   3   0 938   1   9   2   7  16]
 [ 9   2   0  21   3 820   9   2  24   2]
 [ 10  3   2   0   7  11 920   0   5   0]
 [ 3   7  21   9   8   1   0 969   1   9]
 [ 6   1   2  10   3   8   4   7 931   2]
 [ 9   7   1  12  31   6   1  19  13 910]]
Precision: 0.9481
Recall: 0.9476

```



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
64 batch size, 15 epochs..

Epoch 1/15
844/844 - 5s - loss: 0.9518 - accuracy: 0.7403 - val_loss: 0.3651 -
val_accuracy: 0.9013 - 5s/epoch - 6ms/step

Epoch 2/15
844/844 - 4s - loss: 0.3593 - accuracy: 0.8961 - val_loss: 0.2474 -
val_accuracy: 0.9287 - 4s/epoch - 5ms/step

Epoch 3/15
844/844 - 4s - loss: 0.2874 - accuracy: 0.9144 - val_loss: 0.2066 -
val_accuracy: 0.9398 - 4s/epoch - 5ms/step

Epoch 4/15
844/844 - 4s - loss: 0.2474 - accuracy: 0.9258 - val_loss: 0.2024 -
val_accuracy: 0.9390 - 4s/epoch - 5ms/step

Epoch 5/15
844/844 - 4s - loss: 0.2177 - accuracy: 0.9353 - val_loss: 0.1583 -
val_accuracy: 0.9577 - 4s/epoch - 5ms/step

Epoch 6/15
844/844 - 4s - loss: 0.1923 - accuracy: 0.9438 - val_loss: 0.1396 -
val_accuracy: 0.9607 - 4s/epoch - 5ms/step

Epoch 7/15
844/844 - 4s - loss: 0.1726 - accuracy: 0.9495 - val_loss: 0.1444 -
val_accuracy: 0.9592 - 4s/epoch - 5ms/step

Epoch 8/15
844/844 - 4s - loss: 0.1572 - accuracy: 0.9546 - val_loss: 0.1183 -
val_accuracy: 0.9665 - 4s/epoch - 5ms/step

Epoch 9/15
844/844 - 4s - loss: 0.1450 - accuracy: 0.9581 - val_loss: 0.1134 -
val_accuracy: 0.9698 - 4s/epoch - 5ms/step

Epoch 10/15
844/844 - 4s - loss: 0.1320 - accuracy: 0.9620 - val_loss: 0.1059 -
val_accuracy: 0.9710 - 4s/epoch - 5ms/step

Epoch 11/15
844/844 - 5s - loss: 0.1226 - accuracy: 0.9651 - val_loss: 0.0979 -
val_accuracy: 0.9730 - 5s/epoch - 5ms/step

Epoch 12/15
844/844 - 5s - loss: 0.1134 - accuracy: 0.9678 - val_loss: 0.0927 -
val_accuracy: 0.9752 - 5s/epoch - 5ms/step

Epoch 13/15
844/844 - 4s - loss: 0.1056 - accuracy: 0.9698 - val_loss: 0.0933 -
val_accuracy: 0.9752 - 4s/epoch - 5ms/step

Epoch 14/15
844/844 - 4s - loss: 0.0993 - accuracy: 0.9716 - val_loss: 0.0845 -
val_accuracy: 0.9773 - 4s/epoch - 5ms/step

Epoch 15/15
844/844 - 4s - loss: 0.0935 - accuracy: 0.9735 - val_loss: 0.0800 -
val_accuracy: 0.9778 - 4s/epoch - 5ms/step

313/313 [=====] - 1s 2ms/step

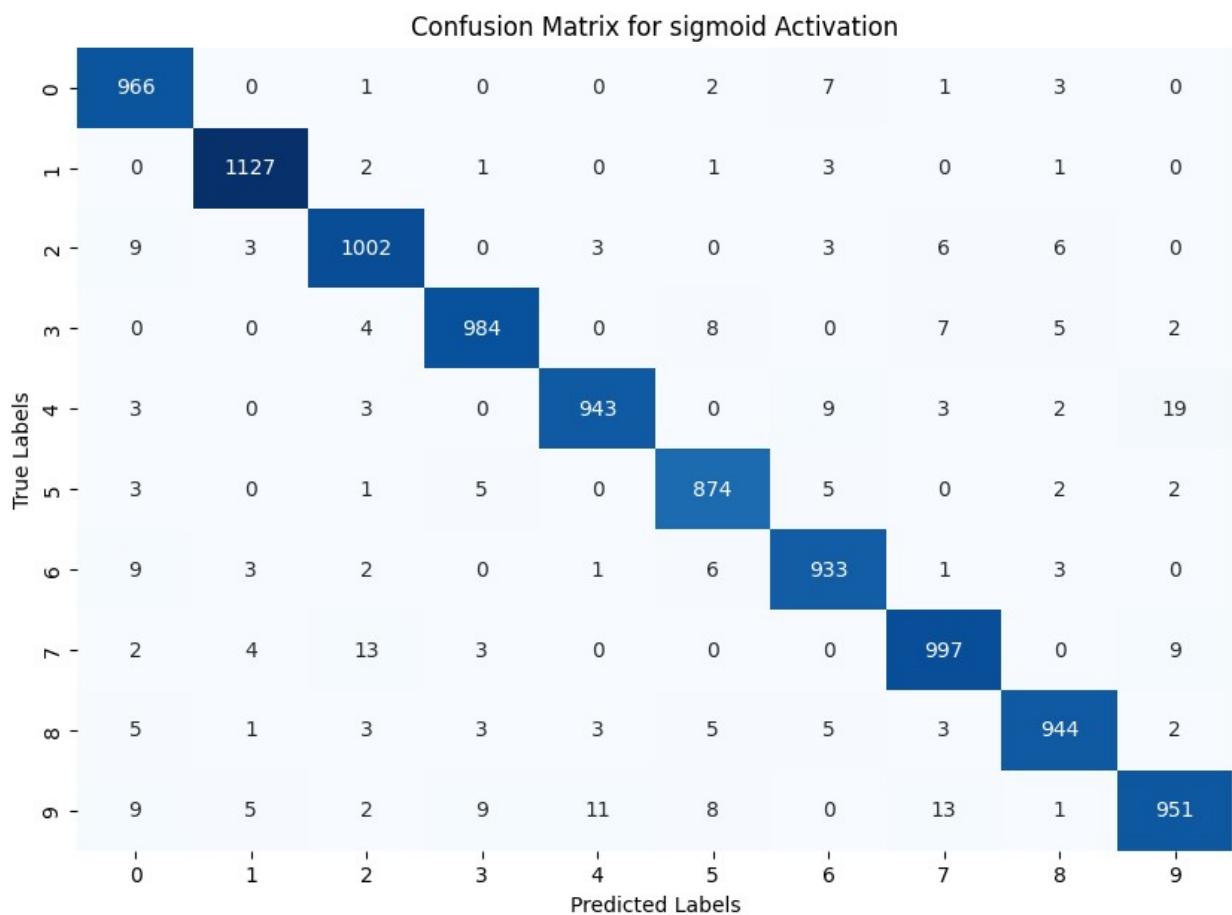
Results for activation function: sigmoid

Confusion Matrix:

```
[[ 966  0   1   0   0   2   7   1   3   0]
 [ 0 1127  2   1   0   1   3   0   1   0]
 [ 9   3 1002  0   3   0   3   6   6   0]
 [ 0   0   4 984  0   8   0   7   5   2]
 [ 3   0   3   0 943  0   9   3   2 19]
 [ 3   0   1   5   0 874  5   0   2   2]
 [ 9   3   2   0   1   6 933  1   3   0]
 [ 2   4 13   3   0   0   0 997  0   9]
 [ 5   1   3   3   3   5   5   3 944  2]
 [ 9   5   2   9 11   8   0 13   1 951]]
```

Precision: 0.9721

Recall: 0.9721



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 64 batch size, 20 epochs..

Epoch 1/20

844/844 - 5s - loss: 0.9043 - accuracy: 0.7487 - val_loss: 0.3261 - val_accuracy: 0.9142 - 5s/epoch - 6ms/step

Epoch 2/20

844/844 - 5s - loss: 0.3562 - accuracy: 0.8957 - val_loss: 0.2560 - val_accuracy: 0.9250 - 5s/epoch - 5ms/step

```
Epoch 3/20
844/844 - 4s - loss: 0.2896 - accuracy: 0.9132 - val_loss: 0.2155 -
val_accuracy: 0.9367 - 4s/epoch - 5ms/step
Epoch 4/20
844/844 - 4s - loss: 0.2530 - accuracy: 0.9246 - val_loss: 0.1821 -
val_accuracy: 0.9475 - 4s/epoch - 5ms/step
Epoch 5/20
844/844 - 4s - loss: 0.2231 - accuracy: 0.9329 - val_loss: 0.1653 -
val_accuracy: 0.9527 - 4s/epoch - 5ms/step
Epoch 6/20
844/844 - 4s - loss: 0.2003 - accuracy: 0.9398 - val_loss: 0.1517 -
val_accuracy: 0.9575 - 4s/epoch - 5ms/step
Epoch 7/20
844/844 - 4s - loss: 0.1818 - accuracy: 0.9461 - val_loss: 0.1473 -
val_accuracy: 0.9585 - 4s/epoch - 5ms/step
Epoch 8/20
844/844 - 4s - loss: 0.1634 - accuracy: 0.9522 - val_loss: 0.1288 -
val_accuracy: 0.9632 - 4s/epoch - 5ms/step
Epoch 9/20
844/844 - 4s - loss: 0.1505 - accuracy: 0.9557 - val_loss: 0.1190 -
val_accuracy: 0.9667 - 4s/epoch - 5ms/step
Epoch 10/20
844/844 - 4s - loss: 0.1376 - accuracy: 0.9599 - val_loss: 0.1148 -
val_accuracy: 0.9675 - 4s/epoch - 5ms/step
Epoch 11/20
844/844 - 5s - loss: 0.1277 - accuracy: 0.9636 - val_loss: 0.1079 -
val_accuracy: 0.9700 - 5s/epoch - 5ms/step
Epoch 12/20
844/844 - 4s - loss: 0.1186 - accuracy: 0.9659 - val_loss: 0.1005 -
val_accuracy: 0.9722 - 4s/epoch - 5ms/step
Epoch 13/20
844/844 - 4s - loss: 0.1112 - accuracy: 0.9686 - val_loss: 0.0949 -
val_accuracy: 0.9742 - 4s/epoch - 5ms/step
Epoch 14/20
844/844 - 4s - loss: 0.1040 - accuracy: 0.9708 - val_loss: 0.0972 -
val_accuracy: 0.9733 - 4s/epoch - 5ms/step
Epoch 15/20
844/844 - 5s - loss: 0.0984 - accuracy: 0.9727 - val_loss: 0.0850 -
val_accuracy: 0.9768 - 5s/epoch - 5ms/step
Epoch 16/20
844/844 - 4s - loss: 0.0915 - accuracy: 0.9749 - val_loss: 0.0895 -
val_accuracy: 0.9758 - 4s/epoch - 5ms/step
Epoch 17/20
844/844 - 4s - loss: 0.0873 - accuracy: 0.9759 - val_loss: 0.0765 -
val_accuracy: 0.9782 - 4s/epoch - 5ms/step
Epoch 18/20
844/844 - 4s - loss: 0.0825 - accuracy: 0.9768 - val_loss: 0.0816 -
val_accuracy: 0.9782 - 4s/epoch - 5ms/step
Epoch 19/20
```

```
844/844 - 5s - loss: 0.0783 - accuracy: 0.9782 - val_loss: 0.0754 -  
val_accuracy: 0.9792 - 5s/epoch - 5ms/step
```

```
Epoch 20/20
```

```
844/844 - 5s - loss: 0.0740 - accuracy: 0.9792 - val_loss: 0.0746 -  
val_accuracy: 0.9797 - 5s/epoch - 5ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: sigmoid
```

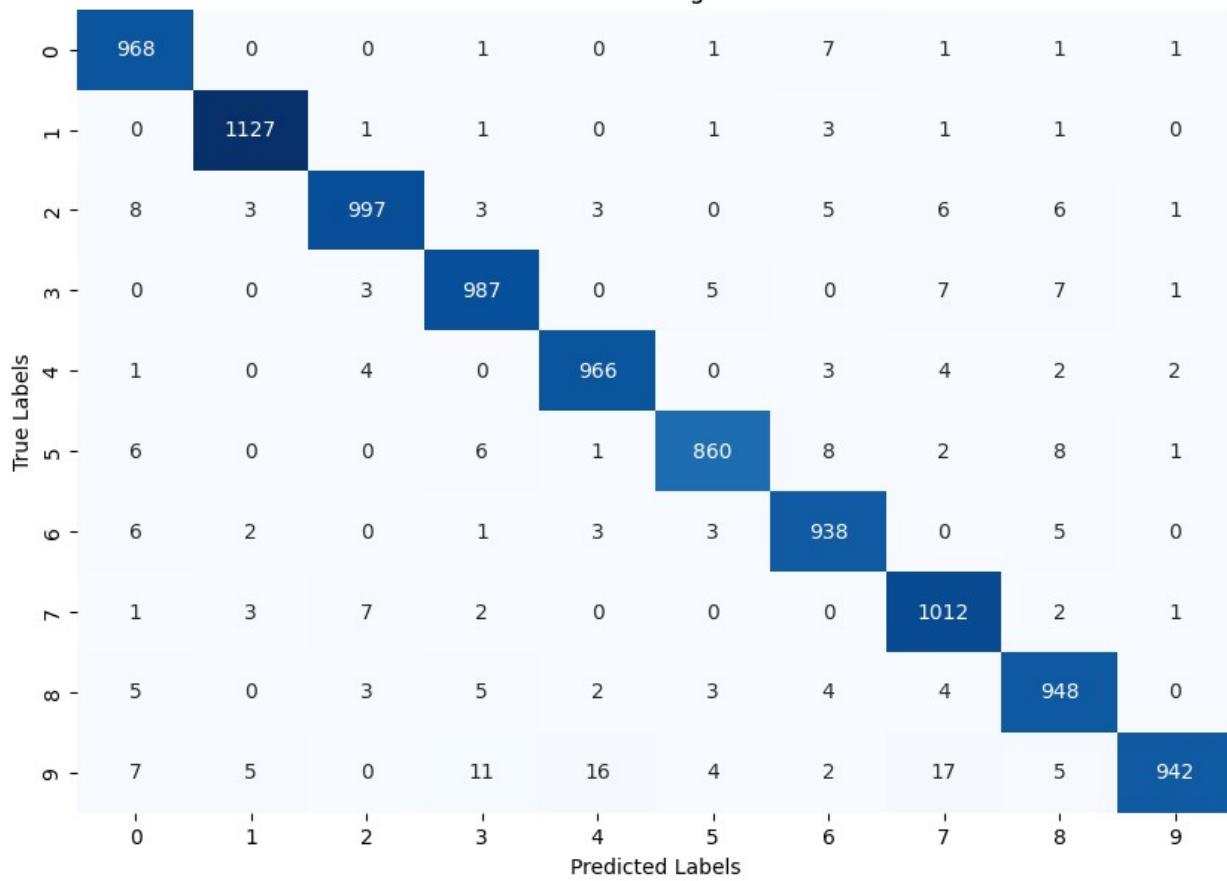
```
Confusion Matrix:
```

```
[[ 968   0   0   1   0   1   7   1   1   1  
[   0 1127   1   1   0   1   3   1   1   0]  
[   8   3 997   3   3   0   5   6   6   1]  
[   0   0   3 987   0   5   0   7   7   1]  
[   1   0   4   0 966   0   3   4   2   2]  
[   6   0   0   6   1 860   8   2   8   1]  
[   6   2   0   1   3   3 938   0   5   0]  
[   1   3   7   2   0   0   0 1012   2   1]  
[   5   0   3   5   2   3   4   4 948   0]  
[   7   5   0 11 16   4   2 17   5 942]]
```

```
Precision: 0.9747
```

```
Recall: 0.9745
```

Confusion Matrix for sigmoid Activation



```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 4s - loss: 1.2424 - accuracy: 0.6650 - val_loss: 0.4906 -
val_accuracy: 0.8907 - 4s/epoch - 9ms/step
Epoch 2/5
422/422 - 3s - loss: 0.4542 - accuracy: 0.8804 - val_loss: 0.3217 -
val_accuracy: 0.9160 - 3s/epoch - 7ms/step
Epoch 3/5
422/422 - 3s - loss: 0.3528 - accuracy: 0.8994 - val_loss: 0.2702 -
val_accuracy: 0.9262 - 3s/epoch - 7ms/step
Epoch 4/5
422/422 - 3s - loss: 0.3068 - accuracy: 0.9114 - val_loss: 0.2301 -
val_accuracy: 0.9363 - 3s/epoch - 7ms/step
Epoch 5/5
422/422 - 3s - loss: 0.2765 - accuracy: 0.9186 - val_loss: 0.2106 -
val_accuracy: 0.9433 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 960     0     1     1     0     6     8     1     3     0]
 [  0 1118     3     2     0     2     4     1     5     0]
 [ 10     2 943     9    10     3    19    16    15     5]
 [  4     1   21 913     0    31     3    19     9     9]
 [  1     5     3     0 910     2    15     2     2    42]
 [  7     3     1   32     5 803    17     5    12     7]
 [  8     3     4     0     5    14 920     2     2     0]
 [  1     9   21     4     6     0     0 964     0    23]
 [  6     5     9   22     8    32    11    14 851    16]
 [ 11     6     2   10    26     9     1    25     4 915]]
```

Precision: 0.9298
Recall: 0.9297

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	960	0	1	1	0	6	8	1	3	0	
1	0	1118	3	2	0	2	4	1	5	0	
2	10	2	943	9	10	3	19	16	15	5	
3	4	1	21	913	0	31	3	19	9	9	
4	1	5	3	0	910	2	15	2	2	42	
5	7	3	1	32	5	803	17	5	12	7	
6	8	3	4	0	5	14	920	2	2	0	
7	1	9	21	4	6	0	0	964	0	23	
8	6	5	9	22	8	32	11	14	851	16	
9	11	6	2	10	26	9	1	25	4	915	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 4s - loss: 1.1775 - accuracy: 0.6974 - val_loss: 0.4643 -
val_accuracy: 0.9002 - 4s/epoch - 10ms/step
Epoch 2/15
422/422 - 3s - loss: 0.4456 - accuracy: 0.8815 - val_loss: 0.3093 -
val_accuracy: 0.9178 - 3s/epoch - 7ms/step
Epoch 3/15
422/422 - 3s - loss: 0.3537 - accuracy: 0.8991 - val_loss: 0.2694 -
val_accuracy: 0.9240 - 3s/epoch - 7ms/step
Epoch 4/15
422/422 - 3s - loss: 0.3104 - accuracy: 0.9103 - val_loss: 0.2308 -
val_accuracy: 0.9328 - 3s/epoch - 7ms/step
Epoch 5/15
422/422 - 3s - loss: 0.2834 - accuracy: 0.9165 - val_loss: 0.2137 -
val_accuracy: 0.9392 - 3s/epoch - 7ms/step
Epoch 6/15
422/422 - 3s - loss: 0.2609 - accuracy: 0.9229 - val_loss: 0.1987 -
val_accuracy: 0.9420 - 3s/epoch - 7ms/step
Epoch 7/15
```

```
422/422 - 3s - loss: 0.2437 - accuracy: 0.9281 - val_loss: 0.1938 -  
val_accuracy: 0.9457 - 3s/epoch - 7ms/step  
Epoch 8/15  
422/422 - 3s - loss: 0.2283 - accuracy: 0.9320 - val_loss: 0.1911 -  
val_accuracy: 0.9452 - 3s/epoch - 7ms/step  
Epoch 9/15  
422/422 - 3s - loss: 0.2150 - accuracy: 0.9362 - val_loss: 0.1695 -  
val_accuracy: 0.9530 - 3s/epoch - 7ms/step  
Epoch 10/15  
422/422 - 3s - loss: 0.2049 - accuracy: 0.9393 - val_loss: 0.1619 -  
val_accuracy: 0.9552 - 3s/epoch - 7ms/step  
Epoch 11/15  
422/422 - 3s - loss: 0.1945 - accuracy: 0.9421 - val_loss: 0.1551 -  
val_accuracy: 0.9585 - 3s/epoch - 7ms/step  
Epoch 12/15  
422/422 - 3s - loss: 0.1857 - accuracy: 0.9453 - val_loss: 0.1459 -  
val_accuracy: 0.9598 - 3s/epoch - 7ms/step  
Epoch 13/15  
422/422 - 3s - loss: 0.1764 - accuracy: 0.9474 - val_loss: 0.1435 -  
val_accuracy: 0.9600 - 3s/epoch - 7ms/step  
Epoch 14/15  
422/422 - 3s - loss: 0.1699 - accuracy: 0.9510 - val_loss: 0.1414 -  
val_accuracy: 0.9632 - 3s/epoch - 7ms/step  
Epoch 15/15  
422/422 - 3s - loss: 0.1617 - accuracy: 0.9526 - val_loss: 0.1290 -  
val_accuracy: 0.9657 - 3s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 969  0  0  1  0  2  2  2  4  0]  
[ 0 1111  3  2  0  1  5  2 11  0]  
[ 9  0 957 11  8  2  5 17 21  2]  
[ 1  0  3 964  0  8  0 15 13  6]  
[ 1  2  6  0 928  0  4  6  2 33]  
[ 8  2  1 14  2 835  7  3 14  6]  
[ 10 3  2  1  7 10 917  2  6  0]  
[ 1  5 15  6  3  0  0 980  1 17]  
[ 5  1  1 14  4  6  5  9 924  5]  
[ 9  5  1 14 12  5  0 16  3 944]]  
Precision: 0.9532  
Recall: 0.9529
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	969	0	0	1	0	2	2	2	4	0	
1	0	1111	3	2	0	1	5	2	11	0	
2	9	0	957	11	8	2	5	17	21	2	
3	1	0	3	964	0	8	0	15	13	6	
4	1	2	6	0	928	0	4	6	2	33	
5	8	2	1	14	2	835	7	3	14	6	
6	10	3	2	1	7	10	917	2	6	0	
7	1	5	15	6	3	0	0	980	1	17	
8	5	1	1	14	4	6	5	9	924	5	
9	9	5	1	14	12	5	0	16	3	944	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

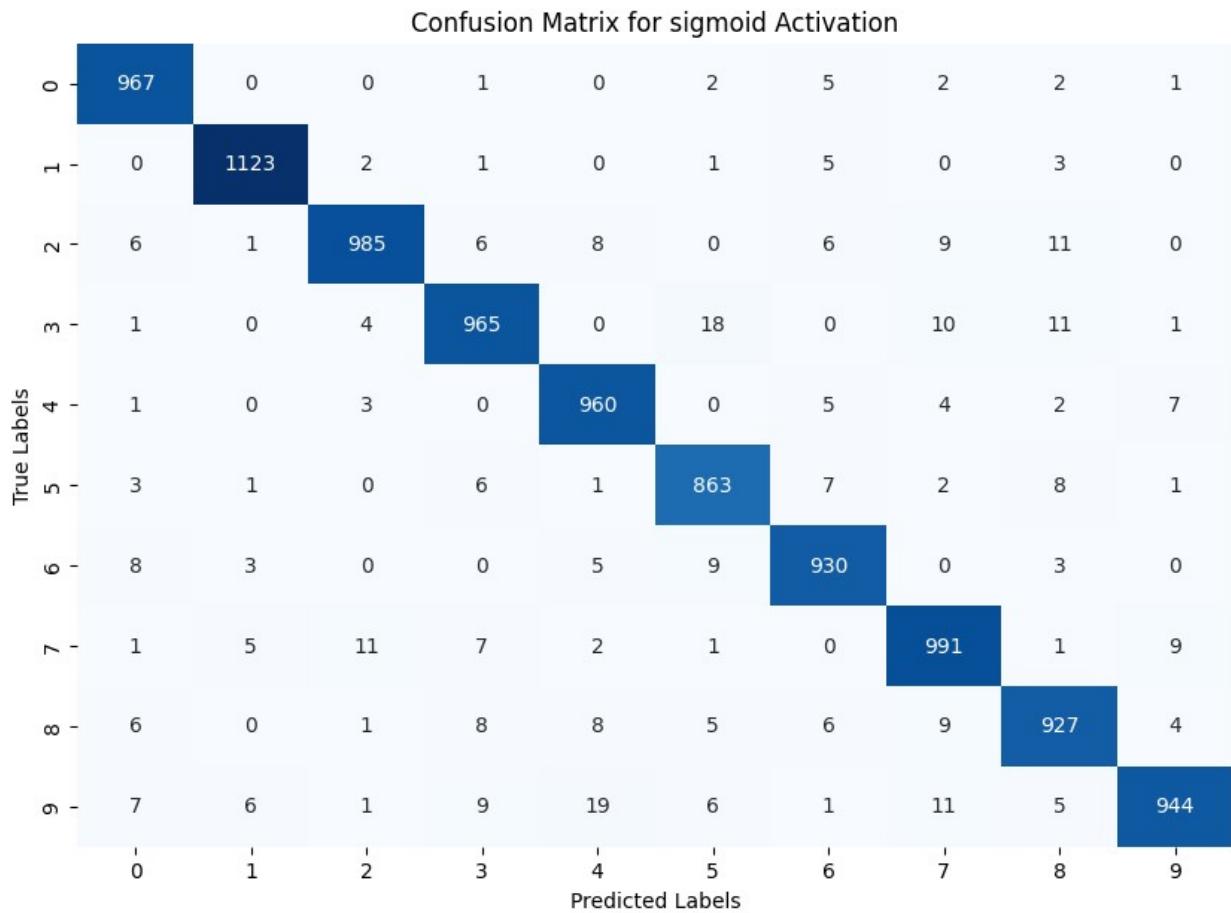
```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 4s - loss: 1.2282 - accuracy: 0.6797 - val_loss: 0.4928 -
val_accuracy: 0.8880 - 4s/epoch - 9ms/step
Epoch 2/20
422/422 - 3s - loss: 0.4479 - accuracy: 0.8818 - val_loss: 0.3156 -
val_accuracy: 0.9143 - 3s/epoch - 7ms/step
Epoch 3/20
422/422 - 3s - loss: 0.3505 - accuracy: 0.9004 - val_loss: 0.2657 -
val_accuracy: 0.9240 - 3s/epoch - 7ms/step
Epoch 4/20
422/422 - 3s - loss: 0.3069 - accuracy: 0.9102 - val_loss: 0.2347 -
val_accuracy: 0.9332 - 3s/epoch - 7ms/step
Epoch 5/20
422/422 - 3s - loss: 0.2781 - accuracy: 0.9183 - val_loss: 0.2094 -
val_accuracy: 0.9398 - 3s/epoch - 7ms/step
Epoch 6/20
422/422 - 3s - loss: 0.2558 - accuracy: 0.9242 - val_loss: 0.1946 -
val_accuracy: 0.9438 - 3s/epoch - 7ms/step
Epoch 7/20
```

```
422/422 - 3s - loss: 0.2376 - accuracy: 0.9302 - val_loss: 0.1958 -  
val_accuracy: 0.9417 - 3s/epoch - 7ms/step  
Epoch 8/20  
422/422 - 3s - loss: 0.2218 - accuracy: 0.9350 - val_loss: 0.1740 -  
val_accuracy: 0.9520 - 3s/epoch - 7ms/step  
Epoch 9/20  
422/422 - 3s - loss: 0.2057 - accuracy: 0.9402 - val_loss: 0.1595 -  
val_accuracy: 0.9545 - 3s/epoch - 7ms/step  
Epoch 10/20  
422/422 - 3s - loss: 0.1941 - accuracy: 0.9436 - val_loss: 0.1506 -  
val_accuracy: 0.9585 - 3s/epoch - 7ms/step  
Epoch 11/20  
422/422 - 3s - loss: 0.1831 - accuracy: 0.9467 - val_loss: 0.1429 -  
val_accuracy: 0.9618 - 3s/epoch - 7ms/step  
Epoch 12/20  
422/422 - 3s - loss: 0.1748 - accuracy: 0.9485 - val_loss: 0.1358 -  
val_accuracy: 0.9637 - 3s/epoch - 7ms/step  
Epoch 13/20  
422/422 - 3s - loss: 0.1658 - accuracy: 0.9516 - val_loss: 0.1299 -  
val_accuracy: 0.9642 - 3s/epoch - 7ms/step  
Epoch 14/20  
422/422 - 3s - loss: 0.1576 - accuracy: 0.9541 - val_loss: 0.1282 -  
val_accuracy: 0.9627 - 3s/epoch - 7ms/step  
Epoch 15/20  
422/422 - 3s - loss: 0.1501 - accuracy: 0.9562 - val_loss: 0.1205 -  
val_accuracy: 0.9668 - 3s/epoch - 7ms/step  
Epoch 16/20  
422/422 - 3s - loss: 0.1435 - accuracy: 0.9584 - val_loss: 0.1113 -  
val_accuracy: 0.9715 - 3s/epoch - 7ms/step  
Epoch 17/20  
422/422 - 3s - loss: 0.1363 - accuracy: 0.9605 - val_loss: 0.1113 -  
val_accuracy: 0.9697 - 3s/epoch - 7ms/step  
Epoch 18/20  
422/422 - 3s - loss: 0.1316 - accuracy: 0.9622 - val_loss: 0.1079 -  
val_accuracy: 0.9697 - 3s/epoch - 7ms/step  
Epoch 19/20  
422/422 - 3s - loss: 0.1257 - accuracy: 0.9644 - val_loss: 0.1024 -  
val_accuracy: 0.9730 - 3s/epoch - 7ms/step  
Epoch 20/20  
422/422 - 3s - loss: 0.1215 - accuracy: 0.9652 - val_loss: 0.0984 -  
val_accuracy: 0.9735 - 3s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 967 0 0 1 0 2 5 2 2 1]  
[ 0 1123 2 1 0 1 5 0 3 0]  
[ 6 1 985 6 8 0 6 9 11 0]  
[ 1 0 4 965 0 18 0 10 11 1]  
[ 1 0 3 0 960 0 5 4 2 7]]
```

```
[ 3 1 0 6 1 863 7 2 8 1]
[ 8 3 0 0 5 9 930 0 3 0]
[ 1 5 11 7 2 1 0 991 1 9]
[ 6 0 1 8 8 5 6 9 927 4]
[ 7 6 1 9 19 6 1 11 5 944]]
```

Precision: 0.9656

Recall: 0.9655



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 3s - loss: 1.6858 - accuracy: 0.5660 - val_loss: 0.9670 - val_accuracy: 0.8308 - 3s/epoch - 15ms/step

Epoch 2/5

211/211 - 3s - loss: 0.7412 - accuracy: 0.8276 - val_loss: 0.4887 - val_accuracy: 0.8975 - 3s/epoch - 12ms/step

Epoch 3/5

211/211 - 3s - loss: 0.4941 - accuracy: 0.8738 - val_loss: 0.3689 - val_accuracy: 0.9127 - 3s/epoch - 12ms/step

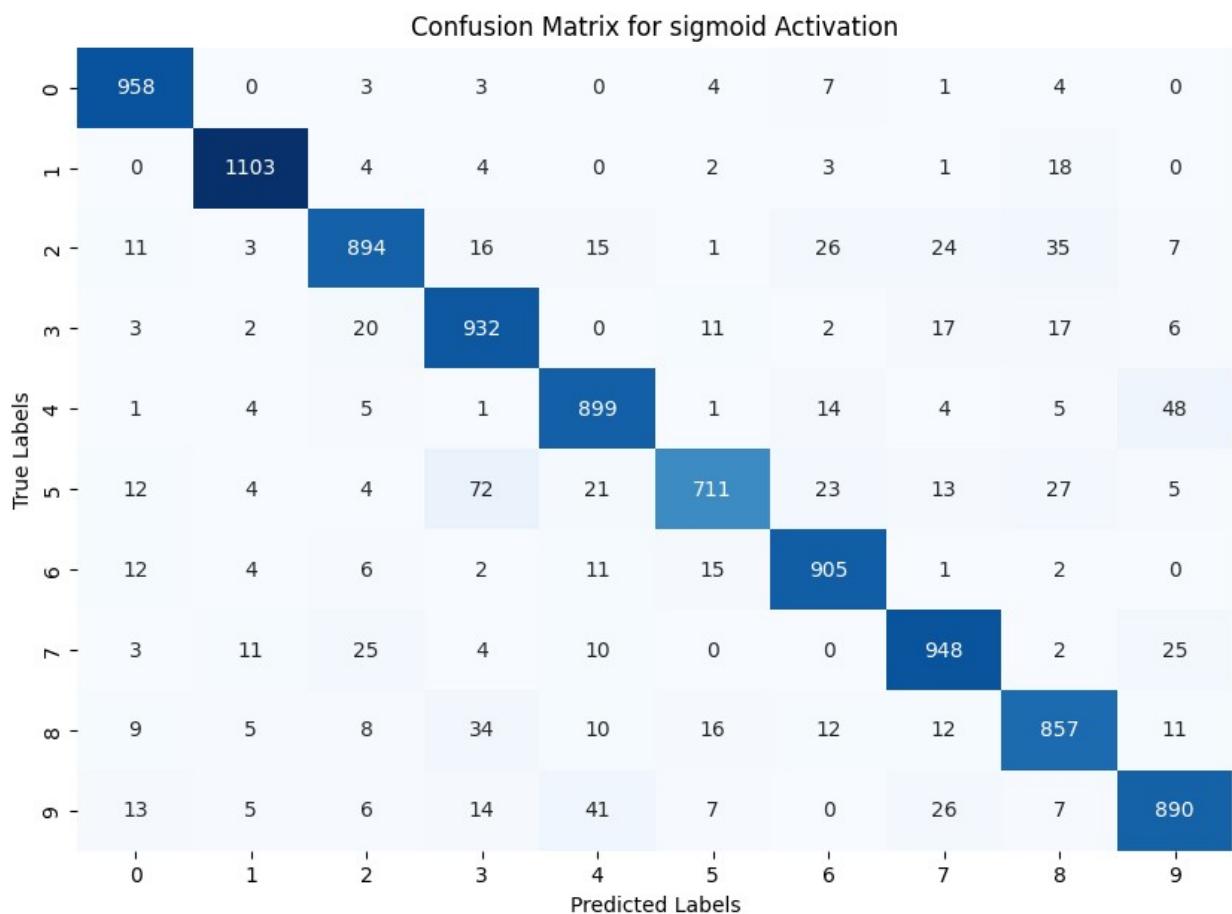
Epoch 4/5

211/211 - 2s - loss: 0.4078 - accuracy: 0.8899 - val_loss: 0.3120 -

```

val_accuracy: 0.9172 - 2s/epoch - 11ms/step
Epoch 5/5
211/211 - 2s - loss: 0.3629 - accuracy: 0.8981 - val_loss: 0.2790 -
val_accuracy: 0.9258 - 2s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 958   0   3   3   0   4   7   1   4   0]
 [  0 1103   4   4   0   2   3   1 18   0]
 [ 11   3 894  16  15   1  26  24  35   7]
 [  3   2 20 932   0  11   2  17  17   6]
 [  1   4   5   1 899   1  14   4   5 48]
 [ 12   4   4  72  21 711  23  13  27   5]
 [ 12   4   6   2 11  15 905   1   2   0]
 [  3  11  25   4 10   0   0 948   2 25]
 [  9   5   8  34  10  16   12  12 857 11]
 [ 13   5   6  14  41   7   0  26   7 890]]
Precision: 0.9102
Recall: 0.9097

```

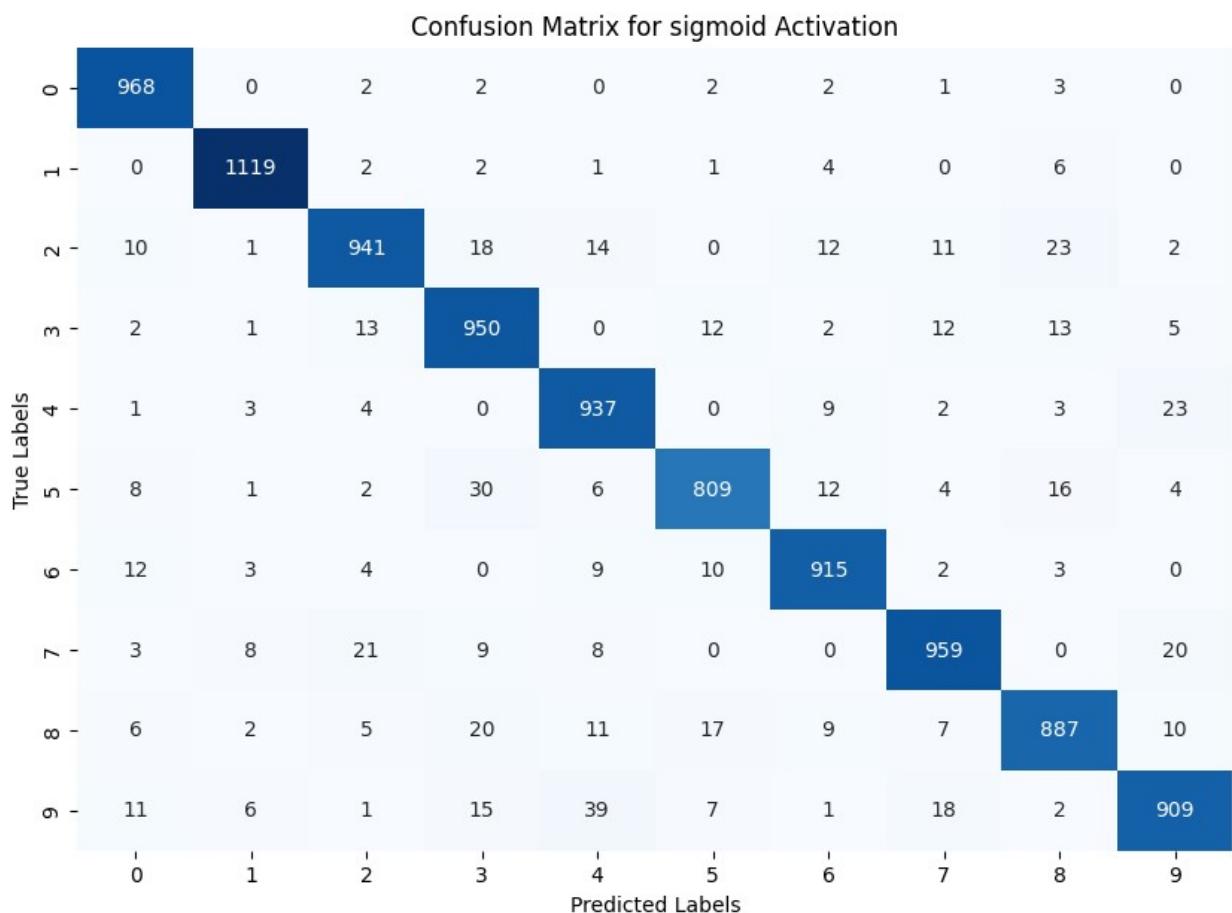


```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 3s - loss: 1.6750 - accuracy: 0.5735 - val_loss: 0.9136 -
val_accuracy: 0.8348 - 3s/epoch - 15ms/step
Epoch 2/15
211/211 - 2s - loss: 0.7190 - accuracy: 0.8308 - val_loss: 0.4891 -
val_accuracy: 0.8888 - 2s/epoch - 12ms/step
Epoch 3/15
211/211 - 3s - loss: 0.4881 - accuracy: 0.8731 - val_loss: 0.3669 -
val_accuracy: 0.9048 - 3s/epoch - 12ms/step
Epoch 4/15
211/211 - 3s - loss: 0.4010 - accuracy: 0.8912 - val_loss: 0.3142 -
val_accuracy: 0.9187 - 3s/epoch - 12ms/step
Epoch 5/15
211/211 - 3s - loss: 0.3586 - accuracy: 0.8990 - val_loss: 0.2836 -
val_accuracy: 0.9220 - 3s/epoch - 12ms/step
Epoch 6/15
211/211 - 2s - loss: 0.3288 - accuracy: 0.9063 - val_loss: 0.2577 -
val_accuracy: 0.9267 - 2s/epoch - 12ms/step
Epoch 7/15
211/211 - 3s - loss: 0.3075 - accuracy: 0.9114 - val_loss: 0.2401 -
val_accuracy: 0.9325 - 3s/epoch - 12ms/step
Epoch 8/15
211/211 - 3s - loss: 0.2899 - accuracy: 0.9157 - val_loss: 0.2274 -
val_accuracy: 0.9373 - 3s/epoch - 12ms/step
Epoch 9/15
211/211 - 3s - loss: 0.2763 - accuracy: 0.9194 - val_loss: 0.2207 -
val_accuracy: 0.9375 - 3s/epoch - 12ms/step
Epoch 10/15
211/211 - 3s - loss: 0.2635 - accuracy: 0.9228 - val_loss: 0.2132 -
val_accuracy: 0.9402 - 3s/epoch - 12ms/step
Epoch 11/15
211/211 - 3s - loss: 0.2527 - accuracy: 0.9263 - val_loss: 0.2001 -
val_accuracy: 0.9428 - 3s/epoch - 12ms/step
Epoch 12/15
211/211 - 2s - loss: 0.2430 - accuracy: 0.9294 - val_loss: 0.1869 -
val_accuracy: 0.9465 - 2s/epoch - 11ms/step
Epoch 13/15
211/211 - 2s - loss: 0.2340 - accuracy: 0.9316 - val_loss: 0.1822 -
val_accuracy: 0.9490 - 2s/epoch - 12ms/step
Epoch 14/15
211/211 - 2s - loss: 0.2242 - accuracy: 0.9341 - val_loss: 0.1831 -
val_accuracy: 0.9465 - 2s/epoch - 11ms/step
Epoch 15/15
211/211 - 2s - loss: 0.2181 - accuracy: 0.9354 - val_loss: 0.1674 -
val_accuracy: 0.9530 - 2s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
```

```
[[ 968  0  2  2  0  2  2  1  3  0]
 [ 0 1119  2  2  1  1  4  0  6  0]
 [ 10  1 941 18 14  0 12 11 23  2]
 [ 2  1 13 950  0 12  2 12 13  5]
 [ 1  3  4  0 937  0  9  2  3 23]
 [ 8  1  2 30  6 809 12  4 16  4]
 [ 12 3  4  0  9 10 915  2  3  0]
 [ 3  8 21  9  8  0  0 959  0 20]
 [ 6  2  5 20 11 17  9  7 887 10]
 [ 11 6  1 15 39  7  1 18  2 909]]
```

Precision: 0.9395

Recall: 0.9394



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 256 batch size, 20 epochs..

Epoch 1/20

211/211 - 3s - loss: 1.7849 - accuracy: 0.5219 - val_loss: 1.0802 - val_accuracy: 0.7622 - 3s/epoch - 14ms/step

Epoch 2/20

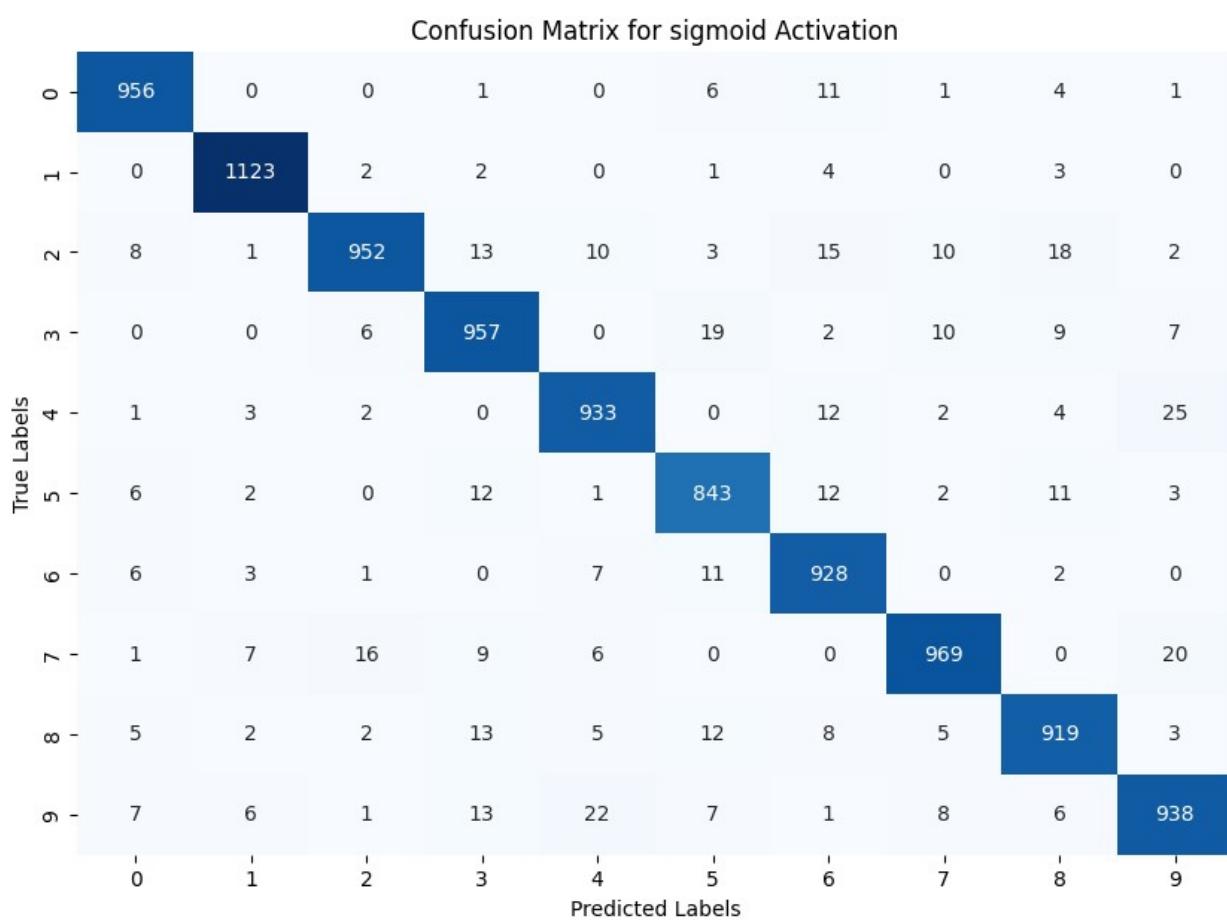
211/211 - 2s - loss: 0.7975 - accuracy: 0.8208 - val_loss: 0.5151 - val_accuracy: 0.8915 - 2s/epoch - 12ms/step

```
Epoch 3/20
211/211 - 2s - loss: 0.5058 - accuracy: 0.8751 - val_loss: 0.3731 -
val_accuracy: 0.9087 - 2s/epoch - 12ms/step
Epoch 4/20
211/211 - 2s - loss: 0.4078 - accuracy: 0.8909 - val_loss: 0.3128 -
val_accuracy: 0.9188 - 2s/epoch - 12ms/step
Epoch 5/20
211/211 - 2s - loss: 0.3570 - accuracy: 0.9018 - val_loss: 0.2755 -
val_accuracy: 0.9257 - 2s/epoch - 12ms/step
Epoch 6/20
211/211 - 3s - loss: 0.3248 - accuracy: 0.9081 - val_loss: 0.2531 -
val_accuracy: 0.9312 - 3s/epoch - 12ms/step
Epoch 7/20
211/211 - 3s - loss: 0.3019 - accuracy: 0.9144 - val_loss: 0.2313 -
val_accuracy: 0.9357 - 3s/epoch - 13ms/step
Epoch 8/20
211/211 - 3s - loss: 0.2831 - accuracy: 0.9190 - val_loss: 0.2200 -
val_accuracy: 0.9395 - 3s/epoch - 12ms/step
Epoch 9/20
211/211 - 2s - loss: 0.2673 - accuracy: 0.9235 - val_loss: 0.2166 -
val_accuracy: 0.9372 - 2s/epoch - 12ms/step
Epoch 10/20
211/211 - 2s - loss: 0.2555 - accuracy: 0.9259 - val_loss: 0.1986 -
val_accuracy: 0.9445 - 2s/epoch - 12ms/step
Epoch 11/20
211/211 - 3s - loss: 0.2430 - accuracy: 0.9302 - val_loss: 0.1871 -
val_accuracy: 0.9478 - 3s/epoch - 12ms/step
Epoch 12/20
211/211 - 3s - loss: 0.2319 - accuracy: 0.9336 - val_loss: 0.1795 -
val_accuracy: 0.9513 - 3s/epoch - 12ms/step
Epoch 13/20
211/211 - 3s - loss: 0.2228 - accuracy: 0.9359 - val_loss: 0.1767 -
val_accuracy: 0.9505 - 3s/epoch - 12ms/step
Epoch 14/20
211/211 - 3s - loss: 0.2138 - accuracy: 0.9377 - val_loss: 0.1653 -
val_accuracy: 0.9535 - 3s/epoch - 12ms/step
Epoch 15/20
211/211 - 2s - loss: 0.2060 - accuracy: 0.9401 - val_loss: 0.1589 -
val_accuracy: 0.9578 - 2s/epoch - 12ms/step
Epoch 16/20
211/211 - 2s - loss: 0.1990 - accuracy: 0.9421 - val_loss: 0.1564 -
val_accuracy: 0.9578 - 2s/epoch - 12ms/step
Epoch 17/20
211/211 - 2s - loss: 0.1918 - accuracy: 0.9444 - val_loss: 0.1531 -
val_accuracy: 0.9568 - 2s/epoch - 12ms/step
Epoch 18/20
211/211 - 2s - loss: 0.1867 - accuracy: 0.9452 - val_loss: 0.1534 -
val_accuracy: 0.9585 - 2s/epoch - 12ms/step
Epoch 19/20
```

```

211/211 - 3s - loss: 0.1794 - accuracy: 0.9478 - val_loss: 0.1427 -
val_accuracy: 0.9623 - 3s/epoch - 12ms/step
Epoch 20/20
211/211 - 3s - loss: 0.1733 - accuracy: 0.9500 - val_loss: 0.1374 -
val_accuracy: 0.9633 - 3s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 956   0   0   1   0   6  11   1   4   1]
 [ 0 1123   2   2   0   1   4   0   3   0]
 [ 8   1 952  13  10   3  15  10  18   2]
 [ 0   0   6 957   0  19   2  10   9   7]
 [ 1   3   2   0 933   0  12   2   4  25]
 [ 6   2   0  12   1 843  12   2  11   3]
 [ 6   3   1   0   7  11  928   0   2   0]
 [ 1   7  16   9   6   0   0 969   0  20]
 [ 5   2   2  13   5  12   8   5 919   3]
 [ 7   6   1 13  22   7   1   8   6 938]]
Precision: 0.9519
Recall: 0.9518

```



```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
64 batch size, 5 epochs..
Epoch 1/5
844/844 - 5s - loss: 0.8149 - accuracy: 0.7620 - val_loss: 0.3253 -
val_accuracy: 0.9053 - 5s/epoch - 6ms/step
Epoch 2/5
844/844 - 5s - loss: 0.3456 - accuracy: 0.8971 - val_loss: 0.2372 -
val_accuracy: 0.9340 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 0.2835 - accuracy: 0.9141 - val_loss: 0.1947 -
val_accuracy: 0.9457 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 0.2416 - accuracy: 0.9274 - val_loss: 0.1788 -
val_accuracy: 0.9468 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.2110 - accuracy: 0.9363 - val_loss: 0.1582 -
val_accuracy: 0.9547 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 951    0    1    2    1    9    8    1    7    0]
 [  0 1124    2    2    0    1    3    0    3    0]
 [  5    3   936    19   17    5    8   12   26    1]
 [  0    1    6   956    1   14    1   10   17    4]
 [  1    3    4    0   956    0    4    2    4    8]
 [  3    5    1   26    5   823    5    4   16    4]
 [  5    4    2    0   11   14  915    1    6    0]
 [  0   10   16    8    9    0    0   968    0   17]
 [  3    2    2   15   10   14    5    5   915    3]
 [  7    7    1   13   54    6    1   13    6  901]]
```

Precision: 0.9452
Recall: 0.9445

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	951	0	1	2	1	9	8	1	7	0	
1	0	1124	2	2	0	1	3	0	3	0	
2	5	3	936	19	17	5	8	12	26	1	
3	0	1	6	956	1	14	1	10	17	4	
4	1	3	4	0	956	0	4	2	4	8	
5	3	5	1	26	5	823	5	4	16	4	
6	5	4	2	0	11	14	915	1	6	0	
7	0	10	16	8	9	0	0	968	0	17	
8	3	2	2	15	10	14	5	5	915	3	
9	7	7	1	13	54	6	1	13	6	901	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 5s - loss: 0.7888 - accuracy: 0.7766 - val_loss: 0.3099 -
val_accuracy: 0.9152 - 5s/epoch - 6ms/step
Epoch 2/15
844/844 - 5s - loss: 0.3362 - accuracy: 0.9004 - val_loss: 0.2266 -
val_accuracy: 0.9373 - 5s/epoch - 5ms/step
Epoch 3/15
844/844 - 5s - loss: 0.2765 - accuracy: 0.9173 - val_loss: 0.2066 -
val_accuracy: 0.9385 - 5s/epoch - 5ms/step
Epoch 4/15
844/844 - 5s - loss: 0.2376 - accuracy: 0.9290 - val_loss: 0.1842 -
val_accuracy: 0.9467 - 5s/epoch - 6ms/step
Epoch 5/15
844/844 - 5s - loss: 0.2070 - accuracy: 0.9376 - val_loss: 0.1508 -
val_accuracy: 0.9585 - 5s/epoch - 6ms/step
Epoch 6/15
844/844 - 5s - loss: 0.1815 - accuracy: 0.9455 - val_loss: 0.1349 -
val_accuracy: 0.9615 - 5s/epoch - 6ms/step
Epoch 7/15
```

```
844/844 - 5s - loss: 0.1623 - accuracy: 0.9522 - val_loss: 0.1298 -  
val_accuracy: 0.9645 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.1480 - accuracy: 0.9568 - val_loss: 0.1190 -  
val_accuracy: 0.9687 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.1342 - accuracy: 0.9607 - val_loss: 0.1043 -  
val_accuracy: 0.9718 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.1230 - accuracy: 0.9642 - val_loss: 0.1080 -  
val_accuracy: 0.9703 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.1132 - accuracy: 0.9673 - val_loss: 0.0915 -  
val_accuracy: 0.9762 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.1047 - accuracy: 0.9693 - val_loss: 0.0913 -  
val_accuracy: 0.9753 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.0975 - accuracy: 0.9714 - val_loss: 0.0814 -  
val_accuracy: 0.9782 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.0904 - accuracy: 0.9740 - val_loss: 0.0804 -  
val_accuracy: 0.9797 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.0844 - accuracy: 0.9760 - val_loss: 0.0770 -  
val_accuracy: 0.9818 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 971  0  0  1  0  1  2  2  1  2]  
[  0 1126  2  1  0  1  3  1  1  0]  
[  4  4 1005  4  4  0  2  7  1  1]  
[  0  0  2 997  0  2  0  6  2  1]  
[  1  0  7  0 963  0  2  2  2  5]  
[  4  1  0 14  1 860  6  2  2  2]  
[  6  3  1  1  6  3 932  1  5  0]  
[  1  3  9  6  0  0  0 1004  0  5]  
[  5  0  3 12  5  1  2  5 937  4]  
[  5  6  2 13  13  0  0  11  3 956]]  
Precision: 0.9753  
Recall: 0.9751
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	971	0	0	1	0	1	2	2	1	2	
1	0	1126	2	1	0	1	3	1	1	0	
2	4	4	1005	4	4	0	2	7	1	1	
3	0	0	2	997	0	2	0	6	2	1	
4	1	0	7	0	963	0	2	2	2	5	
5	4	1	0	14	1	860	6	2	2	2	
6	6	3	1	1	6	3	932	1	5	0	
7	1	3	9	6	0	0	0	1004	0	5	
8	5	0	3	12	5	1	2	5	937	4	
9	5	6	2	13	13	0	0	11	3	956	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

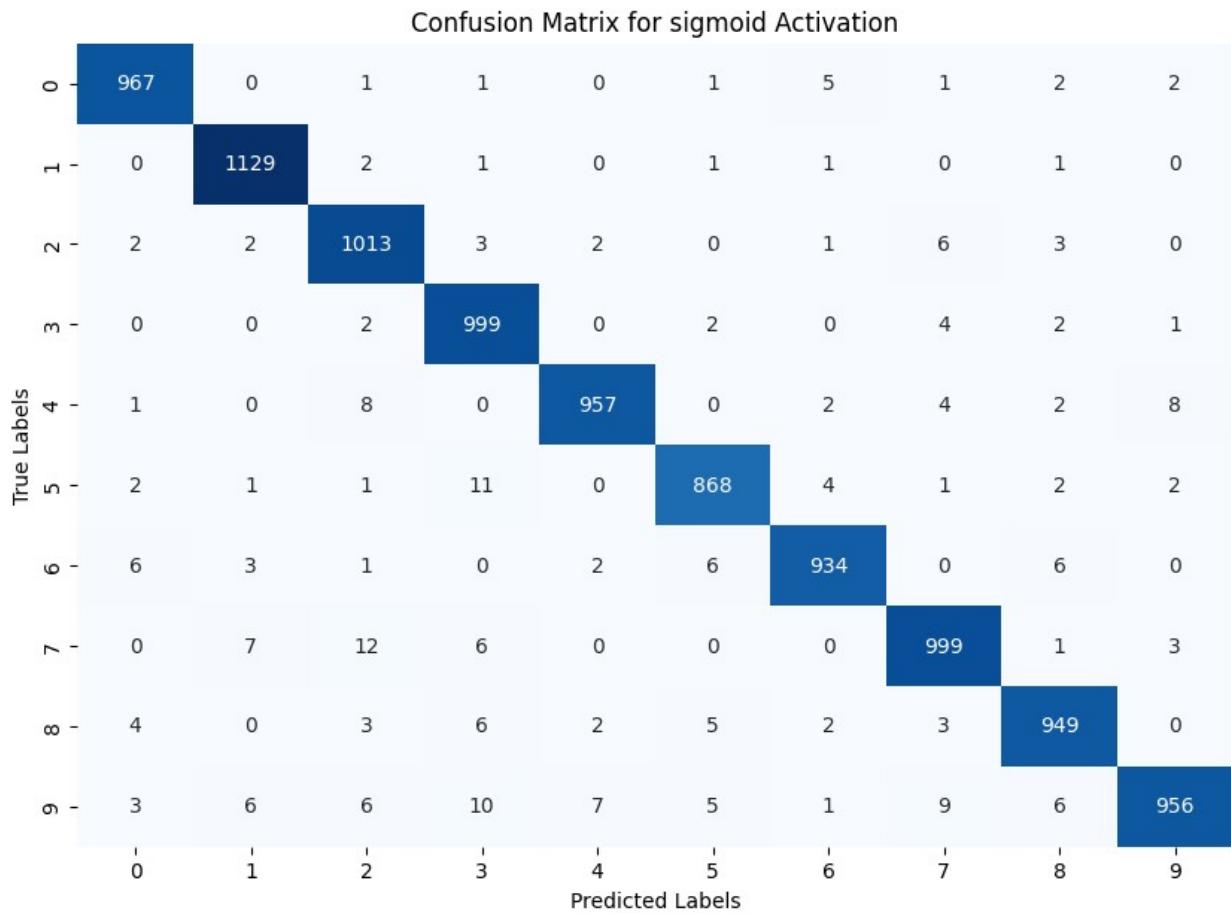
```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 5s - loss: 0.8452 - accuracy: 0.7576 - val_loss: 0.3031 -
val_accuracy: 0.9168 - 5s/epoch - 6ms/step
Epoch 2/20
844/844 - 5s - loss: 0.3429 - accuracy: 0.8966 - val_loss: 0.2510 -
val_accuracy: 0.9258 - 5s/epoch - 6ms/step
Epoch 3/20
844/844 - 5s - loss: 0.2794 - accuracy: 0.9160 - val_loss: 0.2072 -
val_accuracy: 0.9412 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: 0.2388 - accuracy: 0.9279 - val_loss: 0.1760 -
val_accuracy: 0.9495 - 5s/epoch - 5ms/step
Epoch 5/20
844/844 - 5s - loss: 0.2071 - accuracy: 0.9388 - val_loss: 0.1643 -
val_accuracy: 0.9502 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 0.1841 - accuracy: 0.9444 - val_loss: 0.1488 -
val_accuracy: 0.9575 - 5s/epoch - 6ms/step
Epoch 7/20
```

```
844/844 - 5s - loss: 0.1652 - accuracy: 0.9504 - val_loss: 0.1286 -  
val_accuracy: 0.9652 - 5s/epoch - 6ms/step  
Epoch 8/20  
844/844 - 5s - loss: 0.1497 - accuracy: 0.9562 - val_loss: 0.1218 -  
val_accuracy: 0.9647 - 5s/epoch - 6ms/step  
Epoch 9/20  
844/844 - 5s - loss: 0.1354 - accuracy: 0.9603 - val_loss: 0.1040 -  
val_accuracy: 0.9735 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: 0.1253 - accuracy: 0.9633 - val_loss: 0.1152 -  
val_accuracy: 0.9688 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: 0.1150 - accuracy: 0.9659 - val_loss: 0.0932 -  
val_accuracy: 0.9765 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 5s - loss: 0.1068 - accuracy: 0.9688 - val_loss: 0.0934 -  
val_accuracy: 0.9752 - 5s/epoch - 6ms/step  
Epoch 13/20  
844/844 - 5s - loss: 0.0996 - accuracy: 0.9711 - val_loss: 0.0859 -  
val_accuracy: 0.9778 - 5s/epoch - 6ms/step  
Epoch 14/20  
844/844 - 5s - loss: 0.0930 - accuracy: 0.9734 - val_loss: 0.0820 -  
val_accuracy: 0.9782 - 5s/epoch - 6ms/step  
Epoch 15/20  
844/844 - 5s - loss: 0.0870 - accuracy: 0.9751 - val_loss: 0.0770 -  
val_accuracy: 0.9807 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: 0.0828 - accuracy: 0.9764 - val_loss: 0.0736 -  
val_accuracy: 0.9800 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: 0.0762 - accuracy: 0.9789 - val_loss: 0.0740 -  
val_accuracy: 0.9805 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: 0.0731 - accuracy: 0.9794 - val_loss: 0.0700 -  
val_accuracy: 0.9820 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: 0.0691 - accuracy: 0.9804 - val_loss: 0.0677 -  
val_accuracy: 0.9825 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: 0.0654 - accuracy: 0.9813 - val_loss: 0.0689 -  
val_accuracy: 0.9810 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 967 0 1 1 0 1 5 1 2 2]  
[ 0 1129 2 1 0 1 1 0 1 0]  
[ 2 2 1013 3 2 0 1 6 3 0]  
[ 0 0 2 999 0 2 0 4 2 1]  
[ 1 0 8 0 957 0 2 4 2 8]]
```

```
[ 2 1 1 11 0 868 4 1 2 2]
[ 6 3 1 0 2 6 934 0 6 0]
[ 0 7 12 6 0 0 0 999 1 3]
[ 4 0 3 6 2 5 2 3 949 0]
[ 3 6 6 10 7 5 1 9 6 956]]
```

Precision: 0.9772

Recall: 0.9771



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 4s - loss: 1.2165 - accuracy: 0.6691 - val_loss: 0.4839 - val_accuracy: 0.8820 - 4s/epoch - 9ms/step

Epoch 2/5

422/422 - 3s - loss: 0.4475 - accuracy: 0.8767 - val_loss: 0.3139 - val_accuracy: 0.9148 - 3s/epoch - 8ms/step

Epoch 3/5

422/422 - 3s - loss: 0.3529 - accuracy: 0.8974 - val_loss: 0.2615 - val_accuracy: 0.9228 - 3s/epoch - 8ms/step

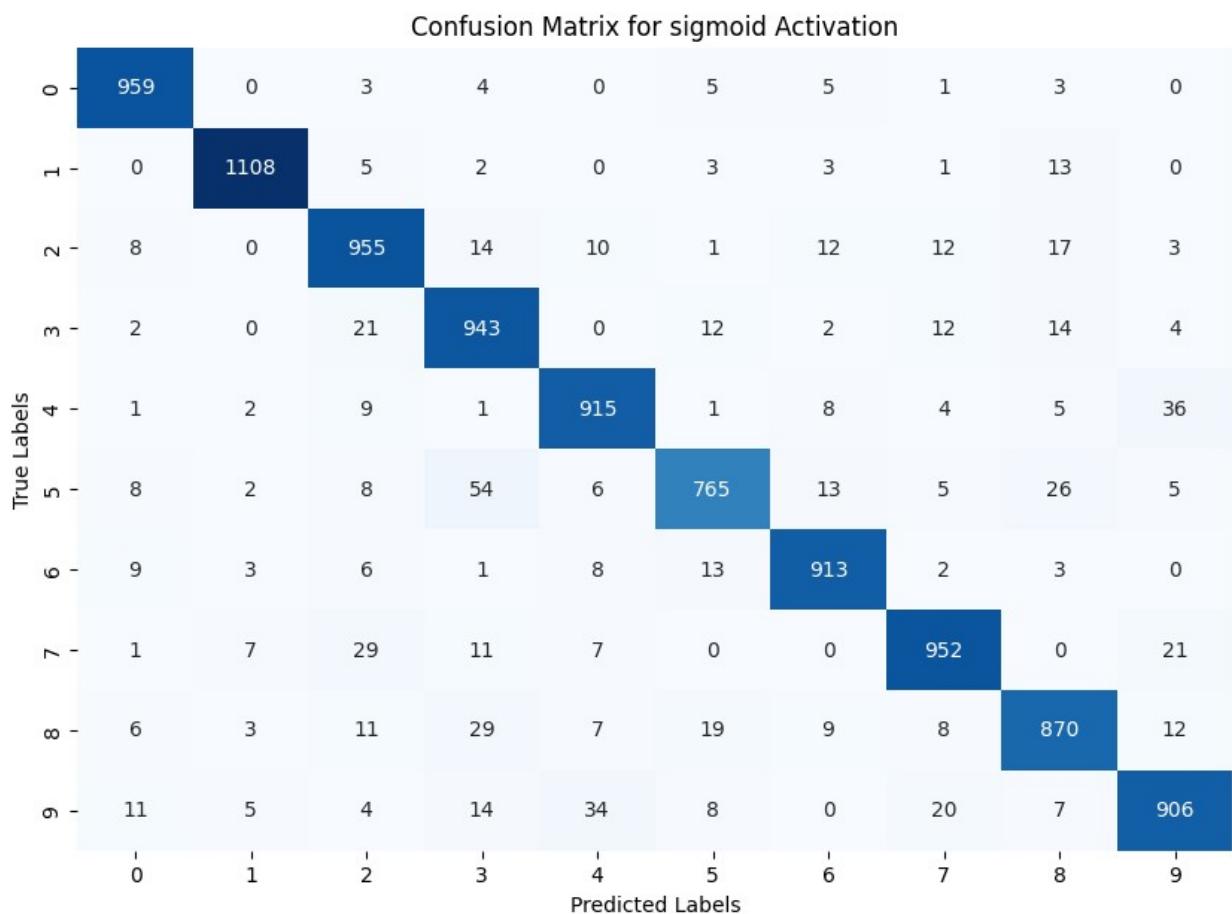
Epoch 4/5

422/422 - 3s - loss: 0.3128 - accuracy: 0.9077 - val_loss: 0.2351 -

```

val_accuracy: 0.9313 - 3s/epoch - 8ms/step
Epoch 5/5
422/422 - 3s - loss: 0.2837 - accuracy: 0.9165 - val_loss: 0.2134 -
val_accuracy: 0.9393 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 959   0   3   4   0   5   5   1   3   0]
 [ 0 1108   5   2   0   3   3   1  13   0]
 [ 8   0 955  14  10   1  12  12  17   3]
 [ 2   0  21 943   0  12   2  12  14   4]
 [ 1   2   9   1 915   1   8   4   5  36]
 [ 8   2   8   54   6 765   13   5  26   5]
 [ 9   3   6   1   8 13 913   2   3   0]
 [ 1   7  29  11   7   0   0 952   0  21]
 [ 6   3  11  29   7  19   9   8 870  12]
 [ 11  5   4  14  34   8   0  20   7 906]]
Precision: 0.9288
Recall: 0.9286

```

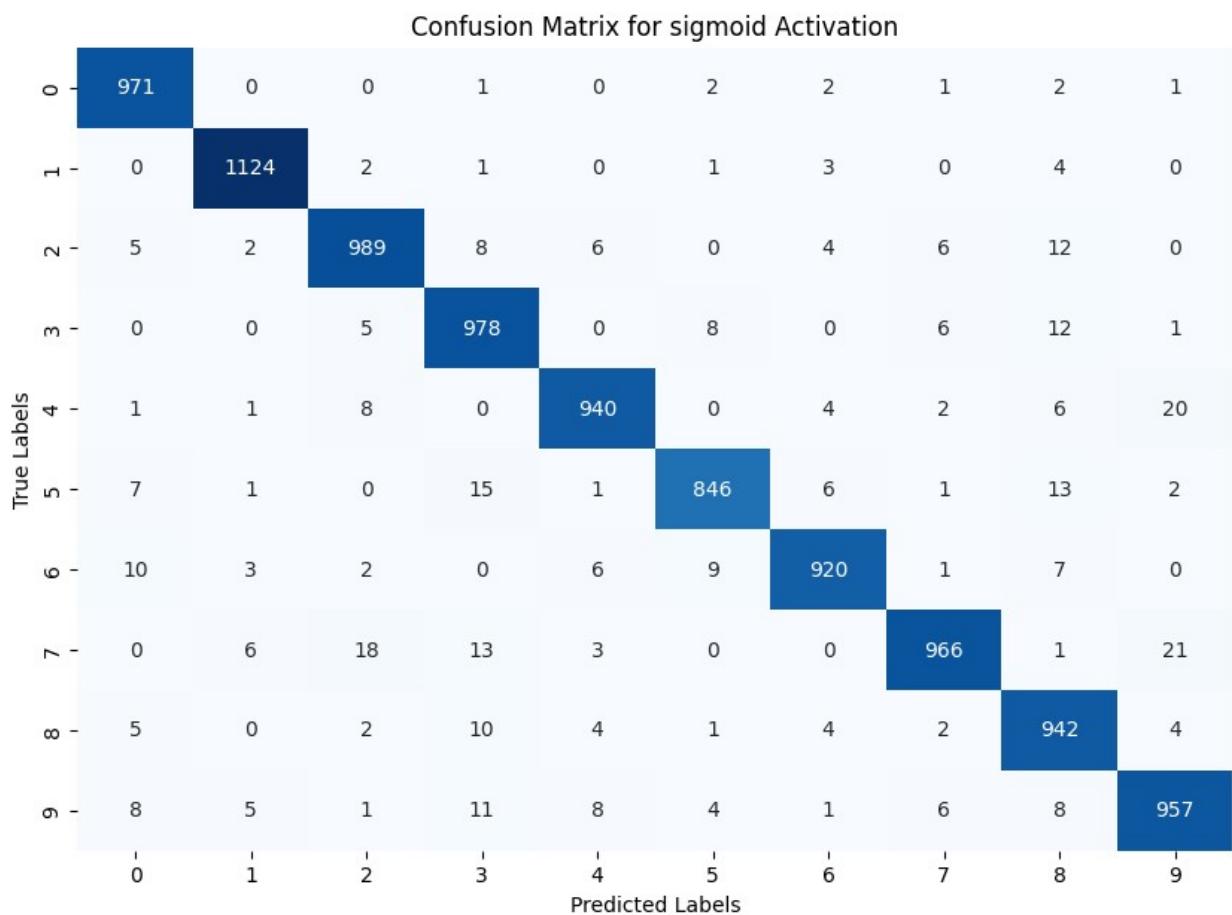


```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epoch..  
Epoch 1/15  
422/422 - 4s - loss: 1.2065 - accuracy: 0.6741 - val_loss: 0.4541 -
val_accuracy: 0.8943 - 4s/epoch - 9ms/step  
Epoch 2/15  
422/422 - 3s - loss: 0.4326 - accuracy: 0.8814 - val_loss: 0.2977 -
val_accuracy: 0.9165 - 3s/epoch - 8ms/step  
Epoch 3/15  
422/422 - 3s - loss: 0.3414 - accuracy: 0.9007 - val_loss: 0.2705 -
val_accuracy: 0.9225 - 3s/epoch - 8ms/step  
Epoch 4/15  
422/422 - 3s - loss: 0.3003 - accuracy: 0.9120 - val_loss: 0.2285 -
val_accuracy: 0.9357 - 3s/epoch - 8ms/step  
Epoch 5/15  
422/422 - 3s - loss: 0.2710 - accuracy: 0.9202 - val_loss: 0.2038 -
val_accuracy: 0.9437 - 3s/epoch - 8ms/step  
Epoch 6/15  
422/422 - 3s - loss: 0.2475 - accuracy: 0.9268 - val_loss: 0.1887 -
val_accuracy: 0.9453 - 3s/epoch - 8ms/step  
Epoch 7/15  
422/422 - 3s - loss: 0.2312 - accuracy: 0.9321 - val_loss: 0.1766 -
val_accuracy: 0.9520 - 3s/epoch - 8ms/step  
Epoch 8/15  
422/422 - 3s - loss: 0.2126 - accuracy: 0.9374 - val_loss: 0.1674 -
val_accuracy: 0.9523 - 3s/epoch - 8ms/step  
Epoch 9/15  
422/422 - 3s - loss: 0.1980 - accuracy: 0.9411 - val_loss: 0.1575 -
val_accuracy: 0.9560 - 3s/epoch - 8ms/step  
Epoch 10/15  
422/422 - 3s - loss: 0.1863 - accuracy: 0.9452 - val_loss: 0.1408 -
val_accuracy: 0.9608 - 3s/epoch - 8ms/step  
Epoch 11/15  
422/422 - 3s - loss: 0.1725 - accuracy: 0.9494 - val_loss: 0.1320 -
val_accuracy: 0.9660 - 3s/epoch - 8ms/step  
Epoch 12/15  
422/422 - 3s - loss: 0.1622 - accuracy: 0.9525 - val_loss: 0.1354 -
val_accuracy: 0.9623 - 3s/epoch - 8ms/step  
Epoch 13/15  
422/422 - 3s - loss: 0.1524 - accuracy: 0.9555 - val_loss: 0.1179 -
val_accuracy: 0.9682 - 3s/epoch - 8ms/step  
Epoch 14/15  
422/422 - 3s - loss: 0.1442 - accuracy: 0.9583 - val_loss: 0.1131 -
val_accuracy: 0.9688 - 3s/epoch - 8ms/step  
Epoch 15/15  
422/422 - 3s - loss: 0.1350 - accuracy: 0.9604 - val_loss: 0.1069 -
val_accuracy: 0.9733 - 3s/epoch - 8ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:
```

```
[[ 971  0  0  1  0  2  2  1  2  1]
 [ 0 1124  2  1  0  1  3  0  4  0]
 [ 5  2 989  8  6  0  4  6 12  0]
 [ 0  0  5 978  0  8  0  6 12  1]
 [ 1  1  8  0 940  0  4  2  6 20]
 [ 7  1  0 15  1 846  6  1 13  2]
 [10  3  2  0  6  9 920  1  7  0]
 [ 0  6 18 13  3  0  0 966  1 21]
 [ 5  0  2 10  4  1  4  2 942  4]
 [ 8  5  1 11  8  4  1  6  8 957]]
```

Precision: 0.9635

Recall: 0.9633



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 128 batch size, 20 epochs..

Epoch 1/20

422/422 - 4s - loss: 1.1732 - accuracy: 0.6809 - val_loss: 0.4390 - val_accuracy: 0.8967 - 4s/epoch - 9ms/step

Epoch 2/20

422/422 - 3s - loss: 0.4303 - accuracy: 0.8798 - val_loss: 0.3032 - val_accuracy: 0.9147 - 3s/epoch - 8ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.3473 - accuracy: 0.8985 - val_loss: 0.2600 -
val_accuracy: 0.9237 - 3s/epoch - 7ms/step
Epoch 4/20
422/422 - 3s - loss: 0.3063 - accuracy: 0.9097 - val_loss: 0.2373 -
val_accuracy: 0.9320 - 3s/epoch - 7ms/step
Epoch 5/20
422/422 - 3s - loss: 0.2813 - accuracy: 0.9171 - val_loss: 0.2142 -
val_accuracy: 0.9383 - 3s/epoch - 7ms/step
Epoch 6/20
422/422 - 3s - loss: 0.2594 - accuracy: 0.9228 - val_loss: 0.1992 -
val_accuracy: 0.9435 - 3s/epoch - 8ms/step
Epoch 7/20
422/422 - 3s - loss: 0.2399 - accuracy: 0.9291 - val_loss: 0.1839 -
val_accuracy: 0.9475 - 3s/epoch - 8ms/step
Epoch 8/20
422/422 - 3s - loss: 0.2236 - accuracy: 0.9336 - val_loss: 0.1744 -
val_accuracy: 0.9482 - 3s/epoch - 8ms/step
Epoch 9/20
422/422 - 3s - loss: 0.2098 - accuracy: 0.9379 - val_loss: 0.1691 -
val_accuracy: 0.9500 - 3s/epoch - 8ms/step
Epoch 10/20
422/422 - 3s - loss: 0.1969 - accuracy: 0.9415 - val_loss: 0.1506 -
val_accuracy: 0.9577 - 3s/epoch - 8ms/step
Epoch 11/20
422/422 - 3s - loss: 0.1845 - accuracy: 0.9449 - val_loss: 0.1581 -
val_accuracy: 0.9542 - 3s/epoch - 8ms/step
Epoch 12/20
422/422 - 3s - loss: 0.1738 - accuracy: 0.9476 - val_loss: 0.1391 -
val_accuracy: 0.9598 - 3s/epoch - 8ms/step
Epoch 13/20
422/422 - 3s - loss: 0.1658 - accuracy: 0.9509 - val_loss: 0.1261 -
val_accuracy: 0.9655 - 3s/epoch - 8ms/step
Epoch 14/20
422/422 - 3s - loss: 0.1558 - accuracy: 0.9537 - val_loss: 0.1194 -
val_accuracy: 0.9675 - 3s/epoch - 8ms/step
Epoch 15/20
422/422 - 3s - loss: 0.1477 - accuracy: 0.9569 - val_loss: 0.1165 -
val_accuracy: 0.9705 - 3s/epoch - 8ms/step
Epoch 16/20
422/422 - 3s - loss: 0.1398 - accuracy: 0.9594 - val_loss: 0.1165 -
val_accuracy: 0.9700 - 3s/epoch - 8ms/step
Epoch 17/20
422/422 - 3s - loss: 0.1348 - accuracy: 0.9609 - val_loss: 0.1092 -
val_accuracy: 0.9730 - 3s/epoch - 8ms/step
Epoch 18/20
422/422 - 3s - loss: 0.1279 - accuracy: 0.9628 - val_loss: 0.1062 -
val_accuracy: 0.9710 - 3s/epoch - 8ms/step
Epoch 19/20
```

```
422/422 - 3s - loss: 0.1230 - accuracy: 0.9640 - val_loss: 0.1030 -  
val_accuracy: 0.9720 - 3s/epoch - 8ms/step
```

```
Epoch 20/20
```

```
422/422 - 3s - loss: 0.1169 - accuracy: 0.9657 - val_loss: 0.0962 -  
val_accuracy: 0.9732 - 3s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: sigmoid
```

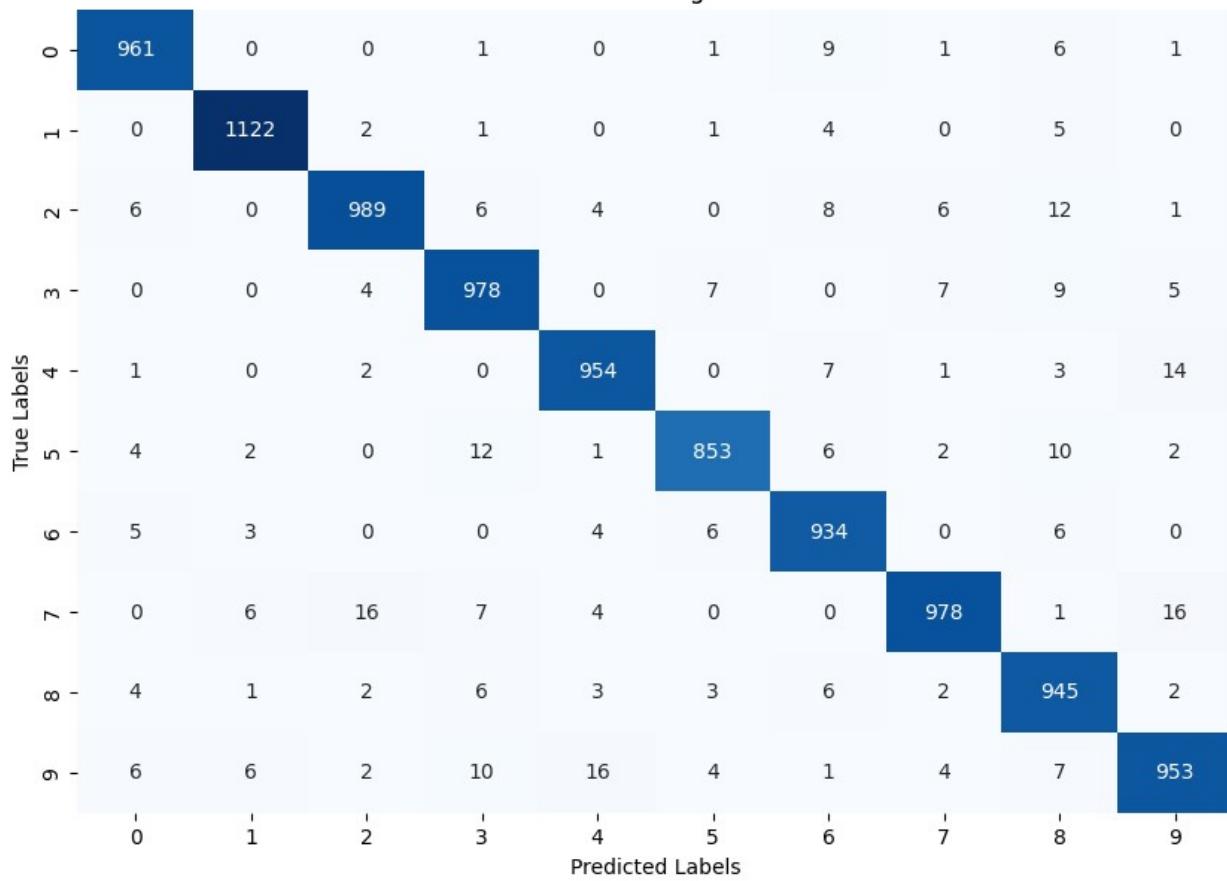
```
Confusion Matrix:
```

```
[[ 961   0   0   1   0   1   9   1   6   1]
 [ 0 1122   2   1   0   1   4   0   5   0]
 [ 6   0 989   6   4   0   8   6 12   1]
 [ 0   0   4 978   0   7   0   7   9   5]
 [ 1   0   2   0 954   0   7   1   3 14]
 [ 4   2   0 12   1 853   6   2 10   2]
 [ 5   3   0   0   4   6 934   0   6   0]
 [ 0   6 16   7   4   0   0 978   1 16]
 [ 4   1   2   6   3   3   6   2 945   2]
 [ 6   6   2 10 16   4   1   4   7 953]]
```

```
Precision: 0.9668
```

```
Recall: 0.9667
```

Confusion Matrix for sigmoid Activation



```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 3s - loss: 1.6881 - accuracy: 0.5639 - val_loss: 0.8990 -
val_accuracy: 0.8405 - 3s/epoch - 16ms/step
Epoch 2/5
211/211 - 3s - loss: 0.6829 - accuracy: 0.8397 - val_loss: 0.4605 -
val_accuracy: 0.8948 - 3s/epoch - 13ms/step
Epoch 3/5
211/211 - 3s - loss: 0.4634 - accuracy: 0.8763 - val_loss: 0.3432 -
val_accuracy: 0.9115 - 3s/epoch - 13ms/step
Epoch 4/5
211/211 - 3s - loss: 0.3879 - accuracy: 0.8920 - val_loss: 0.3034 -
val_accuracy: 0.9158 - 3s/epoch - 13ms/step
Epoch 5/5
211/211 - 3s - loss: 0.3498 - accuracy: 0.9004 - val_loss: 0.2696 -
val_accuracy: 0.9247 - 3s/epoch - 13ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 959     0     5     3     0     4     4     1     4     0]
 [  0 1108     5     2     0     3     3     0    14     0]
 [ 11     4  928    17     9     1    18    13    26     5]
 [  3     2    25  925     0    18     3    14    17     3]
 [  2     7     8     0   888     4    18     1     9    45]
 [ 12     3     9    59     7   749    20     7    21     5]
 [ 12     4    11     2    10    19   898     0     2     0]
 [  5    17    37     5     9     0     0   919     3    33]
 [  9     5    14    31     7    27    11     8   850    12]
 [ 14     6     9    14    32     9     0    20     8  897]]
```

Precision: 0.9121
Recall: 0.9121

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	959	0	5	3	0	4	4	1	4	0	
0 -	959	0	5	3	0	4	4	1	4	0	
1 -	0	1108	5	2	0	3	3	0	14	0	
2 -	11	4	928	17	9	1	18	13	26	5	
3 -	3	2	25	925	0	18	3	14	17	3	
4 -	2	7	8	0	888	4	18	1	9	45	
5 -	12	3	9	59	7	749	20	7	21	5	
6 -	12	4	11	2	10	19	898	0	2	0	
7 -	5	17	37	5	9	0	0	919	3	33	
8 -	9	5	14	31	7	27	11	8	850	12	
9 -	14	6	9	14	32	9	0	20	8	897	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels

```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 3s - loss: 1.4732 - accuracy: 0.6227 - val_loss: 0.7060 -
val_accuracy: 0.8630 - 3s/epoch - 15ms/step
Epoch 2/15
211/211 - 3s - loss: 0.5909 - accuracy: 0.8539 - val_loss: 0.4060 -
val_accuracy: 0.9000 - 3s/epoch - 12ms/step
Epoch 3/15
211/211 - 2s - loss: 0.4333 - accuracy: 0.8828 - val_loss: 0.3338 -
val_accuracy: 0.9133 - 2s/epoch - 12ms/step
Epoch 4/15
211/211 - 2s - loss: 0.3776 - accuracy: 0.8934 - val_loss: 0.2882 -
val_accuracy: 0.9203 - 2s/epoch - 12ms/step
Epoch 5/15
211/211 - 2s - loss: 0.3440 - accuracy: 0.9009 - val_loss: 0.2695 -
val_accuracy: 0.9242 - 2s/epoch - 12ms/step
Epoch 6/15
211/211 - 2s - loss: 0.3202 - accuracy: 0.9066 - val_loss: 0.2534 -
val_accuracy: 0.9292 - 2s/epoch - 12ms/step
Epoch 7/15
```

```
211/211 - 3s - loss: 0.3033 - accuracy: 0.9106 - val_loss: 0.2376 -  
val_accuracy: 0.9308 - 3s/epoch - 12ms/step  
Epoch 8/15  
211/211 - 3s - loss: 0.2893 - accuracy: 0.9141 - val_loss: 0.2258 -  
val_accuracy: 0.9350 - 3s/epoch - 12ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.2778 - accuracy: 0.9179 - val_loss: 0.2250 -  
val_accuracy: 0.9352 - 3s/epoch - 12ms/step  
Epoch 10/15  
211/211 - 3s - loss: 0.2652 - accuracy: 0.9224 - val_loss: 0.2125 -  
val_accuracy: 0.9387 - 3s/epoch - 12ms/step  
Epoch 11/15  
211/211 - 3s - loss: 0.2550 - accuracy: 0.9245 - val_loss: 0.2006 -  
val_accuracy: 0.9427 - 3s/epoch - 12ms/step  
Epoch 12/15  
211/211 - 3s - loss: 0.2473 - accuracy: 0.9263 - val_loss: 0.1945 -  
val_accuracy: 0.9448 - 3s/epoch - 12ms/step  
Epoch 13/15  
211/211 - 3s - loss: 0.2372 - accuracy: 0.9299 - val_loss: 0.1860 -  
val_accuracy: 0.9450 - 3s/epoch - 12ms/step  
Epoch 14/15  
211/211 - 3s - loss: 0.2304 - accuracy: 0.9316 - val_loss: 0.1917 -  
val_accuracy: 0.9455 - 3s/epoch - 12ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.2223 - accuracy: 0.9346 - val_loss: 0.1783 -  
val_accuracy: 0.9460 - 3s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 968 0 1 1 0 2 2 2 4 0]  
[ 0 1111 2 2 0 1 4 1 14 0]  
[ 9 0 948 6 9 1 14 14 28 3]  
[ 5 2 20 891 0 25 2 19 39 7]  
[ 1 3 5 0 927 1 8 2 10 25]  
[ 8 3 3 23 6 792 14 5 34 4]  
[ 10 3 4 0 6 8 920 3 4 0]  
[ 2 7 22 3 7 0 0 966 2 19]  
[ 6 2 5 7 6 6 8 7 922 5]  
[ 10 6 1 9 30 5 1 19 10 918]]  
Precision: 0.9370  
Recall: 0.9363
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	968	0	1	1	0	2	2	2	4	0	
1	0	1111	2	2	0	1	4	1	14	0	
2	9	0	948	6	9	1	14	14	28	3	
3	5	2	20	891	0	25	2	19	39	7	
4	1	3	5	0	927	1	8	2	10	25	
5	8	3	3	23	6	792	14	5	34	4	
6	10	3	4	0	6	8	920	3	4	0	
7	2	7	22	3	7	0	0	966	2	19	
8	6	2	5	7	6	6	8	7	922	5	
9	10	6	1	9	30	5	1	19	10	918	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels											

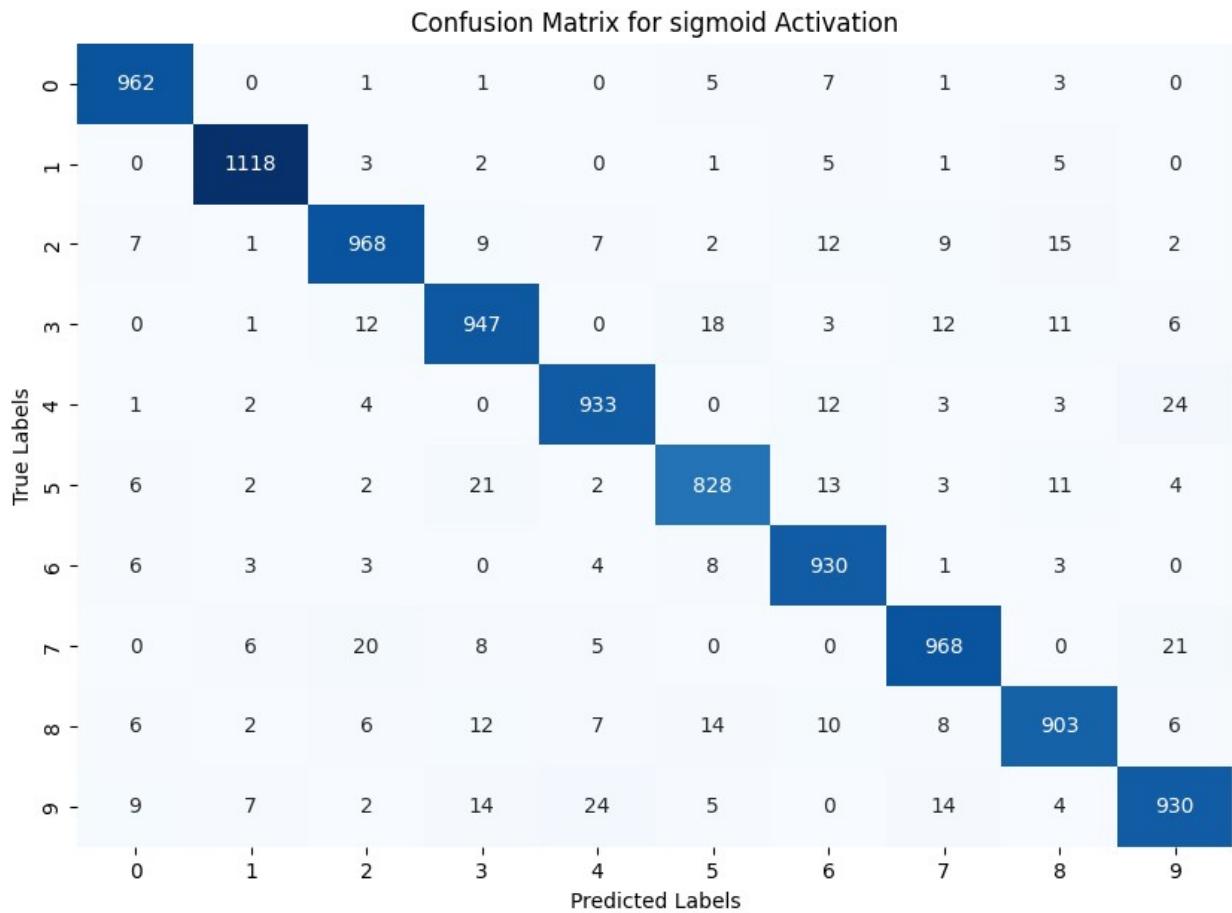
```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 3s - loss: 1.4686 - accuracy: 0.6222 - val_loss: 0.7085 -
val_accuracy: 0.8595 - 3s/epoch - 15ms/step
Epoch 2/20
211/211 - 3s - loss: 0.5967 - accuracy: 0.8546 - val_loss: 0.4079 -
val_accuracy: 0.9048 - 3s/epoch - 12ms/step
Epoch 3/20
211/211 - 3s - loss: 0.4370 - accuracy: 0.8820 - val_loss: 0.3345 -
val_accuracy: 0.9112 - 3s/epoch - 12ms/step
Epoch 4/20
211/211 - 3s - loss: 0.3789 - accuracy: 0.8925 - val_loss: 0.2971 -
val_accuracy: 0.9175 - 3s/epoch - 13ms/step
Epoch 5/20
211/211 - 3s - loss: 0.3459 - accuracy: 0.8990 - val_loss: 0.2671 -
val_accuracy: 0.9262 - 3s/epoch - 12ms/step
Epoch 6/20
211/211 - 3s - loss: 0.3238 - accuracy: 0.9055 - val_loss: 0.2514 -
val_accuracy: 0.9273 - 3s/epoch - 12ms/step
Epoch 7/20
```

```
211/211 - 3s - loss: 0.3082 - accuracy: 0.9094 - val_loss: 0.2367 -  
val_accuracy: 0.9318 - 3s/epoch - 12ms/step  
Epoch 8/20  
211/211 - 3s - loss: 0.2911 - accuracy: 0.9147 - val_loss: 0.2310 -  
val_accuracy: 0.9325 - 3s/epoch - 12ms/step  
Epoch 9/20  
211/211 - 3s - loss: 0.2790 - accuracy: 0.9171 - val_loss: 0.2161 -  
val_accuracy: 0.9380 - 3s/epoch - 12ms/step  
Epoch 10/20  
211/211 - 3s - loss: 0.2665 - accuracy: 0.9207 - val_loss: 0.2121 -  
val_accuracy: 0.9397 - 3s/epoch - 12ms/step  
Epoch 11/20  
211/211 - 3s - loss: 0.2563 - accuracy: 0.9243 - val_loss: 0.2033 -  
val_accuracy: 0.9427 - 3s/epoch - 12ms/step  
Epoch 12/20  
211/211 - 3s - loss: 0.2462 - accuracy: 0.9271 - val_loss: 0.1956 -  
val_accuracy: 0.9445 - 3s/epoch - 12ms/step  
Epoch 13/20  
211/211 - 3s - loss: 0.2380 - accuracy: 0.9290 - val_loss: 0.1981 -  
val_accuracy: 0.9392 - 3s/epoch - 12ms/step  
Epoch 14/20  
211/211 - 3s - loss: 0.2307 - accuracy: 0.9314 - val_loss: 0.1838 -  
val_accuracy: 0.9475 - 3s/epoch - 12ms/step  
Epoch 15/20  
211/211 - 3s - loss: 0.2228 - accuracy: 0.9340 - val_loss: 0.1794 -  
val_accuracy: 0.9495 - 3s/epoch - 12ms/step  
Epoch 16/20  
211/211 - 3s - loss: 0.2135 - accuracy: 0.9365 - val_loss: 0.1750 -  
val_accuracy: 0.9490 - 3s/epoch - 12ms/step  
Epoch 17/20  
211/211 - 3s - loss: 0.2073 - accuracy: 0.9382 - val_loss: 0.1649 -  
val_accuracy: 0.9535 - 3s/epoch - 12ms/step  
Epoch 18/20  
211/211 - 3s - loss: 0.1995 - accuracy: 0.9409 - val_loss: 0.1581 -  
val_accuracy: 0.9568 - 3s/epoch - 12ms/step  
Epoch 19/20  
211/211 - 3s - loss: 0.1948 - accuracy: 0.9428 - val_loss: 0.1622 -  
val_accuracy: 0.9522 - 3s/epoch - 12ms/step  
Epoch 20/20  
211/211 - 3s - loss: 0.1880 - accuracy: 0.9441 - val_loss: 0.1495 -  
val_accuracy: 0.9597 - 3s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 962 0 1 1 0 5 7 1 3 0]  
 [ 0 1118 3 2 0 1 5 1 5 0]  
 [ 7 1 968 9 7 2 12 9 15 2]  
 [ 0 1 12 947 0 18 3 12 11 6]  
 [ 1 2 4 0 933 0 12 3 3 24]]
```

```
[ 6  2  2 21  2 828 13  3 11  4]
[ 6  3  3  0  4  8 930  1  3  0]
[ 0  6 20  8  5  0  0 968  0 21]
[ 6  2  6 12  7 14 10  8 903  6]
[ 9  7  2 14 24  5  0 14  4 930]]
```

Precision: 0.9487

Recall: 0.9487



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 6s - loss: 0.8062 - accuracy: 0.7665 - val_loss: 0.3051 - val_accuracy: 0.9133 - 6s/epoch - 7ms/step

Epoch 2/5

844/844 - 5s - loss: 0.3438 - accuracy: 0.8969 - val_loss: 0.2462 - val_accuracy: 0.9252 - 5s/epoch - 6ms/step

Epoch 3/5

844/844 - 5s - loss: 0.2842 - accuracy: 0.9140 - val_loss: 0.2005 - val_accuracy: 0.9412 - 5s/epoch - 6ms/step

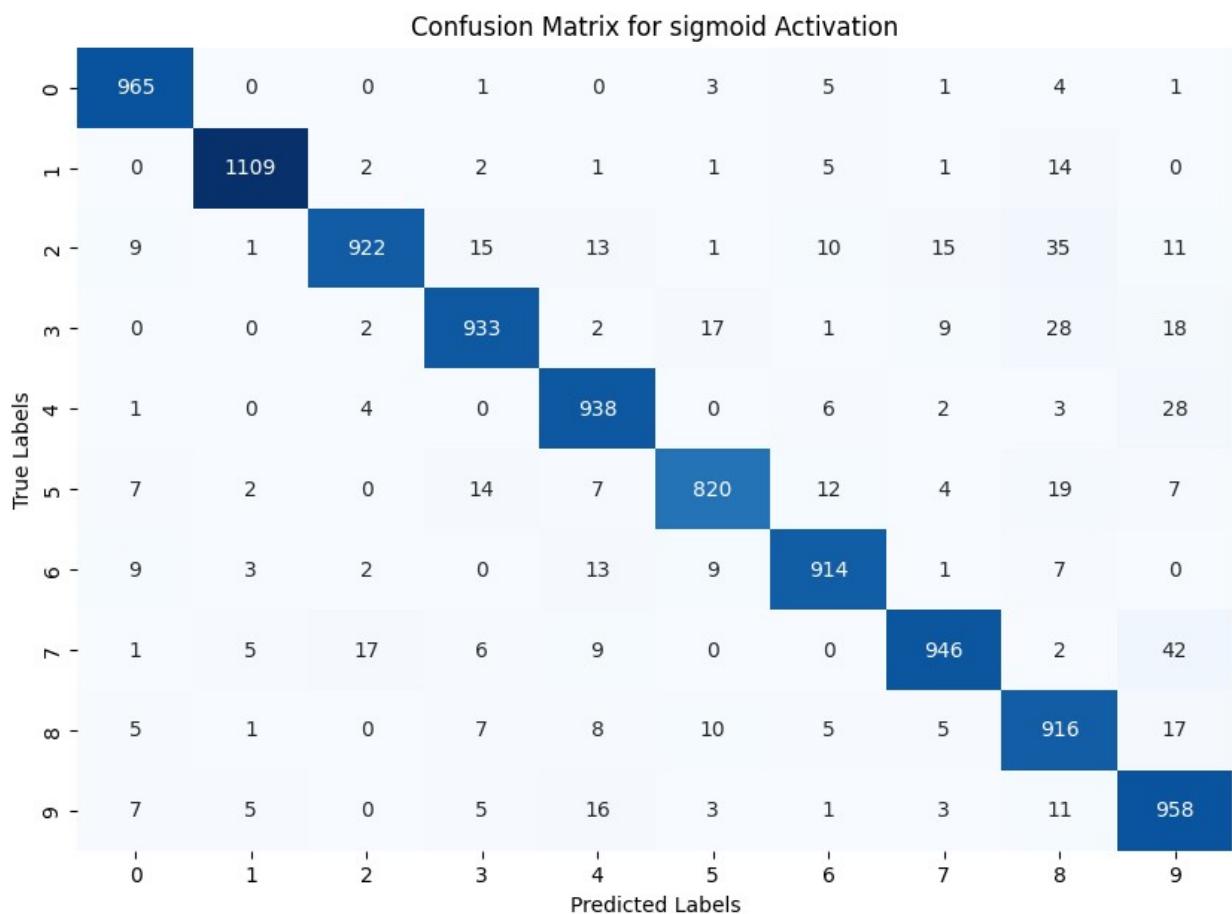
Epoch 4/5

844/844 - 5s - loss: 0.2443 - accuracy: 0.9270 - val_loss: 0.1872 -

```

val_accuracy: 0.9438 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.2141 - accuracy: 0.9358 - val_loss: 0.1591 -
val_accuracy: 0.9542 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 965 0 0 1 0 3 5 1 4 1]
 [ 0 1109 2 2 1 1 5 1 14 0]
 [ 9 1 922 15 13 1 10 15 35 11]
 [ 0 0 2 933 2 17 1 9 28 18]
 [ 1 0 4 0 938 0 6 2 3 28]
 [ 7 2 0 14 7 820 12 4 19 7]
 [ 9 3 2 0 13 9 914 1 7 0]
 [ 1 5 17 6 9 0 0 946 2 42]
 [ 5 1 0 7 8 10 5 5 916 17]
 [ 7 5 0 5 16 3 1 3 11 958]]
Precision: 0.9433
Recall: 0.9421

```



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
64 batch size, 15 epochs..

Epoch 1/15
844/844 - 6s - loss: 0.8090 - accuracy: 0.7660 - val_loss: 0.3164 -
val_accuracy: 0.9072 - 6s/epoch - 7ms/step

Epoch 2/15
844/844 - 5s - loss: 0.3450 - accuracy: 0.8964 - val_loss: 0.2411 -
val_accuracy: 0.9282 - 5s/epoch - 6ms/step

Epoch 3/15
844/844 - 5s - loss: 0.2815 - accuracy: 0.9152 - val_loss: 0.2059 -
val_accuracy: 0.9398 - 5s/epoch - 6ms/step

Epoch 4/15
844/844 - 5s - loss: 0.2424 - accuracy: 0.9272 - val_loss: 0.1730 -
val_accuracy: 0.9517 - 5s/epoch - 6ms/step

Epoch 5/15
844/844 - 5s - loss: 0.2103 - accuracy: 0.9377 - val_loss: 0.1516 -
val_accuracy: 0.9572 - 5s/epoch - 6ms/step

Epoch 6/15
844/844 - 5s - loss: 0.1844 - accuracy: 0.9446 - val_loss: 0.1320 -
val_accuracy: 0.9638 - 5s/epoch - 6ms/step

Epoch 7/15
844/844 - 5s - loss: 0.1639 - accuracy: 0.9510 - val_loss: 0.1273 -
val_accuracy: 0.9660 - 5s/epoch - 6ms/step

Epoch 8/15
844/844 - 5s - loss: 0.1452 - accuracy: 0.9569 - val_loss: 0.1216 -
val_accuracy: 0.9675 - 5s/epoch - 6ms/step

Epoch 9/15
844/844 - 5s - loss: 0.1326 - accuracy: 0.9609 - val_loss: 0.1026 -
val_accuracy: 0.9730 - 5s/epoch - 6ms/step

Epoch 10/15
844/844 - 5s - loss: 0.1206 - accuracy: 0.9639 - val_loss: 0.0933 -
val_accuracy: 0.9760 - 5s/epoch - 6ms/step

Epoch 11/15
844/844 - 5s - loss: 0.1090 - accuracy: 0.9681 - val_loss: 0.0864 -
val_accuracy: 0.9767 - 5s/epoch - 6ms/step

Epoch 12/15
844/844 - 5s - loss: 0.1016 - accuracy: 0.9699 - val_loss: 0.0933 -
val_accuracy: 0.9723 - 5s/epoch - 6ms/step

Epoch 13/15
844/844 - 6s - loss: 0.0940 - accuracy: 0.9723 - val_loss: 0.0764 -
val_accuracy: 0.9805 - 6s/epoch - 7ms/step

Epoch 14/15
844/844 - 5s - loss: 0.0877 - accuracy: 0.9738 - val_loss: 0.0790 -
val_accuracy: 0.9795 - 5s/epoch - 6ms/step

Epoch 15/15
844/844 - 5s - loss: 0.0816 - accuracy: 0.9762 - val_loss: 0.0708 -
val_accuracy: 0.9813 - 5s/epoch - 6ms/step

313/313 [=====] - 1s 2ms/step

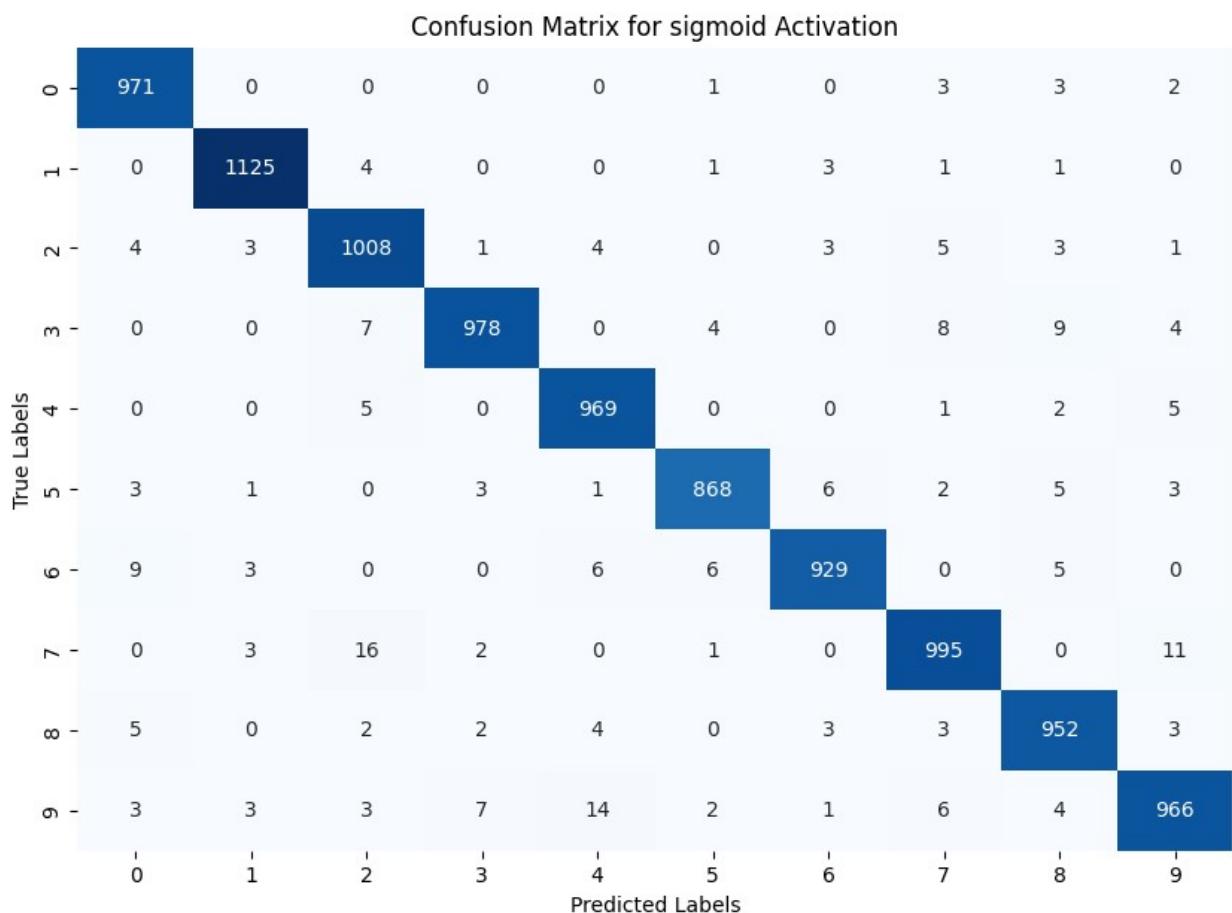
Results for activation function: sigmoid

Confusion Matrix:

```
[[ 971  0  0  0  0  1  0  3  3  3  2]
 [ 0 1125  4  0  0  1  3  1  1  0]
 [ 4  3 1008  1  4  0  3  5  3  3  1]
 [ 0  0  7  978  0  4  0  8  9  4]
 [ 0  0  5  0  969  0  0  1  2  5]
 [ 3  1  0  3  1  868  6  2  5  3]
 [ 9  3  0  0  6  6  929  0  5  0]
 [ 0  3  16  2  0  1  0  995  0  11]
 [ 5  0  2  2  4  0  3  3  952  3]
 [ 3  3  3  7  14  2  1  6  4  966]]
```

Precision: 0.9761

Recall: 0.9761



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 64 batch size, 20 epochs..

Epoch 1/20

844/844 - 6s - loss: 0.7955 - accuracy: 0.7606 - val_loss: 0.3096 - val_accuracy: 0.9088 - 6s/epoch - 7ms/step

Epoch 2/20

844/844 - 5s - loss: 0.3441 - accuracy: 0.8969 - val_loss: 0.2598 - val_accuracy: 0.9242 - 5s/epoch - 6ms/step

```
Epoch 3/20
844/844 - 5s - loss: 0.2851 - accuracy: 0.9141 - val_loss: 0.2021 -
val_accuracy: 0.9390 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: 0.2478 - accuracy: 0.9254 - val_loss: 0.1908 -
val_accuracy: 0.9433 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: 0.2183 - accuracy: 0.9336 - val_loss: 0.1667 -
val_accuracy: 0.9527 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 0.1909 - accuracy: 0.9429 - val_loss: 0.1468 -
val_accuracy: 0.9590 - 5s/epoch - 6ms/step
Epoch 7/20
844/844 - 5s - loss: 0.1697 - accuracy: 0.9491 - val_loss: 0.1322 -
val_accuracy: 0.9642 - 5s/epoch - 6ms/step
Epoch 8/20
844/844 - 5s - loss: 0.1526 - accuracy: 0.9546 - val_loss: 0.1144 -
val_accuracy: 0.9700 - 5s/epoch - 6ms/step
Epoch 9/20
844/844 - 5s - loss: 0.1372 - accuracy: 0.9590 - val_loss: 0.1055 -
val_accuracy: 0.9718 - 5s/epoch - 6ms/step
Epoch 10/20
844/844 - 5s - loss: 0.1260 - accuracy: 0.9629 - val_loss: 0.0966 -
val_accuracy: 0.9747 - 5s/epoch - 6ms/step
Epoch 11/20
844/844 - 5s - loss: 0.1144 - accuracy: 0.9665 - val_loss: 0.0910 -
val_accuracy: 0.9763 - 5s/epoch - 6ms/step
Epoch 12/20
844/844 - 5s - loss: 0.1063 - accuracy: 0.9691 - val_loss: 0.0908 -
val_accuracy: 0.9775 - 5s/epoch - 6ms/step
Epoch 13/20
844/844 - 6s - loss: 0.0978 - accuracy: 0.9709 - val_loss: 0.0832 -
val_accuracy: 0.9770 - 6s/epoch - 7ms/step
Epoch 14/20
844/844 - 6s - loss: 0.0905 - accuracy: 0.9734 - val_loss: 0.0827 -
val_accuracy: 0.9785 - 6s/epoch - 7ms/step
Epoch 15/20
844/844 - 5s - loss: 0.0854 - accuracy: 0.9749 - val_loss: 0.0775 -
val_accuracy: 0.9807 - 5s/epoch - 6ms/step
Epoch 16/20
844/844 - 5s - loss: 0.0791 - accuracy: 0.9774 - val_loss: 0.0743 -
val_accuracy: 0.9815 - 5s/epoch - 6ms/step
Epoch 17/20
844/844 - 5s - loss: 0.0745 - accuracy: 0.9785 - val_loss: 0.0713 -
val_accuracy: 0.9802 - 5s/epoch - 6ms/step
Epoch 18/20
844/844 - 5s - loss: 0.0701 - accuracy: 0.9795 - val_loss: 0.0732 -
val_accuracy: 0.9807 - 5s/epoch - 6ms/step
Epoch 19/20
```

```
844/844 - 5s - loss: 0.0672 - accuracy: 0.9805 - val_loss: 0.0682 -  
val_accuracy: 0.9823 - 5s/epoch - 6ms/step
```

```
Epoch 20/20
```

```
844/844 - 5s - loss: 0.0630 - accuracy: 0.9819 - val_loss: 0.0705 -  
val_accuracy: 0.9803 - 5s/epoch - 6ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: sigmoid
```

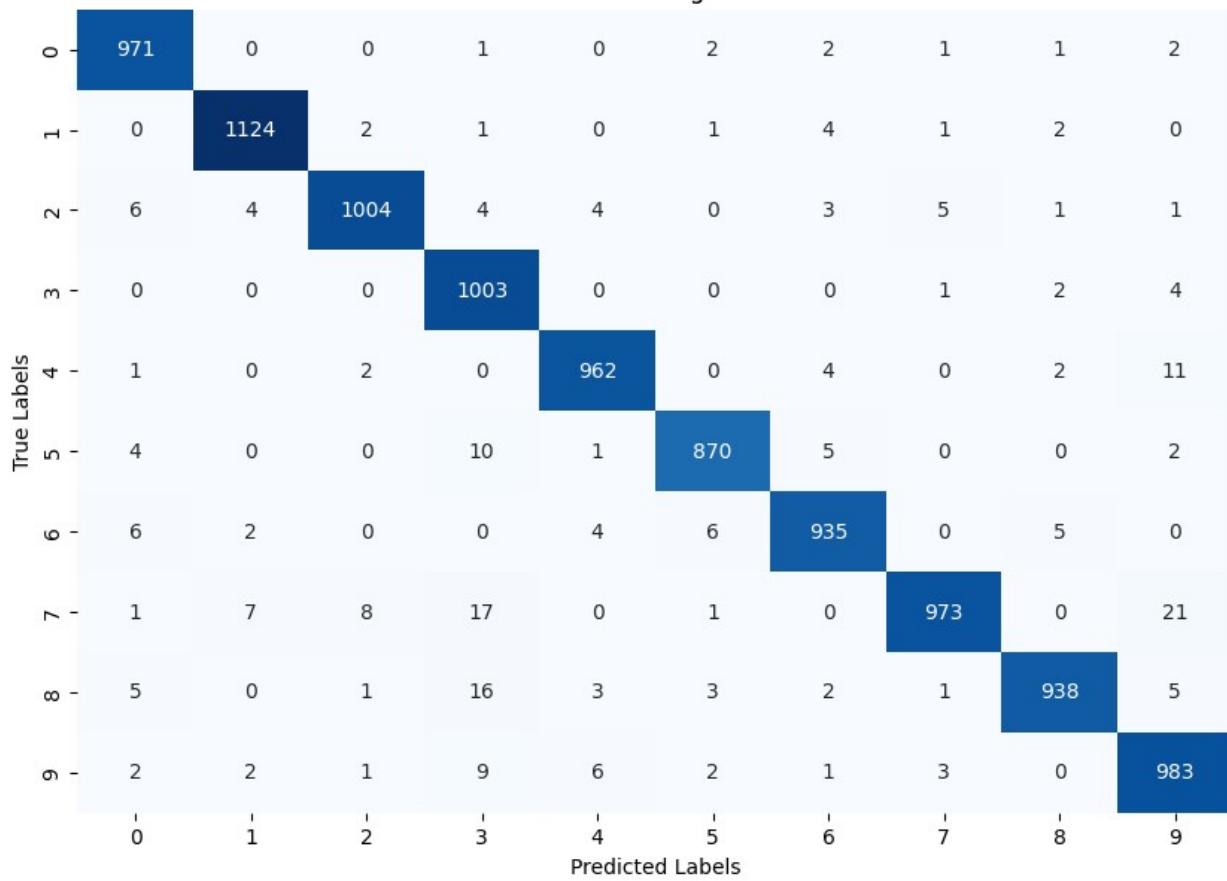
```
Confusion Matrix:
```

```
[[ 971   0   0   1   0   2   2   1   1   2]  
 [ 0 1124   2   1   0   1   4   1   2   0]  
 [ 6   4 1004   4   4   0   3   5   1   1]  
 [ 0   0   0 1003   0   0   0   1   2   4]  
 [ 1   0   2   0 962   0   4   0   2 11]  
 [ 4   0   0   10  1 870   5   0   0   2]  
 [ 6   2   0   0   4   6 935   0   5   0]  
 [ 1   7   8   17  0   1   0 973   0 21]  
 [ 5   0   1   16  3   3   2   1 938   5]  
 [ 2   2   1   9   6   2   1   3   0 983]]
```

```
Precision: 0.9766
```

```
Recall: 0.9763
```

Confusion Matrix for sigmoid Activation



```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 4s - loss: 1.1292 - accuracy: 0.6874 - val_loss: 0.4254 -
val_accuracy: 0.8930 - 4s/epoch - 10ms/step
Epoch 2/5
422/422 - 4s - loss: 0.4199 - accuracy: 0.8806 - val_loss: 0.2938 -
val_accuracy: 0.9157 - 4s/epoch - 9ms/step
Epoch 3/5
422/422 - 4s - loss: 0.3479 - accuracy: 0.8969 - val_loss: 0.2573 -
val_accuracy: 0.9265 - 4s/epoch - 9ms/step
Epoch 4/5
422/422 - 4s - loss: 0.3094 - accuracy: 0.9074 - val_loss: 0.2358 -
val_accuracy: 0.9308 - 4s/epoch - 9ms/step
Epoch 5/5
422/422 - 4s - loss: 0.2844 - accuracy: 0.9153 - val_loss: 0.2129 -
val_accuracy: 0.9378 - 4s/epoch - 9ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 951    0     6     1     0     6    12     1     3     0]
 [  0 1108    5     2     0     2     4     0    14     0]
 [  8    0 949    11     9     1    17    10    24     3]
 [  2    1   27   919     0    23     2    15    17     4]
 [  1    3     6     0   927     1    16     1     3    24]
 [  7    3     7    30     6   786    25     4    20     4]
 [  8    3     4     0     6     8   924     3     2     0]
 [  2    9    37     5    10     0     0   938     1    26]
 [  6    3   13    20     9    19    11     7   874    12]
 [ 10    7     6   13    44     6     0    17     8   898]]
```

Precision: 0.9275
Recall: 0.9274

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	951	0	6	1	0	6	12	1	3	0	
0	0	1108	5	2	0	2	4	0	14	0	
1	8	0	949	11	9	1	17	10	24	3	
2	2	1	27	919	0	23	2	15	17	4	
3	1	3	6	0	927	1	16	1	3	24	
4	7	3	7	30	6	786	25	4	20	4	
5	8	3	4	0	6	8	924	3	2	0	
6	2	9	37	5	10	0	0	938	1	26	
7	6	3	13	20	9	19	11	7	874	12	
8	10	7	6	13	44	6	0	17	8	898	
9	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epochs..
```

```
Epoch 1/15
```

```
422/422 - 4s - loss: 1.0934 - accuracy: 0.7085 - val_loss: 0.4032 -
val_accuracy: 0.9025 - 4s/epoch - 10ms/step
```

```
Epoch 2/15
```

```
422/422 - 4s - loss: 0.4092 - accuracy: 0.8833 - val_loss: 0.2889 -
val_accuracy: 0.9162 - 4s/epoch - 9ms/step
```

```
Epoch 3/15
```

```
422/422 - 4s - loss: 0.3338 - accuracy: 0.9014 - val_loss: 0.2545 -
val_accuracy: 0.9277 - 4s/epoch - 9ms/step
```

```
Epoch 4/15
```

```
422/422 - 4s - loss: 0.3008 - accuracy: 0.9113 - val_loss: 0.2401 -
val_accuracy: 0.9302 - 4s/epoch - 9ms/step
```

```
Epoch 5/15
```

```
422/422 - 4s - loss: 0.2768 - accuracy: 0.9163 - val_loss: 0.2098 -
val_accuracy: 0.9383 - 4s/epoch - 9ms/step
```

```
Epoch 6/15
```

```
422/422 - 4s - loss: 0.2522 - accuracy: 0.9245 - val_loss: 0.1900 -
val_accuracy: 0.9453 - 4s/epoch - 9ms/step
```

```
Epoch 7/15
```

```
422/422 - 4s - loss: 0.2362 - accuracy: 0.9300 - val_loss: 0.1769 -  
val_accuracy: 0.9500 - 4s/epoch - 9ms/step  
Epoch 8/15  
422/422 - 4s - loss: 0.2187 - accuracy: 0.9349 - val_loss: 0.1704 -  
val_accuracy: 0.9517 - 4s/epoch - 9ms/step  
Epoch 9/15  
422/422 - 4s - loss: 0.2031 - accuracy: 0.9395 - val_loss: 0.1563 -  
val_accuracy: 0.9565 - 4s/epoch - 9ms/step  
Epoch 10/15  
422/422 - 4s - loss: 0.1890 - accuracy: 0.9443 - val_loss: 0.1469 -  
val_accuracy: 0.9605 - 4s/epoch - 9ms/step  
Epoch 11/15  
422/422 - 4s - loss: 0.1763 - accuracy: 0.9484 - val_loss: 0.1425 -  
val_accuracy: 0.9627 - 4s/epoch - 9ms/step  
Epoch 12/15  
422/422 - 4s - loss: 0.1674 - accuracy: 0.9507 - val_loss: 0.1297 -  
val_accuracy: 0.9645 - 4s/epoch - 9ms/step  
Epoch 13/15  
422/422 - 4s - loss: 0.1569 - accuracy: 0.9542 - val_loss: 0.1285 -  
val_accuracy: 0.9665 - 4s/epoch - 9ms/step  
Epoch 14/15  
422/422 - 4s - loss: 0.1468 - accuracy: 0.9574 - val_loss: 0.1148 -  
val_accuracy: 0.9695 - 4s/epoch - 9ms/step  
Epoch 15/15  
422/422 - 4s - loss: 0.1382 - accuracy: 0.9598 - val_loss: 0.1091 -  
val_accuracy: 0.9715 - 4s/epoch - 9ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 968 0 2 1 0 1 5 1 2 0]  
[ 0 1127 2 1 0 1 3 0 1 0]  
[ 4 4 990 4 7 0 5 10 7 1]  
[ 0 0 11 966 0 8 0 13 8 4]  
[ 1 1 5 0 957 0 3 2 2 11]  
[ 5 2 3 14 1 848 5 3 8 3]  
[ 9 3 4 1 7 8 921 1 4 0]  
[ 0 11 16 5 2 0 0 983 0 11]  
[ 5 2 7 10 6 2 4 7 926 5]  
[ 8 7 1 12 21 3 1 16 4 936]]  
Precision: 0.9622  
Recall: 0.9622
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	968	0	2	1	0	1	5	1	2	0	
1	0	1127	2	1	0	1	3	0	1	0	
2	4	4	990	4	7	0	5	10	7	1	
3	0	0	11	966	0	8	0	13	8	4	
4	1	1	5	0	957	0	3	2	2	11	
5	5	2	3	14	1	848	5	3	8	3	
6	9	3	4	1	7	8	921	1	4	0	
7	0	11	16	5	2	0	0	983	0	11	
8	5	2	7	10	6	2	4	7	926	5	
9	8	7	1	12	21	3	1	16	4	936	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

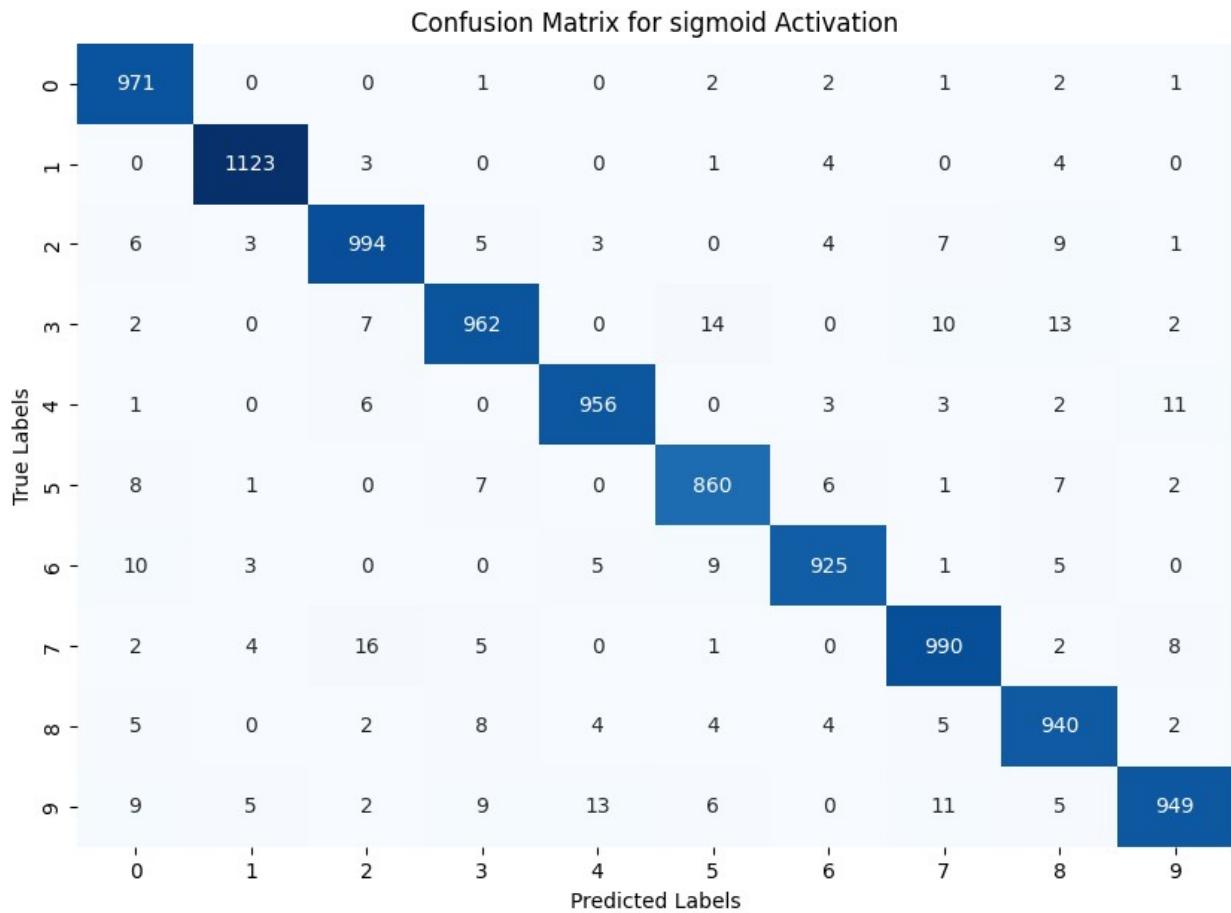
```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 4s - loss: 1.0761 - accuracy: 0.7082 - val_loss: 0.4059 -
val_accuracy: 0.8992 - 4s/epoch - 10ms/step
Epoch 2/20
422/422 - 4s - loss: 0.4125 - accuracy: 0.8822 - val_loss: 0.2940 -
val_accuracy: 0.9127 - 4s/epoch - 9ms/step
Epoch 3/20
422/422 - 4s - loss: 0.3428 - accuracy: 0.8990 - val_loss: 0.2562 -
val_accuracy: 0.9267 - 4s/epoch - 9ms/step
Epoch 4/20
422/422 - 4s - loss: 0.3071 - accuracy: 0.9094 - val_loss: 0.2350 -
val_accuracy: 0.9307 - 4s/epoch - 9ms/step
Epoch 5/20
422/422 - 4s - loss: 0.2837 - accuracy: 0.9159 - val_loss: 0.2227 -
val_accuracy: 0.9323 - 4s/epoch - 9ms/step
Epoch 6/20
422/422 - 4s - loss: 0.2624 - accuracy: 0.9229 - val_loss: 0.2045 -
val_accuracy: 0.9415 - 4s/epoch - 9ms/step
Epoch 7/20
```

```
422/422 - 4s - loss: 0.2456 - accuracy: 0.9264 - val_loss: 0.1958 -  
val_accuracy: 0.9410 - 4s/epoch - 9ms/step  
Epoch 8/20  
422/422 - 4s - loss: 0.2285 - accuracy: 0.9321 - val_loss: 0.1775 -  
val_accuracy: 0.9478 - 4s/epoch - 9ms/step  
Epoch 9/20  
422/422 - 4s - loss: 0.2159 - accuracy: 0.9352 - val_loss: 0.1676 -  
val_accuracy: 0.9523 - 4s/epoch - 9ms/step  
Epoch 10/20  
422/422 - 4s - loss: 0.2015 - accuracy: 0.9399 - val_loss: 0.1629 -  
val_accuracy: 0.9538 - 4s/epoch - 9ms/step  
Epoch 11/20  
422/422 - 4s - loss: 0.1878 - accuracy: 0.9437 - val_loss: 0.1527 -  
val_accuracy: 0.9568 - 4s/epoch - 9ms/step  
Epoch 12/20  
422/422 - 4s - loss: 0.1770 - accuracy: 0.9477 - val_loss: 0.1434 -  
val_accuracy: 0.9597 - 4s/epoch - 9ms/step  
Epoch 13/20  
422/422 - 4s - loss: 0.1673 - accuracy: 0.9507 - val_loss: 0.1281 -  
val_accuracy: 0.9642 - 4s/epoch - 9ms/step  
Epoch 14/20  
422/422 - 4s - loss: 0.1578 - accuracy: 0.9532 - val_loss: 0.1271 -  
val_accuracy: 0.9655 - 4s/epoch - 9ms/step  
Epoch 15/20  
422/422 - 4s - loss: 0.1494 - accuracy: 0.9554 - val_loss: 0.1190 -  
val_accuracy: 0.9663 - 4s/epoch - 9ms/step  
Epoch 16/20  
422/422 - 4s - loss: 0.1408 - accuracy: 0.9588 - val_loss: 0.1181 -  
val_accuracy: 0.9680 - 4s/epoch - 9ms/step  
Epoch 17/20  
422/422 - 4s - loss: 0.1329 - accuracy: 0.9616 - val_loss: 0.1081 -  
val_accuracy: 0.9693 - 4s/epoch - 9ms/step  
Epoch 18/20  
422/422 - 4s - loss: 0.1268 - accuracy: 0.9632 - val_loss: 0.1058 -  
val_accuracy: 0.9722 - 4s/epoch - 9ms/step  
Epoch 19/20  
422/422 - 4s - loss: 0.1216 - accuracy: 0.9641 - val_loss: 0.1022 -  
val_accuracy: 0.9722 - 4s/epoch - 9ms/step  
Epoch 20/20  
422/422 - 4s - loss: 0.1164 - accuracy: 0.9660 - val_loss: 0.0937 -  
val_accuracy: 0.9753 - 4s/epoch - 9ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 971  0  0  1  0  2  2  1  2  1]  
 [ 0 1123  3  0  0  1  4  0  4  0]  
 [ 6  3 994  5  3  0  4  7  9  1]  
 [ 2  0  7 962  0  14  0  10  13  2]  
 [ 1  0  6  0  956  0  3  3  2  11]]
```

```
[ 8  1  0  7  0  860  6  1  7  2]
[ 10 3  0  0  5  9  925  1  5  0]
[ 2  4  16 5  0  1  0  990  2  8]
[ 5  0  2  8  4  4  4  5  940  2]
[ 9  5  2  9  13 6  0  11  5  949]]
```

Precision: 0.9671

Recall: 0.9670



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 4s - loss: 1.4486 - accuracy: 0.6194 - val_loss: 0.6641 - val_accuracy: 0.8758 - 4s/epoch - 17ms/step

Epoch 2/5

211/211 - 3s - loss: 0.5682 - accuracy: 0.8552 - val_loss: 0.3890 - val_accuracy: 0.9017 - 3s/epoch - 14ms/step

Epoch 3/5

211/211 - 3s - loss: 0.4239 - accuracy: 0.8820 - val_loss: 0.3259 - val_accuracy: 0.9122 - 3s/epoch - 14ms/step

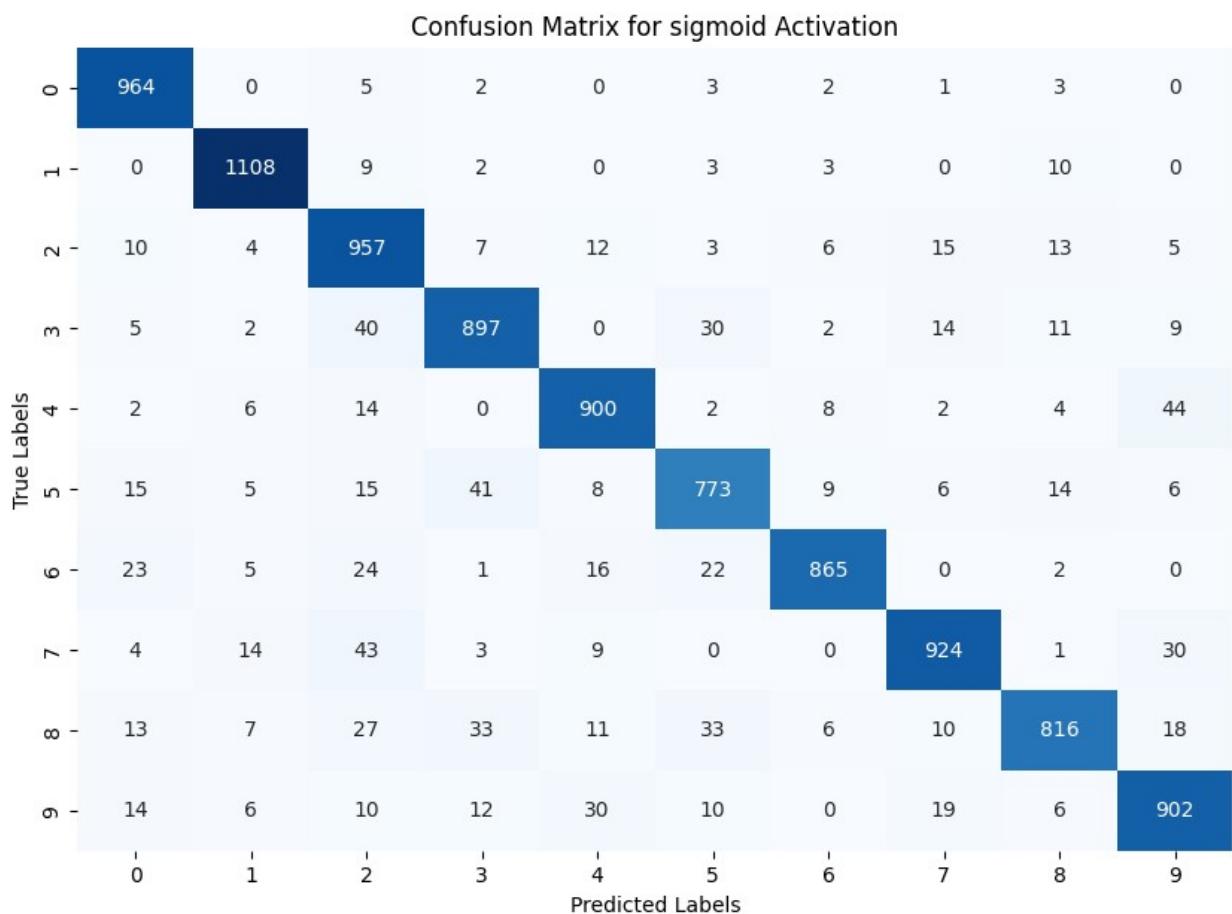
Epoch 4/5

211/211 - 3s - loss: 0.3707 - accuracy: 0.8934 - val_loss: 0.2858 -

```

val_accuracy: 0.9198 - 3s/epoch - 14ms/step
Epoch 5/5
211/211 - 3s - loss: 0.3402 - accuracy: 0.9005 - val_loss: 0.2707 -
val_accuracy: 0.9207 - 3s/epoch - 15ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 964 0 5 2 0 3 2 1 3 0]
 [ 0 1108 9 2 0 3 3 0 10 0]
 [ 10 4 957 7 12 3 6 15 13 5]
 [ 5 2 40 897 0 30 2 14 11 9]
 [ 2 6 14 0 900 2 8 2 4 44]
 [ 15 5 15 41 8 773 9 6 14 6]
 [ 23 5 24 1 16 22 865 0 2 0]
 [ 4 14 43 3 9 0 0 924 1 30]
 [ 13 7 27 33 11 33 6 10 816 18]
 [ 14 6 10 12 30 10 0 19 6 902]]
Precision: 0.9117
Recall: 0.9106

```

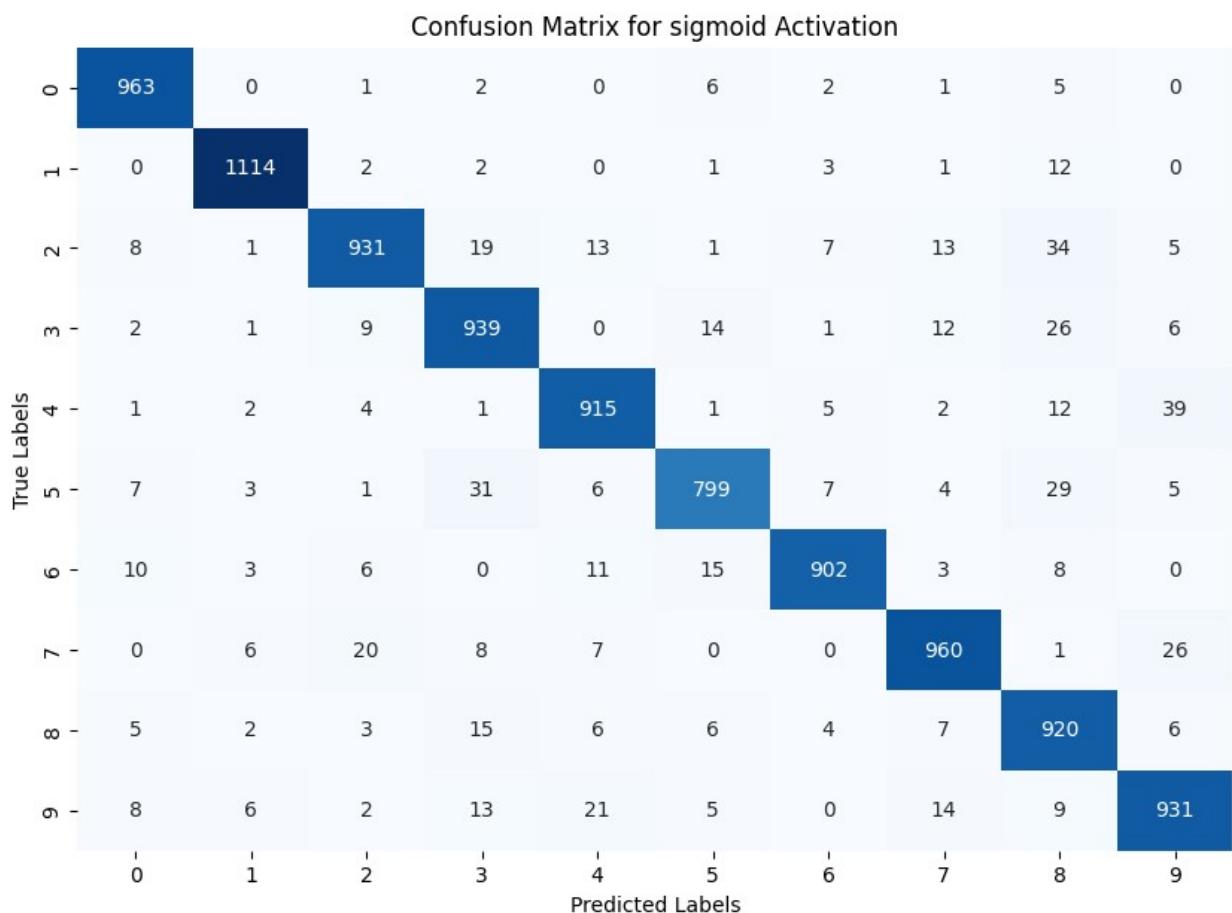


```
Training Model with sigmoid activation, 1 conv_layers, 1 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 4s - loss: 1.5713 - accuracy: 0.5871 - val_loss: 0.7496 -
val_accuracy: 0.8610 - 4s/epoch - 18ms/step
Epoch 2/15
211/211 - 3s - loss: 0.6045 - accuracy: 0.8480 - val_loss: 0.4037 -
val_accuracy: 0.9033 - 3s/epoch - 15ms/step
Epoch 3/15
211/211 - 3s - loss: 0.4341 - accuracy: 0.8788 - val_loss: 0.3301 -
val_accuracy: 0.9113 - 3s/epoch - 15ms/step
Epoch 4/15
211/211 - 3s - loss: 0.3782 - accuracy: 0.8917 - val_loss: 0.2897 -
val_accuracy: 0.9157 - 3s/epoch - 15ms/step
Epoch 5/15
211/211 - 3s - loss: 0.3462 - accuracy: 0.8987 - val_loss: 0.2663 -
val_accuracy: 0.9252 - 3s/epoch - 14ms/step
Epoch 6/15
211/211 - 3s - loss: 0.3227 - accuracy: 0.9046 - val_loss: 0.2522 -
val_accuracy: 0.9253 - 3s/epoch - 15ms/step
Epoch 7/15
211/211 - 3s - loss: 0.3072 - accuracy: 0.9093 - val_loss: 0.2426 -
val_accuracy: 0.9288 - 3s/epoch - 15ms/step
Epoch 8/15
211/211 - 3s - loss: 0.2936 - accuracy: 0.9127 - val_loss: 0.2335 -
val_accuracy: 0.9323 - 3s/epoch - 15ms/step
Epoch 9/15
211/211 - 3s - loss: 0.2817 - accuracy: 0.9163 - val_loss: 0.2199 -
val_accuracy: 0.9358 - 3s/epoch - 15ms/step
Epoch 10/15
211/211 - 3s - loss: 0.2721 - accuracy: 0.9190 - val_loss: 0.2123 -
val_accuracy: 0.9380 - 3s/epoch - 15ms/step
Epoch 11/15
211/211 - 3s - loss: 0.2618 - accuracy: 0.9222 - val_loss: 0.2020 -
val_accuracy: 0.9432 - 3s/epoch - 15ms/step
Epoch 12/15
211/211 - 3s - loss: 0.2526 - accuracy: 0.9258 - val_loss: 0.2063 -
val_accuracy: 0.9380 - 3s/epoch - 15ms/step
Epoch 13/15
211/211 - 3s - loss: 0.2435 - accuracy: 0.9279 - val_loss: 0.1880 -
val_accuracy: 0.9483 - 3s/epoch - 15ms/step
Epoch 14/15
211/211 - 3s - loss: 0.2354 - accuracy: 0.9305 - val_loss: 0.1848 -
val_accuracy: 0.9468 - 3s/epoch - 15ms/step
Epoch 15/15
211/211 - 3s - loss: 0.2278 - accuracy: 0.9328 - val_loss: 0.1799 -
val_accuracy: 0.9483 - 3s/epoch - 15ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
```

```
[[ 963  0  1  2  0  6  2  1  5  0]
 [ 0 1114  2  2  0  1  3  1 12  0]
 [ 8  1 931 19 13  1  7 13 34  5]
 [ 2  1  9 939  0 14  1 12 26  6]
 [ 1  2  4  1 915  1  5  2 12 39]
 [ 7  3  1 31  6 799  7  4 29  5]
 [10  3  6  0 11 15 902  3  8  0]
 [ 0  6 20  8  7  0  0 960  1 26]
 [ 5  2  3 15  6  6  4  7 920  6]
 [ 8  6  2 13 21  5  0 14  9 931]]
```

Precision: 0.9381

Recall: 0.9374



Training Model with sigmoid activation, 1 conv_layers, 1 dense layers, 256 batch size, 20 epochs..

Epoch 1/20

211/211 - 4s - loss: 1.4757 - accuracy: 0.6297 - val_loss: 0.6821 - val_accuracy: 0.8648 - 4s/epoch - 17ms/step

Epoch 2/20

211/211 - 3s - loss: 0.5702 - accuracy: 0.8542 - val_loss: 0.3923 - val_accuracy: 0.9035 - 3s/epoch - 14ms/step

```
Epoch 3/20
211/211 - 3s - loss: 0.4224 - accuracy: 0.8825 - val_loss: 0.3201 -
val_accuracy: 0.9108 - 3s/epoch - 14ms/step
Epoch 4/20
211/211 - 3s - loss: 0.3693 - accuracy: 0.8933 - val_loss: 0.2938 -
val_accuracy: 0.9168 - 3s/epoch - 14ms/step
Epoch 5/20
211/211 - 3s - loss: 0.3402 - accuracy: 0.9018 - val_loss: 0.2657 -
val_accuracy: 0.9238 - 3s/epoch - 14ms/step
Epoch 6/20
211/211 - 3s - loss: 0.3199 - accuracy: 0.9061 - val_loss: 0.2493 -
val_accuracy: 0.9272 - 3s/epoch - 14ms/step
Epoch 7/20
211/211 - 3s - loss: 0.3059 - accuracy: 0.9104 - val_loss: 0.2369 -
val_accuracy: 0.9317 - 3s/epoch - 14ms/step
Epoch 8/20
211/211 - 3s - loss: 0.2929 - accuracy: 0.9133 - val_loss: 0.2280 -
val_accuracy: 0.9337 - 3s/epoch - 14ms/step
Epoch 9/20
211/211 - 3s - loss: 0.2811 - accuracy: 0.9173 - val_loss: 0.2230 -
val_accuracy: 0.9348 - 3s/epoch - 14ms/step
Epoch 10/20
211/211 - 3s - loss: 0.2699 - accuracy: 0.9206 - val_loss: 0.2105 -
val_accuracy: 0.9395 - 3s/epoch - 14ms/step
Epoch 11/20
211/211 - 3s - loss: 0.2624 - accuracy: 0.9224 - val_loss: 0.2009 -
val_accuracy: 0.9445 - 3s/epoch - 14ms/step
Epoch 12/20
211/211 - 3s - loss: 0.2522 - accuracy: 0.9254 - val_loss: 0.1971 -
val_accuracy: 0.9445 - 3s/epoch - 14ms/step
Epoch 13/20
211/211 - 3s - loss: 0.2443 - accuracy: 0.9278 - val_loss: 0.1882 -
val_accuracy: 0.9455 - 3s/epoch - 15ms/step
Epoch 14/20
211/211 - 3s - loss: 0.2357 - accuracy: 0.9311 - val_loss: 0.1835 -
val_accuracy: 0.9488 - 3s/epoch - 15ms/step
Epoch 15/20
211/211 - 3s - loss: 0.2267 - accuracy: 0.9334 - val_loss: 0.1755 -
val_accuracy: 0.9502 - 3s/epoch - 14ms/step
Epoch 16/20
211/211 - 3s - loss: 0.2200 - accuracy: 0.9353 - val_loss: 0.1717 -
val_accuracy: 0.9510 - 3s/epoch - 14ms/step
Epoch 17/20
211/211 - 3s - loss: 0.2130 - accuracy: 0.9375 - val_loss: 0.1651 -
val_accuracy: 0.9538 - 3s/epoch - 14ms/step
Epoch 18/20
211/211 - 3s - loss: 0.2072 - accuracy: 0.9385 - val_loss: 0.1689 -
val_accuracy: 0.9505 - 3s/epoch - 14ms/step
Epoch 19/20
```

```
211/211 - 3s - loss: 0.2002 - accuracy: 0.9414 - val_loss: 0.1548 -  
val_accuracy: 0.9580 - 3s/epoch - 14ms/step
```

```
Epoch 20/20
```

```
211/211 - 3s - loss: 0.1937 - accuracy: 0.9432 - val_loss: 0.1525 -  
val_accuracy: 0.9568 - 3s/epoch - 14ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: sigmoid
```

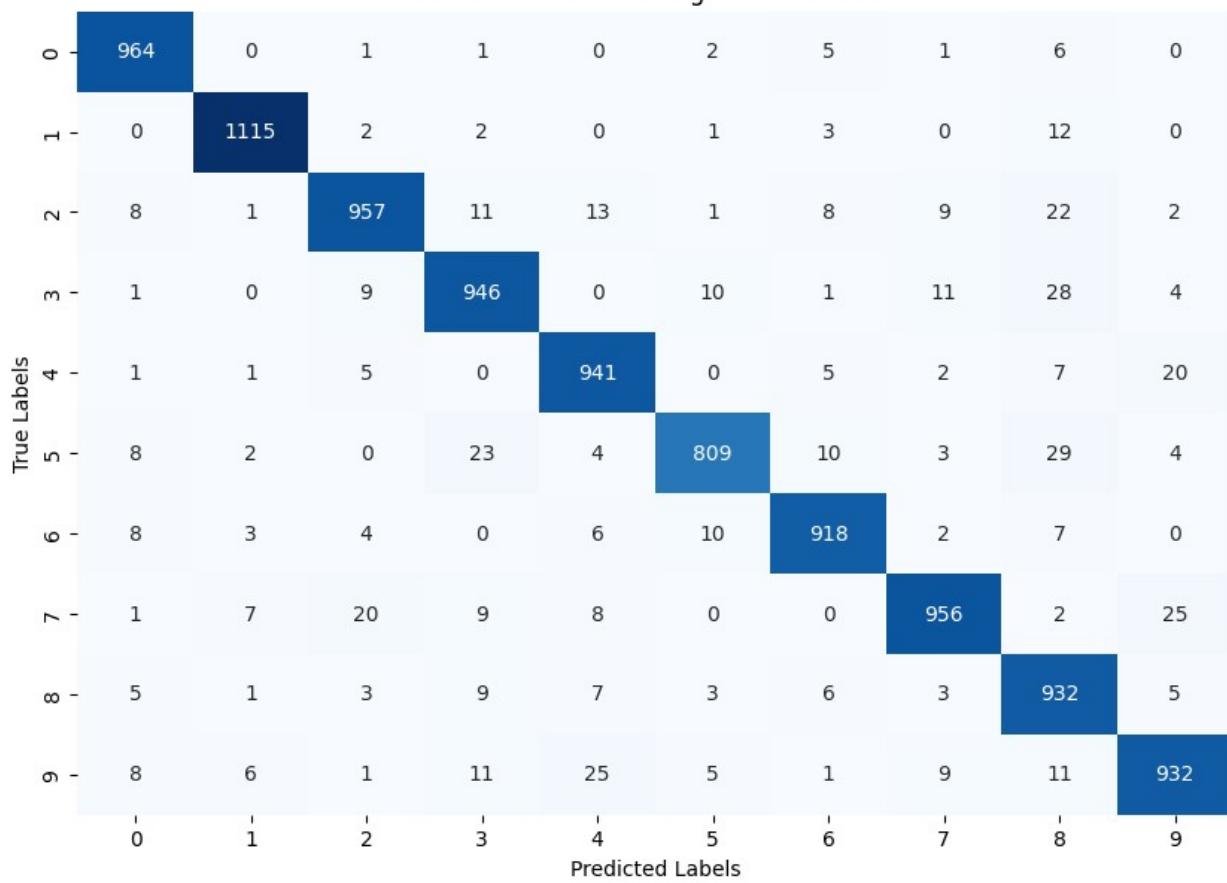
```
Confusion Matrix:
```

```
[[ 964   0   1   1   0   2   5   1   6   0]
 [ 0 1115   2   2   0   1   3   0  12   0]
 [ 8   1 957  11  13   1   8   9  22   2]
 [ 1   0   9 946   0  10   1  11  28   4]
 [ 1   1   5   0 941   0   5   2   7  20]
 [ 8   2   0  23   4 809  10   3  29   4]
 [ 8   3   4   0   6  10 918   2   7   0]
 [ 1   7  20   9   8   0   0 956   2  25]
 [ 5   1   3   9   7   3   6   3 932   5]
 [ 8   6   1 11  25   5   1   9 11 932]]
```

```
Precision: 0.9476
```

```
Recall: 0.9470
```

Confusion Matrix for sigmoid Activation



```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
64 batch size, 5 epochs..
Epoch 1/5
844/844 - 7s - loss: 2.2977 - accuracy: 0.1216 - val_loss: 2.2667 -
val_accuracy: 0.1168 - 7s/epoch - 8ms/step
Epoch 2/5
844/844 - 6s - loss: 1.5979 - accuracy: 0.5089 - val_loss: 0.7387 -
val_accuracy: 0.8133 - 6s/epoch - 7ms/step
Epoch 3/5
844/844 - 6s - loss: 0.5548 - accuracy: 0.8424 - val_loss: 0.3452 -
val_accuracy: 0.9108 - 6s/epoch - 8ms/step
Epoch 4/5
844/844 - 6s - loss: 0.3248 - accuracy: 0.9087 - val_loss: 0.2215 -
val_accuracy: 0.9395 - 6s/epoch - 8ms/step
Epoch 5/5
844/844 - 6s - loss: 0.2259 - accuracy: 0.9354 - val_loss: 0.1589 -
val_accuracy: 0.9595 - 6s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 969     0     2     0     0     1     3     1     4     0]
 [  0 1106     4     2     0     0     4     1    17     1]
 [  9     0 966    11     5     0     4    12    21     4]
 [  0     0     8 968     0    11     0     7    10     6]
 [  1     2     3     1 912     0     9     0     4    50]
 [  9     1     2    25     1 824     3     3    19     5]
 [ 13     3     3     0     6    13 916     0     4     0]
 [  2     5    21     9     1     0     0 954     2    34]
 [  6     0     5    18     3     2     5     5    915    15]
 [ 11     5     0     6     9     4     1     9    14 950]]
```

Precision: 0.9488
Recall: 0.9480

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	969	0	2	0	0	1	3	1	4	0	
1	0	1106	4	2	0	0	4	1	17	1	
2	9	0	966	11	5	0	4	12	21	4	
3	0	0	8	968	0	11	0	7	10	6	
4	1	2	3	1	912	0	9	0	4	50	
5	9	1	2	25	1	824	3	3	19	5	
6	13	3	3	0	6	13	916	0	4	0	
7	2	5	21	9	1	0	0	954	2	34	
8	6	0	5	18	3	2	5	5	915	15	
9	11	5	0	6	9	4	1	9	14	950	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
True Labels	0	1	2	3	4	5	6	7	8	9	Predicted Labels

Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 64 batch size, 15 epochs..

Epoch 1/15

844/844 - 7s - loss: 2.2775 - accuracy: 0.1561 - val_loss: 2.1248 - val_accuracy: 0.3400 - 7s/epoch - 9ms/step

Epoch 2/15

844/844 - 6s - loss: 1.2054 - accuracy: 0.6552 - val_loss: 0.5750 - val_accuracy: 0.8530 - 6s/epoch - 8ms/step

Epoch 3/15

844/844 - 6s - loss: 0.4770 - accuracy: 0.8682 - val_loss: 0.3034 - val_accuracy: 0.9203 - 6s/epoch - 8ms/step

Epoch 4/15

844/844 - 6s - loss: 0.2988 - accuracy: 0.9156 - val_loss: 0.1984 - val_accuracy: 0.9483 - 6s/epoch - 7ms/step

Epoch 5/15

844/844 - 7s - loss: 0.2134 - accuracy: 0.9378 - val_loss: 0.1520 - val_accuracy: 0.9608 - 7s/epoch - 8ms/step

Epoch 6/15

844/844 - 6s - loss: 0.1665 - accuracy: 0.9516 - val_loss: 0.1174 - val_accuracy: 0.9677 - 6s/epoch - 8ms/step

Epoch 7/15

```
844/844 - 6s - loss: 0.1388 - accuracy: 0.9593 - val_loss: 0.1041 -  
val_accuracy: 0.9708 - 6s/epoch - 8ms/step  
Epoch 8/15  
844/844 - 6s - loss: 0.1201 - accuracy: 0.9646 - val_loss: 0.0889 -  
val_accuracy: 0.9743 - 6s/epoch - 8ms/step  
Epoch 9/15  
844/844 - 6s - loss: 0.1072 - accuracy: 0.9685 - val_loss: 0.0819 -  
val_accuracy: 0.9765 - 6s/epoch - 8ms/step  
Epoch 10/15  
844/844 - 6s - loss: 0.0981 - accuracy: 0.9709 - val_loss: 0.0779 -  
val_accuracy: 0.9780 - 6s/epoch - 8ms/step  
Epoch 11/15  
844/844 - 6s - loss: 0.0898 - accuracy: 0.9736 - val_loss: 0.0728 -  
val_accuracy: 0.9787 - 6s/epoch - 8ms/step  
Epoch 12/15  
844/844 - 6s - loss: 0.0840 - accuracy: 0.9753 - val_loss: 0.0688 -  
val_accuracy: 0.9810 - 6s/epoch - 8ms/step  
Epoch 13/15  
844/844 - 6s - loss: 0.0786 - accuracy: 0.9764 - val_loss: 0.0637 -  
val_accuracy: 0.9825 - 6s/epoch - 8ms/step  
Epoch 14/15  
844/844 - 6s - loss: 0.0740 - accuracy: 0.9784 - val_loss: 0.0618 -  
val_accuracy: 0.9830 - 6s/epoch - 8ms/step  
Epoch 15/15  
844/844 - 6s - loss: 0.0703 - accuracy: 0.9790 - val_loss: 0.0588 -  
val_accuracy: 0.9832 - 6s/epoch - 8ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 971  0  0  0  0  0  5  1  3  0]  
[  0 1124  1  1  0  0  4  1  4  0]  
[  5  3 1015  0  3  0  0  4  2  0]  
[  0  0  3 988  0 10  0  6  3  0]  
[  0  0  1  0 973  0  2  1  1  4]  
[  4  0  0  3  0 872  9  1  2  1]  
[  3  2  0  0  2  4 946  0  1  0]  
[  1  2  8  2  0  0  0 1011  2  2]  
[  7  0  4  4  2  1  5  2 947  2]  
[  4  6  0  7 10  5  0  7  0 970]]  
Precision: 0.9817  
Recall: 0.9817
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	971	0	0	0	0	0	5	1	3	0
0	971	0	0	0	0	0	4	1	4	0
1	0	1124	1	1	0	0	4	1	4	0
2	5	3	1015	0	3	0	0	4	2	0
3	0	0	3	988	0	10	0	6	3	0
4	0	0	1	0	973	0	2	1	1	4
5	4	0	0	3	0	872	9	1	2	1
6	3	2	0	0	2	4	946	0	1	0
7	1	2	8	2	0	0	0	1011	2	2
8	7	0	4	4	2	1	5	2	947	2
9	4	6	0	7	10	5	0	7	0	970
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

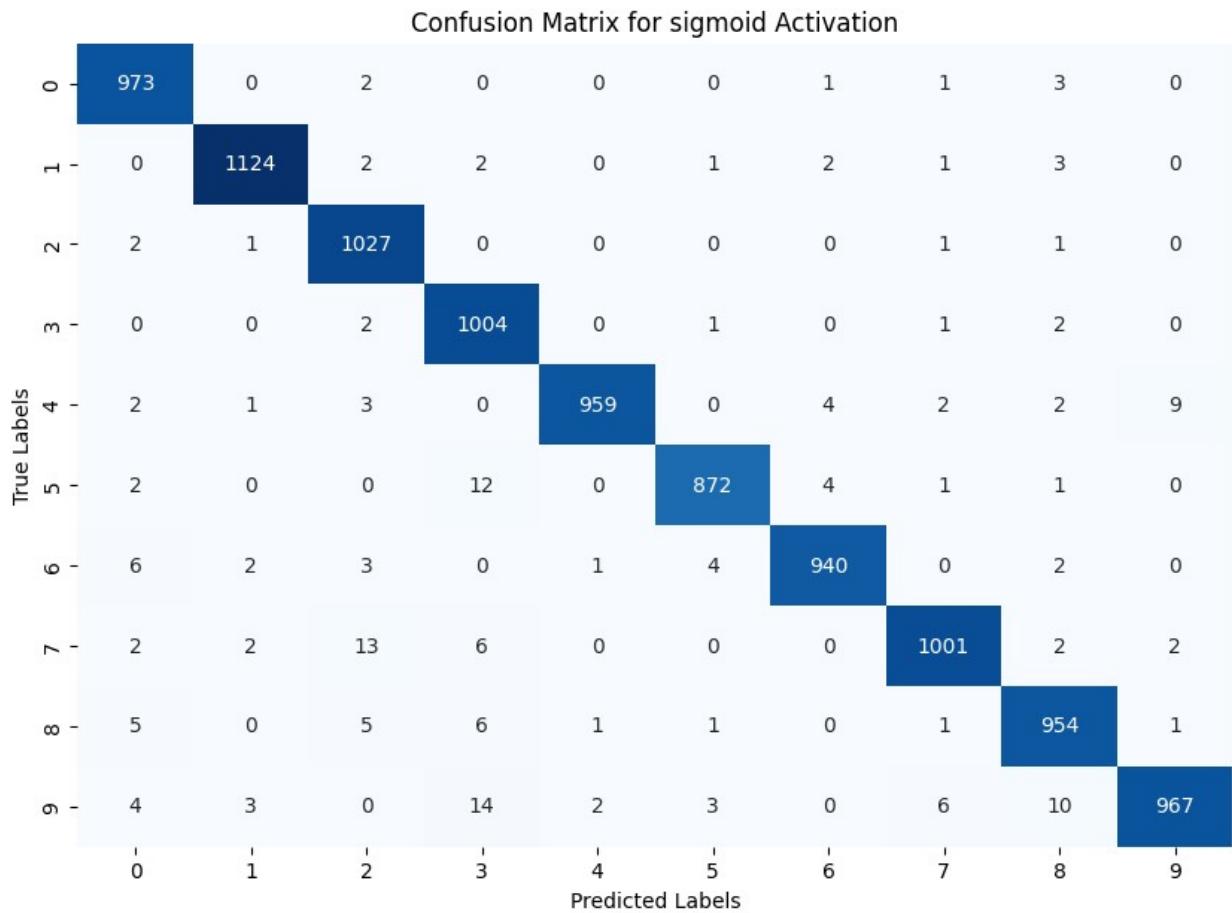
```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 7s - loss: 2.2912 - accuracy: 0.1418 - val_loss: 2.2207 -
val_accuracy: 0.2612 - 7s/epoch - 8ms/step
Epoch 2/20
844/844 - 7s - loss: 1.4176 - accuracy: 0.5817 - val_loss: 0.6522 -
val_accuracy: 0.8373 - 7s/epoch - 8ms/step
Epoch 3/20
844/844 - 7s - loss: 0.5049 - accuracy: 0.8614 - val_loss: 0.3167 -
val_accuracy: 0.9188 - 7s/epoch - 8ms/step
Epoch 4/20
844/844 - 7s - loss: 0.3098 - accuracy: 0.9134 - val_loss: 0.2110 -
val_accuracy: 0.9448 - 7s/epoch - 8ms/step
Epoch 5/20
844/844 - 6s - loss: 0.2258 - accuracy: 0.9349 - val_loss: 0.1601 -
val_accuracy: 0.9563 - 6s/epoch - 8ms/step
Epoch 6/20
844/844 - 6s - loss: 0.1796 - accuracy: 0.9479 - val_loss: 0.1310 -
val_accuracy: 0.9647 - 6s/epoch - 7ms/step
Epoch 7/20
```

```
844/844 - 6s - loss: 0.1491 - accuracy: 0.9557 - val_loss: 0.1108 -  
val_accuracy: 0.9697 - 6s/epoch - 7ms/step  
Epoch 8/20  
844/844 - 6s - loss: 0.1287 - accuracy: 0.9623 - val_loss: 0.0976 -  
val_accuracy: 0.9740 - 6s/epoch - 7ms/step  
Epoch 9/20  
844/844 - 6s - loss: 0.1147 - accuracy: 0.9660 - val_loss: 0.0942 -  
val_accuracy: 0.9730 - 6s/epoch - 7ms/step  
Epoch 10/20  
844/844 - 6s - loss: 0.1032 - accuracy: 0.9700 - val_loss: 0.0814 -  
val_accuracy: 0.9768 - 6s/epoch - 8ms/step  
Epoch 11/20  
844/844 - 6s - loss: 0.0946 - accuracy: 0.9718 - val_loss: 0.0751 -  
val_accuracy: 0.9788 - 6s/epoch - 8ms/step  
Epoch 12/20  
844/844 - 6s - loss: 0.0873 - accuracy: 0.9740 - val_loss: 0.0737 -  
val_accuracy: 0.9790 - 6s/epoch - 8ms/step  
Epoch 13/20  
844/844 - 6s - loss: 0.0814 - accuracy: 0.9753 - val_loss: 0.0697 -  
val_accuracy: 0.9798 - 6s/epoch - 7ms/step  
Epoch 14/20  
844/844 - 6s - loss: 0.0762 - accuracy: 0.9773 - val_loss: 0.0657 -  
val_accuracy: 0.9808 - 6s/epoch - 7ms/step  
Epoch 15/20  
844/844 - 6s - loss: 0.0719 - accuracy: 0.9785 - val_loss: 0.0661 -  
val_accuracy: 0.9808 - 6s/epoch - 8ms/step  
Epoch 16/20  
844/844 - 6s - loss: 0.0683 - accuracy: 0.9794 - val_loss: 0.0617 -  
val_accuracy: 0.9822 - 6s/epoch - 8ms/step  
Epoch 17/20  
844/844 - 6s - loss: 0.0641 - accuracy: 0.9811 - val_loss: 0.0581 -  
val_accuracy: 0.9832 - 6s/epoch - 8ms/step  
Epoch 18/20  
844/844 - 7s - loss: 0.0609 - accuracy: 0.9817 - val_loss: 0.0560 -  
val_accuracy: 0.9823 - 7s/epoch - 8ms/step  
Epoch 19/20  
844/844 - 6s - loss: 0.0588 - accuracy: 0.9826 - val_loss: 0.0562 -  
val_accuracy: 0.9838 - 6s/epoch - 8ms/step  
Epoch 20/20  
844/844 - 6s - loss: 0.0556 - accuracy: 0.9839 - val_loss: 0.0590 -  
val_accuracy: 0.9830 - 6s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 973 0 2 0 0 0 1 1 3 0]  
[ 0 1124 2 2 0 1 2 1 3 0]  
[ 2 1 1027 0 0 0 0 1 1 0]  
[ 0 0 2 1004 0 1 0 1 2 0]  
[ 2 1 3 0 959 0 4 2 2 9]]
```

```
[ 2 0 0 12 0 872 4 1 1 0]
[ 6 2 3 0 1 4 940 0 2 0]
[ 2 2 13 6 0 0 0 1001 2 2]
[ 5 0 5 6 1 1 0 1 954 1]
[ 4 3 0 14 2 3 0 6 10 967]]
```

Precision: 0.9823

Recall: 0.9821



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 5s - loss: 2.3012 - accuracy: 0.1182 - val_loss: 2.2929 - val_accuracy: 0.1828 - 5s/epoch - 13ms/step

Epoch 2/5

422/422 - 5s - loss: 2.2551 - accuracy: 0.2222 - val_loss: 2.1550 - val_accuracy: 0.4750 - 5s/epoch - 11ms/step

Epoch 3/5

422/422 - 4s - loss: 1.7163 - accuracy: 0.4819 - val_loss: 1.2130 - val_accuracy: 0.6672 - 4s/epoch - 11ms/step

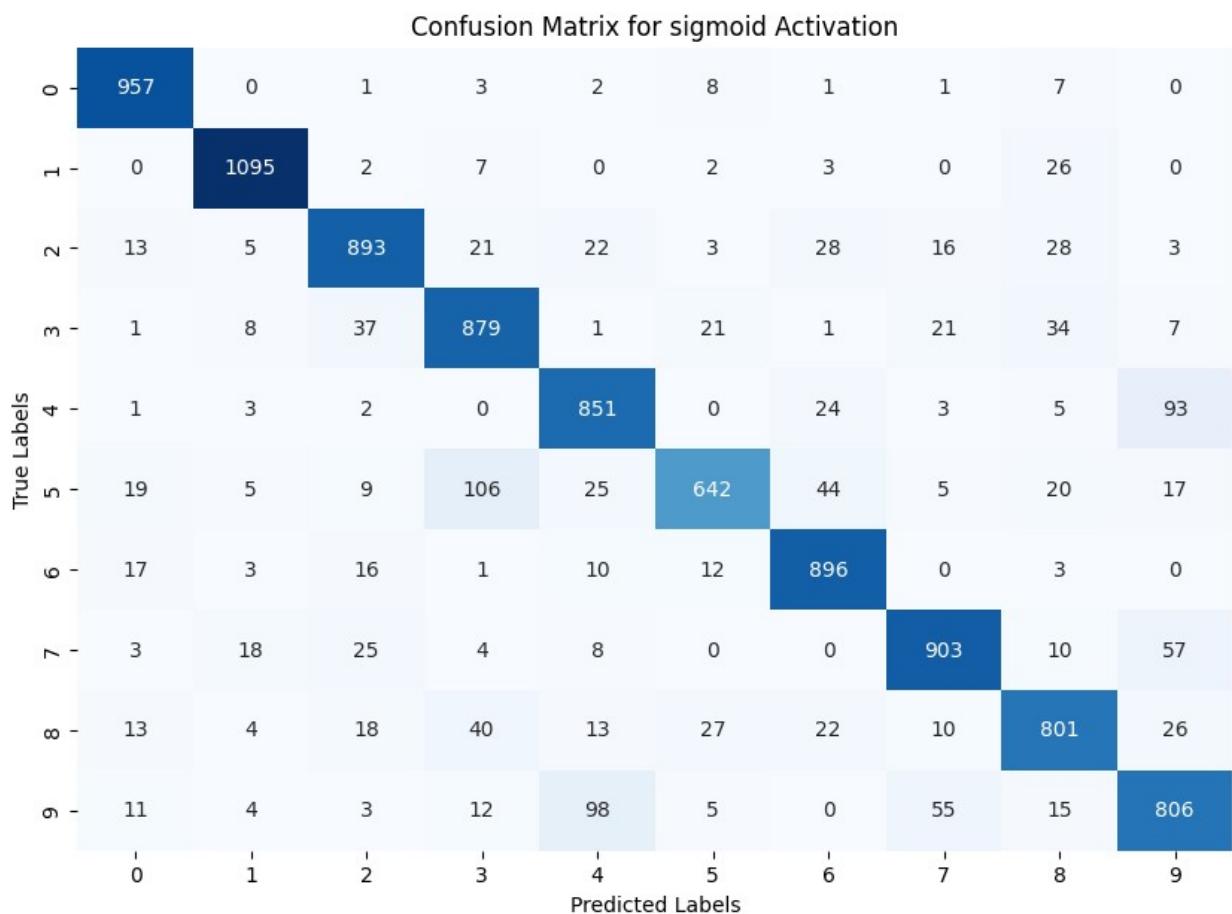
Epoch 4/5

422/422 - 4s - loss: 0.9442 - accuracy: 0.7356 - val_loss: 0.6442 -

```

val_accuracy: 0.8382 - 4s/epoch - 10ms/step
Epoch 5/5
422/422 - 4s - loss: 0.5806 - accuracy: 0.8396 - val_loss: 0.4243 -
val_accuracy: 0.8878 - 4s/epoch - 10ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 957   0   1   3   2   8   1   1   7   0]
 [  0 1095   2   7   0   2   3   0   26   0]
 [ 13   5 893  21  22   3  28  16  28   3]
 [  1   8  37 879   1  21   1  21  34   7]
 [  1   3   2   0 851   0  24   3   5 93]
 [ 19   5   9 106  25 642  44   5  20 17]
 [ 17   3  16   1 10  12 896   0   3   0]
 [  3  18  25   4   8   0   0 903  10 57]
 [ 13   4  18  40  13  27  22  10 801 26]
 [ 11   4   3  12  98   5   0  55  15 806]]
Precision: 0.8728
Recall: 0.8723

```

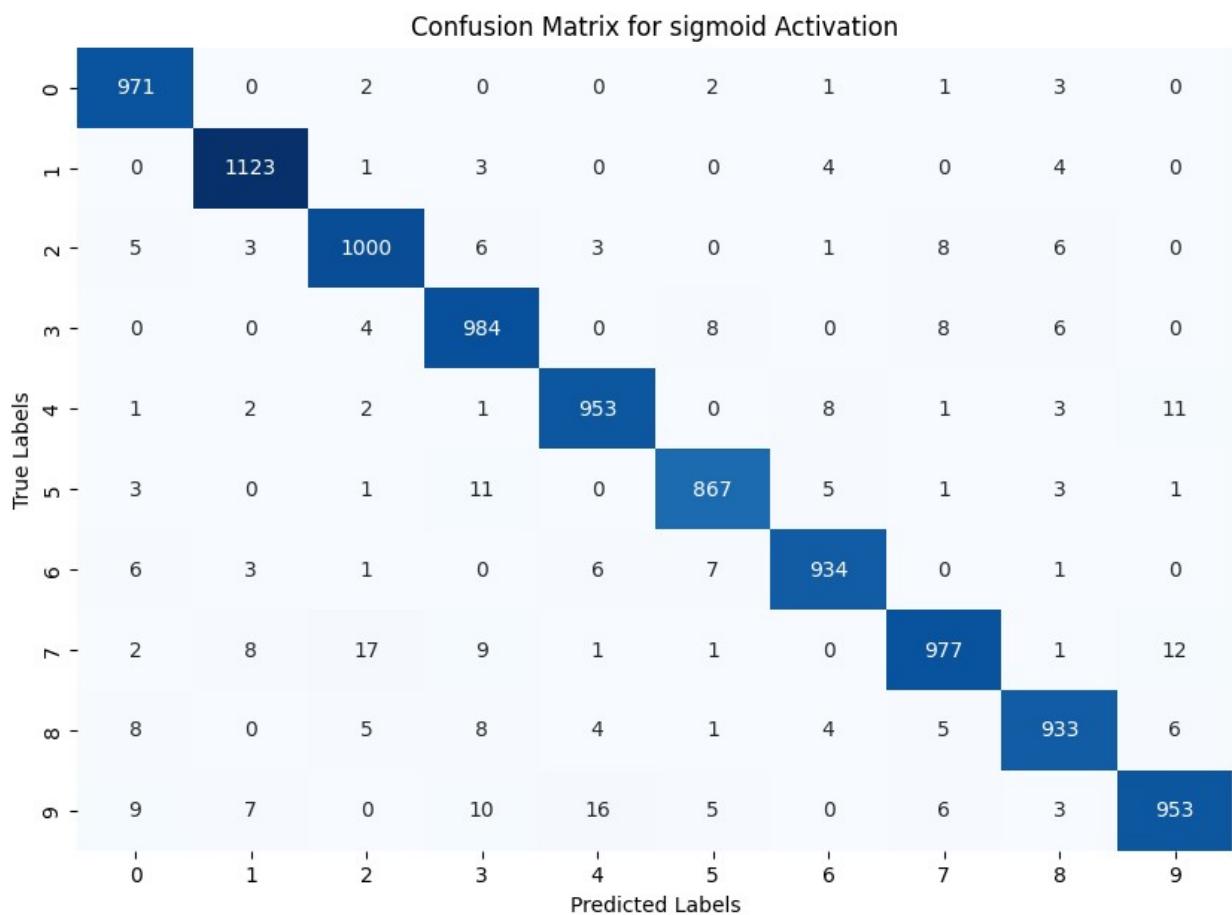


```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 5s - loss: 2.3010 - accuracy: 0.1178 - val_loss: 2.2929 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 2/15
422/422 - 4s - loss: 2.2671 - accuracy: 0.1874 - val_loss: 2.2017 -
val_accuracy: 0.3485 - 4s/epoch - 10ms/step
Epoch 3/15
422/422 - 4s - loss: 1.8334 - accuracy: 0.4661 - val_loss: 1.2807 -
val_accuracy: 0.6938 - 4s/epoch - 10ms/step
Epoch 4/15
422/422 - 4s - loss: 0.9194 - accuracy: 0.7497 - val_loss: 0.6135 -
val_accuracy: 0.8417 - 4s/epoch - 10ms/step
Epoch 5/15
422/422 - 4s - loss: 0.5529 - accuracy: 0.8486 - val_loss: 0.3976 -
val_accuracy: 0.8982 - 4s/epoch - 10ms/step
Epoch 6/15
422/422 - 4s - loss: 0.4053 - accuracy: 0.8873 - val_loss: 0.3002 -
val_accuracy: 0.9228 - 4s/epoch - 11ms/step
Epoch 7/15
422/422 - 4s - loss: 0.3205 - accuracy: 0.9099 - val_loss: 0.2377 -
val_accuracy: 0.9397 - 4s/epoch - 10ms/step
Epoch 8/15
422/422 - 4s - loss: 0.2639 - accuracy: 0.9252 - val_loss: 0.1947 -
val_accuracy: 0.9497 - 4s/epoch - 11ms/step
Epoch 9/15
422/422 - 4s - loss: 0.2244 - accuracy: 0.9358 - val_loss: 0.1672 -
val_accuracy: 0.9557 - 4s/epoch - 11ms/step
Epoch 10/15
422/422 - 4s - loss: 0.1941 - accuracy: 0.9444 - val_loss: 0.1432 -
val_accuracy: 0.9623 - 4s/epoch - 11ms/step
Epoch 11/15
422/422 - 4s - loss: 0.1706 - accuracy: 0.9512 - val_loss: 0.1250 -
val_accuracy: 0.9667 - 4s/epoch - 11ms/step
Epoch 12/15
422/422 - 4s - loss: 0.1534 - accuracy: 0.9562 - val_loss: 0.1157 -
val_accuracy: 0.9692 - 4s/epoch - 10ms/step
Epoch 13/15
422/422 - 4s - loss: 0.1388 - accuracy: 0.9604 - val_loss: 0.1049 -
val_accuracy: 0.9708 - 4s/epoch - 10ms/step
Epoch 14/15
422/422 - 4s - loss: 0.1281 - accuracy: 0.9632 - val_loss: 0.0975 -
val_accuracy: 0.9733 - 4s/epoch - 10ms/step
Epoch 15/15
422/422 - 4s - loss: 0.1185 - accuracy: 0.9656 - val_loss: 0.0914 -
val_accuracy: 0.9748 - 4s/epoch - 10ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
```

```
[[ 971  0  2  0  0  2  1  1  3  0]
 [ 0 1123  1  3  0  0  4  0  4  0]
 [ 5  3 1000  6  3  0  1  8  6  0]
 [ 0  0  4  984  0  8  0  8  6  0]
 [ 1  2  2  1  953  0  8  1  3  11]
 [ 3  0  1  11  0  867  5  1  3  1]
 [ 6  3  1  0  6  7  934  0  1  0]
 [ 2  8  17  9  1  1  0  977  1  12]
 [ 8  0  5  8  4  1  4  5  933  6]
 [ 9  7  0  10  16  5  0  6  3  953]]
```

Precision: 0.9695

Recall: 0.9695



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 128 batch size, 20 epochs..

Epoch 1/20

422/422 - 5s - loss: 2.3019 - accuracy: 0.1111 - val_loss: 2.2934 - val_accuracy: 0.1050 - 5s/epoch - 13ms/step

Epoch 2/20

422/422 - 5s - loss: 2.2758 - accuracy: 0.1753 - val_loss: 2.2371 - val_accuracy: 0.1152 - 5s/epoch - 11ms/step

```
Epoch 3/20
422/422 - 5s - loss: 1.9609 - accuracy: 0.4114 - val_loss: 1.4642 -
val_accuracy: 0.6343 - 5s/epoch - 11ms/step
Epoch 4/20
422/422 - 5s - loss: 1.0169 - accuracy: 0.7446 - val_loss: 0.6221 -
val_accuracy: 0.8578 - 5s/epoch - 11ms/step
Epoch 5/20
422/422 - 5s - loss: 0.5457 - accuracy: 0.8569 - val_loss: 0.3879 -
val_accuracy: 0.9023 - 5s/epoch - 11ms/step
Epoch 6/20
422/422 - 5s - loss: 0.3850 - accuracy: 0.8947 - val_loss: 0.2796 -
val_accuracy: 0.9280 - 5s/epoch - 11ms/step
Epoch 7/20
422/422 - 5s - loss: 0.2995 - accuracy: 0.9164 - val_loss: 0.2203 -
val_accuracy: 0.9433 - 5s/epoch - 11ms/step
Epoch 8/20
422/422 - 5s - loss: 0.2468 - accuracy: 0.9304 - val_loss: 0.1834 -
val_accuracy: 0.9523 - 5s/epoch - 11ms/step
Epoch 9/20
422/422 - 5s - loss: 0.2101 - accuracy: 0.9405 - val_loss: 0.1558 -
val_accuracy: 0.9597 - 5s/epoch - 11ms/step
Epoch 10/20
422/422 - 5s - loss: 0.1843 - accuracy: 0.9475 - val_loss: 0.1380 -
val_accuracy: 0.9637 - 5s/epoch - 11ms/step
Epoch 11/20
422/422 - 5s - loss: 0.1636 - accuracy: 0.9525 - val_loss: 0.1225 -
val_accuracy: 0.9683 - 5s/epoch - 11ms/step
Epoch 12/20
422/422 - 5s - loss: 0.1484 - accuracy: 0.9565 - val_loss: 0.1156 -
val_accuracy: 0.9697 - 5s/epoch - 11ms/step
Epoch 13/20
422/422 - 5s - loss: 0.1353 - accuracy: 0.9610 - val_loss: 0.1037 -
val_accuracy: 0.9737 - 5s/epoch - 11ms/step
Epoch 14/20
422/422 - 4s - loss: 0.1256 - accuracy: 0.9638 - val_loss: 0.1001 -
val_accuracy: 0.9718 - 4s/epoch - 11ms/step
Epoch 15/20
422/422 - 5s - loss: 0.1173 - accuracy: 0.9659 - val_loss: 0.0907 -
val_accuracy: 0.9765 - 5s/epoch - 11ms/step
Epoch 16/20
422/422 - 5s - loss: 0.1100 - accuracy: 0.9681 - val_loss: 0.0871 -
val_accuracy: 0.9787 - 5s/epoch - 11ms/step
Epoch 17/20
422/422 - 4s - loss: 0.1043 - accuracy: 0.9698 - val_loss: 0.0813 -
val_accuracy: 0.9788 - 4s/epoch - 10ms/step
Epoch 18/20
422/422 - 5s - loss: 0.0984 - accuracy: 0.9712 - val_loss: 0.0787 -
val_accuracy: 0.9800 - 5s/epoch - 11ms/step
Epoch 19/20
```

```
422/422 - 5s - loss: 0.0939 - accuracy: 0.9720 - val_loss: 0.0790 -  
val_accuracy: 0.9798 - 5s/epoch - 11ms/step
```

```
Epoch 20/20
```

```
422/422 - 4s - loss: 0.0897 - accuracy: 0.9733 - val_loss: 0.0736 -  
val_accuracy: 0.9813 - 4s/epoch - 11ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: sigmoid
```

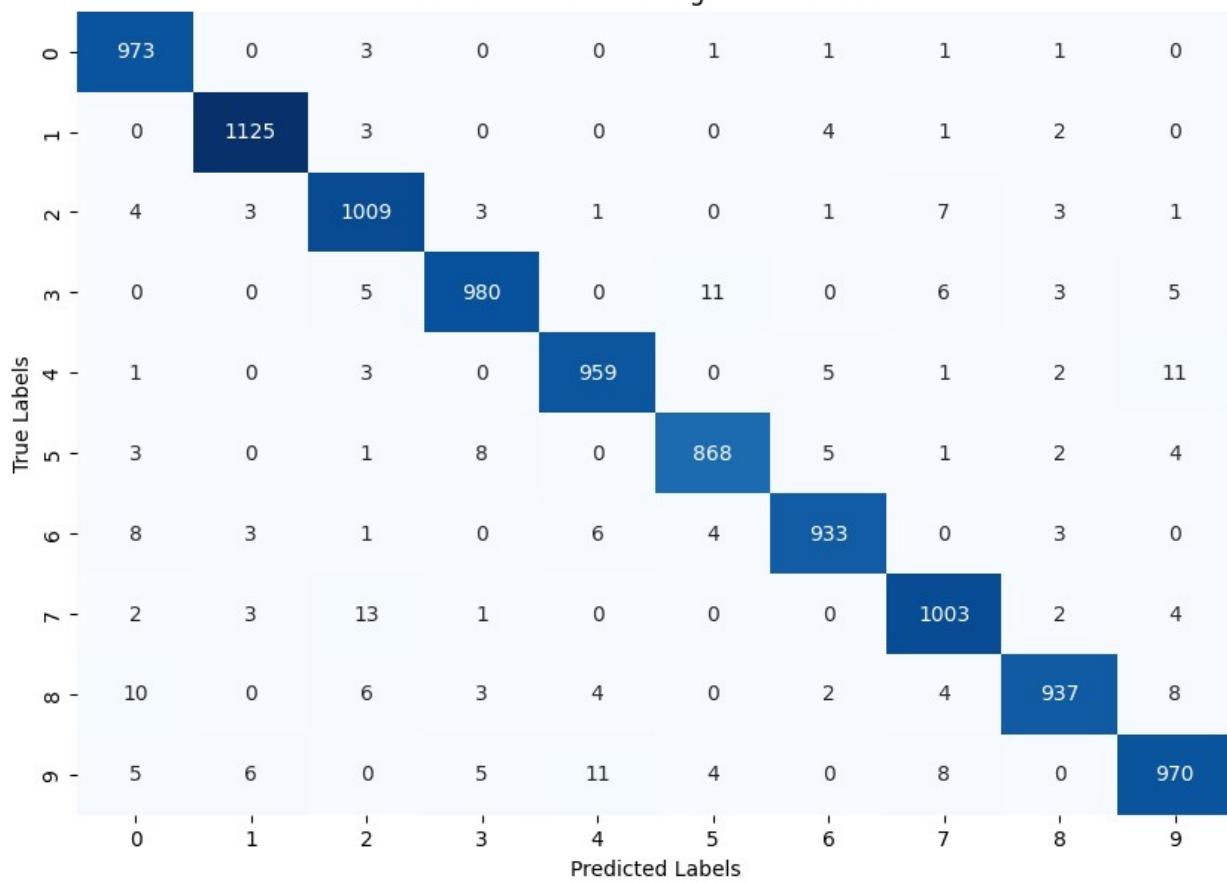
```
Confusion Matrix:
```

```
[[ 973   0   3   0   0   1   1   1   1   0]
 [ 0 1125   3   0   0   0   4   1   2   0]
 [ 4   3 1009   3   1   0   1   7   3   1]
 [ 0   0   5 980   0  11   0   6   3   5]
 [ 1   0   3   0 959   0   5   1   2  11]
 [ 3   0   1   8   0 868   5   1   2   4]
 [ 8   3   1   0   6   4 933   0   3   0]
 [ 2   3  13   1   0   0   0 1003   2   4]
 [ 10  0   6   3   4   0   2   4 937   8]
 [ 5   6   0   5  11   4   0   8   0 970]]
```

```
Precision: 0.9757
```

```
Recall: 0.9757
```

Confusion Matrix for sigmoid Activation



```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 4s - loss: 2.3074 - accuracy: 0.1140 - val_loss: 2.2990 -
val_accuracy: 0.0978 - 4s/epoch - 21ms/step
Epoch 2/5
211/211 - 4s - loss: 2.2951 - accuracy: 0.1166 - val_loss: 2.2909 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 3/5
211/211 - 4s - loss: 2.2840 - accuracy: 0.1343 - val_loss: 2.2726 -
val_accuracy: 0.1772 - 4s/epoch - 17ms/step
Epoch 4/5
211/211 - 4s - loss: 2.2495 - accuracy: 0.2331 - val_loss: 2.2089 -
val_accuracy: 0.2378 - 4s/epoch - 18ms/step
Epoch 5/5
211/211 - 4s - loss: 2.0789 - accuracy: 0.4120 - val_loss: 1.8477 -
val_accuracy: 0.5380 - 4s/epoch - 17ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 936    5   13    5    1    0   10    7    2    1]
 [  0 1129    5    0    0    0    1    0    0    0]
 [ 61  215  675    2    3    0   38   30    0    8]
 [ 62  312  239  323    0    0   10   43    6   15]
 [ 30   63    5    0  408    0   31  144    1  300]
 [ 239  203    67  102   13    0   49  124   28   67]
 [ 184   90   56    1   19    0  586   11    2    9]
 [  4  248   27    0   79    0    2  625    1   42]
 [ 85  243  251   36    1    0   51   33  163  111]
 [ 48  128    4    2  149    0    4  187    1  486]]
```

Precision: 0.5351
Recall: 0.5331

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	936	5	13	5	1	0	10	7	2	1
0	936	5	13	5	1	0	10	7	2	1
1	0	1129	5	0	0	0	1	0	0	0
2	61	215	675	2	3	0	38	30	0	8
3	62	312	239	323	0	0	10	43	6	15
4	30	63	5	0	408	0	31	144	1	300
5	239	203	67	102	13	0	49	124	28	67
6	184	90	56	1	19	0	586	11	2	9
7	4	248	27	0	79	0	2	625	1	42
8	85	243	251	36	1	0	51	33	163	111
9	48	128	4	2	149	0	4	187	1	486
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 4s - loss: 2.3052 - accuracy: 0.1129 - val_loss: 2.2990 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 2/15
211/211 - 4s - loss: 2.2927 - accuracy: 0.1414 - val_loss: 2.2846 -
val_accuracy: 0.1113 - 4s/epoch - 17ms/step
Epoch 3/15
211/211 - 4s - loss: 2.2742 - accuracy: 0.1863 - val_loss: 2.2545 -
val_accuracy: 0.3572 - 4s/epoch - 17ms/step
Epoch 4/15
211/211 - 4s - loss: 2.2022 - accuracy: 0.3065 - val_loss: 2.1024 -
val_accuracy: 0.4718 - 4s/epoch - 17ms/step
Epoch 5/15
211/211 - 4s - loss: 1.8649 - accuracy: 0.5045 - val_loss: 1.5363 -
val_accuracy: 0.6568 - 4s/epoch - 17ms/step
Epoch 6/15
211/211 - 4s - loss: 1.2528 - accuracy: 0.6651 - val_loss: 0.9537 -
val_accuracy: 0.7588 - 4s/epoch - 17ms/step
Epoch 7/15
```

```
211/211 - 4s - loss: 0.8783 - accuracy: 0.7542 - val_loss: 0.7104 -  
val_accuracy: 0.8272 - 4s/epoch - 17ms/step  
Epoch 8/15  
211/211 - 4s - loss: 0.7000 - accuracy: 0.8086 - val_loss: 0.5674 -  
val_accuracy: 0.8570 - 4s/epoch - 17ms/step  
Epoch 9/15  
211/211 - 4s - loss: 0.5828 - accuracy: 0.8413 - val_loss: 0.4691 -  
val_accuracy: 0.8803 - 4s/epoch - 17ms/step  
Epoch 10/15  
211/211 - 4s - loss: 0.4976 - accuracy: 0.8637 - val_loss: 0.3979 -  
val_accuracy: 0.8975 - 4s/epoch - 17ms/step  
Epoch 11/15  
211/211 - 4s - loss: 0.4323 - accuracy: 0.8810 - val_loss: 0.3425 -  
val_accuracy: 0.9117 - 4s/epoch - 17ms/step  
Epoch 12/15  
211/211 - 4s - loss: 0.3800 - accuracy: 0.8954 - val_loss: 0.3012 -  
val_accuracy: 0.9212 - 4s/epoch - 17ms/step  
Epoch 13/15  
211/211 - 4s - loss: 0.3388 - accuracy: 0.9064 - val_loss: 0.2668 -  
val_accuracy: 0.9282 - 4s/epoch - 17ms/step  
Epoch 14/15  
211/211 - 4s - loss: 0.3045 - accuracy: 0.9149 - val_loss: 0.2384 -  
val_accuracy: 0.9358 - 4s/epoch - 17ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.2760 - accuracy: 0.9227 - val_loss: 0.2157 -  
val_accuracy: 0.9433 - 3s/epoch - 16ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 965  0  1  0  0  3  3  1  7  0]  
[  0 1111  3  2  0  2  6  0  11  0]  
[ 12  1 947  9  7  1  15  15  21  4]  
[  1  2  21 931  0  20  0  13  16  6]  
[  1  3  5  0 906  0  17  0  2  48]  
[ 21  2  6  26  2 792  15  4  17  7]  
[ 13  3  4  0  10  14 912  0  2  0]  
[  2  10  24  4  5  0  0 945  2  36]  
[  6  6  8  13  11  15  13  7 877  18]  
[ 10  6  3  10  24  3  0  20  9 924]]  
Precision: 0.9311  
Recall: 0.9310
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	965	0	1	0	0	3	3	1	7	0	
1	0	1111	3	2	0	2	6	0	11	0	
2	12	1	947	9	7	1	15	15	21	4	
3	1	2	21	931	0	20	0	13	16	6	
4	1	3	5	0	906	0	17	0	2	48	
5	21	2	6	26	2	792	15	4	17	7	
6	13	3	4	0	10	14	912	0	2	0	
7	2	10	24	4	5	0	0	945	2	36	
8	6	6	8	13	11	15	13	7	877	18	
9	10	6	3	10	24	3	0	20	9	924	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels											

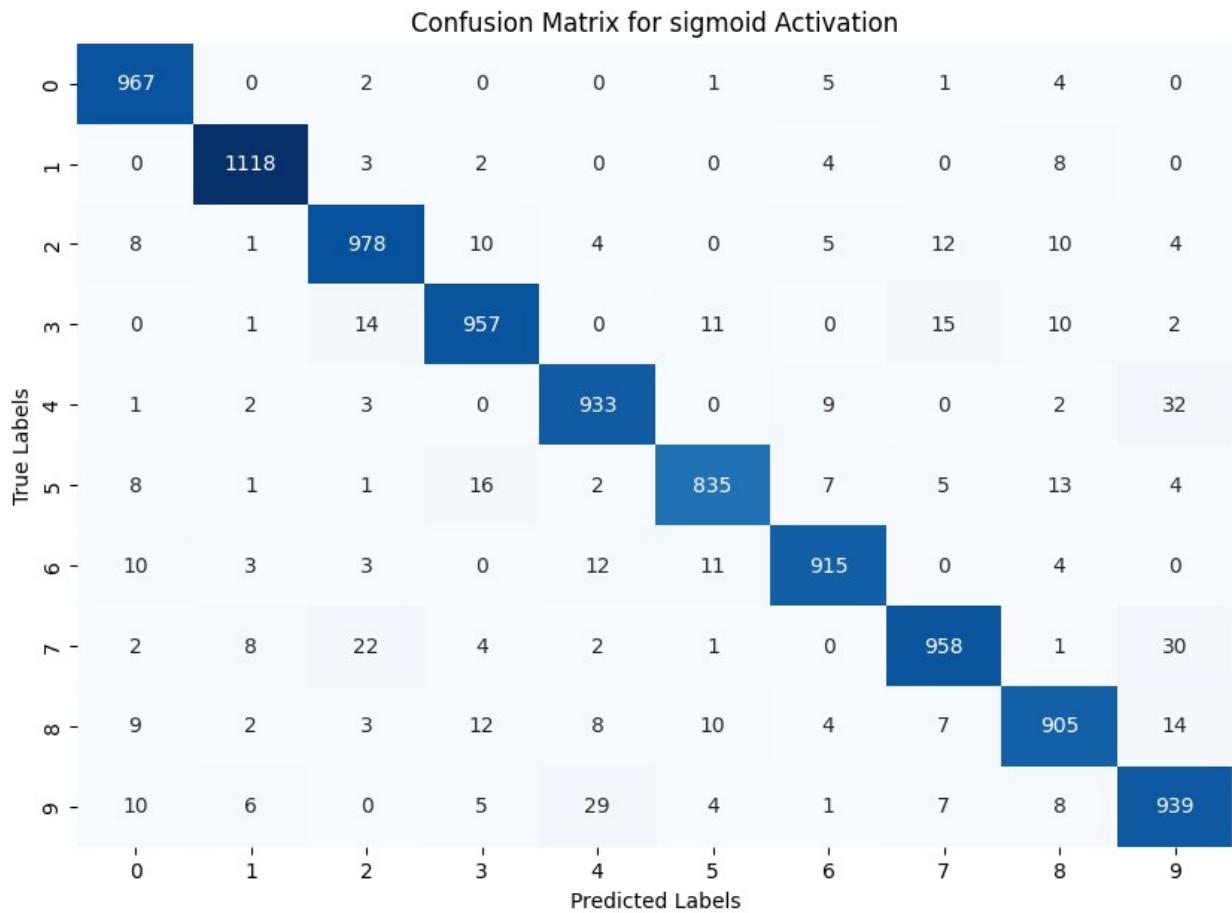
```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 4s - loss: 2.3107 - accuracy: 0.1116 - val_loss: 2.3003 -
val_accuracy: 0.0978 - 4s/epoch - 21ms/step
Epoch 2/20
211/211 - 4s - loss: 2.2985 - accuracy: 0.1198 - val_loss: 2.2948 -
val_accuracy: 0.1083 - 4s/epoch - 17ms/step
Epoch 3/20
211/211 - 4s - loss: 2.2915 - accuracy: 0.1357 - val_loss: 2.2854 -
val_accuracy: 0.2055 - 4s/epoch - 17ms/step
Epoch 4/20
211/211 - 4s - loss: 2.2747 - accuracy: 0.1761 - val_loss: 2.2570 -
val_accuracy: 0.3162 - 4s/epoch - 17ms/step
Epoch 5/20
211/211 - 4s - loss: 2.2034 - accuracy: 0.2783 - val_loss: 2.1035 -
val_accuracy: 0.3360 - 4s/epoch - 17ms/step
Epoch 6/20
211/211 - 4s - loss: 1.8966 - accuracy: 0.4470 - val_loss: 1.6397 -
val_accuracy: 0.5612 - 4s/epoch - 17ms/step
Epoch 7/20
```

```
211/211 - 4s - loss: 1.3948 - accuracy: 0.6166 - val_loss: 1.0979 -  
val_accuracy: 0.7165 - 4s/epoch - 17ms/step  
Epoch 8/20  
211/211 - 4s - loss: 0.9606 - accuracy: 0.7357 - val_loss: 0.7550 -  
val_accuracy: 0.8135 - 4s/epoch - 18ms/step  
Epoch 9/20  
211/211 - 4s - loss: 0.7183 - accuracy: 0.8036 - val_loss: 0.5709 -  
val_accuracy: 0.8637 - 4s/epoch - 17ms/step  
Epoch 10/20  
211/211 - 4s - loss: 0.5781 - accuracy: 0.8413 - val_loss: 0.4582 -  
val_accuracy: 0.8858 - 4s/epoch - 17ms/step  
Epoch 11/20  
211/211 - 4s - loss: 0.4864 - accuracy: 0.8649 - val_loss: 0.3869 -  
val_accuracy: 0.9030 - 4s/epoch - 17ms/step  
Epoch 12/20  
211/211 - 4s - loss: 0.4220 - accuracy: 0.8831 - val_loss: 0.3336 -  
val_accuracy: 0.9155 - 4s/epoch - 17ms/step  
Epoch 13/20  
211/211 - 3s - loss: 0.3733 - accuracy: 0.8961 - val_loss: 0.2928 -  
val_accuracy: 0.9252 - 3s/epoch - 17ms/step  
Epoch 14/20  
211/211 - 4s - loss: 0.3341 - accuracy: 0.9069 - val_loss: 0.2608 -  
val_accuracy: 0.9320 - 4s/epoch - 17ms/step  
Epoch 15/20  
211/211 - 3s - loss: 0.3019 - accuracy: 0.9157 - val_loss: 0.2354 -  
val_accuracy: 0.9395 - 3s/epoch - 17ms/step  
Epoch 16/20  
211/211 - 4s - loss: 0.2746 - accuracy: 0.9233 - val_loss: 0.2128 -  
val_accuracy: 0.9468 - 4s/epoch - 17ms/step  
Epoch 17/20  
211/211 - 4s - loss: 0.2514 - accuracy: 0.9293 - val_loss: 0.1933 -  
val_accuracy: 0.9520 - 4s/epoch - 17ms/step  
Epoch 18/20  
211/211 - 4s - loss: 0.2318 - accuracy: 0.9346 - val_loss: 0.1780 -  
val_accuracy: 0.9560 - 4s/epoch - 17ms/step  
Epoch 19/20  
211/211 - 4s - loss: 0.2151 - accuracy: 0.9390 - val_loss: 0.1647 -  
val_accuracy: 0.9598 - 4s/epoch - 17ms/step  
Epoch 20/20  
211/211 - 4s - loss: 0.1998 - accuracy: 0.9436 - val_loss: 0.1544 -  
val_accuracy: 0.9612 - 4s/epoch - 17ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 967 0 2 0 0 1 5 1 4 0]  
 [ 0 1118 3 2 0 0 4 0 8 0]  
 [ 8 1 978 10 4 0 5 12 10 4]  
 [ 0 1 14 957 0 11 0 15 10 2]  
 [ 1 2 3 0 933 0 9 0 2 32]]
```

```
[ 8  1  1 16  2 835  7  5 13  4]
[10  3  3  0 12 11 915  0  4  0]
[ 2  8 22  4  2  1  0 958  1 30]
[ 9  2  3 12  8 10  4  7 905 14]
[10  6  0  5 29  4  1  7  8 939]]
```

Precision: 0.9506

Recall: 0.9505



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 7s - loss: 2.3016 - accuracy: 0.1217 - val_loss: 2.2527 - val_accuracy: 0.3840 - 7s/epoch - 9ms/step

Epoch 2/5

844/844 - 7s - loss: 1.4845 - accuracy: 0.5466 - val_loss: 0.6041 - val_accuracy: 0.8382 - 7s/epoch - 8ms/step

Epoch 3/5

844/844 - 6s - loss: 0.4768 - accuracy: 0.8624 - val_loss: 0.2934 - val_accuracy: 0.9193 - 6s/epoch - 8ms/step

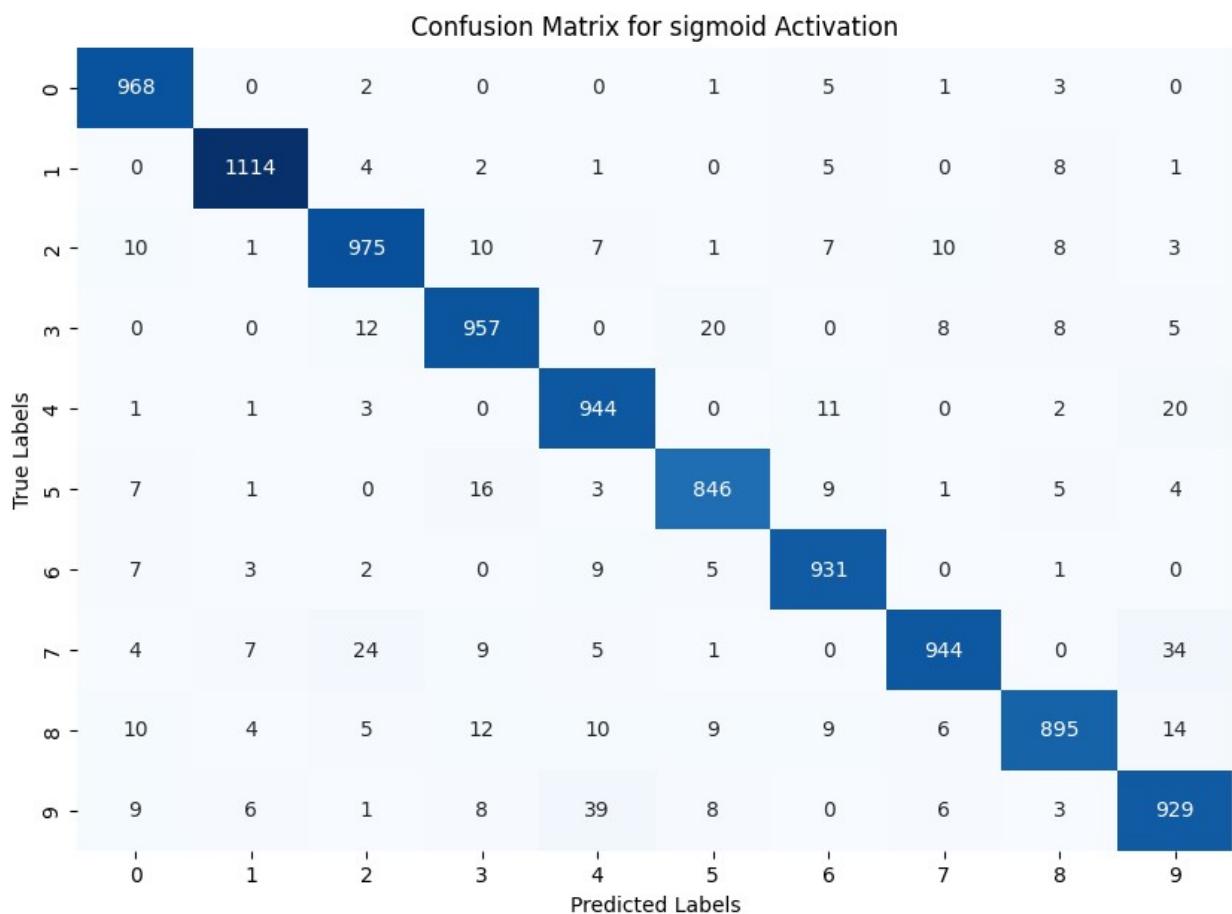
Epoch 4/5

844/844 - 6s - loss: 0.2952 - accuracy: 0.9143 - val_loss: 0.1994 -

```

val_accuracy: 0.9440 - 6s/epoch - 7ms/step
Epoch 5/5
844/844 - 7s - loss: 0.2122 - accuracy: 0.9382 - val_loss: 0.1430 -
val_accuracy: 0.9598 - 7s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 968   0   2   0   0   1   5   1   3   0]
 [  0 1114   4   2   1   0   5   0   8   1]
 [ 10   1 975  10   7   1   7  10   8   3]
 [  0   0 12 957   0  20   0   8   8   5]
 [  1   1   3   0 944   0  11   0   2  20]
 [  7   1   0  16   3 846   9   1   5   4]
 [  7   3   2   0   9   5 931   0   1   0]
 [  4   7  24   9   5   1   0 944   0  34]
 [ 10   4   5  12  10   9   9   6 895  14]
 [  9   6   1   8  39   8   0   6   3 929]]
Precision: 0.9505
Recall: 0.9503

```



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
64 batch size, 15 epochs..

Epoch 1/15
844/844 - 7s - loss: 2.3012 - accuracy: 0.1216 - val_loss: 2.2667 -
val_accuracy: 0.1000 - 7s/epoch - 9ms/step

Epoch 2/15
844/844 - 7s - loss: 1.4959 - accuracy: 0.5515 - val_loss: 0.5588 -
val_accuracy: 0.8585 - 7s/epoch - 8ms/step

Epoch 3/15
844/844 - 6s - loss: 0.4470 - accuracy: 0.8730 - val_loss: 0.2815 -
val_accuracy: 0.9192 - 6s/epoch - 8ms/step

Epoch 4/15
844/844 - 7s - loss: 0.2744 - accuracy: 0.9197 - val_loss: 0.1849 -
val_accuracy: 0.9460 - 7s/epoch - 8ms/step

Epoch 5/15
844/844 - 7s - loss: 0.1997 - accuracy: 0.9418 - val_loss: 0.1420 -
val_accuracy: 0.9573 - 7s/epoch - 8ms/step

Epoch 6/15
844/844 - 7s - loss: 0.1597 - accuracy: 0.9531 - val_loss: 0.1161 -
val_accuracy: 0.9662 - 7s/epoch - 8ms/step

Epoch 7/15
844/844 - 7s - loss: 0.1348 - accuracy: 0.9595 - val_loss: 0.1036 -
val_accuracy: 0.9700 - 7s/epoch - 8ms/step

Epoch 8/15
844/844 - 7s - loss: 0.1173 - accuracy: 0.9652 - val_loss: 0.0948 -
val_accuracy: 0.9718 - 7s/epoch - 8ms/step

Epoch 9/15
844/844 - 7s - loss: 0.1069 - accuracy: 0.9679 - val_loss: 0.0814 -
val_accuracy: 0.9775 - 7s/epoch - 8ms/step

Epoch 10/15
844/844 - 6s - loss: 0.0962 - accuracy: 0.9714 - val_loss: 0.0752 -
val_accuracy: 0.9790 - 6s/epoch - 8ms/step

Epoch 11/15
844/844 - 6s - loss: 0.0907 - accuracy: 0.9730 - val_loss: 0.0750 -
val_accuracy: 0.9780 - 6s/epoch - 8ms/step

Epoch 12/15
844/844 - 7s - loss: 0.0836 - accuracy: 0.9747 - val_loss: 0.0689 -
val_accuracy: 0.9802 - 7s/epoch - 8ms/step

Epoch 13/15
844/844 - 7s - loss: 0.0796 - accuracy: 0.9759 - val_loss: 0.0814 -
val_accuracy: 0.9763 - 7s/epoch - 8ms/step

Epoch 14/15
844/844 - 7s - loss: 0.0749 - accuracy: 0.9771 - val_loss: 0.0631 -
val_accuracy: 0.9825 - 7s/epoch - 8ms/step

Epoch 15/15
844/844 - 7s - loss: 0.0699 - accuracy: 0.9791 - val_loss: 0.0627 -
val_accuracy: 0.9832 - 7s/epoch - 8ms/step

313/313 [=====] - 1s 3ms/step

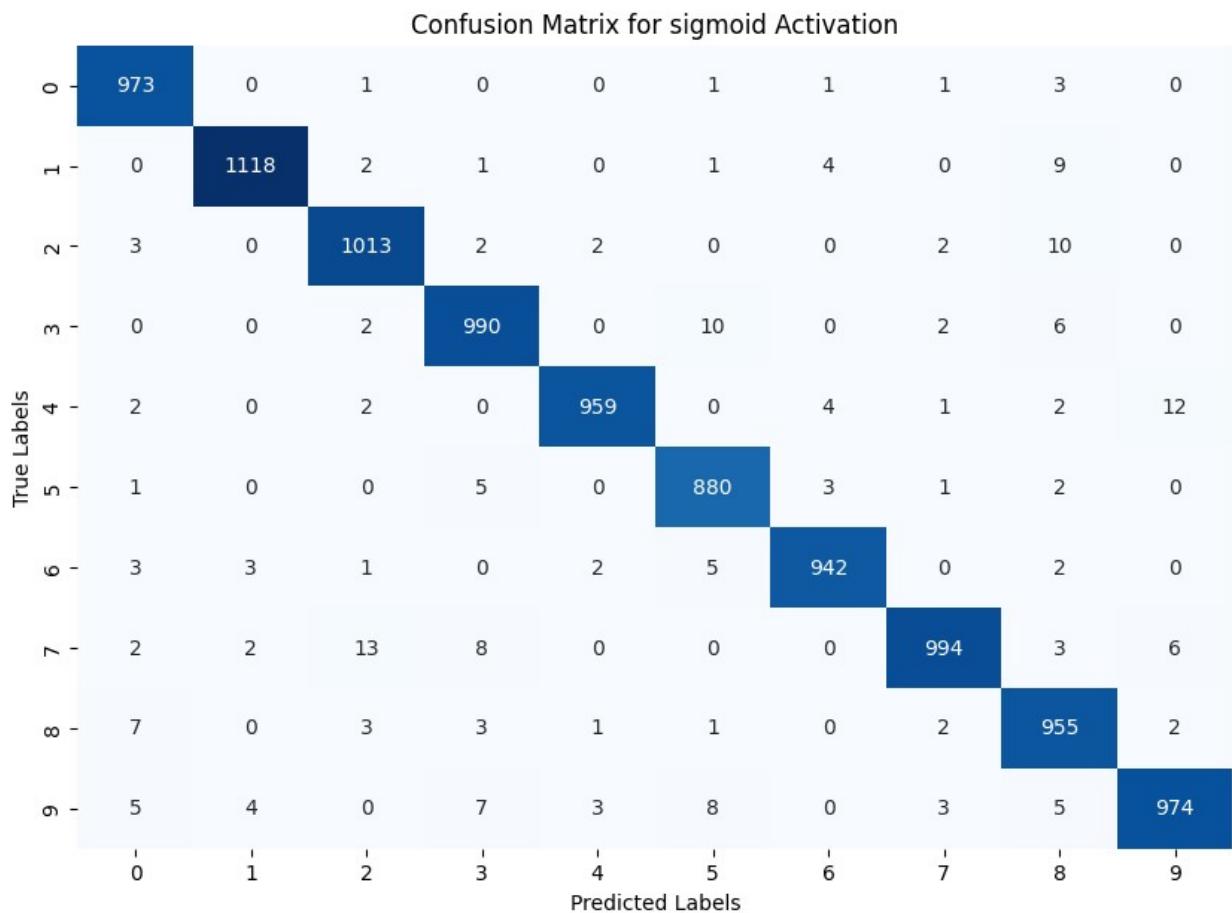
Results for activation function: sigmoid

Confusion Matrix:

```
[[ 973  0   1   0   0   1   1   1   3   0]
 [ 0 1118  2   1   0   1   4   0   9   0]
 [ 3   0 1013  2   2   0   0   2   10  0]
 [ 0   0   2 990  0   10  0   2   6   0]
 [ 2   0   2   0 959  0   4   1   2   12]
 [ 1   0   0   5   0 880  3   1   2   0]
 [ 3   3   1   0   2   5 942  0   2   0]
 [ 2   2   13  8   0   0   0 994  3   6]
 [ 7   0   3   3   1   1   0   2 955  2]
 [ 5   4   0   7   3   8   0   3   5 974]]
```

Precision: 0.9799

Recall: 0.9798



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 64 batch size, 20 epochs..

Epoch 1/20

844/844 - 7s - loss: 2.2992 - accuracy: 0.1274 - val_loss: 2.2414 - val_accuracy: 0.3593 - 7s/epoch - 9ms/step

Epoch 2/20

844/844 - 6s - loss: 1.3306 - accuracy: 0.6135 - val_loss: 0.5314 - val_accuracy: 0.8712 - 6s/epoch - 8ms/step

```
Epoch 3/20
844/844 - 6s - loss: 0.4358 - accuracy: 0.8773 - val_loss: 0.2769 -
val_accuracy: 0.9228 - 6s/epoch - 8ms/step
Epoch 4/20
844/844 - 6s - loss: 0.2813 - accuracy: 0.9188 - val_loss: 0.1901 -
val_accuracy: 0.9475 - 6s/epoch - 8ms/step
Epoch 5/20
844/844 - 7s - loss: 0.2106 - accuracy: 0.9379 - val_loss: 0.1497 -
val_accuracy: 0.9602 - 7s/epoch - 8ms/step
Epoch 6/20
844/844 - 7s - loss: 0.1683 - accuracy: 0.9499 - val_loss: 0.1303 -
val_accuracy: 0.9625 - 7s/epoch - 8ms/step
Epoch 7/20
844/844 - 6s - loss: 0.1418 - accuracy: 0.9575 - val_loss: 0.1015 -
val_accuracy: 0.9720 - 6s/epoch - 7ms/step
Epoch 8/20
844/844 - 6s - loss: 0.1224 - accuracy: 0.9632 - val_loss: 0.0960 -
val_accuracy: 0.9725 - 6s/epoch - 8ms/step
Epoch 9/20
844/844 - 7s - loss: 0.1097 - accuracy: 0.9674 - val_loss: 0.0844 -
val_accuracy: 0.9763 - 7s/epoch - 8ms/step
Epoch 10/20
844/844 - 6s - loss: 0.0981 - accuracy: 0.9709 - val_loss: 0.0795 -
val_accuracy: 0.9772 - 6s/epoch - 8ms/step
Epoch 11/20
844/844 - 7s - loss: 0.0903 - accuracy: 0.9732 - val_loss: 0.0764 -
val_accuracy: 0.9783 - 7s/epoch - 8ms/step
Epoch 12/20
844/844 - 7s - loss: 0.0839 - accuracy: 0.9751 - val_loss: 0.0699 -
val_accuracy: 0.9787 - 7s/epoch - 8ms/step
Epoch 13/20
844/844 - 7s - loss: 0.0783 - accuracy: 0.9769 - val_loss: 0.0665 -
val_accuracy: 0.9805 - 7s/epoch - 8ms/step
Epoch 14/20
844/844 - 7s - loss: 0.0735 - accuracy: 0.9782 - val_loss: 0.0678 -
val_accuracy: 0.9802 - 7s/epoch - 8ms/step
Epoch 15/20
844/844 - 7s - loss: 0.0698 - accuracy: 0.9791 - val_loss: 0.0624 -
val_accuracy: 0.9818 - 7s/epoch - 8ms/step
Epoch 16/20
844/844 - 7s - loss: 0.0658 - accuracy: 0.9805 - val_loss: 0.0680 -
val_accuracy: 0.9812 - 7s/epoch - 8ms/step
Epoch 17/20
844/844 - 7s - loss: 0.0624 - accuracy: 0.9816 - val_loss: 0.0617 -
val_accuracy: 0.9808 - 7s/epoch - 8ms/step
Epoch 18/20
844/844 - 7s - loss: 0.0600 - accuracy: 0.9815 - val_loss: 0.0587 -
val_accuracy: 0.9838 - 7s/epoch - 8ms/step
Epoch 19/20
```

```
844/844 - 7s - loss: 0.0572 - accuracy: 0.9830 - val_loss: 0.0559 -  
val_accuracy: 0.9835 - 7s/epoch - 8ms/step
```

```
Epoch 20/20
```

```
844/844 - 6s - loss: 0.0549 - accuracy: 0.9832 - val_loss: 0.0574 -  
val_accuracy: 0.9837 - 6s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

```
Results for activation function: sigmoid
```

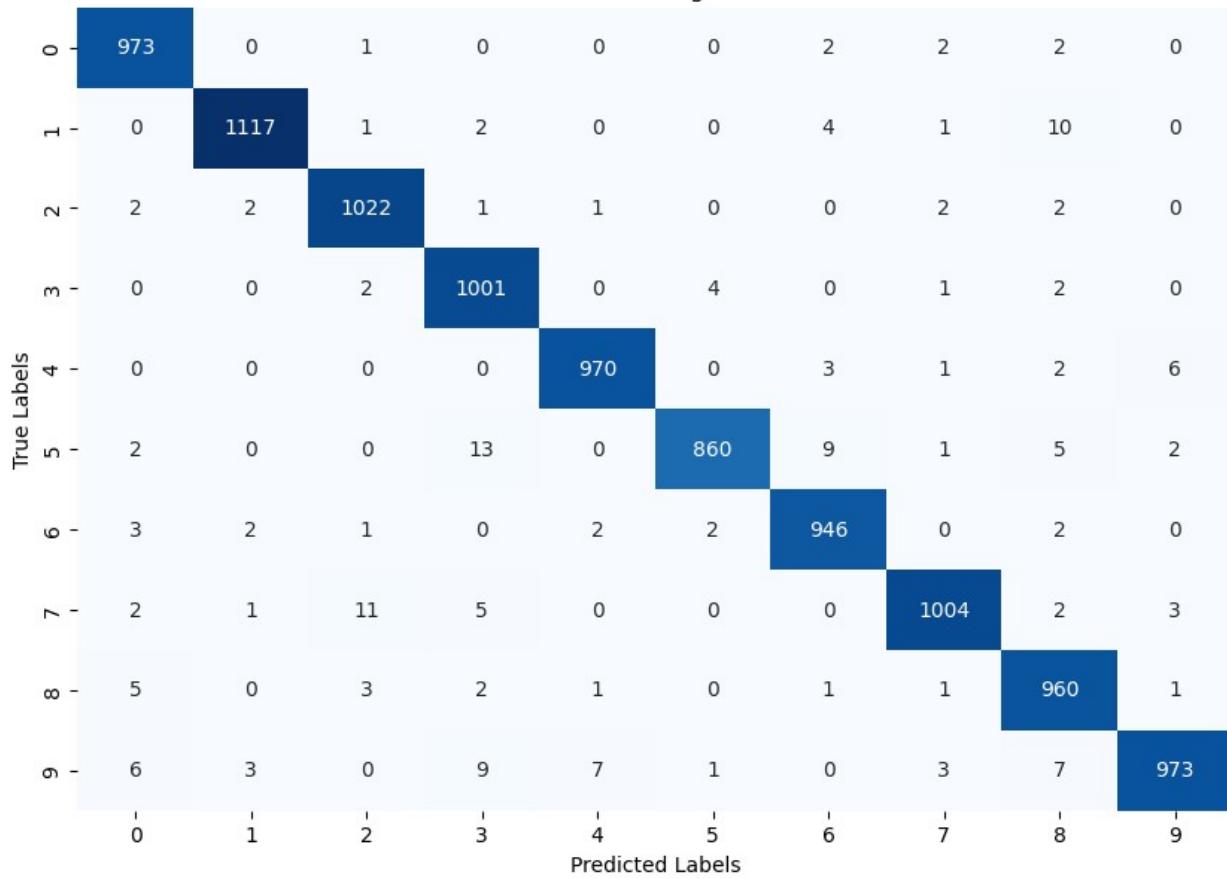
```
Confusion Matrix:
```

```
[[ 973   0   1   0   0   0   2   2   2   0]
 [ 0 1117   1   2   0   0   4   1  10   0]
 [ 2   2 1022   1   1   0   0   2   2   0]
 [ 0   0   2 1001   0   4   0   1   2   0]
 [ 0   0   0   0  970   0   3   1   2   6]
 [ 2   0   0   13   0  860   9   1   5   2]
 [ 3   2   1   0   2   2  946   0   2   0]
 [ 2   1   11   5   0   0   0 1004   2   3]
 [ 5   0   3   2   1   0   1   1  960   1]
 [ 6   3   0   9   7   1   0   3   7  973]]
```

```
Precision: 0.9827
```

```
Recall: 0.9826
```

Confusion Matrix for sigmoid Activation



```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 5s - loss: 2.3078 - accuracy: 0.1095 - val_loss: 2.2915 -
val_accuracy: 0.1113 - 5s/epoch - 13ms/step
Epoch 2/5
422/422 - 5s - loss: 2.2675 - accuracy: 0.1742 - val_loss: 2.1917 -
val_accuracy: 0.3662 - 5s/epoch - 11ms/step
Epoch 3/5
422/422 - 5s - loss: 1.6712 - accuracy: 0.5526 - val_loss: 0.9436 -
val_accuracy: 0.7732 - 5s/epoch - 12ms/step
Epoch 4/5
422/422 - 5s - loss: 0.7150 - accuracy: 0.8071 - val_loss: 0.4767 -
val_accuracy: 0.8765 - 5s/epoch - 11ms/step
Epoch 5/5
422/422 - 5s - loss: 0.4717 - accuracy: 0.8678 - val_loss: 0.3466 -
val_accuracy: 0.9107 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 956     0     3     2     0     8     3     1     7     0]
 [  0 1087     2     7     0     2     3     1    33     0]
 [ 14     3  904     8    17     3    24    16    37     6]
 [  2     5    26  899     1    24     1    18    30     4]
 [  1     2     3     0  860     2    28     0     8    78]
 [ 13     4     8    68     9   721    37     5    19     8]
 [ 14     3    23     1   14    22   875     0     6     0]
 [  4    15    30     4     6     1     0  914     6    48]
 [  8     3     7    32    11    27    18     3  852    13]
 [ 13     5     8     8    50    13     0    33    18  861]]
```

Precision: 0.8931
Recall: 0.8929

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	956	0	3	2	0	8	3	1	7	0	
1	0	1087	2	7	0	2	3	1	33	0	
2	14	3	904	8	17	3	24	16	37	6	
3	2	5	26	899	1	24	1	18	30	4	
4	1	2	3	0	860	2	28	0	8	78	
5	13	4	8	68	9	721	37	5	19	8	
6	14	3	23	1	14	22	875	0	6	0	
7	4	15	30	4	6	1	0	914	6	48	
8	8	3	7	32	11	27	18	3	852	13	
9	13	5	8	8	50	13	0	33	18	861	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 6s - loss: 2.3054 - accuracy: 0.1059 - val_loss: 2.3001 -
val_accuracy: 0.1858 - 6s/epoch - 14ms/step
Epoch 2/15
422/422 - 5s - loss: 2.2763 - accuracy: 0.1618 - val_loss: 2.2302 -
val_accuracy: 0.0958 - 5s/epoch - 11ms/step
Epoch 3/15
422/422 - 5s - loss: 1.8126 - accuracy: 0.4686 - val_loss: 1.1445 -
val_accuracy: 0.7072 - 5s/epoch - 11ms/step
Epoch 4/15
422/422 - 5s - loss: 0.8185 - accuracy: 0.7757 - val_loss: 0.5315 -
val_accuracy: 0.8625 - 5s/epoch - 11ms/step
Epoch 5/15
422/422 - 5s - loss: 0.4879 - accuracy: 0.8620 - val_loss: 0.3529 -
val_accuracy: 0.9072 - 5s/epoch - 11ms/step
Epoch 6/15
422/422 - 5s - loss: 0.3646 - accuracy: 0.8955 - val_loss: 0.2680 -
val_accuracy: 0.9268 - 5s/epoch - 11ms/step
Epoch 7/15
```

```
422/422 - 5s - loss: 0.2957 - accuracy: 0.9148 - val_loss: 0.2192 -  
val_accuracy: 0.9405 - 5s/epoch - 11ms/step  
Epoch 8/15  
422/422 - 5s - loss: 0.2483 - accuracy: 0.9272 - val_loss: 0.1890 -  
val_accuracy: 0.9488 - 5s/epoch - 11ms/step  
Epoch 9/15  
422/422 - 5s - loss: 0.2141 - accuracy: 0.9373 - val_loss: 0.1610 -  
val_accuracy: 0.9560 - 5s/epoch - 11ms/step  
Epoch 10/15  
422/422 - 5s - loss: 0.1886 - accuracy: 0.9446 - val_loss: 0.1457 -  
val_accuracy: 0.9612 - 5s/epoch - 11ms/step  
Epoch 11/15  
422/422 - 5s - loss: 0.1677 - accuracy: 0.9514 - val_loss: 0.1261 -  
val_accuracy: 0.9678 - 5s/epoch - 11ms/step  
Epoch 12/15  
422/422 - 5s - loss: 0.1522 - accuracy: 0.9555 - val_loss: 0.1160 -  
val_accuracy: 0.9697 - 5s/epoch - 11ms/step  
Epoch 13/15  
422/422 - 5s - loss: 0.1395 - accuracy: 0.9589 - val_loss: 0.1091 -  
val_accuracy: 0.9715 - 5s/epoch - 11ms/step  
Epoch 14/15  
422/422 - 5s - loss: 0.1291 - accuracy: 0.9619 - val_loss: 0.1006 -  
val_accuracy: 0.9728 - 5s/epoch - 11ms/step  
Epoch 15/15  
422/422 - 5s - loss: 0.1209 - accuracy: 0.9637 - val_loss: 0.0942 -  
val_accuracy: 0.9740 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 970  0  2  0  0  2  3  1  2  0]  
[ 0 1110  2  3  0  0  5  0  15  0]  
[ 9  0 993  5  2  0  2  9  12  0]  
[ 1  0  2 981  0  9  0  8  7  2]  
[ 1  0  2  1 943  0  11  1  2  21]  
[ 5  0  1  8  0 865  6  1  4  2]  
[ 7  3  1  0  3  4 937  0  3  0]  
[ 2  5  19  8  0  1  0 974  2  17]  
[ 7  0  4  5  3  0  5  3 942  5]  
[ 9  7  0 10  12  5  0  5  4 957]]  
Precision: 0.9673  
Recall: 0.9672
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	970	0	2	0	0	2	3	1	2	0	
0	970	0	2	0	0	2	3	1	2	0	
1	0	1110	2	3	0	0	5	0	15	0	
2	9	0	993	5	2	0	2	9	12	0	
3	1	0	2	981	0	9	0	8	7	2	
4	1	0	2	1	943	0	11	1	2	21	
5	5	0	1	8	0	865	6	1	4	2	
6	7	3	1	0	3	4	937	0	3	0	
7	2	5	19	8	0	1	0	974	2	17	
8	7	0	4	5	3	0	5	3	942	5	
9	9	7	0	10	12	5	0	5	4	957	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 5s - loss: 2.3077 - accuracy: 0.1087 - val_loss: 2.2967 -
val_accuracy: 0.1417 - 5s/epoch - 13ms/step
Epoch 2/20
422/422 - 5s - loss: 2.2831 - accuracy: 0.1512 - val_loss: 2.2485 -
val_accuracy: 0.2708 - 5s/epoch - 11ms/step
Epoch 3/20
422/422 - 5s - loss: 1.9539 - accuracy: 0.4442 - val_loss: 1.2693 -
val_accuracy: 0.6985 - 5s/epoch - 11ms/step
Epoch 4/20
422/422 - 5s - loss: 0.8625 - accuracy: 0.7754 - val_loss: 0.5328 -
val_accuracy: 0.8687 - 5s/epoch - 11ms/step
Epoch 5/20
422/422 - 5s - loss: 0.4959 - accuracy: 0.8638 - val_loss: 0.3557 -
val_accuracy: 0.9067 - 5s/epoch - 11ms/step
Epoch 6/20
422/422 - 5s - loss: 0.3713 - accuracy: 0.8947 - val_loss: 0.2718 -
val_accuracy: 0.9272 - 5s/epoch - 11ms/step
Epoch 7/20
```

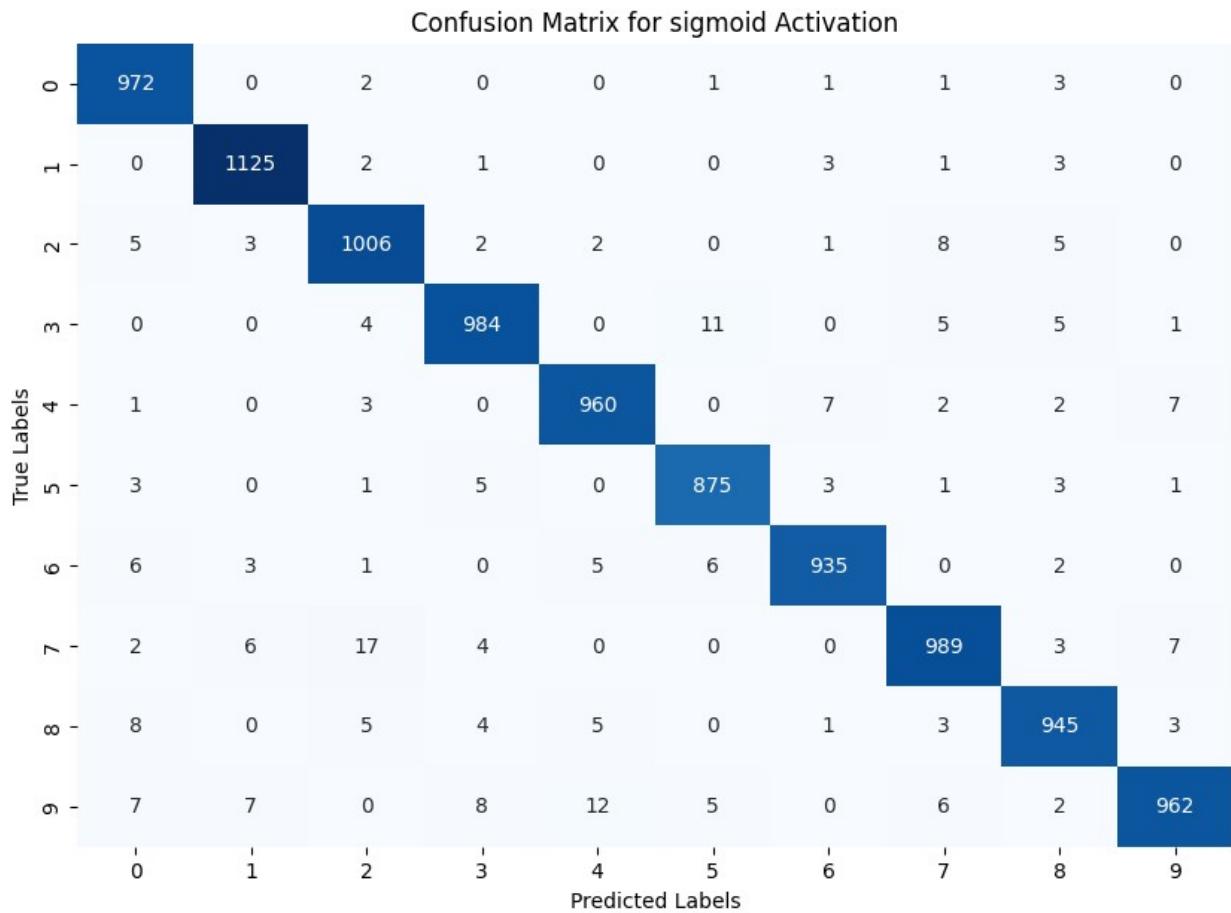
```

422/422 - 5s - loss: 0.3009 - accuracy: 0.9133 - val_loss: 0.2236 -
val_accuracy: 0.9382 - 5s/epoch - 11ms/step
Epoch 8/20
422/422 - 5s - loss: 0.2511 - accuracy: 0.9279 - val_loss: 0.1873 -
val_accuracy: 0.9487 - 5s/epoch - 11ms/step
Epoch 9/20
422/422 - 5s - loss: 0.2150 - accuracy: 0.9375 - val_loss: 0.1645 -
val_accuracy: 0.9557 - 5s/epoch - 11ms/step
Epoch 10/20
422/422 - 5s - loss: 0.1886 - accuracy: 0.9446 - val_loss: 0.1428 -
val_accuracy: 0.9610 - 5s/epoch - 11ms/step
Epoch 11/20
422/422 - 5s - loss: 0.1680 - accuracy: 0.9504 - val_loss: 0.1256 -
val_accuracy: 0.9662 - 5s/epoch - 11ms/step
Epoch 12/20
422/422 - 5s - loss: 0.1516 - accuracy: 0.9547 - val_loss: 0.1164 -
val_accuracy: 0.9680 - 5s/epoch - 11ms/step
Epoch 13/20
422/422 - 5s - loss: 0.1387 - accuracy: 0.9589 - val_loss: 0.1074 -
val_accuracy: 0.9703 - 5s/epoch - 11ms/step
Epoch 14/20
422/422 - 5s - loss: 0.1280 - accuracy: 0.9621 - val_loss: 0.0981 -
val_accuracy: 0.9738 - 5s/epoch - 11ms/step
Epoch 15/20
422/422 - 5s - loss: 0.1188 - accuracy: 0.9646 - val_loss: 0.0929 -
val_accuracy: 0.9755 - 5s/epoch - 11ms/step
Epoch 16/20
422/422 - 5s - loss: 0.1116 - accuracy: 0.9667 - val_loss: 0.0934 -
val_accuracy: 0.9748 - 5s/epoch - 11ms/step
Epoch 17/20
422/422 - 5s - loss: 0.1055 - accuracy: 0.9689 - val_loss: 0.0865 -
val_accuracy: 0.9750 - 5s/epoch - 11ms/step
Epoch 18/20
422/422 - 5s - loss: 0.1000 - accuracy: 0.9701 - val_loss: 0.0825 -
val_accuracy: 0.9768 - 5s/epoch - 11ms/step
Epoch 19/20
422/422 - 5s - loss: 0.0949 - accuracy: 0.9719 - val_loss: 0.0759 -
val_accuracy: 0.9797 - 5s/epoch - 11ms/step
Epoch 20/20
422/422 - 5s - loss: 0.0912 - accuracy: 0.9727 - val_loss: 0.0731 -
val_accuracy: 0.9798 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 972   0   2   0   0   1   1   1   3   0]
 [  0 1125   2   1   0   0   3   1   3   0]
 [  5   3 1006   2   2   0   1   8   5   0]
 [  0   0   4  984   0  11   0   5   5   1]
 [  1   0   3   0  960   0   7   2   2   7]]
```

```
[ 3 0 1 5 0 875 3 1 3 1
[ 6 3 1 0 5 6 935 0 2 0
[ 2 6 17 4 0 0 0 989 3 7
[ 8 0 5 4 5 0 1 3 945 3
[ 7 7 0 8 12 5 0 6 2 962]]
```

Precision: 0.9753

Recall: 0.9753



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 5s - loss: 2.3066 - accuracy: 0.1071 - val_loss: 2.3014 - val_accuracy: 0.1050 - 5s/epoch - 22ms/step

Epoch 2/5

211/211 - 4s - loss: 2.2965 - accuracy: 0.1214 - val_loss: 2.2913 - val_accuracy: 0.1050 - 4s/epoch - 18ms/step

Epoch 3/5

211/211 - 4s - loss: 2.2856 - accuracy: 0.1557 - val_loss: 2.2771 - val_accuracy: 0.1045 - 4s/epoch - 18ms/step

Epoch 4/5

211/211 - 4s - loss: 2.2552 - accuracy: 0.2290 - val_loss: 2.2220 -

```
val_accuracy: 0.4287 - 4s/epoch - 19ms/step
Epoch 5/5
211/211 - 4s - loss: 2.1124 - accuracy: 0.4225 - val_loss: 1.8991 -
val_accuracy: 0.5505 - 4s/epoch - 19ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 941   7  12   1   8   0  10   1   0   0]
 [  0 1130   2   1   1   0   1   0   0   0]
 [ 72 177 671   0  49   0  55   8   0   0]
 [149 237 225 306  65   0  13  15   0   0]
 [  8   74   2   0 878   0  20   0   0   0]
 [249 247 100 101 140   0  35  20   0   0]
 [ 60   84  16   0  83   0 715   0   0   0]
 [  8 216  16   0 127   0   0 661   0   0]
 [ 93 317 206  17 302   0  24   9   6   0]
 [ 27 148   5   3 805   0   0  21   0   0]]]
Precision: 0.5366
Recall: 0.5308

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9
True Labels	941	7	12	1	8	0	10	1	0	0
0	0	1130	2	1	1	0	1	0	0	0
1	72	177	671	0	49	0	55	8	0	0
2	149	237	225	306	65	0	13	15	0	0
3	8	74	2	0	878	0	20	0	0	0
4	249	247	100	101	140	0	35	20	0	0
5	60	84	16	0	83	0	715	0	0	0
6	8	216	16	0	127	0	0	661	0	0
7	93	317	206	17	302	0	24	9	6	0
8	27	148	5	3	805	0	0	21	0	0
9	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 4s - loss: 2.3074 - accuracy: 0.1201 - val_loss: 2.2978 -
val_accuracy: 0.1638 - 4s/epoch - 21ms/step
Epoch 2/15
211/211 - 4s - loss: 2.2915 - accuracy: 0.1444 - val_loss: 2.2846 -
val_accuracy: 0.0952 - 4s/epoch - 17ms/step
Epoch 3/15
211/211 - 4s - loss: 2.2662 - accuracy: 0.2118 - val_loss: 2.2340 -
val_accuracy: 0.3247 - 4s/epoch - 17ms/step
Epoch 4/15
211/211 - 4s - loss: 2.1350 - accuracy: 0.3329 - val_loss: 1.9477 -
val_accuracy: 0.4525 - 4s/epoch - 17ms/step
Epoch 5/15
211/211 - 4s - loss: 1.6504 - accuracy: 0.5387 - val_loss: 1.3007 -
val_accuracy: 0.6623 - 4s/epoch - 17ms/step
Epoch 6/15
211/211 - 3s - loss: 1.0776 - accuracy: 0.7011 - val_loss: 0.8100 -
val_accuracy: 0.7953 - 3s/epoch - 17ms/step
Epoch 7/15
```

```
211/211 - 4s - loss: 0.7465 - accuracy: 0.7921 - val_loss: 0.5817 -  
val_accuracy: 0.8483 - 4s/epoch - 17ms/step  
Epoch 8/15  
211/211 - 4s - loss: 0.5899 - accuracy: 0.8321 - val_loss: 0.4720 -  
val_accuracy: 0.8713 - 4s/epoch - 17ms/step  
Epoch 9/15  
211/211 - 4s - loss: 0.5025 - accuracy: 0.8551 - val_loss: 0.4015 -  
val_accuracy: 0.8902 - 4s/epoch - 17ms/step  
Epoch 10/15  
211/211 - 4s - loss: 0.4410 - accuracy: 0.8723 - val_loss: 0.3504 -  
val_accuracy: 0.9047 - 4s/epoch - 17ms/step  
Epoch 11/15  
211/211 - 4s - loss: 0.3932 - accuracy: 0.8873 - val_loss: 0.3091 -  
val_accuracy: 0.9160 - 4s/epoch - 18ms/step  
Epoch 12/15  
211/211 - 4s - loss: 0.3532 - accuracy: 0.8985 - val_loss: 0.2777 -  
val_accuracy: 0.9253 - 4s/epoch - 18ms/step  
Epoch 13/15  
211/211 - 4s - loss: 0.3208 - accuracy: 0.9069 - val_loss: 0.2497 -  
val_accuracy: 0.9337 - 4s/epoch - 18ms/step  
Epoch 14/15  
211/211 - 4s - loss: 0.2927 - accuracy: 0.9156 - val_loss: 0.2277 -  
val_accuracy: 0.9388 - 4s/epoch - 18ms/step  
Epoch 15/15  
211/211 - 4s - loss: 0.2683 - accuracy: 0.9225 - val_loss: 0.2088 -  
val_accuracy: 0.9442 - 4s/epoch - 17ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 964  0  1  1  0  4  4  1  5  0]  
[  0 1108  3  2  0  2  6  1 13  0]  
[ 10  2 943  8 11  1 17 11 24  5]  
[  2  2 22 928  0 20  0 13 17  6]  
[  1  2  4  1 920  1 15  0  3 35]  
[ 13  4  4 20  6 807 17  5 13  3]  
[ 10  3  3  0  9 13 918  0  2  0]  
[  3 10 29  5  9  0  0 936  4 32]  
[  8  4  4 15 12 14  9  7 887 14]  
[ 12  7  3  7 50  5  0 22 13 890]]  
Precision: 0.9301  
Recall: 0.9301
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	964	0	1	1	0	4	4	1	5	0	
0 -	0	1108	3	2	0	2	6	1	13	0	
1 -	10	2	943	8	11	1	17	11	24	5	
2 -	2	2	22	928	0	20	0	13	17	6	
3 -	1	2	4	1	920	1	15	0	3	35	
4 -	13	4	4	20	6	807	17	5	13	3	
5 -	10	3	3	0	9	13	918	0	2	0	
6 -	3	10	29	5	9	0	0	936	4	32	
7 -	8	4	4	15	12	14	9	7	887	14	
8 -	12	7	3	7	50	5	0	22	13	890	
9 -	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

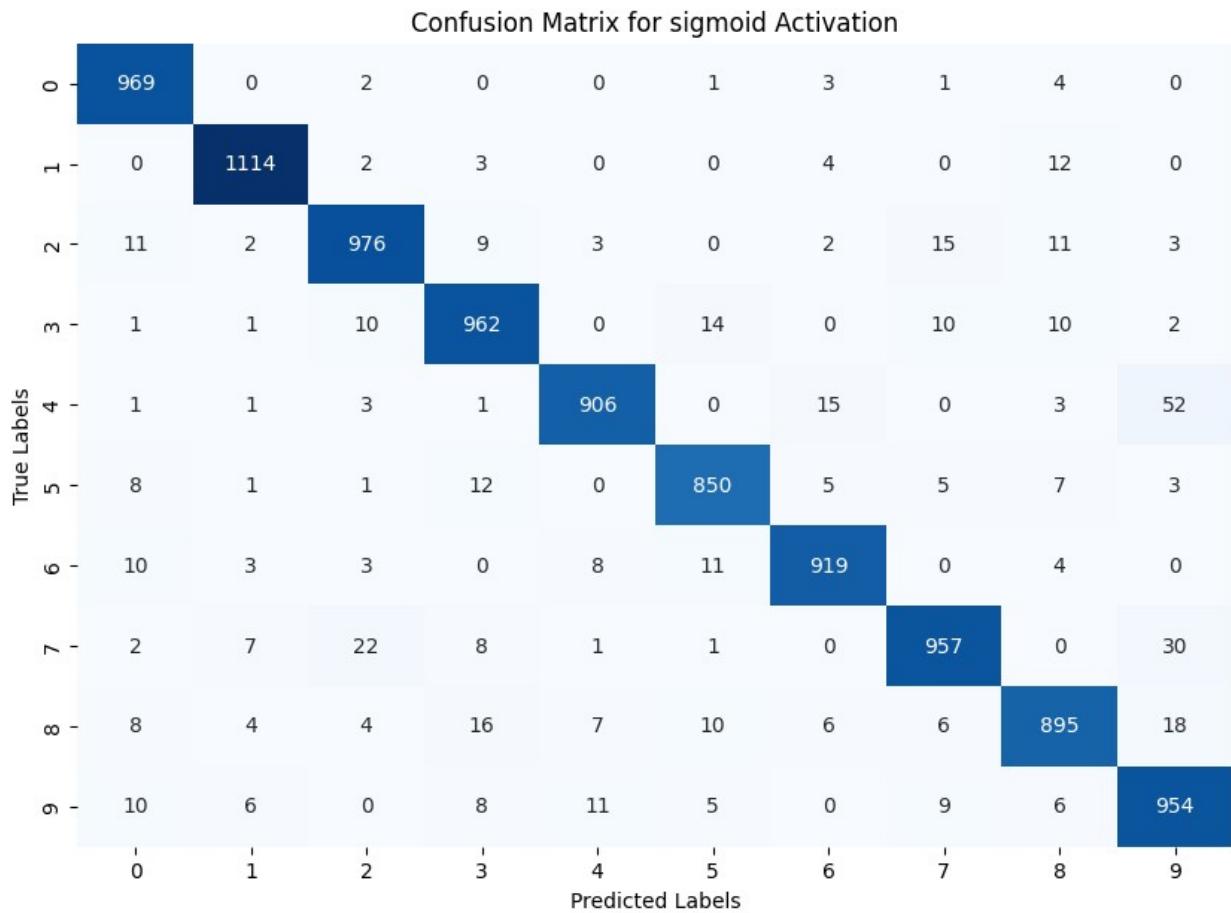
```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 4s - loss: 2.3228 - accuracy: 0.1093 - val_loss: 2.3010 -
val_accuracy: 0.1000 - 4s/epoch - 21ms/step
Epoch 2/20
211/211 - 4s - loss: 2.2982 - accuracy: 0.1202 - val_loss: 2.2930 -
val_accuracy: 0.2623 - 4s/epoch - 18ms/step
Epoch 3/20
211/211 - 4s - loss: 2.2888 - accuracy: 0.1447 - val_loss: 2.2800 -
val_accuracy: 0.2082 - 4s/epoch - 18ms/step
Epoch 4/20
211/211 - 4s - loss: 2.2633 - accuracy: 0.2042 - val_loss: 2.2364 -
val_accuracy: 0.2023 - 4s/epoch - 18ms/step
Epoch 5/20
211/211 - 4s - loss: 2.1512 - accuracy: 0.3297 - val_loss: 1.9945 -
val_accuracy: 0.4060 - 4s/epoch - 18ms/step
Epoch 6/20
211/211 - 4s - loss: 1.7109 - accuracy: 0.5367 - val_loss: 1.3740 -
val_accuracy: 0.6742 - 4s/epoch - 18ms/step
Epoch 7/20
```

```
211/211 - 4s - loss: 1.1243 - accuracy: 0.7106 - val_loss: 0.8320 -  
val_accuracy: 0.8062 - 4s/epoch - 18ms/step  
Epoch 8/20  
211/211 - 4s - loss: 0.7492 - accuracy: 0.7961 - val_loss: 0.5781 -  
val_accuracy: 0.8537 - 4s/epoch - 18ms/step  
Epoch 9/20  
211/211 - 4s - loss: 0.5802 - accuracy: 0.8366 - val_loss: 0.4596 -  
val_accuracy: 0.8767 - 4s/epoch - 18ms/step  
Epoch 10/20  
211/211 - 4s - loss: 0.4873 - accuracy: 0.8619 - val_loss: 0.3872 -  
val_accuracy: 0.8952 - 4s/epoch - 18ms/step  
Epoch 11/20  
211/211 - 4s - loss: 0.4225 - accuracy: 0.8785 - val_loss: 0.3361 -  
val_accuracy: 0.9108 - 4s/epoch - 17ms/step  
Epoch 12/20  
211/211 - 4s - loss: 0.3730 - accuracy: 0.8931 - val_loss: 0.2952 -  
val_accuracy: 0.9195 - 4s/epoch - 17ms/step  
Epoch 13/20  
211/211 - 4s - loss: 0.3339 - accuracy: 0.9037 - val_loss: 0.2614 -  
val_accuracy: 0.9283 - 4s/epoch - 17ms/step  
Epoch 14/20  
211/211 - 4s - loss: 0.3021 - accuracy: 0.9119 - val_loss: 0.2367 -  
val_accuracy: 0.9330 - 4s/epoch - 17ms/step  
Epoch 15/20  
211/211 - 4s - loss: 0.2753 - accuracy: 0.9203 - val_loss: 0.2178 -  
val_accuracy: 0.9398 - 4s/epoch - 18ms/step  
Epoch 16/20  
211/211 - 4s - loss: 0.2520 - accuracy: 0.9263 - val_loss: 0.1962 -  
val_accuracy: 0.9483 - 4s/epoch - 18ms/step  
Epoch 17/20  
211/211 - 4s - loss: 0.2324 - accuracy: 0.9322 - val_loss: 0.1793 -  
val_accuracy: 0.9520 - 4s/epoch - 17ms/step  
Epoch 18/20  
211/211 - 4s - loss: 0.2160 - accuracy: 0.9370 - val_loss: 0.1672 -  
val_accuracy: 0.9553 - 4s/epoch - 17ms/step  
Epoch 19/20  
211/211 - 4s - loss: 0.2002 - accuracy: 0.9416 - val_loss: 0.1548 -  
val_accuracy: 0.9580 - 4s/epoch - 17ms/step  
Epoch 20/20  
211/211 - 4s - loss: 0.1877 - accuracy: 0.9452 - val_loss: 0.1439 -  
val_accuracy: 0.9607 - 4s/epoch - 18ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 969 0 2 0 0 1 3 1 4 0]  
[ 0 1114 2 3 0 0 4 0 12 0]  
[ 11 2 976 9 3 0 2 15 11 3]  
[ 1 1 10 962 0 14 0 10 10 2]  
[ 1 1 3 1 906 0 15 0 3 52]]
```

```
[ 8  1  1 12  0 850  5  5  7  3]
[10  3  3  0  8 11 919  0  4  0]
[ 2  7 22  8  1  1  0 957  0 30]
[ 8  4  4 16  7 10  6  6 895 18]
[10  6  0  8 11  5  0  9  6 954]]
```

Precision: 0.9505

Recall: 0.9502



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 7s - loss: 2.2854 - accuracy: 0.1436 - val_loss: 2.0518 - val_accuracy: 0.4588 - 7s/epoch - 9ms/step

Epoch 2/5

844/844 - 7s - loss: 0.9885 - accuracy: 0.7135 - val_loss: 0.4376 - val_accuracy: 0.8822 - 7s/epoch - 8ms/step

Epoch 3/5

844/844 - 7s - loss: 0.3895 - accuracy: 0.8867 - val_loss: 0.2473 - val_accuracy: 0.9315 - 7s/epoch - 8ms/step

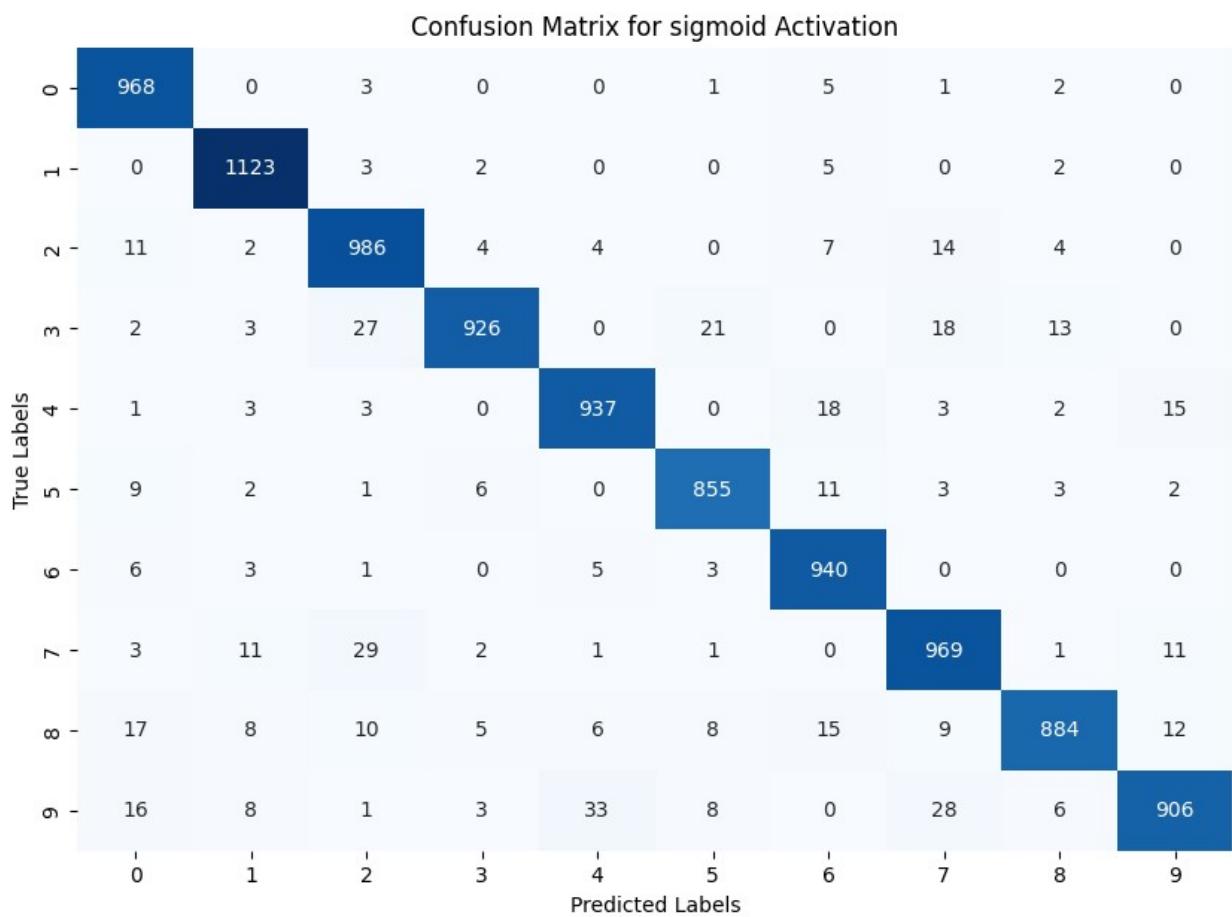
Epoch 4/5

844/844 - 7s - loss: 0.2556 - accuracy: 0.9249 - val_loss: 0.1866 -

```

val_accuracy: 0.9463 - 7s/epoch - 8ms/step
Epoch 5/5
844/844 - 7s - loss: 0.1932 - accuracy: 0.9417 - val_loss: 0.1455 -
val_accuracy: 0.9583 - 7s/epoch - 8ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 968   0   3   0   0   1   5   1   2   0]
 [  0 1123   3   2   0   0   5   0   2   0]
 [ 11   2 986   4   4   0   7  14   4   0]
 [  2   3 27 926   0  21   0  18  13   0]
 [  1   3   3   0 937   0  18   3   2  15]
 [  9   2   1   6   0 855   11   3   3   2]
 [  6   3   1   0   5   3 940   0   0   0]
 [  3  11  29   2   1   1   0 969   1  11]
 [ 17   8  10   5   6   8  15   9 884  12]
 [ 16   8   1   3  33   8   0  28   6 906]]
Precision: 0.9499
Recall: 0.9494

```



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
64 batch size, 15 epochs..

Epoch 1/15
844/844 - 7s - loss: 2.3017 - accuracy: 0.1282 - val_loss: 2.1973 -
val_accuracy: 0.2143 - 7s/epoch - 9ms/step

Epoch 2/15
844/844 - 6s - loss: 1.1948 - accuracy: 0.6481 - val_loss: 0.4656 -
val_accuracy: 0.8685 - 6s/epoch - 8ms/step

Epoch 3/15
844/844 - 7s - loss: 0.4120 - accuracy: 0.8780 - val_loss: 0.2541 -
val_accuracy: 0.9280 - 7s/epoch - 8ms/step

Epoch 4/15
844/844 - 7s - loss: 0.2668 - accuracy: 0.9211 - val_loss: 0.1789 -
val_accuracy: 0.9497 - 7s/epoch - 8ms/step

Epoch 5/15
844/844 - 7s - loss: 0.1951 - accuracy: 0.9422 - val_loss: 0.1351 -
val_accuracy: 0.9658 - 7s/epoch - 8ms/step

Epoch 6/15
844/844 - 7s - loss: 0.1560 - accuracy: 0.9538 - val_loss: 0.1103 -
val_accuracy: 0.9695 - 7s/epoch - 8ms/step

Epoch 7/15
844/844 - 7s - loss: 0.1309 - accuracy: 0.9605 - val_loss: 0.0995 -
val_accuracy: 0.9722 - 7s/epoch - 8ms/step

Epoch 8/15
844/844 - 7s - loss: 0.1163 - accuracy: 0.9650 - val_loss: 0.0892 -
val_accuracy: 0.9755 - 7s/epoch - 8ms/step

Epoch 9/15
844/844 - 7s - loss: 0.1049 - accuracy: 0.9687 - val_loss: 0.0791 -
val_accuracy: 0.9777 - 7s/epoch - 8ms/step

Epoch 10/15
844/844 - 7s - loss: 0.0947 - accuracy: 0.9706 - val_loss: 0.0767 -
val_accuracy: 0.9792 - 7s/epoch - 8ms/step

Epoch 11/15
844/844 - 7s - loss: 0.0876 - accuracy: 0.9733 - val_loss: 0.0772 -
val_accuracy: 0.9785 - 7s/epoch - 8ms/step

Epoch 12/15
844/844 - 7s - loss: 0.0827 - accuracy: 0.9746 - val_loss: 0.0720 -
val_accuracy: 0.9795 - 7s/epoch - 8ms/step

Epoch 13/15
844/844 - 7s - loss: 0.0773 - accuracy: 0.9763 - val_loss: 0.0634 -
val_accuracy: 0.9813 - 7s/epoch - 8ms/step

Epoch 14/15
844/844 - 7s - loss: 0.0731 - accuracy: 0.9779 - val_loss: 0.0668 -
val_accuracy: 0.9818 - 7s/epoch - 8ms/step

Epoch 15/15
844/844 - 7s - loss: 0.0695 - accuracy: 0.9790 - val_loss: 0.0606 -
val_accuracy: 0.9822 - 7s/epoch - 8ms/step

313/313 [=====] - 1s 3ms/step

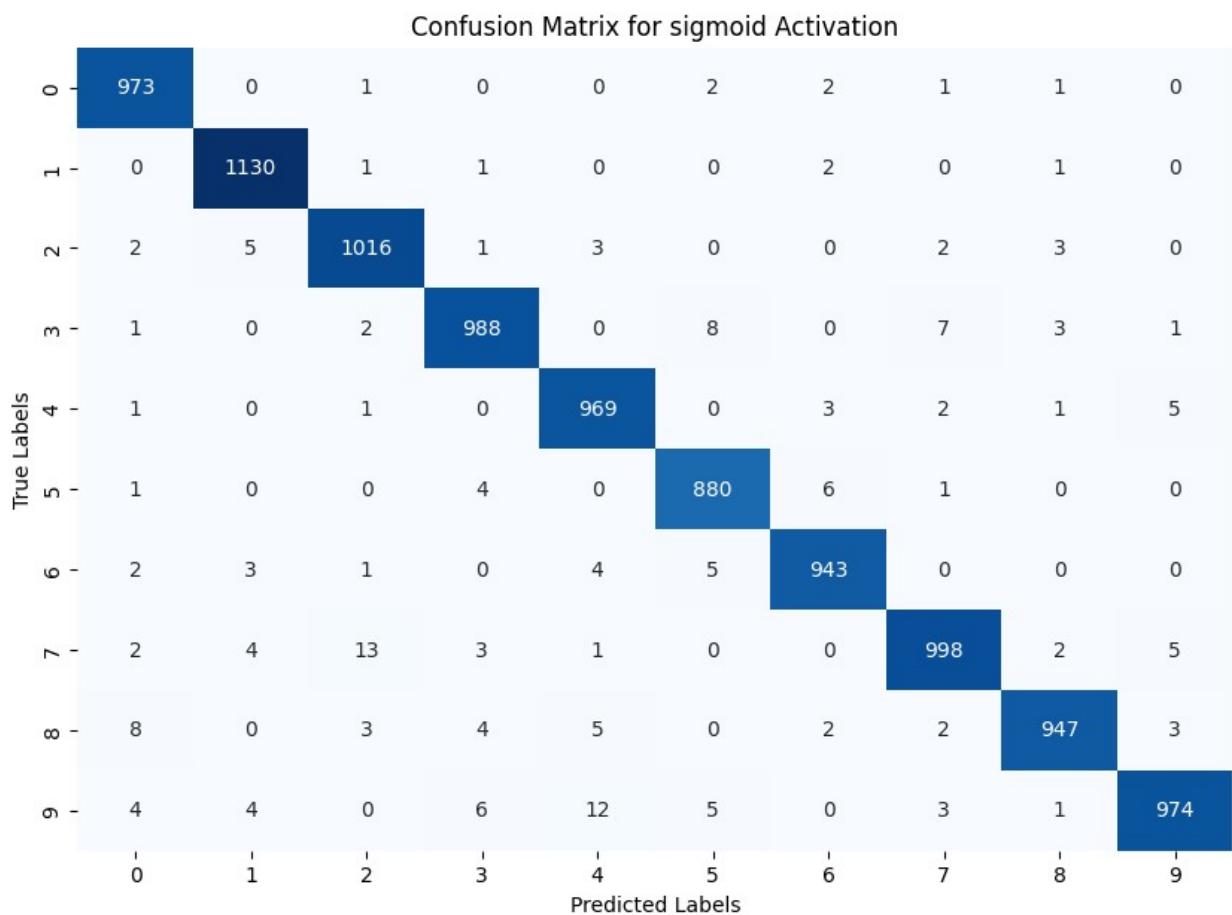
Results for activation function: sigmoid

Confusion Matrix:

```
[[ 973 0 1 0 0 2 2 1 1 0]
 [ 0 1130 1 1 0 0 2 0 1 0]
 [ 2 5 1016 1 3 0 0 2 3 0]
 [ 1 0 2 988 0 8 0 7 3 1]
 [ 1 0 1 0 969 0 3 2 1 5]
 [ 1 0 0 4 0 880 6 1 0 0]
 [ 2 3 1 0 4 5 943 0 0 0]
 [ 2 4 13 3 1 0 0 998 2 5]
 [ 8 0 3 4 5 0 2 2 947 3]
 [ 4 4 0 6 12 5 0 3 1 974]]
```

Precision: 0.9818

Recall: 0.9818



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 64 batch size, 20 epochs..

Epoch 1/20

844/844 - 8s - loss: 2.2816 - accuracy: 0.1513 - val_loss: 2.0305 - val_accuracy: 0.3938 - 8s/epoch - 9ms/step

Epoch 2/20

844/844 - 7s - loss: 0.9506 - accuracy: 0.7289 - val_loss: 0.3901 - val_accuracy: 0.8943 - 7s/epoch - 8ms/step

```
Epoch 3/20
844/844 - 7s - loss: 0.3628 - accuracy: 0.8941 - val_loss: 0.2391 -
val_accuracy: 0.9327 - 7s/epoch - 8ms/step
Epoch 4/20
844/844 - 7s - loss: 0.2533 - accuracy: 0.9251 - val_loss: 0.1734 -
val_accuracy: 0.9513 - 7s/epoch - 8ms/step
Epoch 5/20
844/844 - 7s - loss: 0.1956 - accuracy: 0.9418 - val_loss: 0.1372 -
val_accuracy: 0.9625 - 7s/epoch - 8ms/step
Epoch 6/20
844/844 - 7s - loss: 0.1586 - accuracy: 0.9523 - val_loss: 0.1112 -
val_accuracy: 0.9705 - 7s/epoch - 8ms/step
Epoch 7/20
844/844 - 7s - loss: 0.1341 - accuracy: 0.9594 - val_loss: 0.0957 -
val_accuracy: 0.9723 - 7s/epoch - 8ms/step
Epoch 8/20
844/844 - 7s - loss: 0.1172 - accuracy: 0.9644 - val_loss: 0.0963 -
val_accuracy: 0.9720 - 7s/epoch - 8ms/step
Epoch 9/20
844/844 - 7s - loss: 0.1047 - accuracy: 0.9681 - val_loss: 0.0794 -
val_accuracy: 0.9780 - 7s/epoch - 8ms/step
Epoch 10/20
844/844 - 7s - loss: 0.0949 - accuracy: 0.9711 - val_loss: 0.0774 -
val_accuracy: 0.9783 - 7s/epoch - 8ms/step
Epoch 11/20
844/844 - 7s - loss: 0.0878 - accuracy: 0.9728 - val_loss: 0.0706 -
val_accuracy: 0.9812 - 7s/epoch - 8ms/step
Epoch 12/20
844/844 - 7s - loss: 0.0818 - accuracy: 0.9750 - val_loss: 0.0737 -
val_accuracy: 0.9783 - 7s/epoch - 8ms/step
Epoch 13/20
844/844 - 7s - loss: 0.0768 - accuracy: 0.9759 - val_loss: 0.0692 -
val_accuracy: 0.9803 - 7s/epoch - 8ms/step
Epoch 14/20
844/844 - 7s - loss: 0.0718 - accuracy: 0.9776 - val_loss: 0.0620 -
val_accuracy: 0.9822 - 7s/epoch - 8ms/step
Epoch 15/20
844/844 - 7s - loss: 0.0678 - accuracy: 0.9788 - val_loss: 0.0624 -
val_accuracy: 0.9817 - 7s/epoch - 8ms/step
Epoch 16/20
844/844 - 7s - loss: 0.0661 - accuracy: 0.9796 - val_loss: 0.0589 -
val_accuracy: 0.9830 - 7s/epoch - 8ms/step
Epoch 17/20
844/844 - 7s - loss: 0.0609 - accuracy: 0.9814 - val_loss: 0.0620 -
val_accuracy: 0.9817 - 7s/epoch - 8ms/step
Epoch 18/20
844/844 - 7s - loss: 0.0596 - accuracy: 0.9818 - val_loss: 0.0563 -
val_accuracy: 0.9833 - 7s/epoch - 8ms/step
Epoch 19/20
```

```
844/844 - 7s - loss: 0.0563 - accuracy: 0.9828 - val_loss: 0.0552 -  
val_accuracy: 0.9848 - 7s/epoch - 8ms/step
```

```
Epoch 20/20
```

```
844/844 - 7s - loss: 0.0535 - accuracy: 0.9831 - val_loss: 0.0535 -  
val_accuracy: 0.9847 - 7s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 3ms/step
```

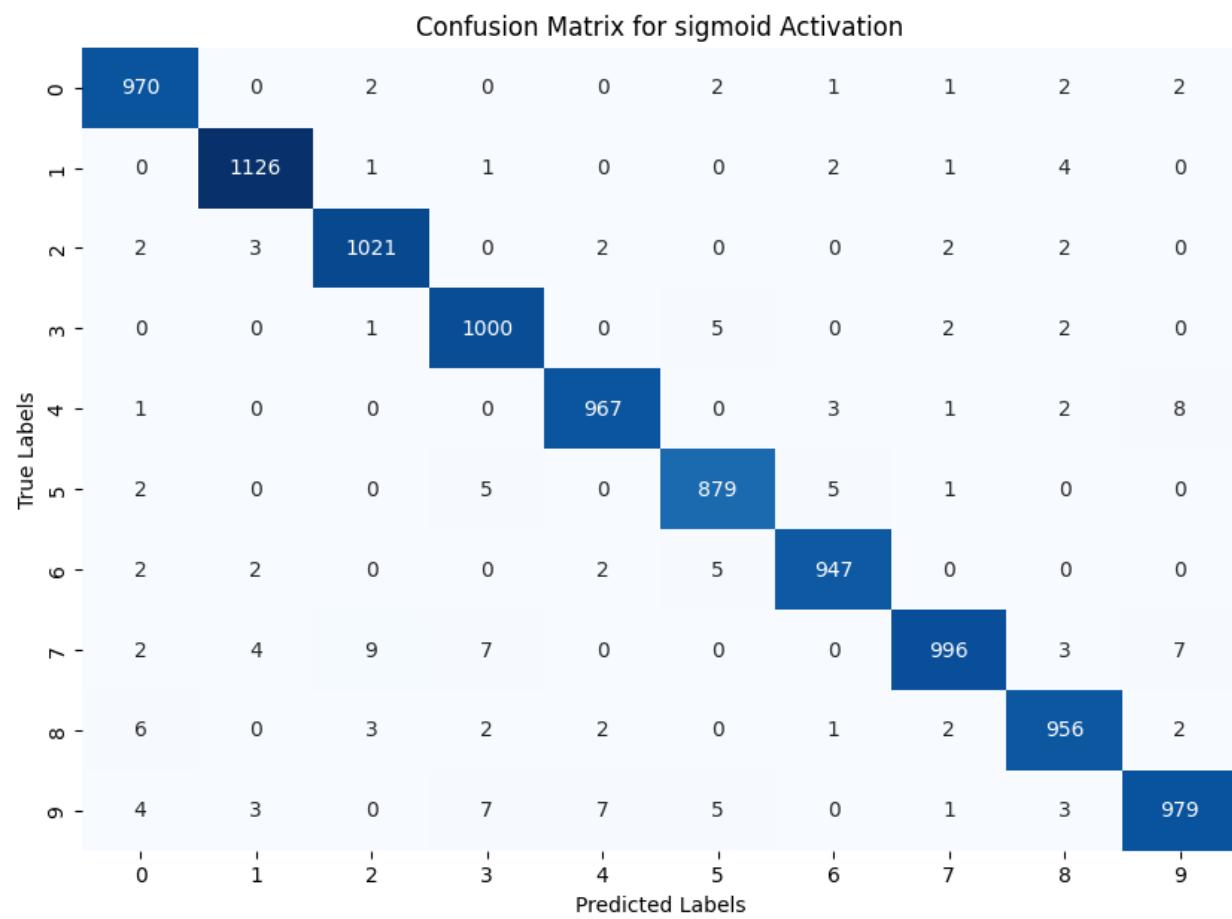
```
Results for activation function: sigmoid
```

```
Confusion Matrix:
```

```
[[ 970   0   2   0   0   2   1   1   2   2  
[  0 1126   1   1   0   0   2   1   4   0]  
[  2   3 1021   0   2   0   0   2   2   0]  
[  0   0   1 1000   0   5   0   2   2   0]  
[  1   0   0   0   967   0   3   1   2   8]  
[  2   0   0   5   0   879   5   1   0   0]  
[  2   2   0   0   2   5   947   0   0   0]  
[  2   4   9   7   0   0   0   996   3   7]  
[  6   0   3   2   2   0   1   2   956   2]  
[  4   3   0   7   7   5   0   1   3   979]]
```

```
Precision: 0.9841
```

```
Recall: 0.9841
```



```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 6s - loss: 2.3101 - accuracy: 0.1161 - val_loss: 2.2983 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 2/5
422/422 - 5s - loss: 2.2406 - accuracy: 0.2095 - val_loss: 2.0793 -
val_accuracy: 0.3075 - 5s/epoch - 11ms/step
Epoch 3/5
422/422 - 5s - loss: 1.3762 - accuracy: 0.6104 - val_loss: 0.7588 -
val_accuracy: 0.8173 - 5s/epoch - 11ms/step
Epoch 4/5
422/422 - 5s - loss: 0.6133 - accuracy: 0.8280 - val_loss: 0.4152 -
val_accuracy: 0.8897 - 5s/epoch - 11ms/step
Epoch 5/5
422/422 - 5s - loss: 0.4227 - accuracy: 0.8773 - val_loss: 0.3041 -
val_accuracy: 0.9178 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 967    0    2    1    0    4    1    1    4    0]
 [  0 1105    3    5    0    2    3    0   17    0]
 [ 15    8  924    9   13    4   19   15   16    9]
 [  2    6   27  911    0   25    1   18   16    4]
 [  2    7    4    0  882    2   21    0    3   61]
 [ 24    5    6   52   10  756   18    2   14    5]
 [ 22    4    7    0    9   23  890    0    3    0]
 [  5   19   30    5    6    0    0  906    4   53]
 [ 18    9    5   43   14   42   22    9  796   16]
 [ 16    6    4    8   67   11    0   25   13  859]]
```

Precision: 0.8996
Recall: 0.8996

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	967	0	2	1	0	4	1	1	4	0	
1	0	1105	3	5	0	2	3	0	17	0	
2	15	8	924	9	13	4	19	15	16	9	
3	2	6	27	911	0	25	1	18	16	4	
4	2	7	4	0	882	2	21	0	3	61	
5	24	5	6	52	10	756	18	2	14	5	
6	22	4	7	0	9	23	890	0	3	0	
7	5	19	30	5	6	0	0	906	4	53	
8	18	9	5	43	14	42	22	9	796	16	
9	16	6	4	8	67	11	0	25	13	859	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
True Labels	Predicted Labels										

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 6s - loss: 2.3115 - accuracy: 0.1119 - val_loss: 2.3002 -
val_accuracy: 0.1715 - 6s/epoch - 14ms/step
Epoch 2/15
422/422 - 5s - loss: 2.2663 - accuracy: 0.1805 - val_loss: 2.1749 -
val_accuracy: 0.2557 - 5s/epoch - 11ms/step
Epoch 3/15
422/422 - 5s - loss: 1.5807 - accuracy: 0.5359 - val_loss: 0.8770 -
val_accuracy: 0.7748 - 5s/epoch - 12ms/step
Epoch 4/15
422/422 - 5s - loss: 0.6811 - accuracy: 0.8046 - val_loss: 0.4669 -
val_accuracy: 0.8652 - 5s/epoch - 12ms/step
Epoch 5/15
422/422 - 5s - loss: 0.4499 - accuracy: 0.8700 - val_loss: 0.3223 -
val_accuracy: 0.9118 - 5s/epoch - 11ms/step
Epoch 6/15
422/422 - 5s - loss: 0.3438 - accuracy: 0.9000 - val_loss: 0.2508 -
val_accuracy: 0.9323 - 5s/epoch - 12ms/step
Epoch 7/15
```

```
422/422 - 5s - loss: 0.2804 - accuracy: 0.9178 - val_loss: 0.2081 -  
val_accuracy: 0.9427 - 5s/epoch - 12ms/step  
Epoch 8/15  
422/422 - 5s - loss: 0.2375 - accuracy: 0.9298 - val_loss: 0.1750 -  
val_accuracy: 0.9538 - 5s/epoch - 11ms/step  
Epoch 9/15  
422/422 - 5s - loss: 0.2058 - accuracy: 0.9400 - val_loss: 0.1509 -  
val_accuracy: 0.9590 - 5s/epoch - 12ms/step  
Epoch 10/15  
422/422 - 5s - loss: 0.1813 - accuracy: 0.9465 - val_loss: 0.1357 -  
val_accuracy: 0.9642 - 5s/epoch - 12ms/step  
Epoch 11/15  
422/422 - 5s - loss: 0.1628 - accuracy: 0.9516 - val_loss: 0.1231 -  
val_accuracy: 0.9673 - 5s/epoch - 12ms/step  
Epoch 12/15  
422/422 - 5s - loss: 0.1475 - accuracy: 0.9563 - val_loss: 0.1106 -  
val_accuracy: 0.9703 - 5s/epoch - 11ms/step  
Epoch 13/15  
422/422 - 5s - loss: 0.1355 - accuracy: 0.9598 - val_loss: 0.1024 -  
val_accuracy: 0.9720 - 5s/epoch - 12ms/step  
Epoch 14/15  
422/422 - 5s - loss: 0.1250 - accuracy: 0.9630 - val_loss: 0.0969 -  
val_accuracy: 0.9725 - 5s/epoch - 11ms/step  
Epoch 15/15  
422/422 - 5s - loss: 0.1166 - accuracy: 0.9657 - val_loss: 0.0924 -  
val_accuracy: 0.9743 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 969  0  3  0  0  2  2  1  3  0]  
[ 0 1112  3  2  0  0  4  0  14  0]  
[ 5  1 1011  5  2  0  0  2  6  0]  
[ 0  0  4 985  0 11  0  2  7  1]  
[ 1  1  2  0 943  0 10  1  4 20]  
[ 3  0  1  7  0 870  4  1  4  2]  
[ 6  3  2  0  5  8 932  0  2  0]  
[ 2  6  27 12  1  1  0 953  3 23]  
[ 7  0  4  8  3  3  0  1 945  3]  
[ 8  5  0  8 12  6  0  4  5 961]]  
Precision: 0.9683  
Recall: 0.9681
```

Confusion Matrix for sigmoid Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	0	969	0	3	0	0	2	2	1	3	0
0	0	1112	3	2	0	0	4	0	14	0	
1	5	1	1011	5	2	0	0	2	6	0	
2	0	0	4	985	0	11	0	2	7	1	
3	1	1	2	0	943	0	10	1	4	20	
4	3	0	1	7	0	870	4	1	4	2	
5	6	3	2	0	5	8	932	0	2	0	
6	2	6	27	12	1	1	0	953	3	23	
7	7	0	4	8	3	3	0	1	945	3	
8	8	5	0	8	12	6	0	4	5	961	
9	1	2	3	4	5	6	7	8	9		
	0	1	2	3	4	5	6	7	8	9	

```

Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 6s - loss: 2.3149 - accuracy: 0.1135 - val_loss: 2.2986 -
val_accuracy: 0.0992 - 6s/epoch - 13ms/step
Epoch 2/20
422/422 - 5s - loss: 2.2460 - accuracy: 0.1935 - val_loss: 2.0880 -
val_accuracy: 0.3990 - 5s/epoch - 12ms/step
Epoch 3/20
422/422 - 5s - loss: 1.4291 - accuracy: 0.5937 - val_loss: 0.7747 -
val_accuracy: 0.8053 - 5s/epoch - 11ms/step
Epoch 4/20
422/422 - 5s - loss: 0.6324 - accuracy: 0.8164 - val_loss: 0.4419 -
val_accuracy: 0.8787 - 5s/epoch - 11ms/step
Epoch 5/20
422/422 - 5s - loss: 0.4462 - accuracy: 0.8681 - val_loss: 0.3292 -
val_accuracy: 0.9152 - 5s/epoch - 11ms/step
Epoch 6/20
422/422 - 5s - loss: 0.3557 - accuracy: 0.8969 - val_loss: 0.2626 -
val_accuracy: 0.9280 - 5s/epoch - 11ms/step
Epoch 7/20

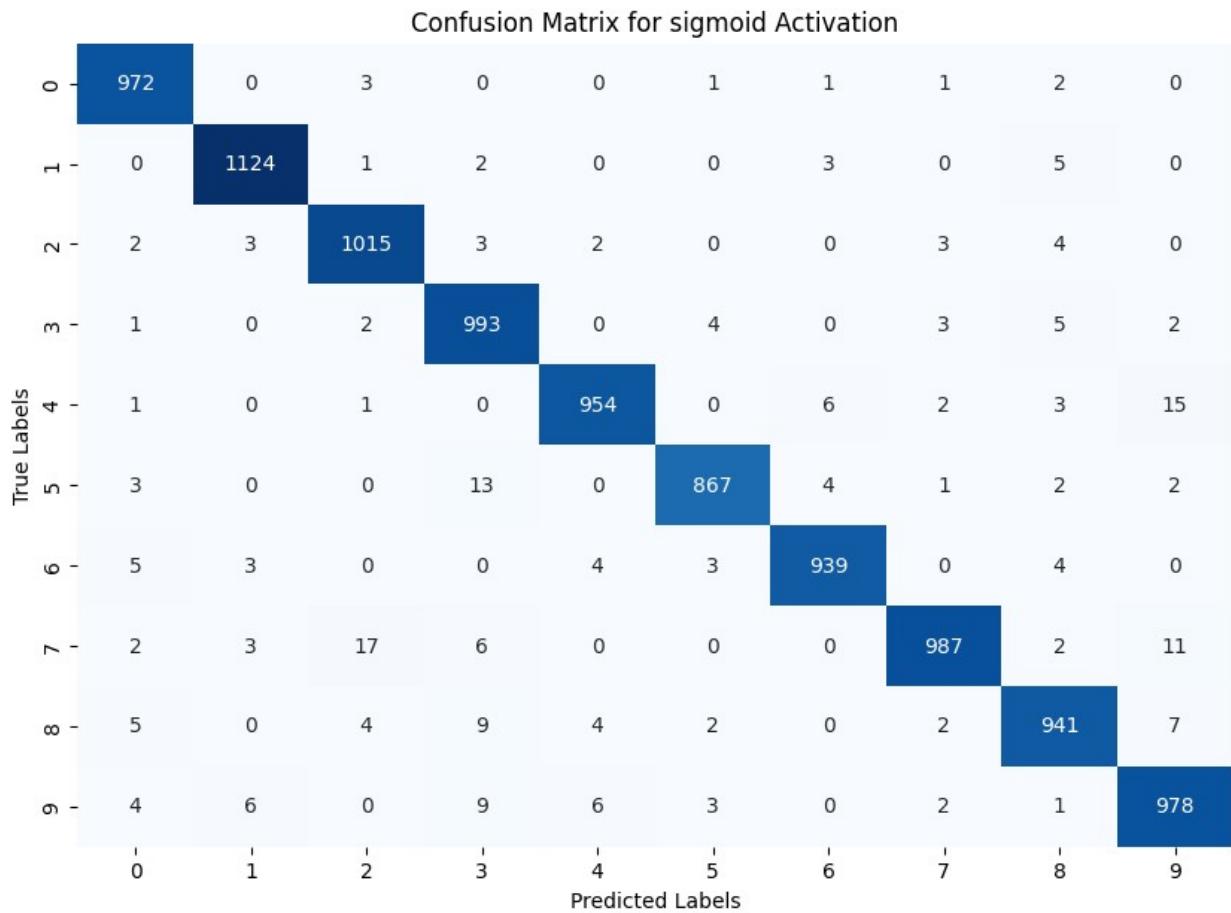
```

```
422/422 - 5s - loss: 0.2950 - accuracy: 0.9136 - val_loss: 0.2188 -  
val_accuracy: 0.9367 - 5s/epoch - 11ms/step  
Epoch 8/20  
422/422 - 5s - loss: 0.2502 - accuracy: 0.9258 - val_loss: 0.1802 -  
val_accuracy: 0.9507 - 5s/epoch - 11ms/step  
Epoch 9/20  
422/422 - 5s - loss: 0.2156 - accuracy: 0.9359 - val_loss: 0.1561 -  
val_accuracy: 0.9582 - 5s/epoch - 11ms/step  
Epoch 10/20  
422/422 - 5s - loss: 0.1893 - accuracy: 0.9436 - val_loss: 0.1435 -  
val_accuracy: 0.9598 - 5s/epoch - 11ms/step  
Epoch 11/20  
422/422 - 5s - loss: 0.1685 - accuracy: 0.9504 - val_loss: 0.1240 -  
val_accuracy: 0.9668 - 5s/epoch - 11ms/step  
Epoch 12/20  
422/422 - 5s - loss: 0.1517 - accuracy: 0.9557 - val_loss: 0.1135 -  
val_accuracy: 0.9693 - 5s/epoch - 11ms/step  
Epoch 13/20  
422/422 - 5s - loss: 0.1391 - accuracy: 0.9593 - val_loss: 0.1038 -  
val_accuracy: 0.9723 - 5s/epoch - 11ms/step  
Epoch 14/20  
422/422 - 5s - loss: 0.1281 - accuracy: 0.9621 - val_loss: 0.0981 -  
val_accuracy: 0.9733 - 5s/epoch - 11ms/step  
Epoch 15/20  
422/422 - 5s - loss: 0.1183 - accuracy: 0.9654 - val_loss: 0.0929 -  
val_accuracy: 0.9755 - 5s/epoch - 11ms/step  
Epoch 16/20  
422/422 - 5s - loss: 0.1109 - accuracy: 0.9674 - val_loss: 0.0899 -  
val_accuracy: 0.9740 - 5s/epoch - 11ms/step  
Epoch 17/20  
422/422 - 5s - loss: 0.1050 - accuracy: 0.9684 - val_loss: 0.0815 -  
val_accuracy: 0.9777 - 5s/epoch - 11ms/step  
Epoch 18/20  
422/422 - 5s - loss: 0.0995 - accuracy: 0.9709 - val_loss: 0.0822 -  
val_accuracy: 0.9777 - 5s/epoch - 11ms/step  
Epoch 19/20  
422/422 - 5s - loss: 0.0944 - accuracy: 0.9716 - val_loss: 0.0781 -  
val_accuracy: 0.9788 - 5s/epoch - 11ms/step  
Epoch 20/20  
422/422 - 5s - loss: 0.0905 - accuracy: 0.9729 - val_loss: 0.0735 -  
val_accuracy: 0.9803 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 972 0 3 0 0 1 1 1 2 0 ]  
 [ 0 1124 1 2 0 0 3 0 5 0 ]  
 [ 2 3 1015 3 2 0 0 3 4 0 ]  
 [ 1 0 2 993 0 4 0 3 5 2 ]  
 [ 1 0 1 0 954 0 6 2 3 15 ]
```

```
[ 3 0 0 13 0 867 4 1 2 2
[ 5 3 0 0 4 3 939 0 4 0
[ 2 3 17 6 0 0 0 987 2 11]
[ 5 0 4 9 4 2 0 2 941 7]
[ 4 6 0 9 6 3 0 2 1 978]]
```

Precision: 0.9771

Recall: 0.9770



Training Model with sigmoid activation, 2 conv_layers, 2 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 5s - loss: 2.3084 - accuracy: 0.1083 - val_loss: 2.3051 - val_accuracy: 0.1050 - 5s/epoch - 21ms/step

Epoch 2/5

211/211 - 4s - loss: 2.2998 - accuracy: 0.1164 - val_loss: 2.2961 - val_accuracy: 0.1047 - 4s/epoch - 18ms/step

Epoch 3/5

211/211 - 4s - loss: 2.2866 - accuracy: 0.1435 - val_loss: 2.2813 - val_accuracy: 0.1328 - 4s/epoch - 18ms/step

Epoch 4/5

211/211 - 4s - loss: 2.2490 - accuracy: 0.2209 - val_loss: 2.2037 -

```
val_accuracy: 0.1613 - 4s/epoch - 18ms/step
Epoch 5/5
211/211 - 4s - loss: 2.0224 - accuracy: 0.3973 - val_loss: 1.7136 -
val_accuracy: 0.5510 - 4s/epoch - 18ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 946   3   2   5  10   0   8   1   5   0]
 [  0 1126   3   5   0   0   1   0   0   0]
 [ 242  138  439  103  59   0  29  22   0   0]
 [ 153  128   35  622  19   0   6  40   7   0]
 [  80   54   1   0  825   0   9  11   0   2]
 [ 270   92   15  159  254   0  28  40  34   0]
 [ 328   66   6   3  111   0  442   1   1   0]
 [  28  159   13   7  130   0   0  689   2   0]
 [ 165  122   44  155  273   0  25  24  166   0]
 [  79   97   0   5  733   0   0  88   2   5]]]
Precision: 0.5798
Recall: 0.5260

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	946	3	2	5	10	0	8	1	5	0	
0	0	1126	3	5	0	0	1	0	0	0	
1	242	138	439	103	59	0	29	22	0	0	
2	153	128	35	622	19	0	6	40	7	0	
3	80	54	1	0	825	0	9	11	0	2	
4	270	92	15	159	254	0	28	40	34	0	
5	328	66	6	3	111	0	442	1	1	0	
6	28	159	13	7	130	0	0	689	2	0	
7	165	122	44	155	273	0	25	24	166	0	
8	79	97	0	5	733	0	0	88	2	5	
9	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 5s - loss: 2.3118 - accuracy: 0.1093 - val_loss: 2.2987 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 2/15
211/211 - 4s - loss: 2.2960 - accuracy: 0.1315 - val_loss: 2.2875 -
val_accuracy: 0.2008 - 4s/epoch - 18ms/step
Epoch 3/15
211/211 - 4s - loss: 2.2775 - accuracy: 0.1468 - val_loss: 2.2573 -
val_accuracy: 0.2042 - 4s/epoch - 18ms/step
Epoch 4/15
211/211 - 4s - loss: 2.2066 - accuracy: 0.2836 - val_loss: 2.1078 -
val_accuracy: 0.3668 - 4s/epoch - 18ms/step
Epoch 5/15
211/211 - 4s - loss: 1.7882 - accuracy: 0.5303 - val_loss: 1.3301 -
val_accuracy: 0.6997 - 4s/epoch - 18ms/step
Epoch 6/15
211/211 - 4s - loss: 1.0342 - accuracy: 0.7347 - val_loss: 0.7360 -
val_accuracy: 0.8223 - 4s/epoch - 18ms/step
Epoch 7/15
```

```
211/211 - 4s - loss: 0.6688 - accuracy: 0.8203 - val_loss: 0.5098 -  
val_accuracy: 0.8677 - 4s/epoch - 18ms/step  
Epoch 8/15  
211/211 - 4s - loss: 0.5178 - accuracy: 0.8558 - val_loss: 0.4014 -  
val_accuracy: 0.8968 - 4s/epoch - 19ms/step  
Epoch 9/15  
211/211 - 4s - loss: 0.4346 - accuracy: 0.8762 - val_loss: 0.3367 -  
val_accuracy: 0.9117 - 4s/epoch - 19ms/step  
Epoch 10/15  
211/211 - 4s - loss: 0.3773 - accuracy: 0.8915 - val_loss: 0.2918 -  
val_accuracy: 0.9213 - 4s/epoch - 19ms/step  
Epoch 11/15  
211/211 - 4s - loss: 0.3347 - accuracy: 0.9034 - val_loss: 0.2595 -  
val_accuracy: 0.9303 - 4s/epoch - 19ms/step  
Epoch 12/15  
211/211 - 4s - loss: 0.3017 - accuracy: 0.9133 - val_loss: 0.2351 -  
val_accuracy: 0.9383 - 4s/epoch - 18ms/step  
Epoch 13/15  
211/211 - 4s - loss: 0.2751 - accuracy: 0.9199 - val_loss: 0.2126 -  
val_accuracy: 0.9432 - 4s/epoch - 19ms/step  
Epoch 14/15  
211/211 - 4s - loss: 0.2512 - accuracy: 0.9270 - val_loss: 0.1927 -  
val_accuracy: 0.9485 - 4s/epoch - 18ms/step  
Epoch 15/15  
211/211 - 4s - loss: 0.2317 - accuracy: 0.9322 - val_loss: 0.1764 -  
val_accuracy: 0.9530 - 4s/epoch - 19ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 966  0  2  1  0  2  4  1  4  0]  
[ 0 1105  3  2  0  1  4  1  19  0]  
[ 9  0 967  9  5  0  7  15  15  5]  
[ 1  1 14 956  0  15  0  8  12  3]  
[ 1  2  3  1 921  0  13  0  3  38]  
[ 11  1  1 25  3 820  10  4  13  4]  
[ 7  3  4  0  7 13 921  0  3  0]  
[ 3  8 23  9  6  1  0  942  3  33]  
[ 8  3  2 22  8 12  10  8  888  13]  
[ 12  6  5  9 29  8  0  16  8  916]]  
Precision: 0.9402  
Recall: 0.9402
```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	966	0	2	1	0	2	4	1	4	0	
1	0	1105	3	2	0	1	4	1	19	0	
2	9	0	967	9	5	0	7	15	15	5	
3	1	1	14	956	0	15	0	8	12	3	
4	1	2	3	1	921	0	13	0	3	38	
5	11	1	1	25	3	820	10	4	13	4	
6	7	3	4	0	7	13	921	0	3	0	
7	3	8	23	9	6	1	0	942	3	33	
8	8	3	2	22	8	12	10	8	888	13	
9	12	6	5	9	29	8	0	16	8	916	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```
Training Model with sigmoid activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 5s - loss: 2.3123 - accuracy: 0.1076 - val_loss: 2.3007 -
val_accuracy: 0.1937 - 5s/epoch - 21ms/step
Epoch 2/20
211/211 - 4s - loss: 2.2947 - accuracy: 0.1346 - val_loss: 2.2878 -
val_accuracy: 0.0978 - 4s/epoch - 18ms/step
Epoch 3/20
211/211 - 4s - loss: 2.2667 - accuracy: 0.1706 - val_loss: 2.2350 -
val_accuracy: 0.2435 - 4s/epoch - 18ms/step
Epoch 4/20
211/211 - 4s - loss: 2.1183 - accuracy: 0.3420 - val_loss: 1.9003 -
val_accuracy: 0.3942 - 4s/epoch - 18ms/step
Epoch 5/20
211/211 - 4s - loss: 1.5227 - accuracy: 0.5746 - val_loss: 1.1286 -
val_accuracy: 0.7142 - 4s/epoch - 18ms/step
Epoch 6/20
211/211 - 4s - loss: 0.9390 - accuracy: 0.7457 - val_loss: 0.6895 -
val_accuracy: 0.8213 - 4s/epoch - 18ms/step
Epoch 7/20
```

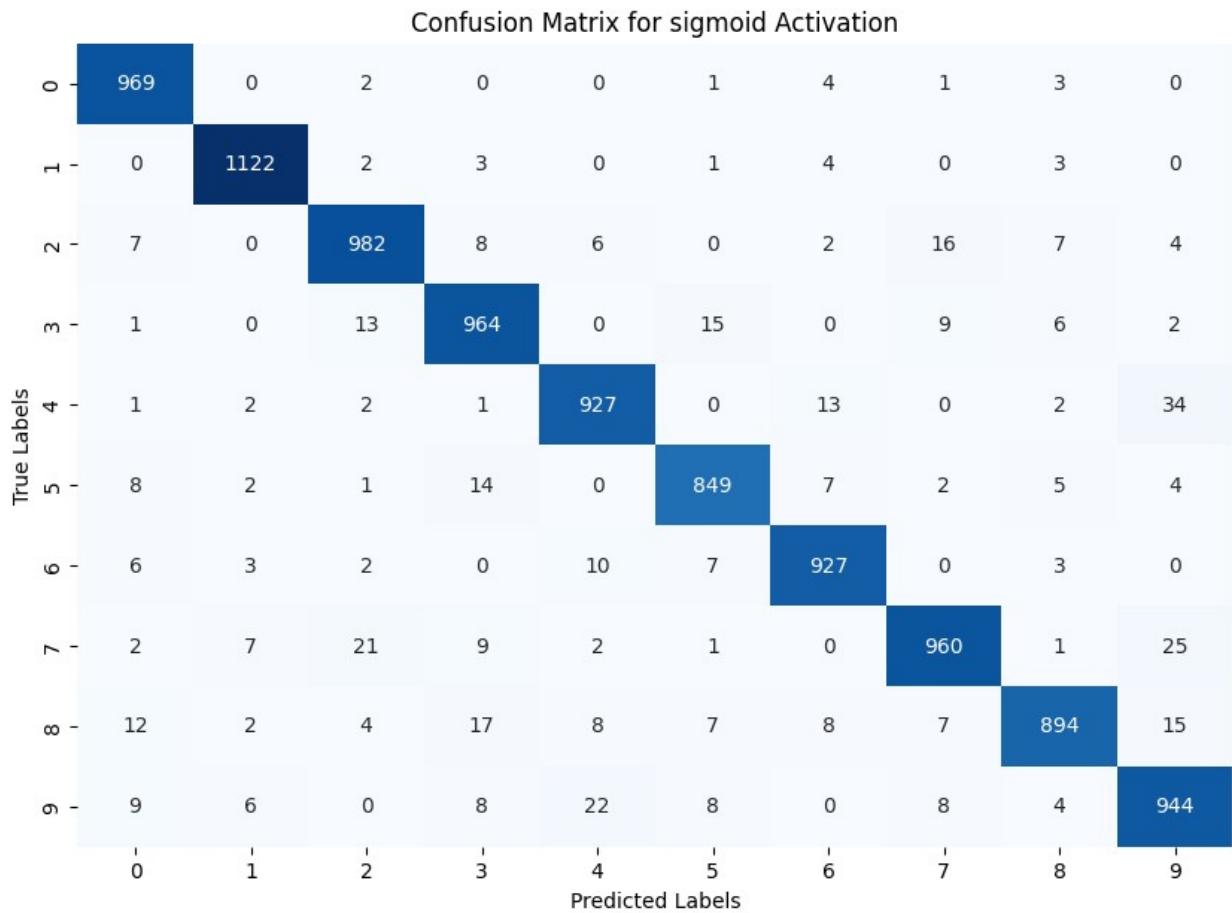
```

211/211 - 4s - loss: 0.6565 - accuracy: 0.8131 - val_loss: 0.5133 -
val_accuracy: 0.8598 - 4s/epoch - 18ms/step
Epoch 8/20
211/211 - 4s - loss: 0.5292 - accuracy: 0.8463 - val_loss: 0.4141 -
val_accuracy: 0.8863 - 4s/epoch - 18ms/step
Epoch 9/20
211/211 - 4s - loss: 0.4505 - accuracy: 0.8690 - val_loss: 0.3578 -
val_accuracy: 0.9020 - 4s/epoch - 18ms/step
Epoch 10/20
211/211 - 4s - loss: 0.3945 - accuracy: 0.8859 - val_loss: 0.3099 -
val_accuracy: 0.9162 - 4s/epoch - 18ms/step
Epoch 11/20
211/211 - 4s - loss: 0.3528 - accuracy: 0.8984 - val_loss: 0.2744 -
val_accuracy: 0.9255 - 4s/epoch - 18ms/step
Epoch 12/20
211/211 - 4s - loss: 0.3198 - accuracy: 0.9074 - val_loss: 0.2489 -
val_accuracy: 0.9315 - 4s/epoch - 18ms/step
Epoch 13/20
211/211 - 4s - loss: 0.2921 - accuracy: 0.9150 - val_loss: 0.2248 -
val_accuracy: 0.9362 - 4s/epoch - 18ms/step
Epoch 14/20
211/211 - 4s - loss: 0.2673 - accuracy: 0.9218 - val_loss: 0.2063 -
val_accuracy: 0.9420 - 4s/epoch - 18ms/step
Epoch 15/20
211/211 - 4s - loss: 0.2471 - accuracy: 0.9274 - val_loss: 0.1914 -
val_accuracy: 0.9478 - 4s/epoch - 18ms/step
Epoch 16/20
211/211 - 4s - loss: 0.2293 - accuracy: 0.9326 - val_loss: 0.1765 -
val_accuracy: 0.9513 - 4s/epoch - 18ms/step
Epoch 17/20
211/211 - 4s - loss: 0.2145 - accuracy: 0.9372 - val_loss: 0.1643 -
val_accuracy: 0.9545 - 4s/epoch - 18ms/step
Epoch 18/20
211/211 - 4s - loss: 0.1996 - accuracy: 0.9413 - val_loss: 0.1535 -
val_accuracy: 0.9570 - 4s/epoch - 18ms/step
Epoch 19/20
211/211 - 4s - loss: 0.1879 - accuracy: 0.9448 - val_loss: 0.1445 -
val_accuracy: 0.9605 - 4s/epoch - 18ms/step
Epoch 20/20
211/211 - 4s - loss: 0.1762 - accuracy: 0.9478 - val_loss: 0.1349 -
val_accuracy: 0.9632 - 4s/epoch - 18ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 969   0   2   0   0   1   4   1   3   0]
 [  0 1122   2   3   0   1   4   0   3   0]
 [  7   0 982   8   6   0   2  16   7   4]
 [  1   0  13 964   0  15   0   9   6   2]
 [  1   2   2   1 927   0  13   0   2  34]]
```

```
[ 8  2  1 14  0 849  7  2  5  4]
[ 6  3  2  0 10  7 927  0  3  0]
[ 2  7 21  9  2  1  0 960  1 25]
[12  2  4 17  8  7  8  7 894 15]
[ 9  6  0  8 22  8  0  8  4 944]]
```

Precision: 0.9539

Recall: 0.9538



Training Model with sigmoid activation, 3 conv_layers, 3 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 8s - loss: 2.3103 - accuracy: 0.1062 - val_loss: 2.3128 - val_accuracy: 0.0978 - 8s/epoch - 9ms/step

Epoch 2/5

844/844 - 7s - loss: 2.3067 - accuracy: 0.1064 - val_loss: 2.3076 - val_accuracy: 0.1113 - 7s/epoch - 8ms/step

Epoch 3/5

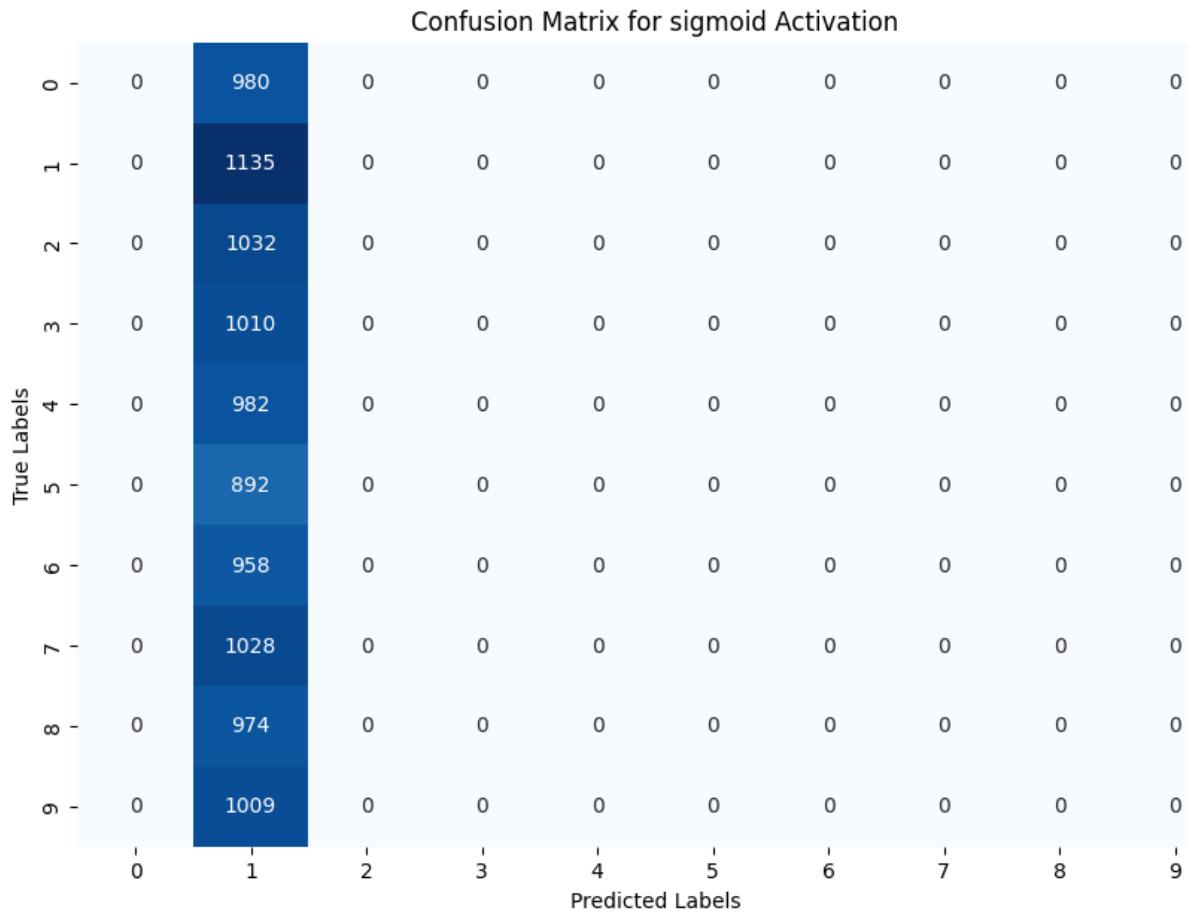
844/844 - 7s - loss: 2.3063 - accuracy: 0.1049 - val_loss: 2.3036 - val_accuracy: 0.0995 - 7s/epoch - 8ms/step

Epoch 4/5

844/844 - 7s - loss: 2.3056 - accuracy: 0.1073 - val_loss: 2.3058 -

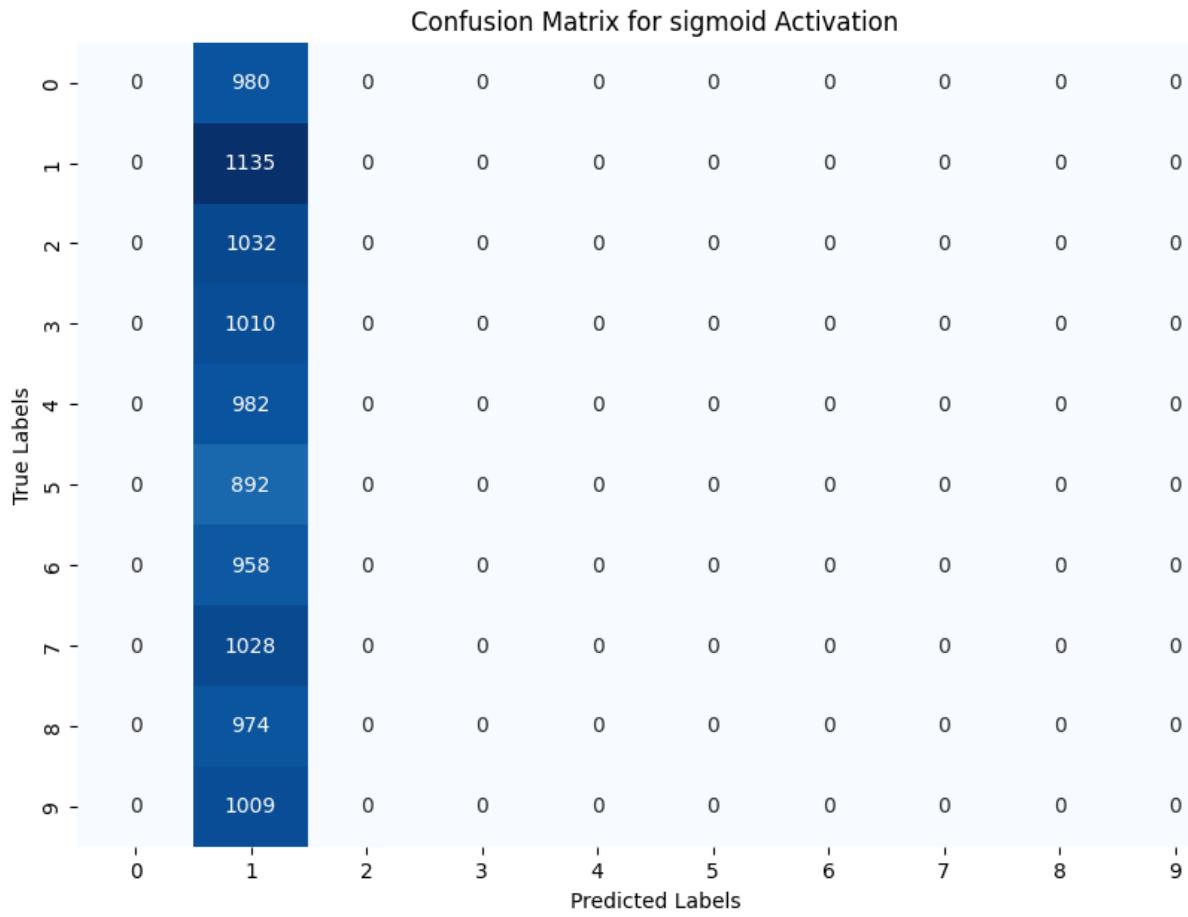
```
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 5/5
844/844 - 7s - loss: 2.3048 - accuracy: 0.1077 - val_loss: 2.3039 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 8s - loss: 2.3108 - accuracy: 0.1036 - val_loss: 2.3028 -
val_accuracy: 0.1045 - 8s/epoch - 9ms/step
Epoch 2/15
844/844 - 7s - loss: 2.3070 - accuracy: 0.1064 - val_loss: 2.3039 -
val_accuracy: 0.1113 - 7s/epoch - 8ms/step
Epoch 3/15
844/844 - 6s - loss: 2.3064 - accuracy: 0.1049 - val_loss: 2.3071 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 4/15
844/844 - 6s - loss: 2.3055 - accuracy: 0.1064 - val_loss: 2.3056 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 5/15
844/844 - 7s - loss: 2.3050 - accuracy: 0.1069 - val_loss: 2.3049 -
val_accuracy: 0.1000 - 7s/epoch - 8ms/step
Epoch 6/15
844/844 - 7s - loss: 2.3047 - accuracy: 0.1080 - val_loss: 2.3034 -
val_accuracy: 0.1113 - 7s/epoch - 8ms/step
Epoch 7/15
```

```
844/844 - 7s - loss: 2.3041 - accuracy: 0.1081 - val_loss: 2.3042 -  
val_accuracy: 0.1045 - 7s/epoch - 8ms/step  
Epoch 8/15  
844/844 - 7s - loss: 2.3036 - accuracy: 0.1095 - val_loss: 2.3033 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 9/15  
844/844 - 7s - loss: 2.3035 - accuracy: 0.1080 - val_loss: 2.3030 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 10/15  
844/844 - 7s - loss: 2.3027 - accuracy: 0.1106 - val_loss: 2.3049 -  
val_accuracy: 0.0960 - 7s/epoch - 8ms/step  
Epoch 11/15  
844/844 - 7s - loss: 2.3026 - accuracy: 0.1110 - val_loss: 2.3053 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 12/15  
844/844 - 7s - loss: 2.3023 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.0960 - 7s/epoch - 8ms/step  
Epoch 13/15  
844/844 - 7s - loss: 2.3016 - accuracy: 0.1124 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 14/15  
844/844 - 7s - loss: 2.3011 - accuracy: 0.1129 - val_loss: 2.3008 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 15/15  
844/844 - 7s - loss: 2.2999 - accuracy: 0.1194 - val_loss: 2.2998 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



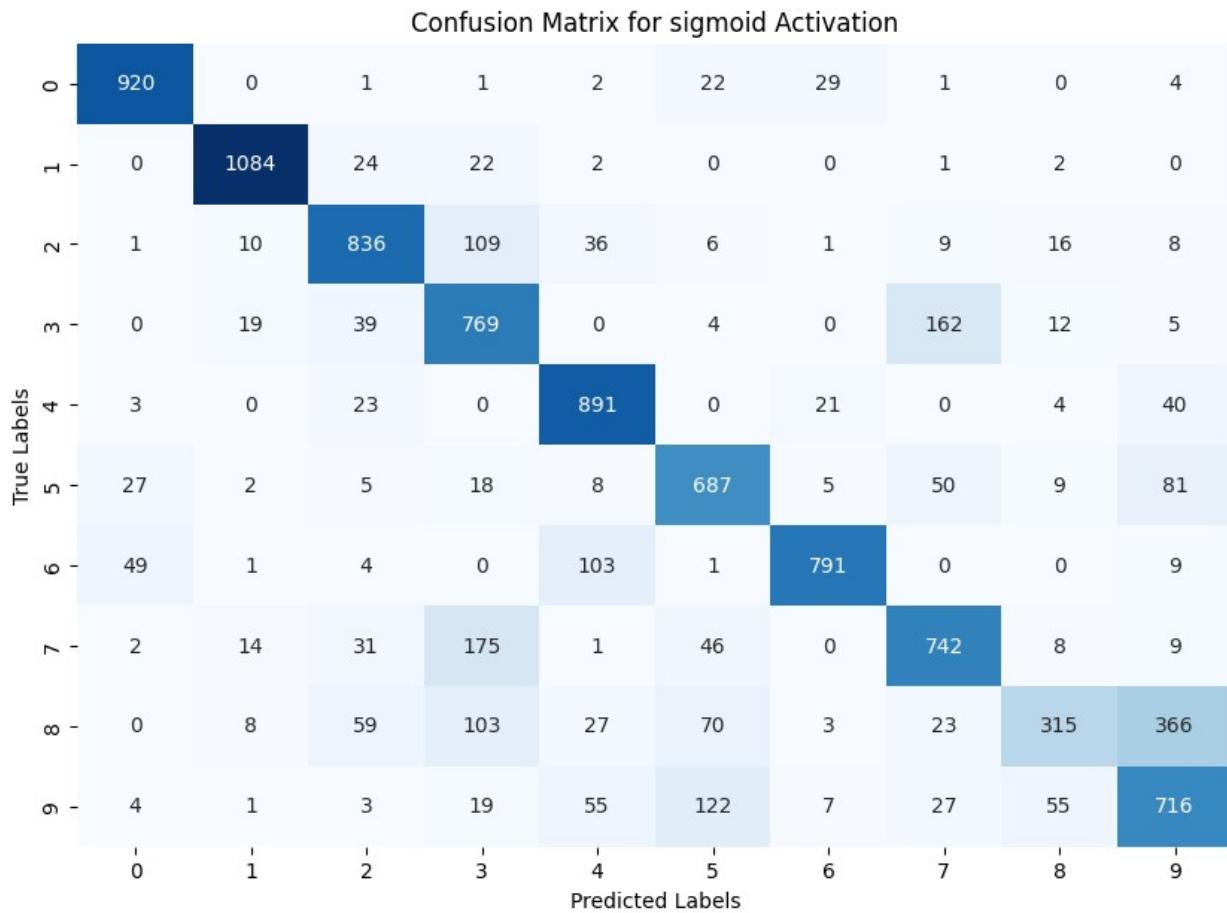
```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 8s - loss: 2.3087 - accuracy: 0.1020 - val_loss: 2.3084 -
val_accuracy: 0.0952 - 8s/epoch - 9ms/step
Epoch 2/20
844/844 - 6s - loss: 2.3061 - accuracy: 0.1054 - val_loss: 2.3030 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 3/20
844/844 - 7s - loss: 2.3057 - accuracy: 0.1056 - val_loss: 2.3083 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 4/20
844/844 - 6s - loss: 2.3052 - accuracy: 0.1051 - val_loss: 2.3043 -
val_accuracy: 0.0960 - 6s/epoch - 8ms/step
Epoch 5/20
844/844 - 7s - loss: 2.3042 - accuracy: 0.1069 - val_loss: 2.3041 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 6/20
844/844 - 6s - loss: 2.3039 - accuracy: 0.1072 - val_loss: 2.3053 -
val_accuracy: 0.0960 - 6s/epoch - 8ms/step
Epoch 7/20
```

```
844/844 - 7s - loss: 2.3035 - accuracy: 0.1079 - val_loss: 2.3030 -  
val_accuracy: 0.0995 - 7s/epoch - 8ms/step  
Epoch 8/20  
844/844 - 6s - loss: 2.3037 - accuracy: 0.1083 - val_loss: 2.3044 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 9/20  
844/844 - 6s - loss: 2.3031 - accuracy: 0.1104 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 10/20  
844/844 - 7s - loss: 2.3027 - accuracy: 0.1110 - val_loss: 2.3035 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 11/20  
844/844 - 7s - loss: 2.3025 - accuracy: 0.1099 - val_loss: 2.3051 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 12/20  
844/844 - 6s - loss: 2.3022 - accuracy: 0.1111 - val_loss: 2.3013 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 13/20  
844/844 - 6s - loss: 2.3017 - accuracy: 0.1119 - val_loss: 2.3008 -  
val_accuracy: 0.1050 - 6s/epoch - 7ms/step  
Epoch 14/20  
844/844 - 6s - loss: 2.3008 - accuracy: 0.1157 - val_loss: 2.2997 -  
val_accuracy: 0.1247 - 6s/epoch - 8ms/step  
Epoch 15/20  
844/844 - 6s - loss: 2.2999 - accuracy: 0.1172 - val_loss: 2.2984 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 16/20  
844/844 - 6s - loss: 2.2960 - accuracy: 0.1297 - val_loss: 2.2925 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 17/20  
844/844 - 6s - loss: 2.2453 - accuracy: 0.1778 - val_loss: 2.0734 -  
val_accuracy: 0.2357 - 6s/epoch - 8ms/step  
Epoch 18/20  
844/844 - 6s - loss: 1.8224 - accuracy: 0.2955 - val_loss: 1.6200 -  
val_accuracy: 0.3417 - 6s/epoch - 8ms/step  
Epoch 19/20  
844/844 - 6s - loss: 1.4630 - accuracy: 0.4723 - val_loss: 1.1674 -  
val_accuracy: 0.6172 - 6s/epoch - 8ms/step  
Epoch 20/20  
844/844 - 6s - loss: 0.9747 - accuracy: 0.6877 - val_loss: 0.7613 -  
val_accuracy: 0.7785 - 6s/epoch - 8ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 920 0 1 1 2 22 29 1 0 4]  
[ 0 1084 24 22 2 0 0 1 2 0]  
[ 1 10 836 109 36 6 1 9 16 8]  
[ 0 19 39 769 0 4 0 162 12 5]  
[ 3 0 23 0 891 0 21 0 4 40]]
```

```
[ 27  2  5  18  8  687  5  50  9  81]
[ 49  1  4  0 103  1 791  0  0  9]
[ 2  14 31 175  1  46  0 742  8  9]
[ 0  8 59 103 27  70  3 23 315 366]
[ 4  1  3 19  55 122  7 27 55 716]]
```

Precision: 0.7822

Recall: 0.7751



Training Model with sigmoid activation, 3 conv_layers, 3 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 6s - loss: 2.3073 - accuracy: 0.1054 - val_loss: 2.3031 - val_accuracy: 0.1050 - 6s/epoch - 13ms/step

Epoch 2/5

422/422 - 5s - loss: 2.3038 - accuracy: 0.1074 - val_loss: 2.3041 - val_accuracy: 0.1000 - 5s/epoch - 11ms/step

Epoch 3/5

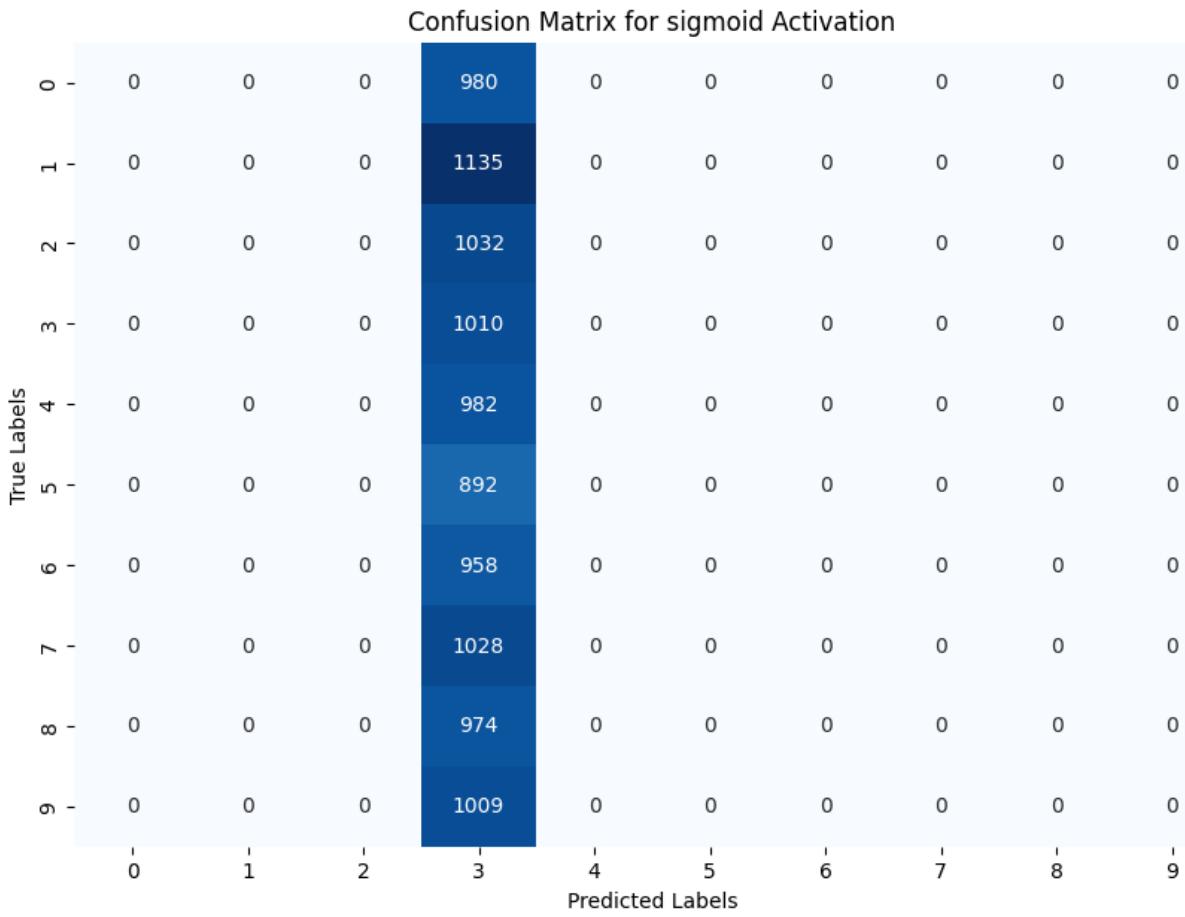
422/422 - 5s - loss: 2.3043 - accuracy: 0.1052 - val_loss: 2.3038 - val_accuracy: 0.1045 - 5s/epoch - 11ms/step

Epoch 4/5

422/422 - 5s - loss: 2.3040 - accuracy: 0.1074 - val_loss: 2.3059 -

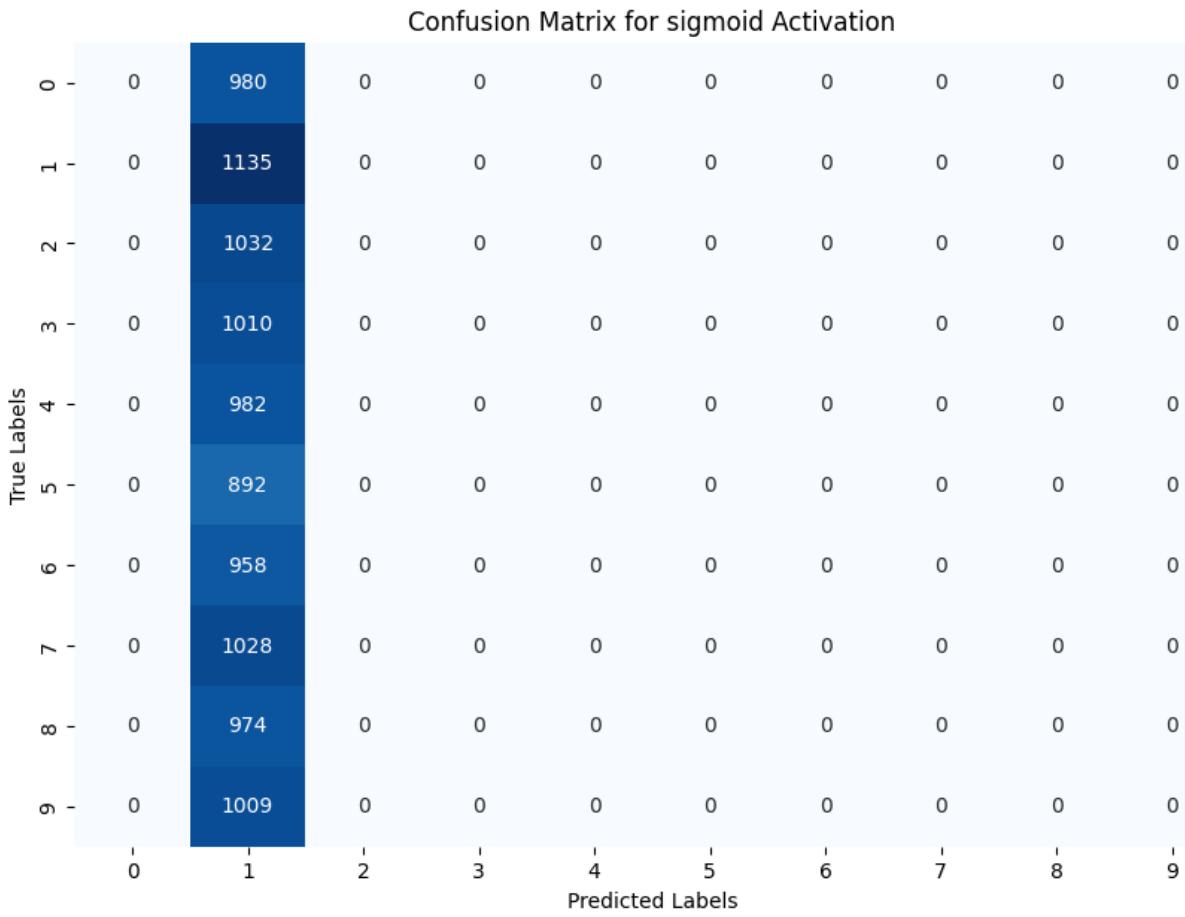
```
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 5/5
422/422 - 5s - loss: 2.3041 - accuracy: 0.1080 - val_loss: 2.3047 -
val_accuracy: 0.1045 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 0  0  0 980  0  0  0  0  0]
 [ 0  0  0 1135  0  0  0  0  0]
 [ 0  0  0 1032  0  0  0  0  0]
 [ 0  0  0 1010  0  0  0  0  0]
 [ 0  0  0 982  0  0  0  0  0]
 [ 0  0  0 892  0  0  0  0  0]
 [ 0  0  0 958  0  0  0  0  0]
 [ 0  0  0 1028  0  0  0  0  0]
 [ 0  0  0 974  0  0  0  0  0]
 [ 0  0  0 1009  0  0  0  0  0]]
Precision: 0.0102
Recall: 0.1010

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 6s - loss: 2.3074 - accuracy: 0.1069 - val_loss: 2.3105 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 2/15
422/422 - 5s - loss: 2.3044 - accuracy: 0.1069 - val_loss: 2.3071 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 3/15
422/422 - 5s - loss: 2.3037 - accuracy: 0.1071 - val_loss: 2.3082 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 4/15
422/422 - 5s - loss: 2.3038 - accuracy: 0.1084 - val_loss: 2.3112 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 5/15
422/422 - 5s - loss: 2.3038 - accuracy: 0.1086 - val_loss: 2.3028 -
val_accuracy: 0.1113 - 5s/epoch - 11ms/step
Epoch 6/15
422/422 - 5s - loss: 2.3034 - accuracy: 0.1082 - val_loss: 2.3053 -
val_accuracy: 0.1045 - 5s/epoch - 11ms/step
Epoch 7/15
```

```
422/422 - 5s - loss: 2.3039 - accuracy: 0.1081 - val_loss: 2.3040 -  
val_accuracy: 0.1113 - 5s/epoch - 11ms/step  
Epoch 8/15  
422/422 - 5s - loss: 2.3036 - accuracy: 0.1074 - val_loss: 2.3038 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 9/15  
422/422 - 5s - loss: 2.3035 - accuracy: 0.1082 - val_loss: 2.3037 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 10/15  
422/422 - 5s - loss: 2.3028 - accuracy: 0.1102 - val_loss: 2.3044 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 11/15  
422/422 - 5s - loss: 2.3030 - accuracy: 0.1081 - val_loss: 2.3069 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 12/15  
422/422 - 5s - loss: 2.3031 - accuracy: 0.1076 - val_loss: 2.3035 -  
val_accuracy: 0.1045 - 5s/epoch - 11ms/step  
Epoch 13/15  
422/422 - 5s - loss: 2.3025 - accuracy: 0.1112 - val_loss: 2.3036 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 14/15  
422/422 - 5s - loss: 2.3028 - accuracy: 0.1099 - val_loss: 2.3029 -  
val_accuracy: 0.0952 - 5s/epoch - 11ms/step  
Epoch 15/15  
422/422 - 5s - loss: 2.3026 - accuracy: 0.1107 - val_loss: 2.3032 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```

Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 6s - loss: 2.3082 - accuracy: 0.1064 - val_loss: 2.3052 -
val_accuracy: 0.1113 - 6s/epoch - 13ms/step
Epoch 2/20
422/422 - 5s - loss: 2.3041 - accuracy: 0.1078 - val_loss: 2.3029 -
val_accuracy: 0.1045 - 5s/epoch - 11ms/step
Epoch 3/20
422/422 - 5s - loss: 2.3043 - accuracy: 0.1061 - val_loss: 2.3051 -
val_accuracy: 0.1113 - 5s/epoch - 11ms/step
Epoch 4/20
422/422 - 5s - loss: 2.3042 - accuracy: 0.1064 - val_loss: 2.3037 -
val_accuracy: 0.0978 - 5s/epoch - 11ms/step
Epoch 5/20
422/422 - 5s - loss: 2.3039 - accuracy: 0.1065 - val_loss: 2.3056 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 6/20
422/422 - 5s - loss: 2.3040 - accuracy: 0.1051 - val_loss: 2.3030 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 7/20

```

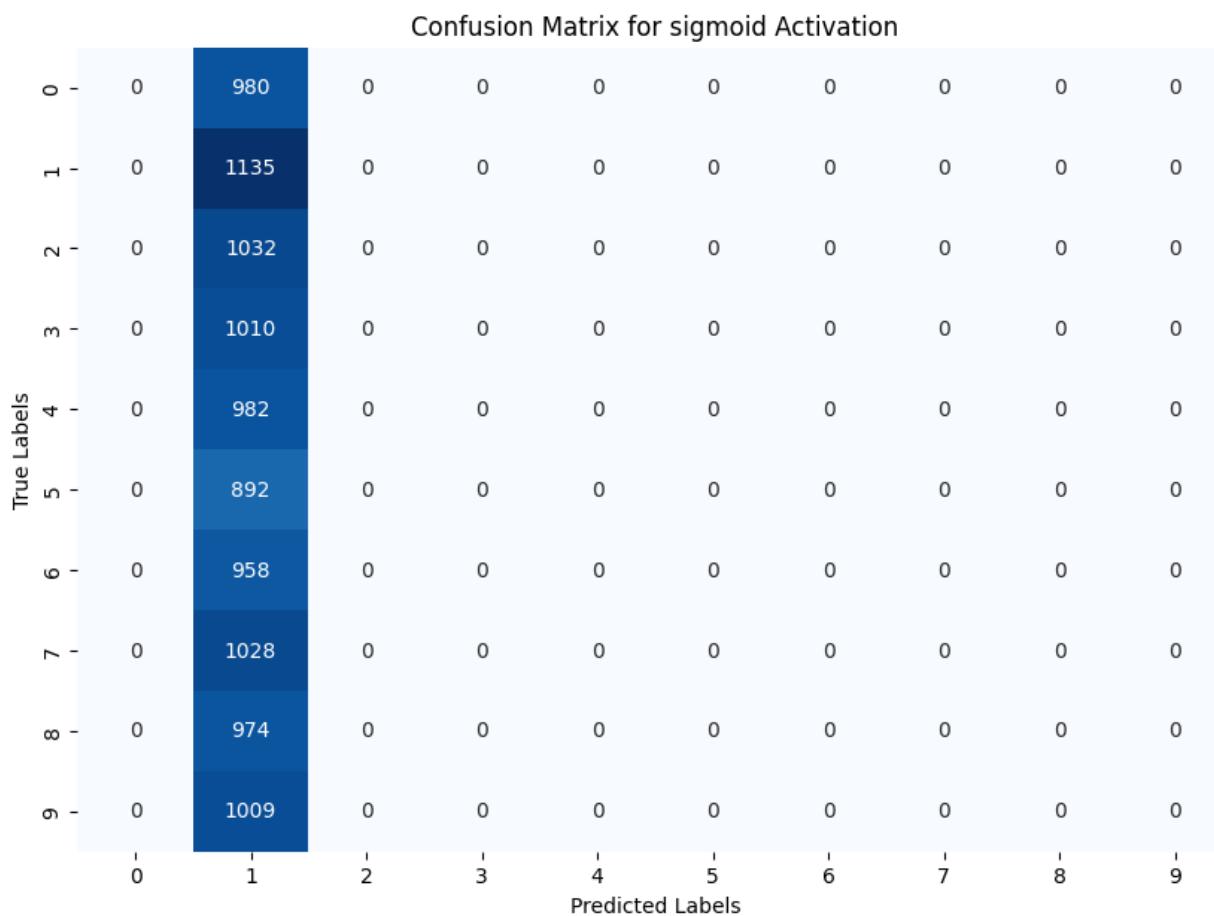
```
422/422 - 5s - loss: 2.3037 - accuracy: 0.1068 - val_loss: 2.3025 -  
val_accuracy: 0.0992 - 5s/epoch - 12ms/step  
Epoch 8/20  
422/422 - 5s - loss: 2.3038 - accuracy: 0.1072 - val_loss: 2.3024 -  
val_accuracy: 0.0952 - 5s/epoch - 11ms/step  
Epoch 9/20  
422/422 - 5s - loss: 2.3035 - accuracy: 0.1071 - val_loss: 2.3020 -  
val_accuracy: 0.1113 - 5s/epoch - 11ms/step  
Epoch 10/20  
422/422 - 5s - loss: 2.3043 - accuracy: 0.1084 - val_loss: 2.3032 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 11/20  
422/422 - 5s - loss: 2.3034 - accuracy: 0.1091 - val_loss: 2.3023 -  
val_accuracy: 0.0995 - 5s/epoch - 11ms/step  
Epoch 12/20  
422/422 - 5s - loss: 2.3032 - accuracy: 0.1081 - val_loss: 2.3028 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 13/20  
422/422 - 5s - loss: 2.3038 - accuracy: 0.1077 - val_loss: 2.3038 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 14/20  
422/422 - 5s - loss: 2.3027 - accuracy: 0.1097 - val_loss: 2.3038 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 15/20  
422/422 - 5s - loss: 2.3027 - accuracy: 0.1089 - val_loss: 2.3044 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 16/20  
422/422 - 5s - loss: 2.3027 - accuracy: 0.1096 - val_loss: 2.3040 -  
val_accuracy: 0.0952 - 5s/epoch - 11ms/step  
Epoch 17/20  
422/422 - 5s - loss: 2.3028 - accuracy: 0.1091 - val_loss: 2.3051 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 18/20  
422/422 - 5s - loss: 2.3025 - accuracy: 0.1112 - val_loss: 2.3038 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 19/20  
422/422 - 5s - loss: 2.3023 - accuracy: 0.1109 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 20/20  
422/422 - 5s - loss: 2.3021 - accuracy: 0.1122 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 5s - loss: 2.3081 - accuracy: 0.1100 - val_loss: 2.3041 -
val_accuracy: 0.0992 - 5s/epoch - 26ms/step
Epoch 2/5
211/211 - 4s - loss: 2.3035 - accuracy: 0.1066 - val_loss: 2.3036 -
```

```
val_accuracy: 0.1113 - 4s/epoch - 19ms/step
Epoch 3/5
211/211 - 4s - loss: 2.3029 - accuracy: 0.1085 - val_loss: 2.3028 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 4/5
211/211 - 4s - loss: 2.3027 - accuracy: 0.1099 - val_loss: 2.3043 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 5/5
211/211 - 4s - loss: 2.3025 - accuracy: 0.1107 - val_loss: 2.3040 -
val_accuracy: 0.1000 - 4s/epoch - 19ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 0  0 980  0  0  0  0  0  0  0]
 [ 0  0 1135  0  0  0  0  0  0  0]
 [ 0  0 1032  0  0  0  0  0  0  0]
 [ 0  0 1010  0  0  0  0  0  0  0]
 [ 0  0 982  0  0  0  0  0  0  0]
 [ 0  0 892  0  0  0  0  0  0  0]
 [ 0  0 958  0  0  0  0  0  0  0]
 [ 0  0 1028  0  0  0  0  0  0  0]
 [ 0  0 974  0  0  0  0  0  0  0]
 [ 0  0 1009  0  0  0  0  0  0  0]]]
Precision: 0.0107
Recall: 0.1032

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for sigmoid Activation										
	0 -	1 -	2 -	3 -	4 -	5 -	6 -	7 -	8 -	9 -
True Labels	0	0	0	980	0	0	0	0	0	0
0 -	0	0	0	980	0	0	0	0	0	0
1 -	0	0	0	1135	0	0	0	0	0	0
2 -	0	0	0	1032	0	0	0	0	0	0
3 -	0	0	0	1010	0	0	0	0	0	0
4 -	0	0	0	982	0	0	0	0	0	0
5 -	0	0	0	892	0	0	0	0	0	0
6 -	0	0	0	958	0	0	0	0	0	0
7 -	0	0	0	1028	0	0	0	0	0	0
8 -	0	0	0	974	0	0	0	0	0	0
9 -	0	0	0	1009	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9
	Predicted Labels									

```

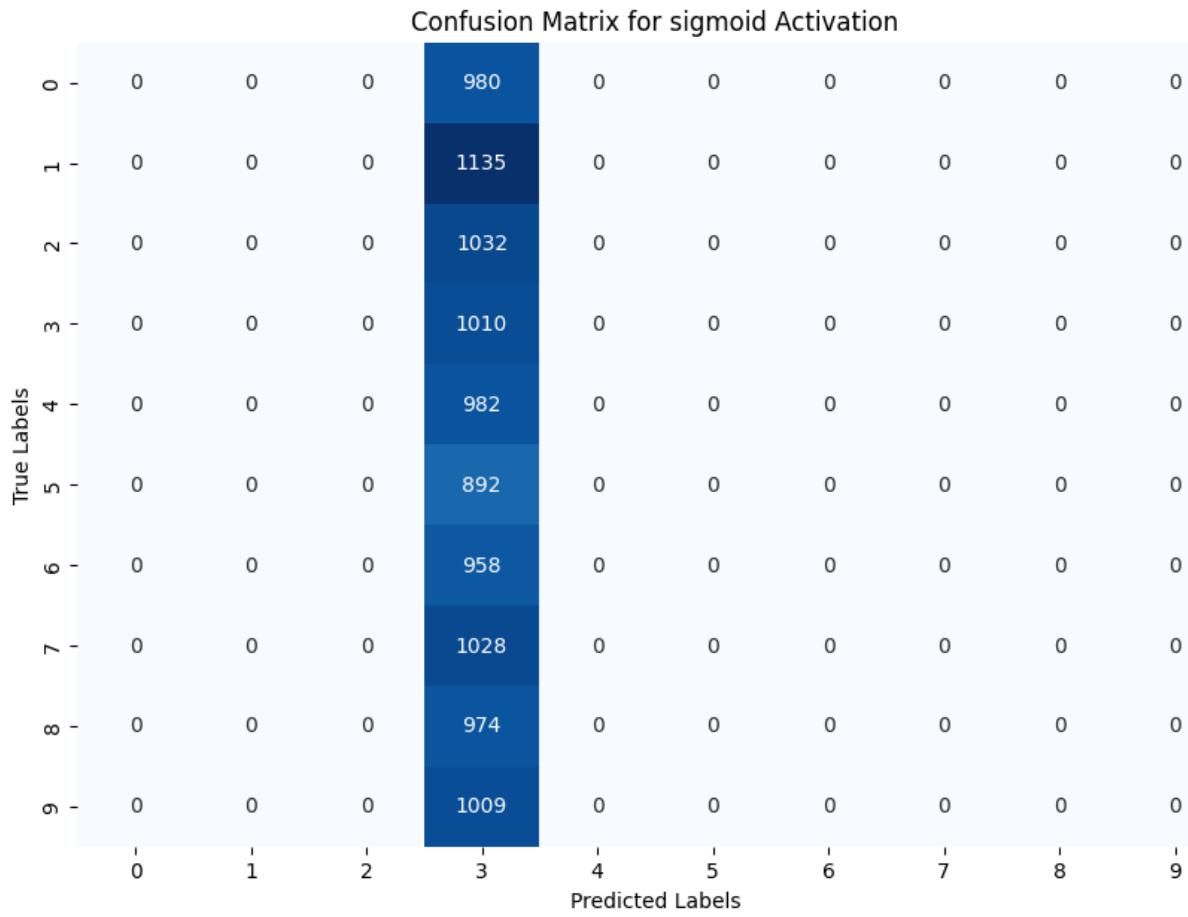
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 5s - loss: 2.3071 - accuracy: 0.1077 - val_loss: 2.3057 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 2/15
211/211 - 4s - loss: 2.3026 - accuracy: 0.1116 - val_loss: 2.3037 -
val_accuracy: 0.1000 - 4s/epoch - 18ms/step
Epoch 3/15
211/211 - 4s - loss: 2.3026 - accuracy: 0.1096 - val_loss: 2.3026 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 4/15
211/211 - 4s - loss: 2.3026 - accuracy: 0.1097 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 5/15
211/211 - 4s - loss: 2.3026 - accuracy: 0.1094 - val_loss: 2.3029 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 6/15
211/211 - 4s - loss: 2.3025 - accuracy: 0.1098 - val_loss: 2.3031 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 7/15

```

```
211/211 - 4s - loss: 2.3026 - accuracy: 0.1095 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 8/15  
211/211 - 4s - loss: 2.3025 - accuracy: 0.1116 - val_loss: 2.3041 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 9/15  
211/211 - 4s - loss: 2.3023 - accuracy: 0.1105 - val_loss: 2.3032 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 10/15  
211/211 - 4s - loss: 2.3024 - accuracy: 0.1119 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 11/15  
211/211 - 4s - loss: 2.3029 - accuracy: 0.1079 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 12/15  
211/211 - 4s - loss: 2.3022 - accuracy: 0.1105 - val_loss: 2.3035 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 13/15  
211/211 - 4s - loss: 2.3026 - accuracy: 0.1110 - val_loss: 2.3039 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 14/15  
211/211 - 4s - loss: 2.3019 - accuracy: 0.1125 - val_loss: 2.3050 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 15/15  
211/211 - 4s - loss: 2.3020 - accuracy: 0.1107 - val_loss: 2.3024 -  
val_accuracy: 0.1045 - 4s/epoch - 18ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0  0  0  980  0  0  0  0  0  0]  
[ 0  0  0 1135  0  0  0  0  0  0]  
[ 0  0  0 1032  0  0  0  0  0  0]  
[ 0  0  0 1010  0  0  0  0  0  0]  
[ 0  0  0 982  0  0  0  0  0  0]  
[ 0  0  0 892  0  0  0  0  0  0]  
[ 0  0  0 958  0  0  0  0  0  0]  
[ 0  0  0 1028  0  0  0  0  0  0]  
[ 0  0  0 974  0  0  0  0  0  0]  
[ 0  0  0 1009  0  0  0  0  0  0]]  
Precision: 0.0102  
Recall: 0.1010  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```



```

Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 5s - loss: 2.3096 - accuracy: 0.1100 - val_loss: 2.3041 -
val_accuracy: 0.1045 - 5s/epoch - 23ms/step
Epoch 2/20
211/211 - 4s - loss: 2.3033 - accuracy: 0.1060 - val_loss: 2.3027 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 3/20
211/211 - 4s - loss: 2.3025 - accuracy: 0.1100 - val_loss: 2.3081 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 4/20
211/211 - 4s - loss: 2.3029 - accuracy: 0.1106 - val_loss: 2.3072 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 5/20
211/211 - 4s - loss: 2.3031 - accuracy: 0.1099 - val_loss: 2.3023 -
val_accuracy: 0.0952 - 4s/epoch - 18ms/step
Epoch 6/20
211/211 - 4s - loss: 2.3026 - accuracy: 0.1092 - val_loss: 2.3037 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 7/20

```

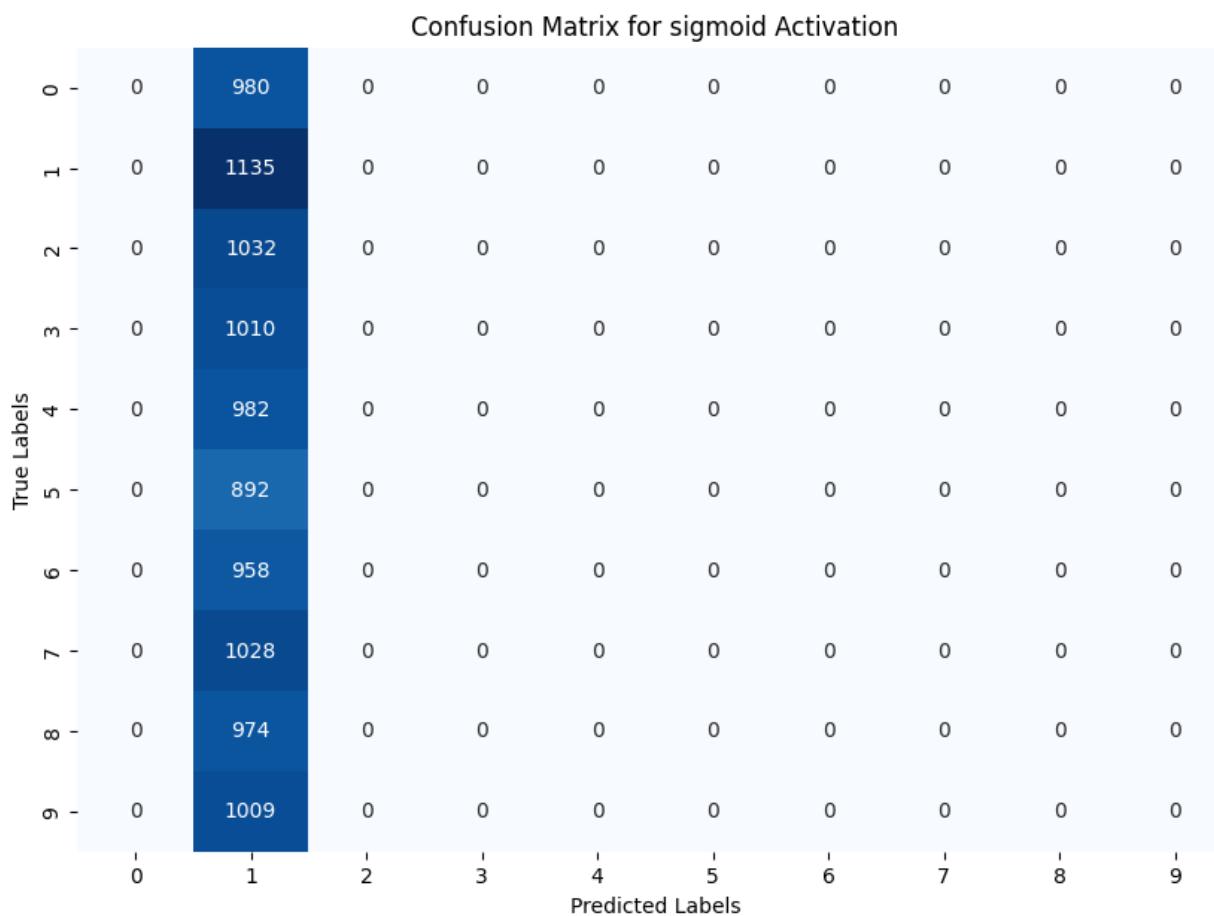
```
211/211 - 4s - loss: 2.3030 - accuracy: 0.1096 - val_loss: 2.3028 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 8/20  
211/211 - 4s - loss: 2.3030 - accuracy: 0.1100 - val_loss: 2.3030 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 9/20  
211/211 - 4s - loss: 2.3028 - accuracy: 0.1101 - val_loss: 2.3061 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 10/20  
211/211 - 4s - loss: 2.3027 - accuracy: 0.1111 - val_loss: 2.3035 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 11/20  
211/211 - 4s - loss: 2.3028 - accuracy: 0.1096 - val_loss: 2.3031 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 12/20  
211/211 - 4s - loss: 2.3026 - accuracy: 0.1108 - val_loss: 2.3028 -  
val_accuracy: 0.1113 - 4s/epoch - 18ms/step  
Epoch 13/20  
211/211 - 4s - loss: 2.3030 - accuracy: 0.1066 - val_loss: 2.3054 -  
val_accuracy: 0.0960 - 4s/epoch - 18ms/step  
Epoch 14/20  
211/211 - 4s - loss: 2.3026 - accuracy: 0.1094 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 15/20  
211/211 - 4s - loss: 2.3027 - accuracy: 0.1093 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 16/20  
211/211 - 4s - loss: 2.3024 - accuracy: 0.1109 - val_loss: 2.3039 -  
val_accuracy: 0.1113 - 4s/epoch - 18ms/step  
Epoch 17/20  
211/211 - 4s - loss: 2.3028 - accuracy: 0.1083 - val_loss: 2.3025 -  
val_accuracy: 0.1113 - 4s/epoch - 18ms/step  
Epoch 18/20  
211/211 - 4s - loss: 2.3025 - accuracy: 0.1103 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 19/20  
211/211 - 4s - loss: 2.3026 - accuracy: 0.1102 - val_loss: 2.3017 -  
val_accuracy: 0.1113 - 4s/epoch - 18ms/step  
Epoch 20/20  
211/211 - 4s - loss: 2.3024 - accuracy: 0.1103 - val_loss: 2.3030 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

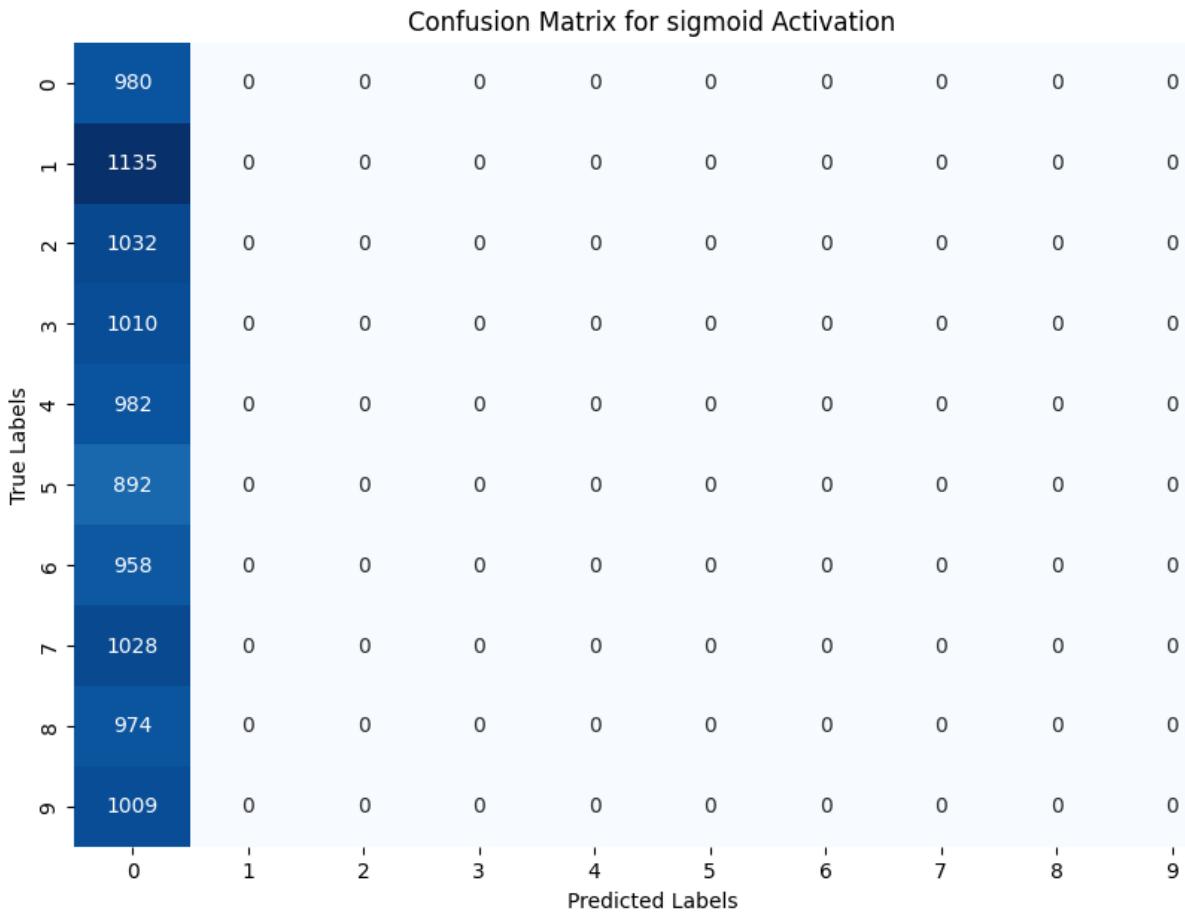
844/844 - 8s - loss: 2.3133 - accuracy: 0.1042 - val_loss: 2.3149 -
val_accuracy: 0.1113 - 8s/epoch - 9ms/step

Epoch 2/5

844/844 - 7s - loss: 2.3108 - accuracy: 0.1027 - val_loss: 2.3079 -

```
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 3/5
844/844 - 7s - loss: 2.3085 - accuracy: 0.1065 - val_loss: 2.3092 -
val_accuracy: 0.0978 - 7s/epoch - 8ms/step
Epoch 4/5
844/844 - 7s - loss: 2.3071 - accuracy: 0.1052 - val_loss: 2.3115 -
val_accuracy: 0.1045 - 7s/epoch - 8ms/step
Epoch 5/5
844/844 - 7s - loss: 2.3057 - accuracy: 0.1058 - val_loss: 2.3021 -
val_accuracy: 0.0978 - 7s/epoch - 8ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 980  0  0  0  0  0  0  0  0  0]
 [1135  0  0  0  0  0  0  0  0  0]
 [1032  0  0  0  0  0  0  0  0  0]
 [1010  0  0  0  0  0  0  0  0  0]
 [ 982  0  0  0  0  0  0  0  0  0]
 [ 892  0  0  0  0  0  0  0  0  0]
 [ 958  0  0  0  0  0  0  0  0  0]
 [1028  0  0  0  0  0  0  0  0  0]
 [ 974  0  0  0  0  0  0  0  0  0]
 [1009  0  0  0  0  0  0  0  0  0]]
Precision: 0.0096
Recall: 0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```

Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 8s - loss: 2.3139 - accuracy: 0.1021 - val_loss: 2.3156 -
val_accuracy: 0.0978 - 8s/epoch - 9ms/step
Epoch 2/15
844/844 - 7s - loss: 2.3101 - accuracy: 0.1039 - val_loss: 2.3043 -
val_accuracy: 0.1000 - 7s/epoch - 8ms/step
Epoch 3/15
844/844 - 7s - loss: 2.3088 - accuracy: 0.1022 - val_loss: 2.3141 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 4/15
844/844 - 7s - loss: 2.3082 - accuracy: 0.1042 - val_loss: 2.3045 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 5/15
844/844 - 7s - loss: 2.3060 - accuracy: 0.1079 - val_loss: 2.3059 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 6/15
844/844 - 7s - loss: 2.3051 - accuracy: 0.1059 - val_loss: 2.3035 -
val_accuracy: 0.0960 - 7s/epoch - 8ms/step
Epoch 7/15

```

```
844/844 - 7s - loss: 2.3043 - accuracy: 0.1087 - val_loss: 2.3049 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 8/15  
844/844 - 7s - loss: 2.3031 - accuracy: 0.1098 - val_loss: 2.3017 -  
val_accuracy: 0.1113 - 7s/epoch - 8ms/step  
Epoch 9/15  
844/844 - 7s - loss: 2.3024 - accuracy: 0.1113 - val_loss: 2.3099 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 10/15  
844/844 - 7s - loss: 2.3022 - accuracy: 0.1113 - val_loss: 2.3030 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 11/15  
844/844 - 7s - loss: 2.3016 - accuracy: 0.1171 - val_loss: 2.3004 -  
val_accuracy: 0.1557 - 7s/epoch - 8ms/step  
Epoch 12/15  
844/844 - 7s - loss: 2.2996 - accuracy: 0.1165 - val_loss: 2.2983 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 13/15  
844/844 - 7s - loss: 2.2959 - accuracy: 0.1310 - val_loss: 2.2895 -  
val_accuracy: 0.1715 - 7s/epoch - 8ms/step  
Epoch 14/15  
844/844 - 7s - loss: 2.2368 - accuracy: 0.1833 - val_loss: 2.0599 -  
val_accuracy: 0.2118 - 7s/epoch - 8ms/step  
Epoch 15/15  
844/844 - 7s - loss: 1.8126 - accuracy: 0.3292 - val_loss: 1.5848 -  
val_accuracy: 0.4130 - 7s/epoch - 8ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 82  0  0  30  2  360  0  306  90  110]  
[ 0 1065  0  64  0  0  0  6  0  0]  
[ 0 283  0  578  7  27  0  104  6  27]  
[ 0 111  0  688  0  14  0  189  1  7]  
[ 0  0  0  2  657  17  0  10  2  294]  
[ 0  26  0  101  39  190  0  223  17  296]  
[ 1  3  0  16  36  237  0  40  34  591]  
[ 0  39  0  239  0  68  0  544  0  138]  
[ 0  20  0  177  8  215  0  334  33  187]  
[ 0  2  0  13  114  48  0  61  0  771]]  
Precision: 0.3809  
Recall: 0.4030  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9
0	82	0	0	30	2	360	0	306	90	110
1	0	1065	0	64	0	0	0	6	0	0
2	0	283	0	578	7	27	0	104	6	27
3	0	111	0	688	0	14	0	189	1	7
4	0	0	0	2	657	17	0	10	2	294
5	0	26	0	101	39	190	0	223	17	296
6	1	3	0	16	36	237	0	40	34	591
7	0	39	0	239	0	68	0	544	0	138
8	0	20	0	177	8	215	0	334	33	187
9	0	2	0	13	114	48	0	61	0	771
	0	1	2	3	4	5	6	7	8	9
	True Labels									Predicted Labels

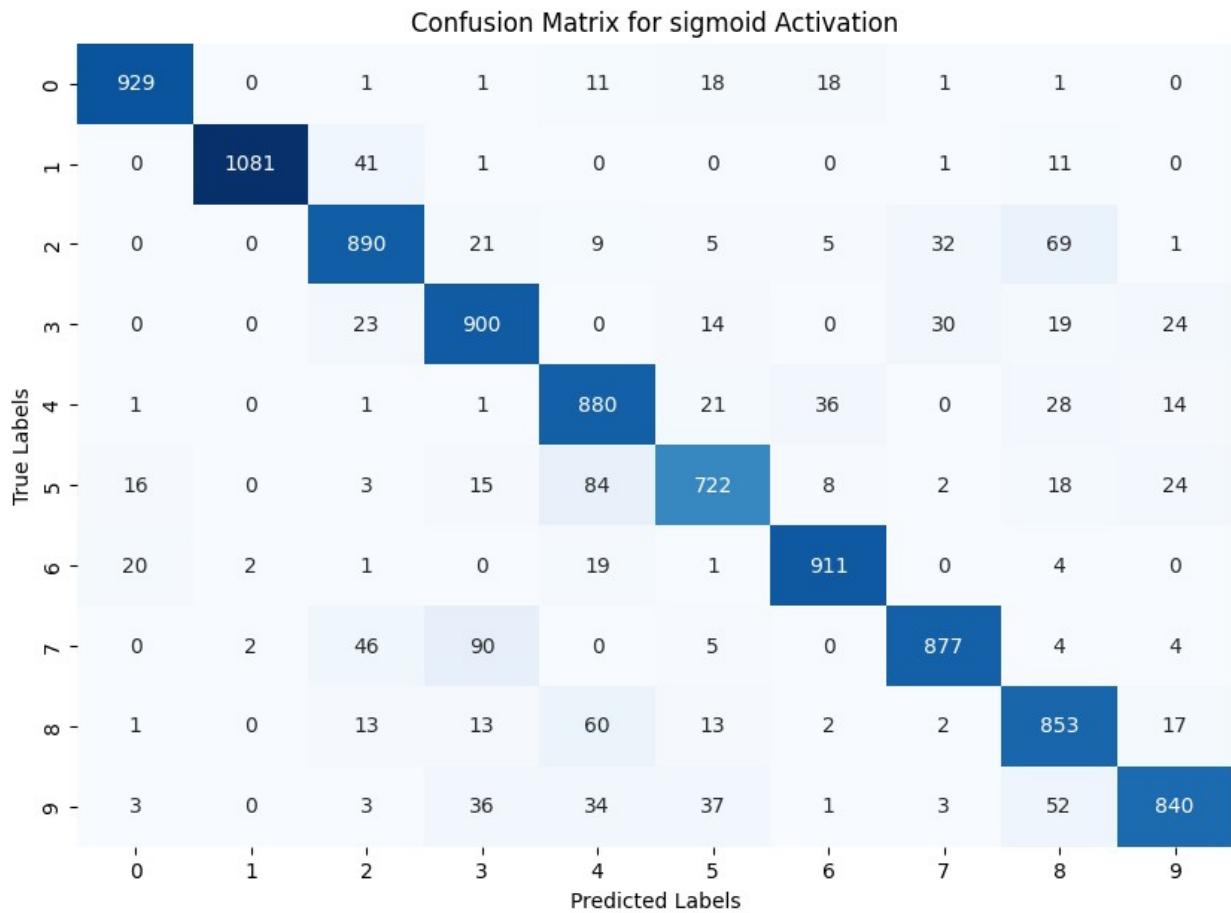
```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 8s - loss: 2.3147 - accuracy: 0.1030 - val_loss: 2.3114 -
val_accuracy: 0.1000 - 8s/epoch - 9ms/step
Epoch 2/20
844/844 - 7s - loss: 2.3105 - accuracy: 0.1059 - val_loss: 2.3078 -
val_accuracy: 0.0960 - 7s/epoch - 8ms/step
Epoch 3/20
844/844 - 7s - loss: 2.3089 - accuracy: 0.1051 - val_loss: 2.3061 -
val_accuracy: 0.1113 - 7s/epoch - 8ms/step
Epoch 4/20
844/844 - 7s - loss: 2.3075 - accuracy: 0.1045 - val_loss: 2.3079 -
val_accuracy: 0.0960 - 7s/epoch - 8ms/step
Epoch 5/20
844/844 - 7s - loss: 2.3062 - accuracy: 0.1053 - val_loss: 2.3098 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 6/20
844/844 - 7s - loss: 2.3052 - accuracy: 0.1084 - val_loss: 2.3044 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 7/20
```

```
844/844 - 7s - loss: 2.3039 - accuracy: 0.1095 - val_loss: 2.3045 -  
val_accuracy: 0.1968 - 7s/epoch - 8ms/step  
Epoch 8/20  
844/844 - 7s - loss: 2.3035 - accuracy: 0.1061 - val_loss: 2.3027 -  
val_accuracy: 0.1052 - 7s/epoch - 8ms/step  
Epoch 9/20  
844/844 - 7s - loss: 2.3030 - accuracy: 0.1127 - val_loss: 2.3028 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 10/20  
844/844 - 7s - loss: 2.3023 - accuracy: 0.1110 - val_loss: 2.3042 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 11/20  
844/844 - 7s - loss: 2.3018 - accuracy: 0.1141 - val_loss: 2.3009 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 12/20  
844/844 - 7s - loss: 2.3008 - accuracy: 0.1180 - val_loss: 2.3012 -  
val_accuracy: 0.1068 - 7s/epoch - 8ms/step  
Epoch 13/20  
844/844 - 7s - loss: 2.2993 - accuracy: 0.1242 - val_loss: 2.2988 -  
val_accuracy: 0.1045 - 7s/epoch - 8ms/step  
Epoch 14/20  
844/844 - 7s - loss: 2.2929 - accuracy: 0.1374 - val_loss: 2.2824 -  
val_accuracy: 0.2265 - 7s/epoch - 8ms/step  
Epoch 15/20  
844/844 - 7s - loss: 2.1504 - accuracy: 0.2070 - val_loss: 1.9136 -  
val_accuracy: 0.2565 - 7s/epoch - 8ms/step  
Epoch 16/20  
844/844 - 7s - loss: 1.7239 - accuracy: 0.3222 - val_loss: 1.5253 -  
val_accuracy: 0.3890 - 7s/epoch - 8ms/step  
Epoch 17/20  
844/844 - 7s - loss: 1.4219 - accuracy: 0.4300 - val_loss: 1.2757 -  
val_accuracy: 0.4852 - 7s/epoch - 8ms/step  
Epoch 18/20  
844/844 - 7s - loss: 1.2202 - accuracy: 0.5270 - val_loss: 1.1323 -  
val_accuracy: 0.5708 - 7s/epoch - 8ms/step  
Epoch 19/20  
844/844 - 7s - loss: 0.9490 - accuracy: 0.6843 - val_loss: 0.7079 -  
val_accuracy: 0.8027 - 7s/epoch - 8ms/step  
Epoch 20/20  
844/844 - 7s - loss: 0.6201 - accuracy: 0.8292 - val_loss: 0.4498 -  
val_accuracy: 0.8952 - 7s/epoch - 8ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 929   0   1   1  11   18   18   1   1   0]  
 [  0 1081   41   1   0   0   0   1   11   0]  
 [  0   0 890   21   9   5   5  32   69   1]  
 [  0   0   23 900   0  14   0  30   19  24]  
 [  1   0   1   1  880   21   36   0  28  14]]
```

```
[ 16   0   3  15   84  722    8   2   18   24]
[ 20   2   1   0   19   1  911    0   4   0]
[ 0   2   46  90   0   5   0   877   4   4]
[ 1   0  13  13   60  13    2   2  853  17]
[ 3   0   3  36   34  37    1   3   52  840]]
```

Precision: 0.8912

Recall: 0.8883



Training Model with sigmoid activation, 3 conv_layers, 3 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 6s - loss: 2.3112 - accuracy: 0.1042 - val_loss: 2.3038 - val_accuracy: 0.0952 - 6s/epoch - 14ms/step

Epoch 2/5

422/422 - 5s - loss: 2.3080 - accuracy: 0.1040 - val_loss: 2.3069 - val_accuracy: 0.1050 - 5s/epoch - 11ms/step

Epoch 3/5

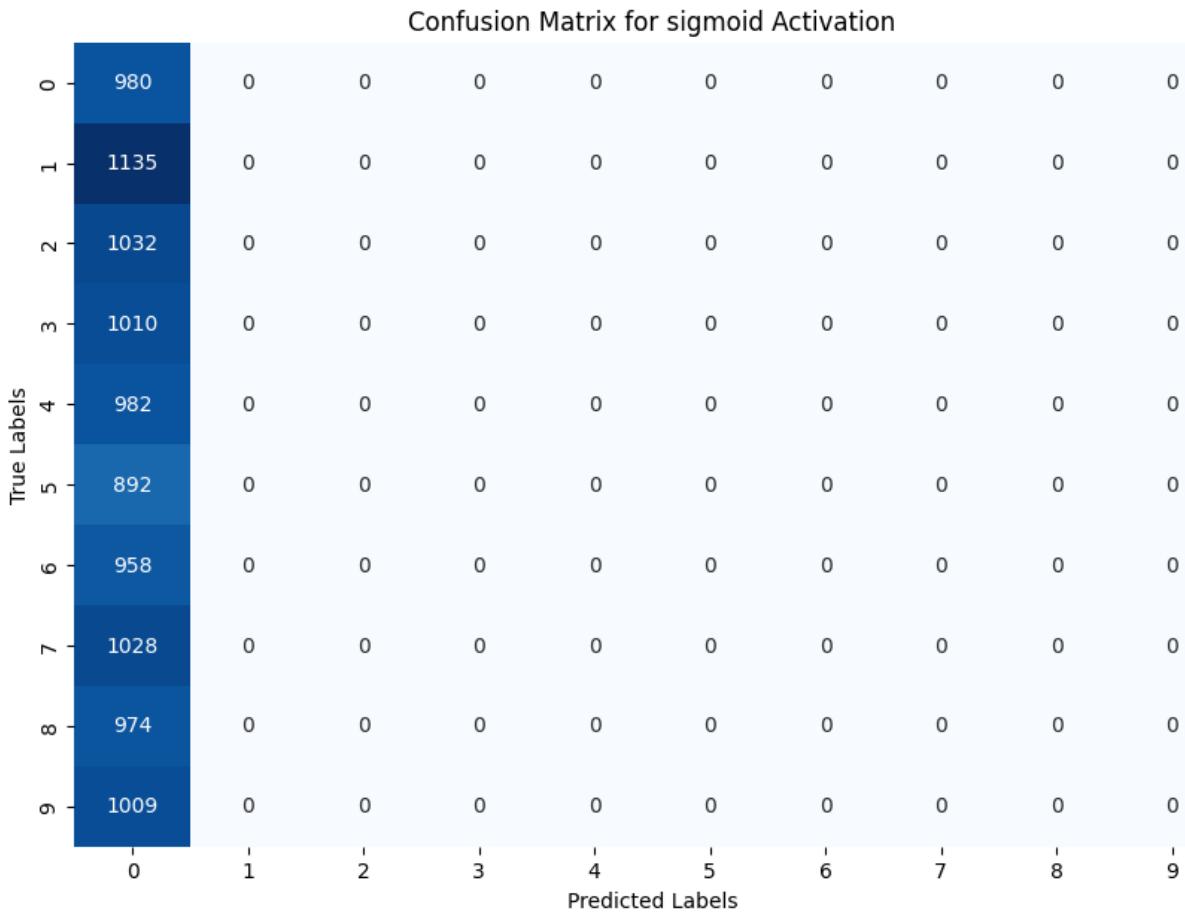
422/422 - 5s - loss: 2.3068 - accuracy: 0.1043 - val_loss: 2.3096 - val_accuracy: 0.1050 - 5s/epoch - 11ms/step

Epoch 4/5

422/422 - 5s - loss: 2.3068 - accuracy: 0.1056 - val_loss: 2.3056 -

```
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 5/5
422/422 - 5s - loss: 2.3066 - accuracy: 0.1047 - val_loss: 2.3079 -
val_accuracy: 0.0978 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 980   0   0   0   0   0   0   0   0   0]
 [1135   0   0   0   0   0   0   0   0   0]
 [1032   0   0   0   0   0   0   0   0   0]
 [1010   0   0   0   0   0   0   0   0   0]
 [ 982   0   0   0   0   0   0   0   0   0]
 [ 892   0   0   0   0   0   0   0   0   0]
 [ 958   0   0   0   0   0   0   0   0   0]
 [1028   0   0   0   0   0   0   0   0   0]
 [ 974   0   0   0   0   0   0   0   0   0]
 [1009   0   0   0   0   0   0   0   0   0]]
Precision: 0.0096
Recall: 0.0980

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

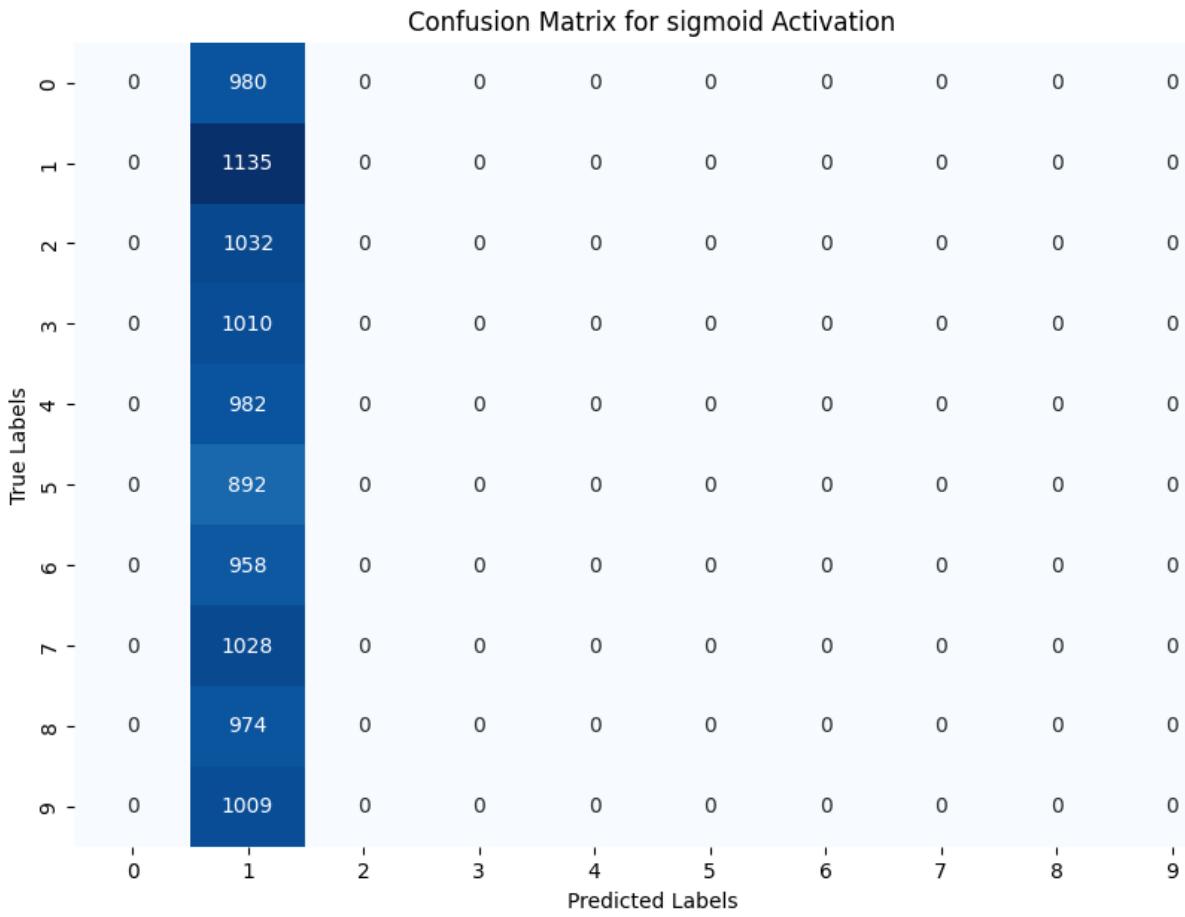


```

Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 6s - loss: 2.3095 - accuracy: 0.1055 - val_loss: 2.3111 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 2/15
422/422 - 5s - loss: 2.3069 - accuracy: 0.1042 - val_loss: 2.3073 -
val_accuracy: 0.1000 - 5s/epoch - 11ms/step
Epoch 3/15
422/422 - 5s - loss: 2.3070 - accuracy: 0.1066 - val_loss: 2.3079 -
val_accuracy: 0.0992 - 5s/epoch - 11ms/step
Epoch 4/15
422/422 - 5s - loss: 2.3062 - accuracy: 0.1068 - val_loss: 2.3094 -
val_accuracy: 0.1113 - 5s/epoch - 12ms/step
Epoch 5/15
422/422 - 5s - loss: 2.3059 - accuracy: 0.1038 - val_loss: 2.3034 -
val_accuracy: 0.0952 - 5s/epoch - 12ms/step
Epoch 6/15
422/422 - 5s - loss: 2.3055 - accuracy: 0.1065 - val_loss: 2.3058 -
val_accuracy: 0.1045 - 5s/epoch - 12ms/step
Epoch 7/15

```

```
422/422 - 5s - loss: 2.3058 - accuracy: 0.1060 - val_loss: 2.3051 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 8/15  
422/422 - 5s - loss: 2.3051 - accuracy: 0.1060 - val_loss: 2.3125 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 9/15  
422/422 - 5s - loss: 2.3054 - accuracy: 0.1054 - val_loss: 2.3064 -  
val_accuracy: 0.0960 - 5s/epoch - 12ms/step  
Epoch 10/15  
422/422 - 5s - loss: 2.3049 - accuracy: 0.1051 - val_loss: 2.3049 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 11/15  
422/422 - 5s - loss: 2.3044 - accuracy: 0.1057 - val_loss: 2.3034 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 12/15  
422/422 - 5s - loss: 2.3047 - accuracy: 0.1071 - val_loss: 2.3043 -  
val_accuracy: 0.0978 - 5s/epoch - 12ms/step  
Epoch 13/15  
422/422 - 5s - loss: 2.3046 - accuracy: 0.1073 - val_loss: 2.3052 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 14/15  
422/422 - 5s - loss: 2.3038 - accuracy: 0.1083 - val_loss: 2.3049 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 15/15  
422/422 - 5s - loss: 2.3034 - accuracy: 0.1093 - val_loss: 2.3045 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```

Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 6s - loss: 2.3121 - accuracy: 0.1051 - val_loss: 2.3068 -
val_accuracy: 0.1000 - 6s/epoch - 14ms/step
Epoch 2/20
422/422 - 5s - loss: 2.3077 - accuracy: 0.1068 - val_loss: 2.3073 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 3/20
422/422 - 5s - loss: 2.3072 - accuracy: 0.1042 - val_loss: 2.3148 -
val_accuracy: 0.0978 - 5s/epoch - 11ms/step
Epoch 4/20
422/422 - 5s - loss: 2.3061 - accuracy: 0.1051 - val_loss: 2.3055 -
val_accuracy: 0.1000 - 5s/epoch - 11ms/step
Epoch 5/20
422/422 - 5s - loss: 2.3064 - accuracy: 0.1068 - val_loss: 2.3070 -
val_accuracy: 0.1045 - 5s/epoch - 11ms/step
Epoch 6/20
422/422 - 5s - loss: 2.3060 - accuracy: 0.1073 - val_loss: 2.3072 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 7/20

```

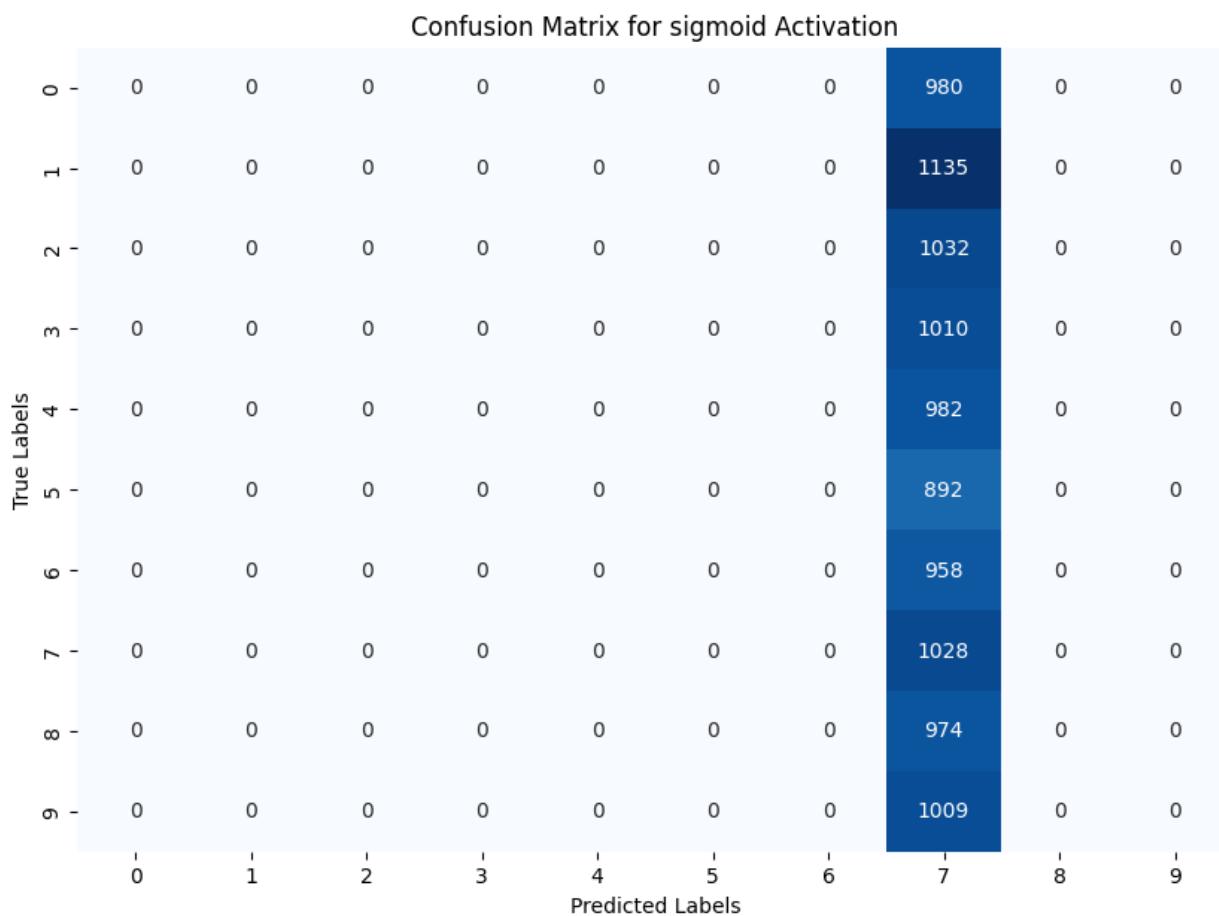
```
422/422 - 5s - loss: 2.3061 - accuracy: 0.1069 - val_loss: 2.3050 -  
val_accuracy: 0.0952 - 5s/epoch - 11ms/step  
Epoch 8/20  
422/422 - 5s - loss: 2.3053 - accuracy: 0.1088 - val_loss: 2.3061 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 9/20  
422/422 - 5s - loss: 2.3049 - accuracy: 0.1056 - val_loss: 2.3087 -  
val_accuracy: 0.1000 - 5s/epoch - 11ms/step  
Epoch 10/20  
422/422 - 5s - loss: 2.3046 - accuracy: 0.1074 - val_loss: 2.3098 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 11/20  
422/422 - 5s - loss: 2.3047 - accuracy: 0.1081 - val_loss: 2.3029 -  
val_accuracy: 0.1045 - 5s/epoch - 11ms/step  
Epoch 12/20  
422/422 - 5s - loss: 2.3040 - accuracy: 0.1084 - val_loss: 2.3037 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 13/20  
422/422 - 5s - loss: 2.3038 - accuracy: 0.1079 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 14/20  
422/422 - 5s - loss: 2.3031 - accuracy: 0.1126 - val_loss: 2.3030 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 15/20  
422/422 - 5s - loss: 2.3035 - accuracy: 0.1097 - val_loss: 2.3032 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 16/20  
422/422 - 5s - loss: 2.3027 - accuracy: 0.1100 - val_loss: 2.3021 -  
val_accuracy: 0.1077 - 5s/epoch - 11ms/step  
Epoch 17/20  
422/422 - 5s - loss: 2.3012 - accuracy: 0.1121 - val_loss: 2.2997 -  
val_accuracy: 0.1000 - 5s/epoch - 11ms/step  
Epoch 18/20  
422/422 - 5s - loss: 2.3001 - accuracy: 0.1155 - val_loss: 2.3005 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 19/20  
422/422 - 5s - loss: 2.2980 - accuracy: 0.1266 - val_loss: 2.2965 -  
val_accuracy: 0.1918 - 5s/epoch - 11ms/step  
Epoch 20/20  
422/422 - 5s - loss: 2.2936 - accuracy: 0.1420 - val_loss: 2.2876 -  
val_accuracy: 0.1113 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0  0  0  0  0  0  0  980  0  0]  
 [ 0  0  0  0  0  0  0  1135  0  0]  
 [ 0  0  0  0  0  0  0  1032  0  0]  
 [ 0  0  0  0  0  0  0  1010  0  0]  
 [ 0  0  0  0  0  0  0  982  0  0]]
```

```
[ 0  0  0  0  0  0  0  892  0  0]
[ 0  0  0  0  0  0  0  958  0  0]
[ 0  0  0  0  0  0  0 1028  0  0]
[ 0  0  0  0  0  0  0  974  0  0]
[ 0  0  0  0  0  0  0 1009  0  0]]
```

Precision: 0.0106

Recall: 0.1028

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 5 epochs..

Epoch 1/5

211/211 - 5s - loss: 2.3112 - accuracy: 0.1072 - val_loss: 2.3072 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step

Epoch 2/5

211/211 - 4s - loss: 2.3042 - accuracy: 0.1094 - val_loss: 2.3031 -

```
val_accuracy: 0.1045 - 4s/epoch - 19ms/step
Epoch 3/5
211/211 - 4s - loss: 2.3042 - accuracy: 0.1067 - val_loss: 2.3042 -
val_accuracy: 0.1113 - 4s/epoch - 19ms/step
Epoch 4/5
211/211 - 4s - loss: 2.3040 - accuracy: 0.1091 - val_loss: 2.3042 -
val_accuracy: 0.0992 - 4s/epoch - 18ms/step
Epoch 5/5
211/211 - 4s - loss: 2.3040 - accuracy: 0.1075 - val_loss: 2.3089 -
val_accuracy: 0.0960 - 4s/epoch - 19ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 0  0  0  0  0  0  0  0  0  980]
 [ 0  0  0  0  0  0  0  0  0  1135]
 [ 0  0  0  0  0  0  0  0  0  1032]
 [ 0  0  0  0  0  0  0  0  0  1010]
 [ 0  0  0  0  0  0  0  0  0  982]
 [ 0  0  0  0  0  0  0  0  0  892]
 [ 0  0  0  0  0  0  0  0  0  958]
 [ 0  0  0  0  0  0  0  0  0  1028]
 [ 0  0  0  0  0  0  0  0  0  974]
 [ 0  0  0  0  0  0  0  0  0  1009]]
Precision: 0.0102
Recall: 0.1009

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for sigmoid Activation

	0 -	1 -	2 -	3 -	4 -	5 -	6 -	7 -	8 -	9	980
True Labels	0	0	0	0	0	0	0	0	0	0	1135
	1 -	0	0	0	0	0	0	0	0	0	1032
	2 -	0	0	0	0	0	0	0	0	0	1010
	3 -	0	0	0	0	0	0	0	0	0	982
	4 -	0	0	0	0	0	0	0	0	0	892
	5 -	0	0	0	0	0	0	0	0	0	958
	6 -	0	0	0	0	0	0	0	0	0	1028
	7 -	0	0	0	0	0	0	0	0	0	974
	8 -	0	0	0	0	0	0	0	0	0	1009
	9 -	0	0	0	0	0	0	0	0	0	
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 5s - loss: 2.3144 - accuracy: 0.1030 - val_loss: 2.3050 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 2/15
211/211 - 4s - loss: 2.3044 - accuracy: 0.1073 - val_loss: 2.3077 -
val_accuracy: 0.0960 - 4s/epoch - 19ms/step
Epoch 3/15
211/211 - 4s - loss: 2.3043 - accuracy: 0.1043 - val_loss: 2.3033 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 4/15
211/211 - 4s - loss: 2.3045 - accuracy: 0.1076 - val_loss: 2.3041 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 5/15
211/211 - 4s - loss: 2.3046 - accuracy: 0.1065 - val_loss: 2.3083 -
val_accuracy: 0.1045 - 4s/epoch - 19ms/step
Epoch 6/15
211/211 - 4s - loss: 2.3038 - accuracy: 0.1078 - val_loss: 2.3040 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 7/15
```

```
211/211 - 4s - loss: 2.3047 - accuracy: 0.1085 - val_loss: 2.3040 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 8/15  
211/211 - 4s - loss: 2.3041 - accuracy: 0.1084 - val_loss: 2.3045 -  
val_accuracy: 0.0978 - 4s/epoch - 19ms/step  
Epoch 9/15  
211/211 - 4s - loss: 2.3041 - accuracy: 0.1074 - val_loss: 2.3055 -  
val_accuracy: 0.0960 - 4s/epoch - 18ms/step  
Epoch 10/15  
211/211 - 4s - loss: 2.3041 - accuracy: 0.1062 - val_loss: 2.3045 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 11/15  
211/211 - 4s - loss: 2.3038 - accuracy: 0.1074 - val_loss: 2.3043 -  
val_accuracy: 0.1113 - 4s/epoch - 18ms/step  
Epoch 12/15  
211/211 - 4s - loss: 2.3033 - accuracy: 0.1077 - val_loss: 2.3038 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 13/15  
211/211 - 4s - loss: 2.3042 - accuracy: 0.1051 - val_loss: 2.3049 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 14/15  
211/211 - 4s - loss: 2.3037 - accuracy: 0.1096 - val_loss: 2.3041 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 15/15  
211/211 - 4s - loss: 2.3040 - accuracy: 0.1071 - val_loss: 2.3044 -  
val_accuracy: 0.0960 - 4s/epoch - 18ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0  0  0  0  0  0  0  0  0  980]  
[ 0  0  0  0  0  0  0  0  0  1135]  
[ 0  0  0  0  0  0  0  0  0  1032]  
[ 0  0  0  0  0  0  0  0  0  1010]  
[ 0  0  0  0  0  0  0  0  0  982]  
[ 0  0  0  0  0  0  0  0  0  892]  
[ 0  0  0  0  0  0  0  0  0  958]  
[ 0  0  0  0  0  0  0  0  0  1028]  
[ 0  0  0  0  0  0  0  0  0  974]  
[ 0  0  0  0  0  0  0  0  0  1009]]  
Precision: 0.0102  
Recall: 0.1009  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for sigmoid Activation

	0 -	1 -	2 -	3 -	4 -	5 -	6 -	7 -	8 -	9	980
True Labels	0	0	0	0	0	0	0	0	0	0	1135
	1	0	0	0	0	0	0	0	0	0	1032
	2	0	0	0	0	0	0	0	0	0	1010
	3	0	0	0	0	0	0	0	0	0	982
	4	0	0	0	0	0	0	0	0	0	892
	5	0	0	0	0	0	0	0	0	0	958
	6	0	0	0	0	0	0	0	0	0	1028
	7	0	0	0	0	0	0	0	0	0	974
	8	0	0	0	0	0	0	0	0	0	1009
	9	0	0	0	0	0	0	0	0	0	9
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	

```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 5s - loss: 2.3098 - accuracy: 0.1082 - val_loss: 2.3065 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 2/20
211/211 - 4s - loss: 2.3047 - accuracy: 0.1076 - val_loss: 2.3037 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 3/20
211/211 - 4s - loss: 2.3041 - accuracy: 0.1060 - val_loss: 2.3043 -
val_accuracy: 0.1045 - 4s/epoch - 18ms/step
Epoch 4/20
211/211 - 4s - loss: 2.3044 - accuracy: 0.1072 - val_loss: 2.3064 -
val_accuracy: 0.1045 - 4s/epoch - 18ms/step
Epoch 5/20
211/211 - 4s - loss: 2.3045 - accuracy: 0.1064 - val_loss: 2.3050 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 6/20
211/211 - 4s - loss: 2.3045 - accuracy: 0.1056 - val_loss: 2.3023 -
val_accuracy: 0.1113 - 4s/epoch - 18ms/step
Epoch 7/20
```

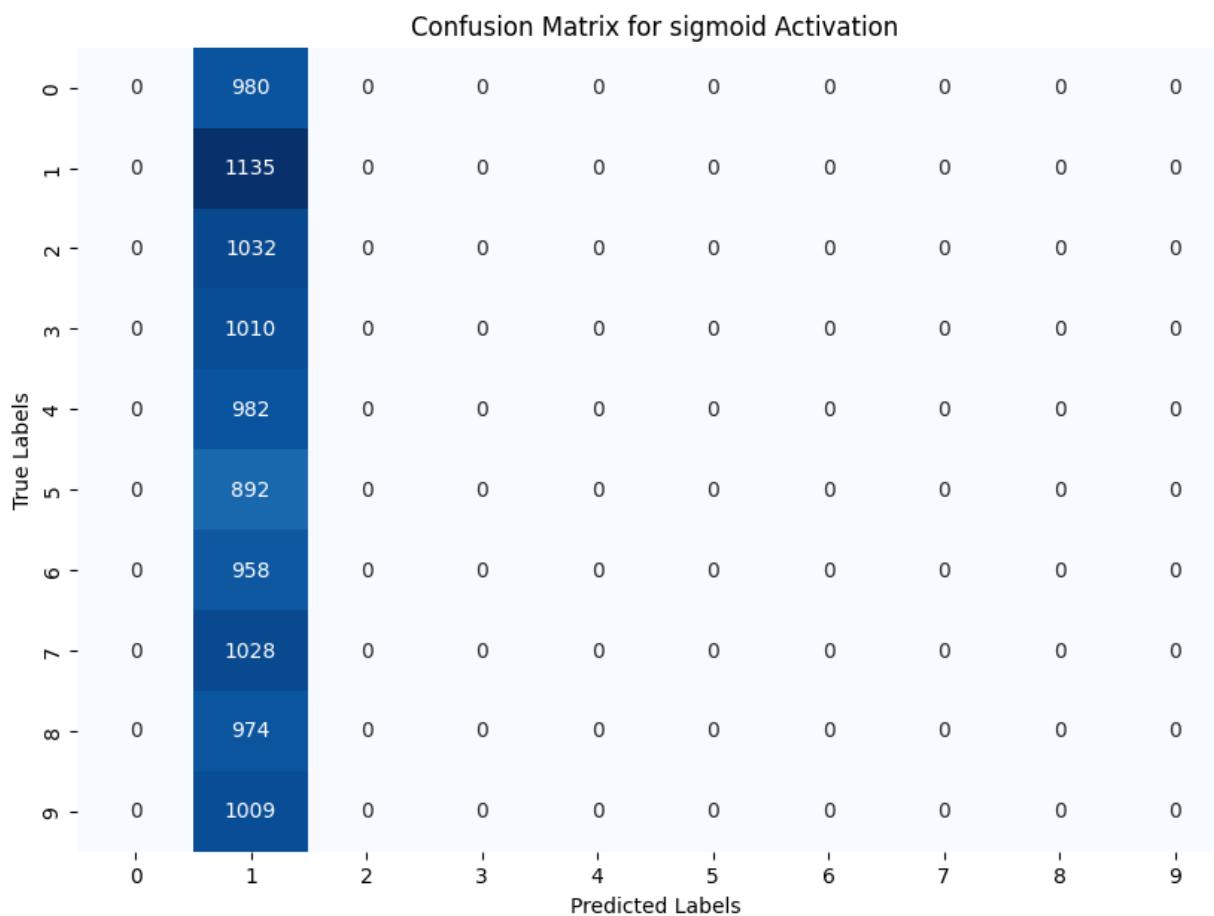
```
211/211 - 4s - loss: 2.3038 - accuracy: 0.1079 - val_loss: 2.3043 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 8/20  
211/211 - 4s - loss: 2.3045 - accuracy: 0.1041 - val_loss: 2.3063 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 9/20  
211/211 - 4s - loss: 2.3040 - accuracy: 0.1070 - val_loss: 2.3046 -  
val_accuracy: 0.1045 - 4s/epoch - 18ms/step  
Epoch 10/20  
211/211 - 4s - loss: 2.3042 - accuracy: 0.1091 - val_loss: 2.3058 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 11/20  
211/211 - 4s - loss: 2.3037 - accuracy: 0.1093 - val_loss: 2.3046 -  
val_accuracy: 0.1000 - 4s/epoch - 19ms/step  
Epoch 12/20  
211/211 - 4s - loss: 2.3035 - accuracy: 0.1067 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 13/20  
211/211 - 4s - loss: 2.3038 - accuracy: 0.1096 - val_loss: 2.3033 -  
val_accuracy: 0.1113 - 4s/epoch - 19ms/step  
Epoch 14/20  
211/211 - 4s - loss: 2.3037 - accuracy: 0.1084 - val_loss: 2.3034 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 15/20  
211/211 - 4s - loss: 2.3044 - accuracy: 0.1079 - val_loss: 2.3057 -  
val_accuracy: 0.0952 - 4s/epoch - 19ms/step  
Epoch 16/20  
211/211 - 4s - loss: 2.3045 - accuracy: 0.1083 - val_loss: 2.3039 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 17/20  
211/211 - 4s - loss: 2.3032 - accuracy: 0.1085 - val_loss: 2.3040 -  
val_accuracy: 0.1045 - 4s/epoch - 19ms/step  
Epoch 18/20  
211/211 - 4s - loss: 2.3038 - accuracy: 0.1050 - val_loss: 2.3034 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 19/20  
211/211 - 4s - loss: 2.3038 - accuracy: 0.1070 - val_loss: 2.3048 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 20/20  
211/211 - 4s - loss: 2.3028 - accuracy: 0.1098 - val_loss: 2.3069 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

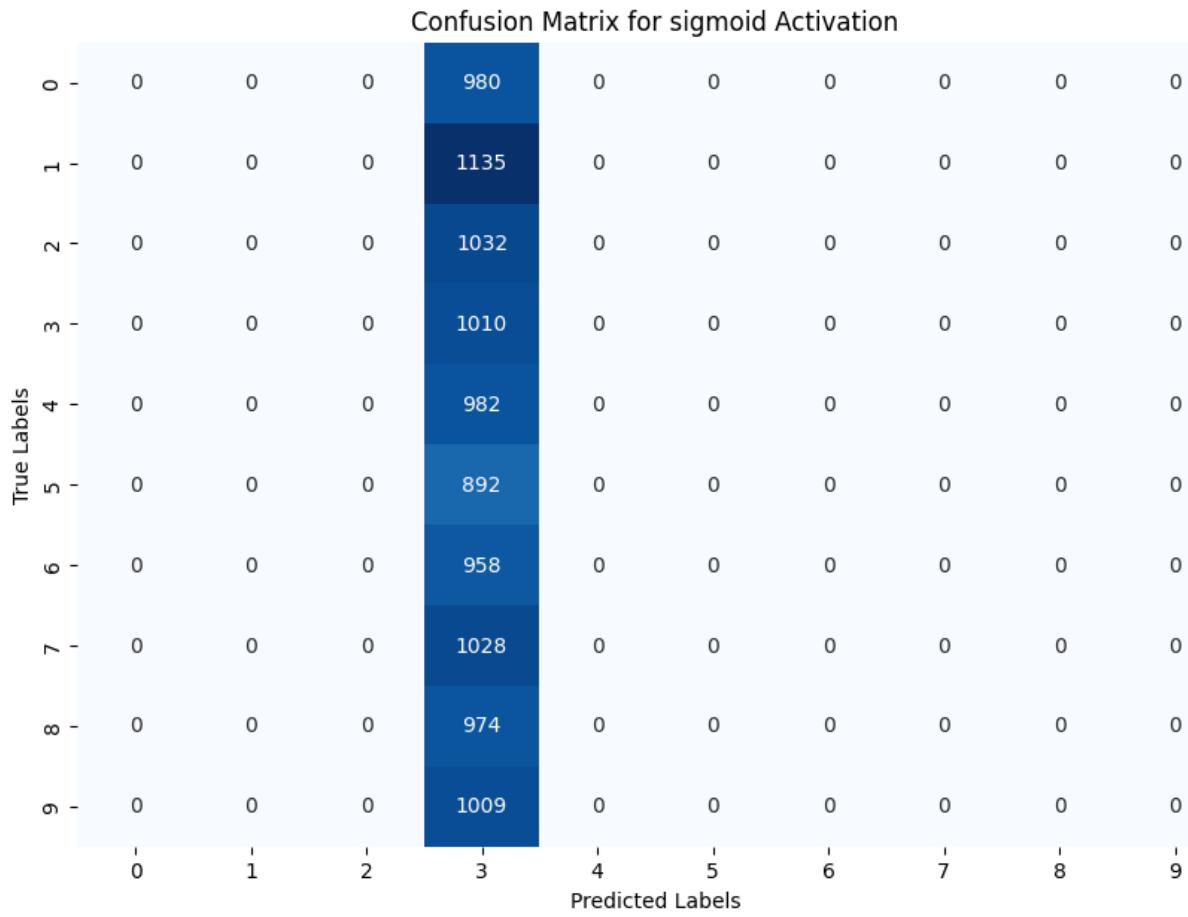
844/844 - 8s - loss: 2.3245 - accuracy: 0.1023 - val_loss: 2.3121 -
val_accuracy: 0.1113 - 8s/epoch - 10ms/step

Epoch 2/5

844/844 - 7s - loss: 2.3152 - accuracy: 0.1036 - val_loss: 2.3048 -

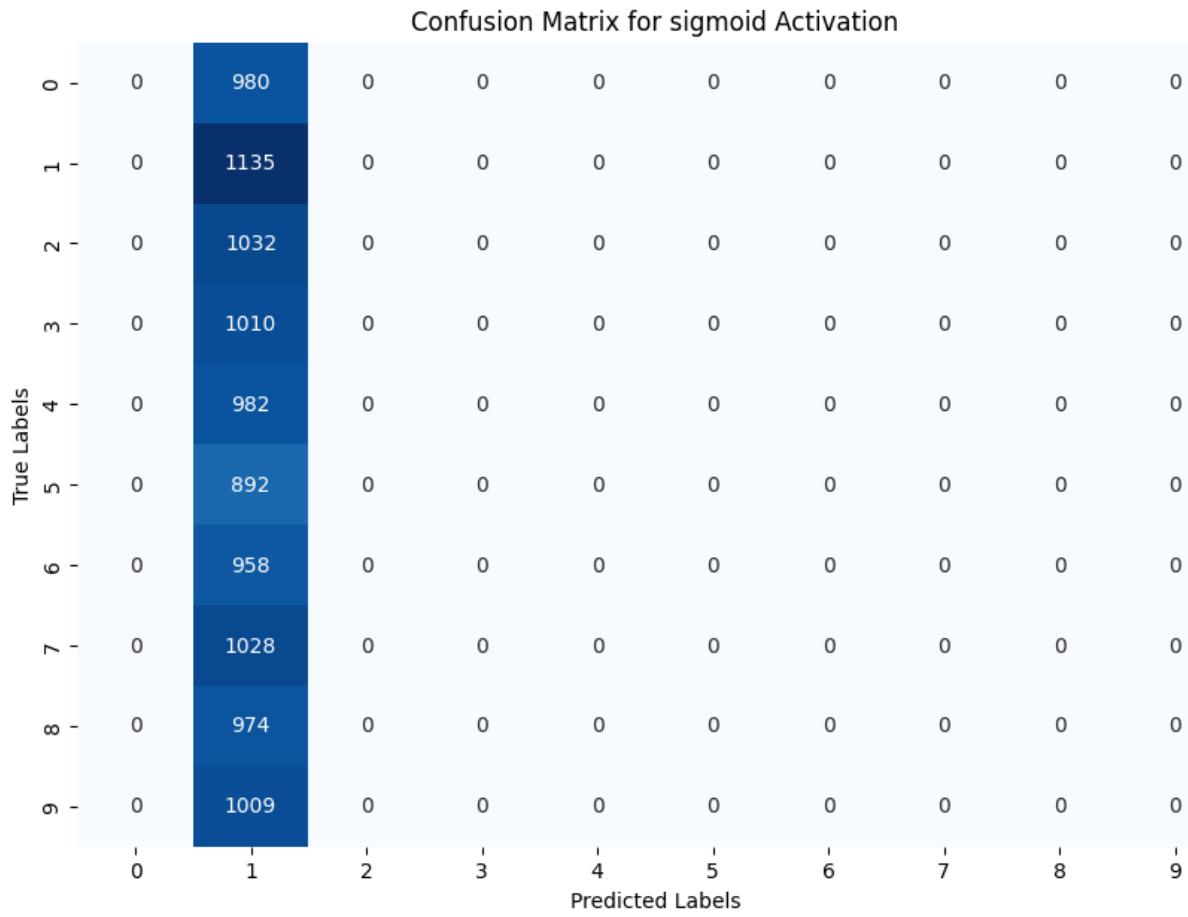
```
val_accuracy: 0.1113 - 7s/epoch - 9ms/step
Epoch 3/5
844/844 - 7s - loss: 2.3102 - accuracy: 0.1019 - val_loss: 2.3042 -
val_accuracy: 0.0952 - 7s/epoch - 9ms/step
Epoch 4/5
844/844 - 7s - loss: 2.3070 - accuracy: 0.1053 - val_loss: 2.3067 -
val_accuracy: 0.1050 - 7s/epoch - 9ms/step
Epoch 5/5
844/844 - 7s - loss: 2.3051 - accuracy: 0.1073 - val_loss: 2.3033 -
val_accuracy: 0.1045 - 7s/epoch - 9ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 0  0  0 980  0  0  0  0  0  0]
 [ 0  0  0 1135  0  0  0  0  0  0]
 [ 0  0  0 1032  0  0  0  0  0  0]
 [ 0  0  0 1010  0  0  0  0  0  0]
 [ 0  0  0 982  0  0  0  0  0  0]
 [ 0  0  0 892  0  0  0  0  0  0]
 [ 0  0  0 958  0  0  0  0  0  0]
 [ 0  0  0 1028  0  0  0  0  0  0]
 [ 0  0  0 974  0  0  0  0  0  0]
 [ 0  0  0 1009  0  0  0  0  0  0]]]
Precision: 0.0102
Recall: 0.1010

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 8s - loss: 2.3262 - accuracy: 0.1035 - val_loss: 2.3205 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 2/15
844/844 - 7s - loss: 2.3151 - accuracy: 0.1029 - val_loss: 2.3047 -
val_accuracy: 0.1050 - 7s/epoch - 9ms/step
Epoch 3/15
844/844 - 7s - loss: 2.3096 - accuracy: 0.1046 - val_loss: 2.3100 -
val_accuracy: 0.1050 - 7s/epoch - 9ms/step
Epoch 4/15
844/844 - 7s - loss: 2.3067 - accuracy: 0.1059 - val_loss: 2.3049 -
val_accuracy: 0.1050 - 7s/epoch - 9ms/step
Epoch 5/15
844/844 - 7s - loss: 2.3050 - accuracy: 0.1080 - val_loss: 2.3033 -
val_accuracy: 0.1050 - 7s/epoch - 9ms/step
Epoch 6/15
844/844 - 7s - loss: 2.3042 - accuracy: 0.1075 - val_loss: 2.3032 -
val_accuracy: 0.1050 - 7s/epoch - 9ms/step
Epoch 7/15
```

```
844/844 - 7s - loss: 2.3031 - accuracy: 0.1089 - val_loss: 2.3032 -  
val_accuracy: 0.0992 - 7s/epoch - 9ms/step  
Epoch 8/15  
844/844 - 7s - loss: 2.3028 - accuracy: 0.1077 - val_loss: 2.3055 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 9/15  
844/844 - 7s - loss: 2.3028 - accuracy: 0.1115 - val_loss: 2.3029 -  
val_accuracy: 0.1105 - 7s/epoch - 9ms/step  
Epoch 10/15  
844/844 - 8s - loss: 2.3020 - accuracy: 0.1103 - val_loss: 2.3011 -  
val_accuracy: 0.1113 - 8s/epoch - 9ms/step  
Epoch 11/15  
844/844 - 7s - loss: 2.3019 - accuracy: 0.1122 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 12/15  
844/844 - 7s - loss: 2.3015 - accuracy: 0.1147 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 13/15  
844/844 - 7s - loss: 2.3012 - accuracy: 0.1136 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 14/15  
844/844 - 7s - loss: 2.3003 - accuracy: 0.1170 - val_loss: 2.3016 -  
val_accuracy: 0.1103 - 7s/epoch - 9ms/step  
Epoch 15/15  
844/844 - 7s - loss: 2.2992 - accuracy: 0.1206 - val_loss: 2.2987 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 8s - loss: 2.3233 - accuracy: 0.1016 - val_loss: 2.3094 -
val_accuracy: 0.0992 - 8s/epoch - 10ms/step
Epoch 2/20
844/844 - 8s - loss: 2.3161 - accuracy: 0.1053 - val_loss: 2.3068 -
val_accuracy: 0.1045 - 8s/epoch - 9ms/step
Epoch 3/20
844/844 - 8s - loss: 2.3089 - accuracy: 0.1059 - val_loss: 2.3099 -
val_accuracy: 0.0978 - 8s/epoch - 9ms/step
Epoch 4/20
844/844 - 8s - loss: 2.3065 - accuracy: 0.1054 - val_loss: 2.3036 -
val_accuracy: 0.0960 - 8s/epoch - 9ms/step
Epoch 5/20
844/844 - 7s - loss: 2.3050 - accuracy: 0.1068 - val_loss: 2.3059 -
val_accuracy: 0.1050 - 7s/epoch - 9ms/step
Epoch 6/20
844/844 - 8s - loss: 2.3035 - accuracy: 0.1103 - val_loss: 2.3023 -
val_accuracy: 0.1113 - 8s/epoch - 9ms/step
Epoch 7/20
```

```
844/844 - 7s - loss: 2.3030 - accuracy: 0.1104 - val_loss: 2.3016 -  
val_accuracy: 0.1045 - 7s/epoch - 9ms/step  
Epoch 8/20  
844/844 - 8s - loss: 2.3029 - accuracy: 0.1084 - val_loss: 2.3048 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 9/20  
844/844 - 7s - loss: 2.3023 - accuracy: 0.1108 - val_loss: 2.3044 -  
val_accuracy: 0.0952 - 7s/epoch - 9ms/step  
Epoch 10/20  
844/844 - 7s - loss: 2.3022 - accuracy: 0.1108 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 11/20  
844/844 - 7s - loss: 2.3021 - accuracy: 0.1123 - val_loss: 2.3036 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 12/20  
844/844 - 7s - loss: 2.3016 - accuracy: 0.1113 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 13/20  
844/844 - 7s - loss: 2.3015 - accuracy: 0.1134 - val_loss: 2.3014 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 14/20  
844/844 - 7s - loss: 2.3012 - accuracy: 0.1142 - val_loss: 2.3015 -  
val_accuracy: 0.1045 - 7s/epoch - 9ms/step  
Epoch 15/20  
844/844 - 7s - loss: 2.3007 - accuracy: 0.1172 - val_loss: 2.3011 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 16/20  
844/844 - 7s - loss: 2.3001 - accuracy: 0.1218 - val_loss: 2.3005 -  
val_accuracy: 0.1050 - 7s/epoch - 9ms/step  
Epoch 17/20  
844/844 - 8s - loss: 2.2983 - accuracy: 0.1246 - val_loss: 2.2969 -  
val_accuracy: 0.1113 - 8s/epoch - 9ms/step  
Epoch 18/20  
844/844 - 7s - loss: 2.2921 - accuracy: 0.1481 - val_loss: 2.2802 -  
val_accuracy: 0.1018 - 7s/epoch - 9ms/step  
Epoch 19/20  
844/844 - 7s - loss: 2.1115 - accuracy: 0.2487 - val_loss: 1.7526 -  
val_accuracy: 0.3447 - 7s/epoch - 9ms/step  
Epoch 20/20  
844/844 - 7s - loss: 1.5004 - accuracy: 0.4619 - val_loss: 1.1688 -  
val_accuracy: 0.6112 - 7s/epoch - 9ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 865 0 4 0 60 4 7 1 2 37]  
 [ 0 1097 30 3 1 0 1 1 2 0]  
 [ 13 25 426 177 100 0 9 19 192 71]  
 [ 2 1 28 534 4 2 0 271 30 138]  
 [ 81 2 25 19 605 6 73 1 81 89]]
```

```
[ 141  11  56  52  81  31   8   19   70  423]
[ 73   33    7   0 113   0 729   0    1   2]
[ 1    1  28 114   0   0   0 807   2  75]
[ 23   6 126 178 143  13   2   22 223 238]
[ 30   0   8 120   9   4   1   71  17 749]]
```

Precision: 0.6079

Recall: 0.6066

Confusion Matrix for sigmoid Activation

	0	1	2	3	4	5	6	7	8	9	
0	865	0	4	0	60	4	7	1	2	37	
1	0	1097	30	3	1	0	1	1	2	0	
2	13	25	426	177	100	0	9	19	192	71	
3	2	1	28	534	4	2	0	271	30	138	
4	81	2	25	19	605	6	73	1	81	89	
5	141	11	56	52	81	31	8	19	70	423	
6	73	33	7	0	113	0	729	0	1	2	
7	1	1	28	114	0	0	0	807	2	75	
8	23	6	126	178	143	13	2	22	223	238	
9	30	0	8	120	9	4	1	71	17	749	
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

Training Model with sigmoid activation, 3 conv_layers, 3 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 6s - loss: 2.3161 - accuracy: 0.1059 - val_loss: 2.3198 - val_accuracy: 0.0960 - 6s/epoch - 14ms/step

Epoch 2/5

422/422 - 5s - loss: 2.3133 - accuracy: 0.1053 - val_loss: 2.3264 - val_accuracy: 0.1050 - 5s/epoch - 12ms/step

Epoch 3/5

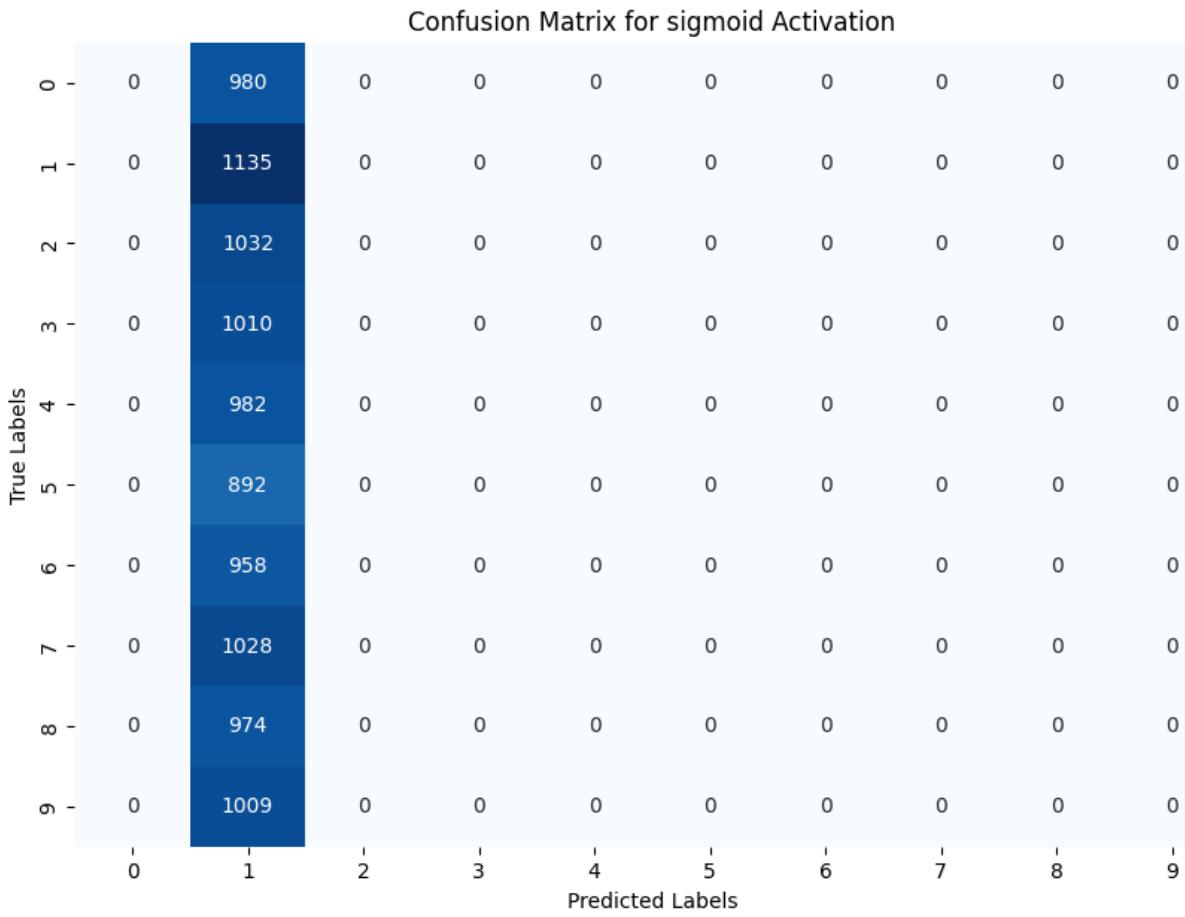
422/422 - 5s - loss: 2.3112 - accuracy: 0.1006 - val_loss: 2.3146 - val_accuracy: 0.1045 - 5s/epoch - 12ms/step

Epoch 4/5

422/422 - 5s - loss: 2.3102 - accuracy: 0.1051 - val_loss: 2.3091 -

```
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 5/5
422/422 - 5s - loss: 2.3111 - accuracy: 0.1037 - val_loss: 2.3064 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 6s - loss: 2.3161 - accuracy: 0.1040 - val_loss: 2.3103 -
val_accuracy: 0.1113 - 6s/epoch - 14ms/step
Epoch 2/15
422/422 - 5s - loss: 2.3114 - accuracy: 0.1039 - val_loss: 2.3168 -
val_accuracy: 0.0952 - 5s/epoch - 12ms/step
Epoch 3/15
422/422 - 5s - loss: 2.3136 - accuracy: 0.1034 - val_loss: 2.3102 -
val_accuracy: 0.1045 - 5s/epoch - 12ms/step
Epoch 4/15
422/422 - 5s - loss: 2.3103 - accuracy: 0.1026 - val_loss: 2.3061 -
val_accuracy: 0.1045 - 5s/epoch - 12ms/step
Epoch 5/15
422/422 - 5s - loss: 2.3086 - accuracy: 0.1054 - val_loss: 2.3072 -
val_accuracy: 0.1113 - 5s/epoch - 12ms/step
Epoch 6/15
422/422 - 5s - loss: 2.3075 - accuracy: 0.1070 - val_loss: 2.3142 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 7/15
```

```
422/422 - 5s - loss: 2.3069 - accuracy: 0.1053 - val_loss: 2.3067 -  
val_accuracy: 0.0992 - 5s/epoch - 12ms/step  
Epoch 8/15  
422/422 - 5s - loss: 2.3061 - accuracy: 0.1067 - val_loss: 2.3061 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 9/15  
422/422 - 5s - loss: 2.3052 - accuracy: 0.1060 - val_loss: 2.3112 -  
val_accuracy: 0.1000 - 5s/epoch - 12ms/step  
Epoch 10/15  
422/422 - 5s - loss: 2.3056 - accuracy: 0.1077 - val_loss: 2.3074 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 11/15  
422/422 - 5s - loss: 2.3048 - accuracy: 0.1054 - val_loss: 2.3017 -  
val_accuracy: 0.1833 - 5s/epoch - 12ms/step  
Epoch 12/15  
422/422 - 5s - loss: 2.3036 - accuracy: 0.1081 - val_loss: 2.3051 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 13/15  
422/422 - 5s - loss: 2.3031 - accuracy: 0.1085 - val_loss: 2.3012 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 14/15  
422/422 - 5s - loss: 2.3021 - accuracy: 0.1123 - val_loss: 2.3031 -  
val_accuracy: 0.1667 - 5s/epoch - 12ms/step  
Epoch 15/15  
422/422 - 5s - loss: 2.3018 - accuracy: 0.1144 - val_loss: 2.2999 -  
val_accuracy: 0.1113 - 5s/epoch - 12ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0  0  0  0  0  0  0  980  0  0]  
[ 0  0  0  0  0  0  0  1135  0  0]  
[ 0  0  0  0  0  0  0  1032  0  0]  
[ 0  0  0  0  0  0  0  1010  0  0]  
[ 0  0  0  0  0  0  0  982  0  0]  
[ 0  0  0  0  0  0  0  892  0  0]  
[ 0  0  0  0  0  0  0  958  0  0]  
[ 0  0  0  0  0  0  0  1028  0  0]  
[ 0  0  0  0  0  0  0  974  0  0]  
[ 0  0  0  0  0  0  0  1009  0  0]]  
Precision: 0.0106  
Recall: 0.1028  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for sigmoid Activation										
True Labels	0 -	1	2	3	4	5	6	7	8	9
0 -	980	0	0	0	0	0	0	0	0	0
1 -	1135	0	0	0	0	0	0	0	0	0
2 -	1032	0	0	0	0	0	0	0	0	0
3 -	1010	0	0	0	0	0	0	0	0	0
4 -	982	0	0	0	0	0	0	0	0	0
5 -	892	0	0	0	0	0	0	0	0	0
6 -	958	0	0	0	0	0	0	0	0	0
7 -	1028	0	0	0	0	0	0	0	0	0
8 -	974	0	0	0	0	0	0	0	0	0
9 -	1009	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9
	Predicted Labels									

```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 6s - loss: 2.3205 - accuracy: 0.1045 - val_loss: 2.3110 -
val_accuracy: 0.1045 - 6s/epoch - 14ms/step
Epoch 2/20
422/422 - 5s - loss: 2.3160 - accuracy: 0.1044 - val_loss: 2.3085 -
val_accuracy: 0.1000 - 5s/epoch - 12ms/step
Epoch 3/20
422/422 - 5s - loss: 2.3118 - accuracy: 0.1029 - val_loss: 2.3135 -
val_accuracy: 0.0995 - 5s/epoch - 12ms/step
Epoch 4/20
422/422 - 5s - loss: 2.3111 - accuracy: 0.1044 - val_loss: 2.3181 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 5/20
422/422 - 5s - loss: 2.3095 - accuracy: 0.1069 - val_loss: 2.3201 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 6/20
422/422 - 5s - loss: 2.3087 - accuracy: 0.1040 - val_loss: 2.3095 -
val_accuracy: 0.1113 - 5s/epoch - 12ms/step
Epoch 7/20
```

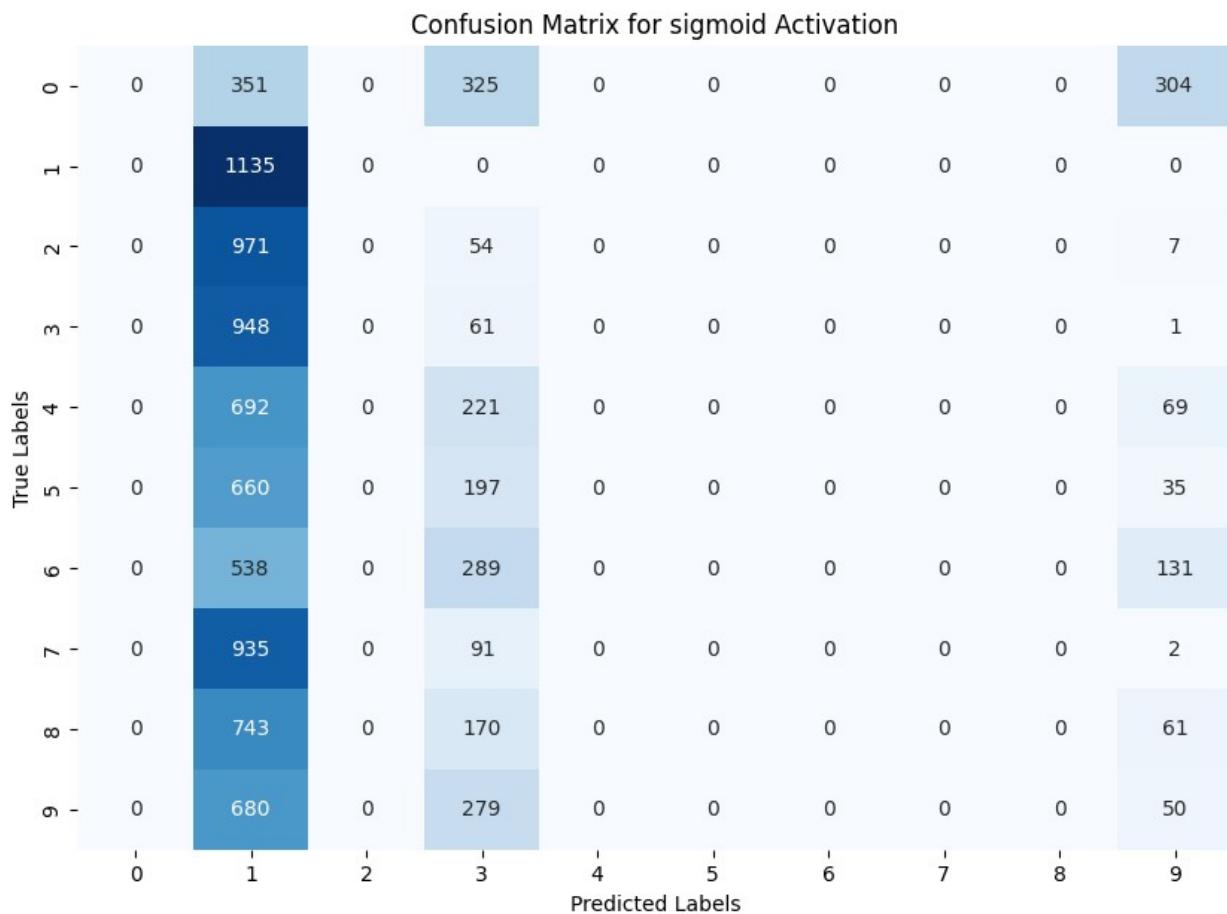
```
422/422 - 5s - loss: 2.3071 - accuracy: 0.1055 - val_loss: 2.3076 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 8/20  
422/422 - 5s - loss: 2.3060 - accuracy: 0.1037 - val_loss: 2.3073 -  
val_accuracy: 0.1603 - 5s/epoch - 12ms/step  
Epoch 9/20  
422/422 - 5s - loss: 2.3057 - accuracy: 0.1076 - val_loss: 2.3102 -  
val_accuracy: 0.0992 - 5s/epoch - 12ms/step  
Epoch 10/20  
422/422 - 5s - loss: 2.3055 - accuracy: 0.1057 - val_loss: 2.3079 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 11/20  
422/422 - 5s - loss: 2.3050 - accuracy: 0.1069 - val_loss: 2.3051 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 12/20  
422/422 - 5s - loss: 2.3041 - accuracy: 0.1070 - val_loss: 2.3060 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 13/20  
422/422 - 5s - loss: 2.3040 - accuracy: 0.1092 - val_loss: 2.3075 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 14/20  
422/422 - 5s - loss: 2.3032 - accuracy: 0.1104 - val_loss: 2.3051 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 15/20  
422/422 - 5s - loss: 2.3030 - accuracy: 0.1116 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 16/20  
422/422 - 5s - loss: 2.3020 - accuracy: 0.1124 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 17/20  
422/422 - 5s - loss: 2.3020 - accuracy: 0.1101 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 18/20  
422/422 - 5s - loss: 2.3008 - accuracy: 0.1180 - val_loss: 2.3041 -  
val_accuracy: 0.1943 - 5s/epoch - 12ms/step  
Epoch 19/20  
422/422 - 5s - loss: 2.2992 - accuracy: 0.1190 - val_loss: 2.2977 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 20/20  
422/422 - 5s - loss: 2.2957 - accuracy: 0.1307 - val_loss: 2.2943 -  
val_accuracy: 0.1152 - 5s/epoch - 12ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0 351 0 325 0 0 0 0 304]  
[ 0 1135 0 0 0 0 0 0 0]  
[ 0 971 0 54 0 0 0 0 7]  
[ 0 948 0 61 0 0 0 0 1]  
[ 0 692 0 221 0 0 0 0 69]]
```

```
[ 0 660 0 197 0 0 0 0 0 35]
[ 0 538 0 289 0 0 0 0 0 131]
[ 0 935 0 91 0 0 0 0 0 2]
[ 0 743 0 170 0 0 0 0 0 61]
[ 0 680 0 279 0 0 0 0 0 50]]
```

Precision: 0.0281

Recall: 0.1246

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 5 epochs..

Epoch 1/5

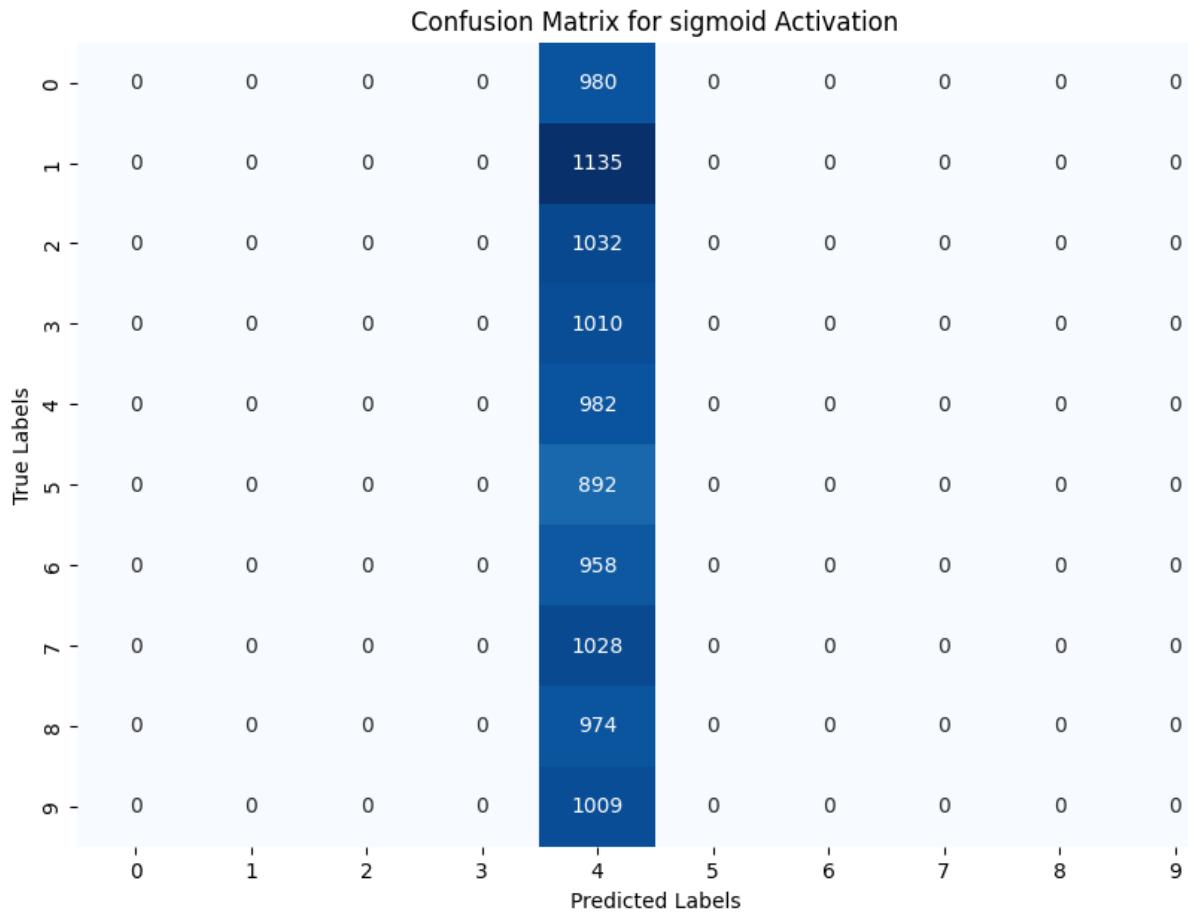
211/211 - 5s - loss: 2.3086 - accuracy: 0.1062 - val_loss: 2.3058 -
val_accuracy: 0.1113 - 5s/epoch - 24ms/step

Epoch 2/5

211/211 - 4s - loss: 2.3070 - accuracy: 0.1063 - val_loss: 2.3132 -

```
val_accuracy: 0.1050 - 4s/epoch - 20ms/step
Epoch 3/5
211/211 - 4s - loss: 2.3086 - accuracy: 0.1079 - val_loss: 2.3025 -
val_accuracy: 0.1113 - 4s/epoch - 19ms/step
Epoch 4/5
211/211 - 4s - loss: 2.3069 - accuracy: 0.1056 - val_loss: 2.3028 -
val_accuracy: 0.1000 - 4s/epoch - 19ms/step
Epoch 5/5
211/211 - 4s - loss: 2.3064 - accuracy: 0.1060 - val_loss: 2.3082 -
val_accuracy: 0.0992 - 4s/epoch - 20ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: sigmoid
Confusion Matrix:
[[ 0  0  0  0 980  0  0  0  0  0]
 [ 0  0  0  0 1135  0  0  0  0  0]
 [ 0  0  0  0 1032  0  0  0  0  0]
 [ 0  0  0  0 1010  0  0  0  0  0]
 [ 0  0  0  0 982  0  0  0  0  0]
 [ 0  0  0  0 892  0  0  0  0  0]
 [ 0  0  0  0 958  0  0  0  0  0]
 [ 0  0  0  0 1028  0  0  0  0  0]
 [ 0  0  0  0 974  0  0  0  0  0]
 [ 0  0  0  0 1009  0  0  0  0  0]]]
Precision: 0.0096
Recall: 0.0982

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 5s - loss: 2.3084 - accuracy: 0.1061 - val_loss: 2.3052 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 2/15
211/211 - 4s - loss: 2.3070 - accuracy: 0.1049 - val_loss: 2.3060 -
val_accuracy: 0.1050 - 4s/epoch - 20ms/step
Epoch 3/15
211/211 - 4s - loss: 2.3072 - accuracy: 0.1042 - val_loss: 2.3105 -
val_accuracy: 0.1050 - 4s/epoch - 20ms/step
Epoch 4/15
211/211 - 4s - loss: 2.3076 - accuracy: 0.1033 - val_loss: 2.3039 -
val_accuracy: 0.1050 - 4s/epoch - 20ms/step
Epoch 5/15
211/211 - 4s - loss: 2.3080 - accuracy: 0.1066 - val_loss: 2.3136 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 6/15
211/211 - 4s - loss: 2.3074 - accuracy: 0.1051 - val_loss: 2.3137 -
val_accuracy: 0.0952 - 4s/epoch - 20ms/step
Epoch 7/15
```

```
211/211 - 4s - loss: 2.3066 - accuracy: 0.1067 - val_loss: 2.3039 -  
val_accuracy: 0.1113 - 4s/epoch - 20ms/step  
Epoch 8/15  
211/211 - 4s - loss: 2.3056 - accuracy: 0.1065 - val_loss: 2.3133 -  
val_accuracy: 0.1000 - 4s/epoch - 20ms/step  
Epoch 9/15  
211/211 - 4s - loss: 2.3066 - accuracy: 0.1081 - val_loss: 2.3165 -  
val_accuracy: 0.0992 - 4s/epoch - 20ms/step  
Epoch 10/15  
211/211 - 4s - loss: 2.3073 - accuracy: 0.1039 - val_loss: 2.3095 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 11/15  
211/211 - 4s - loss: 2.3051 - accuracy: 0.1069 - val_loss: 2.3045 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 12/15  
211/211 - 4s - loss: 2.3056 - accuracy: 0.1057 - val_loss: 2.3060 -  
val_accuracy: 0.0978 - 4s/epoch - 20ms/step  
Epoch 13/15  
211/211 - 4s - loss: 2.3053 - accuracy: 0.1075 - val_loss: 2.3052 -  
val_accuracy: 0.1045 - 4s/epoch - 20ms/step  
Epoch 14/15  
211/211 - 4s - loss: 2.3060 - accuracy: 0.1066 - val_loss: 2.3089 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 15/15  
211/211 - 4s - loss: 2.3055 - accuracy: 0.1036 - val_loss: 2.3059 -  
val_accuracy: 0.1113 - 4s/epoch - 20ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0  0  0  0  0  0  0  980  0  0]  
[ 0  0  0  0  0  0  0  1135  0  0]  
[ 0  0  0  0  0  0  0  1032  0  0]  
[ 0  0  0  0  0  0  0  1010  0  0]  
[ 0  0  0  0  0  0  0  982  0  0]  
[ 0  0  0  0  0  0  0  892  0  0]  
[ 0  0  0  0  0  0  0  958  0  0]  
[ 0  0  0  0  0  0  0  1028  0  0]  
[ 0  0  0  0  0  0  0  974  0  0]  
[ 0  0  0  0  0  0  0  1009  0  0]]  
Precision: 0.0106  
Recall: 0.1028  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for sigmoid Activation

	0 -	1 -	2 -	3 -	4 -	5 -	6 -	7 -	8 -	9 -	
True Labels	0	0	0	0	0	0	0	0	0	0	980
	0	0	0	0	0	0	0	0	0	0	1135
	0	0	0	0	0	0	0	0	0	0	1032
	0	0	0	0	0	0	0	0	0	0	1010
	0	0	0	0	0	0	0	0	0	0	982
	0	0	0	0	0	0	0	0	0	0	892
	0	0	0	0	0	0	0	0	0	0	958
	0	0	0	0	0	0	0	0	0	0	1028
	0	0	0	0	0	0	0	0	0	0	974
	0	0	0	0	0	0	0	0	0	0	1009
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels

```
Training Model with sigmoid activation, 3 conv_layers, 3 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 5s - loss: 2.3179 - accuracy: 0.1035 - val_loss: 2.3037 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 2/20
211/211 - 4s - loss: 2.3082 - accuracy: 0.1045 - val_loss: 2.3141 -
val_accuracy: 0.0978 - 4s/epoch - 20ms/step
Epoch 3/20
211/211 - 4s - loss: 2.3078 - accuracy: 0.1043 - val_loss: 2.3059 -
val_accuracy: 0.0960 - 4s/epoch - 20ms/step
Epoch 4/20
211/211 - 4s - loss: 2.3073 - accuracy: 0.1042 - val_loss: 2.3072 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 5/20
211/211 - 4s - loss: 2.3089 - accuracy: 0.1055 - val_loss: 2.3150 -
val_accuracy: 0.0978 - 4s/epoch - 20ms/step
Epoch 6/20
211/211 - 4s - loss: 2.3085 - accuracy: 0.1032 - val_loss: 2.3114 -
val_accuracy: 0.1113 - 4s/epoch - 20ms/step
Epoch 7/20
```

```
211/211 - 4s - loss: 2.3080 - accuracy: 0.1050 - val_loss: 2.3054 -  
val_accuracy: 0.0978 - 4s/epoch - 19ms/step  
Epoch 8/20  
211/211 - 4s - loss: 2.3067 - accuracy: 0.1062 - val_loss: 2.3063 -  
val_accuracy: 0.0992 - 4s/epoch - 19ms/step  
Epoch 9/20  
211/211 - 4s - loss: 2.3073 - accuracy: 0.1049 - val_loss: 2.3050 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 10/20  
211/211 - 4s - loss: 2.3061 - accuracy: 0.1078 - val_loss: 2.3114 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 11/20  
211/211 - 4s - loss: 2.3065 - accuracy: 0.1077 - val_loss: 2.3045 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 12/20  
211/211 - 4s - loss: 2.3056 - accuracy: 0.1037 - val_loss: 2.3042 -  
val_accuracy: 0.0978 - 4s/epoch - 21ms/step  
Epoch 13/20  
211/211 - 4s - loss: 2.3066 - accuracy: 0.1046 - val_loss: 2.3046 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 14/20  
211/211 - 4s - loss: 2.3067 - accuracy: 0.1055 - val_loss: 2.3094 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 15/20  
211/211 - 4s - loss: 2.3060 - accuracy: 0.1051 - val_loss: 2.3113 -  
val_accuracy: 0.1050 - 4s/epoch - 21ms/step  
Epoch 16/20  
211/211 - 4s - loss: 2.3054 - accuracy: 0.1067 - val_loss: 2.3072 -  
val_accuracy: 0.1050 - 4s/epoch - 20ms/step  
Epoch 17/20  
211/211 - 4s - loss: 2.3046 - accuracy: 0.1092 - val_loss: 2.3074 -  
val_accuracy: 0.1045 - 4s/epoch - 21ms/step  
Epoch 18/20  
211/211 - 4s - loss: 2.3041 - accuracy: 0.1080 - val_loss: 2.3046 -  
val_accuracy: 0.1000 - 4s/epoch - 21ms/step  
Epoch 19/20  
211/211 - 4s - loss: 2.3040 - accuracy: 0.1097 - val_loss: 2.3065 -  
val_accuracy: 0.0978 - 4s/epoch - 20ms/step  
Epoch 20/20  
211/211 - 4s - loss: 2.3032 - accuracy: 0.1119 - val_loss: 2.3059 -  
val_accuracy: 0.0960 - 4s/epoch - 20ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: sigmoid  
Confusion Matrix:  
[[ 0  0  0  0  0  0  0  0  980]  
 [ 0  0  0  0  0  0  0  0  1135]  
 [ 0  0  0  0  0  0  0  0  1032]  
 [ 0  0  0  0  0  0  0  0  1010]  
 [ 0  0  0  0  0  0  0  0  982]]
```

```
[ 0  0  0  0  0  0  0  0  0  892]
[ 0  0  0  0  0  0  0  0  0  958]
[ 0  0  0  0  0  0  0  0  0  1028]
[ 0  0  0  0  0  0  0  0  0  974]
[ 0  0  0  0  0  0  0  0  0  1009]]
```

Precision: 0.0102

Recall: 0.1009

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for sigmoid Activation										
True Labels	0 -	0	0	0	0	0	0	0	0	980
	1 -	0	0	0	0	0	0	0	0	1135
	2 -	0	0	0	0	0	0	0	0	1032
	3 -	0	0	0	0	0	0	0	0	1010
	4 -	0	0	0	0	0	0	0	0	982
	5 -	0	0	0	0	0	0	0	0	892
	6 -	0	0	0	0	0	0	0	0	958
	7 -	0	0	0	0	0	0	0	0	1028
	8 -	0	0	0	0	0	0	0	0	974
	9 -	0	1	2	3	4	5	6	7	8
Predicted Labels										

Training Model with softmax activation, 1 conv_layers, 1 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

844/844 - 6s - loss: 2.3010 - accuracy: 0.1123 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step

Epoch 2/5

844/844 - 6s - loss: 2.2996 - accuracy: 0.1132 - val_loss: 2.2995 -

```
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 3/5
844/844 - 6s - loss: 2.2974 - accuracy: 0.1132 - val_loss: 2.2962 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 4/5
844/844 - 5s - loss: 2.2910 - accuracy: 0.1209 - val_loss: 2.2755 -
val_accuracy: 0.2480 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 2.0954 - accuracy: 0.2451 - val_loss: 1.8604 -
val_accuracy: 0.3340 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 949   2   9   3   1   0  14   2   0   0]
 [  4 1112   5   7   0   0   1   6   0   0]
 [ 748   55   54   43   1   0  88   37   4   2]
 [ 247   211   82  123   4   0 106  218  15   4]
 [  27   20   22   14   7   0  18  813  45  16]
 [ 171   80   77  119   8   0  85  303  34  15]
 [ 868   6   21   13   0   0  46   3   1   0]
 [  2   74   7   6   2   0   6  924   6   1]
 [ 126   80   83  73  11   0  62  442  64  33]
 [ 15   13   11   2   2   0   7  929  15  15]]]
Precision: 0.2582
Recall: 0.3294

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation

	0	1	2	3	4	5	6	7	8	9	
True Labels	949	2	9	3	1	0	14	2	0	0	0
0	949	2	9	3	1	0	14	2	0	0	0
1	4	1112	5	7	0	0	1	6	0	0	0
2	748	55	54	43	1	0	88	37	4	2	
3	247	211	82	123	4	0	106	218	15	4	
4	27	20	22	14	7	0	18	813	45	16	
5	171	80	77	119	8	0	85	303	34	15	
6	868	6	21	13	0	0	46	3	1	0	
7	2	74	7	6	2	0	6	924	6	1	
8	126	80	83	73	11	0	62	442	64	33	
9	15	13	11	2	2	0	7	929	15	15	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels

```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 6s - loss: 2.3011 - accuracy: 0.1124 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 2/15
844/844 - 6s - loss: 2.2998 - accuracy: 0.1132 - val_loss: 2.2995 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 3/15
844/844 - 6s - loss: 2.2982 - accuracy: 0.1132 - val_loss: 2.2975 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 4/15
844/844 - 6s - loss: 2.2955 - accuracy: 0.1132 - val_loss: 2.2931 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 5/15
844/844 - 6s - loss: 2.2819 - accuracy: 0.1611 - val_loss: 2.2368 -
val_accuracy: 0.2010 - 6s/epoch - 7ms/step
Epoch 6/15
844/844 - 6s - loss: 2.0738 - accuracy: 0.2447 - val_loss: 1.8447 -
val_accuracy: 0.3353 - 6s/epoch - 7ms/step
Epoch 7/15
```

```
844/844 - 6s - loss: 1.5282 - accuracy: 0.4440 - val_loss: 1.2992 -  
val_accuracy: 0.4940 - 6s/epoch - 7ms/step  
Epoch 8/15  
844/844 - 6s - loss: 1.2431 - accuracy: 0.5008 - val_loss: 1.1537 -  
val_accuracy: 0.5342 - 6s/epoch - 7ms/step  
Epoch 9/15  
844/844 - 6s - loss: 1.1453 - accuracy: 0.5370 - val_loss: 1.0838 -  
val_accuracy: 0.5765 - 6s/epoch - 7ms/step  
Epoch 10/15  
844/844 - 6s - loss: 1.0844 - accuracy: 0.5644 - val_loss: 1.0327 -  
val_accuracy: 0.5778 - 6s/epoch - 7ms/step  
Epoch 11/15  
844/844 - 6s - loss: 1.0383 - accuracy: 0.5901 - val_loss: 0.9924 -  
val_accuracy: 0.6203 - 6s/epoch - 7ms/step  
Epoch 12/15  
844/844 - 6s - loss: 1.0002 - accuracy: 0.6047 - val_loss: 0.9551 -  
val_accuracy: 0.6447 - 6s/epoch - 7ms/step  
Epoch 13/15  
844/844 - 6s - loss: 0.9706 - accuracy: 0.6220 - val_loss: 0.9181 -  
val_accuracy: 0.6525 - 6s/epoch - 7ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.9420 - accuracy: 0.6350 - val_loss: 0.8929 -  
val_accuracy: 0.6942 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 6s - loss: 0.9137 - accuracy: 0.6510 - val_loss: 0.8580 -  
val_accuracy: 0.7052 - 6s/epoch - 7ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 870  0  20  0  15  0  74  1  0  0]  
[  0 1119  2  1  1  1  3  0  7  1]  
[ 30  9 583  5  17  49 106  1 211 21]  
[  0  0  6 922  0  36  0  2 35  9]  
[ 14  2  2  0 836  0  19  21 12 76]  
[  1  2 42 635  0 110  9  0 84  9]  
[ 623  3 25  0  4  0 300  0  3  0]  
[  0  6  0  1  3  1  2 919 18 78]  
[  4 31 295 37 15 62  44  2 445 39]  
[  1 11  5 15 62  6  3 269 25 612]]  
Precision: 0.6586  
Recall: 0.6716
```

Confusion Matrix for softmax Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	0	870	0	20	0	15	0	74	1	0	0
1	0	1119	2	1	1	1	3	0	7	1	
2	30	9	583	5	17	49	106	1	211	21	
3	0	0	6	922	0	36	0	2	35	9	
4	14	2	2	0	836	0	19	21	12	76	
5	1	2	42	635	0	110	9	0	84	9	
6	623	3	25	0	4	0	300	0	3	0	
7	0	6	0	1	3	1	2	919	18	78	
8	4	31	295	37	15	62	44	2	445	39	
9	1	11	5	15	62	6	3	269	25	612	
0	1	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```

Training Model with softmax activation, 1 conv_layers, 1 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 6s - loss: 2.3011 - accuracy: 0.1128 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 2/20
844/844 - 6s - loss: 2.2996 - accuracy: 0.1132 - val_loss: 2.2998 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 3/20
844/844 - 6s - loss: 2.2976 - accuracy: 0.1132 - val_loss: 2.2963 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 4/20
844/844 - 6s - loss: 2.2933 - accuracy: 0.1133 - val_loss: 2.2893 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 5/20
844/844 - 6s - loss: 2.2210 - accuracy: 0.1997 - val_loss: 2.0106 -
val_accuracy: 0.2740 - 6s/epoch - 7ms/step
Epoch 6/20
844/844 - 6s - loss: 1.7038 - accuracy: 0.3283 - val_loss: 1.4966 -
val_accuracy: 0.3403 - 6s/epoch - 7ms/step
Epoch 7/20

```

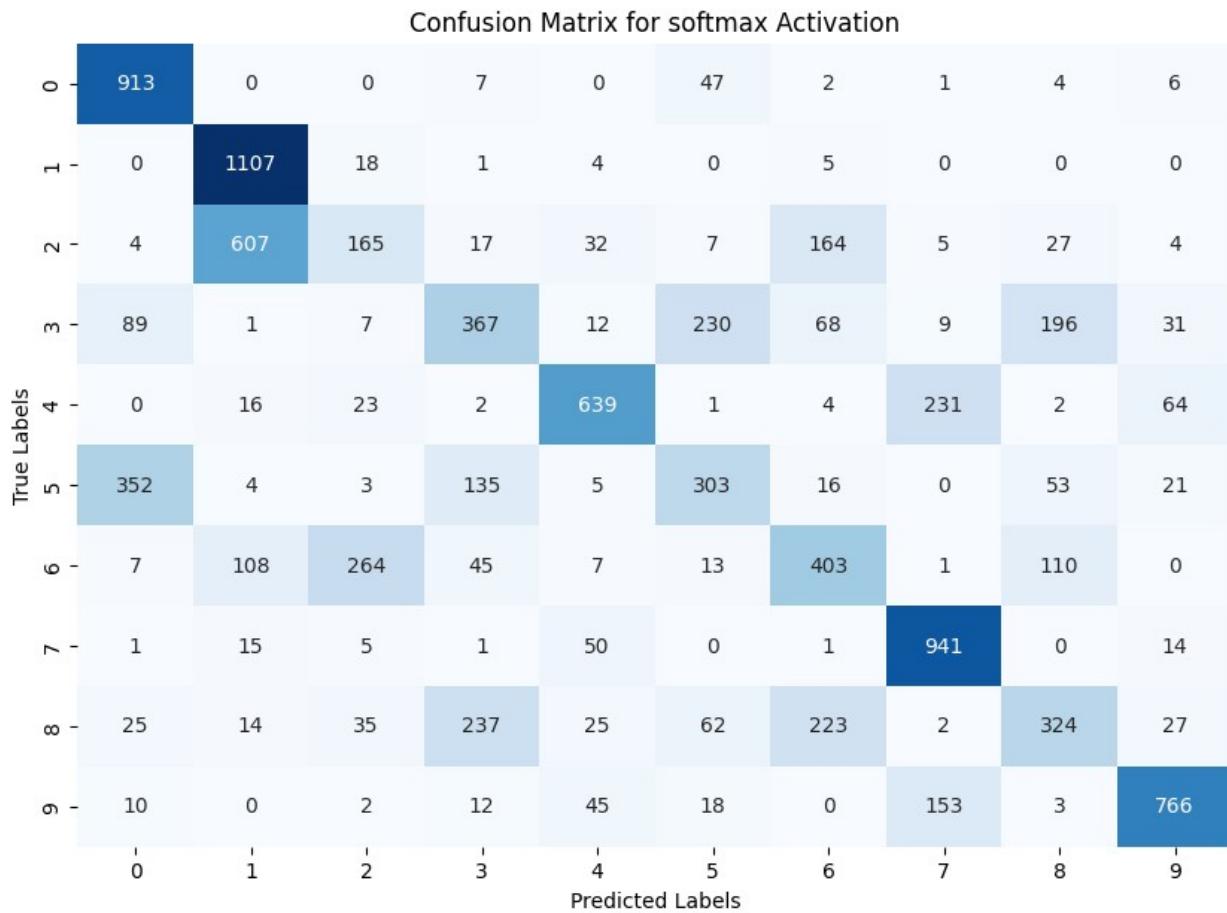
```

844/844 - 6s - loss: 1.4585 - accuracy: 0.3645 - val_loss: 1.3836 -
val_accuracy: 0.3700 - 6s/epoch - 7ms/step
Epoch 8/20
844/844 - 6s - loss: 1.3939 - accuracy: 0.3792 - val_loss: 1.3364 -
val_accuracy: 0.3937 - 6s/epoch - 7ms/step
Epoch 9/20
844/844 - 6s - loss: 1.3660 - accuracy: 0.3915 - val_loss: 1.3184 -
val_accuracy: 0.4005 - 6s/epoch - 7ms/step
Epoch 10/20
844/844 - 6s - loss: 1.3469 - accuracy: 0.4012 - val_loss: 1.2996 -
val_accuracy: 0.4122 - 6s/epoch - 7ms/step
Epoch 11/20
844/844 - 6s - loss: 1.3329 - accuracy: 0.4114 - val_loss: 1.2876 -
val_accuracy: 0.4500 - 6s/epoch - 7ms/step
Epoch 12/20
844/844 - 5s - loss: 1.3184 - accuracy: 0.4278 - val_loss: 1.2812 -
val_accuracy: 0.4715 - 5s/epoch - 6ms/step
Epoch 13/20
844/844 - 5s - loss: 1.3048 - accuracy: 0.4419 - val_loss: 1.2554 -
val_accuracy: 0.4527 - 5s/epoch - 6ms/step
Epoch 14/20
844/844 - 6s - loss: 1.2926 - accuracy: 0.4552 - val_loss: 1.2434 -
val_accuracy: 0.4798 - 6s/epoch - 7ms/step
Epoch 15/20
844/844 - 6s - loss: 1.2784 - accuracy: 0.4743 - val_loss: 1.2264 -
val_accuracy: 0.4910 - 6s/epoch - 7ms/step
Epoch 16/20
844/844 - 6s - loss: 1.2589 - accuracy: 0.4939 - val_loss: 1.2105 -
val_accuracy: 0.4998 - 6s/epoch - 7ms/step
Epoch 17/20
844/844 - 6s - loss: 1.2371 - accuracy: 0.5142 - val_loss: 1.1787 -
val_accuracy: 0.5375 - 6s/epoch - 7ms/step
Epoch 18/20
844/844 - 6s - loss: 1.2077 - accuracy: 0.5389 - val_loss: 1.1620 -
val_accuracy: 0.5718 - 6s/epoch - 7ms/step
Epoch 19/20
844/844 - 6s - loss: 1.1806 - accuracy: 0.5606 - val_loss: 1.1357 -
val_accuracy: 0.5725 - 6s/epoch - 7ms/step
Epoch 20/20
844/844 - 6s - loss: 1.1410 - accuracy: 0.5867 - val_loss: 1.0852 -
val_accuracy: 0.6083 - 6s/epoch - 7ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 913   0   0   7   0   47   2   1   4   6]
 [  0 1107  18   1   4   0   5   0   0   0]
 [  4  607 165  17  32   7 164   5  27   4]
 [ 89   1   7 367  12 230  68   9 196  31]
 [  0  16  23   2 639   1   4 231   2  64]]
```

```
[ 352   4   3 135   5 303  16    0   53   21]
[  7 108 264  45   7  13 403    1 110    0]
[  1 15   5   1  50   0    1 941    0 14]
[ 25 14   35 237  25  62 223    2 324  27]
[ 10   0   2  12  45  18    0 153    3 766]]
```

Precision: 0.5672

Recall: 0.5928



Training Model with softmax activation, 1 conv_layers, 1 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 5s - loss: 2.3012 - accuracy: 0.1126 - val_loss: 2.3013 - val_accuracy: 0.1050 - 5s/epoch - 11ms/step

Epoch 2/5

422/422 - 4s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3003 - val_accuracy: 0.1050 - 4s/epoch - 9ms/step

Epoch 3/5

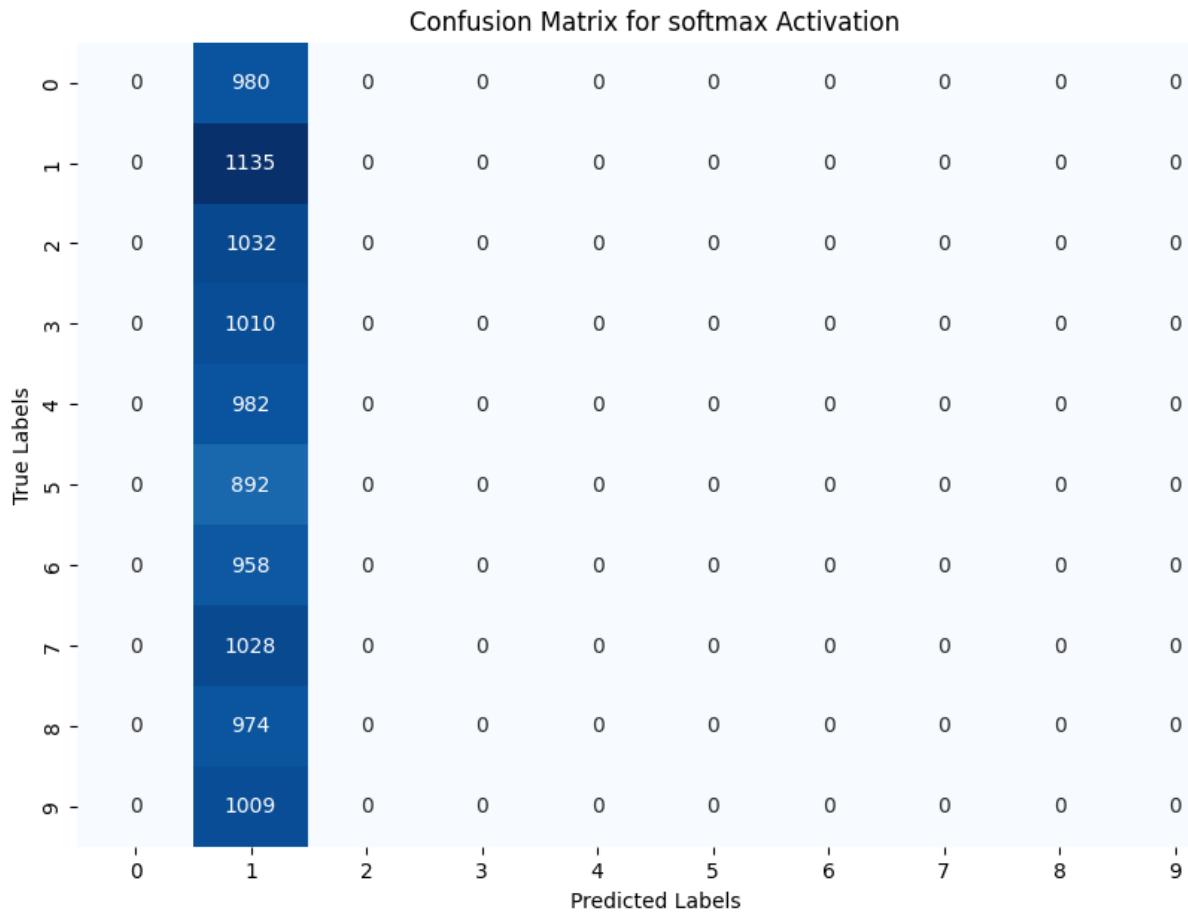
422/422 - 4s - loss: 2.2996 - accuracy: 0.1132 - val_loss: 2.2997 - val_accuracy: 0.1050 - 4s/epoch - 9ms/step

Epoch 4/5

422/422 - 4s - loss: 2.2985 - accuracy: 0.1132 - val_loss: 2.2988 -

```
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 5/5
422/422 - 4s - loss: 2.2969 - accuracy: 0.1132 - val_loss: 2.2962 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 4s - loss: 2.3013 - accuracy: 0.1120 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 4s/epoch - 11ms/step
Epoch 2/15
422/422 - 4s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3010 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 3/15
422/422 - 4s - loss: 2.3003 - accuracy: 0.1132 - val_loss: 2.3004 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 4/15
422/422 - 4s - loss: 2.2998 - accuracy: 0.1132 - val_loss: 2.3001 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 5/15
422/422 - 4s - loss: 2.2993 - accuracy: 0.1132 - val_loss: 2.2996 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 6/15
422/422 - 4s - loss: 2.2987 - accuracy: 0.1132 - val_loss: 2.2992 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 7/15
```

```
422/422 - 4s - loss: 2.2980 - accuracy: 0.1132 - val_loss: 2.2984 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 8/15  
422/422 - 4s - loss: 2.2970 - accuracy: 0.1132 - val_loss: 2.2966 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 9/15  
422/422 - 4s - loss: 2.2956 - accuracy: 0.1132 - val_loss: 2.2950 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 10/15  
422/422 - 4s - loss: 2.2931 - accuracy: 0.1132 - val_loss: 2.2910 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 11/15  
422/422 - 4s - loss: 2.2692 - accuracy: 0.1636 - val_loss: 2.2098 -  
val_accuracy: 0.2037 - 4s/epoch - 9ms/step  
Epoch 12/15  
422/422 - 4s - loss: 2.0862 - accuracy: 0.2538 - val_loss: 1.9152 -  
val_accuracy: 0.3172 - 4s/epoch - 9ms/step  
Epoch 13/15  
422/422 - 4s - loss: 1.8026 - accuracy: 0.3271 - val_loss: 1.6879 -  
val_accuracy: 0.3277 - 4s/epoch - 9ms/step  
Epoch 14/15  
422/422 - 4s - loss: 1.6410 - accuracy: 0.3292 - val_loss: 1.5758 -  
val_accuracy: 0.3270 - 4s/epoch - 9ms/step  
Epoch 15/15  
422/422 - 4s - loss: 1.5507 - accuracy: 0.3314 - val_loss: 1.5159 -  
val_accuracy: 0.3487 - 4s/epoch - 9ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 793  0  0  176  3  0  0  1  7  0]  
[  1 1120  0  14  0  0  0  0  0  0]  
[ 515  15  0  438  25  0  0  14  25  0]  
[ 606  1  0  353  16  0  0  8  26  0]  
[  2  12  0  14  111  0  0  812  31  0]  
[ 427  5  0  402  23  0  0  3  32  0]  
[ 497  5  0  418  9  0  0  3  26  0]  
[  1  22  0  18  44  0  0  928  15  0]  
[ 327  14  0  537  33  0  0  15  48  0]  
[  6  9  0  24  47  0  0  915  8  0]]  
Precision: 0.2369  
Recall: 0.3353  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for softmax Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	793	0	0	176	3	0	0	1	7	0	
0	793	0	0	176	3	0	0	1	7	0	
1	1	1120	0	14	0	0	0	0	0	0	
2	515	15	0	438	25	0	0	14	25	0	
3	606	1	0	353	16	0	0	8	26	0	
4	2	12	0	14	111	0	0	812	31	0	
5	427	5	0	402	23	0	0	3	32	0	
6	497	5	0	418	9	0	0	3	26	0	
7	1	22	0	18	44	0	0	928	15	0	
8	327	14	0	537	33	0	0	15	48	0	
9	6	9	0	24	47	0	0	915	8	0	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	

```

Training Model with softmax activation, 1 conv_layers, 1 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 4s - loss: 2.3012 - accuracy: 0.1122 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 2/20
422/422 - 4s - loss: 2.3004 - accuracy: 0.1132 - val_loss: 2.3008 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 3/20
422/422 - 4s - loss: 2.2997 - accuracy: 0.1132 - val_loss: 2.3000 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 4/20
422/422 - 4s - loss: 2.2990 - accuracy: 0.1132 - val_loss: 2.2993 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 5/20
422/422 - 4s - loss: 2.2981 - accuracy: 0.1132 - val_loss: 2.2979 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 6/20
422/422 - 4s - loss: 2.2969 - accuracy: 0.1132 - val_loss: 2.2966 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 7/20

```

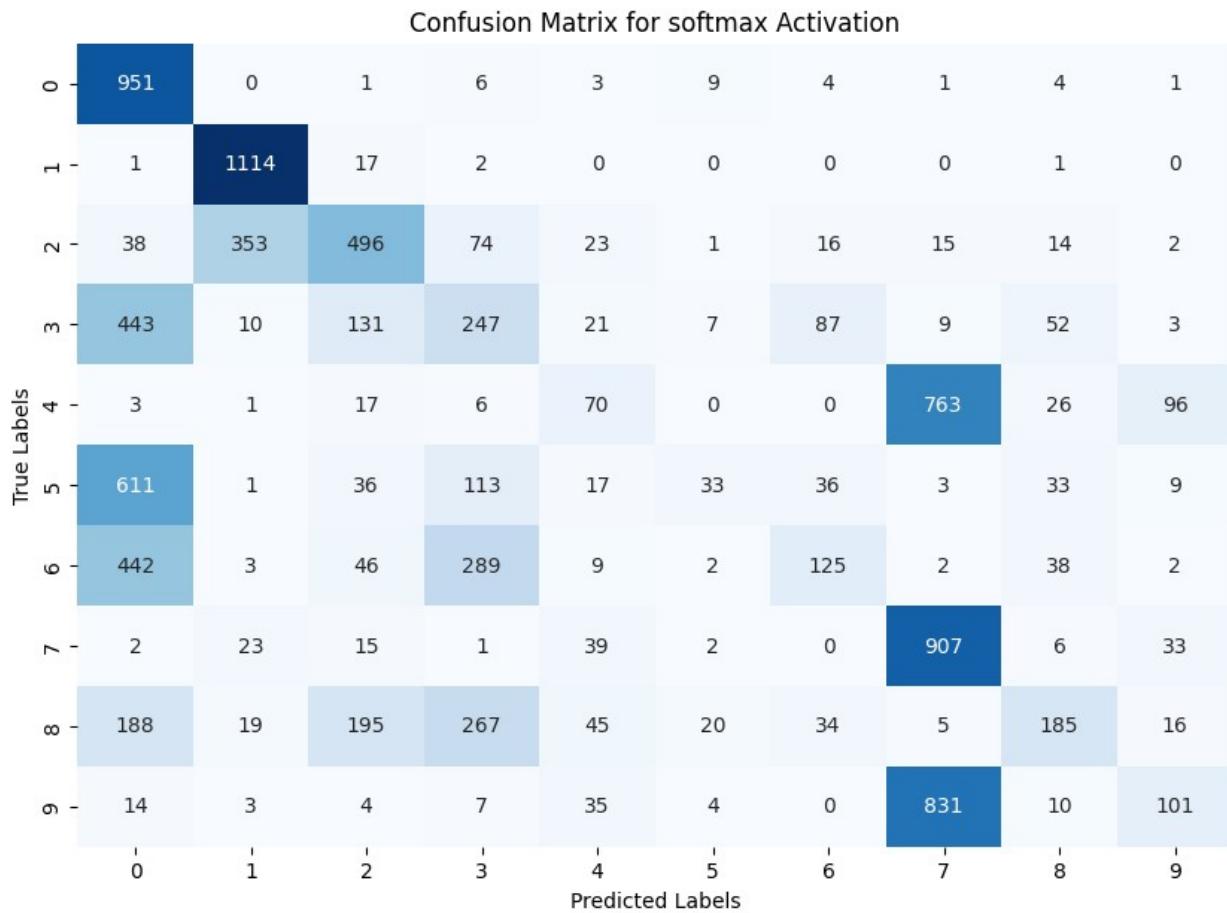
```

422/422 - 4s - loss: 2.2951 - accuracy: 0.1132 - val_loss: 2.2941 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 8/20
422/422 - 4s - loss: 2.2921 - accuracy: 0.1132 - val_loss: 2.2900 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 9/20
422/422 - 4s - loss: 2.2862 - accuracy: 0.1154 - val_loss: 2.2810 -
val_accuracy: 0.1050 - 4s/epoch - 9ms/step
Epoch 10/20
422/422 - 4s - loss: 2.2613 - accuracy: 0.2107 - val_loss: 2.1898 -
val_accuracy: 0.3087 - 4s/epoch - 9ms/step
Epoch 11/20
422/422 - 4s - loss: 1.9661 - accuracy: 0.3167 - val_loss: 1.7664 -
val_accuracy: 0.3282 - 4s/epoch - 9ms/step
Epoch 12/20
422/422 - 4s - loss: 1.6633 - accuracy: 0.3257 - val_loss: 1.5763 -
val_accuracy: 0.3283 - 4s/epoch - 9ms/step
Epoch 13/20
422/422 - 4s - loss: 1.5619 - accuracy: 0.3335 - val_loss: 1.5277 -
val_accuracy: 0.3370 - 4s/epoch - 9ms/step
Epoch 14/20
422/422 - 4s - loss: 1.5284 - accuracy: 0.3396 - val_loss: 1.5061 -
val_accuracy: 0.3338 - 4s/epoch - 9ms/step
Epoch 15/20
422/422 - 4s - loss: 1.5107 - accuracy: 0.3454 - val_loss: 1.4898 -
val_accuracy: 0.3603 - 4s/epoch - 9ms/step
Epoch 16/20
422/422 - 4s - loss: 1.4964 - accuracy: 0.3575 - val_loss: 1.4748 -
val_accuracy: 0.3568 - 4s/epoch - 9ms/step
Epoch 17/20
422/422 - 4s - loss: 1.4796 - accuracy: 0.3694 - val_loss: 1.4526 -
val_accuracy: 0.3802 - 4s/epoch - 9ms/step
Epoch 18/20
422/422 - 4s - loss: 1.4537 - accuracy: 0.3873 - val_loss: 1.4232 -
val_accuracy: 0.4203 - 4s/epoch - 9ms/step
Epoch 19/20
422/422 - 4s - loss: 1.4235 - accuracy: 0.4020 - val_loss: 1.3862 -
val_accuracy: 0.4170 - 4s/epoch - 9ms/step
Epoch 20/20
422/422 - 4s - loss: 1.3970 - accuracy: 0.4119 - val_loss: 1.3602 -
val_accuracy: 0.4420 - 4s/epoch - 9ms/step
313/313 [=====] - 1s 2ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 951   0   1   6   3   9   4   1   4   1]
 [  1 1114  17   2   0   0   0   0   1   0]
 [ 38  353  496  74  23   1  16  15  14   2]
 [ 443   10  131  247  21   7  87   9  52   3]
 [  3    1   17    6   70   0   0  763  26  96]]
```

```
[ 611   1   36  113   17   33   36    3   33   9]
[ 442   3   46  289    9    2  125    2   38   2]
[  2   23   15    1   39    2    0  907    6   33]
[ 188   19  195  267   45   20   34    5  185   16]
[ 14    3    4    7   35    4    0  831   10  101]]
```

Precision: 0.4235

Recall: 0.4229



Training Model with softmax activation, 1 conv_layers, 1 dense layers, 256 batch size, 5 epochs..

Epoch 1/5

211/211 - 4s - loss: 2.3015 - accuracy: 0.1119 - val_loss: 2.3017 - val_accuracy: 0.1050 - 4s/epoch - 18ms/step

Epoch 2/5

211/211 - 3s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3014 - val_accuracy: 0.1050 - 3s/epoch - 15ms/step

Epoch 3/5

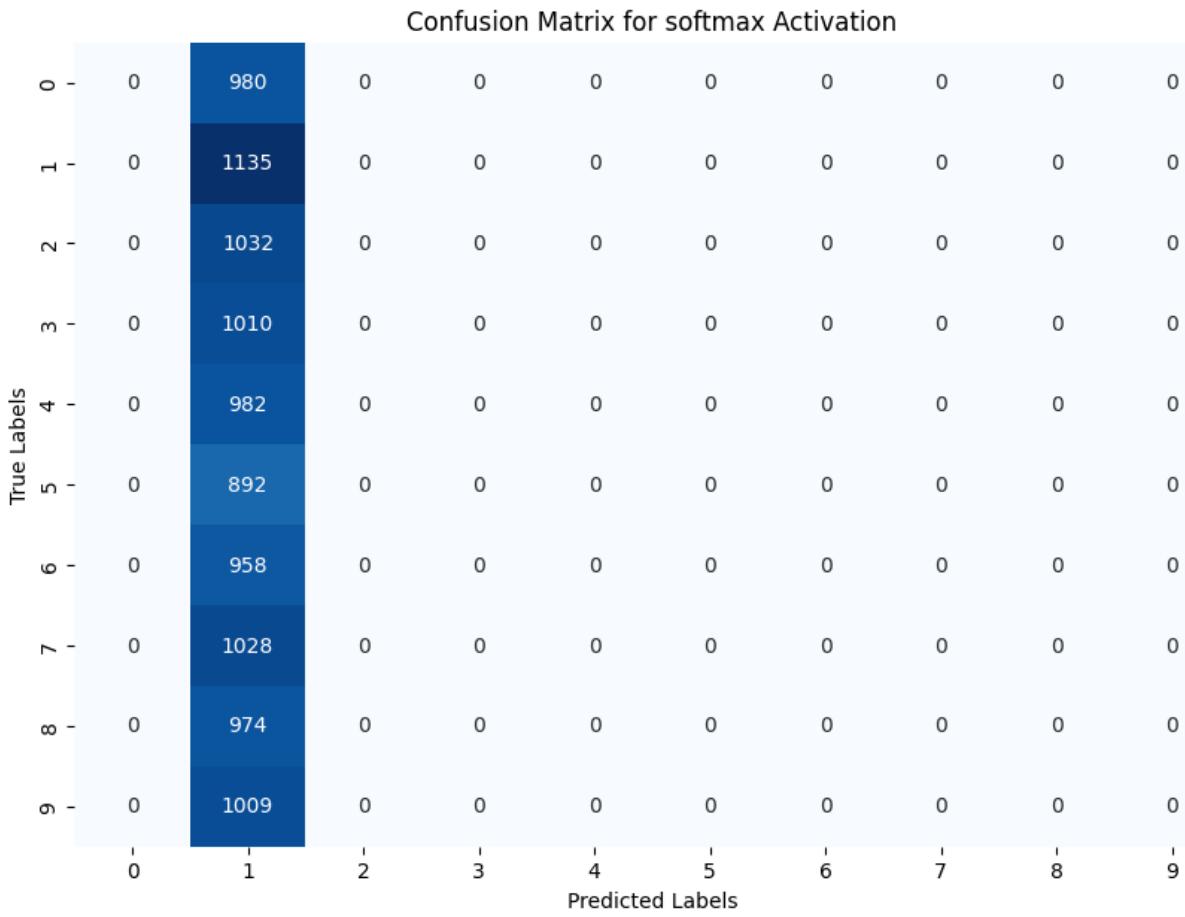
211/211 - 3s - loss: 2.3004 - accuracy: 0.1132 - val_loss: 2.3009 - val_accuracy: 0.1050 - 3s/epoch - 15ms/step

Epoch 4/5

211/211 - 3s - loss: 2.3000 - accuracy: 0.1132 - val_loss: 2.3005 -

```
val_accuracy: 0.1050 - 3s/epoch - 15ms/step
Epoch 5/5
211/211 - 3s - loss: 2.2996 - accuracy: 0.1132 - val_loss: 2.3000 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 4s - loss: 2.3014 - accuracy: 0.1112 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 2/15
211/211 - 3s - loss: 2.3006 - accuracy: 0.1132 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 3s/epoch - 15ms/step
Epoch 3/15
211/211 - 3s - loss: 2.3003 - accuracy: 0.1132 - val_loss: 2.3008 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 4/15
211/211 - 3s - loss: 2.2999 - accuracy: 0.1132 - val_loss: 2.3004 -
val_accuracy: 0.1050 - 3s/epoch - 15ms/step
Epoch 5/15
211/211 - 3s - loss: 2.2995 - accuracy: 0.1132 - val_loss: 2.2997 -
val_accuracy: 0.1050 - 3s/epoch - 15ms/step
Epoch 6/15
211/211 - 3s - loss: 2.2990 - accuracy: 0.1132 - val_loss: 2.2995 -
val_accuracy: 0.1050 - 3s/epoch - 15ms/step
Epoch 7/15
```

```
211/211 - 3s - loss: 2.2984 - accuracy: 0.1132 - val_loss: 2.2986 -  
val_accuracy: 0.1050 - 3s/epoch - 15ms/step  
Epoch 8/15  
211/211 - 3s - loss: 2.2977 - accuracy: 0.1132 - val_loss: 2.2978 -  
val_accuracy: 0.1050 - 3s/epoch - 15ms/step  
Epoch 9/15  
211/211 - 3s - loss: 2.2968 - accuracy: 0.1132 - val_loss: 2.2968 -  
val_accuracy: 0.1050 - 3s/epoch - 15ms/step  
Epoch 10/15  
211/211 - 3s - loss: 2.2955 - accuracy: 0.1132 - val_loss: 2.2950 -  
val_accuracy: 0.1050 - 3s/epoch - 15ms/step  
Epoch 11/15  
211/211 - 3s - loss: 2.2936 - accuracy: 0.1132 - val_loss: 2.2926 -  
val_accuracy: 0.1050 - 3s/epoch - 15ms/step  
Epoch 12/15  
211/211 - 3s - loss: 2.2904 - accuracy: 0.1132 - val_loss: 2.2883 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 13/15  
211/211 - 3s - loss: 2.2822 - accuracy: 0.1352 - val_loss: 2.2677 -  
val_accuracy: 0.1732 - 3s/epoch - 15ms/step  
Epoch 14/15  
211/211 - 3s - loss: 2.2263 - accuracy: 0.2172 - val_loss: 2.1680 -  
val_accuracy: 0.2312 - 3s/epoch - 15ms/step  
Epoch 15/15  
211/211 - 3s - loss: 2.1078 - accuracy: 0.2249 - val_loss: 2.0218 -  
val_accuracy: 0.2177 - 3s/epoch - 15ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 964  12   0   2   0   0   1   0   1   0 ]  
[   9 1121   0   1   0   0   0   0   4   0 ]  
[ 919  86   0  20   0   0   3   0   4   0 ]  
[ 829 131   0  39   0   0   1   0  10   0 ]  
[  26 947   0   4   0   0   5   0   0   0 ]  
[ 498 361   0  27   0   0   0   0   6   0 ]  
[ 752 161   0  25   0   0  10   0  10   0 ]  
[  25 958   0   3   0   0   0  39   3   0 ]  
[ 401 464   0  63   0   0   0   0  46   0 ]  
[  37 964   0   4   0   0   1   0   3   0 ]]  
Precision: 0.2665  
Recall: 0.2219  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	964	12	0	2	0	0	1	0	1	0
	9	1121	0	1	0	0	0	0	4	0
	919	86	0	20	0	0	3	0	4	0
	829	131	0	39	0	0	1	0	10	0
	26	947	0	4	0	0	5	0	0	0
	498	361	0	27	0	0	0	0	6	0
	752	161	0	25	0	0	10	0	10	0
	25	958	0	3	0	0	0	39	3	0
	401	464	0	63	0	0	0	0	46	0
	37	964	0	4	0	0	1	0	3	0

```

Training Model with softmax activation, 1 conv_layers, 1 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 4s - loss: 2.3017 - accuracy: 0.1093 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 2/20
211/211 - 3s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 3/20
211/211 - 3s - loss: 2.3002 - accuracy: 0.1132 - val_loss: 2.3007 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 4/20
211/211 - 3s - loss: 2.2997 - accuracy: 0.1132 - val_loss: 2.3000 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 5/20
211/211 - 4s - loss: 2.2991 - accuracy: 0.1132 - val_loss: 2.2994 -
val_accuracy: 0.1050 - 4s/epoch - 17ms/step
Epoch 6/20
211/211 - 3s - loss: 2.2985 - accuracy: 0.1132 - val_loss: 2.2988 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 7/20

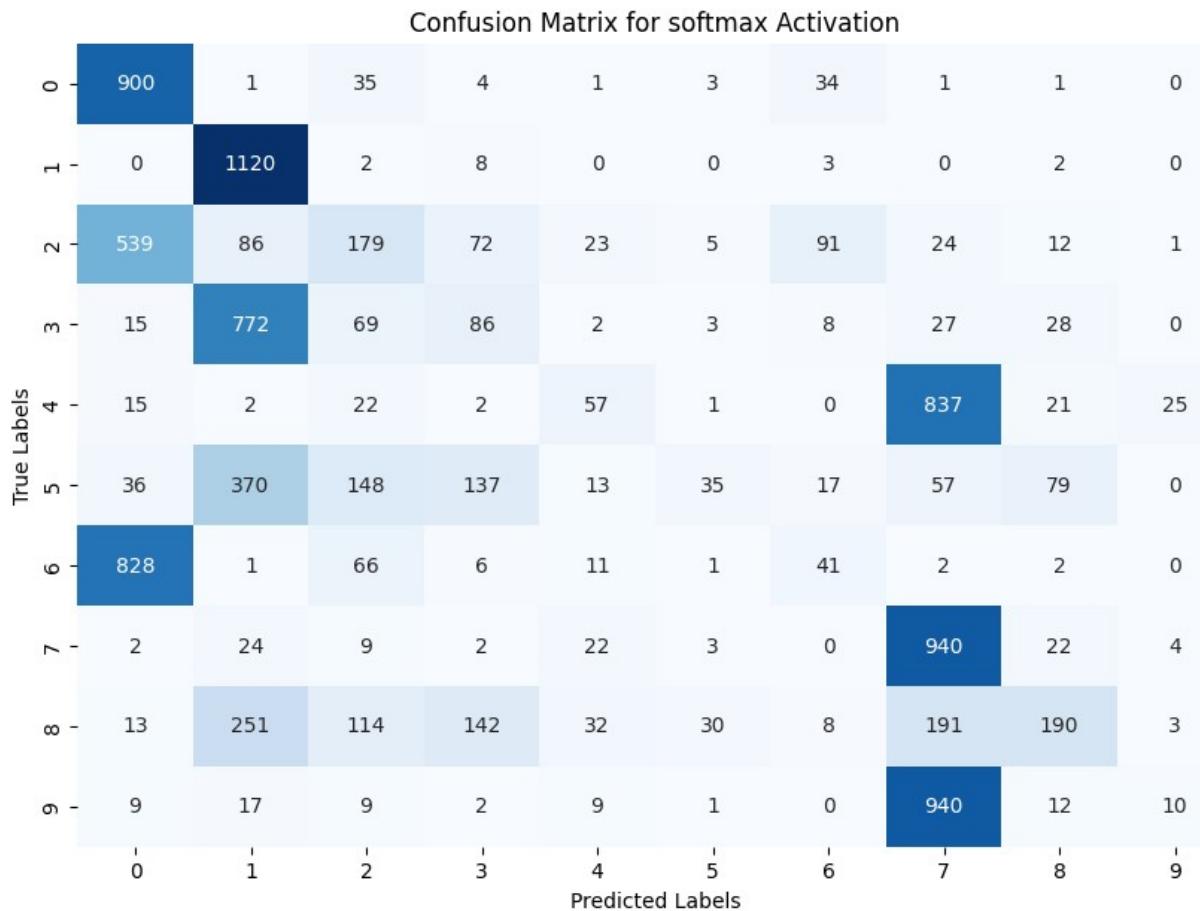
```

```
211/211 - 3s - loss: 2.2977 - accuracy: 0.1132 - val_loss: 2.2978 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 8/20  
211/211 - 3s - loss: 2.2967 - accuracy: 0.1132 - val_loss: 2.2968 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 9/20  
211/211 - 3s - loss: 2.2954 - accuracy: 0.1132 - val_loss: 2.2949 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 10/20  
211/211 - 3s - loss: 2.2935 - accuracy: 0.1132 - val_loss: 2.2926 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 11/20  
211/211 - 3s - loss: 2.2908 - accuracy: 0.1132 - val_loss: 2.2892 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 12/20  
211/211 - 3s - loss: 2.2862 - accuracy: 0.1138 - val_loss: 2.2829 -  
val_accuracy: 0.1058 - 3s/epoch - 15ms/step  
Epoch 13/20  
211/211 - 3s - loss: 2.2741 - accuracy: 0.1680 - val_loss: 2.2560 -  
val_accuracy: 0.1992 - 3s/epoch - 15ms/step  
Epoch 14/20  
211/211 - 3s - loss: 2.2016 - accuracy: 0.2505 - val_loss: 2.1087 -  
val_accuracy: 0.3013 - 3s/epoch - 15ms/step  
Epoch 15/20  
211/211 - 3s - loss: 2.0209 - accuracy: 0.3225 - val_loss: 1.9300 -  
val_accuracy: 0.3203 - 3s/epoch - 15ms/step  
Epoch 16/20  
211/211 - 3s - loss: 1.8706 - accuracy: 0.3309 - val_loss: 1.7989 -  
val_accuracy: 0.3235 - 3s/epoch - 15ms/step  
Epoch 17/20  
211/211 - 3s - loss: 1.7460 - accuracy: 0.3290 - val_loss: 1.6677 -  
val_accuracy: 0.3235 - 3s/epoch - 15ms/step  
Epoch 18/20  
211/211 - 3s - loss: 1.6276 - accuracy: 0.3333 - val_loss: 1.5644 -  
val_accuracy: 0.3302 - 3s/epoch - 16ms/step  
Epoch 19/20  
211/211 - 3s - loss: 1.5504 - accuracy: 0.3455 - val_loss: 1.5008 -  
val_accuracy: 0.3537 - 3s/epoch - 16ms/step  
Epoch 20/20  
211/211 - 3s - loss: 1.5023 - accuracy: 0.3528 - val_loss: 1.4602 -  
val_accuracy: 0.3575 - 3s/epoch - 16ms/step  
313/313 [=====] - 1s 2ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 900 1 35 4 1 3 34 1 1 0]  
 [ 0 1120 2 8 0 0 3 0 2 0]  
 [ 539 86 179 72 23 5 91 24 12 1]  
 [ 15 772 69 86 2 3 8 27 28 0]  
 [ 15 2 22 2 57 1 0 837 21 25]]
```

```
[ 36 370 148 137 13 35 17 57 79 0]
[ 828 1 66 6 11 1 41 2 2 0]
[ 2 24 9 2 22 3 0 940 22 4]
[ 13 251 114 142 32 30 8 191 190 3]
[ 9 17 9 2 9 1 0 940 12 10]]
```

Precision: 0.3287

Recall: 0.3558



Training Model with softmax activation, 1 conv_layers, 1 dense layers, 64 batch size, 5 epochs..

Epoch 1/5

844/844 - 7s - loss: 2.3014 - accuracy: 0.1119 - val_loss: 2.3020 - val_accuracy: 0.1050 - 7s/epoch - 8ms/step

Epoch 2/5

844/844 - 6s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3013 - val_accuracy: 0.1050 - 6s/epoch - 7ms/step

Epoch 3/5

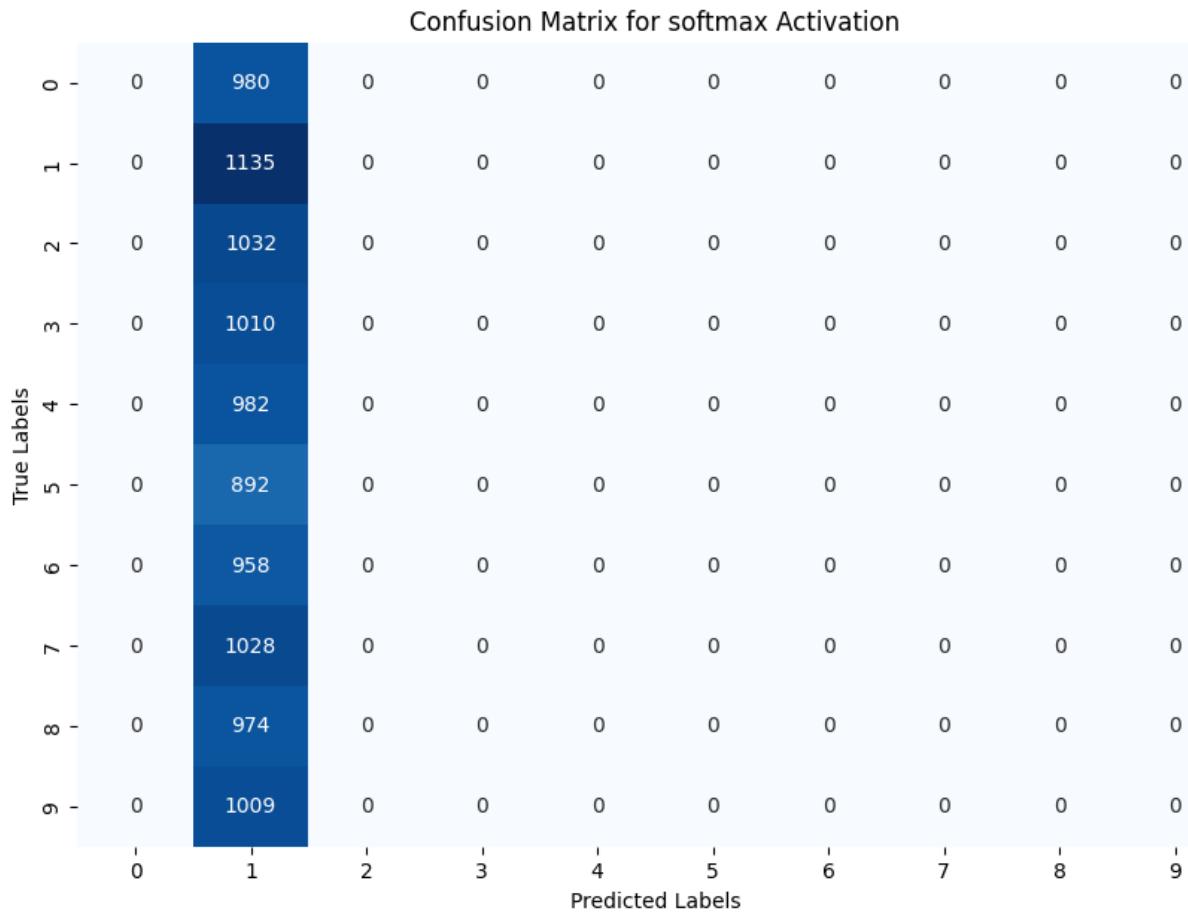
844/844 - 6s - loss: 2.3002 - accuracy: 0.1135 - val_loss: 2.3001 - val_accuracy: 0.1050 - 6s/epoch - 7ms/step

Epoch 4/5

844/844 - 6s - loss: 2.2993 - accuracy: 0.1132 - val_loss: 2.2991 -

```
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 5/5
844/844 - 6s - loss: 2.2979 - accuracy: 0.1132 - val_loss: 2.2970 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 7s - loss: 2.3016 - accuracy: 0.1129 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 2/15
844/844 - 6s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 3/15
844/844 - 6s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 4/15
844/844 - 6s - loss: 2.3006 - accuracy: 0.1126 - val_loss: 2.3009 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 5/15
844/844 - 6s - loss: 2.3001 - accuracy: 0.1133 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 6/15
844/844 - 6s - loss: 2.2995 - accuracy: 0.1132 - val_loss: 2.2996 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 7/15
```

```
844/844 - 6s - loss: 2.2978 - accuracy: 0.1144 - val_loss: 2.2984 -  
val_accuracy: 0.1050 - 6s/epoch - 7ms/step  
Epoch 8/15  
844/844 - 6s - loss: 2.2172 - accuracy: 0.1883 - val_loss: 2.0624 -  
val_accuracy: 0.2245 - 6s/epoch - 7ms/step  
Epoch 9/15  
844/844 - 6s - loss: 1.9380 - accuracy: 0.2507 - val_loss: 1.7853 -  
val_accuracy: 0.3355 - 6s/epoch - 7ms/step  
Epoch 10/15  
844/844 - 6s - loss: 1.6758 - accuracy: 0.3358 - val_loss: 1.5779 -  
val_accuracy: 0.3278 - 6s/epoch - 7ms/step  
Epoch 11/15  
844/844 - 6s - loss: 1.5612 - accuracy: 0.3428 - val_loss: 1.5126 -  
val_accuracy: 0.3520 - 6s/epoch - 7ms/step  
Epoch 12/15  
844/844 - 6s - loss: 1.5084 - accuracy: 0.3583 - val_loss: 1.4714 -  
val_accuracy: 0.3693 - 6s/epoch - 7ms/step  
Epoch 13/15  
844/844 - 6s - loss: 1.4776 - accuracy: 0.3697 - val_loss: 1.4422 -  
val_accuracy: 0.3895 - 6s/epoch - 7ms/step  
Epoch 14/15  
844/844 - 6s - loss: 1.4582 - accuracy: 0.3783 - val_loss: 1.4223 -  
val_accuracy: 0.3953 - 6s/epoch - 7ms/step  
Epoch 15/15  
844/844 - 6s - loss: 1.4373 - accuracy: 0.3901 - val_loss: 1.3998 -  
val_accuracy: 0.4093 - 6s/epoch - 7ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 866 0 45 49 0 0 1 1 17 1]  
[ 2 1119 3 1 0 0 3 1 6 0]  
[ 624 19 119 98 0 0 53 17 90 12]  
[ 331 7 150 235 4 0 46 20 207 10]  
[ 6 3 2 6 6 0 0 887 52 20]  
[ 221 12 137 291 1 0 28 15 179 8]  
[ 782 6 79 39 0 0 26 2 24 0]  
[ 1 14 4 4 4 0 0 944 41 16]  
[ 39 26 26 116 10 0 3 76 607 71]  
[ 6 8 2 9 2 0 0 937 30 15]]  
Precision: 0.3117  
Recall: 0.3937  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for softmax Activation											
	0	1	2	3	4	5	6	7	8	9	
True Labels	0	866	0	45	49	0	0	1	1	17	1
0	866	0	45	49	0	0	1	1	17	1	
1	2	1119	3	1	0	0	3	1	6	0	
2	624	19	119	98	0	0	53	17	90	12	
3	331	7	150	235	4	0	46	20	207	10	
4	6	3	2	6	6	0	0	887	52	20	
5	221	12	137	291	1	0	28	15	179	8	
6	782	6	79	39	0	0	26	2	24	0	
7	1	14	4	4	4	0	0	944	41	16	
8	39	26	26	116	10	0	3	76	607	71	
9	6	8	2	9	2	0	0	937	30	15	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

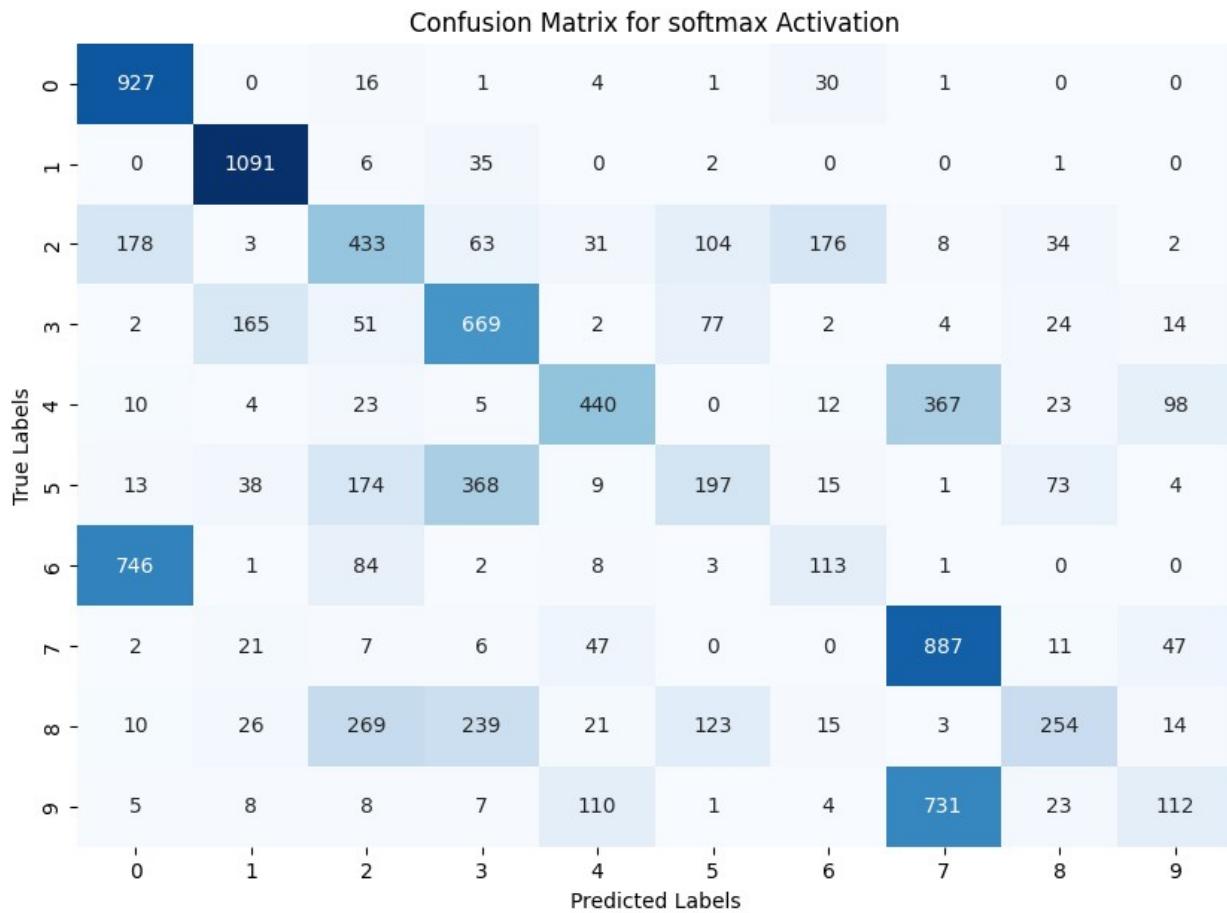
```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 7s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 2/20
844/844 - 6s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3011 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 3/20
844/844 - 6s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3011 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 4/20
844/844 - 6s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3004 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 5/20
844/844 - 6s - loss: 2.3000 - accuracy: 0.1132 - val_loss: 2.3002 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 6/20
844/844 - 6s - loss: 2.2991 - accuracy: 0.1133 - val_loss: 2.2991 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 7/20
```

```
844/844 - 6s - loss: 2.2974 - accuracy: 0.1158 - val_loss: 2.2964 -  
val_accuracy: 0.1050 - 6s/epoch - 7ms/step  
Epoch 8/20  
844/844 - 6s - loss: 2.2938 - accuracy: 0.1132 - val_loss: 2.2911 -  
val_accuracy: 0.1072 - 6s/epoch - 7ms/step  
Epoch 9/20  
844/844 - 6s - loss: 2.2677 - accuracy: 0.1674 - val_loss: 2.1710 -  
val_accuracy: 0.2020 - 6s/epoch - 7ms/step  
Epoch 10/20  
844/844 - 6s - loss: 1.9228 - accuracy: 0.2812 - val_loss: 1.6052 -  
val_accuracy: 0.3235 - 6s/epoch - 7ms/step  
Epoch 11/20  
844/844 - 6s - loss: 1.5064 - accuracy: 0.3523 - val_loss: 1.4140 -  
val_accuracy: 0.3608 - 6s/epoch - 7ms/step  
Epoch 12/20  
844/844 - 6s - loss: 1.4137 - accuracy: 0.3689 - val_loss: 1.3598 -  
val_accuracy: 0.3870 - 6s/epoch - 7ms/step  
Epoch 13/20  
844/844 - 6s - loss: 1.3730 - accuracy: 0.3886 - val_loss: 1.3268 -  
val_accuracy: 0.4062 - 6s/epoch - 7ms/step  
Epoch 14/20  
844/844 - 6s - loss: 1.3450 - accuracy: 0.4076 - val_loss: 1.3101 -  
val_accuracy: 0.4078 - 6s/epoch - 7ms/step  
Epoch 15/20  
844/844 - 6s - loss: 1.3231 - accuracy: 0.4209 - val_loss: 1.2969 -  
val_accuracy: 0.4352 - 6s/epoch - 7ms/step  
Epoch 16/20  
844/844 - 6s - loss: 1.3045 - accuracy: 0.4346 - val_loss: 1.2635 -  
val_accuracy: 0.4398 - 6s/epoch - 7ms/step  
Epoch 17/20  
844/844 - 6s - loss: 1.2868 - accuracy: 0.4446 - val_loss: 1.2642 -  
val_accuracy: 0.4773 - 6s/epoch - 7ms/step  
Epoch 18/20  
844/844 - 6s - loss: 1.2722 - accuracy: 0.4591 - val_loss: 1.2593 -  
val_accuracy: 0.4667 - 6s/epoch - 7ms/step  
Epoch 19/20  
844/844 - 6s - loss: 1.2561 - accuracy: 0.4690 - val_loss: 1.2142 -  
val_accuracy: 0.5117 - 6s/epoch - 7ms/step  
Epoch 20/20  
844/844 - 6s - loss: 1.2345 - accuracy: 0.4899 - val_loss: 1.1960 -  
val_accuracy: 0.5178 - 6s/epoch - 7ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 927   0   16    1    4    1   30    1    0    0]  
 [  0 1091   6   35    0    2    0    0    1    0]  
 [ 178    3  433   63   31  104   176    8   34    2]  
 [  2  165   51  669    2   77    2    4   24   14]  
 [ 10    4   23    5  440    0   12  367   23   98]]
```

```
[ 13  38 174 368  9 197 15  1 73  4]
[ 746  1 84  2  8  3 113  1  0  0]
[ 2 21  7  6 47  0  0 887 11 47]
[ 10 26 269 239 21 123 15  3 254 14]
[ 5  8  8  7 110  1  4 731 23 112]]
```

Precision: 0.4980

Recall: 0.5123



Training Model with softmax activation, 1 conv_layers, 1 dense layers, 128 batch size, 5 epochs..

Epoch 1/5

422/422 - 5s - loss: 2.3014 - accuracy: 0.1113 - val_loss: 2.3025 - val_accuracy: 0.1050 - 5s/epoch - 11ms/step

Epoch 2/5

422/422 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3021 - val_accuracy: 0.1050 - 4s/epoch - 10ms/step

Epoch 3/5

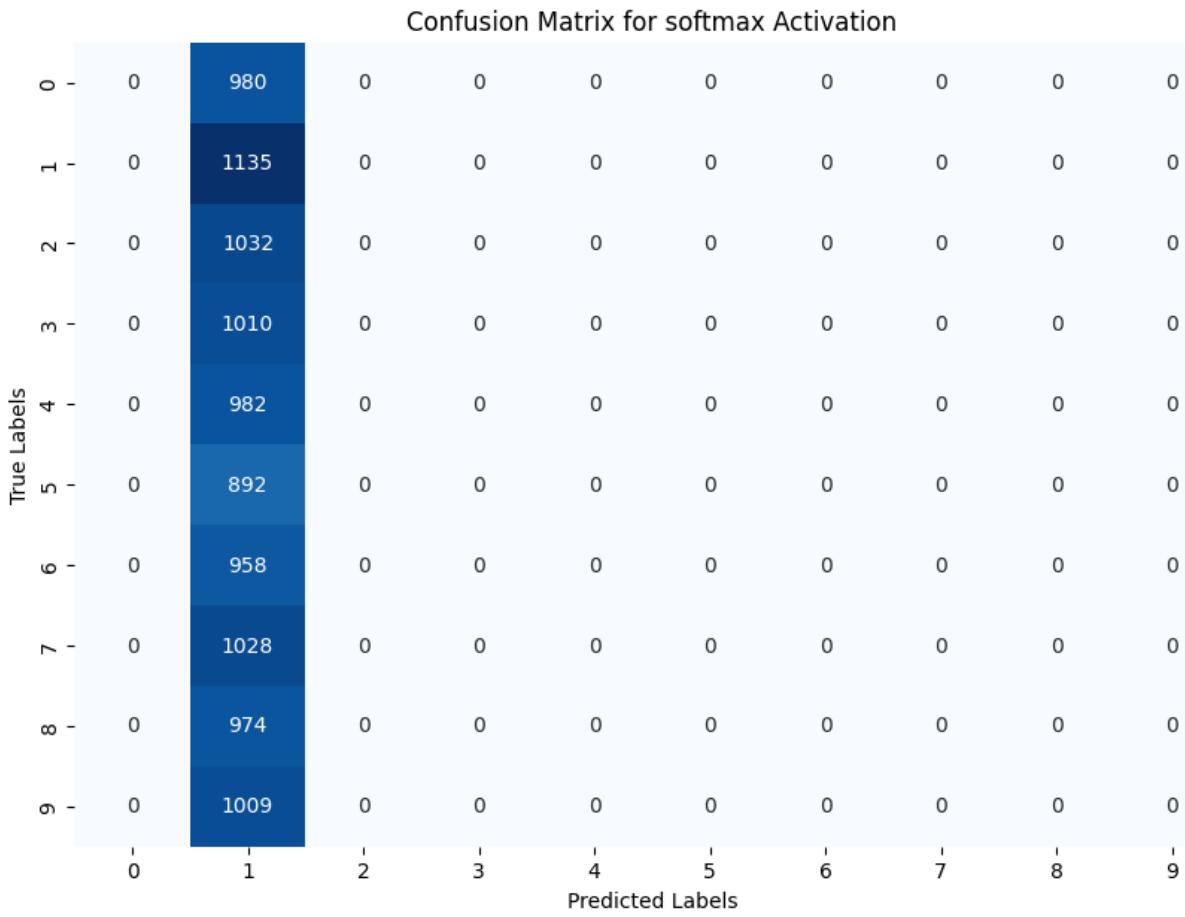
422/422 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3018 - val_accuracy: 0.1050 - 4s/epoch - 10ms/step

Epoch 4/5

422/422 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3016 -

```
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 5/5
422/422 - 4s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

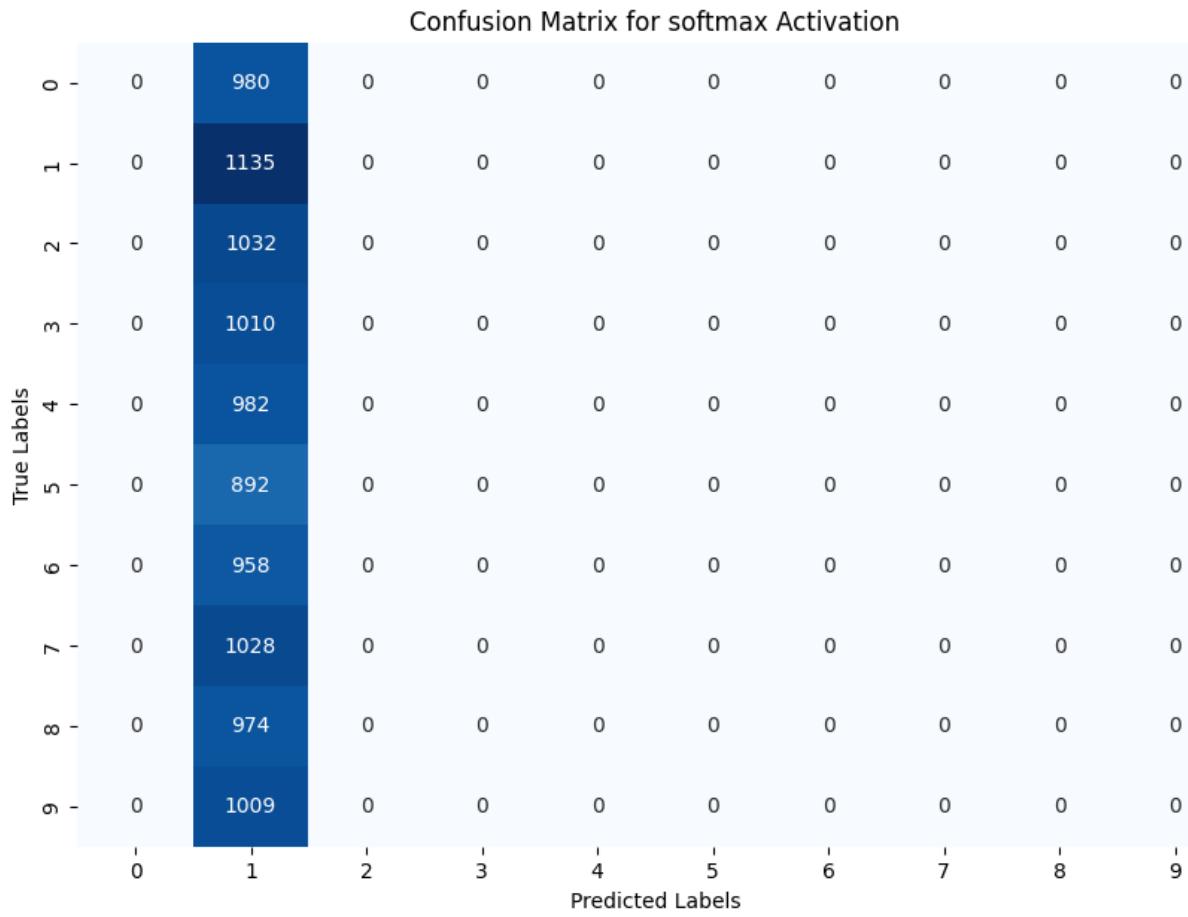


```

Training Model with softmax activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 5s - loss: 2.3014 - accuracy: 0.1122 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 2/15
422/422 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 3/15
422/422 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 4/15
422/422 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3014 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 5/15
422/422 - 4s - loss: 2.3009 - accuracy: 0.1125 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 6/15
422/422 - 4s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 7/15

```

```
422/422 - 4s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 8/15  
422/422 - 4s - loss: 2.3006 - accuracy: 0.1132 - val_loss: 2.3010 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 9/15  
422/422 - 4s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3014 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 10/15  
422/422 - 4s - loss: 2.3004 - accuracy: 0.1132 - val_loss: 2.3008 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 11/15  
422/422 - 4s - loss: 2.3002 - accuracy: 0.1132 - val_loss: 2.3007 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 12/15  
422/422 - 4s - loss: 2.3001 - accuracy: 0.1132 - val_loss: 2.3008 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 13/15  
422/422 - 4s - loss: 2.3000 - accuracy: 0.1132 - val_loss: 2.3004 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 14/15  
422/422 - 4s - loss: 2.2998 - accuracy: 0.1132 - val_loss: 2.3004 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 15/15  
422/422 - 4s - loss: 2.2995 - accuracy: 0.1132 - val_loss: 2.3002 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 5s - loss: 2.3014 - accuracy: 0.1122 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 2/20
422/422 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 3/20
422/422 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 4/20
422/422 - 4s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3014 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 5/20
422/422 - 4s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 6/20
422/422 - 4s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 7/20
```

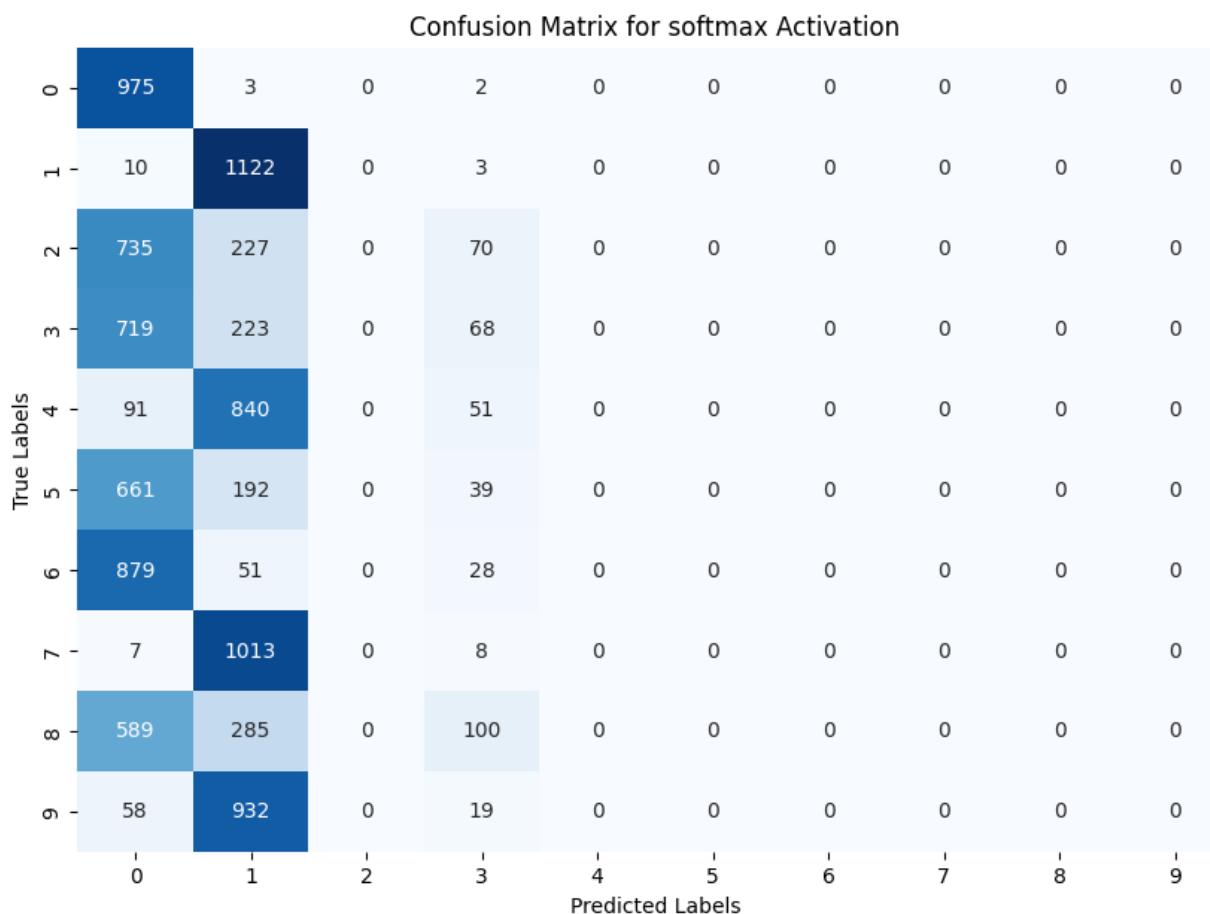
```
422/422 - 4s - loss: 2.3004 - accuracy: 0.1132 - val_loss: 2.3007 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 8/20  
422/422 - 4s - loss: 2.3003 - accuracy: 0.1132 - val_loss: 2.3010 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 9/20  
422/422 - 4s - loss: 2.3001 - accuracy: 0.1132 - val_loss: 2.3005 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 10/20  
422/422 - 4s - loss: 2.2998 - accuracy: 0.1132 - val_loss: 2.3005 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 11/20  
422/422 - 4s - loss: 2.2996 - accuracy: 0.1132 - val_loss: 2.3005 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 12/20  
422/422 - 4s - loss: 2.2993 - accuracy: 0.1132 - val_loss: 2.2996 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 13/20  
422/422 - 4s - loss: 2.2990 - accuracy: 0.1132 - val_loss: 2.2996 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 14/20  
422/422 - 4s - loss: 2.2986 - accuracy: 0.1132 - val_loss: 2.2987 -  
val_accuracy: 0.1050 - 4s/epoch - 9ms/step  
Epoch 15/20  
422/422 - 4s - loss: 2.2981 - accuracy: 0.1132 - val_loss: 2.2981 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 16/20  
422/422 - 4s - loss: 2.2973 - accuracy: 0.1132 - val_loss: 2.2976 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 17/20  
422/422 - 4s - loss: 2.2964 - accuracy: 0.1132 - val_loss: 2.2960 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 18/20  
422/422 - 4s - loss: 2.2948 - accuracy: 0.1132 - val_loss: 2.2942 -  
val_accuracy: 0.1050 - 4s/epoch - 10ms/step  
Epoch 19/20  
422/422 - 4s - loss: 2.2910 - accuracy: 0.1164 - val_loss: 2.2826 -  
val_accuracy: 0.1605 - 4s/epoch - 10ms/step  
Epoch 20/20  
422/422 - 4s - loss: 2.2003 - accuracy: 0.2089 - val_loss: 2.0976 -  
val_accuracy: 0.2085 - 4s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 975   3   0   2   0   0   0   0   0 ]  
 [  10 1122   0   3   0   0   0   0   0 ]  
 [ 735  227   0  70   0   0   0   0   0 ]  
 [ 719  223   0  68   0   0   0   0   0 ]  
 [  91  840   0  51   0   0   0   0   0 ]
```

```
[ 661 192  0  39  0  0  0  0  0  0]
[ 879  51  0  28  0  0  0  0  0  0]
[  7 1013  0   8  0  0  0  0  0  0]
[ 589 285  0 100  0  0  0  0  0  0]
[ 58 932  0  19  0  0  0  0  0  0]]
```

Precision: 0.0640

Recall: 0.2165

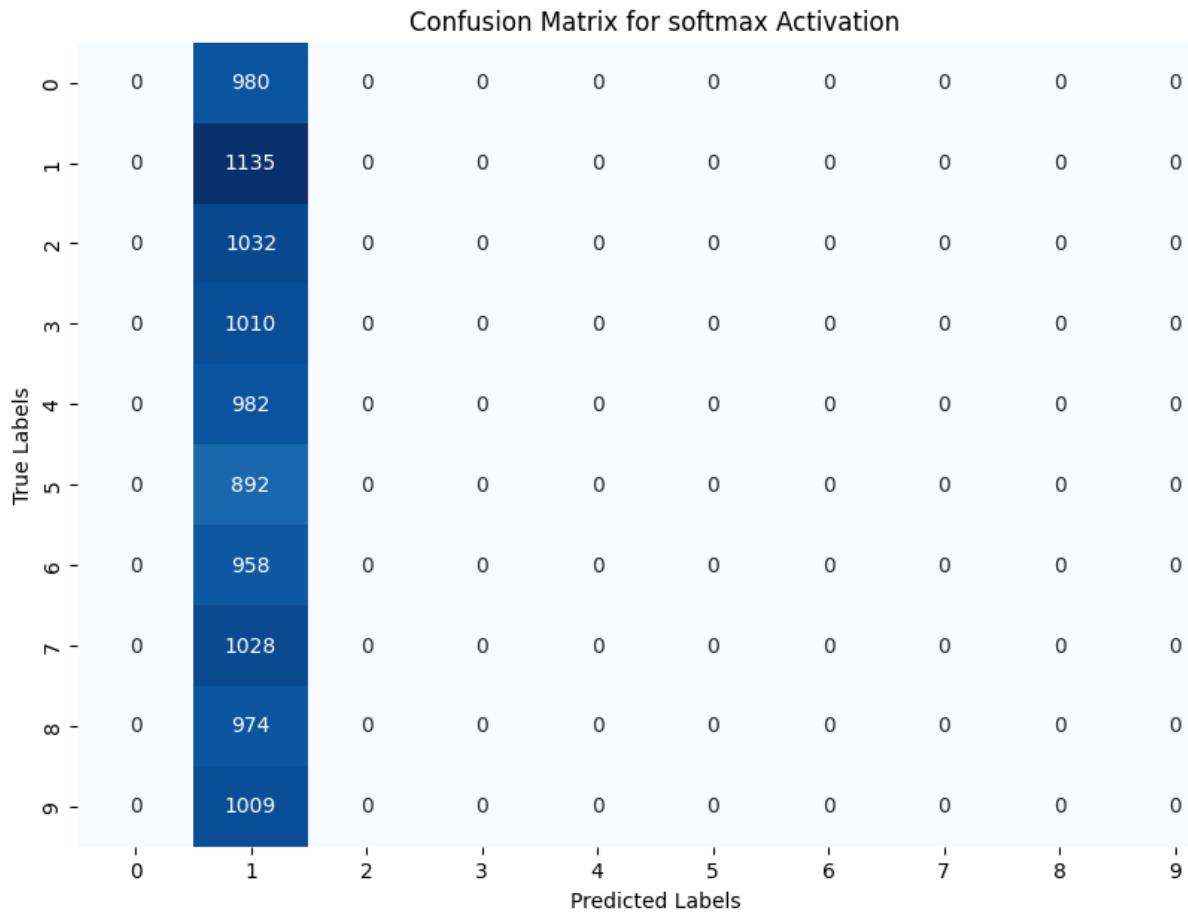
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 1 conv_layers, 1 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 4s - loss: 2.3016 - accuracy: 0.1109 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 2/5
211/211 - 3s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3017 -

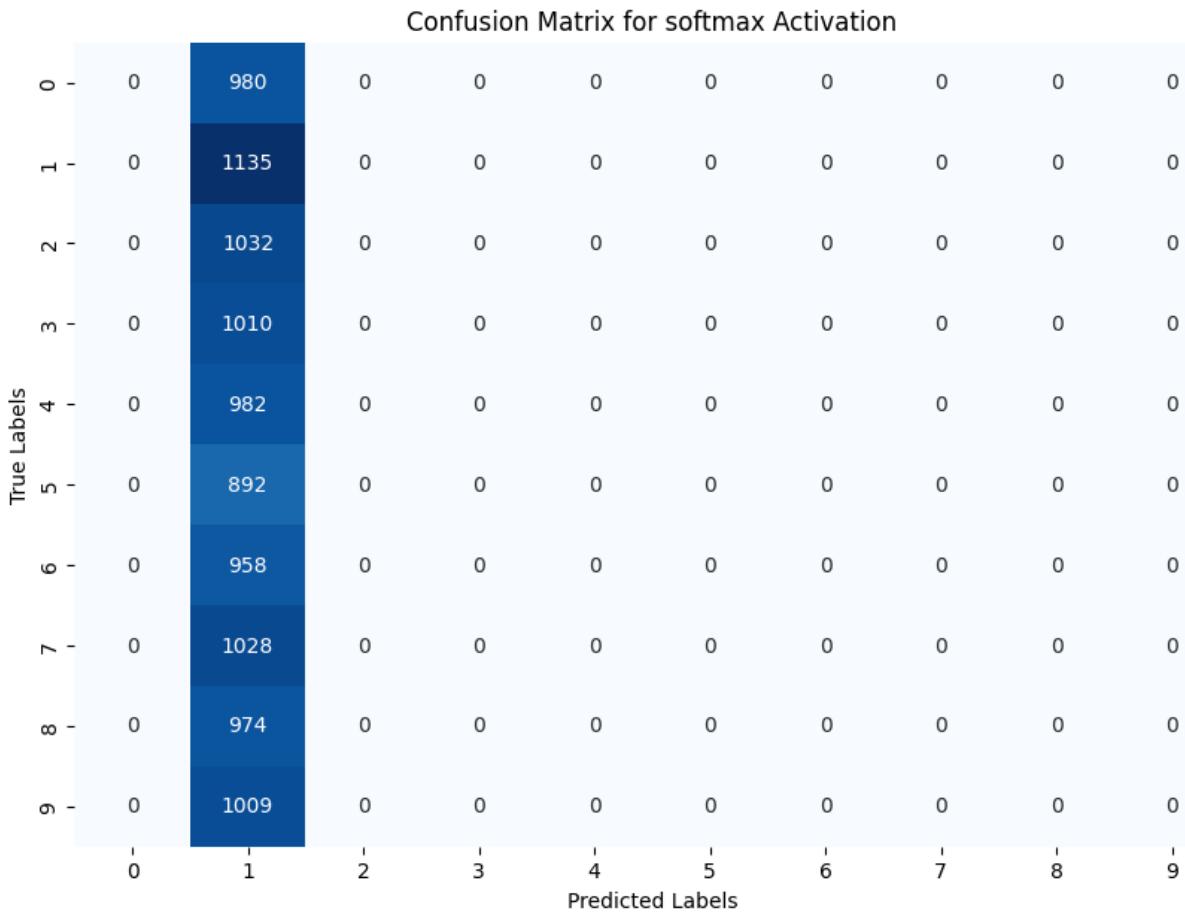
```
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 3/5
211/211 - 3s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 4/5
211/211 - 3s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 5/5
211/211 - 4s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 4s - loss: 2.3016 - accuracy: 0.1113 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 2/15
211/211 - 3s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 3/15
211/211 - 4s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 4s/epoch - 17ms/step
Epoch 4/15
211/211 - 3s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3013 -
val_accuracy: 0.1050 - 3s/epoch - 17ms/step
Epoch 5/15
211/211 - 3s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3012 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 6/15
211/211 - 3s - loss: 2.3003 - accuracy: 0.1132 - val_loss: 2.3009 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 7/15
```

```
211/211 - 3s - loss: 2.3001 - accuracy: 0.1132 - val_loss: 2.3006 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 8/15  
211/211 - 3s - loss: 2.2998 - accuracy: 0.1132 - val_loss: 2.3005 -  
val_accuracy: 0.1050 - 3s/epoch - 17ms/step  
Epoch 9/15  
211/211 - 3s - loss: 2.2995 - accuracy: 0.1132 - val_loss: 2.3002 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 10/15  
211/211 - 3s - loss: 2.2992 - accuracy: 0.1132 - val_loss: 2.2996 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 11/15  
211/211 - 3s - loss: 2.2989 - accuracy: 0.1132 - val_loss: 2.2993 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 12/15  
211/211 - 3s - loss: 2.2984 - accuracy: 0.1132 - val_loss: 2.2988 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 13/15  
211/211 - 3s - loss: 2.2979 - accuracy: 0.1132 - val_loss: 2.2983 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 14/15  
211/211 - 3s - loss: 2.2972 - accuracy: 0.1132 - val_loss: 2.2975 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 15/15  
211/211 - 3s - loss: 2.2965 - accuracy: 0.1132 - val_loss: 2.2965 -  
val_accuracy: 0.1050 - 3s/epoch - 17ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 4s - loss: 2.3016 - accuracy: 0.1115 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 2/20
211/211 - 3s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 3/20
211/211 - 3s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 3s/epoch - 17ms/step
Epoch 4/20
211/211 - 3s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 3s/epoch - 17ms/step
Epoch 5/20
211/211 - 3s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 6/20
211/211 - 3s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 3s/epoch - 16ms/step
Epoch 7/20
```

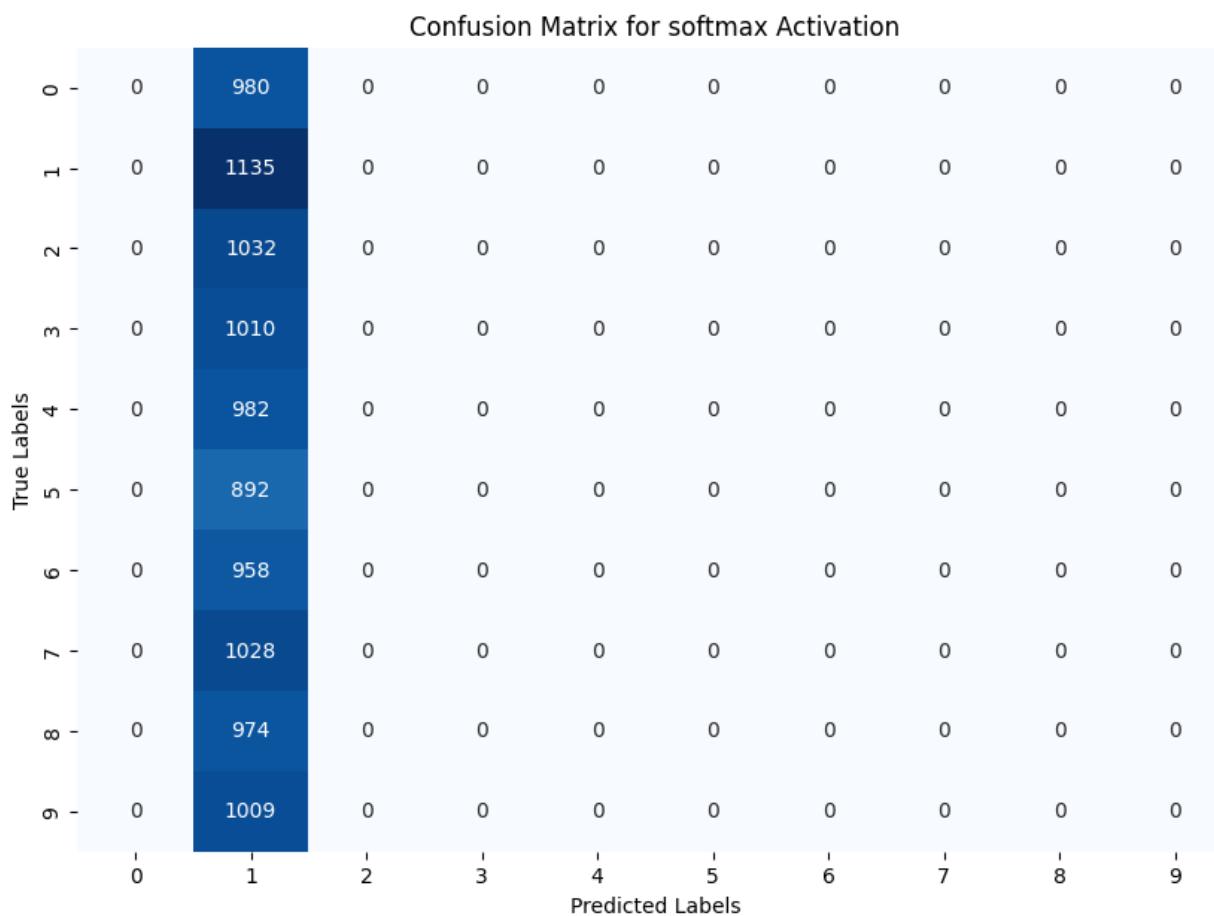
```
211/211 - 3s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3013 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 8/20  
211/211 - 3s - loss: 2.3006 - accuracy: 0.1132 - val_loss: 2.3011 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 9/20  
211/211 - 3s - loss: 2.3004 - accuracy: 0.1132 - val_loss: 2.3009 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 10/20  
211/211 - 3s - loss: 2.3004 - accuracy: 0.1132 - val_loss: 2.3011 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 11/20  
211/211 - 3s - loss: 2.3002 - accuracy: 0.1132 - val_loss: 2.3008 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 12/20  
211/211 - 3s - loss: 2.3001 - accuracy: 0.1132 - val_loss: 2.3009 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 13/20  
211/211 - 3s - loss: 2.3000 - accuracy: 0.1132 - val_loss: 2.3006 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 14/20  
211/211 - 3s - loss: 2.2998 - accuracy: 0.1132 - val_loss: 2.3006 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 15/20  
211/211 - 3s - loss: 2.2997 - accuracy: 0.1132 - val_loss: 2.3002 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 16/20  
211/211 - 3s - loss: 2.2996 - accuracy: 0.1132 - val_loss: 2.3001 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 17/20  
211/211 - 3s - loss: 2.2994 - accuracy: 0.1132 - val_loss: 2.2998 -  
val_accuracy: 0.1050 - 3s/epoch - 15ms/step  
Epoch 18/20  
211/211 - 3s - loss: 2.2991 - accuracy: 0.1132 - val_loss: 2.2997 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 19/20  
211/211 - 3s - loss: 2.2990 - accuracy: 0.1132 - val_loss: 2.2996 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
Epoch 20/20  
211/211 - 3s - loss: 2.2987 - accuracy: 0.1132 - val_loss: 2.2993 -  
val_accuracy: 0.1050 - 3s/epoch - 16ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 1 conv_layers, 1 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

844/844 - 7s - loss: 2.3015 - accuracy: 0.1116 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step

Epoch 2/5

844/844 - 6s - loss: 2.3013 - accuracy: 0.1130 - val_loss: 2.3020 -

```
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 3/5
844/844 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 4/5
844/844 - 6s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 5/5
844/844 - 6s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0	980	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0

```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 7s - loss: 2.3015 - accuracy: 0.1127 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 2/15
844/844 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 3/15
844/844 - 6s - loss: 2.3012 - accuracy: 0.1133 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 4/15
844/844 - 6s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 5/15
844/844 - 6s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 6/15
844/844 - 6s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3007 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 7/15
```

```
844/844 - 6s - loss: 2.3003 - accuracy: 0.1132 - val_loss: 2.3009 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 8/15  
844/844 - 7s - loss: 2.2998 - accuracy: 0.1132 - val_loss: 2.3006 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 9/15  
844/844 - 6s - loss: 2.2991 - accuracy: 0.1149 - val_loss: 2.3001 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 10/15  
844/844 - 7s - loss: 2.2984 - accuracy: 0.1132 - val_loss: 2.2989 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 11/15  
844/844 - 6s - loss: 2.2972 - accuracy: 0.1143 - val_loss: 2.2970 -  
val_accuracy: 0.1973 - 6s/epoch - 8ms/step  
Epoch 12/15  
844/844 - 6s - loss: 2.2952 - accuracy: 0.1266 - val_loss: 2.2934 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 13/15  
844/844 - 6s - loss: 2.1982 - accuracy: 0.1991 - val_loss: 2.0560 -  
val_accuracy: 0.2160 - 6s/epoch - 8ms/step  
Epoch 14/15  
844/844 - 6s - loss: 1.9746 - accuracy: 0.2307 - val_loss: 1.9062 -  
val_accuracy: 0.2267 - 6s/epoch - 8ms/step  
Epoch 15/15  
844/844 - 6s - loss: 1.8661 - accuracy: 0.2629 - val_loss: 1.7679 -  
val_accuracy: 0.3265 - 6s/epoch - 8ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 917    1   43   10    2    1    1    0    5    0]  
[  3 1089    5   17   20    0    0    0    1    0]  
[ 753   30  135   66   25    0    1    0   22    0]  
[ 27   55   35  869    3    3    0    0   18    0]  
[ 67   535   98   79  173    0    7    1   22    0]  
[ 72   85  110   521   62    5    2    1   34    0]  
[ 892    2   43    3    7    0    9    0    2    0]  
[  3  940   13   43   14    0    0    9    6    0]  
[ 116   43  190   461   39    4    1    1  119    0]  
[ 20  738   36  133   71    0    1    5    5    0]]  
Precision: 0.3443  
Recall: 0.3325  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for softmax Activation										
	0	1	2	3	4	5	6	7	8	9
True Labels	917	1	43	10	2	1	1	0	5	0
0	917	1	43	10	2	1	1	0	5	0
1	3	1089	5	17	20	0	0	0	1	0
2	753	30	135	66	25	0	1	0	22	0
3	27	55	35	869	3	3	0	0	18	0
4	67	535	98	79	173	0	7	1	22	0
5	72	85	110	521	62	5	2	1	34	0
6	892	2	43	3	7	0	9	0	2	0
7	3	940	13	43	14	0	0	9	6	0
8	116	43	190	461	39	4	1	1	119	0
9	20	738	36	133	71	0	1	5	5	0
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 7s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 8ms/step
Epoch 2/20
844/844 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3028 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 3/20
844/844 - 6s - loss: 2.3014 - accuracy: 0.1129 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 4/20
844/844 - 6s - loss: 2.3014 - accuracy: 0.1128 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 5/20
844/844 - 6s - loss: 2.3012 - accuracy: 0.1128 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 6/20
844/844 - 6s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 8ms/step
Epoch 7/20
```

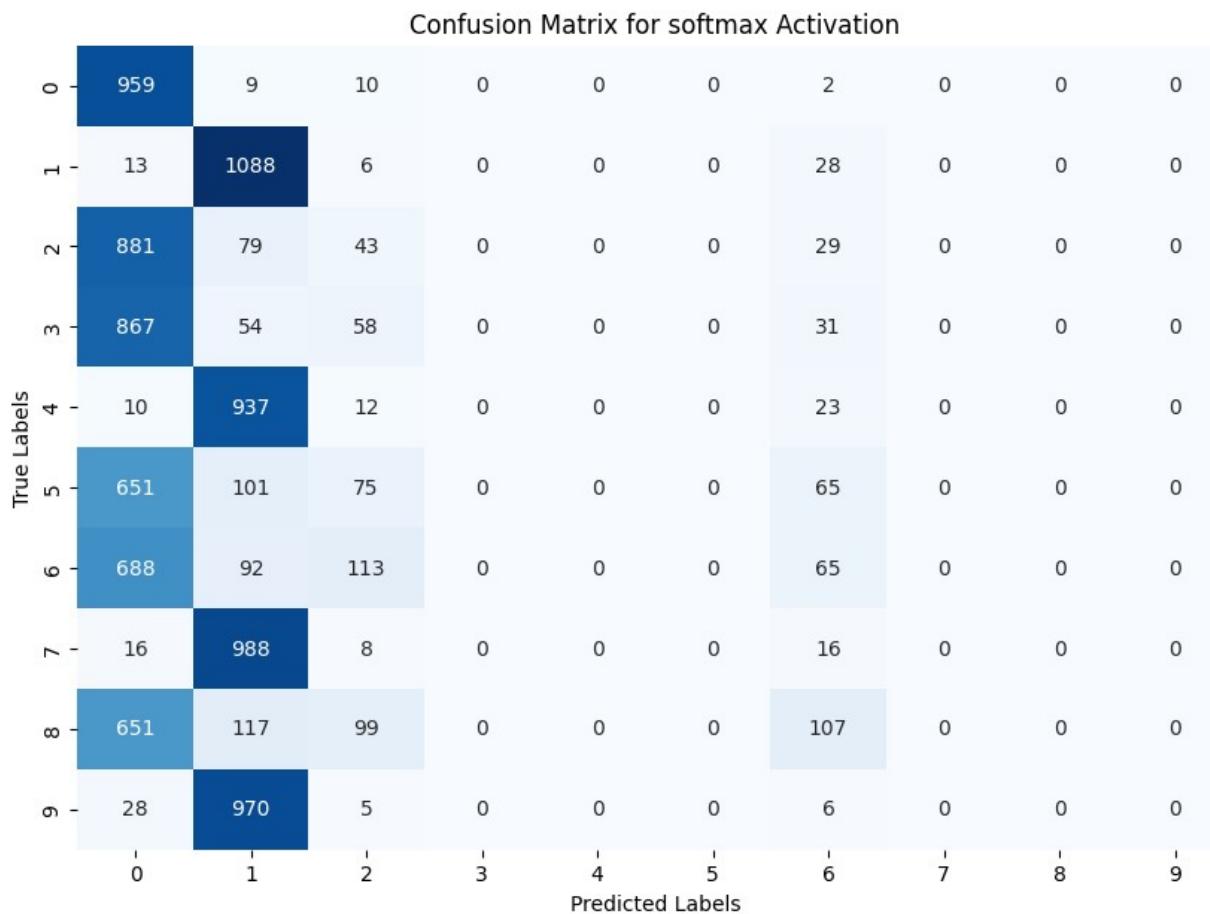
```
844/844 - 6s - loss: 2.3010 - accuracy: 0.1133 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 8/20  
844/844 - 6s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 9/20  
844/844 - 6s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 10/20  
844/844 - 6s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 11/20  
844/844 - 6s - loss: 2.3006 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 12/20  
844/844 - 6s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3010 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 13/20  
844/844 - 6s - loss: 2.3003 - accuracy: 0.1132 - val_loss: 2.3008 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 14/20  
844/844 - 6s - loss: 2.2999 - accuracy: 0.1145 - val_loss: 2.3007 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 15/20  
844/844 - 6s - loss: 2.2996 - accuracy: 0.1132 - val_loss: 2.2998 -  
val_accuracy: 0.1050 - 6s/epoch - 7ms/step  
Epoch 16/20  
844/844 - 6s - loss: 2.2989 - accuracy: 0.1132 - val_loss: 2.2988 -  
val_accuracy: 0.1050 - 6s/epoch - 7ms/step  
Epoch 17/20  
844/844 - 6s - loss: 2.2980 - accuracy: 0.1132 - val_loss: 2.2982 -  
val_accuracy: 0.1050 - 6s/epoch - 8ms/step  
Epoch 18/20  
844/844 - 7s - loss: 2.2967 - accuracy: 0.1132 - val_loss: 2.2961 -  
val_accuracy: 0.1050 - 7s/epoch - 8ms/step  
Epoch 19/20  
844/844 - 6s - loss: 2.2431 - accuracy: 0.1669 - val_loss: 2.0813 -  
val_accuracy: 0.2057 - 6s/epoch - 8ms/step  
Epoch 20/20  
844/844 - 7s - loss: 1.9587 - accuracy: 0.2275 - val_loss: 1.8556 -  
val_accuracy: 0.2112 - 7s/epoch - 8ms/step  
313/313 [=====] - 2s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 959   9  10   0   0   0   2   0   0   0 ]  
 [  13 1088   6   0   0   0  28   0   0   0 ]  
 [ 881   79  43   0   0   0  29   0   0   0 ]  
 [ 867   54  58   0   0   0  31   0   0   0 ]  
 [  10  937  12   0   0   0  23   0   0   0 ]
```

```
[ 651 101 75 0 0 0 65 0 0 0]
[ 688 92 113 0 0 0 65 0 0 0]
[ 16 988 8 0 0 0 16 0 0 0]
[ 651 117 99 0 0 0 107 0 0 0]
[ 28 970 5 0 0 0 6 0 0 0]]
```

Precision: 0.0747

Recall: 0.2155

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 1 conv_layers, 1 dense layers,
128 batch size, 5 epochs..

Epoch 1/5

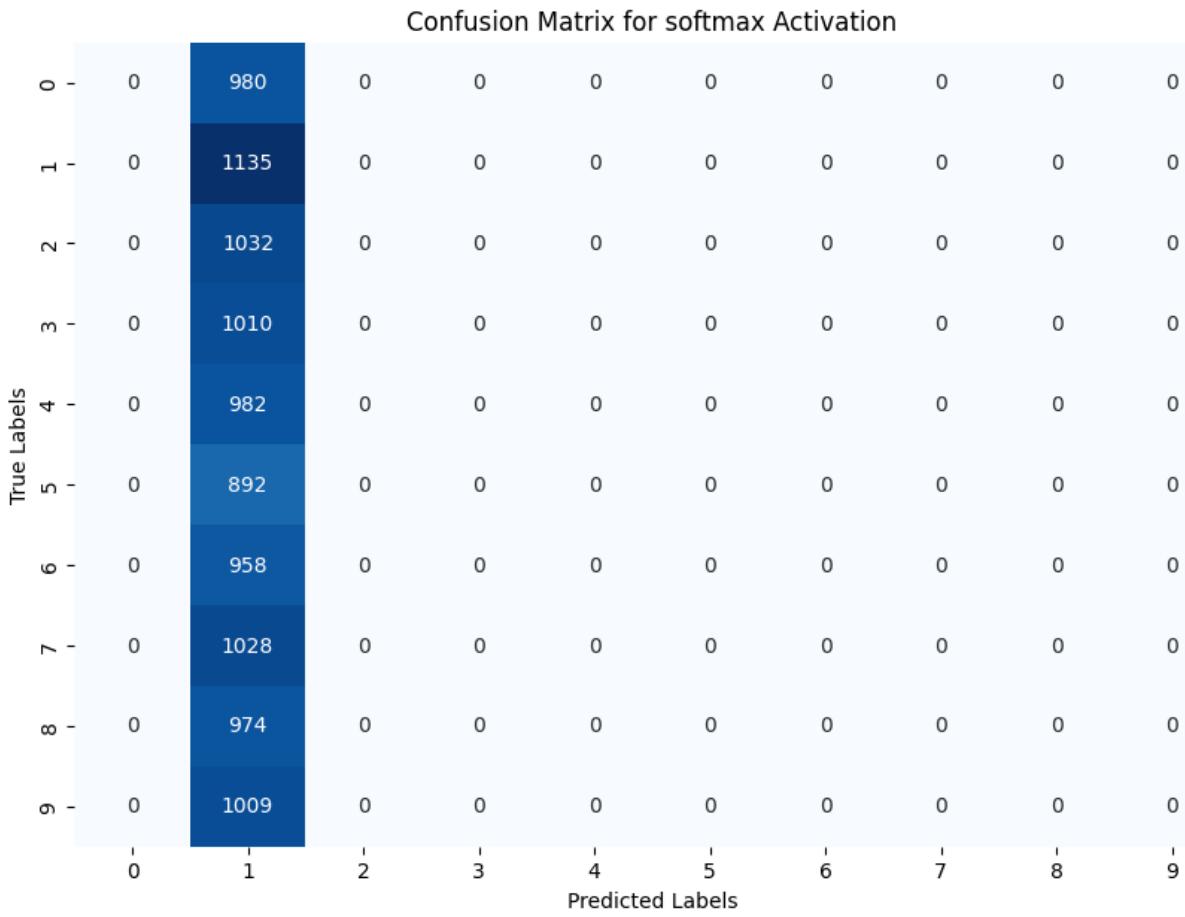
422/422 - 5s - loss: 2.3015 - accuracy: 0.1127 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step

Epoch 2/5

422/422 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -

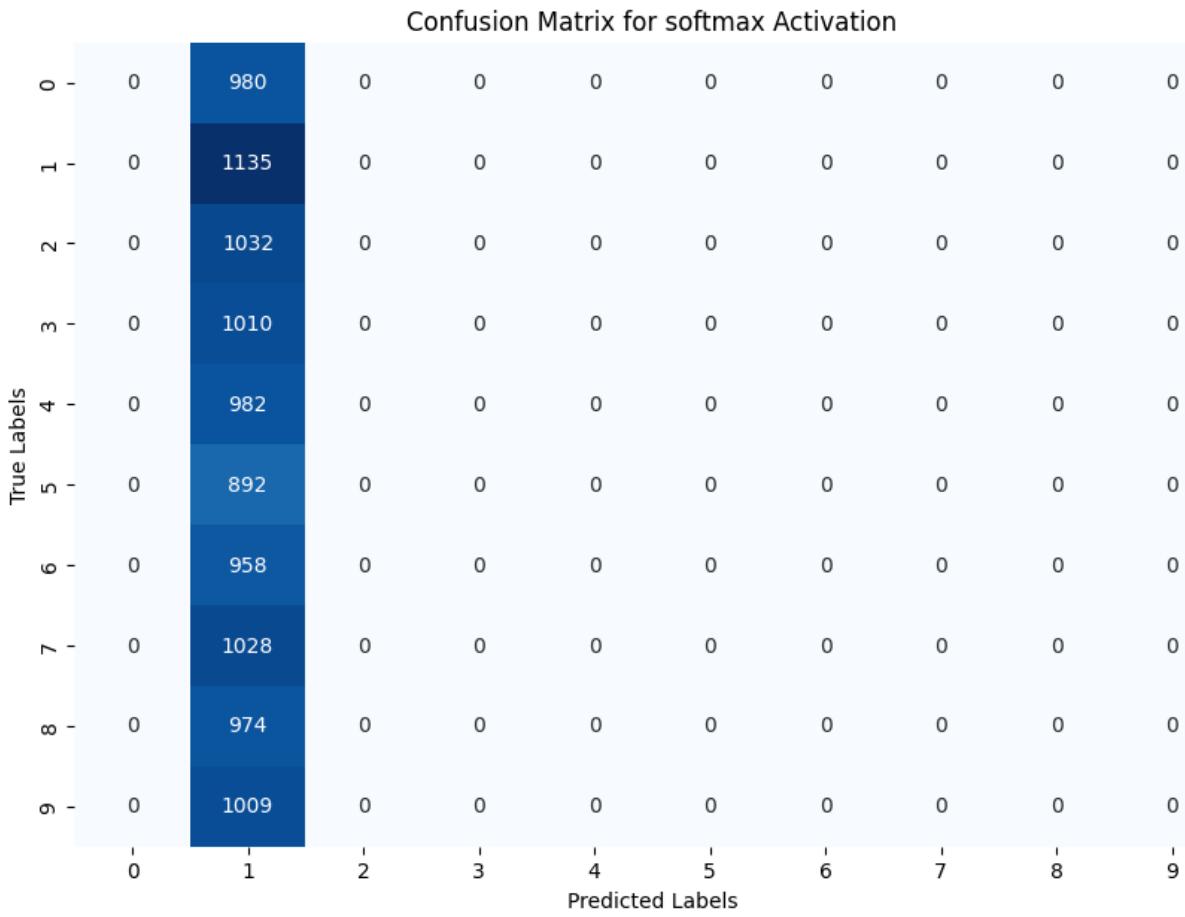
```
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 3/5
422/422 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 4/5
422/422 - 5s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 5/5
422/422 - 5s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 5s - loss: 2.3015 - accuracy: 0.1124 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 2/15
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 3/15
422/422 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 4/15
422/422 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 11ms/step
Epoch 5/15
422/422 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 6/15
422/422 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 4s/epoch - 10ms/step
Epoch 7/15
```

```
422/422 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 4s/epoch - 11ms/step  
Epoch 8/15  
422/422 - 5s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 9/15  
422/422 - 5s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 10/15  
422/422 - 5s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3014 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 11/15  
422/422 - 5s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 12/15  
422/422 - 5s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3011 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 13/15  
422/422 - 5s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3011 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 14/15  
422/422 - 5s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3013 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 15/15  
422/422 - 5s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3012 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 5s - loss: 2.3014 - accuracy: 0.1130 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 12ms/step
Epoch 2/20
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 3/20
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 4/20
422/422 - 5s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 5/20
422/422 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 6/20
422/422 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 11ms/step
Epoch 7/20
```

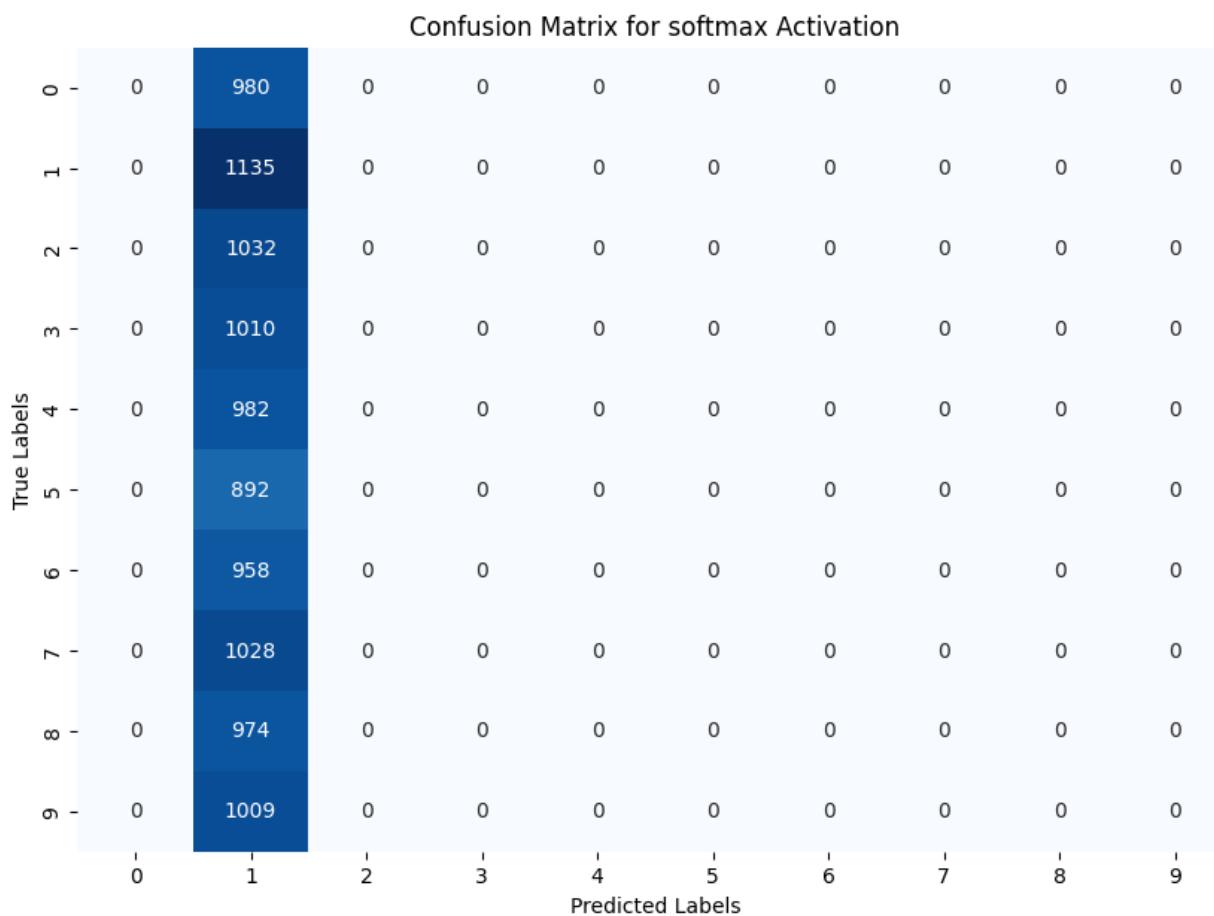
```
422/422 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 4s/epoch - 11ms/step  
Epoch 8/20  
422/422 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 4s/epoch - 11ms/step  
Epoch 9/20  
422/422 - 5s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 10/20  
422/422 - 5s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 11/20  
422/422 - 5s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 12/20  
422/422 - 5s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 13/20  
422/422 - 5s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3014 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 14/20  
422/422 - 5s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3014 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 15/20  
422/422 - 5s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3012 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 16/20  
422/422 - 5s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3014 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 17/20  
422/422 - 5s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3010 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 18/20  
422/422 - 5s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3013 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 19/20  
422/422 - 5s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
Epoch 20/20  
422/422 - 5s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3012 -  
val_accuracy: 0.1050 - 5s/epoch - 11ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

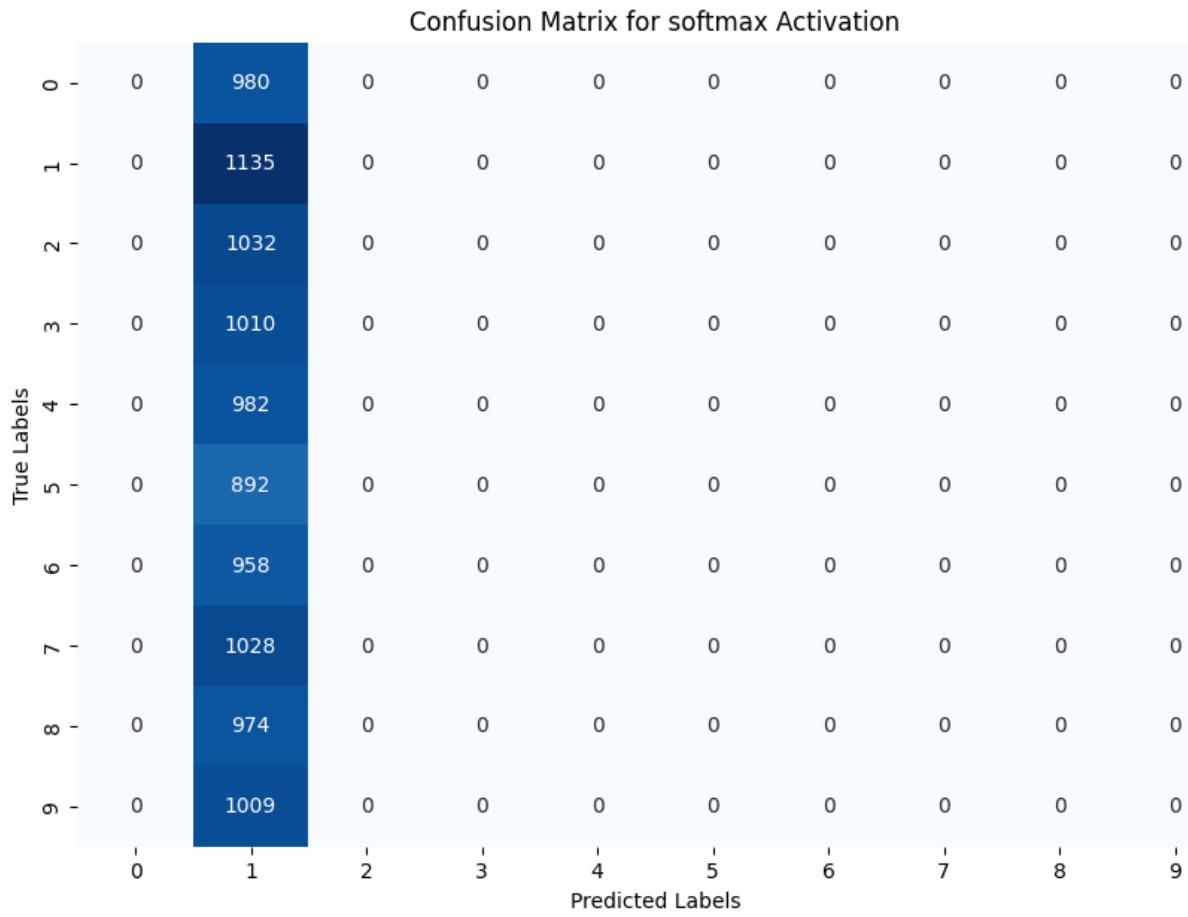
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 5s - loss: 2.3015 - accuracy: 0.1122 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 21ms/step
Epoch 2/5
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -
```

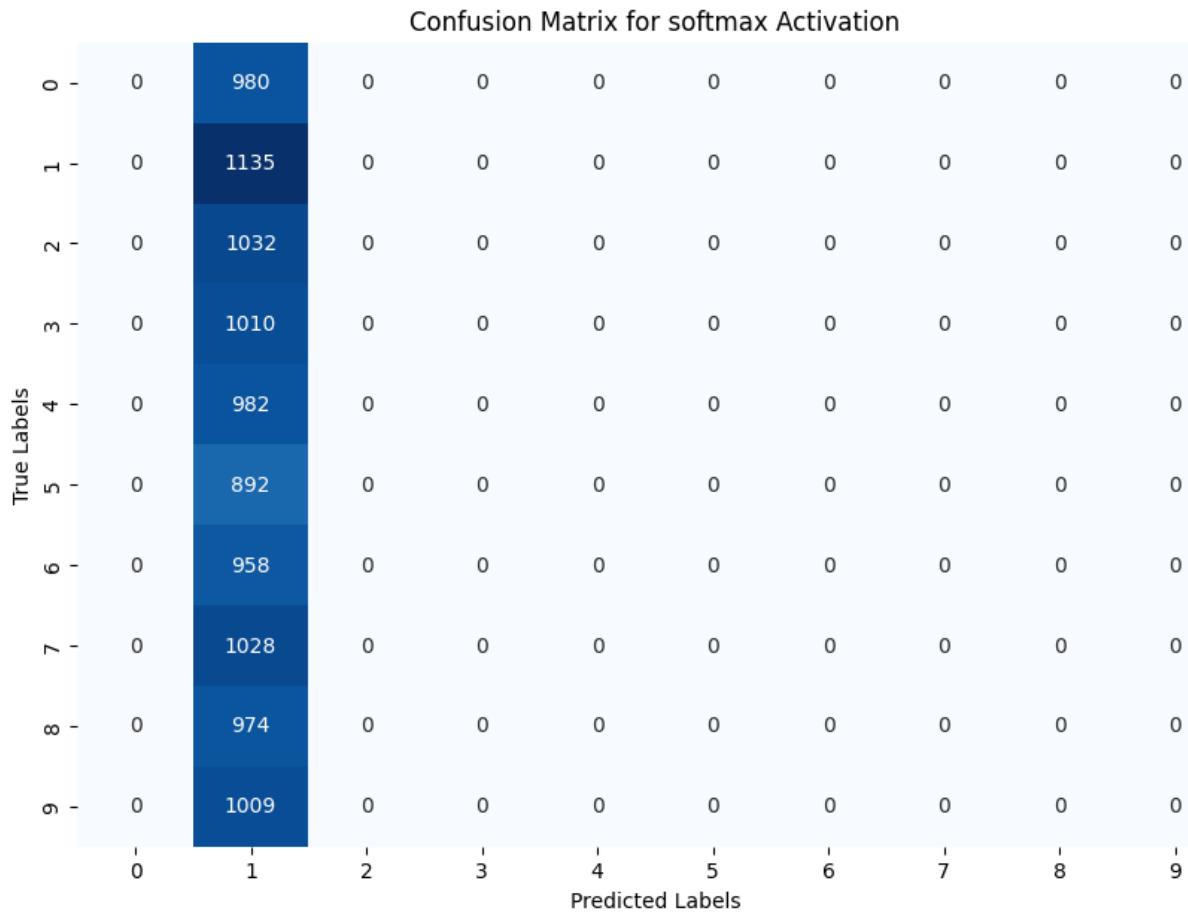
```
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 3/5
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 4/5
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 5/5
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 4s - loss: 2.3016 - accuracy: 0.1119 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 2/15
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 3/15
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 4/15
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 5/15
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 6/15
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 4s/epoch - 17ms/step
Epoch 7/15
```

```
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 4s/epoch - 17ms/step  
Epoch 8/15  
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 9/15  
211/211 - 4s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 10/15  
211/211 - 4s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3014 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 11/15  
211/211 - 4s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 12/15  
211/211 - 4s - loss: 2.3007 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 13/15  
211/211 - 4s - loss: 2.3006 - accuracy: 0.1132 - val_loss: 2.3013 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 14/15  
211/211 - 4s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3012 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 15/15  
211/211 - 4s - loss: 2.3005 - accuracy: 0.1132 - val_loss: 2.3011 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 1 conv_layers, 1 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 4s - loss: 2.3016 - accuracy: 0.1119 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 2/20
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 3/20
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 4/20
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 5/20
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 4s/epoch - 18ms/step
Epoch 6/20
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 19ms/step
Epoch 7/20
```

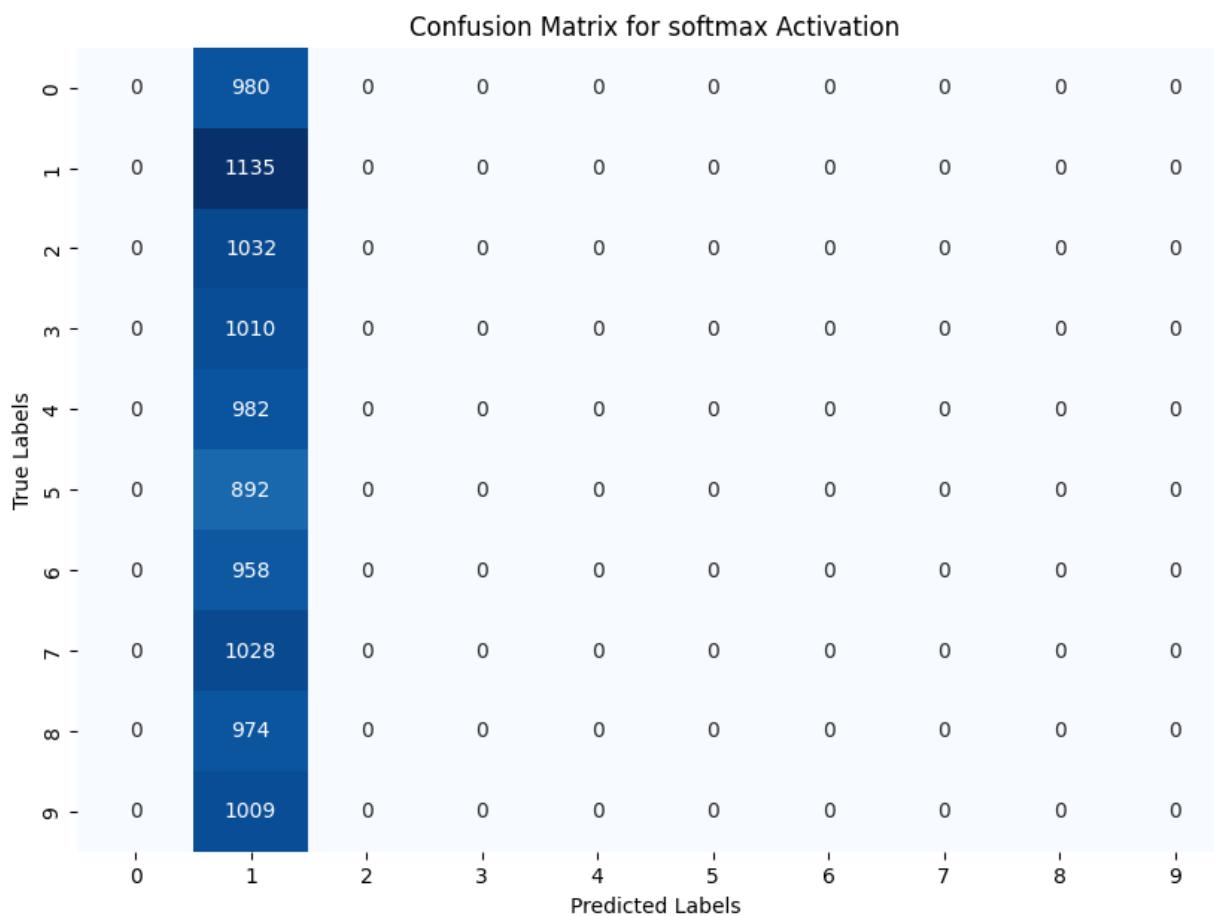
```
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 8/20  
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 9/20  
211/211 - 4s - loss: 2.3011 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 10/20  
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 11/20  
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 12/20  
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 13/20  
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 14/20  
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 15/20  
211/211 - 4s - loss: 2.3010 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 16/20  
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 17/20  
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 4s/epoch - 19ms/step  
Epoch 18/20  
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 19/20  
211/211 - 4s - loss: 2.3009 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
Epoch 20/20  
211/211 - 4s - loss: 2.3008 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 4s/epoch - 18ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

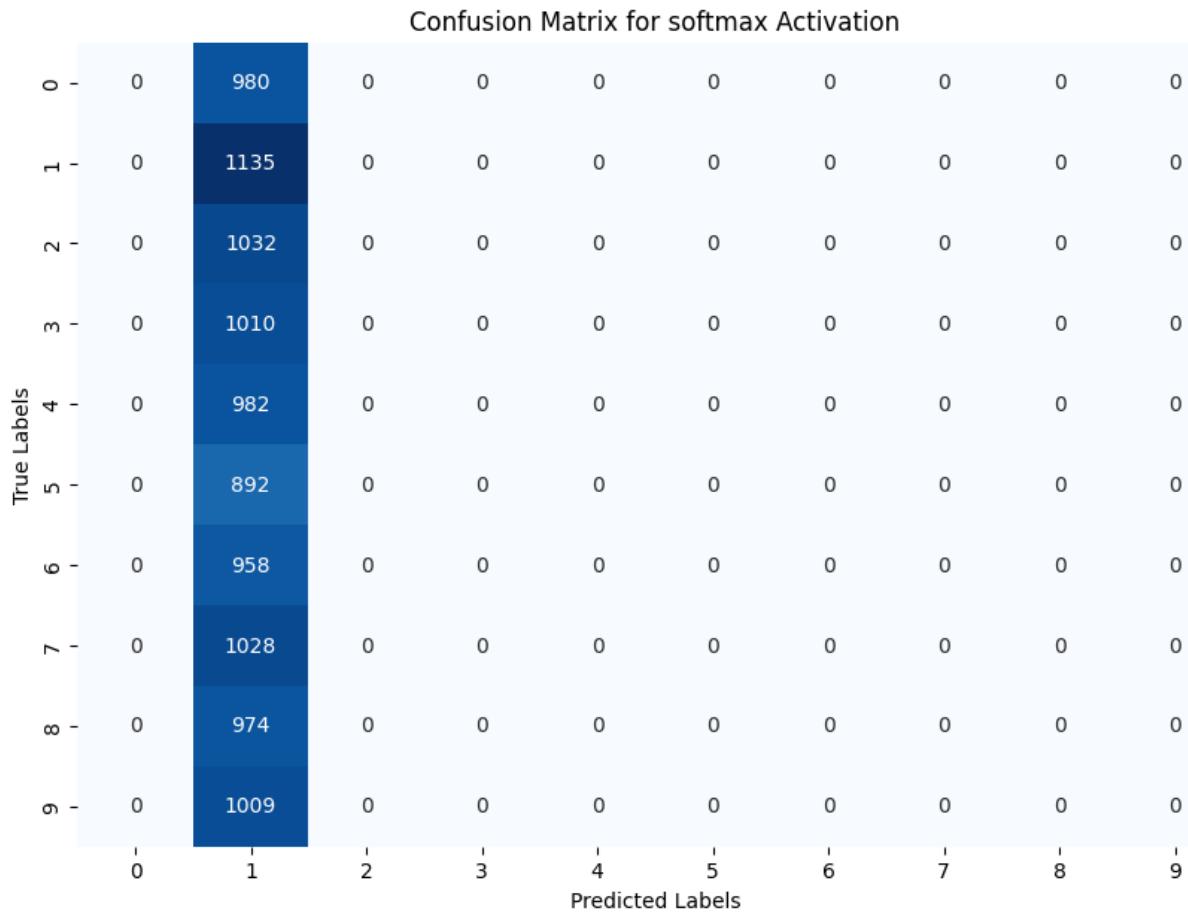
844/844 - 9s - loss: 2.3015 - accuracy: 0.1123 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step

Epoch 2/5

844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3016 -

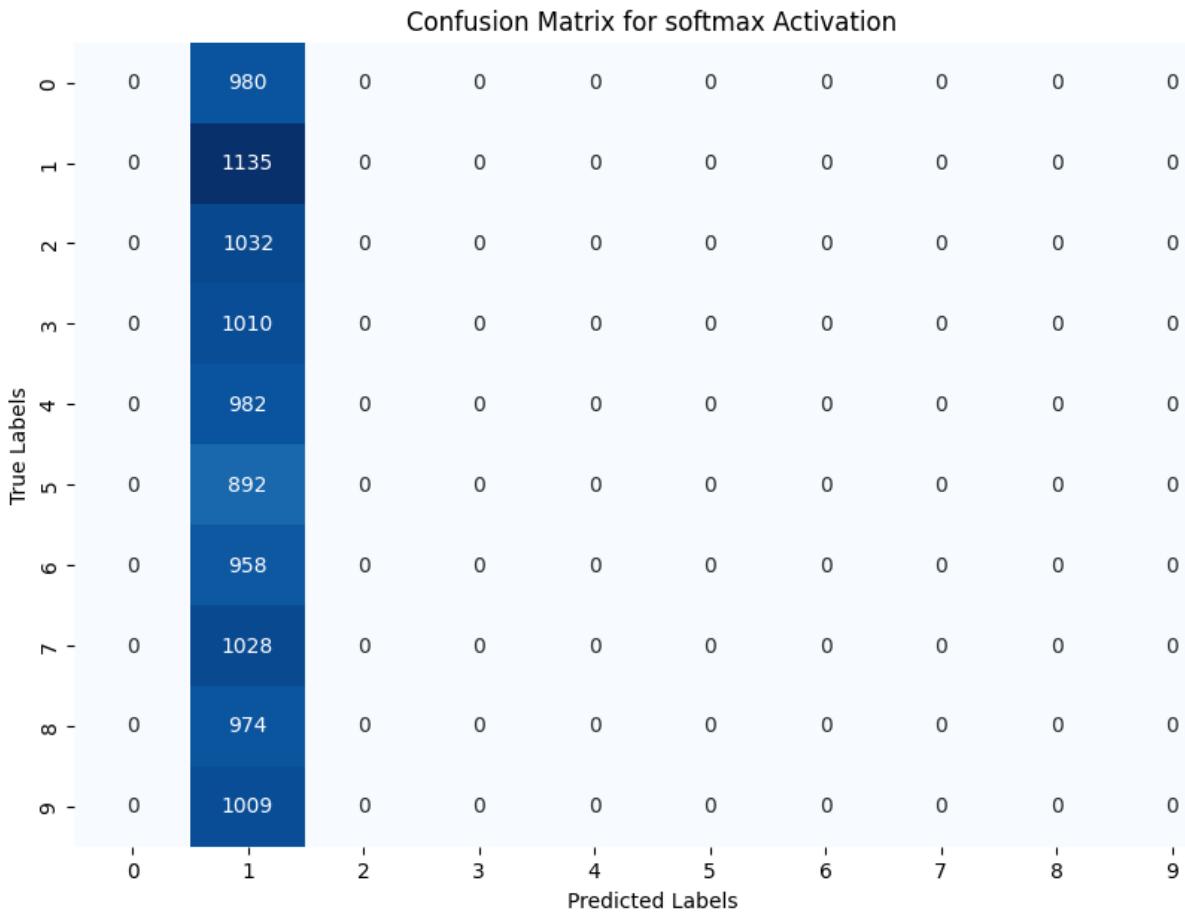
```
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 3/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 4/5
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3027 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 9s - loss: 2.3015 - accuracy: 0.1117 - val_loss: 2.3026 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 2/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/15
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 4/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 6/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 7/15
```

```
844/844 - 8s - loss: 2.3015 - accuracy: 0.1130 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 8/15  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 9/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 10/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 11/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1126 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 12/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 13/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1129 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 14/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 15/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1121 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 2/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 3/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1126 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 4/20
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 5/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 6/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 7/20
```

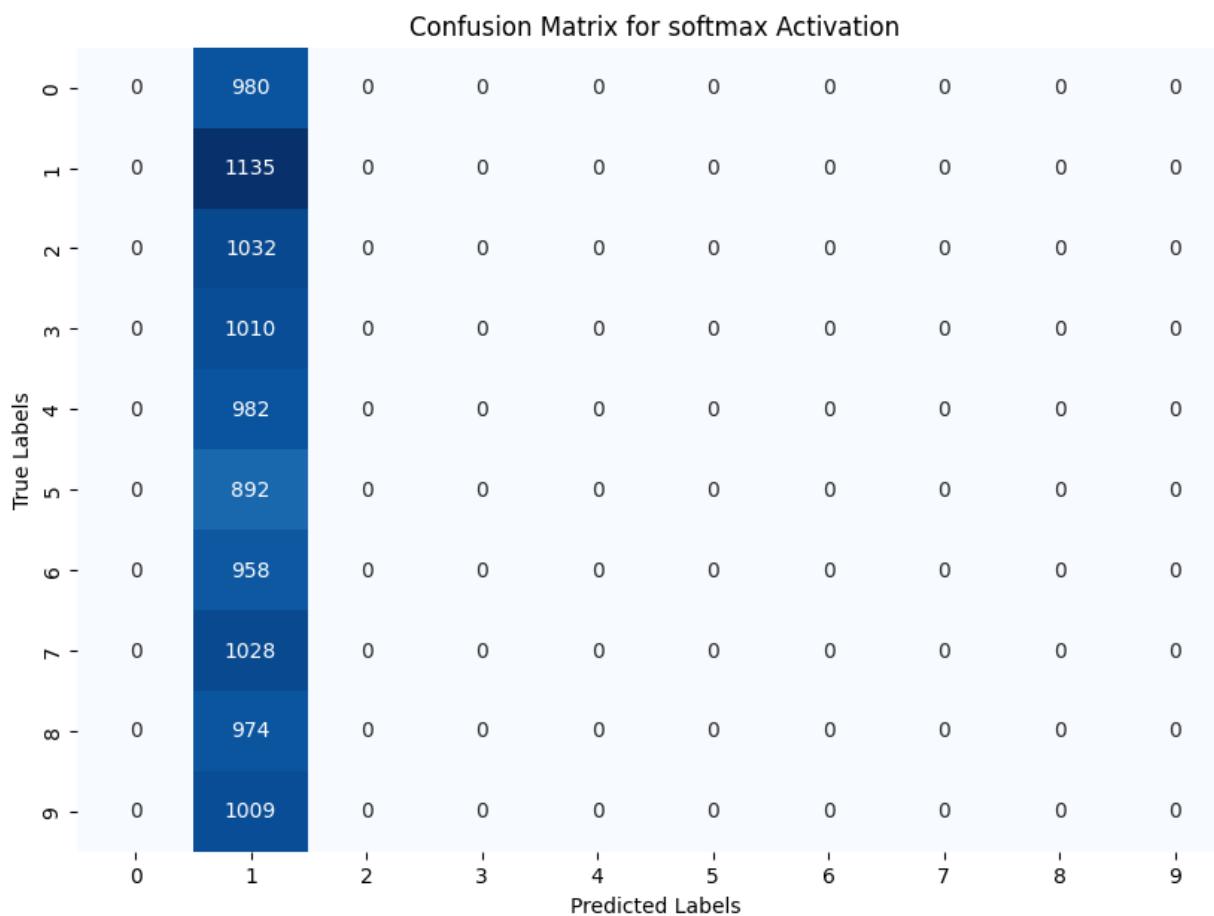
```
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 8/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 9/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 10/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 11/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 12/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1126 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 13/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 14/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 15/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 16/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 17/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 18/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 19/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 20/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

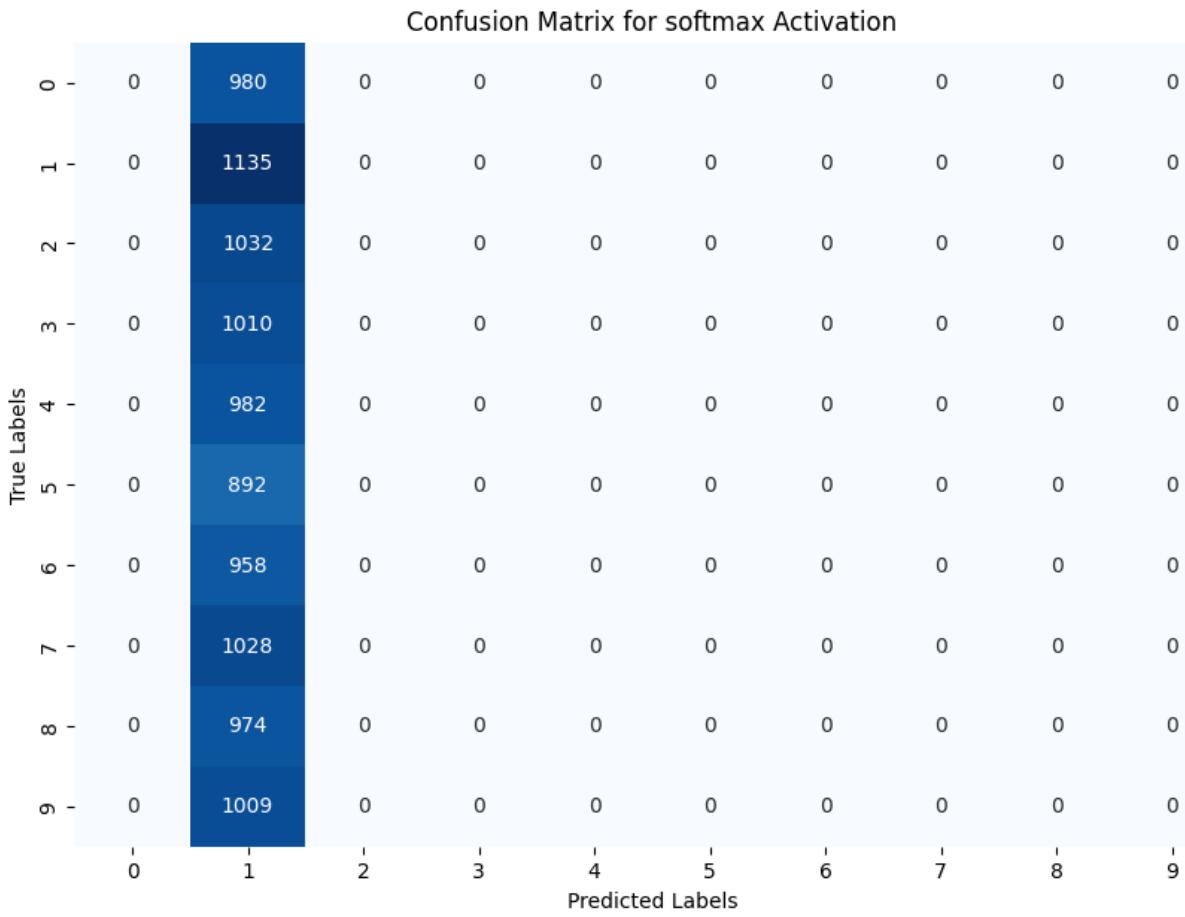
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 6s - loss: 2.3015 - accuracy: 0.1126 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 2/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
```

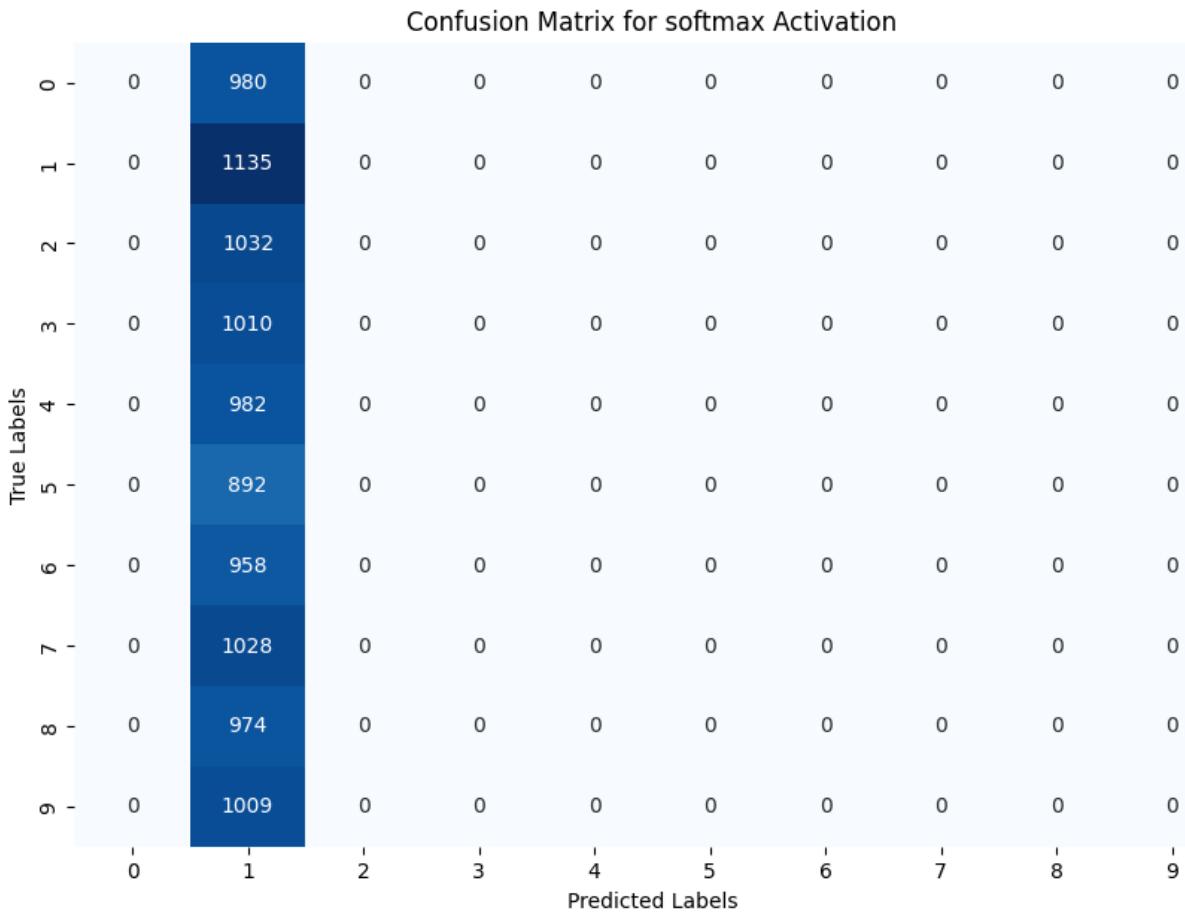
```
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 3/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 4/5
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 5/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 6s - loss: 2.3014 - accuracy: 0.1118 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 2/15
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 3/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 4/15
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 5/15
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 6/15
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 7/15
```

```
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 8/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 9/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 10/15  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 11/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 12/15  
422/422 - 5s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 13/15  
422/422 - 5s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 14/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 15/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```

Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 6s - loss: 2.3015 - accuracy: 0.1118 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 2/20
422/422 - 5s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 3/20
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 4/20
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 5/20
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 6/20
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 13ms/step
Epoch 7/20

```

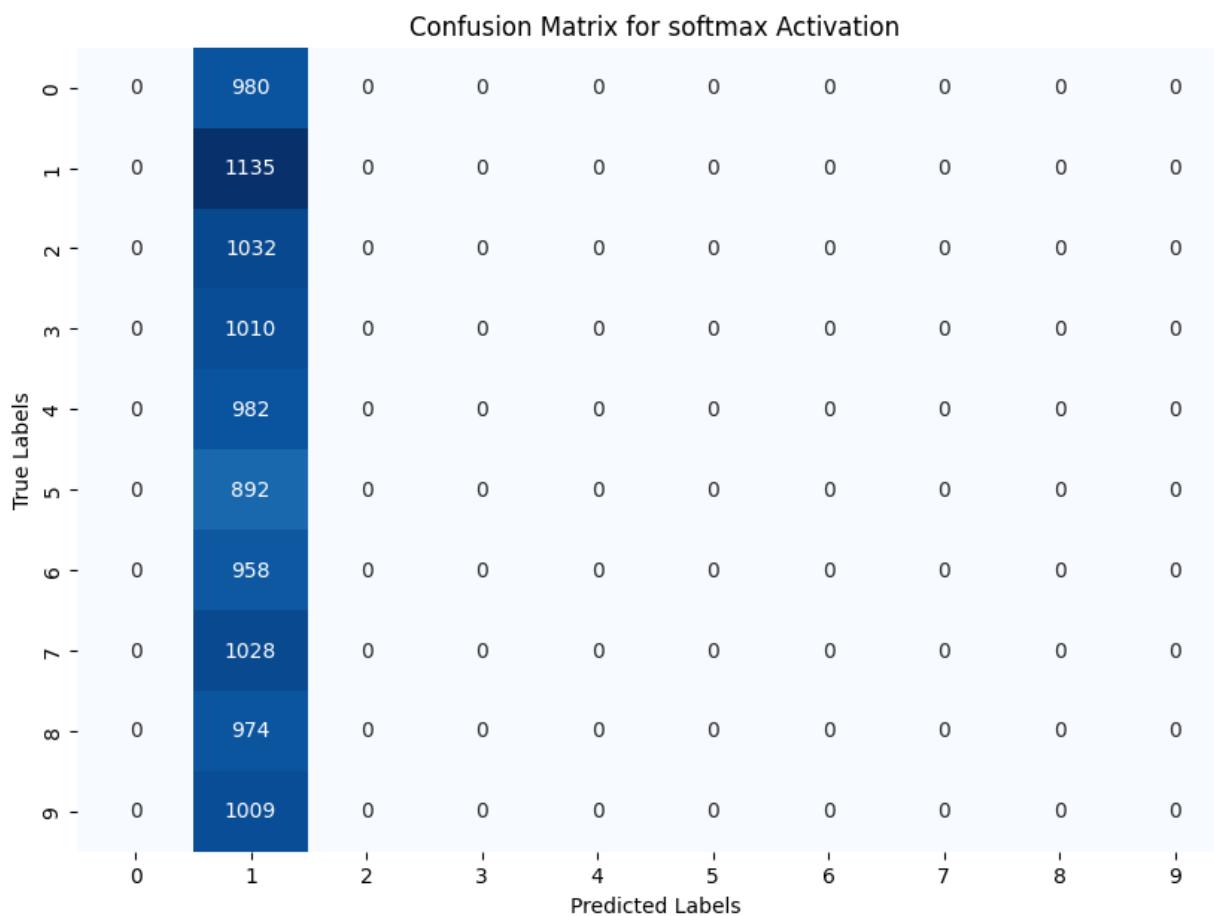
```
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 8/20  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 9/20  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 10/20  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 11/20  
422/422 - 5s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 12/20  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 12ms/step  
Epoch 13/20  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 14/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 15/20  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 16/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 17/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 18/20  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 19/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 20/20  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

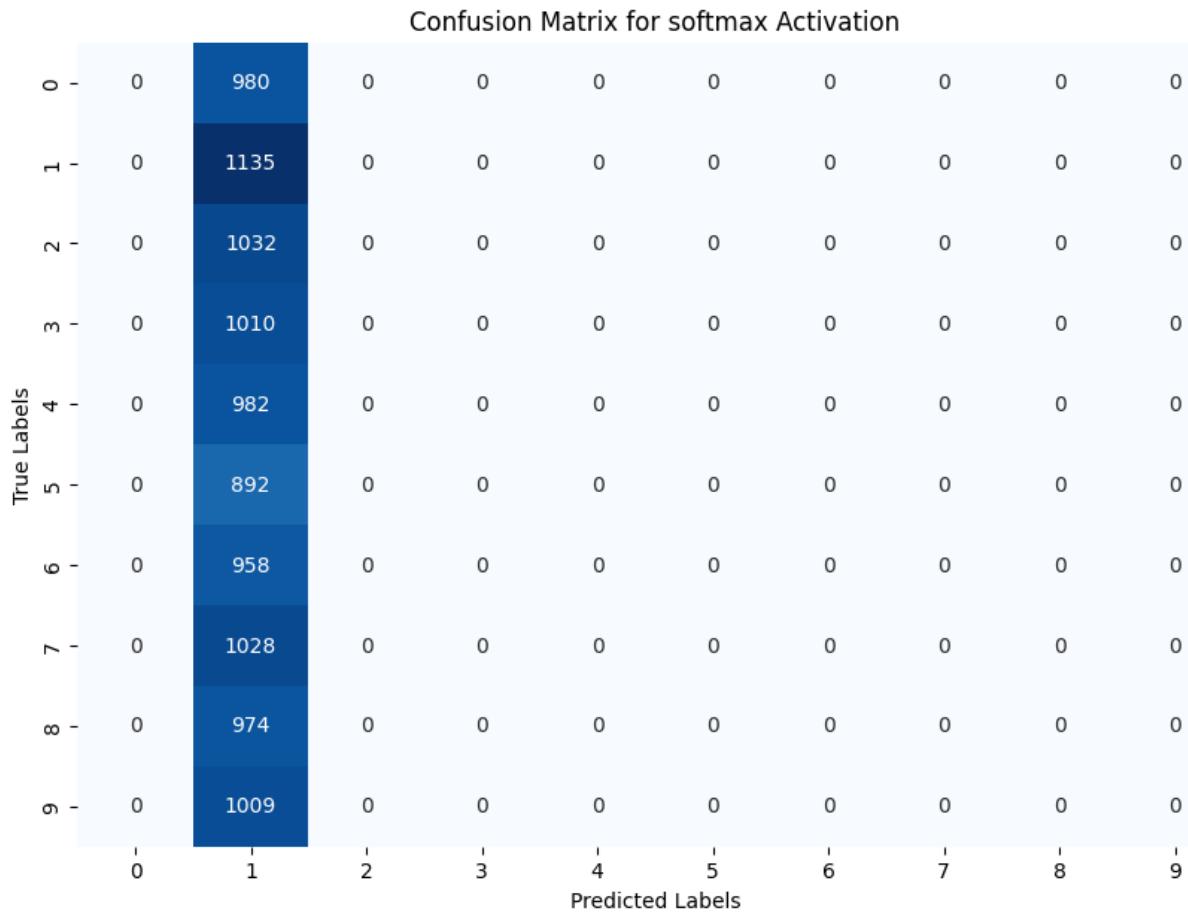
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 5s - loss: 2.3017 - accuracy: 0.1108 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 2/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
```

```
val_accuracy: 0.1050 - 5s/epoch - 21ms/step
Epoch 3/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 4/5
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 5/5
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 5s - loss: 2.3016 - accuracy: 0.1109 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 2/15
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 4s/epoch - 20ms/step
Epoch 3/15
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 4/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 21ms/step
Epoch 5/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 6/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 21ms/step
Epoch 7/15
```

```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 8/15  
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 4s/epoch - 21ms/step  
Epoch 9/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 21ms/step  
Epoch 10/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 11/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 12/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 13/15  
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 4s/epoch - 21ms/step  
Epoch 14/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 15/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 21ms/step  
313/313 [=====] - 2s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0 - 0	980	0	0	0	0	0	0	0	0
	1 - 0	1135	0	0	0	0	0	0	0	0
	2 - 0	1032	0	0	0	0	0	0	0	0
	3 - 0	1010	0	0	0	0	0	0	0	0
	4 - 0	982	0	0	0	0	0	0	0	0
	5 - 0	892	0	0	0	0	0	0	0	0
	6 - 0	958	0	0	0	0	0	0	0	0
	7 - 0	1028	0	0	0	0	0	0	0	0
	8 - 0	974	0	0	0	0	0	0	0	0
	9 - 0	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 5s - loss: 2.3016 - accuracy: 0.1103 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 26ms/step
Epoch 2/20
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 3/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 4/20
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 5/20
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 6/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 21ms/step
Epoch 7/20

```

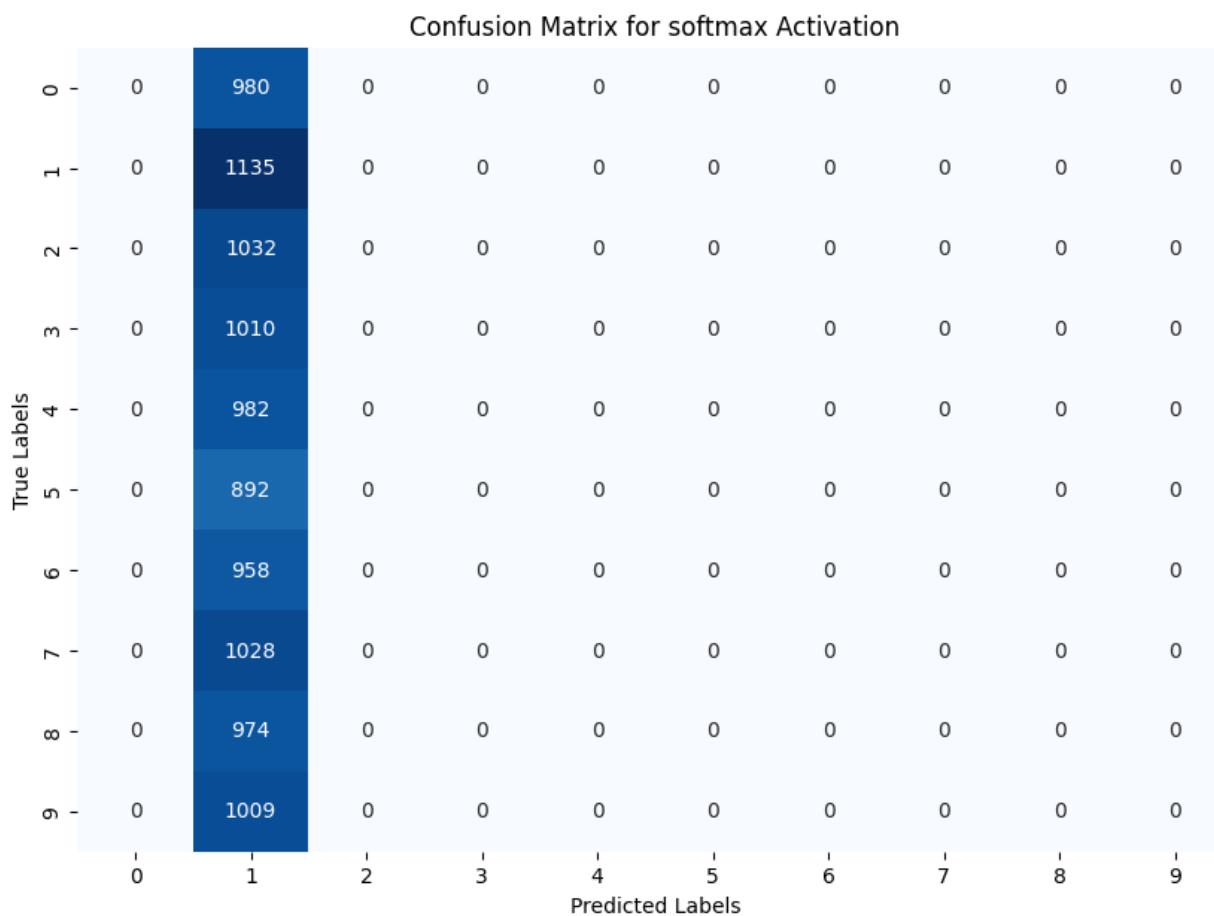
```
211/211 - 4s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 4s/epoch - 21ms/step  
Epoch 8/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 21ms/step  
Epoch 9/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 21ms/step  
Epoch 10/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 11/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 12/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 13/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 14/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 21ms/step  
Epoch 15/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 16/20  
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 4s/epoch - 21ms/step  
Epoch 17/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 18/20  
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 4s/epoch - 21ms/step  
Epoch 19/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 20/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

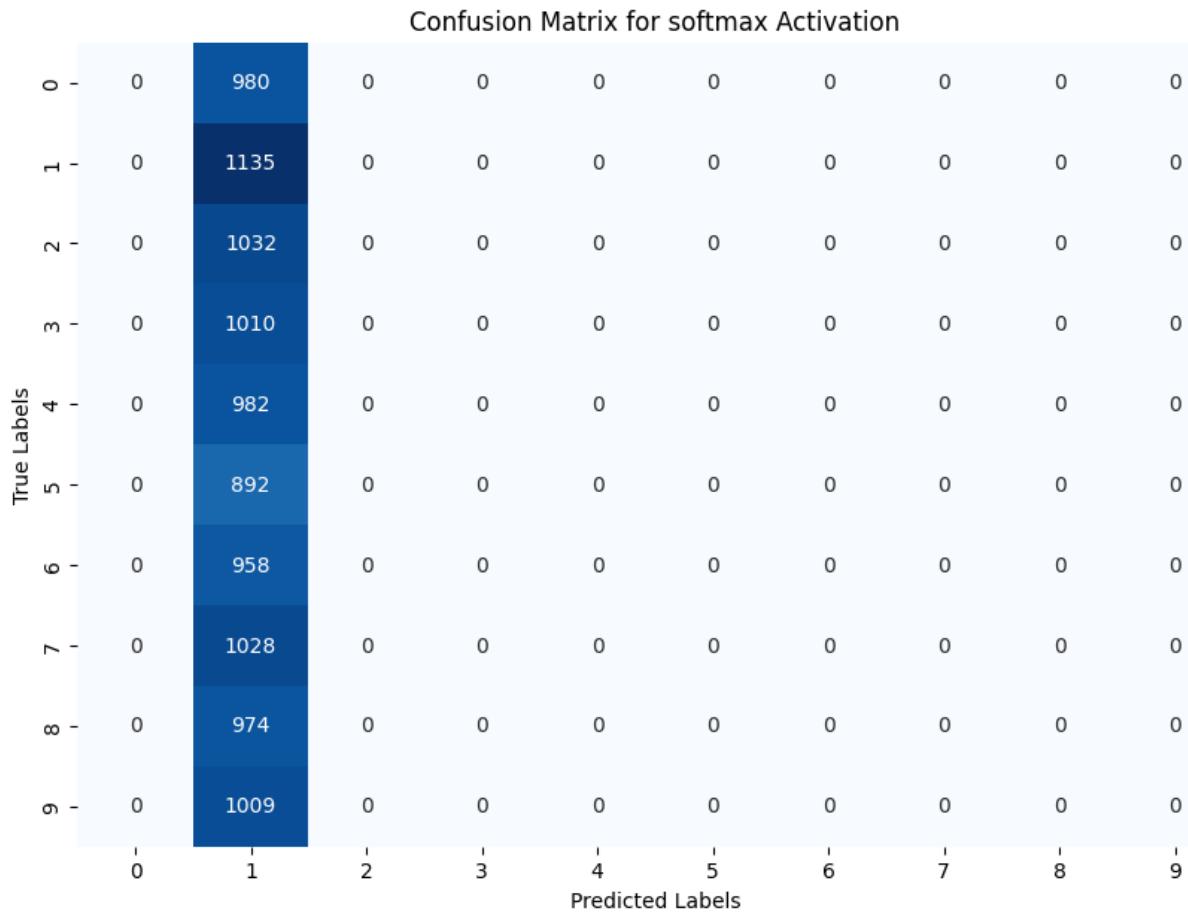
844/844 - 8s - loss: 2.3015 - accuracy: 0.1133 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step

Epoch 2/5

844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -

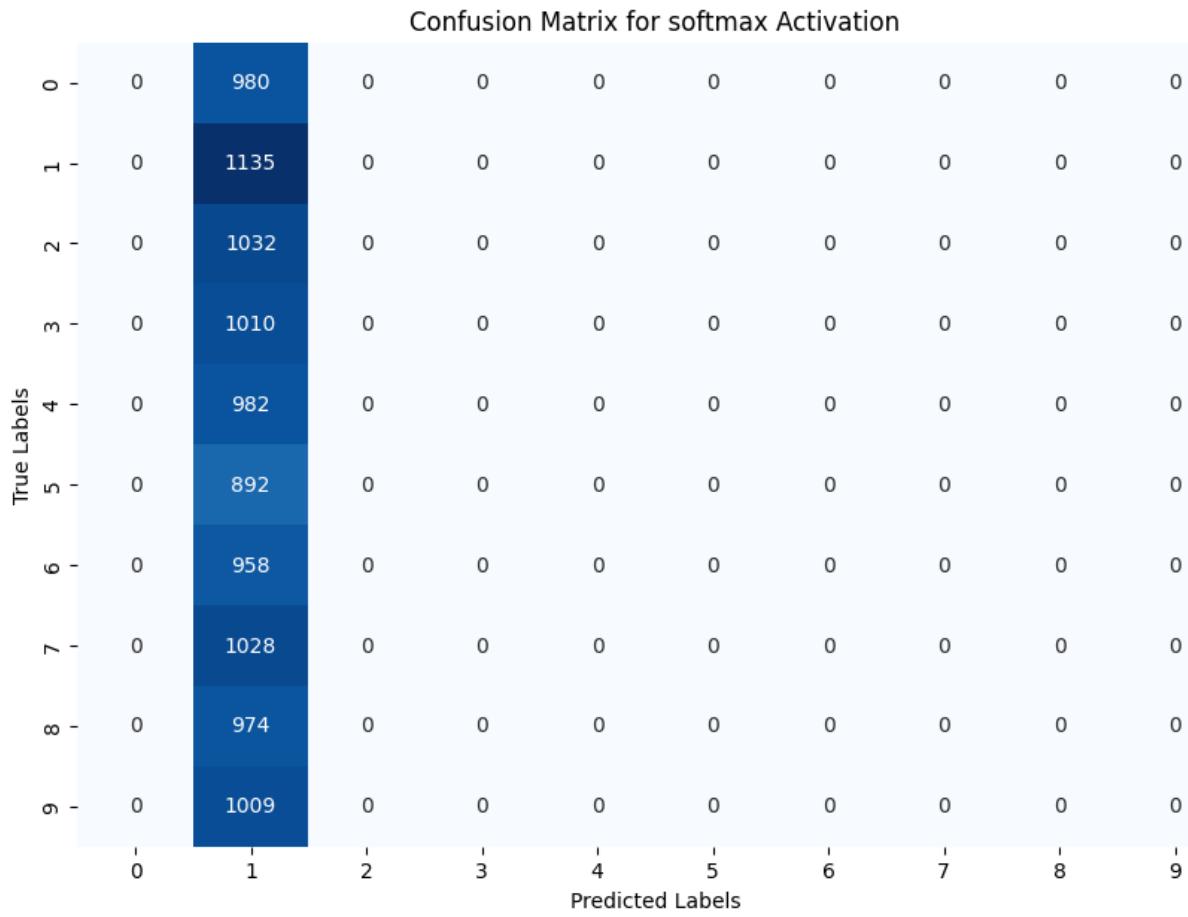
```
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 9s - loss: 2.3016 - accuracy: 0.1125 - val_loss: 2.3028 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 2/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1131 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/15
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3026 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 6/15
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 7/15
```

```
844/844 - 8s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 8/15  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 9/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 10/15  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 11/15  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 12/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 13/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 14/15  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 15/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1122 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 2/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/20
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/20
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 5/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 6/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3027 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 7/20
```

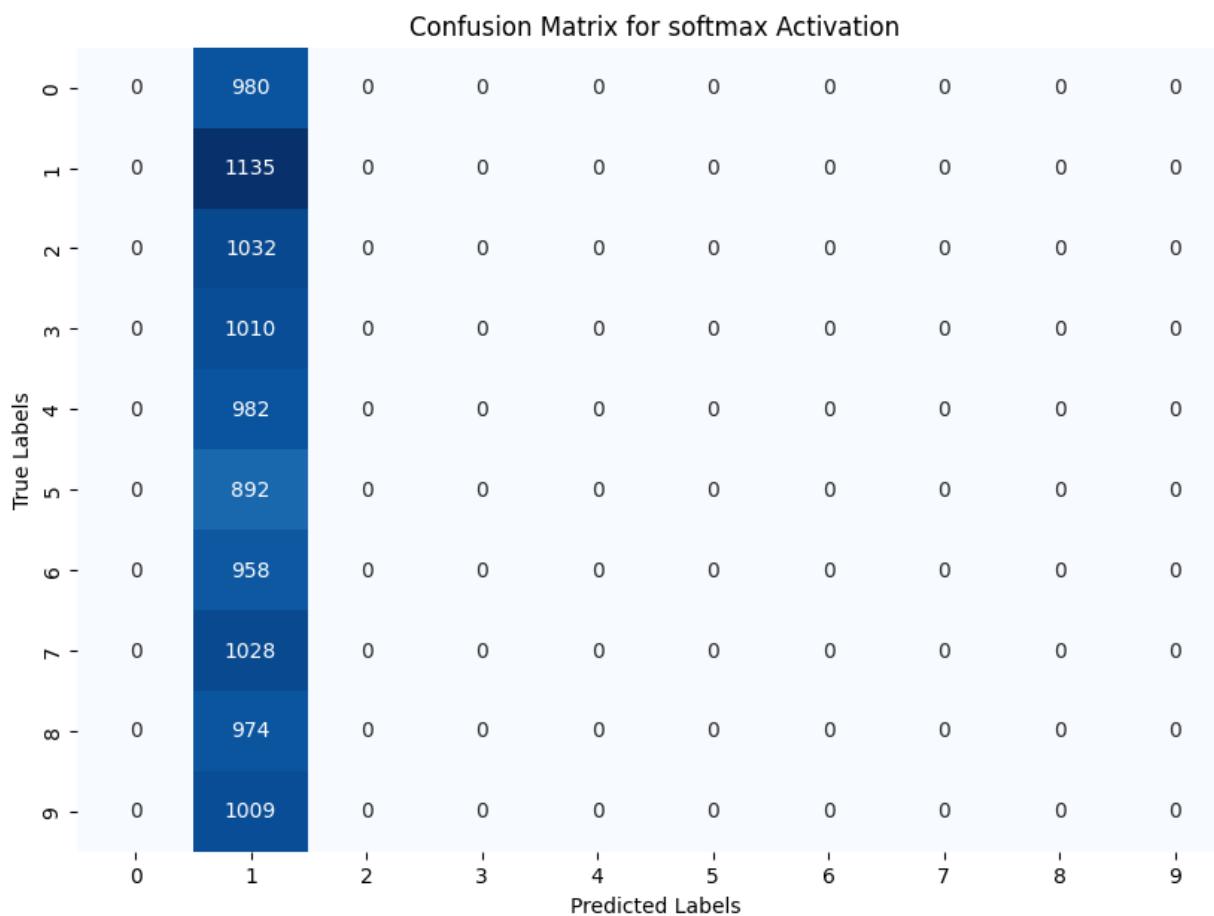
```
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 8/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 9/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 10/20  
844/844 - 8s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3028 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 11/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 12/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 13/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 14/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 15/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 16/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 17/20  
844/844 - 8s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 18/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 19/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 20/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

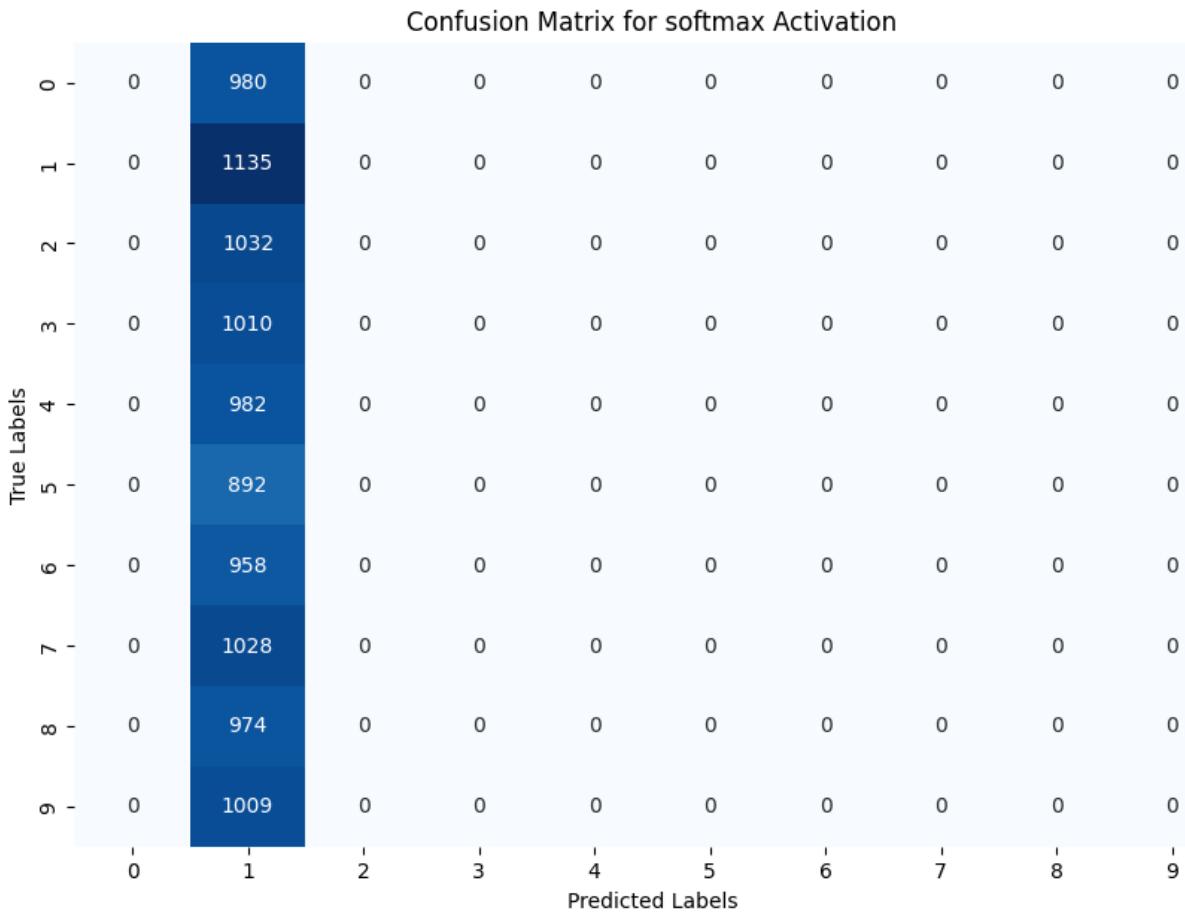
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 6s - loss: 2.3015 - accuracy: 0.1122 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 2/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -
```

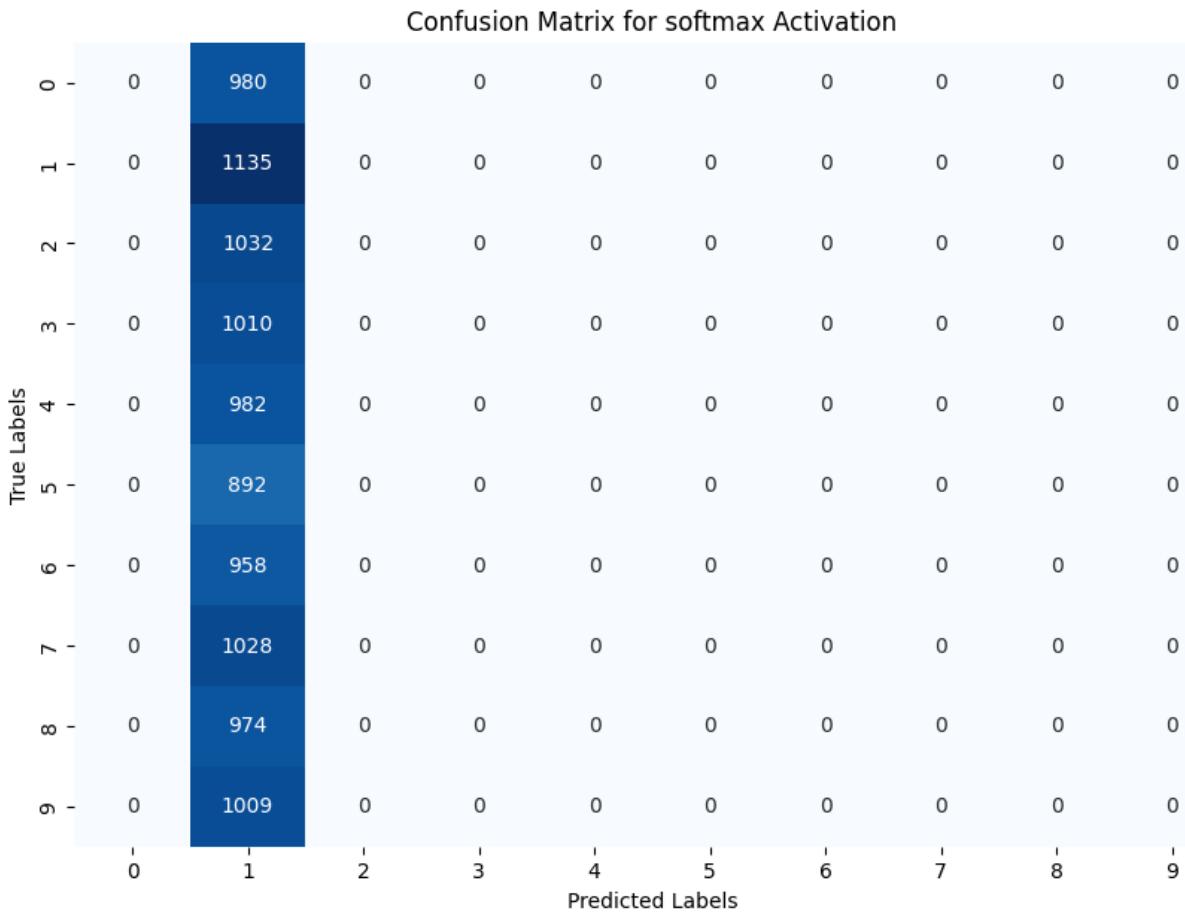
```
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 3/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 4/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 5/5
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 6s - loss: 2.3015 - accuracy: 0.1124 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 2/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 3/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 4/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 5/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 6/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 13ms/step
Epoch 7/15
```

```
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 8/15  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 9/15  
422/422 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 13ms/step  
Epoch 10/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 11/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 12/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 13/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 14/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 15/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
313/313 [=====] - 1s 4ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 7s - loss: 2.3015 - accuracy: 0.1119 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 7s/epoch - 17ms/step
Epoch 2/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 3/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 4/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 5/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 6/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 7/20
```

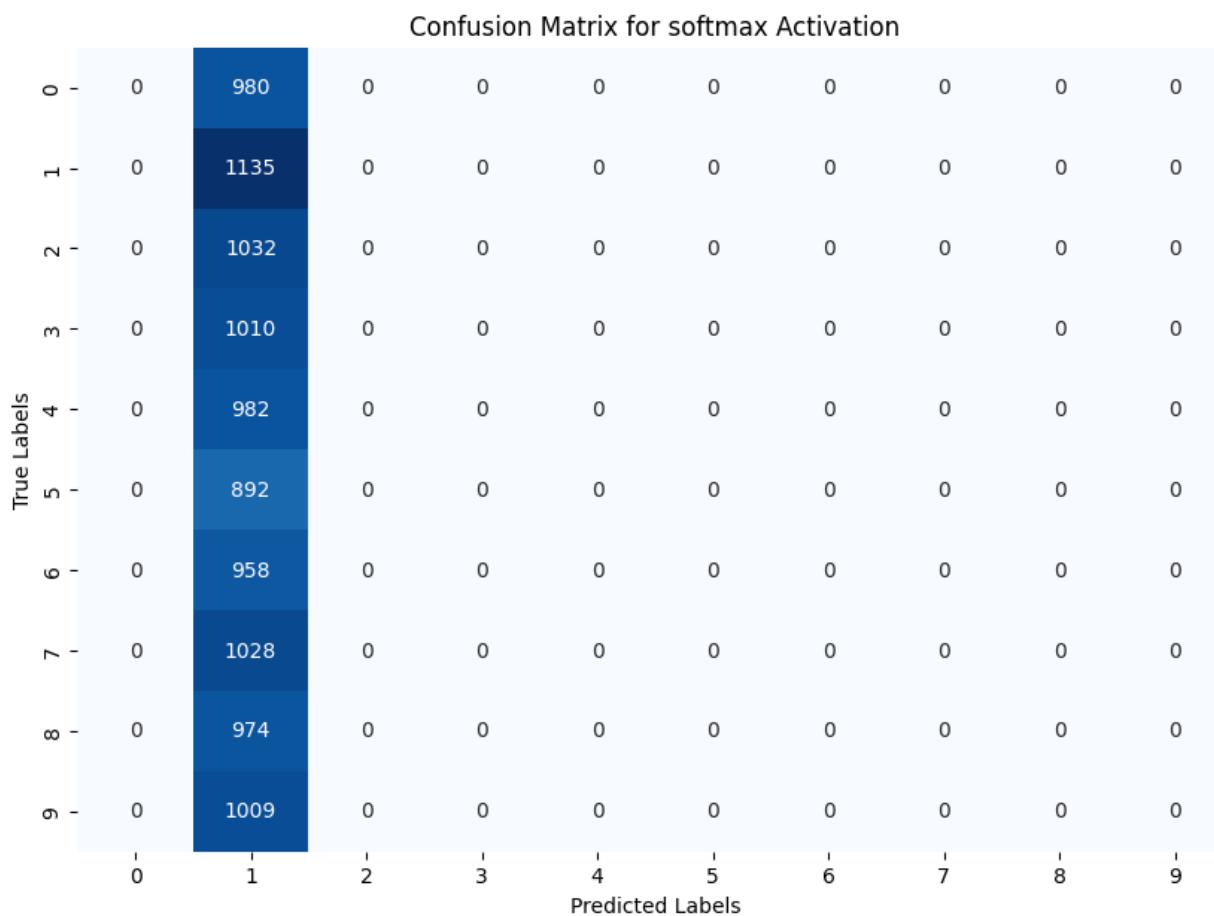
```
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 8/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 9/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 10/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 11/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 12/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 13/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 14/20  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 15/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 16/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 17/20  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 18/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 19/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 13ms/step  
Epoch 20/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 5 epochs..

Epoch 1/5

211/211 - 6s - loss: 2.3016 - accuracy: 0.1108 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 26ms/step

Epoch 2/5

211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -

```
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 3/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 4/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 5/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	980	0	0	0	0	0	0	0	0	0
	1135	0	0	0	0	0	0	0	0	0
	1032	0	0	0	0	0	0	0	0	0
	1010	0	0	0	0	0	0	0	0	0
	982	0	0	0	0	0	0	0	0	0
	892	0	0	0	0	0	0	0	0	0
	958	0	0	0	0	0	0	0	0	0
	1028	0	0	0	0	0	0	0	0	0
	974	0	0	0	0	0	0	0	0	0
	1009	0	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 5s - loss: 2.3016 - accuracy: 0.1121 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 26ms/step
Epoch 2/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 3/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 4/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 5/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 6/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 7/15

```

```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 8/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 9/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 10/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 11/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 12/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 13/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 14/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 15/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0	980	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 6s - loss: 2.3016 - accuracy: 0.1118 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 26ms/step
Epoch 2/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 3/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 4/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 5/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 6/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 7/20

```

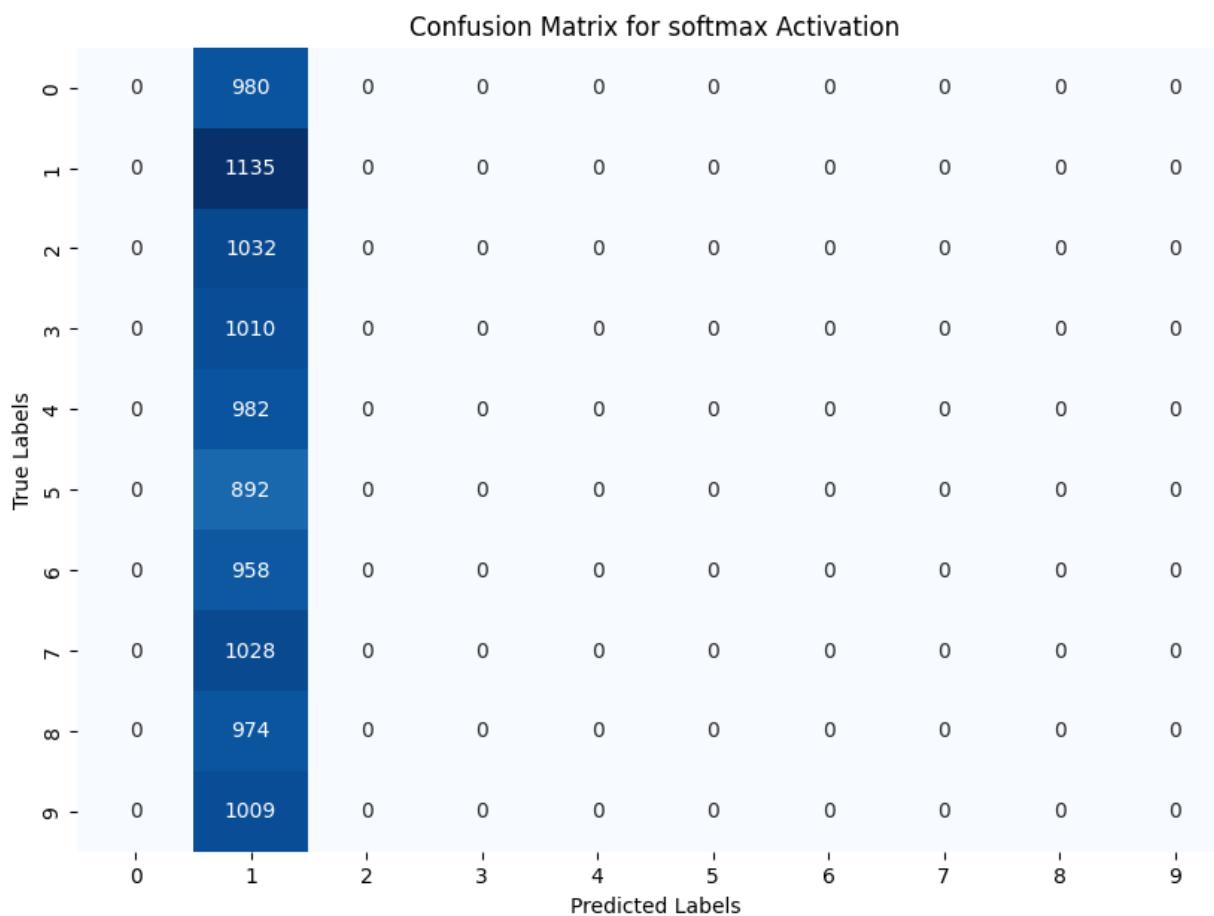
```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 8/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 9/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 10/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 11/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 12/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 13/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 14/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 21ms/step  
Epoch 15/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 16/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 17/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 18/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 19/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 20/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

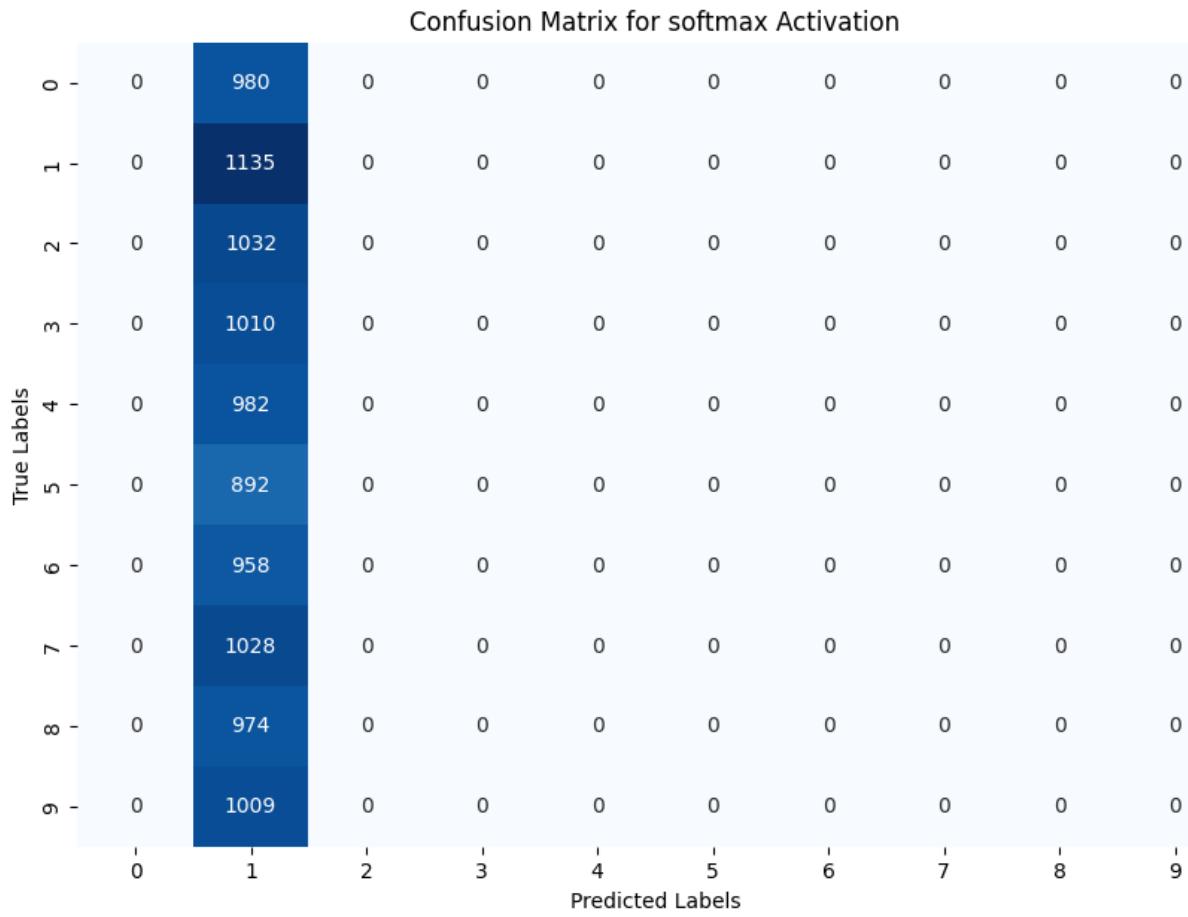
844/844 - 9s - loss: 2.3015 - accuracy: 0.1129 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step

Epoch 2/5

844/844 - 8s - loss: 2.3015 - accuracy: 0.1125 - val_loss: 2.3027 -

```
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/5
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/5
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3030 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 9s - loss: 2.3016 - accuracy: 0.1119 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 2/15
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3028 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 6/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 7/15
```

```
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 8/15  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 9/15  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 10/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 11/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 12/15  
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 13/15  
844/844 - 9s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 14/15  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
313/313 [=====] - 1s 4ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0 - 0	980	0	0	0	0	0	0	0	0
	1 - 0	1135	0	0	0	0	0	0	0	0
	2 - 0	1032	0	0	0	0	0	0	0	0
	3 - 0	1010	0	0	0	0	0	0	0	0
	4 - 0	982	0	0	0	0	0	0	0	0
	5 - 0	892	0	0	0	0	0	0	0	0
	6 - 0	958	0	0	0	0	0	0	0	0
	7 - 0	1028	0	0	0	0	0	0	0	0
	8 - 0	974	0	0	0	0	0	0	0	0
	9 - 0	1009	0	0	0	0	0	0	0	0

```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 9s - loss: 2.3016 - accuracy: 0.1126 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 2/20
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 4/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 6/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 7/20
```

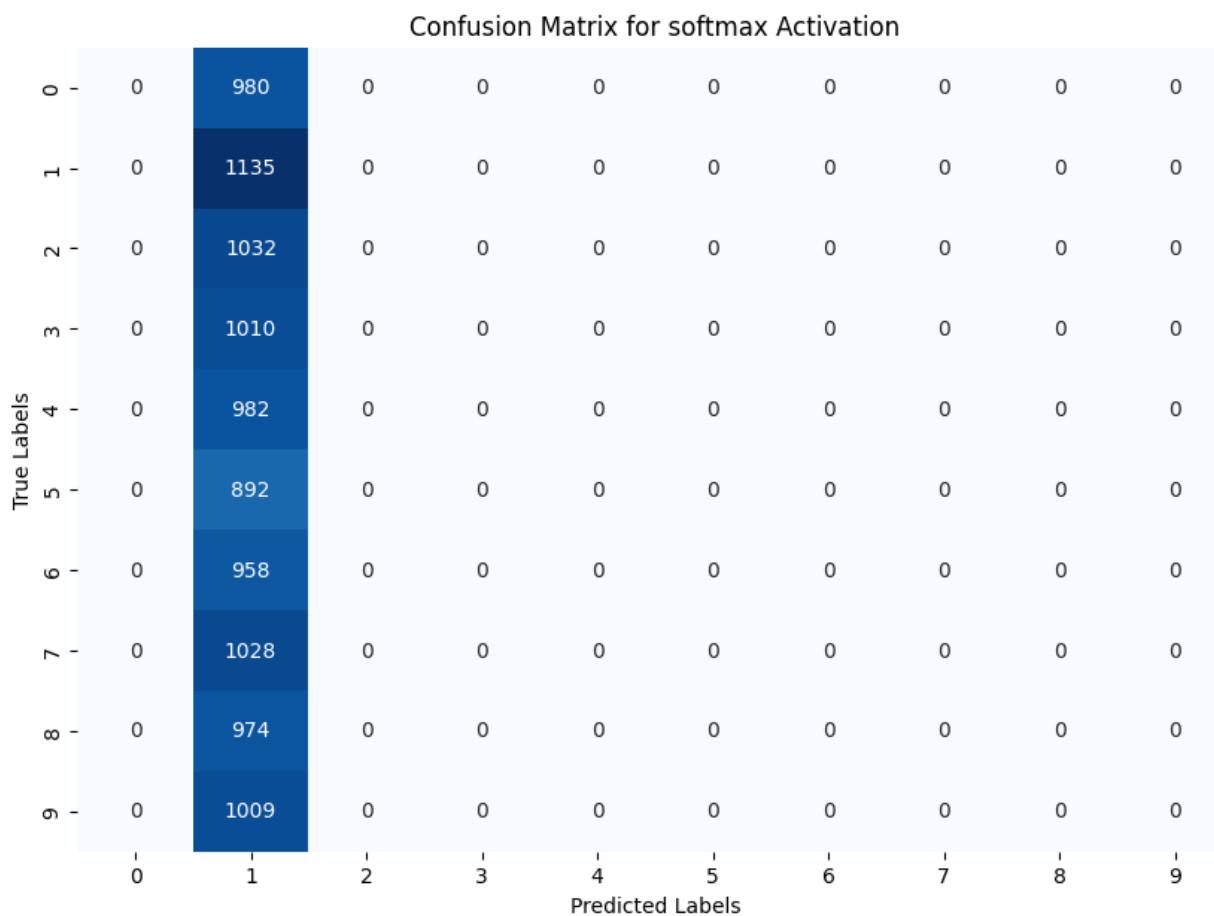
```
844/844 - 8s - loss: 2.3013 - accuracy: 0.1130 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 8/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 9/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 10/20  
844/844 - 8s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 11/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 12/20  
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 13/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 14/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 15/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 16/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1130 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 17/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 18/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 19/20  
844/844 - 8s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 20/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1130 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

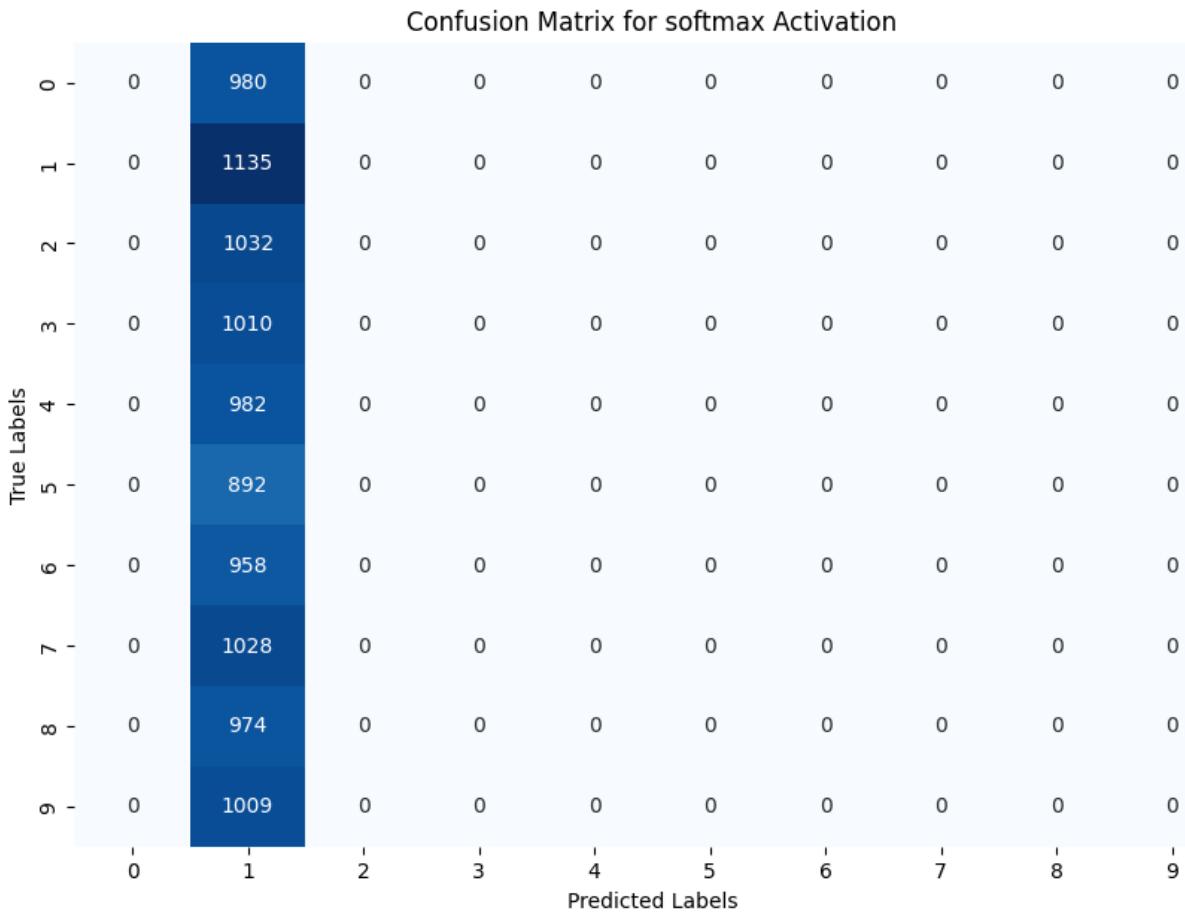
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 8s - loss: 2.3015 - accuracy: 0.1125 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 18ms/step
Epoch 2/5
422/422 - 6s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
```

```
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 3/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 4/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 5/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

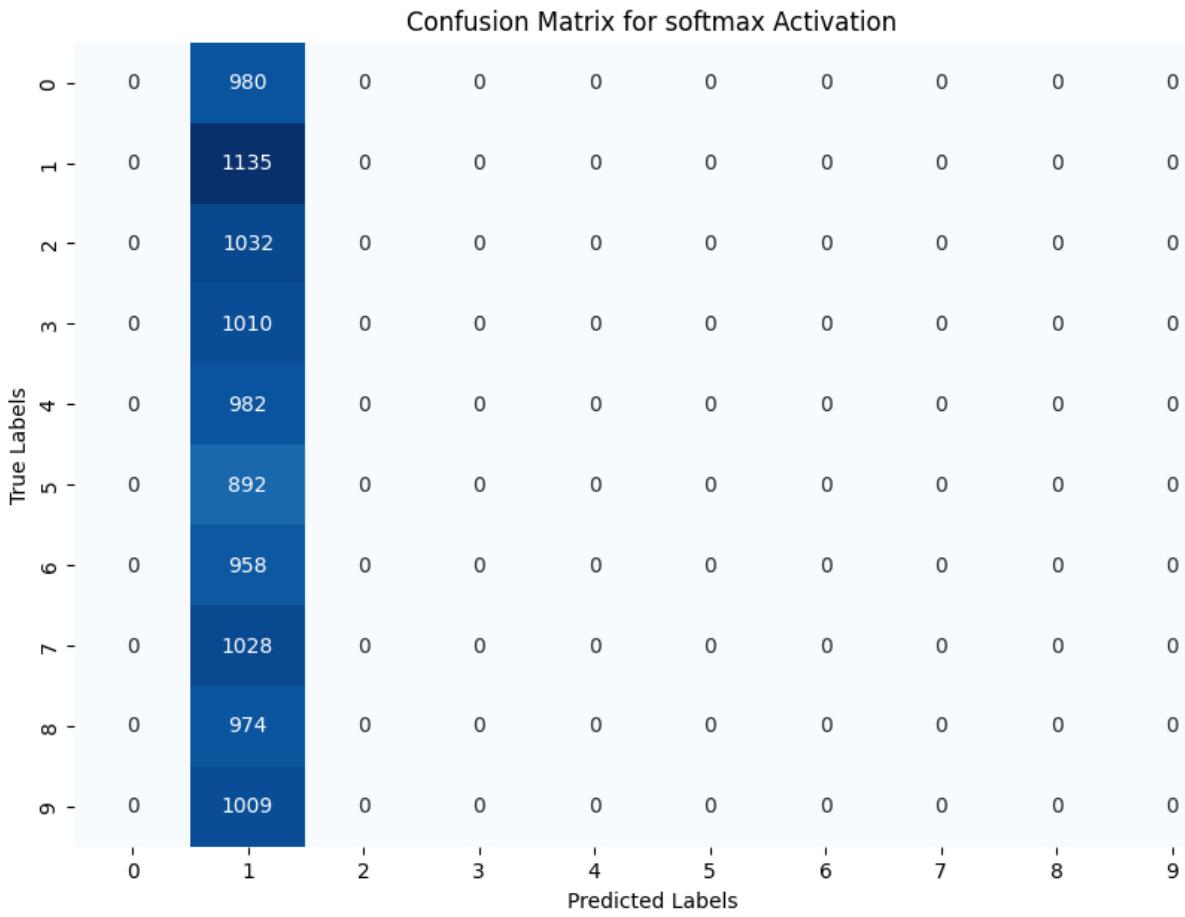


```

Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 7s - loss: 2.3015 - accuracy: 0.1127 - val_loss: 2.3016 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 2/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 3/15
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 4/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 5/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 6/15
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 7/15

```

```
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 8/15  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 9/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 10/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 11/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 12/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 13/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 14/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 15/15  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```

Training Model with softmax activation, 2 conv_layers, 2 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 7s - loss: 2.3015 - accuracy: 0.1124 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 2/20
422/422 - 6s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 3/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 4/20
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 5/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 6/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 7/20

```

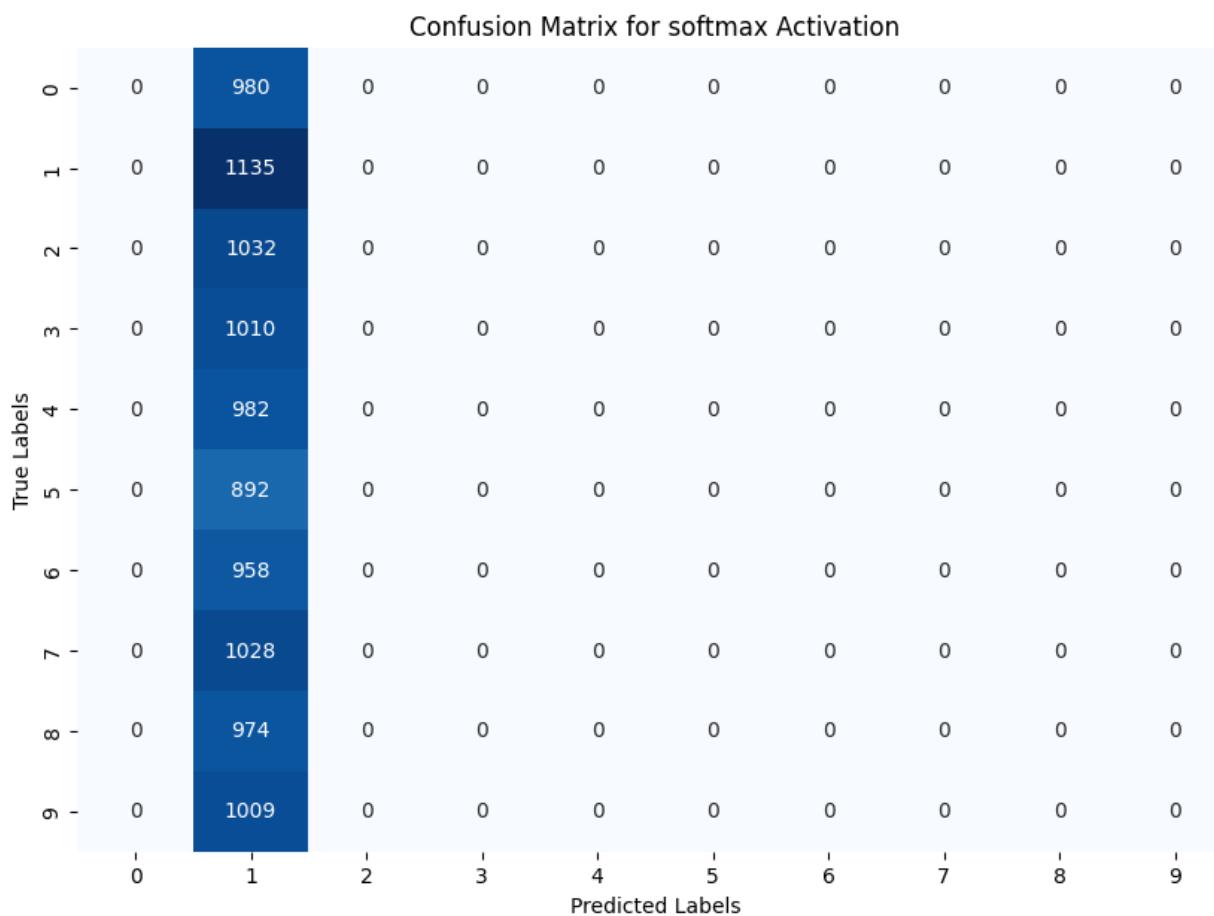
```
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 8/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 9/20  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 10/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 11/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 12/20  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 13/20  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 14/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 15/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 16/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 17/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 18/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 19/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 20/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 6s - loss: 2.3016 - accuracy: 0.1117 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 26ms/step
Epoch 2/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
```

```
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 3/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 4/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 5/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0	980	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 5s - loss: 2.3016 - accuracy: 0.1122 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 26ms/step
Epoch 2/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 3/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 4/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 5/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 6/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 7/15

```

```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 8/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 9/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 10/15  
211/211 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 11/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 12/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 13/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 14/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 15/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0 - 0	980	0	0	0	0	0	0	0	0
	1 - 0	1135	0	0	0	0	0	0	0	0
	2 - 0	1032	0	0	0	0	0	0	0	0
	3 - 0	1010	0	0	0	0	0	0	0	0
	4 - 0	982	0	0	0	0	0	0	0	0
	5 - 0	892	0	0	0	0	0	0	0	0
	6 - 0	958	0	0	0	0	0	0	0	0
	7 - 0	1028	0	0	0	0	0	0	0	0
	8 - 0	974	0	0	0	0	0	0	0	0
	9 - 0	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 2 conv_layers, 2 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 5s - loss: 2.3016 - accuracy: 0.1115 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 26ms/step
Epoch 2/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 21ms/step
Epoch 3/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 4/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 5/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 6/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 7/20

```

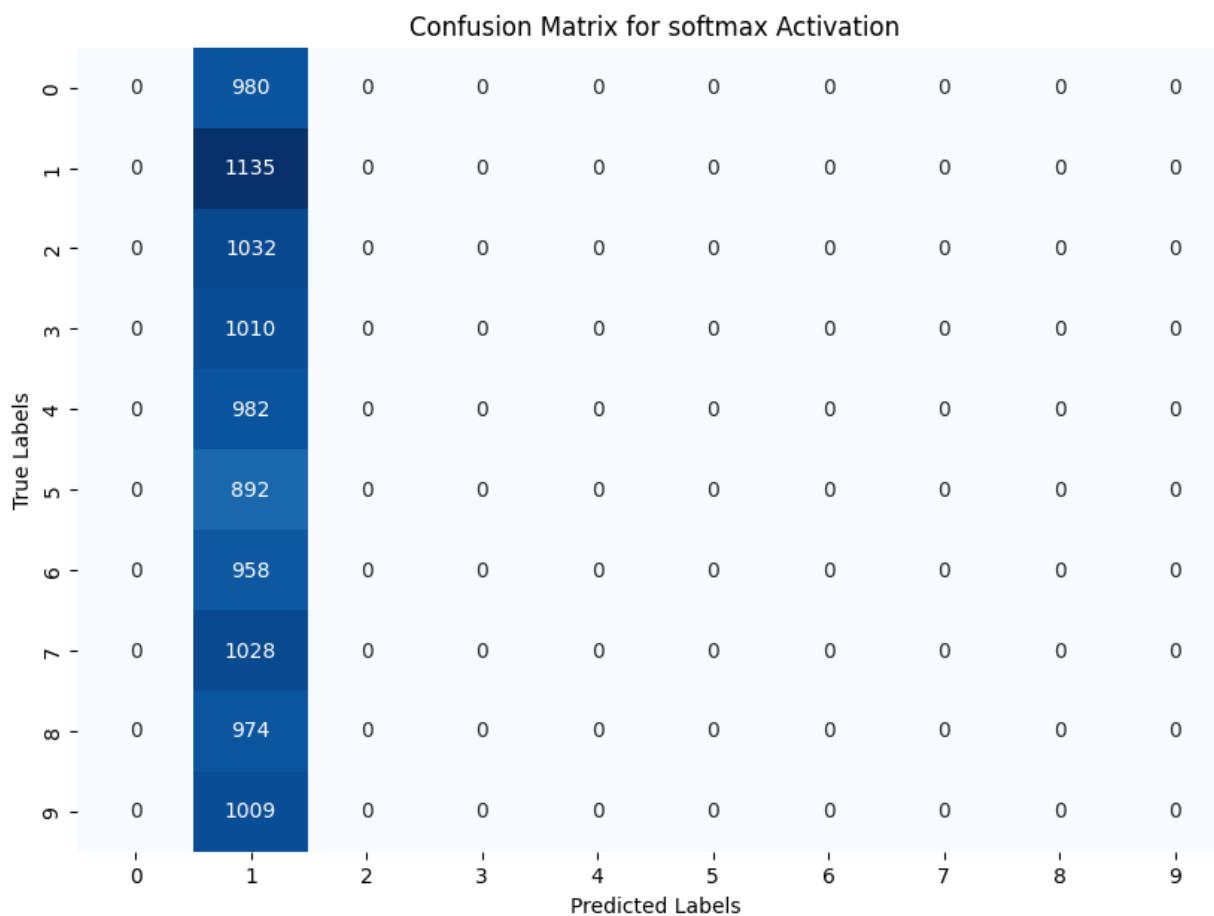
```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 8/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 9/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 10/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 11/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 12/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 13/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 14/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 15/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 16/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 17/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 18/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 19/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 20/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

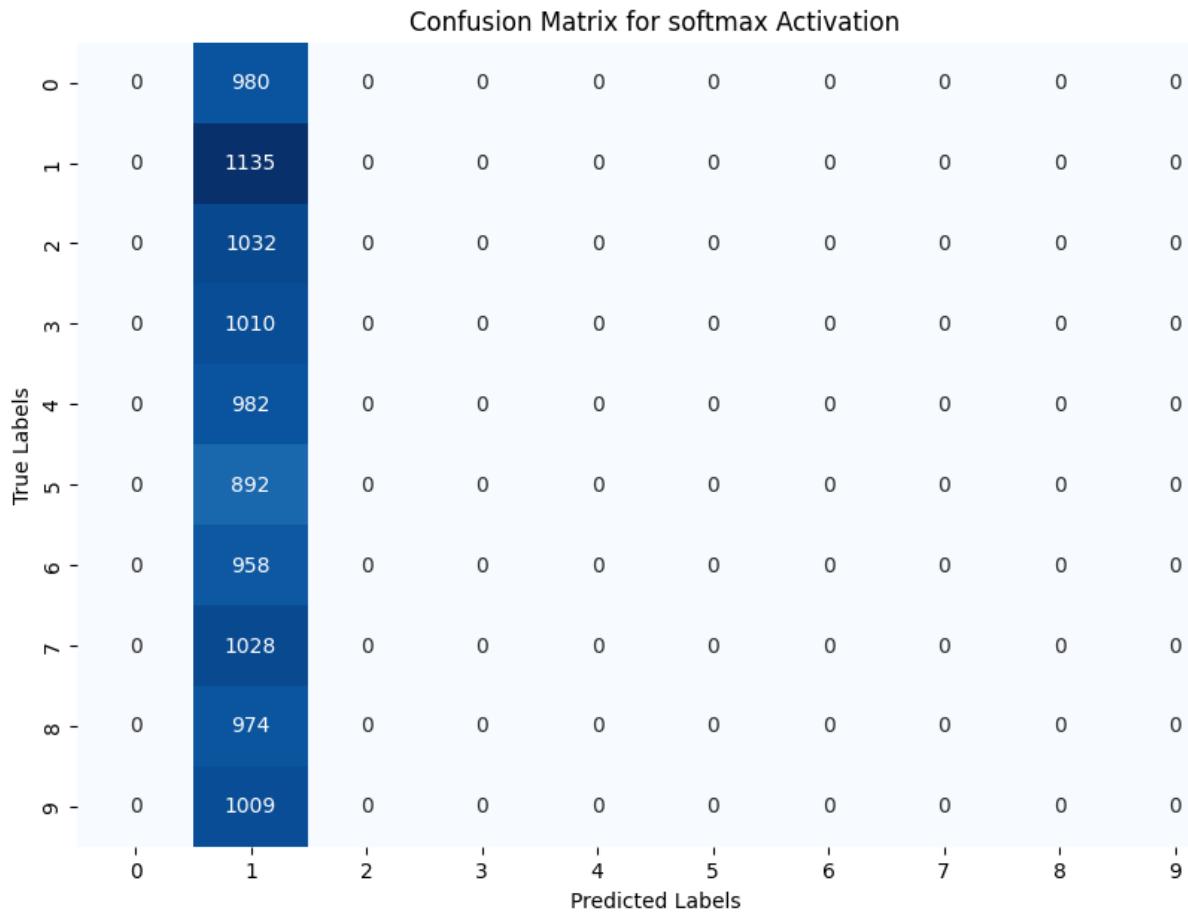
844/844 - 9s - loss: 2.3015 - accuracy: 0.1119 - val_loss: 2.3026 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step

Epoch 2/5

844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -

```
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3027 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/5
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

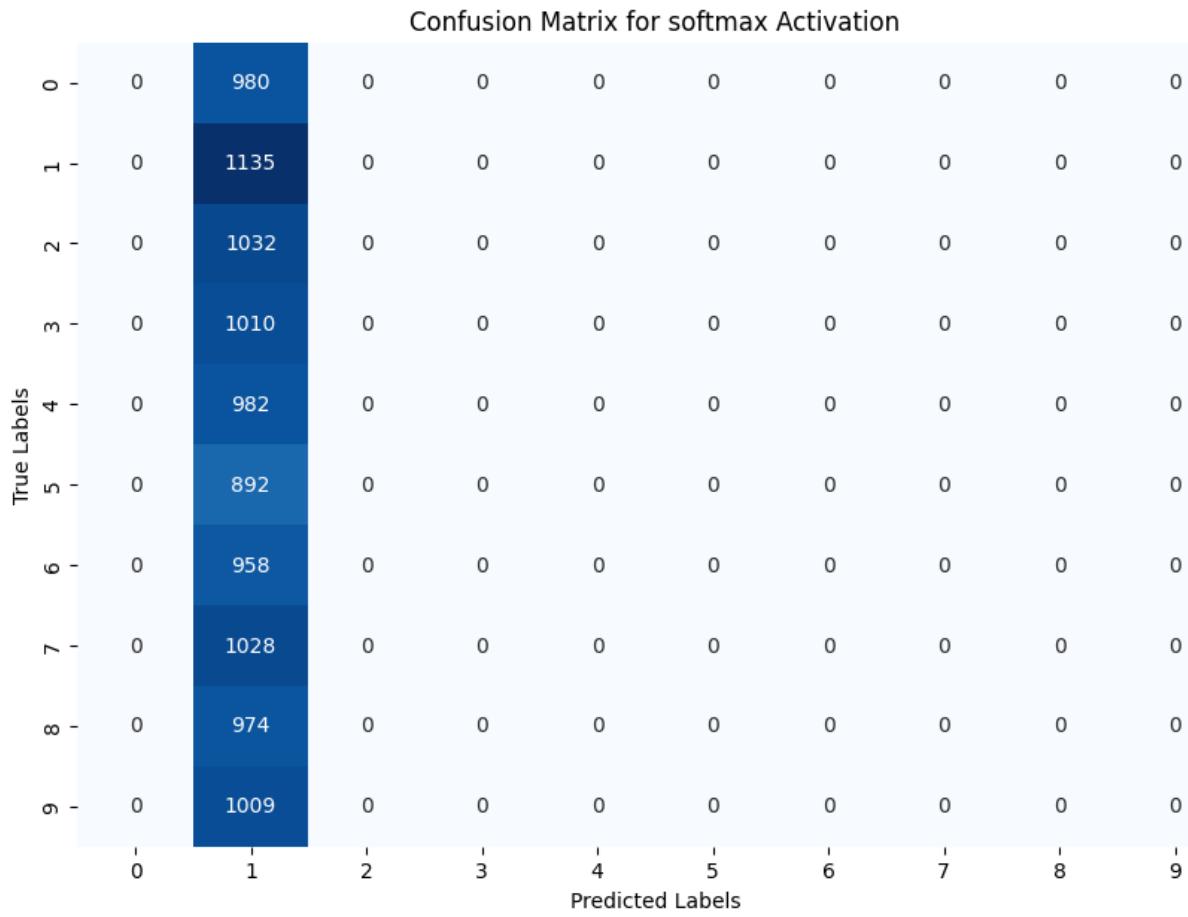


```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 9s - loss: 2.3017 - accuracy: 0.1125 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 2/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3028 -
val_accuracy: 0.1050 - 8s/epoch - 9ms/step
Epoch 4/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/15
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 6/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 7/15
```

```
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 8/15  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 9/15  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 10/15  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 11/15  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3029 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 12/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 13/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1126 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 14/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 15/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```


```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3027 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 2/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3015 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1124 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 6/20
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 7/20
```

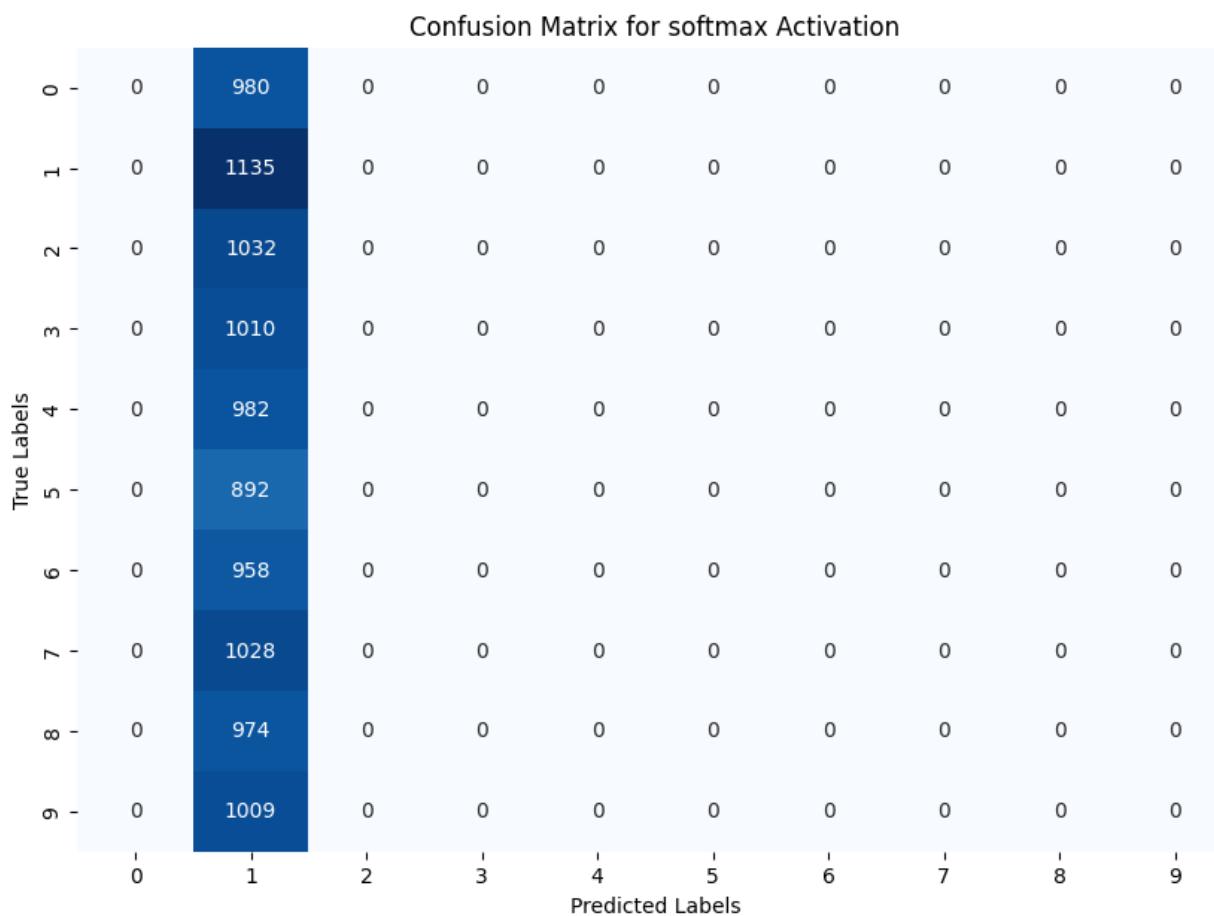
```
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 8s/epoch - 9ms/step  
Epoch 8/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 9/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 10/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 11/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 12/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3016 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 13/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 14/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 15/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 16/20  
844/844 - 8s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 17/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 18/20  
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 19/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 20/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

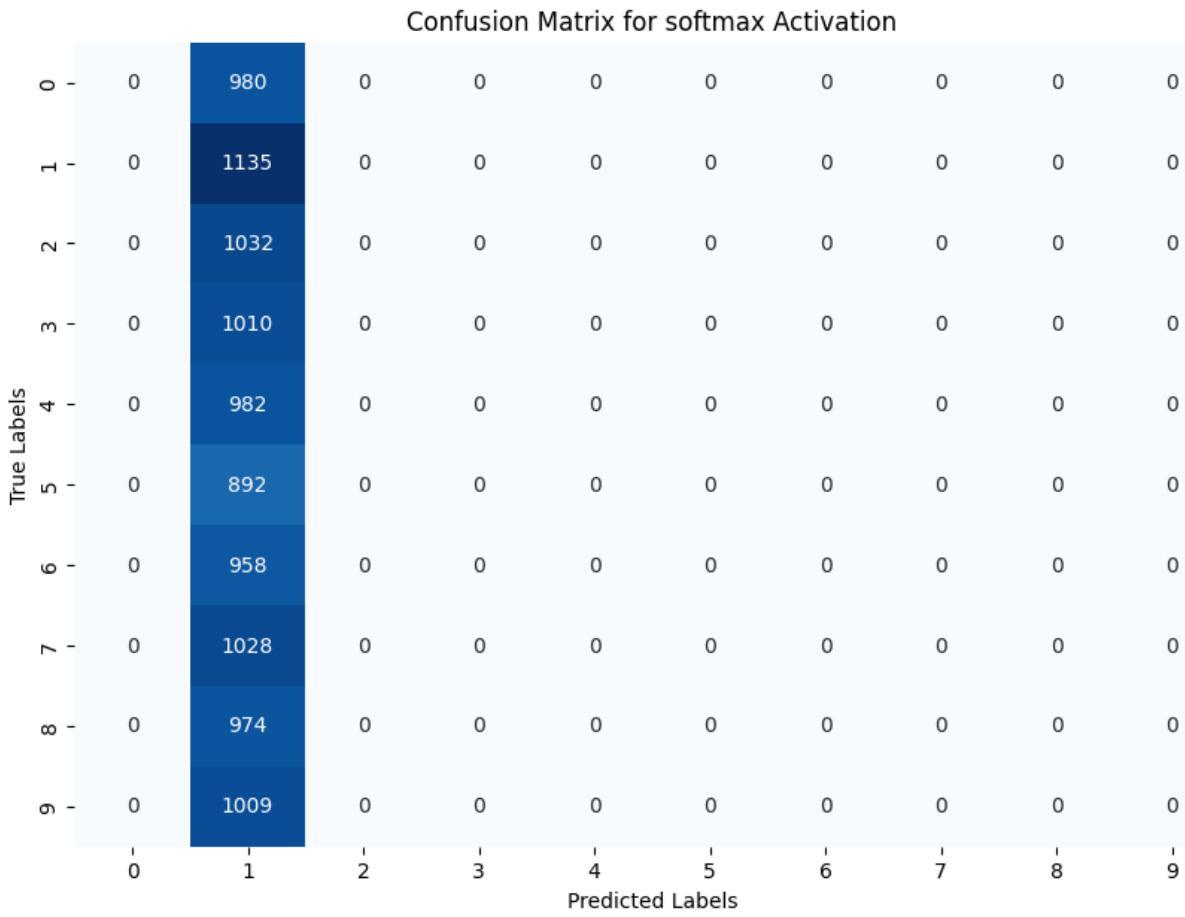
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 7s - loss: 2.3014 - accuracy: 0.1124 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 2/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -
```

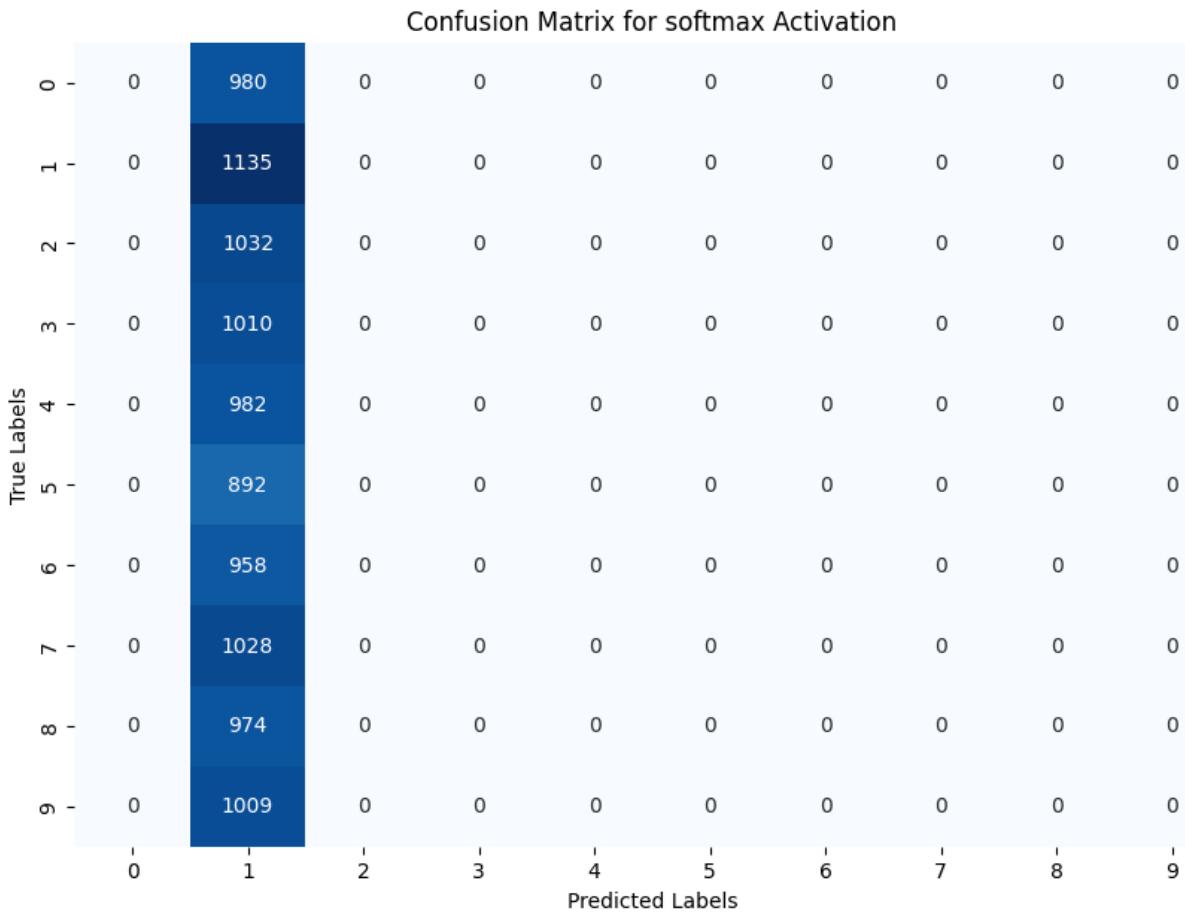
```
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 3/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 4/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 5/5
422/422 - 6s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 7s - loss: 2.3015 - accuracy: 0.1122 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 2/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 3/15
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 4/15
422/422 - 6s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 5/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 6/15
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 7/15
```

```
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 8/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 9/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 10/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 11/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 12/15  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 13/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 14/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 15/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 6s - loss: 2.3015 - accuracy: 0.1116 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 2/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 3/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 4/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 5/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 6/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 7/20
```

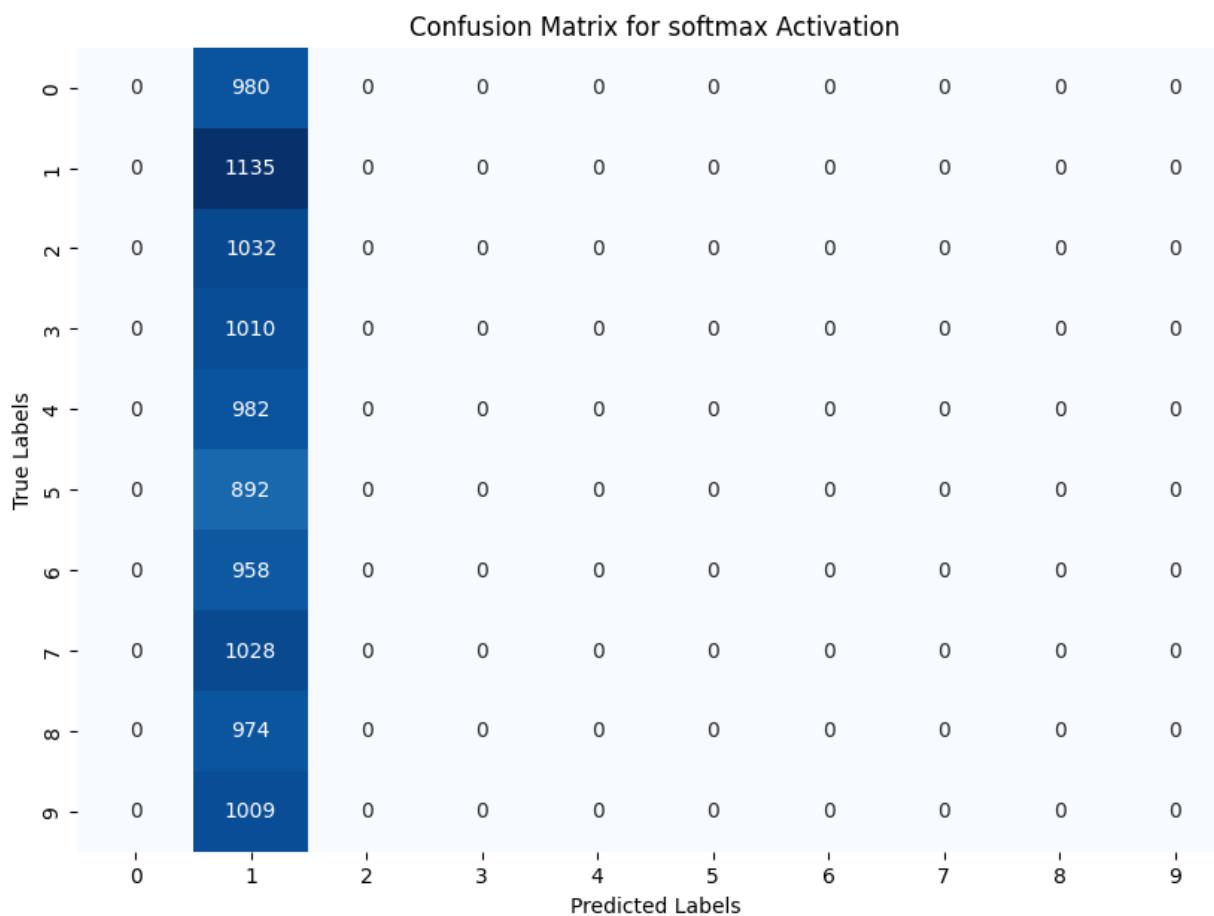
```
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 8/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 9/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 10/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 11/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 12/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 13/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 14/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 15/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 16/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 17/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 18/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 19/20  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 20/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

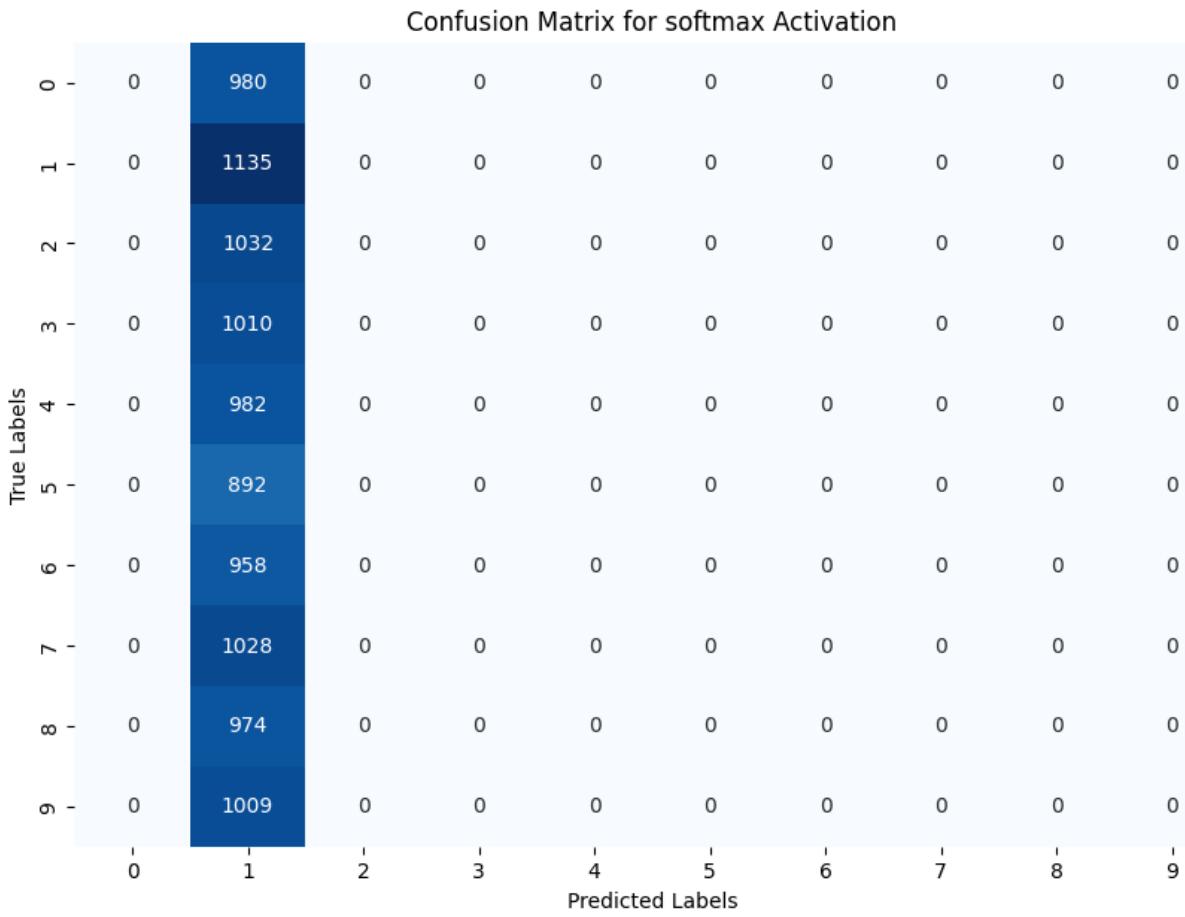
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 6s - loss: 2.3017 - accuracy: 0.1115 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 27ms/step
Epoch 2/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
```

```
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 3/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 4/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 5/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 7s - loss: 2.3018 - accuracy: 0.1099 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 31ms/step
Epoch 2/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 3/15
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 4/15
211/211 - 4s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 4s/epoch - 21ms/step
Epoch 5/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 6/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 7/15
```

```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 8/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 9/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 10/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 11/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 12/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 13/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 14/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 15/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0	980	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 6s - loss: 2.3016 - accuracy: 0.1108 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 27ms/step
Epoch 2/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 3/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 4/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 5/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 6/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 7/20

```

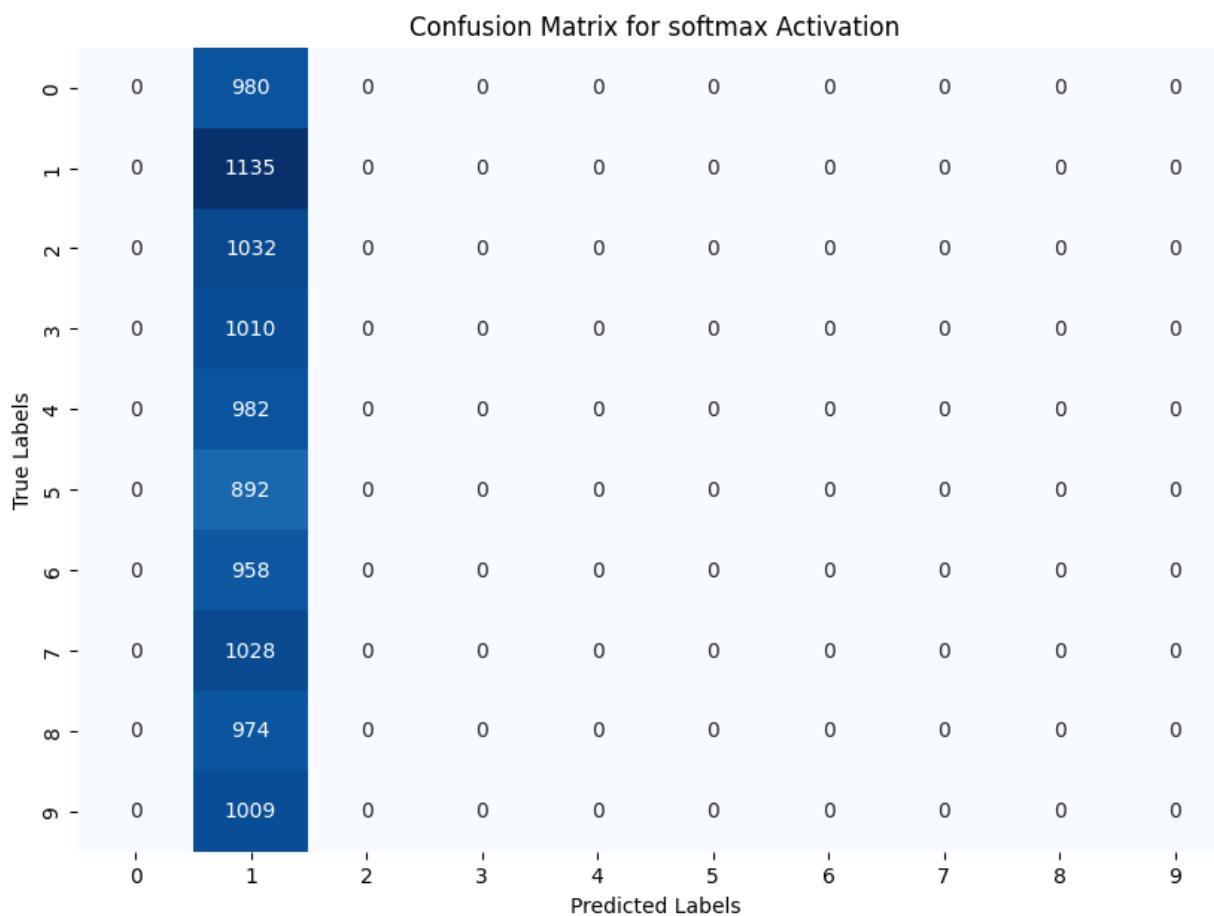
```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 8/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 9/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 10/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 11/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 12/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 13/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 14/20  
211/211 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 15/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 16/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 17/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 22ms/step  
Epoch 18/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 19/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 20/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

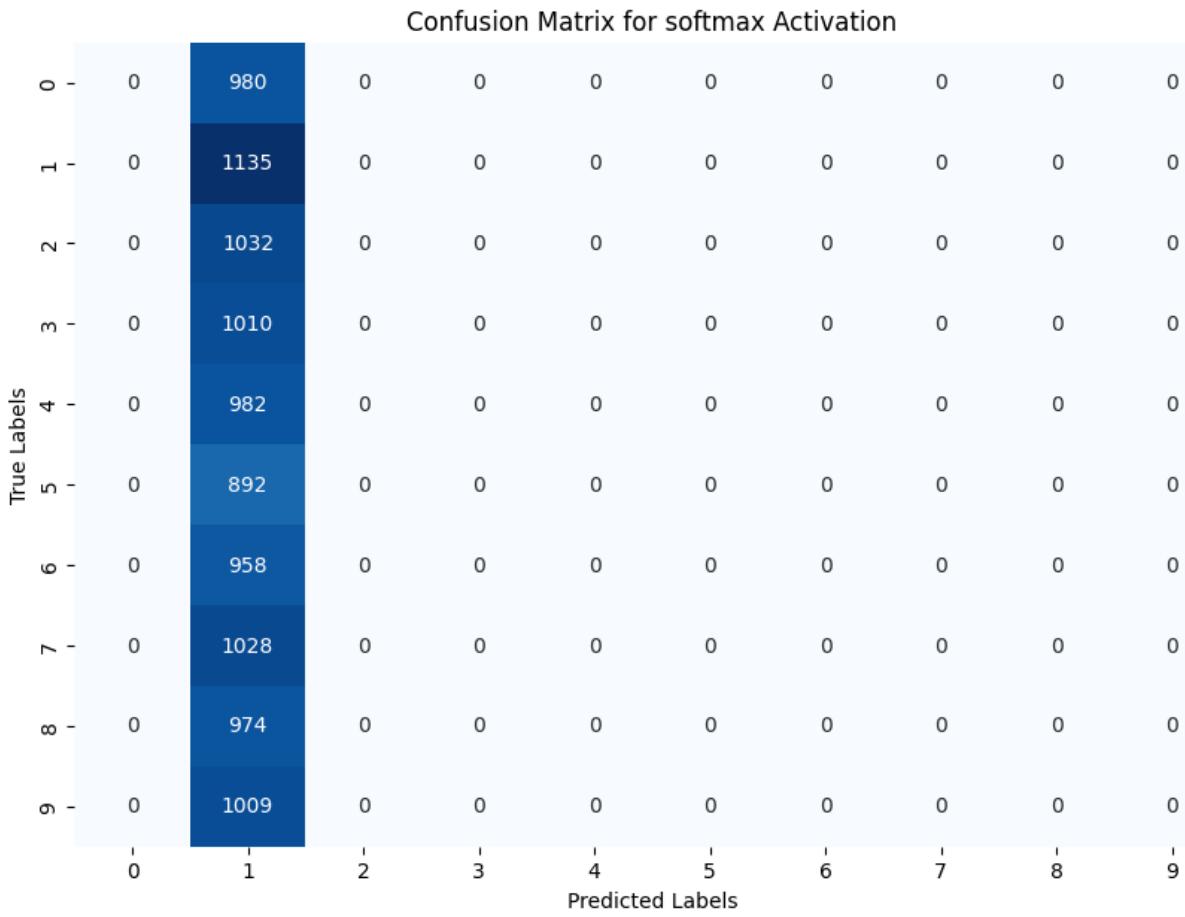
844/844 - 9s - loss: 2.3015 - accuracy: 0.1130 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step

Epoch 2/5

844/844 - 8s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3023 -

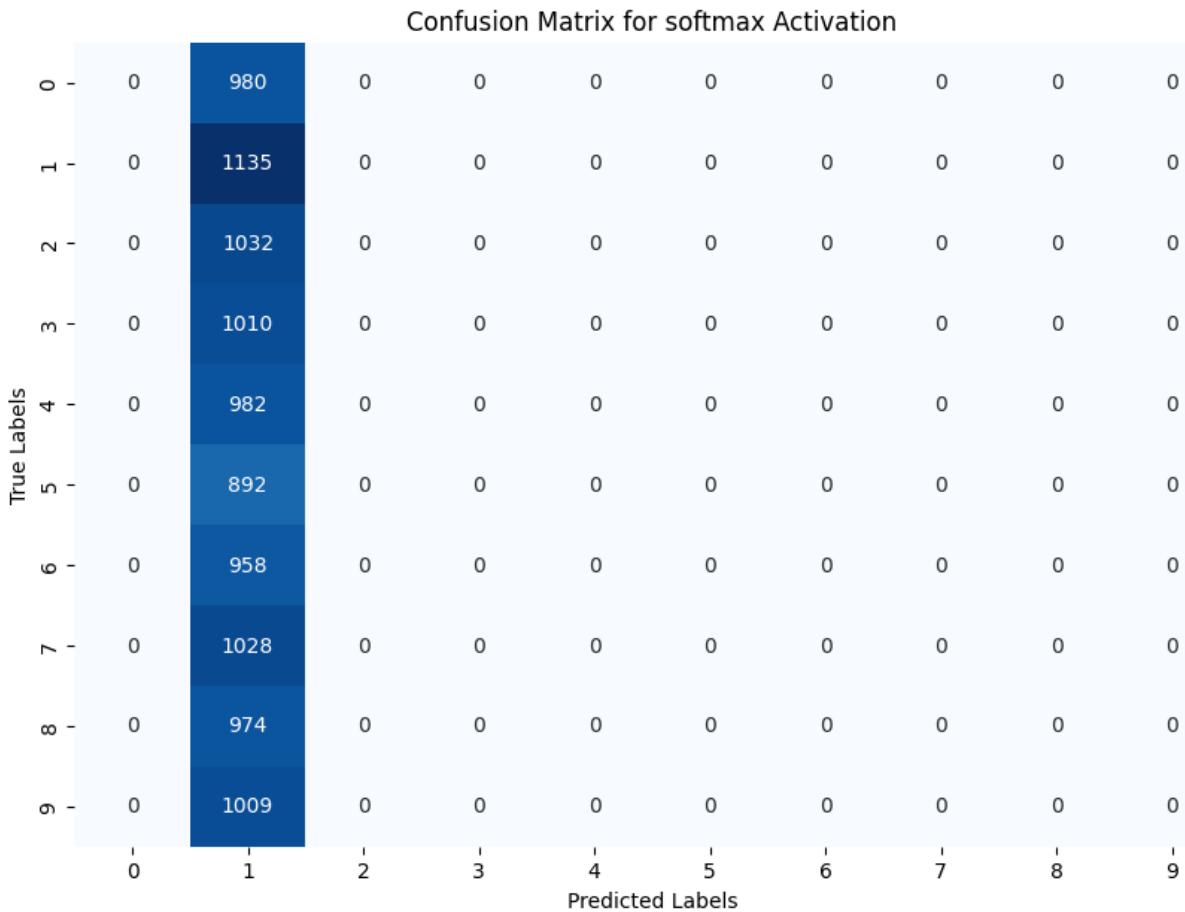
```
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/5
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/5
844/844 - 8s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/5
844/844 - 8s - loss: 2.3016 - accuracy: 0.1123 - val_loss: 2.3027 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 9s - loss: 2.3016 - accuracy: 0.1127 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 2/15
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 3/15
844/844 - 8s - loss: 2.3014 - accuracy: 0.1125 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 4/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 5/15
844/844 - 8s - loss: 2.3015 - accuracy: 0.1129 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 6/15
844/844 - 8s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3028 -
val_accuracy: 0.1050 - 8s/epoch - 10ms/step
Epoch 7/15
```

```
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 8/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 9/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 10/15  
844/844 - 8s - loss: 2.3014 - accuracy: 0.1126 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 11/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 12/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 13/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 14/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1130 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 15/15  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 9s - loss: 2.3017 - accuracy: 0.1126 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 2/20
844/844 - 9s - loss: 2.3014 - accuracy: 0.1128 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 3/20
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 4/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3026 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 5/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 6/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 9s/epoch - 10ms/step
Epoch 7/20
```

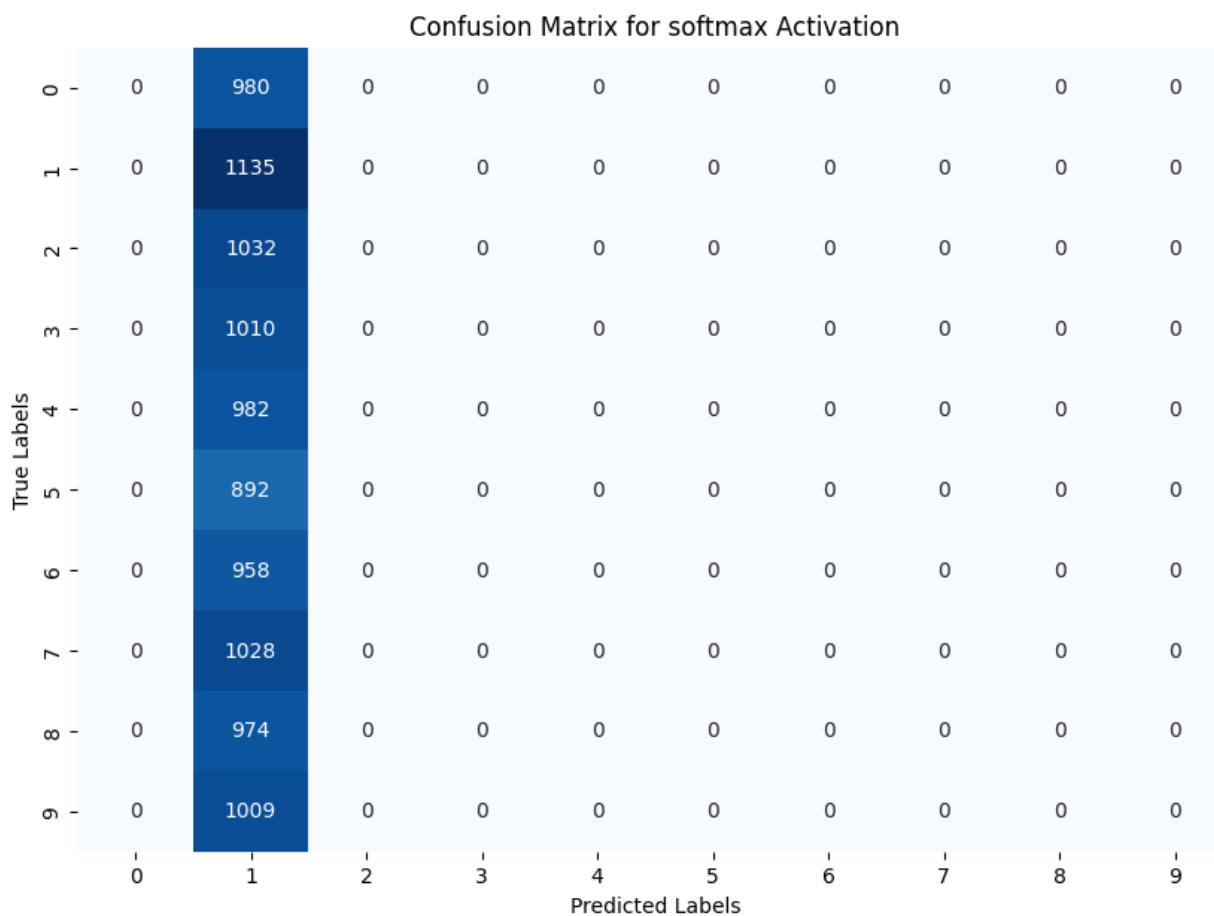
```
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 8/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 9/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 10/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1127 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 11/20  
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 12/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 13/20  
844/844 - 9s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 14/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1125 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 15/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 16/20  
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 17/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 18/20  
844/844 - 8s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 8s/epoch - 10ms/step  
Epoch 19/20  
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
Epoch 20/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1126 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 9s/epoch - 10ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

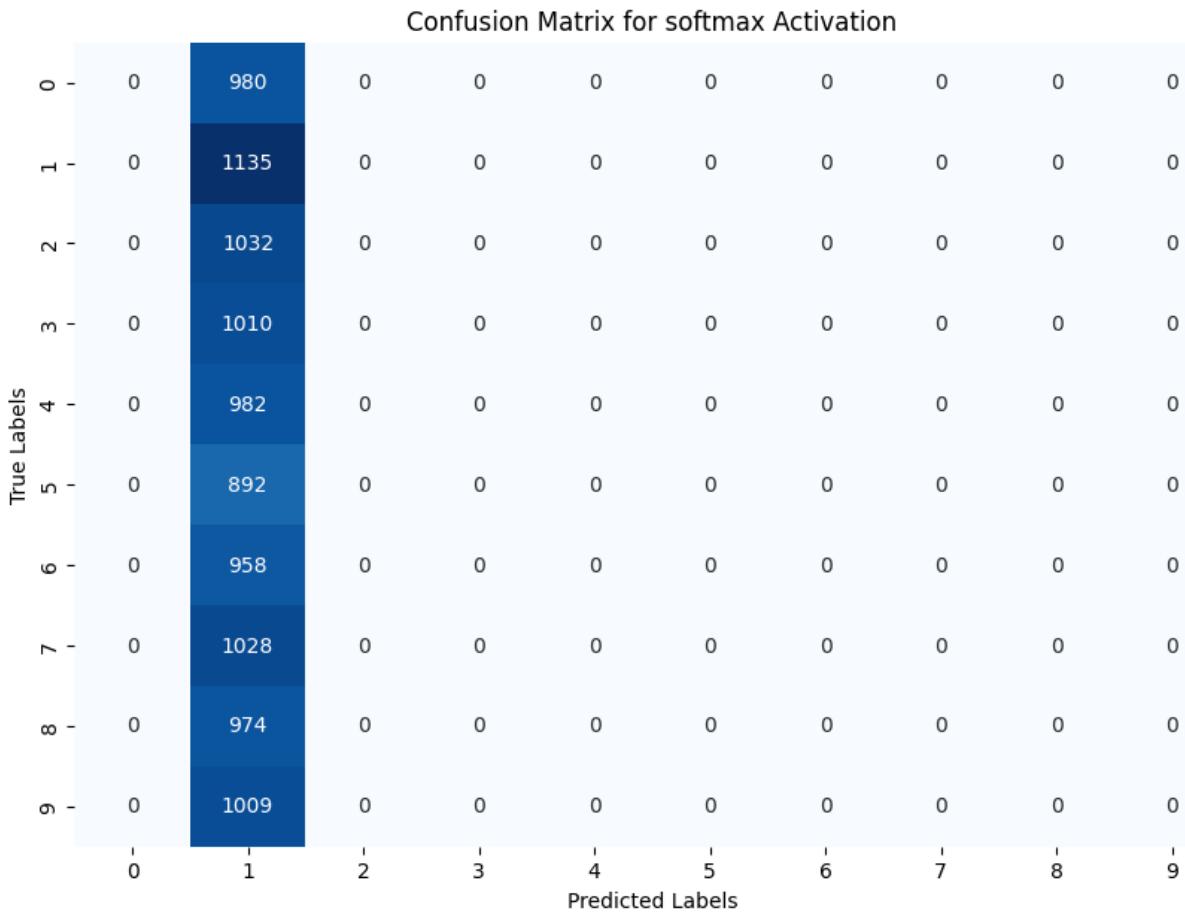
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 7s - loss: 2.3015 - accuracy: 0.1125 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 7s/epoch - 17ms/step
Epoch 2/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -
```

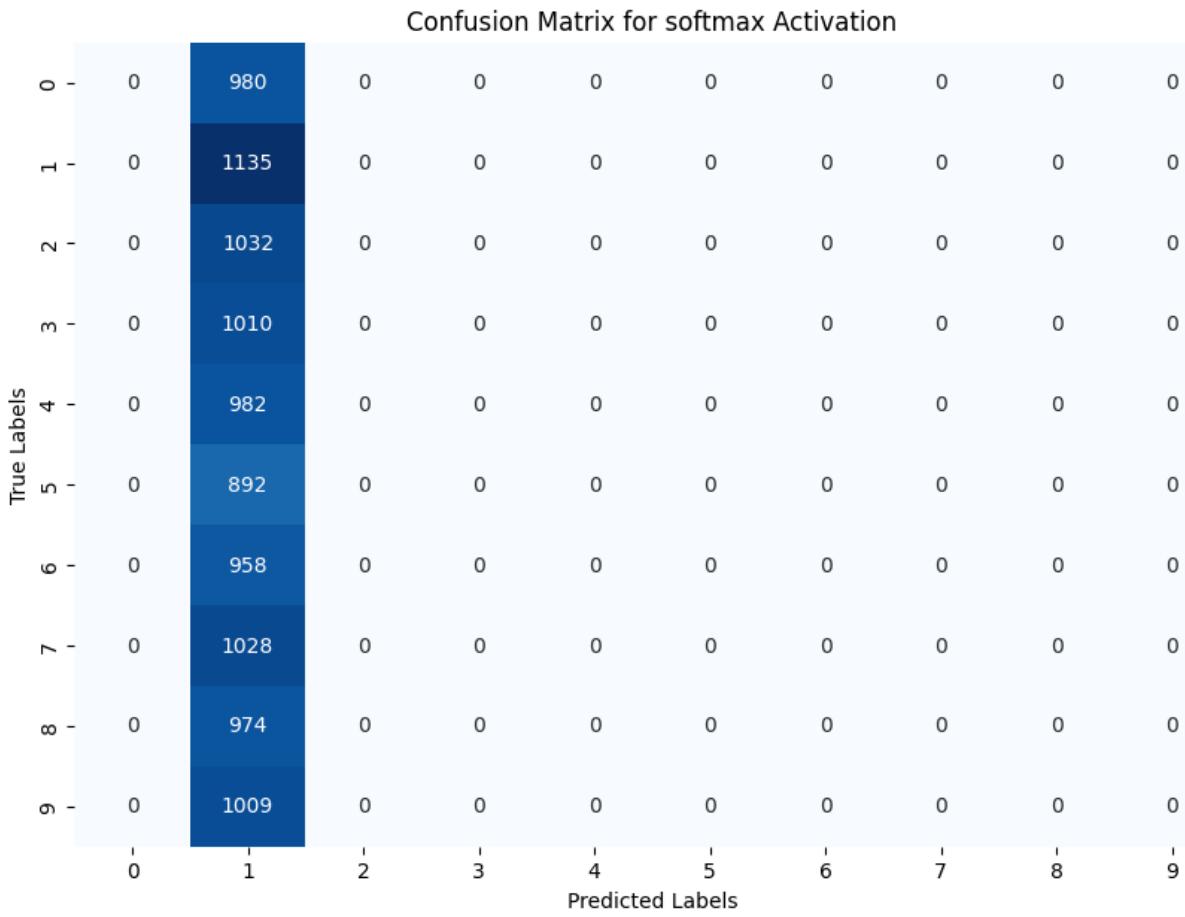
```
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 3/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 4/5
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 5/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3027 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 7s - loss: 2.3016 - accuracy: 0.1116 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 7s/epoch - 17ms/step
Epoch 2/15
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 3/15
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 4/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 5/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 6/15
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 7/15
```

```
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 8/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 9/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 10/15  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 11/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3028 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 12/15  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 13/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 14/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 14ms/step  
Epoch 15/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```

Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 7s - loss: 2.3015 - accuracy: 0.1117 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 7s/epoch - 17ms/step
Epoch 2/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 6s/epoch - 14ms/step
Epoch 3/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3025 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 4/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 5/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 6/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3026 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 7/20

```

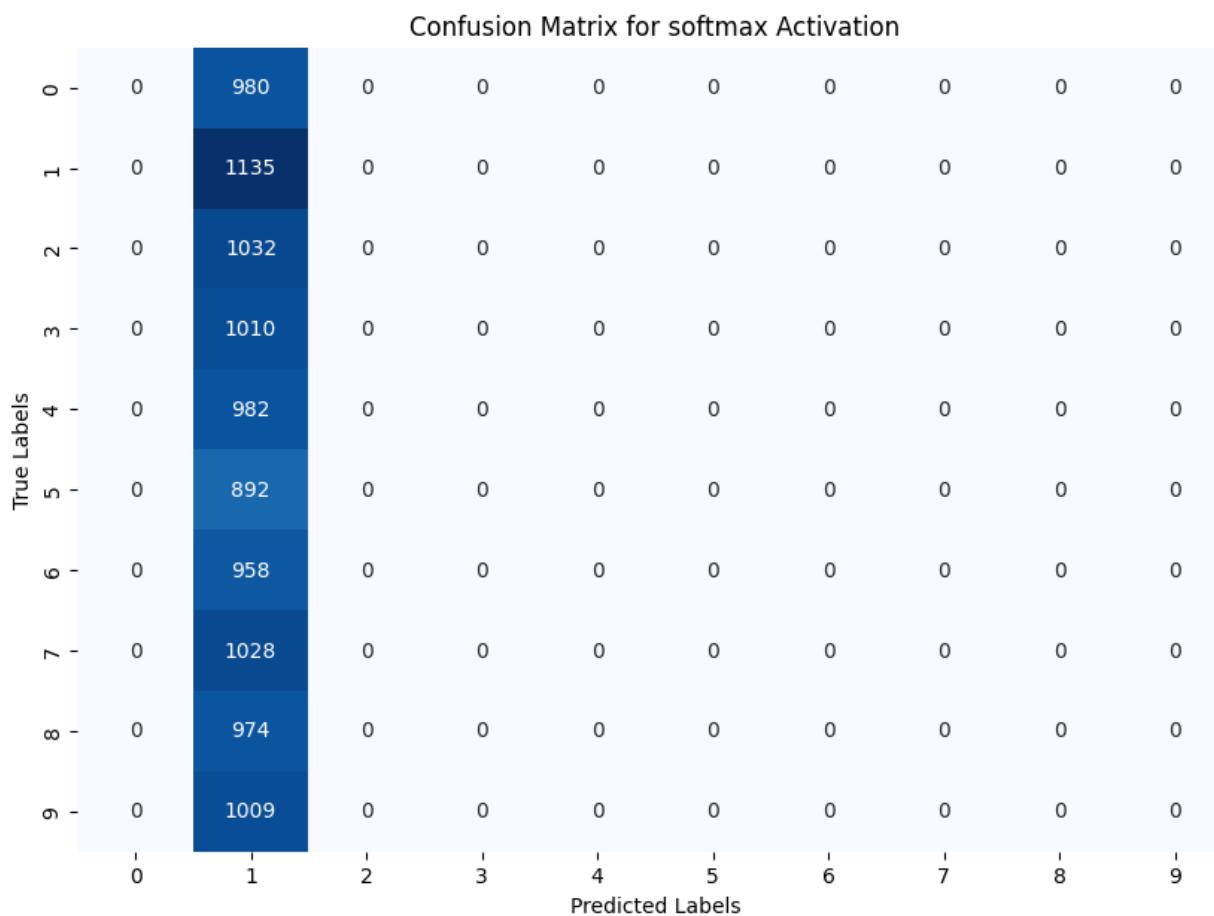
```
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 8/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 9/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 10/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 11/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 12/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 13/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 14/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 15/20  
422/422 - 6s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 16/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 17/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 18/20  
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 19/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 20/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 6s - loss: 2.3018 - accuracy: 0.1105 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 29ms/step
Epoch 2/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
```

```
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 3/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 4/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 5/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
313/313 [=====] - 1s 3ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0	980	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 6s - loss: 2.3016 - accuracy: 0.1111 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 28ms/step
Epoch 2/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 22ms/step
Epoch 3/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 4/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 5/15
211/211 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 23ms/step
Epoch 6/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 7/15

```

```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 8/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 9/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 10/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 11/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 12/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 13/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 14/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 15/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0	980	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 6s - loss: 2.3017 - accuracy: 0.1110 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 28ms/step
Epoch 2/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 3/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 4/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 5/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 6/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 7/20

```

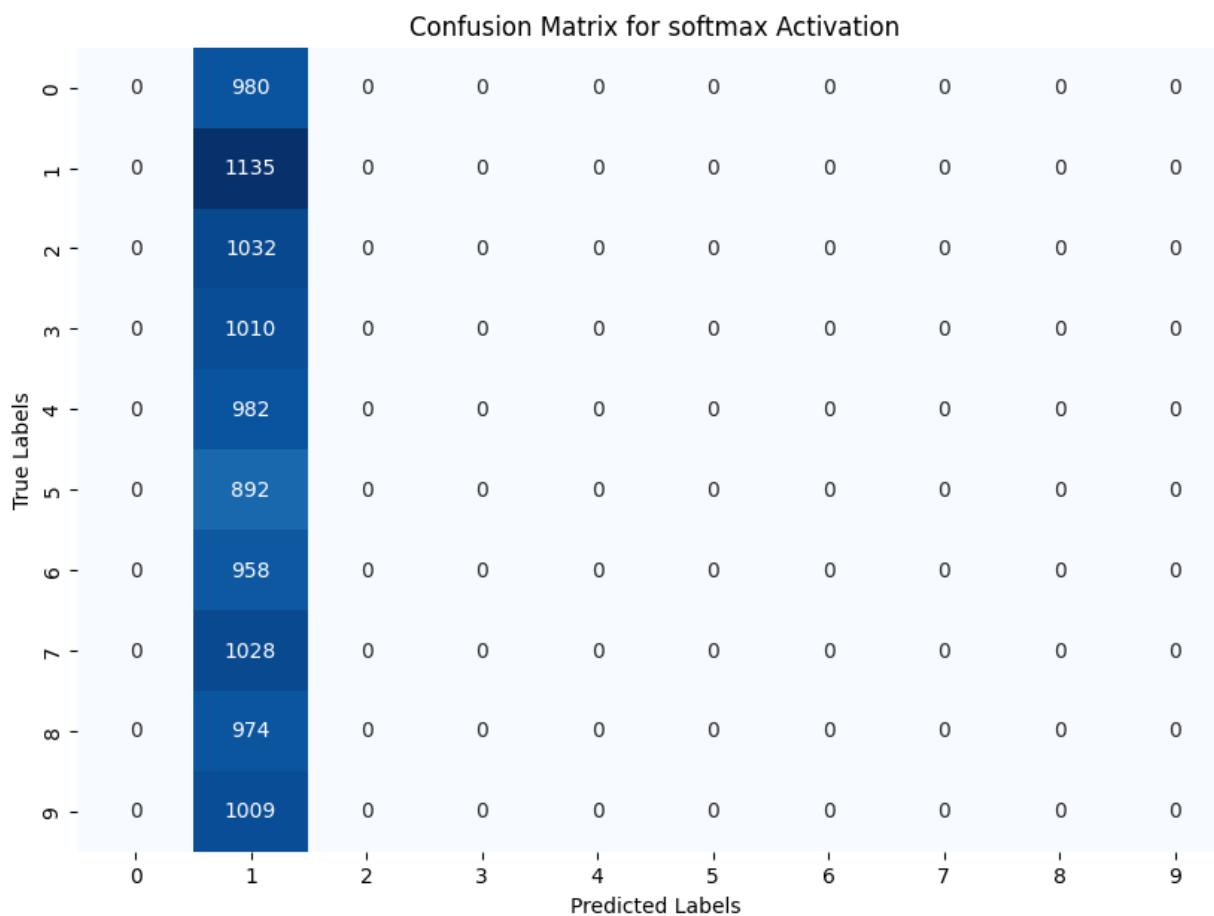
```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 8/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 9/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 10/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 11/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 12/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 13/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 14/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 15/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 16/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 17/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 18/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 19/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
Epoch 20/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 23ms/step  
313/313 [=====] - 1s 3ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 5 epochs..

Epoch 1/5

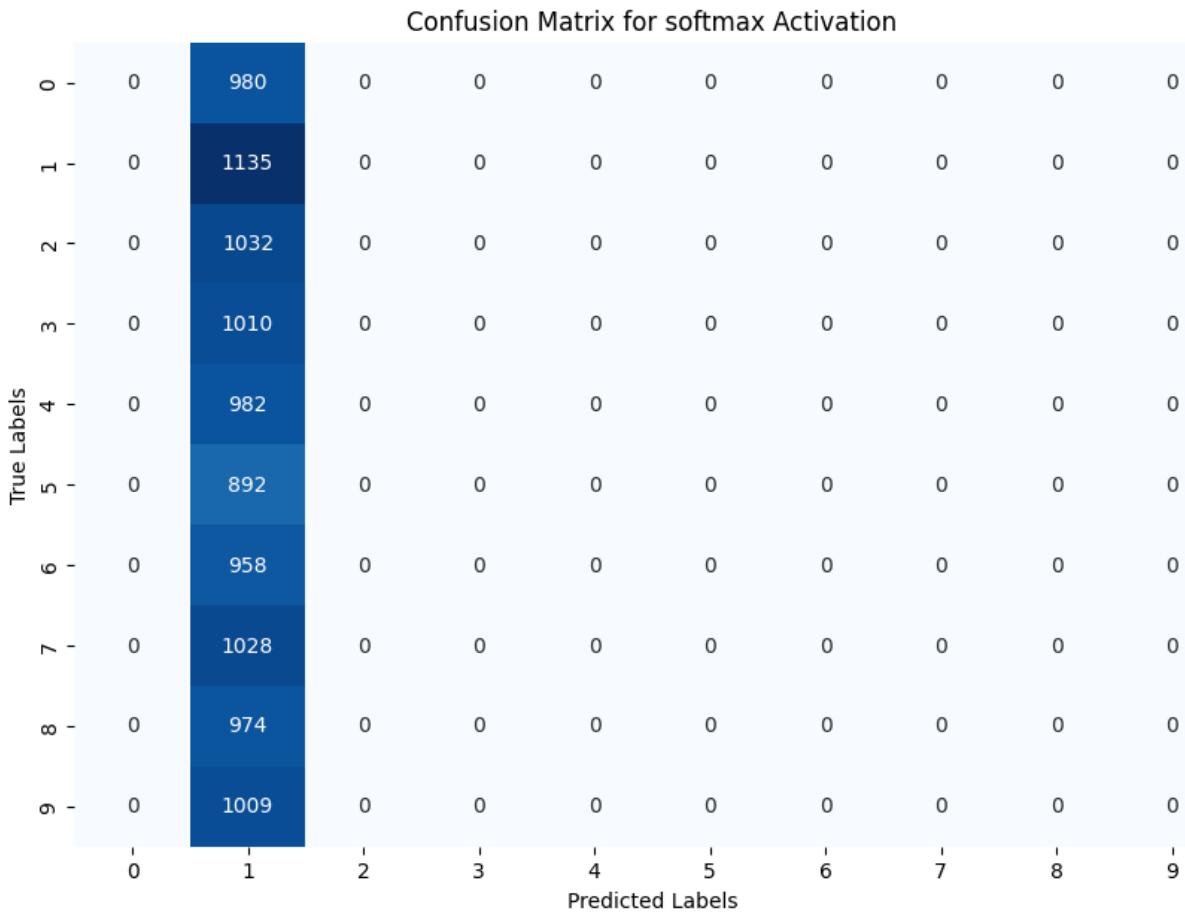
844/844 - 10s - loss: 2.3015 - accuracy: 0.1126 - val_loss: 2.3026 -
val_accuracy: 0.1050 - 10s/epoch - 12ms/step

Epoch 2/5

844/844 - 9s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3020 -

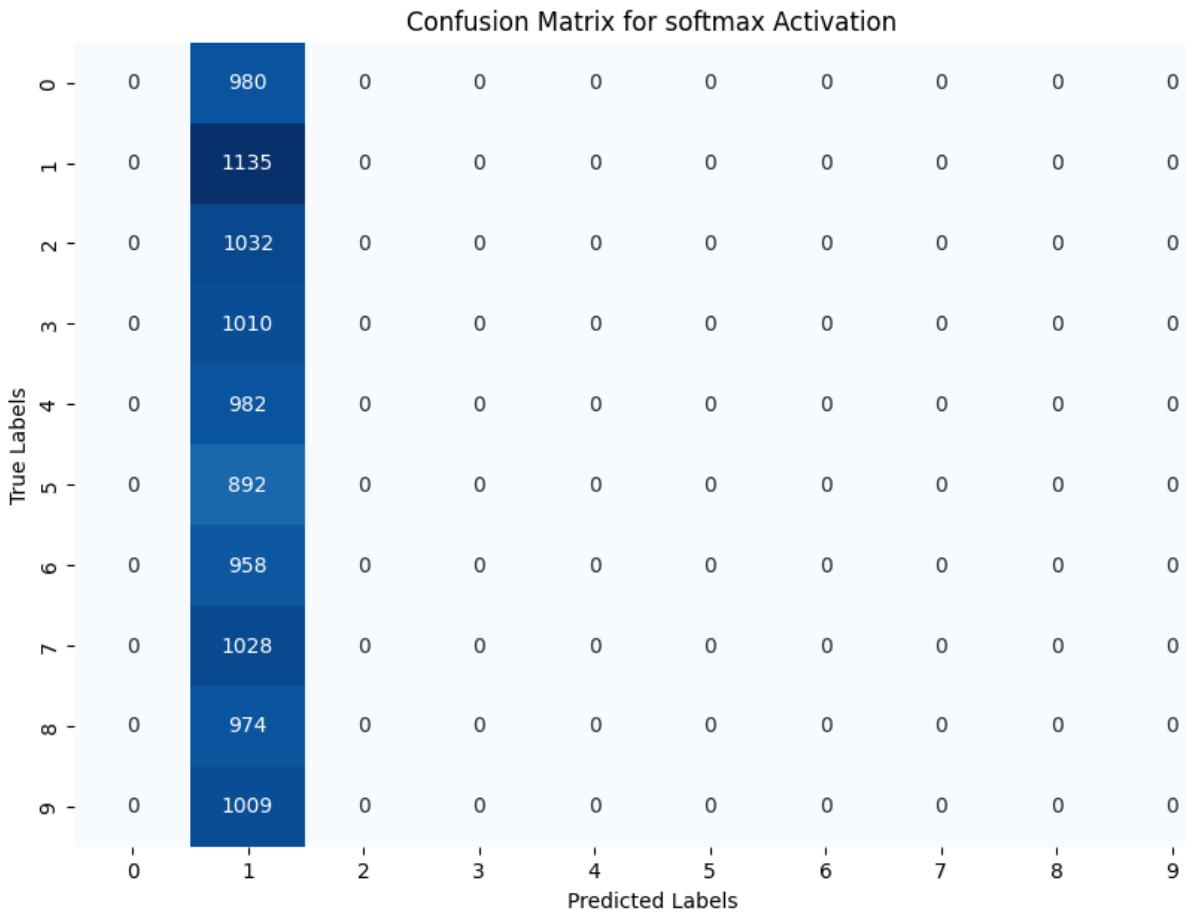
```
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 3/5
844/844 - 9s - loss: 2.3015 - accuracy: 0.1129 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 4/5
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3030 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 5/5
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
313/313 [=====] - 1s 4ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 15 epochs..
Epoch 1/15
844/844 - 10s - loss: 2.3017 - accuracy: 0.1128 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 10s/epoch - 12ms/step
Epoch 2/15
844/844 - 9s - loss: 2.3015 - accuracy: 0.1129 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 3/15
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 4/15
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 5/15
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 6/15
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 7/15
```

```
844/844 - 9s - loss: 2.3014 - accuracy: 0.1124 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 8/15  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 9/15  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 10/15  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 11/15  
844/844 - 9s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 12/15  
844/844 - 9s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 13/15  
844/844 - 9s - loss: 2.3014 - accuracy: 0.1130 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 14/15  
844/844 - 9s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 15/15  
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
313/313 [=====] - 1s 4ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
64 batch size, 20 epochs..
Epoch 1/20
844/844 - 11s - loss: 2.3017 - accuracy: 0.1127 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 11s/epoch - 13ms/step
Epoch 2/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 3/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 4/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3023 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 5/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1121 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 6/20
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -
val_accuracy: 0.1050 - 9s/epoch - 11ms/step
Epoch 7/20
```

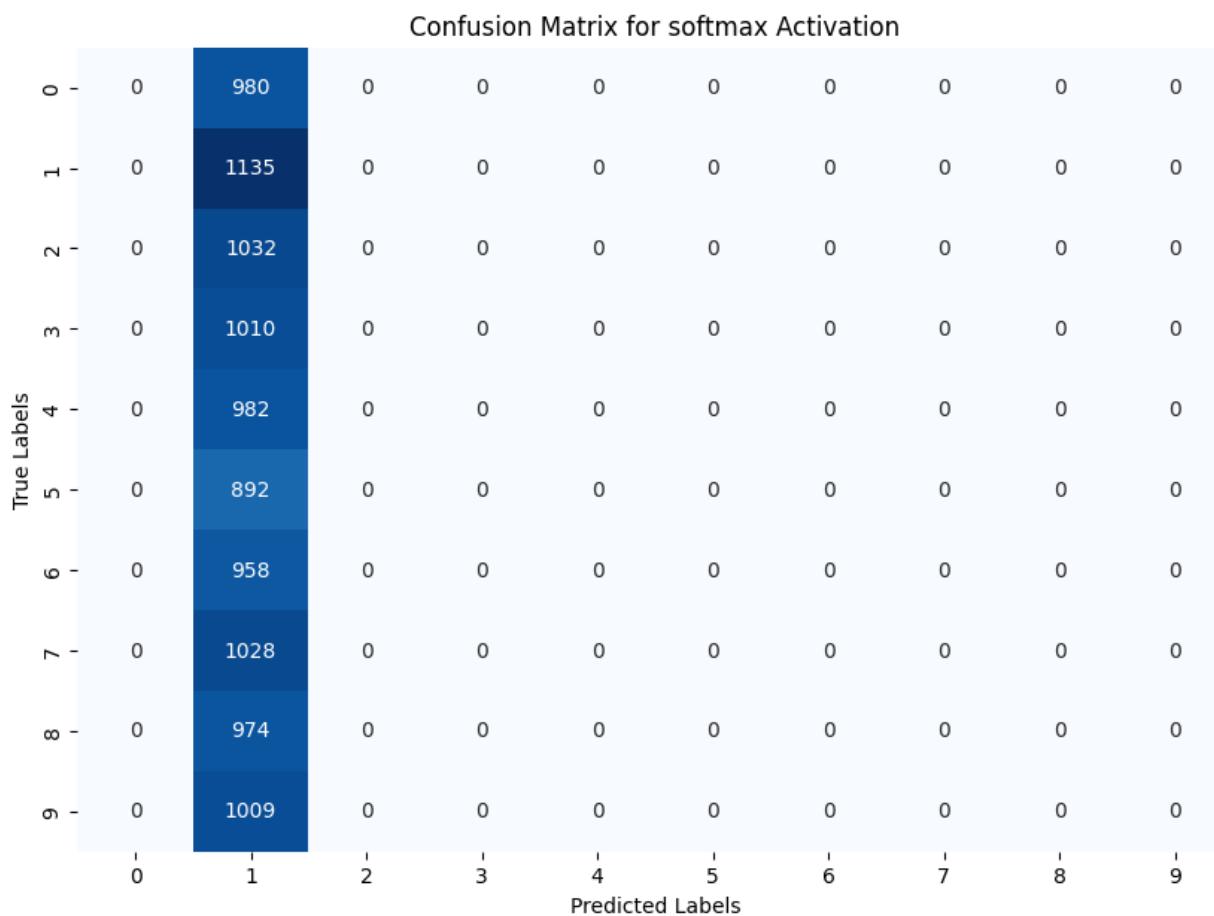
```
844/844 - 9s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 8/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 9/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 10/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 11/20  
844/844 - 9s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 12/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 13/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3015 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 14/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3027 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 15/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 16/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 17/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1127 - val_loss: 2.3026 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 18/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1128 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 19/20  
844/844 - 9s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
Epoch 20/20  
844/844 - 9s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 9s/epoch - 11ms/step  
313/313 [=====] - 1s 4ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

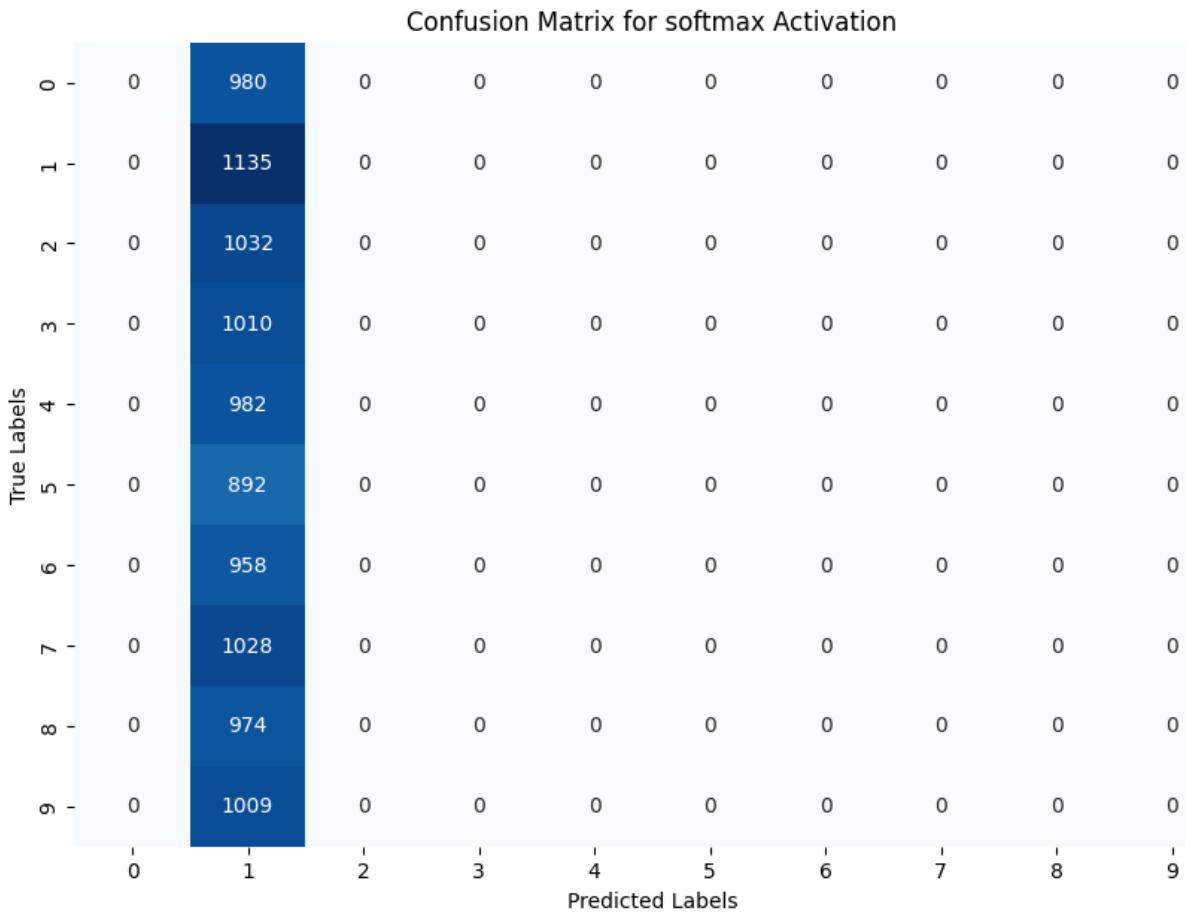
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 5 epochs..
Epoch 1/5
422/422 - 8s - loss: 2.3015 - accuracy: 0.1122 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 8s/epoch - 19ms/step
Epoch 2/5
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
```

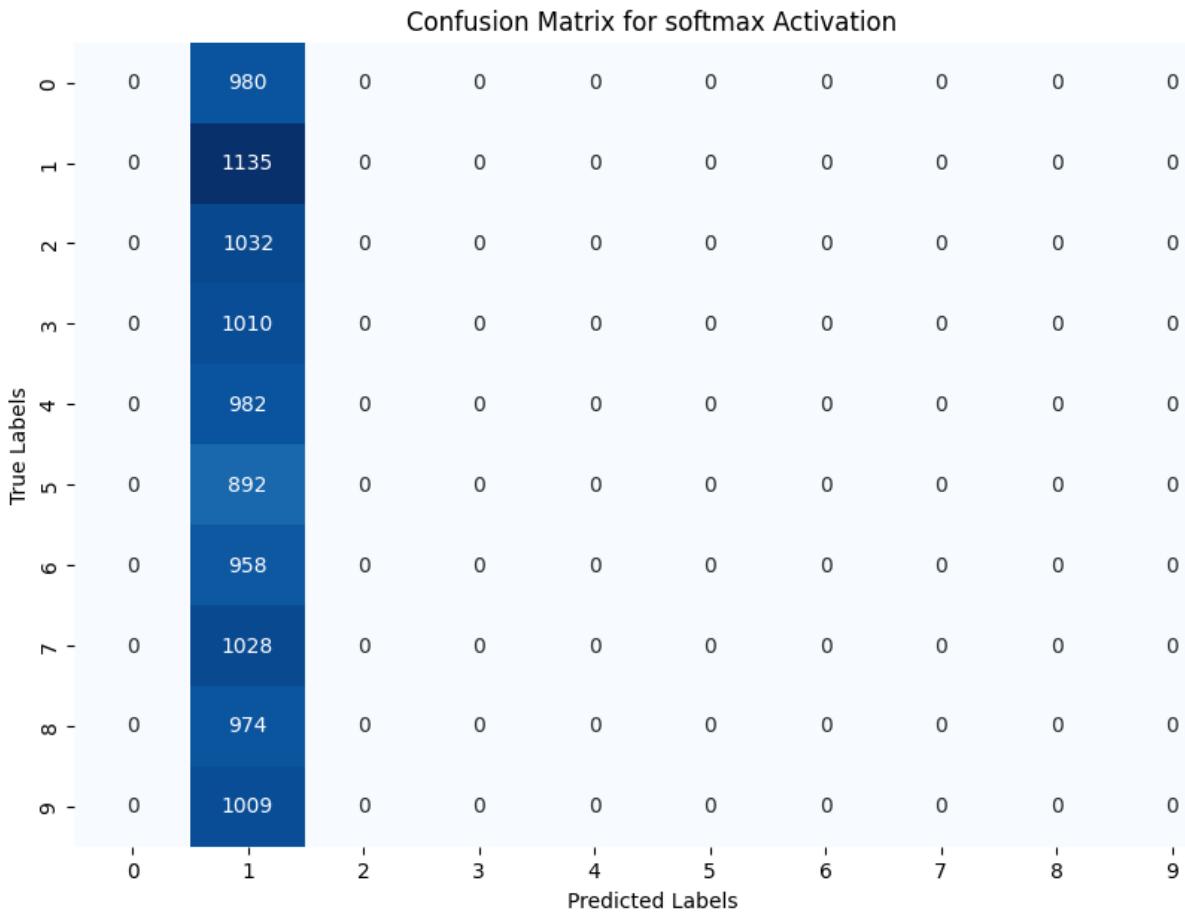
```
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 3/5
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 4/5
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 5/5
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
313/313 [=====] - 1s 4ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 15 epochs..
Epoch 1/15
422/422 - 8s - loss: 2.3016 - accuracy: 0.1119 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 8s/epoch - 18ms/step
Epoch 2/15
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 3/15
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 4/15
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 5/15
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 7s/epoch - 15ms/step
Epoch 6/15
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 7s/epoch - 16ms/step
Epoch 7/15
```

```
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 8/15  
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 7s/epoch - 16ms/step  
Epoch 9/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 10/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 11/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 12/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 13/15  
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 7s/epoch - 15ms/step  
Epoch 14/15  
422/422 - 7s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 7s/epoch - 15ms/step  
Epoch 15/15  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
313/313 [=====] - 1s 4ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```

Training Model with softmax activation, 3 conv_layers, 3 dense layers,
128 batch size, 20 epochs..
Epoch 1/20
422/422 - 7s - loss: 2.3015 - accuracy: 0.1126 - val_loss: 2.3017 -
val_accuracy: 0.1050 - 7s/epoch - 18ms/step
Epoch 2/20
422/422 - 6s - loss: 2.3014 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 3/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 4/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 5/20
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 15ms/step
Epoch 6/20
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -
val_accuracy: 0.1050 - 7s/epoch - 15ms/step
Epoch 7/20

```

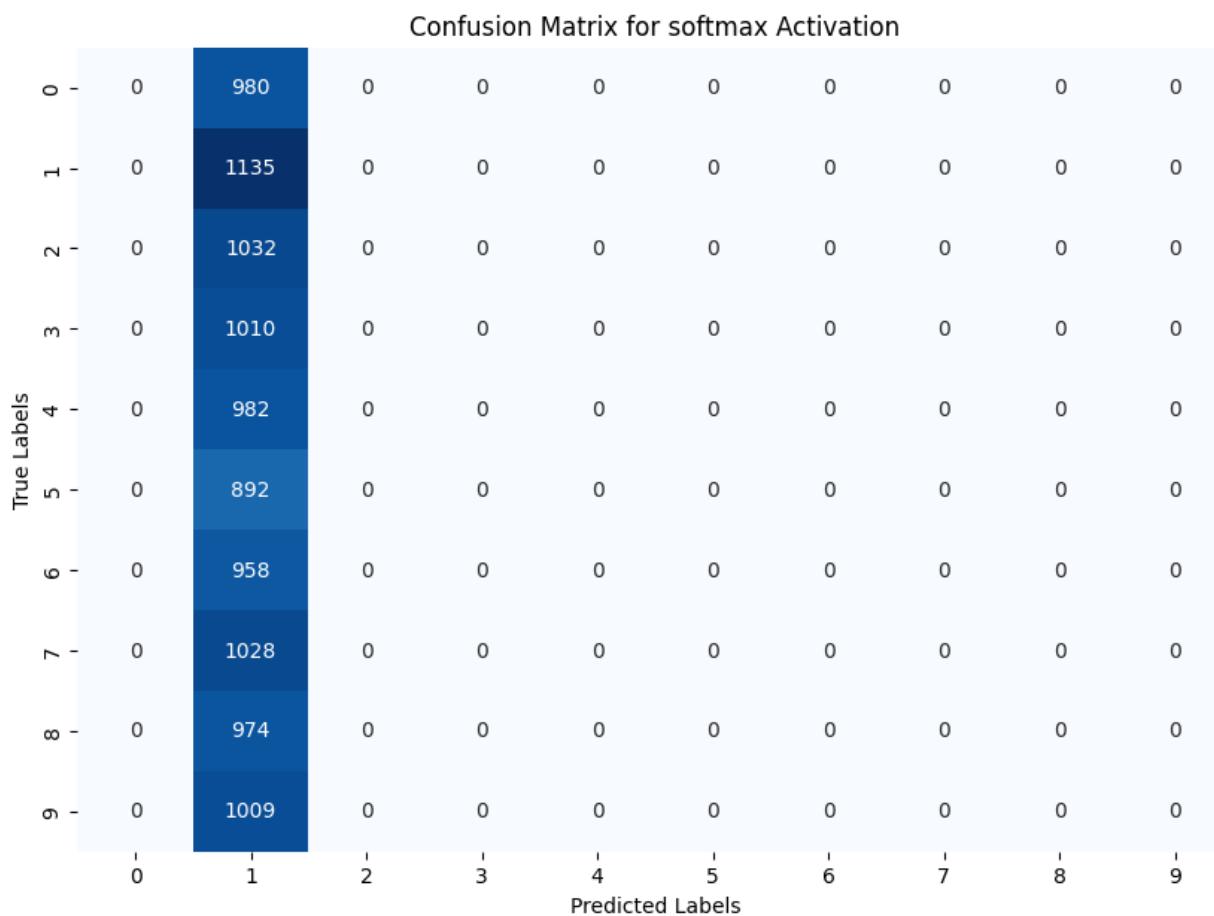
```
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3024 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 8/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 9/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 10/20  
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3023 -  
val_accuracy: 0.1050 - 7s/epoch - 16ms/step  
Epoch 11/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 12/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 13/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 14/20  
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 7s/epoch - 15ms/step  
Epoch 15/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 16/20  
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 7s/epoch - 15ms/step  
Epoch 17/20  
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3017 -  
val_accuracy: 0.1050 - 7s/epoch - 15ms/step  
Epoch 18/20  
422/422 - 7s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 7s/epoch - 16ms/step  
Epoch 19/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
Epoch 20/20  
422/422 - 6s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3025 -  
val_accuracy: 0.1050 - 6s/epoch - 15ms/step  
313/313 [=====] - 1s 4ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 5 epochs..
Epoch 1/5
211/211 - 6s - loss: 2.3016 - accuracy: 0.1114 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 29ms/step
Epoch 2/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
```

```
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 3/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 4/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
Epoch 5/5
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 24ms/step
313/313 [=====] - 1s 4ms/step
Results for activation function: softmax
Confusion Matrix:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0	980	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 15 epochs..
Epoch 1/15
211/211 - 6s - loss: 2.3015 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 6s/epoch - 29ms/step
Epoch 2/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 3/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 4/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 5/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 26ms/step
Epoch 6/15
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 7/15

```

```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 8/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 9/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 10/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 11/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 26ms/step  
Epoch 12/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 13/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 14/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3022 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 15/15  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 26ms/step  
313/313 [=====] - 1s 4ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0 0]  
[ 0 892 0 0 0 0 0 0 0 0]  
[ 0 958 0 0 0 0 0 0 0 0]  
[ 0 1028 0 0 0 0 0 0 0 0]  
[ 0 974 0 0 0 0 0 0 0 0]  
[ 0 1009 0 0 0 0 0 0 0 0]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation										
True Labels	0	1	2	3	4	5	6	7	8	9
	0	980	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0

```

Training Model with softmax activation, 3 conv_layers, 3 dense layers,
256 batch size, 20 epochs..
Epoch 1/20
211/211 - 6s - loss: 2.3016 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 6s/epoch - 29ms/step
Epoch 2/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 3/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 4/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 5/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 6/20
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -
val_accuracy: 0.1050 - 5s/epoch - 25ms/step
Epoch 7/20

```

```
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 26ms/step  
Epoch 8/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 9/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 10/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 11/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3021 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 12/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 13/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 14/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 15/20  
211/211 - 5s - loss: 2.3013 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 16/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 24ms/step  
Epoch 17/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 18/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3018 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 19/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3019 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
Epoch 20/20  
211/211 - 5s - loss: 2.3012 - accuracy: 0.1132 - val_loss: 2.3020 -  
val_accuracy: 0.1050 - 5s/epoch - 25ms/step  
313/313 [=====] - 1s 4ms/step  
Results for activation function: softmax  
Confusion Matrix:  
[[ 0 980 0 0 0 0 0 0 0 ]]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]
```

```
[ 0 892 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 ]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for softmax Activation

In this experiment, we combined various activation function choices with different network size, depth, epoch and batch sizes. We meticulously selected various activation functions such as relu, sigmoid, and softmax functions. Specifically, the experiment was designed to assess the model's behavior across an array of batch sizes—64, 128, 256—and epochs—5, 15, and 20.

In this experiment, softmax activation function generated the lowest precision and recall values despite the increment of the depth and width of the network. In this case, it showed the worst value when the batch size increased. Conversely, Relu activation function generated the highest

precision and recall score 0.9923 when the convolution and dense layers increase to (2,2) and (2,2). In this scenario, the batch size is 64 and the number of epochs is 20. On the other hand, sigmoid function generated consistently good precision and recall scores, but it generated lowest precision and recall scores when the number of layers and batch size increases.

Activation Function Variation: The choice of activation function significantly influences model performance, with ReLU consistently outperforming other activation functions across various batch sizes and epochs. This underscores ReLU's capability to mitigate the vanishing gradient problem, thereby facilitating better learning. Sigmoid functions exhibit competitive performance but may suffer from saturation issues in deeper networks or with inappropriate initialization, potentially leading to slower convergence or stagnation. Softmax functions in inner layers exhibit poorer performance, highlighting their unsuitability for internal activation due to exponential growth leading to instability or their inherent nature being more suited to output layers for probability distributions.

Impact of Batch Size and Epochs: Larger batch sizes generally lead to faster computation per epoch due to parallelization but may result in reduced model generalization capability. This is evident from the diminishing precision and recall scores as batch size increases, particularly noticeable in configurations using less robust activation functions like sigmoid. Increasing the number of epochs tends to improve model accuracy up to a certain point, beyond which gains diminish. This is indicative of the model's ability to learn more from the data over time, though it also raises the risk of overfitting if not coupled with appropriate regularization techniques.

Network Size and Depth: The experimentation indicates that increasing network depth and size can enhance model performance, particularly when utilizing activation functions like ReLU. This can be attributed to the ability of deeper networks to capture more complex patterns and relationships in the data. However, the benefits of increased size and depth are contingent upon adequate data and proper regularization to prevent overfitting. Moreover, computational costs and training time escalate with network size, emphasizing the need for a balanced approach.

###Experiment with Various Optimizers and Learning Rates for Different Batch Sizes and Epochs:

```
optimizers = ["adam", "nadam", "AdamW"]
learning_rate_num = [0.1, 0.01, 0.001]
from tensorflow.keras.optimizers import AdamW, Adam, RMSprop, Adagrad,
Nadam, Adadelta
! pip install tensorflow_addons
Collecting tensorflow_addons
  Downloading tensorflow_addons-0.23.0-cp310-cp310-
manylinux_2_17_x86_64.manylinux2014_x86_64.whl (611 kB)
                                           611.8/611.8 kB 3.6 MB/s eta
0:00:00
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-
packages (from tensorflow_addons) (24.0)
Collecting typeguard<3.0.0,>=2.7 (from tensorflow_addons)
  Downloading typeguard-2.13.3-py3-none-any.whl (17 kB)
```

```
Installing collected packages: typeguard, tensorflow_addons
Successfully installed tensorflow-addons-0.23.0 typeguard-2.13.3

import tensorflow_addons as tfa

/usr/local/lib/python3.10/dist-packages/tensorflow_addons/utils/
tfa_eol_msg.py:23: UserWarning:

TensorFlow Addons (TFA) has ended development and introduction of new
features.
TFA has entered a minimal maintenance and release mode until a planned
end of life in May 2024.
Please modify downstream libraries to take dependencies from other
repositories in our TensorFlow community (e.g. Keras, Keras-CV, and
Keras-NLP).

For more information see:
https://github.com/tensorflow/addons/issues/2807

    warnings.warn(
/usr/local/lib/python3.10/dist-packages/tensorflow_addons/utils/ensure
_tf_install.py:53: UserWarning: Tensorflow Addons supports using
Python ops for all Tensorflow versions above or equal to 2.13.0 and
strictly below 2.16.0 (nightly versions are not supported).
The versions of TensorFlow you are currently using is 2.12.0 and is
not supported.
Some things might work, some things might not.
If you were to encounter a bug, do not file an issue.
If you want to make sure you're using a tested and supported
configuration, either change the TensorFlow version or the TensorFlow
Addons's version.
You can find the compatibility matrix in TensorFlow Addon's readme:
https://github.com/tensorflow/addons
    warnings.warn(

def baseline_model_opt_lr_exp_2(optimizer_name = "adam", lr = 0.001):
    model = Sequential([
        Conv2D(32,(3,3), activation = "relu", kernel_initializer =
"he_uniform", input_shape = (28,28,1)),
        MaxPooling2D((2,2)),
        Flatten(),
        Dense(100, activation = "relu", kernel_initializer =
"he_uniform"),
        Dense(10, activation = "softmax")

    ])
    # Select the optimizer based on the optimizer_name parameter
    if optimizer_name.lower() == "adam":
        opt = Adam(learning_rate=lr)
    elif optimizer_name.lower() == "nadam":
```

```

        opt = Nadam(learning_rate=lr)
    elif optimizer_name.lower() == "adadelta":
        opt = Adadelta(learning_rate=lr)
    elif optimizer_name.lower() == "rmsprop":
        opt = RMSprop(learning_rate=lr, momentum=0.9)
    elif optimizer_name.lower() == "adamw":
        opt = tfa.optimizers.AdamW(learning_rate=lr, weight_decay =
0.004)
    else:
        raise ValueError("Unsupported Optimizer")
    ## Select the optimizer based on the optimizer name parameter

    model.compile(optimizer = opt, loss = "categorical_crossentropy",
metrics = ["accuracy"])
    return model

import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, precision_score,
recall_score
import numpy as np

for opt in optimizers:
    for learn_r in learning_rate_num:
        for batch in batch_sizes:
            for epoch in epochs_list:
                print(f"Training with {opt} optimizer and the
learning_rate is {learn_r}, {batch} batch size and {epoch} epochs...")
                model =
baseline_model_opt_lr_exp_2(optimizer_name=opt, lr=learn_r)

                history = model.fit(train_norm, y_train, epochs=epoch,
batch_size=batch, validation_split=0.1, verbose=2)

                ### Predictions
                y_pred_prob = model.predict(test_norm)
                y_pred = np.argmax(y_pred_prob, axis=1)

                ### Ensure y_test is not one-hot encoded for confusion
matrix
                if y_test.ndim > 1:
                    y_true = np.argmax(y_test, axis=1)
                else:
                    y_true = y_test

                ### Calculating the confusion matrix
                cm = confusion_matrix(y_true, y_pred)
                precision = precision_score(y_true, y_pred,
average="weighted")
                recall = recall_score(y_true, y_pred,

```

```

average="weighted")

    ### Printing the confusion matrix
    print(f"Confusion Matrix {opt} optimizer and the
learning_rate is {learn_r}, {batch} batch size and {epoch} epochs:")
    print(cm)
    print(f"Precision: {precision:.4f}")
    print(f"Recall: {recall:.4f}")

    ### Plotting the confusion matrix
    plt.figure(figsize=(10, 7))
    sns.heatmap(cm, annot=True, fmt="g", cmap="Blues",
cbar=False)
    plt.xlabel("Predicted Labels")
    plt.ylabel("True Labels")
    plt.title(f"Confusion Matrix for {opt} optimizer and
{learn_r} learning rate, {batch} batch size and {epoch} epochs")
    plt.show()

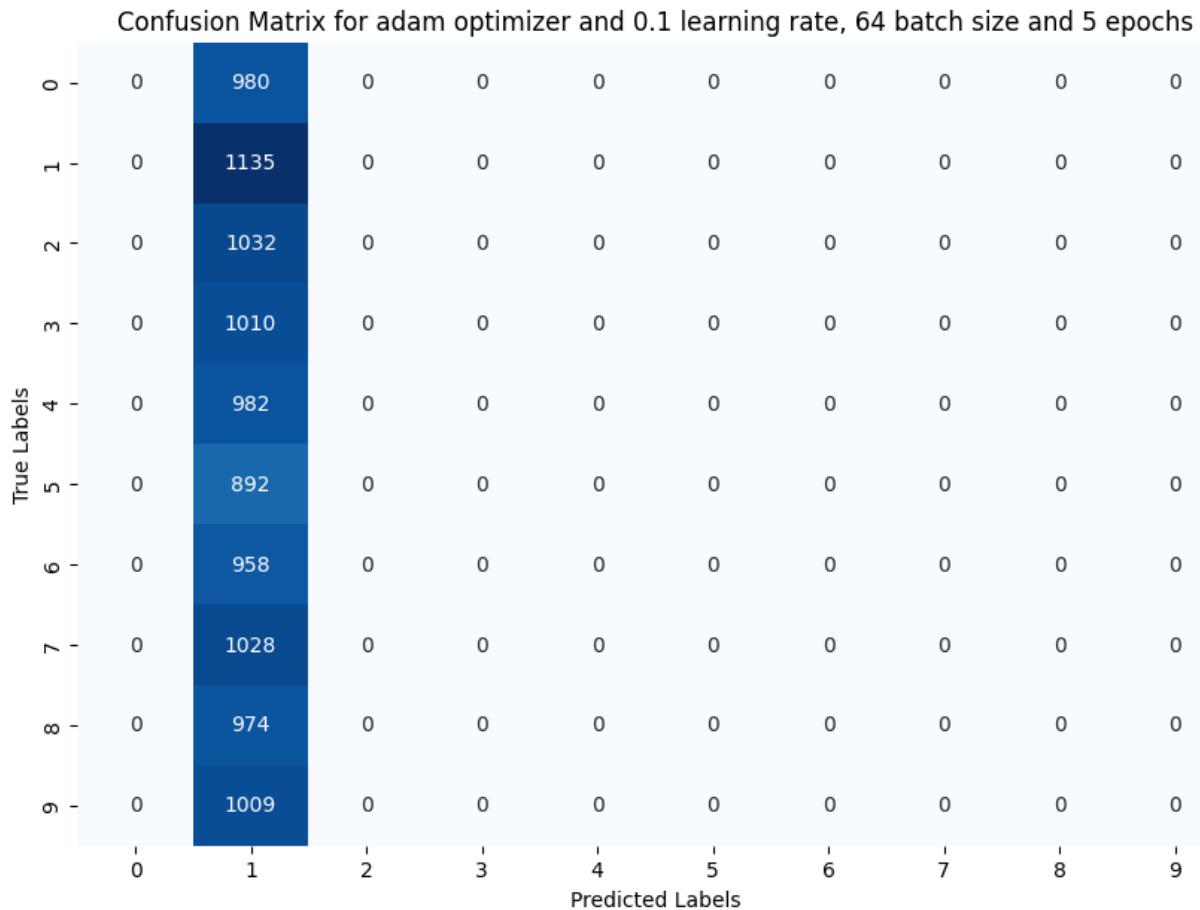
Training with adam optimizer and the learning_rate is 0.1, 64 batch
size and 5 epochs...
Epoch 1/5
844/844 - 6s - loss: 2.6098 - accuracy: 0.1053 - val_loss: 2.3093 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 2/5
844/844 - 5s - loss: 2.3098 - accuracy: 0.1029 - val_loss: 2.3083 -
val_accuracy: 0.1045 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 2.3091 - accuracy: 0.1033 - val_loss: 2.3117 -
val_accuracy: 0.0952 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 2.3102 - accuracy: 0.1032 - val_loss: 2.3112 -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 2.3095 - accuracy: 0.1047 - val_loss: 2.3077 -
val_accuracy: 0.1050 - 5s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.1, 64 batch
size and 5 epochs:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]

```

```
Precision: 0.0129
```

```
Recall: 0.1135
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training with adam optimizer and the learning_rate is 0.1, 64 batch size and 15 epochs...
```

```
Epoch 1/15
```

```
844/844 - 5s - loss: 2.6096 - accuracy: 0.1042 - val_loss: 2.3092 - val_accuracy: 0.0960 - 5s/epoch - 6ms/step
```

```
Epoch 2/15
```

```
844/844 - 5s - loss: 2.3109 - accuracy: 0.1044 - val_loss: 2.3067 - val_accuracy: 0.1113 - 5s/epoch - 6ms/step
```

```
Epoch 3/15
```

```
844/844 - 5s - loss: 2.3105 - accuracy: 0.1042 - val_loss: 2.3129 - val_accuracy: 0.1113 - 5s/epoch - 6ms/step
```

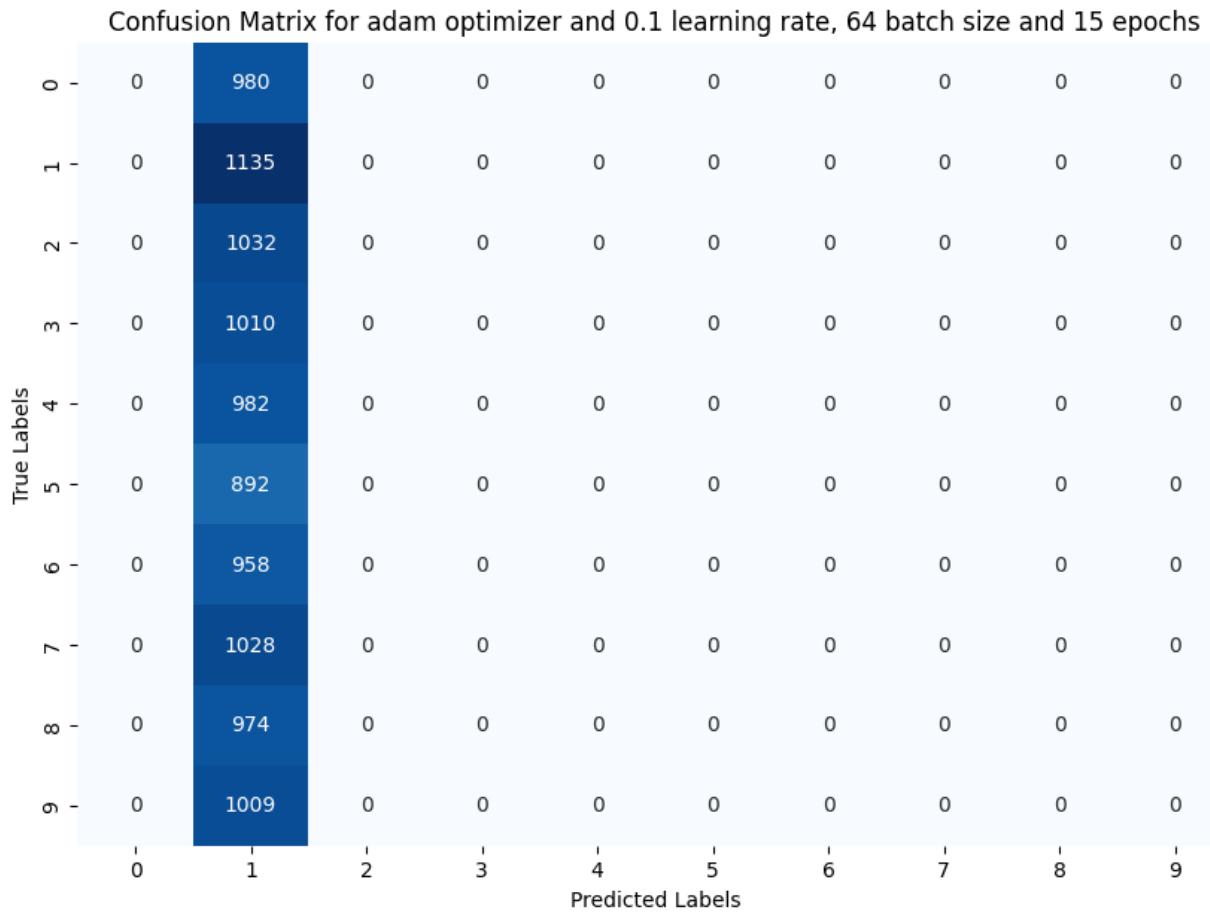
```
Epoch 4/15
```

```
844/844 - 5s - loss: 2.3102 - accuracy: 0.1044 - val_loss: 2.3036 -  
val_accuracy: 0.1113 - 5s/epoch - 6ms/step  
Epoch 5/15  
844/844 - 5s - loss: 2.3115 - accuracy: 0.1033 - val_loss: 2.3159 -  
val_accuracy: 0.0952 - 5s/epoch - 6ms/step  
Epoch 6/15  
844/844 - 5s - loss: 2.3100 - accuracy: 0.1011 - val_loss: 2.3166 -  
val_accuracy: 0.0960 - 5s/epoch - 6ms/step  
Epoch 7/15  
844/844 - 5s - loss: 2.3097 - accuracy: 0.1051 - val_loss: 2.3108 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 2.3092 - accuracy: 0.1047 - val_loss: 2.3211 -  
val_accuracy: 0.1000 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 2.3099 - accuracy: 0.1036 - val_loss: 2.3125 -  
val_accuracy: 0.1045 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 2.3098 - accuracy: 0.1048 - val_loss: 2.3069 -  
val_accuracy: 0.0960 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 2.3094 - accuracy: 0.1052 - val_loss: 2.3109 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 2.3102 - accuracy: 0.1052 - val_loss: 2.3073 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 2.3105 - accuracy: 0.1035 - val_loss: 2.3133 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 2.3105 - accuracy: 0.1060 - val_loss: 2.3173 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 2.3095 - accuracy: 0.1067 - val_loss: 2.3157 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.1, 64 batch  
size and 15 epochs:  
[[ 0 980 0 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 0 ]]
```

```
Precision: 0.0129
```

```
Recall: 0.1135
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```



```
Training with adam optimizer and the learning_rate is 0.1, 64 batch size and 20 epochs...
```

```
Epoch 1/20
```

```
844/844 - 6s - loss: 1.1618 - accuracy: 0.7997 - val_loss: 0.5143 - val_accuracy: 0.8407 - 6s/epoch - 7ms/step
```

```
Epoch 2/20
```

```
844/844 - 5s - loss: 0.5224 - accuracy: 0.8440 - val_loss: 0.4522 - val_accuracy: 0.8792 - 5s/epoch - 5ms/step
```

```
Epoch 3/20
```

```
844/844 - 5s - loss: 0.5174 - accuracy: 0.8471 - val_loss: 0.4786 - val_accuracy: 0.8565 - 5s/epoch - 6ms/step
```

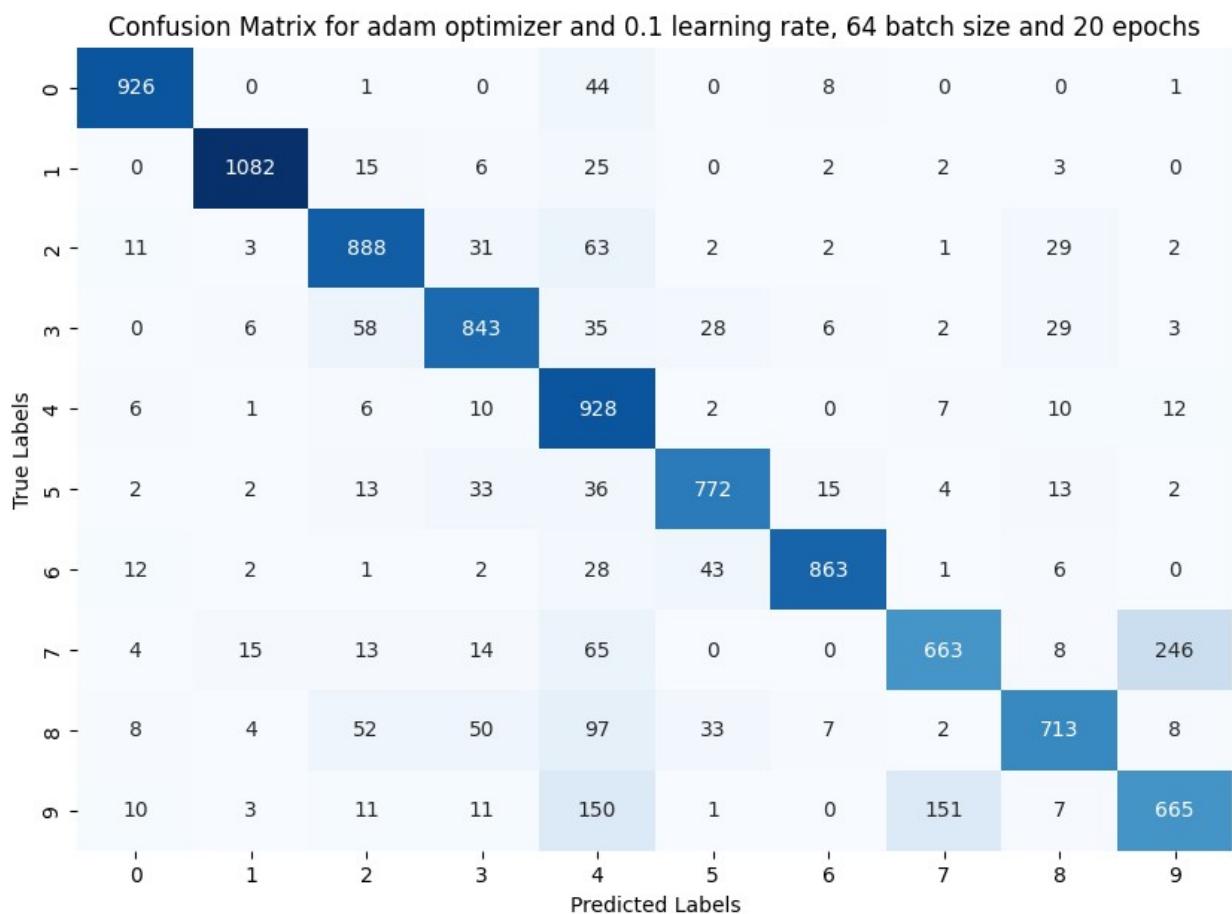
```
Epoch 4/20
```

```
844/844 - 5s - loss: 0.5077 - accuracy: 0.8512 - val_loss: 0.5543 -  
val_accuracy: 0.8173 - 5s/epoch - 6ms/step  
Epoch 5/20  
844/844 - 5s - loss: 0.5343 - accuracy: 0.8477 - val_loss: 0.5605 -  
val_accuracy: 0.8287 - 5s/epoch - 6ms/step  
Epoch 6/20  
844/844 - 5s - loss: 0.5594 - accuracy: 0.8437 - val_loss: 0.4571 -  
val_accuracy: 0.8573 - 5s/epoch - 6ms/step  
Epoch 7/20  
844/844 - 5s - loss: 0.5921 - accuracy: 0.8347 - val_loss: 0.4107 -  
val_accuracy: 0.8720 - 5s/epoch - 6ms/step  
Epoch 8/20  
844/844 - 5s - loss: 0.6086 - accuracy: 0.8342 - val_loss: 0.5584 -  
val_accuracy: 0.8463 - 5s/epoch - 6ms/step  
Epoch 9/20  
844/844 - 5s - loss: 0.6219 - accuracy: 0.8322 - val_loss: 0.5918 -  
val_accuracy: 0.8428 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: 0.6324 - accuracy: 0.8269 - val_loss: 0.5124 -  
val_accuracy: 0.8572 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: 0.6010 - accuracy: 0.8355 - val_loss: 0.7639 -  
val_accuracy: 0.8195 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 5s - loss: 0.6382 - accuracy: 0.8254 - val_loss: 0.6226 -  
val_accuracy: 0.8188 - 5s/epoch - 6ms/step  
Epoch 13/20  
844/844 - 5s - loss: 0.6900 - accuracy: 0.8161 - val_loss: 0.7269 -  
val_accuracy: 0.7895 - 5s/epoch - 6ms/step  
Epoch 14/20  
844/844 - 5s - loss: 0.6309 - accuracy: 0.8281 - val_loss: 0.5413 -  
val_accuracy: 0.8468 - 5s/epoch - 6ms/step  
Epoch 15/20  
844/844 - 5s - loss: 0.7375 - accuracy: 0.8092 - val_loss: 0.5212 -  
val_accuracy: 0.8595 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: 0.5708 - accuracy: 0.8427 - val_loss: 0.4878 -  
val_accuracy: 0.8650 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: 0.6783 - accuracy: 0.8181 - val_loss: 0.5991 -  
val_accuracy: 0.8220 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: 0.6468 - accuracy: 0.8278 - val_loss: 0.6948 -  
val_accuracy: 0.7973 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: 0.7004 - accuracy: 0.8146 - val_loss: 0.6676 -  
val_accuracy: 0.8322 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: 0.6107 - accuracy: 0.8376 - val_loss: 0.5394 -
```

```

val_accuracy: 0.8497 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.1, 64 batch
size and 20 epochs:
[[ 926   0   1   0   44   0   8   0   0   1]
 [  0 1082  15   6  25   0   2   2   3   0]
 [ 11   3 888  31  63   2   2   1  29   2]
 [  0   6  58 843  35  28   6   2  29   3]
 [  6   1   6  10 928   2   0   7  10  12]
 [  2   2  13  33  36 772  15   4  13   2]
 [ 12   2   1   2  28  43 863   1   6   0]
 [  4  15  13  14  65   0   0 663   8 246]
 [  8   4  52  50  97  33   7   2 713   8]
 [ 10   3  11  11 150   1   0 151   7 665]]
Precision: 0.8441
Recall: 0.8343

```



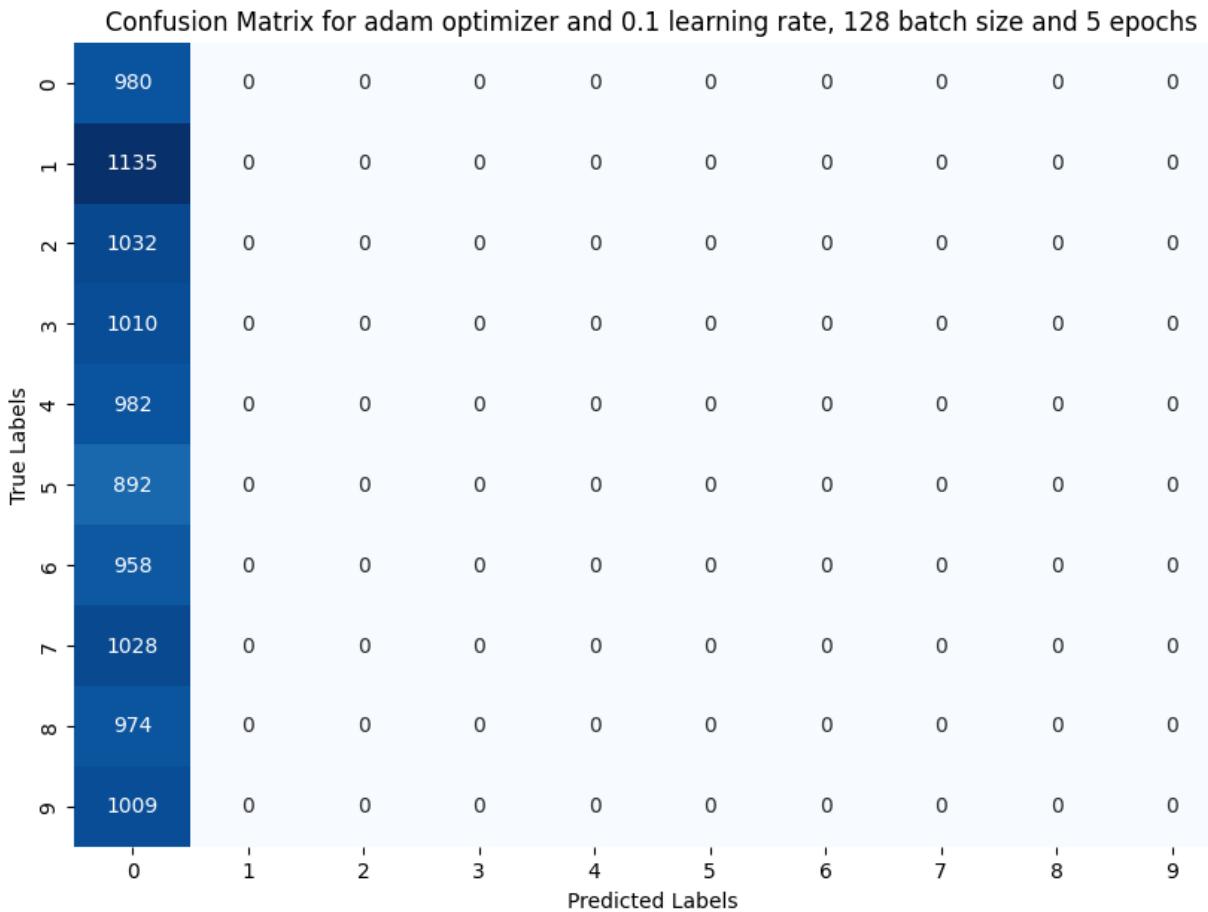
```

Training with adam optimizer and the learning_rate is 0.1, 128 batch
size and 5 epochs...
Epoch 1/5
422/422 - 4s - loss: 2.8364 - accuracy: 0.1063 - val_loss: 2.3120 -

```

```
val_accuracy: 0.1113 - 4s/epoch - 10ms/step
Epoch 2/5
422/422 - 3s - loss: 2.3066 - accuracy: 0.1063 - val_loss: 2.3127 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 3/5
422/422 - 3s - loss: 2.3071 - accuracy: 0.1056 - val_loss: 2.3114 -
val_accuracy: 0.1113 - 3s/epoch - 8ms/step
Epoch 4/5
422/422 - 3s - loss: 2.3066 - accuracy: 0.1073 - val_loss: 2.3107 -
val_accuracy: 0.1045 - 3s/epoch - 8ms/step
Epoch 5/5
422/422 - 3s - loss: 2.3081 - accuracy: 0.1056 - val_loss: 2.3111 -
val_accuracy: 0.0978 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.1, 128
batch size and 5 epochs:
[[ 980    0    0    0    0    0    0    0    0    0]
 [1135    0    0    0    0    0    0    0    0    0]
 [1032    0    0    0    0    0    0    0    0    0]
 [1010    0    0    0    0    0    0    0    0    0]
 [ 982    0    0    0    0    0    0    0    0    0]
 [ 892    0    0    0    0    0    0    0    0    0]
 [ 958    0    0    0    0    0    0    0    0    0]
 [1028    0    0    0    0    0    0    0    0    0]
 [ 974    0    0    0    0    0    0    0    0    0]
 [1009    0    0    0    0    0    0    0    0    0]]
Precision: 0.0096
Recall: 0.0980

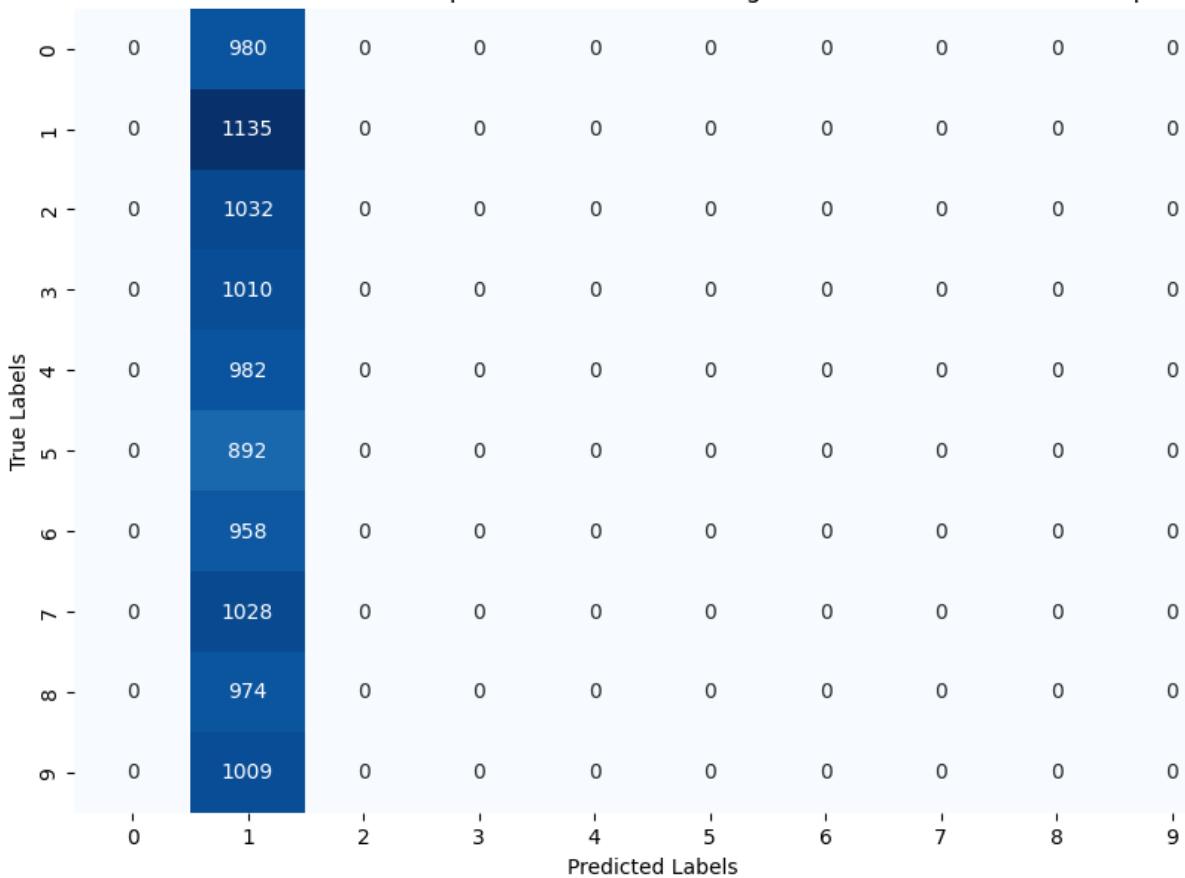
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



```
Training with adam optimizer and the learning_rate is 0.1, 128 batch size and 15 epochs...
Epoch 1/15
422/422 - 4s - loss: 3.3304 - accuracy: 0.1052 - val_loss: 2.3037 -
val_accuracy: 0.0952 - 4s/epoch - 10ms/step
Epoch 2/15
422/422 - 3s - loss: 2.3071 - accuracy: 0.1042 - val_loss: 2.3084 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 3/15
422/422 - 3s - loss: 2.3068 - accuracy: 0.1064 - val_loss: 2.3069 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 4/15
422/422 - 3s - loss: 2.3070 - accuracy: 0.1049 - val_loss: 2.3056 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 5/15
422/422 - 3s - loss: 2.3070 - accuracy: 0.1055 - val_loss: 2.3064 -
val_accuracy: 0.0978 - 3s/epoch - 8ms/step
Epoch 6/15
422/422 - 3s - loss: 2.3074 - accuracy: 0.1031 - val_loss: 2.3040 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 7/15
```

```
422/422 - 3s - loss: 2.3062 - accuracy: 0.1044 - val_loss: 2.3090 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 8/15  
422/422 - 3s - loss: 2.3080 - accuracy: 0.1051 - val_loss: 2.3069 -  
val_accuracy: 0.1045 - 3s/epoch - 8ms/step  
Epoch 9/15  
422/422 - 3s - loss: 2.3075 - accuracy: 0.1055 - val_loss: 2.3104 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 10/15  
422/422 - 3s - loss: 2.3067 - accuracy: 0.1070 - val_loss: 2.3054 -  
val_accuracy: 0.0952 - 3s/epoch - 8ms/step  
Epoch 11/15  
422/422 - 3s - loss: 2.3071 - accuracy: 0.1059 - val_loss: 2.3088 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 12/15  
422/422 - 3s - loss: 2.3073 - accuracy: 0.1051 - val_loss: 2.3083 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 13/15  
422/422 - 3s - loss: 2.3074 - accuracy: 0.1056 - val_loss: 2.3072 -  
val_accuracy: 0.1113 - 3s/epoch - 7ms/step  
Epoch 14/15  
422/422 - 3s - loss: 2.3073 - accuracy: 0.1033 - val_loss: 2.3116 -  
val_accuracy: 0.1050 - 3s/epoch - 7ms/step  
Epoch 15/15  
422/422 - 3s - loss: 2.3063 - accuracy: 0.1069 - val_loss: 2.3077 -  
val_accuracy: 0.1050 - 3s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.1, 128  
batch size and 15 epochs:  
[[ 0 980 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adam optimizer and 0.1 learning rate, 128 batch size and 15 epochs



```
Training with adam optimizer and the learning_rate is 0.1, 128 batch size and 20 epochs...
Epoch 1/20
422/422 - 4s - loss: 2.9825 - accuracy: 0.1081 - val_loss: 2.3063 - val_accuracy: 0.0995 - 4s/epoch - 9ms/step
Epoch 2/20
422/422 - 3s - loss: 2.3064 - accuracy: 0.1037 - val_loss: 2.3081 - val_accuracy: 0.0995 - 3s/epoch - 7ms/step
Epoch 3/20
422/422 - 3s - loss: 2.3065 - accuracy: 0.1061 - val_loss: 2.3073 - val_accuracy: 0.1050 - 3s/epoch - 7ms/step
Epoch 4/20
422/422 - 3s - loss: 2.3072 - accuracy: 0.1066 - val_loss: 2.3041 - val_accuracy: 0.0995 - 3s/epoch - 8ms/step
Epoch 5/20
422/422 - 3s - loss: 2.3077 - accuracy: 0.1029 - val_loss: 2.3034 - val_accuracy: 0.1045 - 3s/epoch - 8ms/step
Epoch 6/20
422/422 - 3s - loss: 2.3066 - accuracy: 0.1056 - val_loss: 2.3087 - val_accuracy: 0.0978 - 3s/epoch - 8ms/step
Epoch 7/20
```

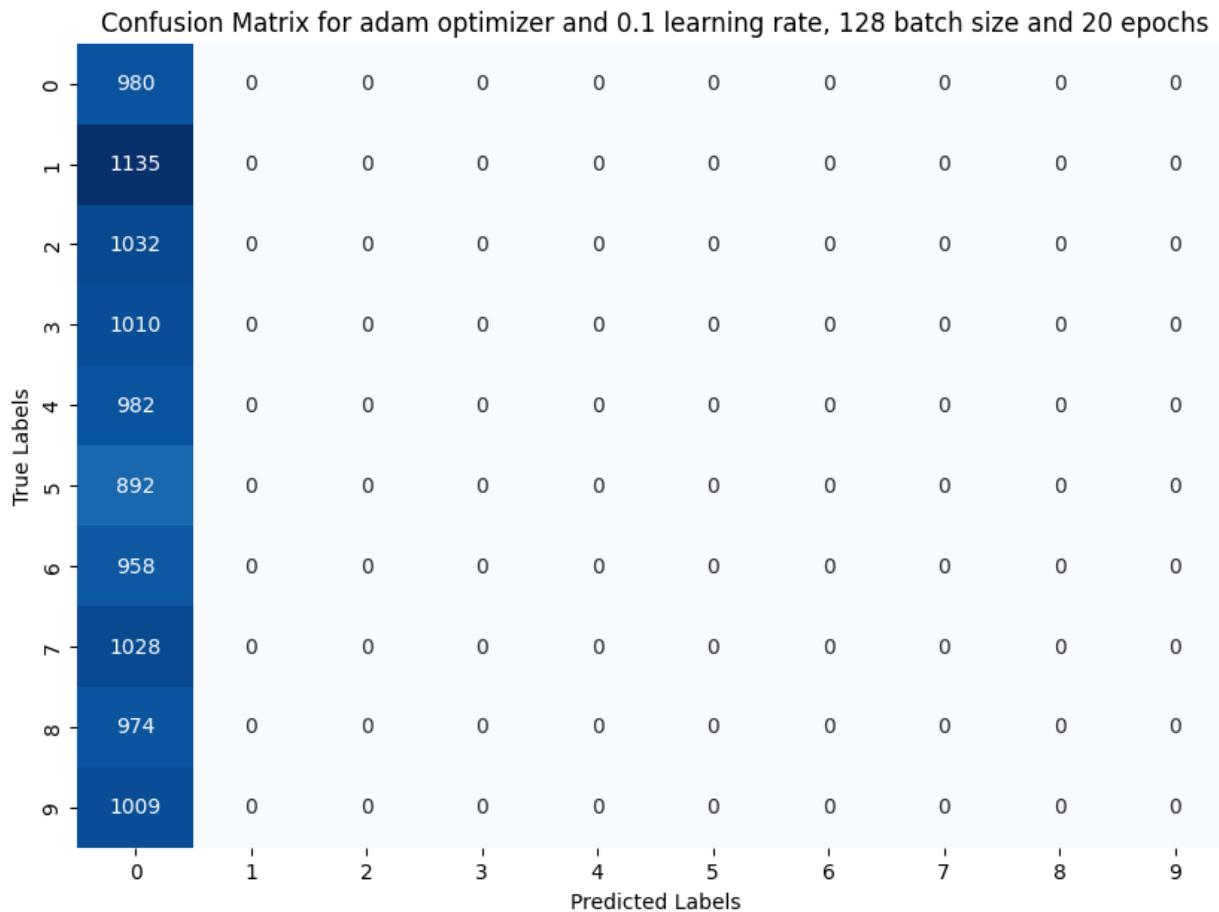
```
422/422 - 3s - loss: 2.3076 - accuracy: 0.1056 - val_loss: 2.3104 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 8/20  
422/422 - 3s - loss: 2.3074 - accuracy: 0.1050 - val_loss: 2.3106 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 9/20  
422/422 - 3s - loss: 2.3077 - accuracy: 0.1036 - val_loss: 2.3158 -  
val_accuracy: 0.1050 - 3s/epoch - 7ms/step  
Epoch 10/20  
422/422 - 3s - loss: 2.3071 - accuracy: 0.1045 - val_loss: 2.3061 -  
val_accuracy: 0.1113 - 3s/epoch - 7ms/step  
Epoch 11/20  
422/422 - 3s - loss: 2.3076 - accuracy: 0.1044 - val_loss: 2.3109 -  
val_accuracy: 0.1050 - 3s/epoch - 7ms/step  
Epoch 12/20  
422/422 - 3s - loss: 2.3072 - accuracy: 0.1057 - val_loss: 2.3027 -  
val_accuracy: 0.1113 - 3s/epoch - 7ms/step  
Epoch 13/20  
422/422 - 3s - loss: 2.3071 - accuracy: 0.1037 - val_loss: 2.3144 -  
val_accuracy: 0.1050 - 3s/epoch - 7ms/step  
Epoch 14/20  
422/422 - 3s - loss: 2.3071 - accuracy: 0.1045 - val_loss: 2.3098 -  
val_accuracy: 0.1113 - 3s/epoch - 7ms/step  
Epoch 15/20  
422/422 - 3s - loss: 2.3068 - accuracy: 0.1040 - val_loss: 2.3088 -  
val_accuracy: 0.0952 - 3s/epoch - 8ms/step  
Epoch 16/20  
422/422 - 3s - loss: 2.3069 - accuracy: 0.1050 - val_loss: 2.3097 -  
val_accuracy: 0.0952 - 3s/epoch - 7ms/step  
Epoch 17/20  
422/422 - 3s - loss: 2.3078 - accuracy: 0.1043 - val_loss: 2.3096 -  
val_accuracy: 0.1045 - 3s/epoch - 7ms/step  
Epoch 18/20  
422/422 - 3s - loss: 2.3075 - accuracy: 0.1051 - val_loss: 2.3062 -  
val_accuracy: 0.0952 - 3s/epoch - 7ms/step  
Epoch 19/20  
422/422 - 3s - loss: 2.3076 - accuracy: 0.1048 - val_loss: 2.3078 -  
val_accuracy: 0.1050 - 3s/epoch - 7ms/step  
Epoch 20/20  
422/422 - 3s - loss: 2.3076 - accuracy: 0.1016 - val_loss: 2.3125 -  
val_accuracy: 0.0978 - 3s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.1, 128  
batch size and 20 epochs:  
[[ 980  0  0  0  0  0  0  0  0 ]  
 [1135  0  0  0  0  0  0  0  0 ]  
 [1032  0  0  0  0  0  0  0  0 ]  
 [1010  0  0  0  0  0  0  0  0 ]  
 [ 982  0  0  0  0  0  0  0  0 ]
```

```
[ 892  0  0  0  0  0  0  0  0  0]  
[ 958  0  0  0  0  0  0  0  0  0]  
[1028  0  0  0  0  0  0  0  0  0]  
[ 974  0  0  0  0  0  0  0  0  0]  
[1009  0  0  0  0  0  0  0  0  0]]
```

Precision: 0.0096

Recall: 0.0980

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))
```



Training with adam optimizer and the learning_rate is 0.1, 256 batch size and 5 epochs...

Epoch 1/5

211/211 - 3s - loss: 4.1260 - accuracy: 0.1099 - val_loss: 2.3038 - val_accuracy: 0.1113 - 3s/epoch - 16ms/step

Epoch 2/5

211/211 - 2s - loss: 2.3047 - accuracy: 0.1071 - val_loss: 2.3031 -

```
val_accuracy: 0.1050 - 2s/epoch - 12ms/step
Epoch 3/5
211/211 - 2s - loss: 2.3053 - accuracy: 0.1053 - val_loss: 2.3079 -
val_accuracy: 0.1050 - 2s/epoch - 11ms/step
Epoch 4/5
211/211 - 2s - loss: 2.3050 - accuracy: 0.1070 - val_loss: 2.3055 -
val_accuracy: 0.1045 - 2s/epoch - 11ms/step
Epoch 5/5
211/211 - 2s - loss: 2.3052 - accuracy: 0.1060 - val_loss: 2.3054 -
val_accuracy: 0.1000 - 2s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.1, 256
batch size and 5 epochs:
[[ 0  0 980  0  0  0  0  0  0  0]
 [ 0  0 1135  0  0  0  0  0  0  0]
 [ 0  0 1032  0  0  0  0  0  0  0]
 [ 0  0 1010  0  0  0  0  0  0  0]
 [ 0  0 982  0  0  0  0  0  0  0]
 [ 0  0 892  0  0  0  0  0  0  0]
 [ 0  0 958  0  0  0  0  0  0  0]
 [ 0  0 1028  0  0  0  0  0  0  0]
 [ 0  0 974  0  0  0  0  0  0  0]
 [ 0  0 1009  0  0  0  0  0  0  0]]]
Precision: 0.0107
Recall: 0.1032

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adam optimizer and 0.1 learning rate, 256 batch size and 5 epochs

	0 -	1 -	2 -	3 -	4 -	5 -	6 -	7 -	8 -	9 -	
True Labels	0	0	0	0	0	0	0	0	0	0	0
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
0 -	980	0	0	0	0	0	0	0	0	0	0
1 -	0	0	1135	0	0	0	0	0	0	0	0
2 -	0	0	1032	0	0	0	0	0	0	0	0
3 -	0	0	1010	0	0	0	0	0	0	0	0
4 -	0	0	982	0	0	0	0	0	0	0	0
5 -	0	0	892	0	0	0	0	0	0	0	0
6 -	0	0	958	0	0	0	0	0	0	0	0
7 -	0	0	1028	0	0	0	0	0	0	0	0
8 -	0	0	974	0	0	0	0	0	0	0	0
9 -	0	0	1009	0	0	0	0	0	0	0	0

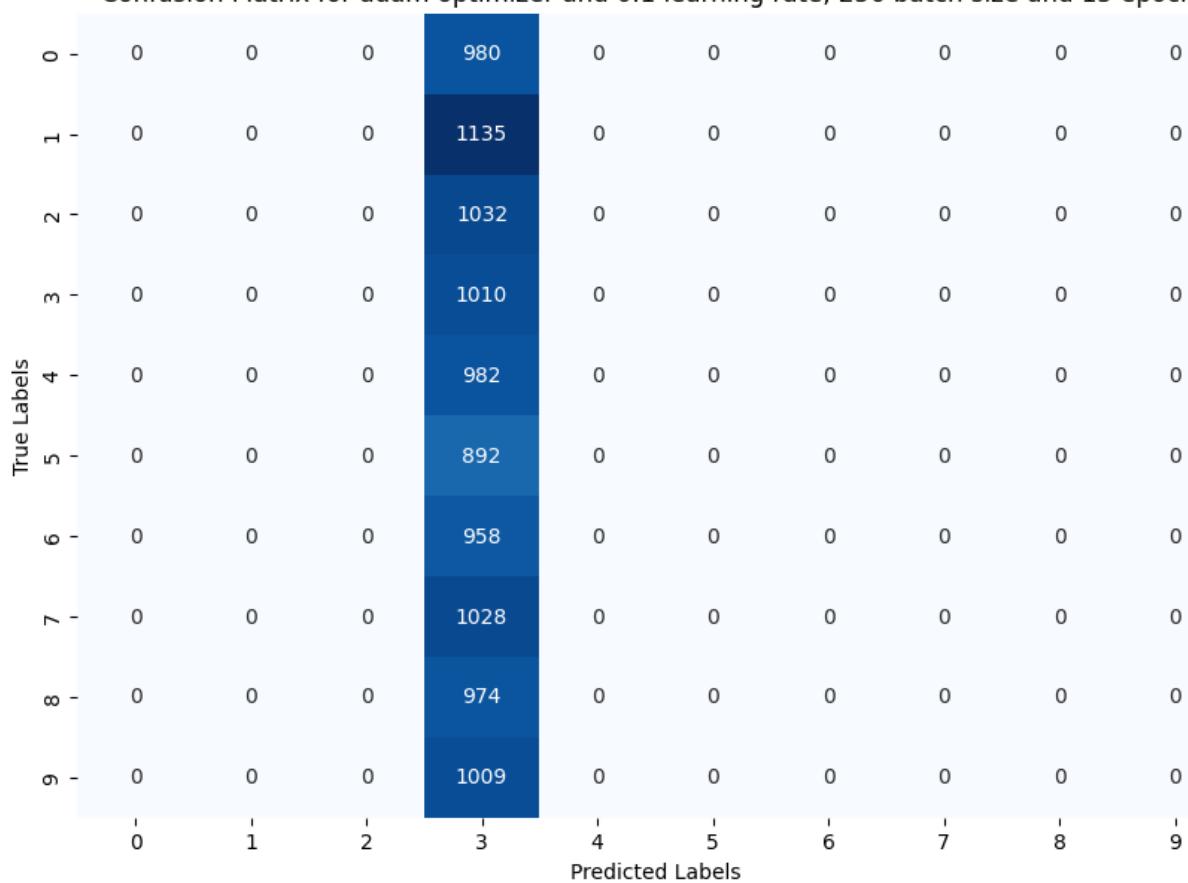
```
Training with adam optimizer and the learning_rate is 0.1, 256 batch size and 15 epochs...
Epoch 1/15
211/211 - 3s - loss: 3.4366 - accuracy: 0.1126 - val_loss: 2.3037 -
val_accuracy: 0.1045 - 3s/epoch - 16ms/step
Epoch 2/15
211/211 - 2s - loss: 2.3048 - accuracy: 0.1099 - val_loss: 2.3037 -
val_accuracy: 0.1050 - 2s/epoch - 12ms/step
Epoch 3/15
211/211 - 3s - loss: 2.3049 - accuracy: 0.1056 - val_loss: 2.3065 -
val_accuracy: 0.1050 - 3s/epoch - 12ms/step
Epoch 4/15
211/211 - 3s - loss: 2.3050 - accuracy: 0.1067 - val_loss: 2.3062 -
val_accuracy: 0.1050 - 3s/epoch - 12ms/step
Epoch 5/15
211/211 - 2s - loss: 2.3059 - accuracy: 0.1062 - val_loss: 2.3047 -
val_accuracy: 0.1113 - 2s/epoch - 12ms/step
Epoch 6/15
211/211 - 3s - loss: 2.3054 - accuracy: 0.1050 - val_loss: 2.3036 -
val_accuracy: 0.1045 - 3s/epoch - 12ms/step
Epoch 7/15
```

```
211/211 - 3s - loss: 2.3043 - accuracy: 0.1062 - val_loss: 2.3056 -  
val_accuracy: 0.0960 - 3s/epoch - 12ms/step  
Epoch 8/15  
211/211 - 3s - loss: 2.3054 - accuracy: 0.1067 - val_loss: 2.3063 -  
val_accuracy: 0.1050 - 3s/epoch - 12ms/step  
Epoch 9/15  
211/211 - 3s - loss: 2.3050 - accuracy: 0.1068 - val_loss: 2.3059 -  
val_accuracy: 0.1050 - 3s/epoch - 12ms/step  
Epoch 10/15  
211/211 - 3s - loss: 2.3056 - accuracy: 0.1081 - val_loss: 2.3071 -  
val_accuracy: 0.1045 - 3s/epoch - 12ms/step  
Epoch 11/15  
211/211 - 3s - loss: 2.3053 - accuracy: 0.1054 - val_loss: 2.3073 -  
val_accuracy: 0.1045 - 3s/epoch - 12ms/step  
Epoch 12/15  
211/211 - 2s - loss: 2.3051 - accuracy: 0.1053 - val_loss: 2.3047 -  
val_accuracy: 0.1050 - 2s/epoch - 12ms/step  
Epoch 13/15  
211/211 - 2s - loss: 2.3072 - accuracy: 0.1051 - val_loss: 2.3068 -  
val_accuracy: 0.1050 - 2s/epoch - 12ms/step  
Epoch 14/15  
211/211 - 3s - loss: 2.3053 - accuracy: 0.1066 - val_loss: 2.3065 -  
val_accuracy: 0.1000 - 3s/epoch - 12ms/step  
Epoch 15/15  
211/211 - 2s - loss: 2.3054 - accuracy: 0.1067 - val_loss: 2.3055 -  
val_accuracy: 0.1045 - 2s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.1, 256  
batch size and 15 epochs:  
[[ 0  0  0 980  0  0  0  0  0  0]  
[ 0  0  0 1135  0  0  0  0  0  0]  
[ 0  0  0 1032  0  0  0  0  0  0]  
[ 0  0  0 1010  0  0  0  0  0  0]  
[ 0  0  0 982  0  0  0  0  0  0]  
[ 0  0  0 892  0  0  0  0  0  0]  
[ 0  0  0 958  0  0  0  0  0  0]  
[ 0  0  0 1028  0  0  0  0  0  0]  
[ 0  0  0 974  0  0  0  0  0  0]  
[ 0  0  0 1009  0  0  0  0  0  0]]  
Precision: 0.0102  
Recall: 0.1010  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for adam optimizer and 0.1 learning rate, 256 batch size and 15 epochs



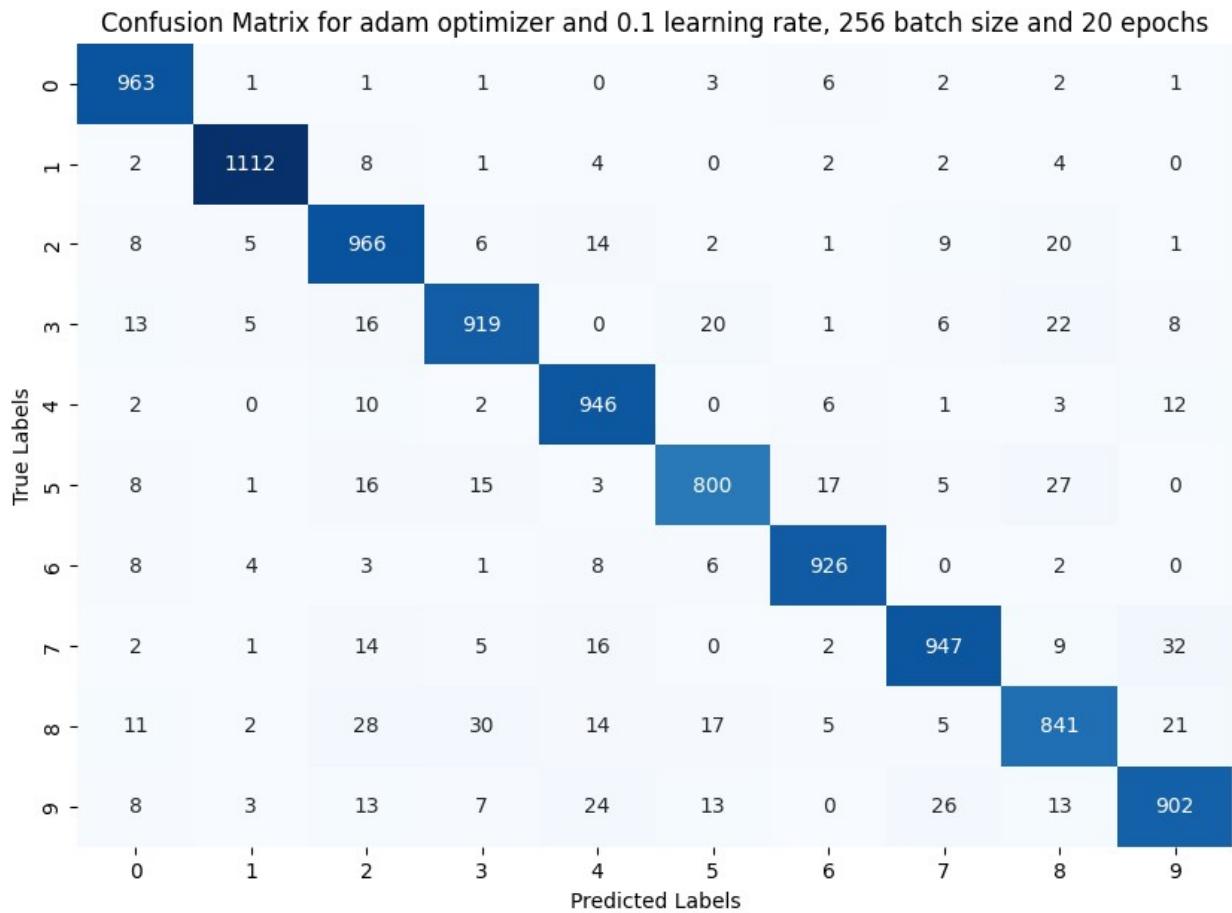
```
Training with adam optimizer and the learning_rate is 0.1, 256 batch size and 20 epochs...
Epoch 1/20
211/211 - 3s - loss: 1.4324 - accuracy: 0.8555 - val_loss: 0.2478 - val_accuracy: 0.9288 - 3s/epoch - 16ms/step
Epoch 2/20
211/211 - 2s - loss: 0.3450 - accuracy: 0.8997 - val_loss: 0.3027 - val_accuracy: 0.9208 - 2s/epoch - 12ms/step
Epoch 3/20
211/211 - 3s - loss: 0.2573 - accuracy: 0.9224 - val_loss: 0.2496 - val_accuracy: 0.9302 - 3s/epoch - 12ms/step
Epoch 4/20
211/211 - 3s - loss: 0.2132 - accuracy: 0.9348 - val_loss: 0.2310 - val_accuracy: 0.9375 - 3s/epoch - 12ms/step
Epoch 5/20
211/211 - 2s - loss: 0.2117 - accuracy: 0.9362 - val_loss: 0.2792 - val_accuracy: 0.9165 - 2s/epoch - 12ms/step
Epoch 6/20
211/211 - 3s - loss: 0.2407 - accuracy: 0.9306 - val_loss: 0.2388 - val_accuracy: 0.9393 - 3s/epoch - 12ms/step
Epoch 7/20
```

```
211/211 - 2s - loss: 0.2007 - accuracy: 0.9396 - val_loss: 0.3405 -  
val_accuracy: 0.9302 - 2s/epoch - 12ms/step  
Epoch 8/20  
211/211 - 2s - loss: 0.2332 - accuracy: 0.9328 - val_loss: 0.2777 -  
val_accuracy: 0.9407 - 2s/epoch - 12ms/step  
Epoch 9/20  
211/211 - 2s - loss: 0.2089 - accuracy: 0.9394 - val_loss: 0.2349 -  
val_accuracy: 0.9380 - 2s/epoch - 12ms/step  
Epoch 10/20  
211/211 - 2s - loss: 0.2187 - accuracy: 0.9380 - val_loss: 0.3505 -  
val_accuracy: 0.9167 - 2s/epoch - 11ms/step  
Epoch 11/20  
211/211 - 2s - loss: 0.2388 - accuracy: 0.9329 - val_loss: 0.2616 -  
val_accuracy: 0.9388 - 2s/epoch - 12ms/step  
Epoch 12/20  
211/211 - 3s - loss: 0.2035 - accuracy: 0.9413 - val_loss: 0.2527 -  
val_accuracy: 0.9317 - 3s/epoch - 12ms/step  
Epoch 13/20  
211/211 - 3s - loss: 0.2193 - accuracy: 0.9373 - val_loss: 0.3018 -  
val_accuracy: 0.9382 - 3s/epoch - 12ms/step  
Epoch 14/20  
211/211 - 3s - loss: 0.2160 - accuracy: 0.9385 - val_loss: 0.2649 -  
val_accuracy: 0.9412 - 3s/epoch - 12ms/step  
Epoch 15/20  
211/211 - 2s - loss: 0.2205 - accuracy: 0.9392 - val_loss: 0.2993 -  
val_accuracy: 0.9300 - 2s/epoch - 12ms/step  
Epoch 16/20  
211/211 - 2s - loss: 0.2157 - accuracy: 0.9409 - val_loss: 0.4648 -  
val_accuracy: 0.9160 - 2s/epoch - 11ms/step  
Epoch 17/20  
211/211 - 2s - loss: 0.2902 - accuracy: 0.9212 - val_loss: 0.3074 -  
val_accuracy: 0.9262 - 2s/epoch - 12ms/step  
Epoch 18/20  
211/211 - 2s - loss: 0.2218 - accuracy: 0.9376 - val_loss: 0.3063 -  
val_accuracy: 0.9332 - 2s/epoch - 12ms/step  
Epoch 19/20  
211/211 - 3s - loss: 0.2713 - accuracy: 0.9290 - val_loss: 0.4076 -  
val_accuracy: 0.9187 - 3s/epoch - 12ms/step  
Epoch 20/20  
211/211 - 2s - loss: 0.2437 - accuracy: 0.9333 - val_loss: 0.2806 -  
val_accuracy: 0.9352 - 2s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.1, 256  
batch size and 20 epochs:  
[[ 963 1 1 1 0 3 6 2 2 1]  
 [ 2 1112 8 1 4 0 2 2 4 0]  
 [ 8 5 966 6 14 2 1 9 20 1]  
 [ 13 5 16 919 0 20 1 6 22 8]  
 [ 2 0 10 2 946 0 6 1 3 12]]
```

```
[ 8  1 16 15  3 800 17  5 27  0]
[ 8  4  3  1  8  6 926  0  2  0]
[ 2  1 14  5 16  0  2 947  9 32]
[ 11 2 28 30 14 17  5  5 841 21]
[ 8  3 13  7 24 13  0 26 13 902]]
```

Precision: 0.9322

Recall: 0.9322



Training with adam optimizer and the learning_rate is 0.1, 16 batch size and 5 epochs...

Epoch 1/5

3375/3375 - 15s - loss: 2.4864 - accuracy: 0.1030 - val_loss: 2.3227 - val_accuracy: 0.1113 - 15s/epoch - 4ms/step

Epoch 2/5

3375/3375 - 14s - loss: 2.3194 - accuracy: 0.1026 - val_loss: 2.3153 - val_accuracy: 0.1050 - 14s/epoch - 4ms/step

Epoch 3/5

3375/3375 - 14s - loss: 2.3197 - accuracy: 0.1016 - val_loss: 2.3168 - val_accuracy: 0.0978 - 14s/epoch - 4ms/step

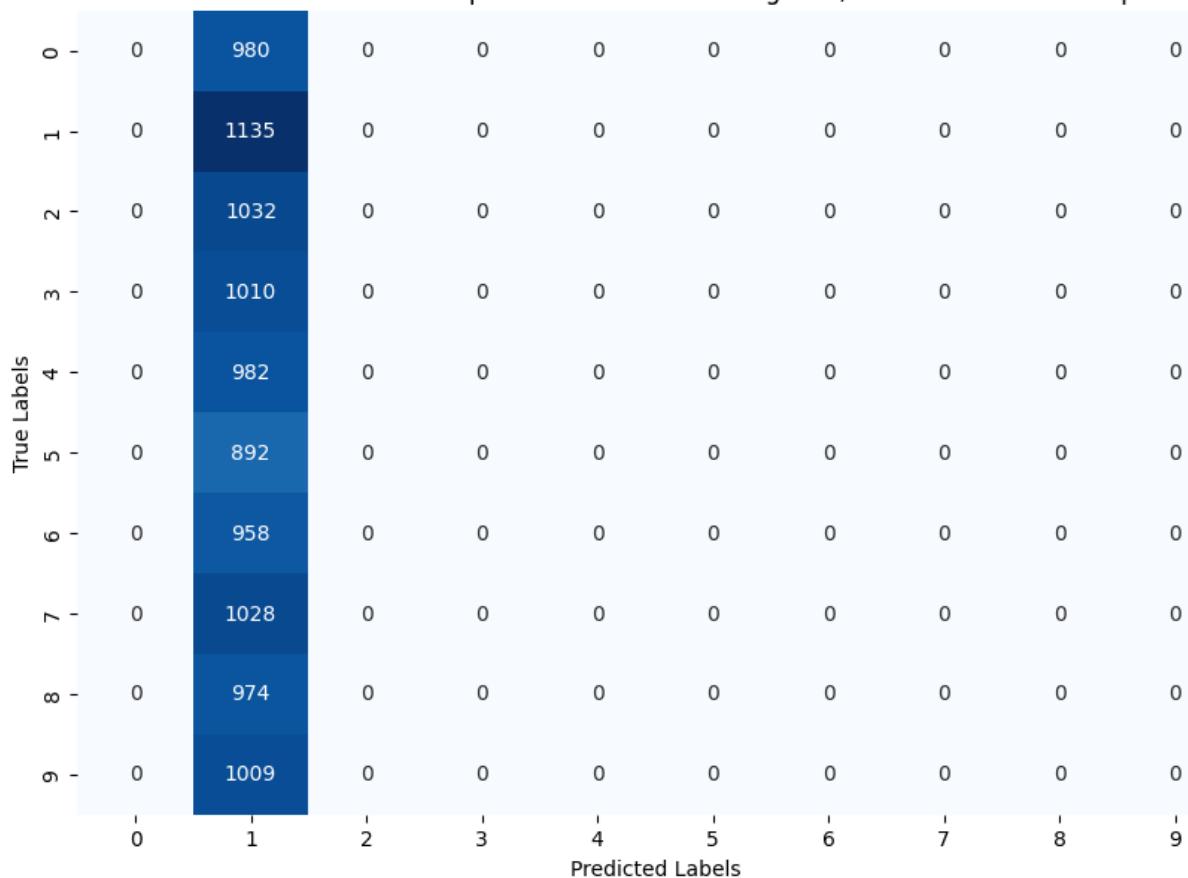
Epoch 4/5

3375/3375 - 14s - loss: 2.3195 - accuracy: 0.1034 - val_loss: 2.3183 -

```
val_accuracy: 0.0978 - 14s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 14s - loss: 2.3196 - accuracy: 0.1024 - val_loss: 2.3335 -
val_accuracy: 0.1050 - 14s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.1, 16 batch
size and 5 epochs:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adam optimizer and 0.1 learning rate, 16 batch size and 5 epochs



```
Training with adam optimizer and the learning_rate is 0.1, 16 batch size and 15 epochs...
```

```
Epoch 1/15
```

```
3375/3375 - 14s - loss: 2.5045 - accuracy: 0.1048 - val_loss: 2.3103 - val_accuracy: 0.1045 - 14s/epoch - 4ms/step
```

```
Epoch 2/15
```

```
3375/3375 - 14s - loss: 2.3198 - accuracy: 0.1016 - val_loss: 2.3194 - val_accuracy: 0.0992 - 14s/epoch - 4ms/step
```

```
Epoch 3/15
```

```
3375/3375 - 14s - loss: 2.3196 - accuracy: 0.1027 - val_loss: 2.3130 - val_accuracy: 0.1045 - 14s/epoch - 4ms/step
```

```
Epoch 4/15
```

```
3375/3375 - 14s - loss: 2.3201 - accuracy: 0.1029 - val_loss: 2.3233 - val_accuracy: 0.0960 - 14s/epoch - 4ms/step
```

```
Epoch 5/15
```

```
3375/3375 - 14s - loss: 2.3182 - accuracy: 0.1041 - val_loss: 2.3284 - val_accuracy: 0.0960 - 14s/epoch - 4ms/step
```

```
Epoch 6/15
```

```
3375/3375 - 14s - loss: 2.3202 - accuracy: 0.1027 - val_loss: 2.3143 - val_accuracy: 0.0978 - 14s/epoch - 4ms/step
```

```
Epoch 7/15
```

```
3375/3375 - 14s - loss: 2.3196 - accuracy: 0.1050 - val_loss: 2.3194 -  
val_accuracy: 0.1045 - 14s/epoch - 4ms/step  
Epoch 8/15  
3375/3375 - 14s - loss: 2.3198 - accuracy: 0.1036 - val_loss: 2.3098 -  
val_accuracy: 0.1113 - 14s/epoch - 4ms/step  
Epoch 9/15  
3375/3375 - 14s - loss: 2.3192 - accuracy: 0.1015 - val_loss: 2.3130 -  
val_accuracy: 0.1050 - 14s/epoch - 4ms/step  
Epoch 10/15  
3375/3375 - 14s - loss: 2.3200 - accuracy: 0.1033 - val_loss: 2.3275 -  
val_accuracy: 0.1050 - 14s/epoch - 4ms/step  
Epoch 11/15  
3375/3375 - 14s - loss: 2.3196 - accuracy: 0.1037 - val_loss: 2.3135 -  
val_accuracy: 0.1050 - 14s/epoch - 4ms/step  
Epoch 12/15  
3375/3375 - 14s - loss: 2.3194 - accuracy: 0.1049 - val_loss: 2.3281 -  
val_accuracy: 0.0995 - 14s/epoch - 4ms/step  
Epoch 13/15  
3375/3375 - 14s - loss: 2.3188 - accuracy: 0.1051 - val_loss: 2.3292 -  
val_accuracy: 0.0960 - 14s/epoch - 4ms/step  
Epoch 14/15  
3375/3375 - 14s - loss: 2.3196 - accuracy: 0.1038 - val_loss: 2.3184 -  
val_accuracy: 0.1050 - 14s/epoch - 4ms/step  
Epoch 15/15  
3375/3375 - 14s - loss: 2.3199 - accuracy: 0.1017 - val_loss: 2.3329 -  
val_accuracy: 0.0960 - 14s/epoch - 4ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.1, 16 batch  
size and 15 epochs:  
[[ 0  0  0  0  0  0  0  0  0  980]  
[ 0  0  0  0  0  0  0  0  0  1135]  
[ 0  0  0  0  0  0  0  0  0  1032]  
[ 0  0  0  0  0  0  0  0  0  1010]  
[ 0  0  0  0  0  0  0  0  0  982]  
[ 0  0  0  0  0  0  0  0  0  892]  
[ 0  0  0  0  0  0  0  0  0  958]  
[ 0  0  0  0  0  0  0  0  0  1028]  
[ 0  0  0  0  0  0  0  0  0  974]  
[ 0  0  0  0  0  0  0  0  0  1009]]  
Precision: 0.0102  
Recall: 0.1009  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for adam optimizer and 0.1 learning rate, 16 batch size and 15 epochs

	0 -	1 -	2 -	3 -	4 -	5 -	6 -	7 -	8 -	9	
True Labels	0	0	0	0	0	0	0	0	0	0	980
0 -	0	0	0	0	0	0	0	0	0	0	1135
1 -	0	0	0	0	0	0	0	0	0	0	1032
2 -	0	0	0	0	0	0	0	0	0	0	1010
3 -	0	0	0	0	0	0	0	0	0	0	982
4 -	0	0	0	0	0	0	0	0	0	0	892
5 -	0	0	0	0	0	0	0	0	0	0	958
6 -	0	0	0	0	0	0	0	0	0	0	1028
7 -	0	0	0	0	0	0	0	0	0	0	974
8 -	0	0	0	0	0	0	0	0	0	0	1009
9	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

```
Training with adam optimizer and the learning_rate is 0.1, 16 batch size and 20 epochs...
```

```
Epoch 1/20
```

```
3375/3375 - 15s - loss: 1.0028 - accuracy: 0.7678 - val_loss: 0.7809 - val_accuracy: 0.7902 - 15s/epoch - 4ms/step
```

```
Epoch 2/20
```

```
3375/3375 - 14s - loss: 1.1024 - accuracy: 0.7295 - val_loss: 1.6747 - val_accuracy: 0.4103 - 14s/epoch - 4ms/step
```

```
Epoch 3/20
```

```
3375/3375 - 13s - loss: 1.0708 - accuracy: 0.6890 - val_loss: 0.8208 - val_accuracy: 0.7483 - 13s/epoch - 4ms/step
```

```
Epoch 4/20
```

```
3375/3375 - 14s - loss: 2.2657 - accuracy: 0.1617 - val_loss: 2.3157 - val_accuracy: 0.0917 - 14s/epoch - 4ms/step
```

```
Epoch 5/20
```

```
3375/3375 - 14s - loss: 2.3231 - accuracy: 0.1051 - val_loss: 2.3255 - val_accuracy: 0.1045 - 14s/epoch - 4ms/step
```

```
Epoch 6/20
```

```
3375/3375 - 14s - loss: 2.3199 - accuracy: 0.1036 - val_loss: 2.3194 - val_accuracy: 0.0952 - 14s/epoch - 4ms/step
```

```
Epoch 7/20
```

```

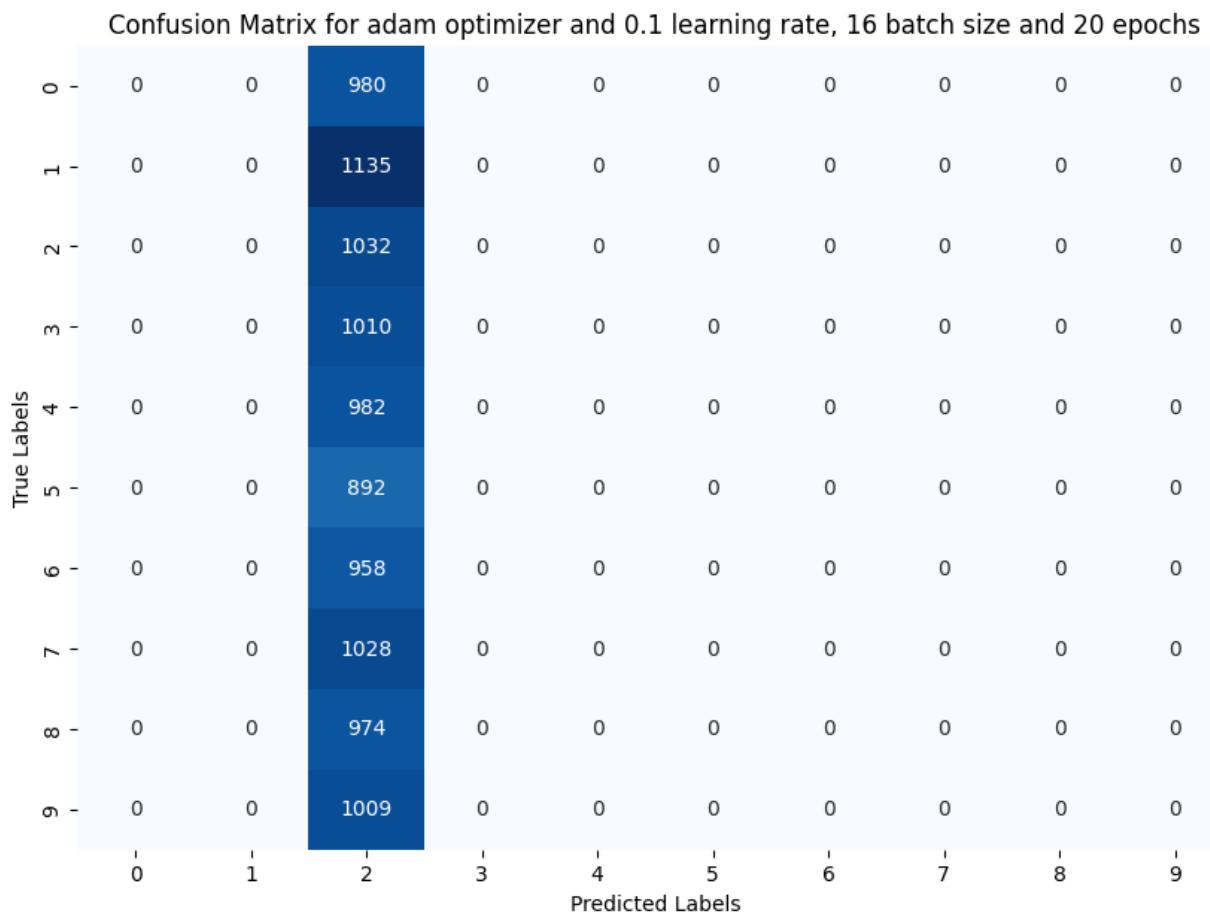
3375/3375 - 14s - loss: 2.3190 - accuracy: 0.1015 - val_loss: 2.3115 -
val_accuracy: 0.1113 - 14s/epoch - 4ms/step
Epoch 8/20
3375/3375 - 14s - loss: 2.3186 - accuracy: 0.1018 - val_loss: 2.3188 -
val_accuracy: 0.0960 - 14s/epoch - 4ms/step
Epoch 9/20
3375/3375 - 14s - loss: 2.3192 - accuracy: 0.1008 - val_loss: 2.3189 -
val_accuracy: 0.0978 - 14s/epoch - 4ms/step
Epoch 10/20
3375/3375 - 14s - loss: 2.3196 - accuracy: 0.1007 - val_loss: 2.3099 -
val_accuracy: 0.0952 - 14s/epoch - 4ms/step
Epoch 11/20
3375/3375 - 14s - loss: 2.3198 - accuracy: 0.1016 - val_loss: 2.3412 -
val_accuracy: 0.0952 - 14s/epoch - 4ms/step
Epoch 12/20
3375/3375 - 14s - loss: 2.3199 - accuracy: 0.1026 - val_loss: 2.3138 -
val_accuracy: 0.1113 - 14s/epoch - 4ms/step
Epoch 13/20
3375/3375 - 14s - loss: 2.3186 - accuracy: 0.1029 - val_loss: 2.3254 -
val_accuracy: 0.0992 - 14s/epoch - 4ms/step
Epoch 14/20
3375/3375 - 14s - loss: 2.3185 - accuracy: 0.1010 - val_loss: 2.3169 -
val_accuracy: 0.1050 - 14s/epoch - 4ms/step
Epoch 15/20
3375/3375 - 14s - loss: 2.3201 - accuracy: 0.1027 - val_loss: 2.3203 -
val_accuracy: 0.0978 - 14s/epoch - 4ms/step
Epoch 16/20
3375/3375 - 14s - loss: 2.3197 - accuracy: 0.1032 - val_loss: 2.3140 -
val_accuracy: 0.1113 - 14s/epoch - 4ms/step
Epoch 17/20
3375/3375 - 14s - loss: 2.3192 - accuracy: 0.1035 - val_loss: 2.3217 -
val_accuracy: 0.0995 - 14s/epoch - 4ms/step
Epoch 18/20
3375/3375 - 14s - loss: 2.3203 - accuracy: 0.1034 - val_loss: 2.3203 -
val_accuracy: 0.0915 - 14s/epoch - 4ms/step
Epoch 19/20
3375/3375 - 14s - loss: 2.3194 - accuracy: 0.1033 - val_loss: 2.3212 -
val_accuracy: 0.1113 - 14s/epoch - 4ms/step
Epoch 20/20
3375/3375 - 14s - loss: 2.3192 - accuracy: 0.1026 - val_loss: 2.3246 -
val_accuracy: 0.1000 - 14s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.1, 16 batch size and 20 epochs:
[[ 0  0 980  0  0  0  0  0  0]
 [ 0  0 1135  0  0  0  0  0  0]
 [ 0  0 1032  0  0  0  0  0  0]
 [ 0  0 1010  0  0  0  0  0  0]
 [ 0  0 982  0  0  0  0  0  0]]
```

```
[ 0  0 892  0  0  0  0  0  0  0]
[ 0  0 958  0  0  0  0  0  0  0]
[ 0  0 1028 0  0  0  0  0  0  0]
[ 0  0 974  0  0  0  0  0  0  0]
[ 0  0 1009 0  0  0  0  0  0  0]]
```

Precision: 0.0107

Recall: 0.1032

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training with adam optimizer and the learning_rate is 0.01, 64 batch size and 5 epochs...

Epoch 1/5

844/844 - 6s - loss: 0.2164 - accuracy: 0.9469 - val_loss: 0.0596 - val_accuracy: 0.9825 - 6s/epoch - 7ms/step

Epoch 2/5

844/844 - 5s - loss: 0.0595 - accuracy: 0.9814 - val_loss: 0.0650 -

```
val_accuracy: 0.9818 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 0.0434 - accuracy: 0.9863 - val_loss: 0.0738 -
val_accuracy: 0.9825 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 0.0385 - accuracy: 0.9877 - val_loss: 0.0902 -
val_accuracy: 0.9772 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.0334 - accuracy: 0.9897 - val_loss: 0.0877 -
val_accuracy: 0.9802 - 5s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.01, 64
batch size and 5 epochs:
[[ 974    1    1    1    0    1    0    0    2    0]
 [  1 1123    3    3    0    1    0    1    3    0]
 [  1    5 1008    4    1    1    0    8    4    0]
 [  0    1    0  996    0    8    0    2    2    1]
 [  0    0    1    0  972    0    4    0    1    4]
 [  2    0    0    5    0  881    2    0    2    0]
 [  7    5    0    0    3    6  934    0    3    0]
 [  0    1    9    5    5    4    0  998    1    5]
 [  3    1    2    3    2    7    0    3  950    3]
 [  4    3    3    9    9   25    0   11    8  937]]
```

Precision: 0.9775
Recall: 0.9773

Confusion Matrix for adam optimizer and 0.01 learning rate, 64 batch size and 5 epochs

	0	1	2	3	4	5	6	7	8	9
True Labels	974	1	1	1	0	1	0	0	2	0
0	1	1123	3	3	0	1	0	1	3	0
1	1	5	1008	4	1	1	0	8	4	0
2	0	1	0	996	0	8	0	2	2	1
3	0	0	1	0	972	0	4	0	1	4
4	2	0	0	5	0	881	2	0	2	0
5	7	5	0	0	3	6	934	0	3	0
6	0	1	9	5	5	4	0	998	1	5
7	3	1	2	3	2	7	0	3	950	3
8	4	3	3	9	9	25	0	11	8	937
9	0	1	2	3	4	5	6	7	8	9
Predicted Labels	0	1	2	3	4	5	6	7	8	9

```
Training with adam optimizer and the learning_rate is 0.01, 64 batch size and 15 epochs...
Epoch 1/15
844/844 - 6s - loss: 0.1665 - accuracy: 0.9544 - val_loss: 0.0745 -
val_accuracy: 0.9795 - 6s/epoch - 7ms/step
Epoch 2/15
844/844 - 5s - loss: 0.0547 - accuracy: 0.9827 - val_loss: 0.0626 -
val_accuracy: 0.9840 - 5s/epoch - 6ms/step
Epoch 3/15
844/844 - 5s - loss: 0.0388 - accuracy: 0.9881 - val_loss: 0.0705 -
val_accuracy: 0.9822 - 5s/epoch - 6ms/step
Epoch 4/15
844/844 - 5s - loss: 0.0334 - accuracy: 0.9895 - val_loss: 0.0735 -
val_accuracy: 0.9820 - 5s/epoch - 6ms/step
Epoch 5/15
844/844 - 5s - loss: 0.0299 - accuracy: 0.9909 - val_loss: 0.0679 -
val_accuracy: 0.9852 - 5s/epoch - 6ms/step
Epoch 6/15
844/844 - 5s - loss: 0.0336 - accuracy: 0.9902 - val_loss: 0.0973 -
val_accuracy: 0.9845 - 5s/epoch - 6ms/step
Epoch 7/15
```

```
844/844 - 5s - loss: 0.0249 - accuracy: 0.9930 - val_loss: 0.1053 -  
val_accuracy: 0.9837 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.0260 - accuracy: 0.9934 - val_loss: 0.1450 -  
val_accuracy: 0.9798 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.0248 - accuracy: 0.9940 - val_loss: 0.1662 -  
val_accuracy: 0.9810 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.0230 - accuracy: 0.9946 - val_loss: 0.1582 -  
val_accuracy: 0.9812 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.0260 - accuracy: 0.9940 - val_loss: 0.1512 -  
val_accuracy: 0.9837 - 5s/epoch - 5ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.0236 - accuracy: 0.9947 - val_loss: 0.1816 -  
val_accuracy: 0.9815 - 5s/epoch - 5ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.0216 - accuracy: 0.9957 - val_loss: 0.1992 -  
val_accuracy: 0.9827 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.0279 - accuracy: 0.9946 - val_loss: 0.2394 -  
val_accuracy: 0.9788 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.0216 - accuracy: 0.9955 - val_loss: 0.2196 -  
val_accuracy: 0.9837 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.01, 64  
batch size and 15 epochs:  
[[ 969 0 4 0 3 0 1 1 1 1 ]  
[ 0 1126 0 0 1 2 2 0 4 0 ]  
[ 1 1 1009 0 5 0 3 8 3 2 ]  
[ 0 0 4 998 0 5 0 0 2 1 ]  
[ 0 1 1 0 967 0 1 1 5 6 ]  
[ 2 0 1 13 0 867 4 0 3 2 ]  
[ 3 2 0 0 8 1 942 0 2 0 ]  
[ 0 7 9 3 3 0 0 996 4 6 ]  
[ 0 2 0 4 1 1 3 2 958 3 ]  
[ 0 0 3 3 16 1 0 4 4 978 ]]  
Precision: 0.9811  
Recall: 0.9810
```

Confusion Matrix for adam optimizer and 0.01 learning rate, 64 batch size and 15 epochs

	0	1	2	3	4	5	6	7	8	9	
0	969	0	4	0	3	0	1	1	1	1	0
1	0	1126	0	0	1	2	2	0	4	0	0
2	1	1	1009	0	5	0	3	8	3	2	0
3	0	0	4	998	0	5	0	0	2	1	0
4	0	1	1	0	967	0	1	1	5	6	0
5	2	0	1	13	0	867	4	0	3	2	0
6	3	2	0	0	8	1	942	0	2	0	0
7	0	7	9	3	3	0	0	996	4	6	0
8	0	2	0	4	1	1	3	2	958	3	0
9	0	0	3	3	16	1	0	4	4	978	0
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	

Training with adam optimizer and the learning_rate is 0.01, 64 batch size and 20 epochs...

Epoch 1/20

844/844 - 6s - loss: 0.1817 - accuracy: 0.9526 - val_loss: 0.0710 - val_accuracy: 0.9782 - 6s/epoch - 7ms/step

Epoch 2/20

844/844 - 5s - loss: 0.0550 - accuracy: 0.9833 - val_loss: 0.0653 - val_accuracy: 0.9822 - 5s/epoch - 6ms/step

Epoch 3/20

844/844 - 5s - loss: 0.0412 - accuracy: 0.9874 - val_loss: 0.0702 - val_accuracy: 0.9818 - 5s/epoch - 6ms/step

Epoch 4/20

844/844 - 5s - loss: 0.0344 - accuracy: 0.9890 - val_loss: 0.1063 - val_accuracy: 0.9768 - 5s/epoch - 6ms/step

Epoch 5/20

844/844 - 5s - loss: 0.0311 - accuracy: 0.9900 - val_loss: 0.0928 - val_accuracy: 0.9808 - 5s/epoch - 5ms/step

Epoch 6/20

844/844 - 5s - loss: 0.0294 - accuracy: 0.9909 - val_loss: 0.0941 - val_accuracy: 0.9822 - 5s/epoch - 5ms/step

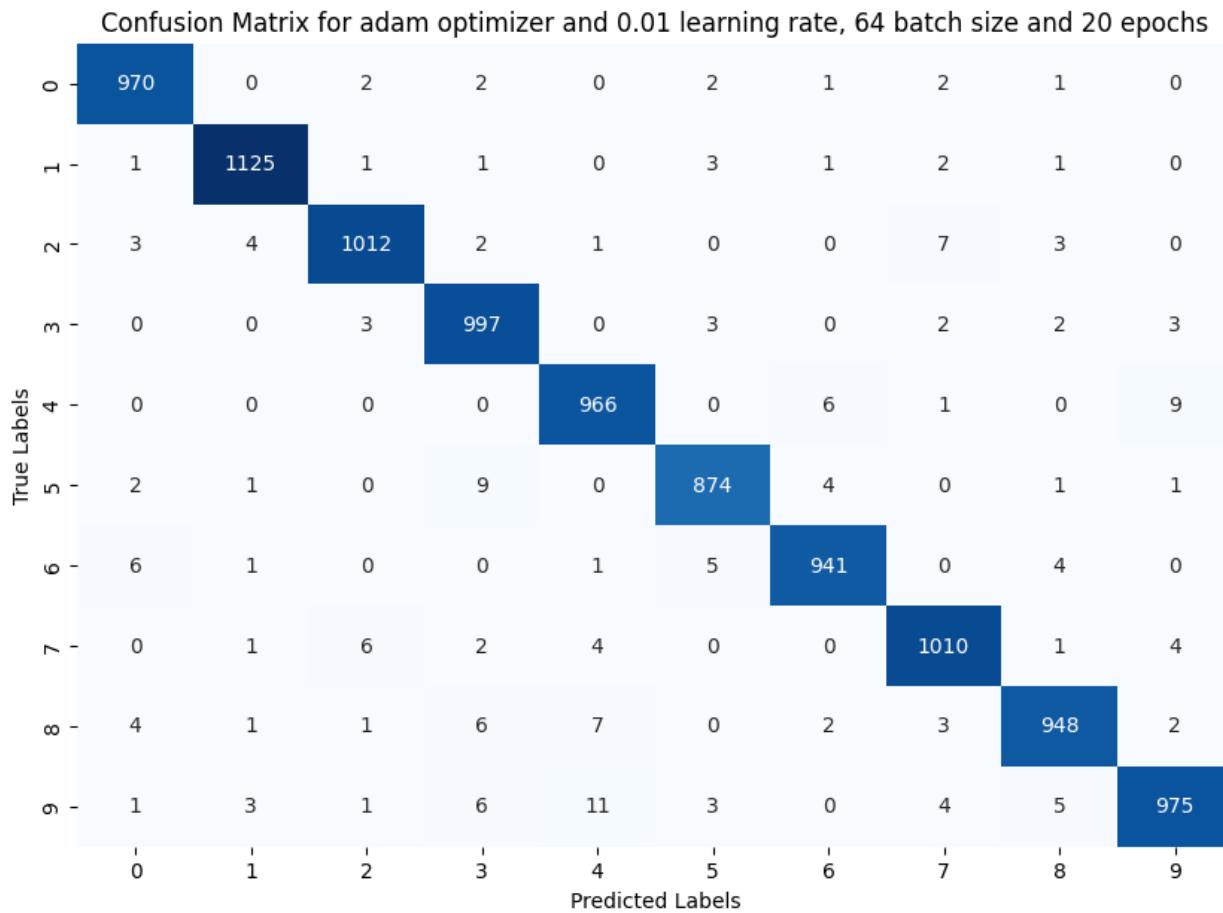
Epoch 7/20

```
844/844 - 5s - loss: 0.0289 - accuracy: 0.9916 - val_loss: 0.1121 -  
val_accuracy: 0.9820 - 5s/epoch - 6ms/step  
Epoch 8/20  
844/844 - 5s - loss: 0.0258 - accuracy: 0.9930 - val_loss: 0.1154 -  
val_accuracy: 0.9838 - 5s/epoch - 6ms/step  
Epoch 9/20  
844/844 - 5s - loss: 0.0300 - accuracy: 0.9921 - val_loss: 0.1324 -  
val_accuracy: 0.9822 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: 0.0231 - accuracy: 0.9943 - val_loss: 0.1337 -  
val_accuracy: 0.9818 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: 0.0194 - accuracy: 0.9952 - val_loss: 0.1162 -  
val_accuracy: 0.9842 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 4s - loss: 0.0210 - accuracy: 0.9946 - val_loss: 0.1234 -  
val_accuracy: 0.9842 - 4s/epoch - 5ms/step  
Epoch 13/20  
844/844 - 5s - loss: 0.0300 - accuracy: 0.9941 - val_loss: 0.1667 -  
val_accuracy: 0.9825 - 5s/epoch - 5ms/step  
Epoch 14/20  
844/844 - 5s - loss: 0.0174 - accuracy: 0.9958 - val_loss: 0.1845 -  
val_accuracy: 0.9830 - 5s/epoch - 5ms/step  
Epoch 15/20  
844/844 - 5s - loss: 0.0209 - accuracy: 0.9959 - val_loss: 0.2393 -  
val_accuracy: 0.9780 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: 0.0285 - accuracy: 0.9947 - val_loss: 0.2436 -  
val_accuracy: 0.9837 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: 0.0248 - accuracy: 0.9956 - val_loss: 0.2866 -  
val_accuracy: 0.9825 - 5s/epoch - 5ms/step  
Epoch 18/20  
844/844 - 5s - loss: 0.0208 - accuracy: 0.9964 - val_loss: 0.2468 -  
val_accuracy: 0.9823 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: 0.0169 - accuracy: 0.9968 - val_loss: 0.2660 -  
val_accuracy: 0.9857 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: 0.0155 - accuracy: 0.9973 - val_loss: 0.2347 -  
val_accuracy: 0.9888 - 5s/epoch - 5ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.01, 64  
batch size and 20 epochs:  
[[ 970 0 2 2 0 2 1 2 1 0]  
 [ 1 1125 1 1 0 3 1 2 1 0]  
 [ 3 4 1012 2 1 0 0 7 3 0]  
 [ 0 0 3 997 0 3 0 2 2 3]  
 [ 0 0 0 0 966 0 6 1 0 9]]
```

```
[ 2 1 0 9 0 874 4 0 1 1
[ 6 1 0 0 1 5 941 0 4 0
[ 0 1 6 2 4 0 0 1010 1 4
[ 4 1 1 6 7 0 2 3 948 2
[ 1 3 1 6 11 3 0 4 5 975]]
```

Precision: 0.9818

Recall: 0.9818



Training with adam optimizer and the learning_rate is 0.01, 128 batch size and 5 epochs...

Epoch 1/5

422/422 - 4s - loss: 0.2581 - accuracy: 0.9405 - val_loss: 0.0653 - val_accuracy: 0.9788 - 4s/epoch - 10ms/step

Epoch 2/5

422/422 - 3s - loss: 0.0534 - accuracy: 0.9836 - val_loss: 0.0524 - val_accuracy: 0.9852 - 3s/epoch - 7ms/step

Epoch 3/5

422/422 - 3s - loss: 0.0350 - accuracy: 0.9887 - val_loss: 0.0628 - val_accuracy: 0.9825 - 3s/epoch - 7ms/step

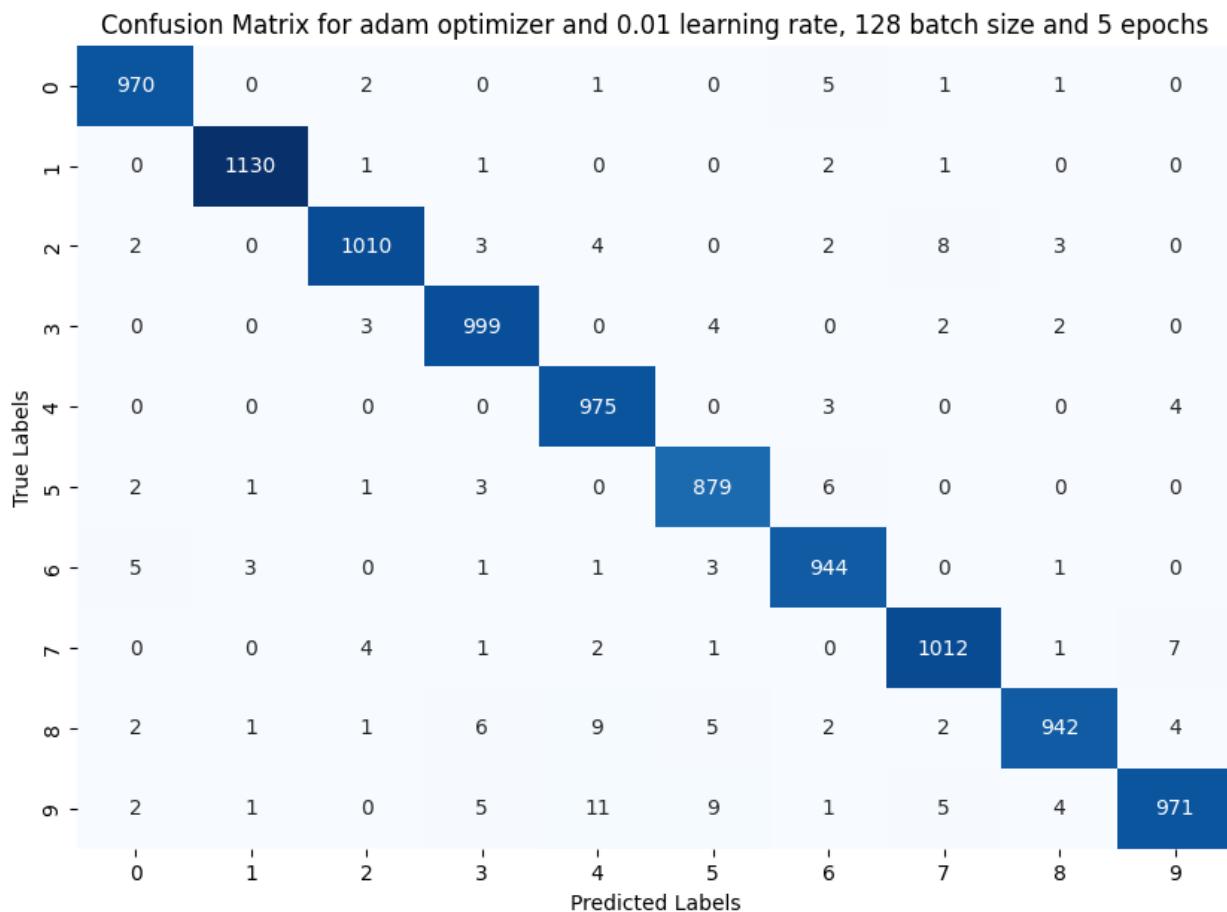
Epoch 4/5

422/422 - 3s - loss: 0.0277 - accuracy: 0.9906 - val_loss: 0.0669 -

```

val_accuracy: 0.9833 - 3s/epoch - 7ms/step
Epoch 5/5
422/422 - 3s - loss: 0.0191 - accuracy: 0.9937 - val_loss: 0.0644 -
val_accuracy: 0.9852 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.01, 128
batch size and 5 epochs:
[[ 970   0   2   0   1   0   5   1   1   0]
 [ 0 1130   1   1   0   0   2   1   0   0]
 [ 2   0 1010   3   4   0   2   8   3   0]
 [ 0   0   3 999   0   4   0   2   2   0]
 [ 0   0   0   0 975   0   3   0   0   4]
 [ 2   1   1   3   0 879   6   0   0   0]
 [ 5   3   0   1   1   3 944   0   1   0]
 [ 0   0   4   1   2   1   0 1012   1   7]
 [ 2   1   1   6   9   5   2   2 942   4]
 [ 2   1   0   5 11   9   1   5   4 971]]
Precision: 0.9833
Recall: 0.9832

```

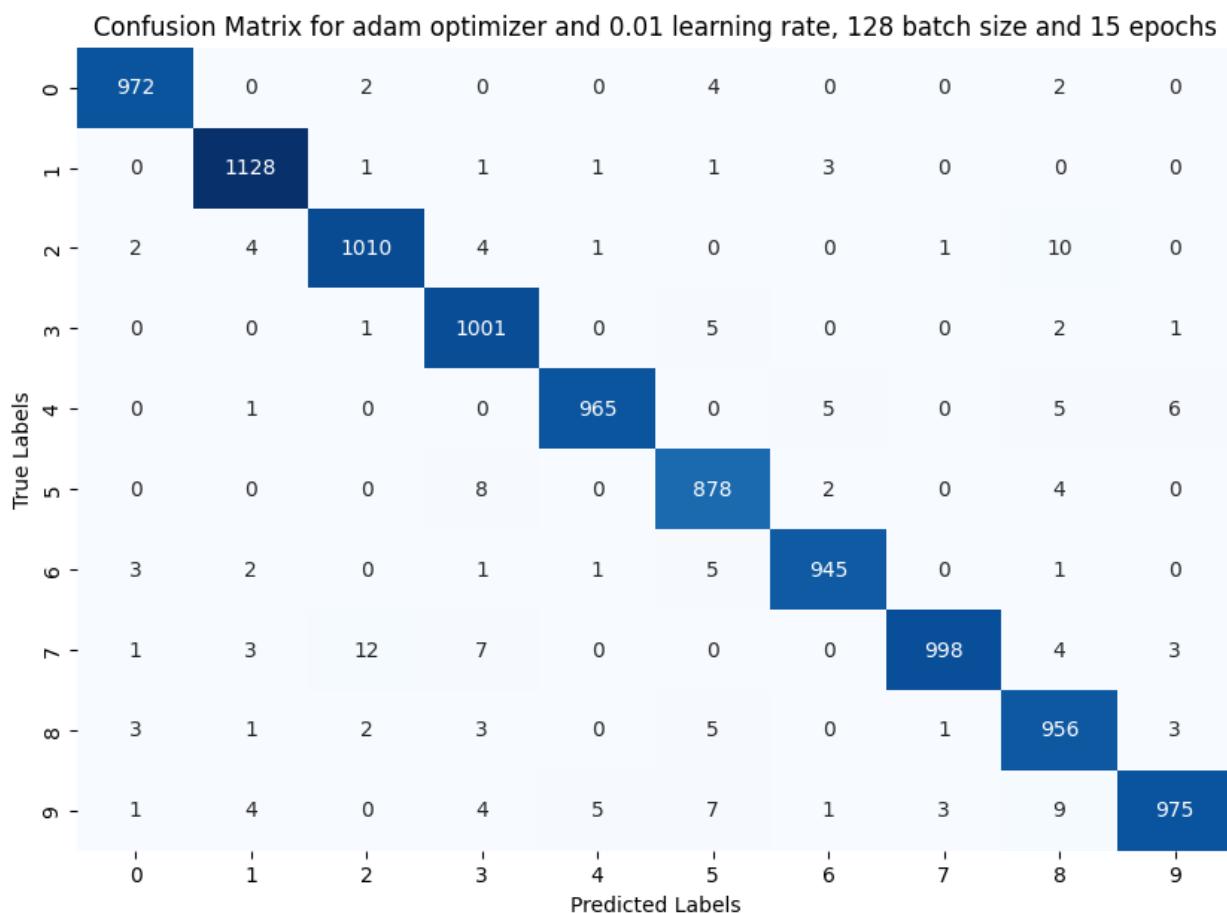


```
Training with adam optimizer and the learning_rate is 0.01, 128 batch size and 15 epochs...
Epoch 1/15
422/422 - 4s - loss: 0.2946 - accuracy: 0.9378 - val_loss: 0.0589 -
val_accuracy: 0.9840 - 4s/epoch - 9ms/step
Epoch 2/15
422/422 - 3s - loss: 0.0557 - accuracy: 0.9823 - val_loss: 0.0634 -
val_accuracy: 0.9818 - 3s/epoch - 7ms/step
Epoch 3/15
422/422 - 3s - loss: 0.0363 - accuracy: 0.9882 - val_loss: 0.0563 -
val_accuracy: 0.9842 - 3s/epoch - 7ms/step
Epoch 4/15
422/422 - 3s - loss: 0.0239 - accuracy: 0.9922 - val_loss: 0.0580 -
val_accuracy: 0.9842 - 3s/epoch - 7ms/step
Epoch 5/15
422/422 - 3s - loss: 0.0241 - accuracy: 0.9917 - val_loss: 0.0654 -
val_accuracy: 0.9845 - 3s/epoch - 7ms/step
Epoch 6/15
422/422 - 3s - loss: 0.0174 - accuracy: 0.9938 - val_loss: 0.0729 -
val_accuracy: 0.9822 - 3s/epoch - 8ms/step
Epoch 7/15
422/422 - 3s - loss: 0.0177 - accuracy: 0.9941 - val_loss: 0.0730 -
val_accuracy: 0.9840 - 3s/epoch - 7ms/step
Epoch 8/15
422/422 - 3s - loss: 0.0146 - accuracy: 0.9955 - val_loss: 0.0794 -
val_accuracy: 0.9852 - 3s/epoch - 7ms/step
Epoch 9/15
422/422 - 3s - loss: 0.0145 - accuracy: 0.9951 - val_loss: 0.0914 -
val_accuracy: 0.9842 - 3s/epoch - 7ms/step
Epoch 10/15
422/422 - 3s - loss: 0.0196 - accuracy: 0.9935 - val_loss: 0.0988 -
val_accuracy: 0.9830 - 3s/epoch - 7ms/step
Epoch 11/15
422/422 - 3s - loss: 0.0155 - accuracy: 0.9956 - val_loss: 0.0958 -
val_accuracy: 0.9833 - 3s/epoch - 7ms/step
Epoch 12/15
422/422 - 3s - loss: 0.0141 - accuracy: 0.9955 - val_loss: 0.0992 -
val_accuracy: 0.9848 - 3s/epoch - 7ms/step
Epoch 13/15
422/422 - 3s - loss: 0.0158 - accuracy: 0.9954 - val_loss: 0.1298 -
val_accuracy: 0.9793 - 3s/epoch - 7ms/step
Epoch 14/15
422/422 - 3s - loss: 0.0178 - accuracy: 0.9952 - val_loss: 0.1238 -
val_accuracy: 0.9832 - 3s/epoch - 7ms/step
Epoch 15/15
422/422 - 3s - loss: 0.0113 - accuracy: 0.9969 - val_loss: 0.1227 -
val_accuracy: 0.9833 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.01, 128 batch size and 15 epochs:
```

```
[[ 972  0   2   0   0   4   0   0   2   0
  [ 0 1128  1   1   1   1   3   0   0   0
  [ 2 4 1010  4   1   0   0   1   10  0
  [ 0 0 1 1001  0   5   0   0   2   1
  [ 0 1 0 0 965  0   5   0   5   6
  [ 0 0 0 8 0 878  2   0   4   0
  [ 3 2 0 1 1 5 945  0   1   0
  [ 1 3 12 7 0 0 0 998  4   3
  [ 3 1 2 3 0 5 0 1 956  3
  [ 1 4 0 4 5 7 1 3 9 975]]
```

Precision: 0.9829

Recall: 0.9828



Training with adam optimizer and the learning_rate is 0.01, 128 batch size and 20 epochs...

Epoch 1/20

422/422 - 4s - loss: 0.3157 - accuracy: 0.9345 - val_loss: 0.0875 - val_accuracy: 0.9752 - 4s/epoch - 9ms/step

Epoch 2/20

422/422 - 3s - loss: 0.0633 - accuracy: 0.9804 - val_loss: 0.0651 - val_accuracy: 0.9823 - 3s/epoch - 7ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.0413 - accuracy: 0.9868 - val_loss: 0.0686 -
val_accuracy: 0.9812 - 3s/epoch - 8ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0297 - accuracy: 0.9904 - val_loss: 0.0740 -
val_accuracy: 0.9805 - 3s/epoch - 7ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0269 - accuracy: 0.9909 - val_loss: 0.0688 -
val_accuracy: 0.9842 - 3s/epoch - 7ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0236 - accuracy: 0.9922 - val_loss: 0.0748 -
val_accuracy: 0.9825 - 3s/epoch - 7ms/step
Epoch 7/20
422/422 - 3s - loss: 0.0216 - accuracy: 0.9928 - val_loss: 0.0795 -
val_accuracy: 0.9835 - 3s/epoch - 7ms/step
Epoch 8/20
422/422 - 3s - loss: 0.0194 - accuracy: 0.9938 - val_loss: 0.0848 -
val_accuracy: 0.9823 - 3s/epoch - 7ms/step
Epoch 9/20
422/422 - 3s - loss: 0.0179 - accuracy: 0.9939 - val_loss: 0.0967 -
val_accuracy: 0.9833 - 3s/epoch - 7ms/step
Epoch 10/20
422/422 - 3s - loss: 0.0191 - accuracy: 0.9940 - val_loss: 0.0974 -
val_accuracy: 0.9847 - 3s/epoch - 7ms/step
Epoch 11/20
422/422 - 3s - loss: 0.0158 - accuracy: 0.9954 - val_loss: 0.1029 -
val_accuracy: 0.9840 - 3s/epoch - 7ms/step
Epoch 12/20
422/422 - 3s - loss: 0.0168 - accuracy: 0.9945 - val_loss: 0.1204 -
val_accuracy: 0.9812 - 3s/epoch - 8ms/step
Epoch 13/20
422/422 - 3s - loss: 0.0159 - accuracy: 0.9954 - val_loss: 0.1276 -
val_accuracy: 0.9802 - 3s/epoch - 8ms/step
Epoch 14/20
422/422 - 3s - loss: 0.0148 - accuracy: 0.9956 - val_loss: 0.1135 -
val_accuracy: 0.9845 - 3s/epoch - 7ms/step
Epoch 15/20
422/422 - 3s - loss: 0.0159 - accuracy: 0.9954 - val_loss: 0.1023 -
val_accuracy: 0.9842 - 3s/epoch - 7ms/step
Epoch 16/20
422/422 - 3s - loss: 0.0180 - accuracy: 0.9950 - val_loss: 0.1247 -
val_accuracy: 0.9825 - 3s/epoch - 7ms/step
Epoch 17/20
422/422 - 3s - loss: 0.0137 - accuracy: 0.9967 - val_loss: 0.1183 -
val_accuracy: 0.9848 - 3s/epoch - 7ms/step
Epoch 18/20
422/422 - 3s - loss: 0.0127 - accuracy: 0.9966 - val_loss: 0.1582 -
val_accuracy: 0.9822 - 3s/epoch - 7ms/step
Epoch 19/20
```

```
422/422 - 3s - loss: 0.0156 - accuracy: 0.9956 - val_loss: 0.1671 -  
val_accuracy: 0.9835 - 3s/epoch - 7ms/step
```

```
Epoch 20/20
```

```
422/422 - 3s - loss: 0.0173 - accuracy: 0.9958 - val_loss: 0.1700 -  
val_accuracy: 0.9830 - 3s/epoch - 7ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

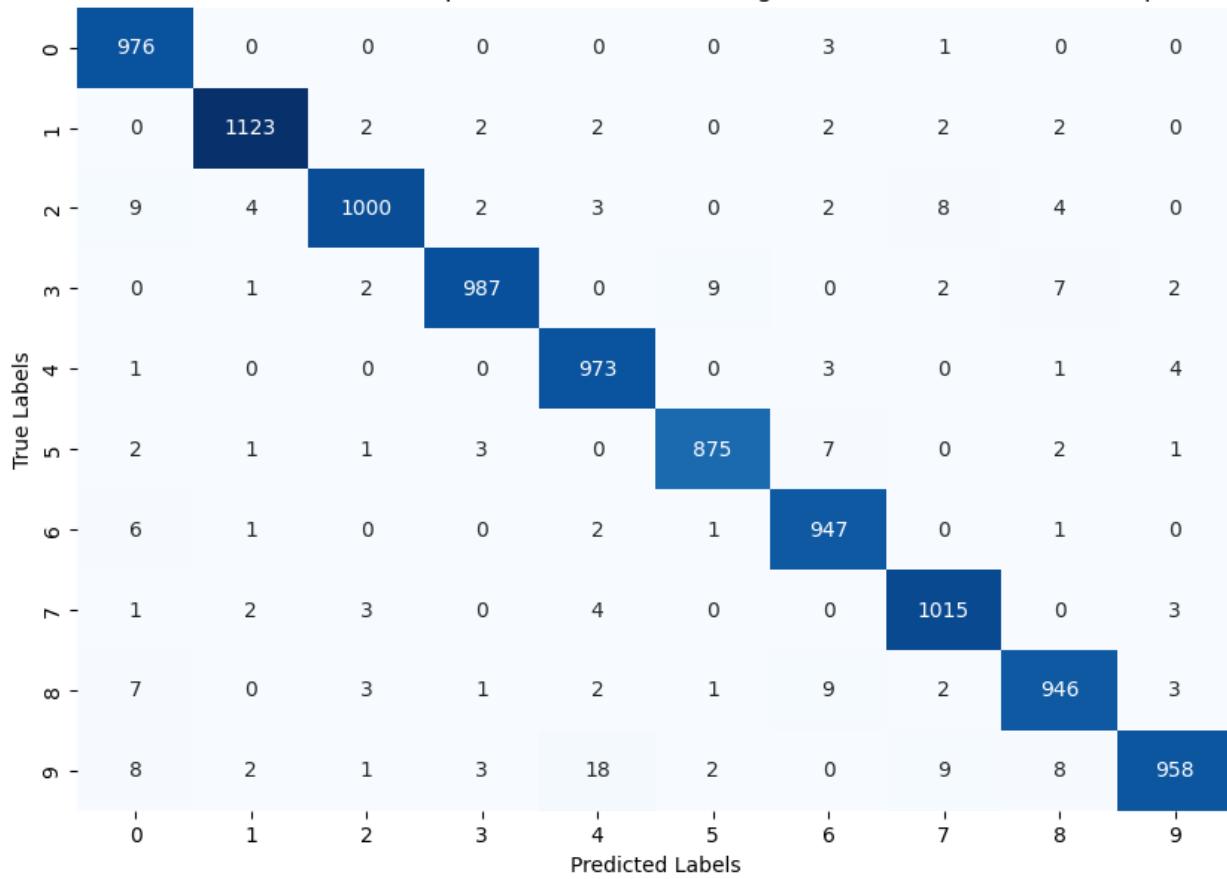
```
Confusion Matrix adam optimizer and the learning_rate is 0.01, 128  
batch size and 20 epochs:
```

```
[[ 976 0 0 0 0 0 3 1 0 0 ]  
[ 0 1123 2 2 2 0 2 2 2 0 ]  
[ 9 4 1000 2 3 0 2 8 4 0 ]  
[ 0 1 2 987 0 9 0 2 7 2 ]  
[ 1 0 0 0 973 0 3 0 1 4 ]  
[ 2 1 1 3 0 875 7 0 2 1 ]  
[ 6 1 0 0 2 1 947 0 1 0 ]  
[ 1 2 3 0 4 0 0 1015 0 3 ]  
[ 7 0 3 1 2 1 9 2 946 3 ]  
[ 8 2 1 3 18 2 0 9 8 958 ]]
```

```
Precision: 0.9801
```

```
Recall: 0.9800
```

Confusion Matrix for adam optimizer and 0.01 learning rate, 128 batch size and 20 epochs



```
Training with adam optimizer and the learning_rate is 0.01, 256 batch size and 5 epochs...
Epoch 1/5
211/211 - 3s - loss: 0.3661 - accuracy: 0.9150 - val_loss: 0.0569 -
val_accuracy: 0.9842 - 3s/epoch - 15ms/step
Epoch 2/5
211/211 - 2s - loss: 0.0574 - accuracy: 0.9819 - val_loss: 0.0621 -
val_accuracy: 0.9820 - 2s/epoch - 11ms/step
Epoch 3/5
211/211 - 2s - loss: 0.0353 - accuracy: 0.9891 - val_loss: 0.0586 -
val_accuracy: 0.9828 - 2s/epoch - 11ms/step
Epoch 4/5
211/211 - 2s - loss: 0.0234 - accuracy: 0.9926 - val_loss: 0.0577 -
val_accuracy: 0.9858 - 2s/epoch - 11ms/step
Epoch 5/5
211/211 - 2s - loss: 0.0168 - accuracy: 0.9946 - val_loss: 0.0550 -
val_accuracy: 0.9857 - 2s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.01, 256 batch size and 5 epochs:
[[ 972    1    1    1    0    1    3    0    1    0]
 [  1 1126    4    0    0    1    1    1    1    0]
 [  2    3 1014    2    0    0    2    6    3    0]
 [  0    0    3 994    0    9    0    3    1    0]
 [  0    1    1    0 964    0    2    1    1   12]
 [  0    0    0    4    0 884    4    0    0    0]
 [  2    1    0    1    1    5 947    0    1    0]
 [  0    1    15   1    0    0    0 1004    2    5]
 [  3    1    6    7    0   10    2    1 942    2]
 [  3    2    1    3    5    7    0    5    5 978]]
```

Precision: 0.9826
Recall: 0.9825

Confusion Matrix for adam optimizer and 0.01 learning rate, 256 batch size and 5 epochs

	0	1	2	3	4	5	6	7	8	9	
True Labels	972	1	1	1	0	1	3	0	1	0	
0											
1	1	1126	4	0	0	1	1	1	1	0	
2	2	3	1014	2	0	0	2	6	3	0	
3	0	0	3	994	0	9	0	3	1	0	
4	0	1	1	0	964	0	2	1	1	12	
5	0	0	0	4	0	884	4	0	0	0	
6	2	1	0	1	1	5	947	0	1	0	
7	0	1	15	1	0	0	0	1004	2	5	
8	3	1	6	7	0	10	2	1	942	2	
9	3	2	1	3	5	7	0	5	5	978	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```
Training with adam optimizer and the learning_rate is 0.01, 256 batch size and 15 epochs...
Epoch 1/15
211/211 - 3s - loss: 0.4066 - accuracy: 0.9165 - val_loss: 0.0705 -
val_accuracy: 0.9782 - 3s/epoch - 15ms/step
Epoch 2/15
211/211 - 2s - loss: 0.0633 - accuracy: 0.9802 - val_loss: 0.0690 -
val_accuracy: 0.9807 - 2s/epoch - 11ms/step
Epoch 3/15
211/211 - 2s - loss: 0.0392 - accuracy: 0.9877 - val_loss: 0.0663 -
val_accuracy: 0.9818 - 2s/epoch - 11ms/step
Epoch 4/15
211/211 - 2s - loss: 0.0285 - accuracy: 0.9904 - val_loss: 0.0561 -
val_accuracy: 0.9862 - 2s/epoch - 11ms/step
Epoch 5/15
211/211 - 2s - loss: 0.0203 - accuracy: 0.9932 - val_loss: 0.0612 -
val_accuracy: 0.9858 - 2s/epoch - 11ms/step
Epoch 6/15
211/211 - 2s - loss: 0.0151 - accuracy: 0.9953 - val_loss: 0.0623 -
val_accuracy: 0.9865 - 2s/epoch - 11ms/step
Epoch 7/15
```

```
211/211 - 2s - loss: 0.0141 - accuracy: 0.9952 - val_loss: 0.0663 -  
val_accuracy: 0.9855 - 2s/epoch - 12ms/step  
Epoch 8/15  
211/211 - 2s - loss: 0.0118 - accuracy: 0.9962 - val_loss: 0.0664 -  
val_accuracy: 0.9860 - 2s/epoch - 12ms/step  
Epoch 9/15  
211/211 - 2s - loss: 0.0136 - accuracy: 0.9954 - val_loss: 0.0698 -  
val_accuracy: 0.9857 - 2s/epoch - 11ms/step  
Epoch 10/15  
211/211 - 2s - loss: 0.0092 - accuracy: 0.9970 - val_loss: 0.0937 -  
val_accuracy: 0.9835 - 2s/epoch - 12ms/step  
Epoch 11/15  
211/211 - 2s - loss: 0.0139 - accuracy: 0.9953 - val_loss: 0.0826 -  
val_accuracy: 0.9848 - 2s/epoch - 11ms/step  
Epoch 12/15  
211/211 - 2s - loss: 0.0104 - accuracy: 0.9965 - val_loss: 0.0823 -  
val_accuracy: 0.9850 - 2s/epoch - 11ms/step  
Epoch 13/15  
211/211 - 2s - loss: 0.0065 - accuracy: 0.9978 - val_loss: 0.0811 -  
val_accuracy: 0.9858 - 2s/epoch - 12ms/step  
Epoch 14/15  
211/211 - 2s - loss: 0.0107 - accuracy: 0.9964 - val_loss: 0.0781 -  
val_accuracy: 0.9860 - 2s/epoch - 12ms/step  
Epoch 15/15  
211/211 - 2s - loss: 0.0073 - accuracy: 0.9975 - val_loss: 0.1006 -  
val_accuracy: 0.9830 - 2s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.01, 256  
batch size and 15 epochs:  
[[ 965 0 2 0 1 1 4 0 2 5]  
[ 0 1125 5 2 1 0 0 0 2 0]  
[ 1 1 1018 4 0 0 0 5 2 1]  
[ 0 0 2 983 0 12 0 3 1 9]  
[ 1 1 1 0 965 0 1 2 0 11]  
[ 1 1 1 7 0 876 4 0 0 2]  
[ 3 3 0 1 1 2 946 0 2 0]  
[ 0 2 11 4 0 1 0 998 1 11]  
[ 3 1 4 3 1 1 2 1 953 5]  
[ 0 1 2 3 4 5 0 1 1 992]]  
Precision: 0.9822  
Recall: 0.9821
```

Confusion Matrix for adam optimizer and 0.01 learning rate, 256 batch size and 15 epochs											
	0	1	2	3	4	5	6	7	8	9	
0	965	0	2	0	1	1	4	0	2	5	0
1	0	1125	5	2	1	0	0	0	2	0	0
2	1	1	1018	4	0	0	0	5	2	1	0
3	0	0	2	983	0	12	0	3	1	9	0
4	1	1	1	0	965	0	1	2	0	11	0
5	1	1	1	7	0	876	4	0	0	2	0
6	3	3	0	1	1	2	946	0	2	0	0
7	0	2	11	4	0	1	0	998	1	11	0
8	3	1	4	3	1	1	2	1	953	5	0
9	0	1	2	3	4	5	0	1	1	992	0
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

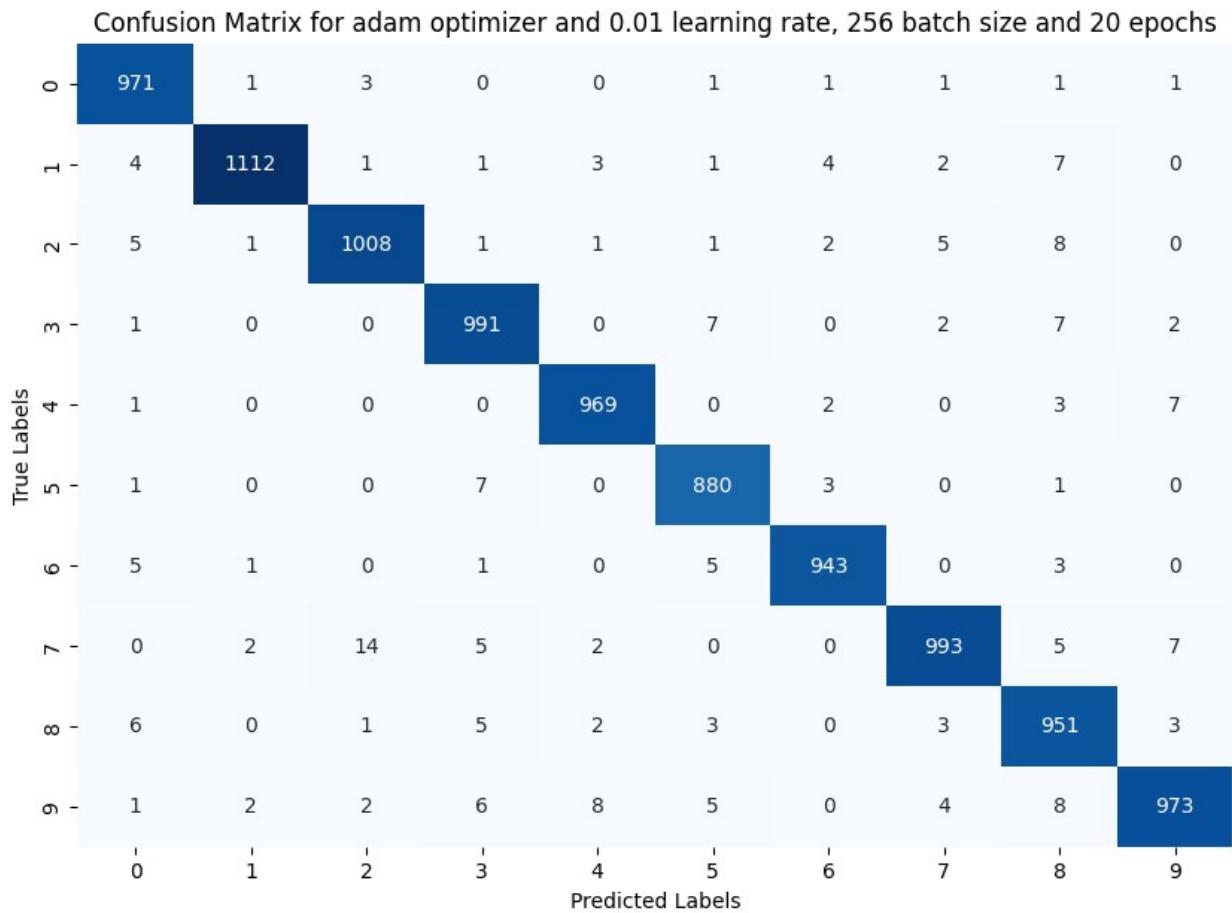
```
Training with adam optimizer and the learning_rate is 0.01, 256 batch size and 20 epochs...
Epoch 1/20
211/211 - 3s - loss: 0.3425 - accuracy: 0.9174 - val_loss: 0.0615 - val_accuracy: 0.9830 - 3s/epoch - 16ms/step
Epoch 2/20
211/211 - 2s - loss: 0.0565 - accuracy: 0.9822 - val_loss: 0.0555 - val_accuracy: 0.9835 - 2s/epoch - 11ms/step
Epoch 3/20
211/211 - 2s - loss: 0.0370 - accuracy: 0.9880 - val_loss: 0.0558 - val_accuracy: 0.9860 - 2s/epoch - 11ms/step
Epoch 4/20
211/211 - 2s - loss: 0.0266 - accuracy: 0.9911 - val_loss: 0.0660 - val_accuracy: 0.9830 - 2s/epoch - 12ms/step
Epoch 5/20
211/211 - 2s - loss: 0.0190 - accuracy: 0.9935 - val_loss: 0.0698 - val_accuracy: 0.9825 - 2s/epoch - 12ms/step
Epoch 6/20
211/211 - 2s - loss: 0.0138 - accuracy: 0.9953 - val_loss: 0.0617 - val_accuracy: 0.9857 - 2s/epoch - 11ms/step
Epoch 7/20
```

```
211/211 - 2s - loss: 0.0152 - accuracy: 0.9947 - val_loss: 0.0688 -  
val_accuracy: 0.9853 - 2s/epoch - 11ms/step  
Epoch 8/20  
211/211 - 2s - loss: 0.0115 - accuracy: 0.9956 - val_loss: 0.0664 -  
val_accuracy: 0.9857 - 2s/epoch - 11ms/step  
Epoch 9/20  
211/211 - 2s - loss: 0.0104 - accuracy: 0.9964 - val_loss: 0.0826 -  
val_accuracy: 0.9833 - 2s/epoch - 11ms/step  
Epoch 10/20  
211/211 - 2s - loss: 0.0115 - accuracy: 0.9962 - val_loss: 0.0976 -  
val_accuracy: 0.9837 - 2s/epoch - 11ms/step  
Epoch 11/20  
211/211 - 2s - loss: 0.0124 - accuracy: 0.9961 - val_loss: 0.0878 -  
val_accuracy: 0.9818 - 2s/epoch - 11ms/step  
Epoch 12/20  
211/211 - 2s - loss: 0.0118 - accuracy: 0.9962 - val_loss: 0.0985 -  
val_accuracy: 0.9842 - 2s/epoch - 11ms/step  
Epoch 13/20  
211/211 - 2s - loss: 0.0157 - accuracy: 0.9948 - val_loss: 0.1280 -  
val_accuracy: 0.9792 - 2s/epoch - 11ms/step  
Epoch 14/20  
211/211 - 2s - loss: 0.0118 - accuracy: 0.9961 - val_loss: 0.0865 -  
val_accuracy: 0.9852 - 2s/epoch - 12ms/step  
Epoch 15/20  
211/211 - 2s - loss: 0.0073 - accuracy: 0.9974 - val_loss: 0.0979 -  
val_accuracy: 0.9858 - 2s/epoch - 11ms/step  
Epoch 16/20  
211/211 - 2s - loss: 0.0055 - accuracy: 0.9982 - val_loss: 0.1035 -  
val_accuracy: 0.9850 - 2s/epoch - 11ms/step  
Epoch 17/20  
211/211 - 2s - loss: 0.0074 - accuracy: 0.9976 - val_loss: 0.0948 -  
val_accuracy: 0.9865 - 2s/epoch - 11ms/step  
Epoch 18/20  
211/211 - 2s - loss: 0.0092 - accuracy: 0.9974 - val_loss: 0.1026 -  
val_accuracy: 0.9840 - 2s/epoch - 11ms/step  
Epoch 19/20  
211/211 - 2s - loss: 0.0105 - accuracy: 0.9968 - val_loss: 0.1019 -  
val_accuracy: 0.9845 - 2s/epoch - 12ms/step  
Epoch 20/20  
211/211 - 2s - loss: 0.0095 - accuracy: 0.9971 - val_loss: 0.1244 -  
val_accuracy: 0.9835 - 2s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.01, 256  
batch size and 20 epochs:  
[[ 971    1     3     0     0     1     1     1     1     1]  
 [   4 1112    1     1     3     1     4     2     7     0]  
 [   5    1 1008    1     1     1     2     5     8     0]  
 [   1    0     0   991     0     7     0     2     7     2]  
 [   1    0     0     0   969     0     2     0     3     7]]
```

```
[ 1 0 0 7 0 880 3 0 1 0]
[ 5 1 0 1 0 5 943 0 3 0]
[ 0 2 14 5 2 0 0 993 5 7]
[ 6 0 1 5 2 3 0 3 951 3]
[ 1 2 2 6 8 5 0 4 8 973]]
```

Precision: 0.9792

Recall: 0.9791



Training with adam optimizer and the learning_rate is 0.01, 16 batch size and 5 epochs...

Epoch 1/5

3375/3375 - 15s - loss: 0.1681 - accuracy: 0.9527 - val_loss: 0.0829 - val_accuracy: 0.9753 - 15s/epoch - 4ms/step

Epoch 2/5

3375/3375 - 14s - loss: 0.0882 - accuracy: 0.9747 - val_loss: 0.1120 - val_accuracy: 0.9707 - 14s/epoch - 4ms/step

Epoch 3/5

3375/3375 - 14s - loss: 0.0766 - accuracy: 0.9795 - val_loss: 0.1157 - val_accuracy: 0.9748 - 14s/epoch - 4ms/step

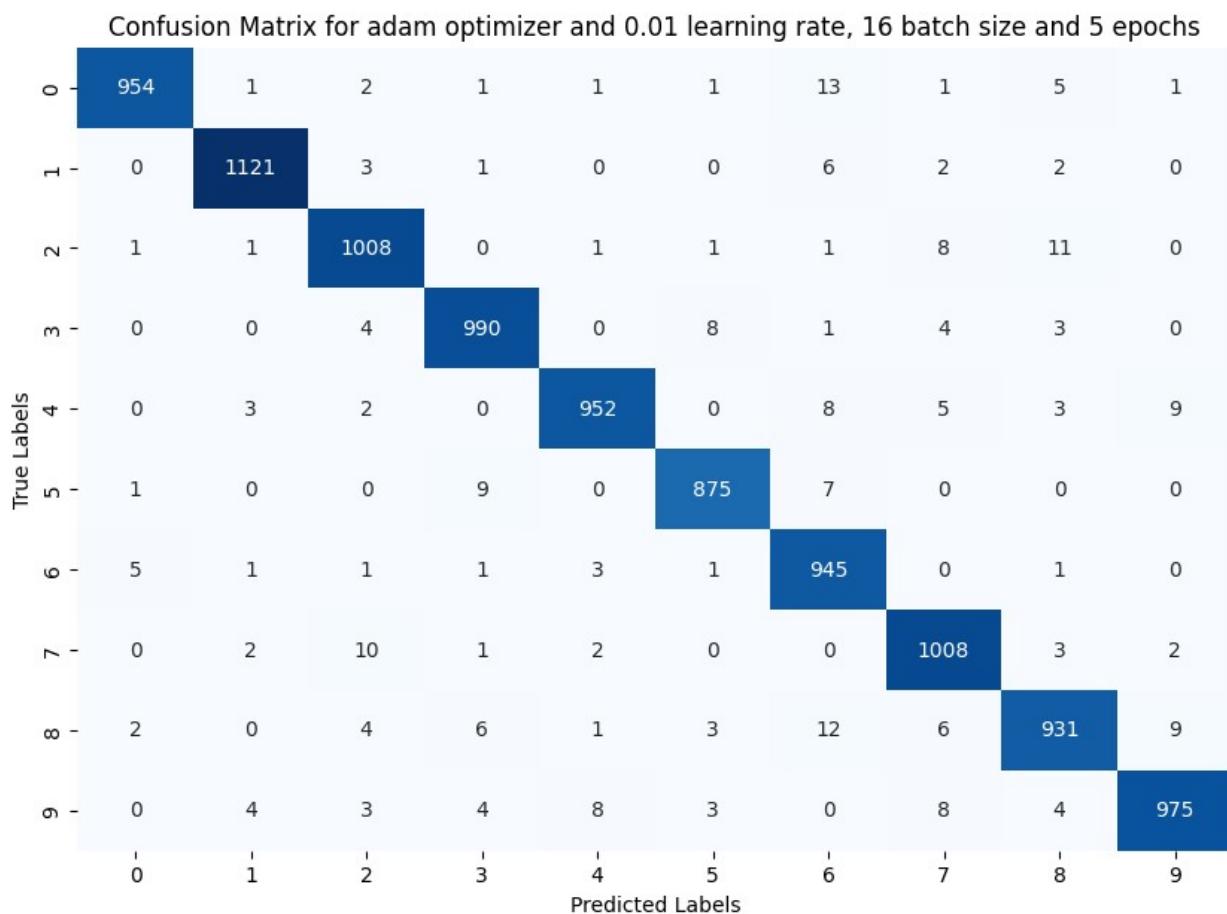
Epoch 4/5

3375/3375 - 14s - loss: 0.0667 - accuracy: 0.9819 - val_loss: 0.1096 -

```

val_accuracy: 0.9787 - 14s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 14s - loss: 0.0599 - accuracy: 0.9845 - val_loss: 0.1258 -
val_accuracy: 0.9783 - 14s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.01, 16
batch size and 5 epochs:
[[ 954   1   2   1   1   1   13   1   5   1]
 [ 0 1121   3   1   0   0   6   2   2   0]
 [ 1   1 1008   0   1   1   1   8   11   0]
 [ 0   0   4 990   0   8   1   4   3   0]
 [ 0   3   2   0 952   0   8   5   3   9]
 [ 1   0   0   9   0 875   7   0   0   0]
 [ 5   1   1   1   3   1 945   0   1   0]
 [ 0   2   10  1   2   0   0 1008   3   2]
 [ 2   0   4   6   1   3   12   6 931   9]
 [ 0   4   3   4   8   3   0   8   4 975]]
Precision: 0.9760
Recall: 0.9759

```



Training with adam optimizer and the learning_rate is 0.01, 16 batch size and 15 epochs...

Epoch 1/15
3375/3375 - 15s - loss: 0.1725 - accuracy: 0.9521 - val_loss: 0.0797 - val_accuracy: 0.9790 - 15s/epoch - 4ms/step

Epoch 2/15
3375/3375 - 14s - loss: 0.0890 - accuracy: 0.9744 - val_loss: 0.0933 - val_accuracy: 0.9760 - 14s/epoch - 4ms/step

Epoch 3/15
3375/3375 - 14s - loss: 0.0740 - accuracy: 0.9794 - val_loss: 0.0820 - val_accuracy: 0.9785 - 14s/epoch - 4ms/step

Epoch 4/15
3375/3375 - 14s - loss: 0.0599 - accuracy: 0.9841 - val_loss: 0.1470 - val_accuracy: 0.9695 - 14s/epoch - 4ms/step

Epoch 5/15
3375/3375 - 14s - loss: 0.0618 - accuracy: 0.9844 - val_loss: 0.1063 - val_accuracy: 0.9778 - 14s/epoch - 4ms/step

Epoch 6/15
3375/3375 - 14s - loss: 0.0563 - accuracy: 0.9868 - val_loss: 0.1275 - val_accuracy: 0.9788 - 14s/epoch - 4ms/step

Epoch 7/15
3375/3375 - 14s - loss: 0.0488 - accuracy: 0.9893 - val_loss: 0.1208 - val_accuracy: 0.9827 - 14s/epoch - 4ms/step

Epoch 8/15
3375/3375 - 14s - loss: 0.0534 - accuracy: 0.9895 - val_loss: 0.1946 - val_accuracy: 0.9792 - 14s/epoch - 4ms/step

Epoch 9/15
3375/3375 - 14s - loss: 0.0546 - accuracy: 0.9896 - val_loss: 0.2080 - val_accuracy: 0.9785 - 14s/epoch - 4ms/step

Epoch 10/15
3375/3375 - 14s - loss: 0.0476 - accuracy: 0.9909 - val_loss: 0.3415 - val_accuracy: 0.9760 - 14s/epoch - 4ms/step

Epoch 11/15
3375/3375 - 14s - loss: 0.0501 - accuracy: 0.9918 - val_loss: 0.2196 - val_accuracy: 0.9828 - 14s/epoch - 4ms/step

Epoch 12/15
3375/3375 - 14s - loss: 0.0439 - accuracy: 0.9922 - val_loss: 0.2714 - val_accuracy: 0.9798 - 14s/epoch - 4ms/step

Epoch 13/15
3375/3375 - 14s - loss: 0.0508 - accuracy: 0.9926 - val_loss: 0.2619 - val_accuracy: 0.9793 - 14s/epoch - 4ms/step

Epoch 14/15
3375/3375 - 14s - loss: 0.0494 - accuracy: 0.9930 - val_loss: 0.2854 - val_accuracy: 0.9835 - 14s/epoch - 4ms/step

Epoch 15/15
3375/3375 - 14s - loss: 0.0502 - accuracy: 0.9935 - val_loss: 0.3767 - val_accuracy: 0.9800 - 14s/epoch - 4ms/step

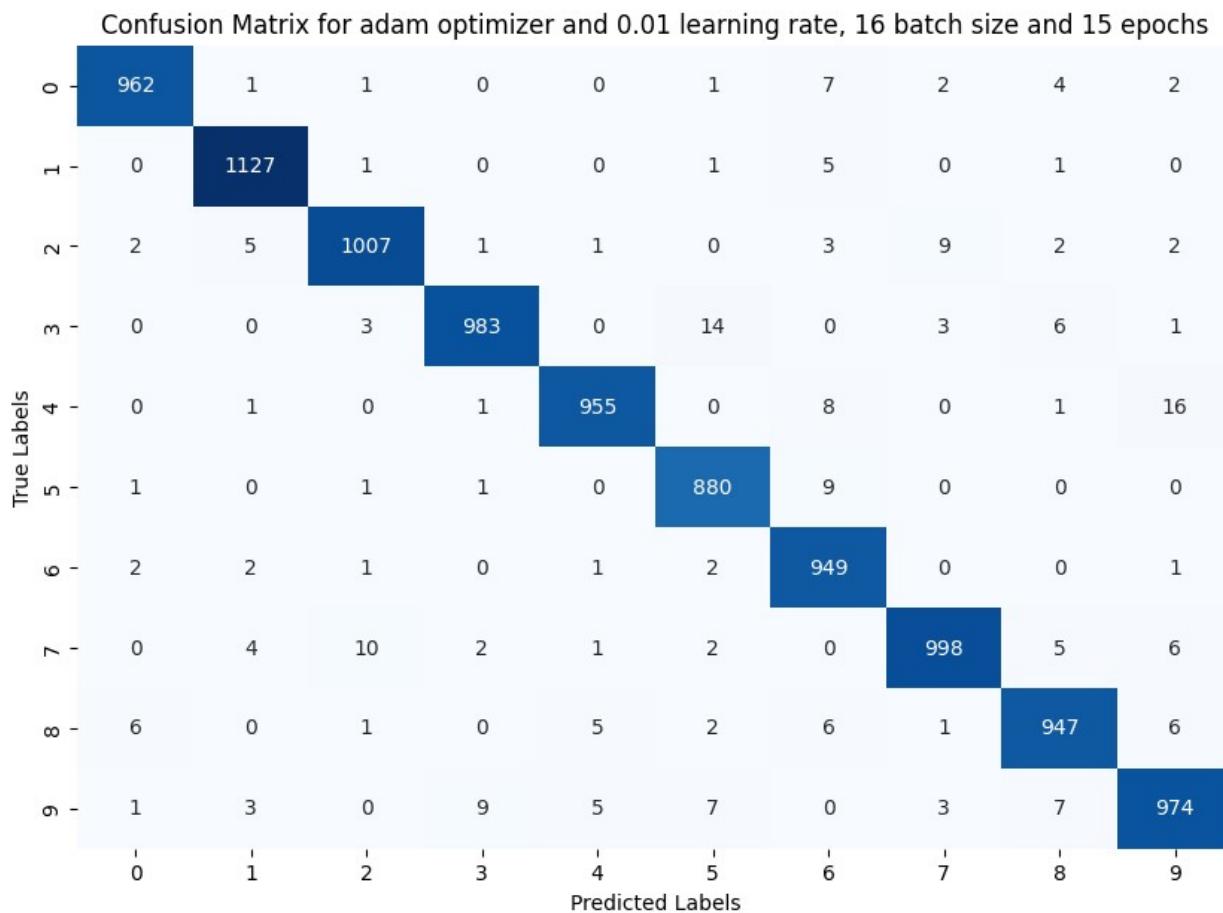
313/313 [=====] - 1s 2ms/step

Confusion Matrix adam optimizer and the learning_rate is 0.01, 16 batch size and 15 epochs:

```
[[ 962  1  1  0  0  1  7  2  4  2]
 [ 0 1127  1  0  0  1  5  0  1  0]
 [ 2  5 1007  1  1  0  3  9  2  2]
 [ 0  0  3  983  0  14  0  3  6  1]
 [ 0  1  0  1  955  0  8  0  1  16]
 [ 1  0  1  1  0  880  9  0  0  0]
 [ 2  2  1  0  1  2  949  0  0  1]
 [ 0  4  10  2  1  2  0  998  5  6]
 [ 6  0  1  0  5  2  6  1  947  6]
 [ 1  3  0  9  5  7  0  3  7  974]]
```

Precision: 0.9783

Recall: 0.9782



Training with adam optimizer and the learning_rate is 0.01, 16 batch size and 20 epochs...

Epoch 1/20

3375/3375 - 14s - loss: 0.1738 - accuracy: 0.9534 - val_loss: 0.1020 - val_accuracy: 0.9720 - 14s/epoch - 4ms/step

Epoch 2/20

3375/3375 - 14s - loss: 0.0935 - accuracy: 0.9737 - val_loss: 0.0854 - val_accuracy: 0.9787 - 14s/epoch - 4ms/step

```
Epoch 3/20
3375/3375 - 14s - loss: 0.0742 - accuracy: 0.9799 - val_loss: 0.1095 -
val_accuracy: 0.9740 - 14s/epoch - 4ms/step
Epoch 4/20
3375/3375 - 14s - loss: 0.0689 - accuracy: 0.9830 - val_loss: 0.1245 -
val_accuracy: 0.9762 - 14s/epoch - 4ms/step
Epoch 5/20
3375/3375 - 14s - loss: 0.0688 - accuracy: 0.9843 - val_loss: 0.1096 -
val_accuracy: 0.9793 - 14s/epoch - 4ms/step
Epoch 6/20
3375/3375 - 14s - loss: 0.0598 - accuracy: 0.9873 - val_loss: 0.1919 -
val_accuracy: 0.9795 - 14s/epoch - 4ms/step
Epoch 7/20
3375/3375 - 13s - loss: 0.0594 - accuracy: 0.9882 - val_loss: 0.1667 -
val_accuracy: 0.9840 - 13s/epoch - 4ms/step
Epoch 8/20
3375/3375 - 14s - loss: 0.0605 - accuracy: 0.9893 - val_loss: 0.1611 -
val_accuracy: 0.9752 - 14s/epoch - 4ms/step
Epoch 9/20
3375/3375 - 14s - loss: 0.0559 - accuracy: 0.9896 - val_loss: 0.1758 -
val_accuracy: 0.9803 - 14s/epoch - 4ms/step
Epoch 10/20
3375/3375 - 14s - loss: 0.0579 - accuracy: 0.9902 - val_loss: 0.2363 -
val_accuracy: 0.9817 - 14s/epoch - 4ms/step
Epoch 11/20
3375/3375 - 14s - loss: 0.0509 - accuracy: 0.9914 - val_loss: 0.2563 -
val_accuracy: 0.9812 - 14s/epoch - 4ms/step
Epoch 12/20
3375/3375 - 14s - loss: 0.0507 - accuracy: 0.9926 - val_loss: 0.2927 -
val_accuracy: 0.9813 - 14s/epoch - 4ms/step
Epoch 13/20
3375/3375 - 14s - loss: 0.0594 - accuracy: 0.9929 - val_loss: 0.3660 -
val_accuracy: 0.9817 - 14s/epoch - 4ms/step
Epoch 14/20
3375/3375 - 14s - loss: 0.0550 - accuracy: 0.9938 - val_loss: 0.4346 -
val_accuracy: 0.9838 - 14s/epoch - 4ms/step
Epoch 15/20
3375/3375 - 14s - loss: 0.0661 - accuracy: 0.9927 - val_loss: 0.5494 -
val_accuracy: 0.9818 - 14s/epoch - 4ms/step
Epoch 16/20
3375/3375 - 14s - loss: 0.0653 - accuracy: 0.9938 - val_loss: 0.3360 -
val_accuracy: 0.9820 - 14s/epoch - 4ms/step
Epoch 17/20
3375/3375 - 14s - loss: 0.0535 - accuracy: 0.9944 - val_loss: 0.6303 -
val_accuracy: 0.9783 - 14s/epoch - 4ms/step
Epoch 18/20
3375/3375 - 14s - loss: 0.0796 - accuracy: 0.9939 - val_loss: 0.5156 -
val_accuracy: 0.9780 - 14s/epoch - 4ms/step
Epoch 19/20
```

3375/3375 - 14s - loss: 0.0674 - accuracy: 0.9948 - val_loss: 0.5067 -
val_accuracy: 0.9853 - 14s/epoch - 4ms/step

Epoch 20/20

3375/3375 - 14s - loss: 0.0714 - accuracy: 0.9949 - val_loss: 0.7090 -
val_accuracy: 0.9827 - 14s/epoch - 4ms/step

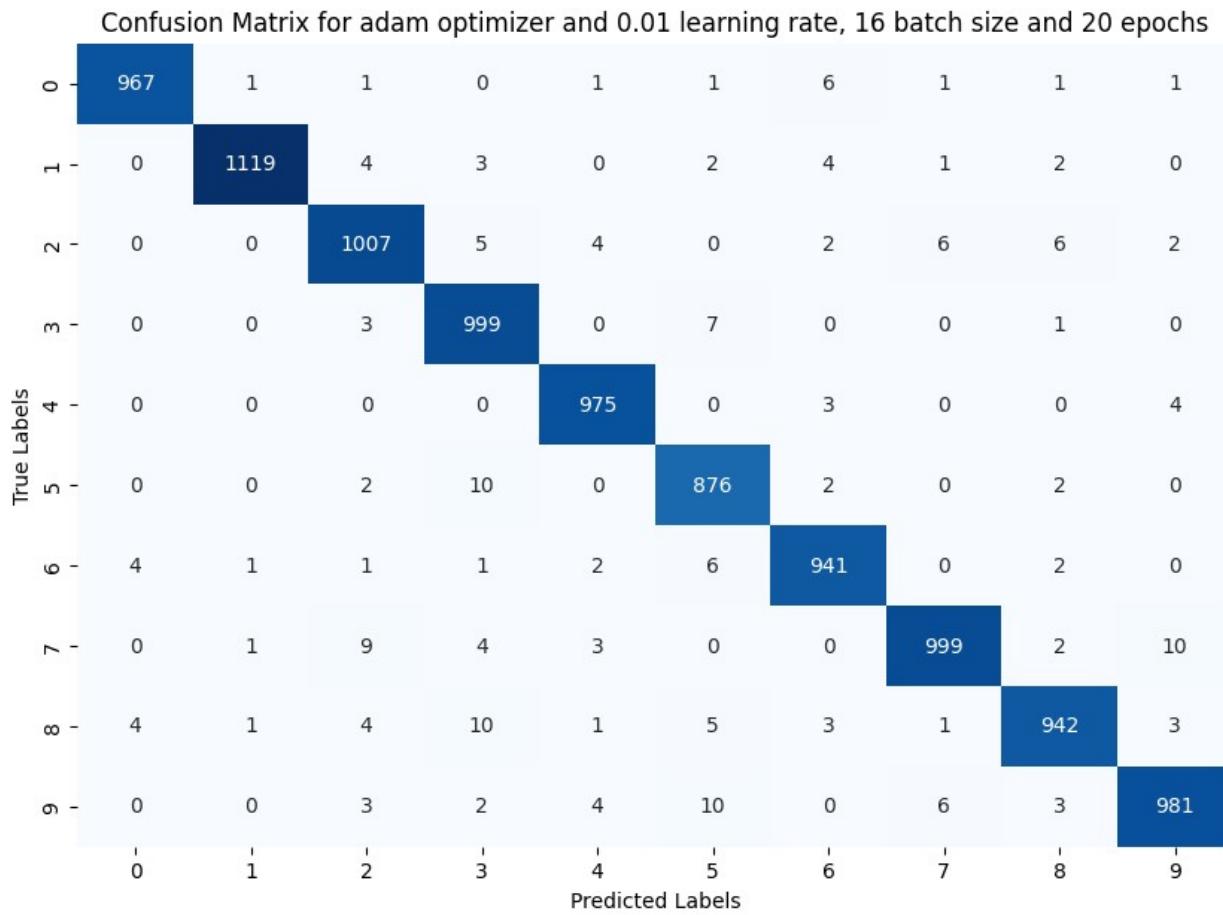
313/313 [=====] - 1s 2ms/step

Confusion Matrix adam optimizer and the learning_rate is 0.01, 16
batch size and 20 epochs:

```
[[ 967   1   1   0   1   1   6   1   1   1]
 [ 0 1119   4   3   0   2   4   1   2   0]
 [ 0   0 1007   5   4   0   2   6   6   2]
 [ 0   0   3 999   0   7   0   0   1   0]
 [ 0   0   0   0 975   0   3   0   0   4]
 [ 0   0   2   10   0 876   2   0   2   0]
 [ 4   1   1   1   2   6 941   0   2   0]
 [ 0   1   9   4   3   0   0 999   2   10]
 [ 4   1   4   10   1   5   3   1 942   3]
 [ 0   0   3   2   4   10   0   6   3 981]]
```

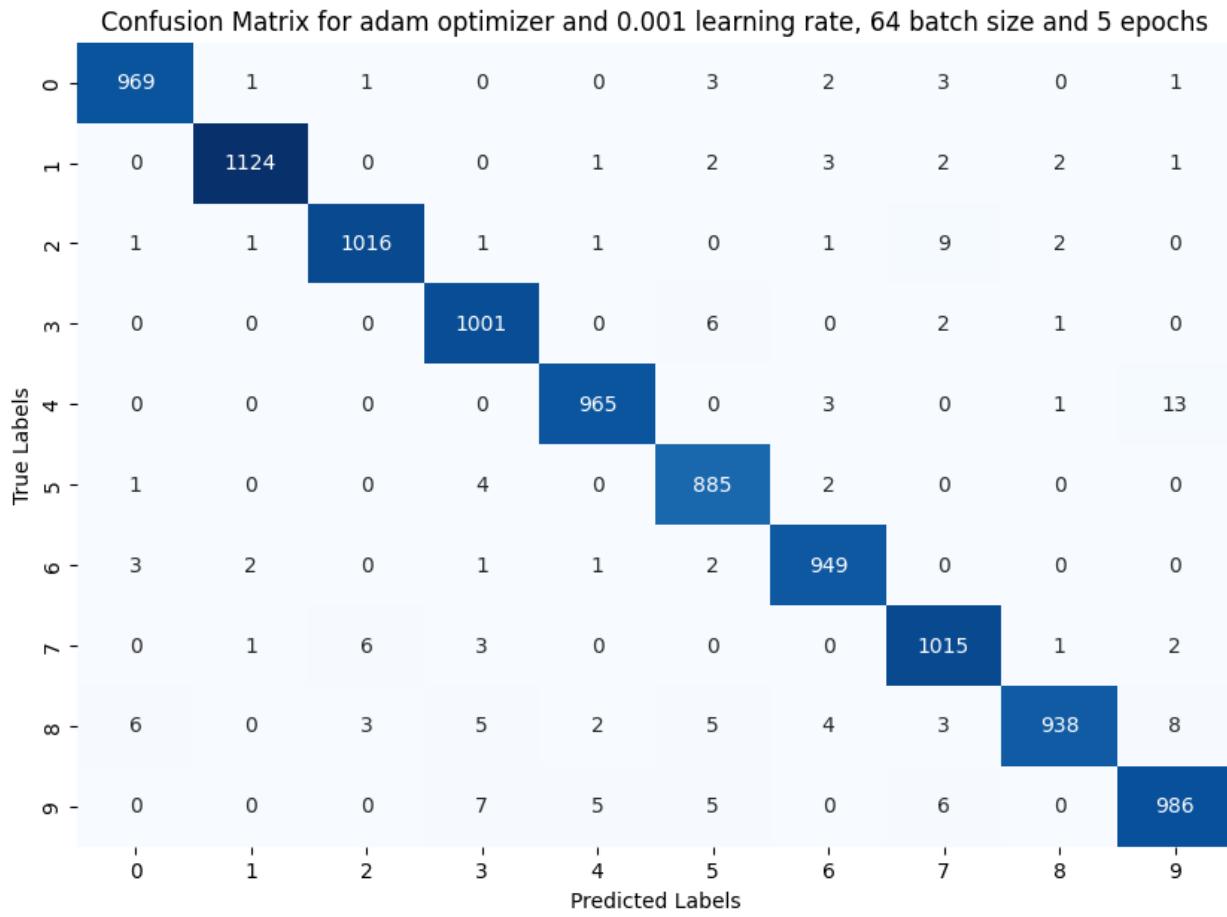
Precision: 0.9807

Recall: 0.9806



```
Training with adam optimizer and the learning_rate is 0.001, 64 batch size and 5 epochs...
Epoch 1/5
844/844 - 6s - loss: 0.1595 - accuracy: 0.9514 - val_loss: 0.0633 -
val_accuracy: 0.9813 - 6s/epoch - 7ms/step
Epoch 2/5
844/844 - 5s - loss: 0.0522 - accuracy: 0.9837 - val_loss: 0.0517 -
val_accuracy: 0.9850 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 0.0323 - accuracy: 0.9902 - val_loss: 0.0491 -
val_accuracy: 0.9863 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 0.0217 - accuracy: 0.9932 - val_loss: 0.0516 -
val_accuracy: 0.9865 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.0151 - accuracy: 0.9952 - val_loss: 0.0526 -
val_accuracy: 0.9860 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.001, 64 batch size and 5 epochs:
[[ 969    1    1    0    0    3    2    3    0    1]
 [  0 1124    0    0    1    2    3    2    2    1]
 [  1    1 1016    1    1    0    1    9    2    0]
 [  0    0    0 1001    0    6    0    2    1    0]
 [  0    0    0    0  965    0    3    0    1   13]
 [  1    0    0    4    0  885    2    0    0    0]
 [  3    2    0    1    1    2  949    0    0    0]
 [  0    1    6    3    0    0    0 1015    1    2]
 [  6    0    3    5    2    5    4    3  938    8]
 [  0    0    0    7    5    5    0    6    0  986]]
```

Precision: 0.9849
Recall: 0.9848



```
Training with adam optimizer and the learning_rate is 0.001, 64 batch size and 15 epochs...
Epoch 1/15
844/844 - 6s - loss: 0.1604 - accuracy: 0.9509 - val_loss: 0.0678 -
val_accuracy: 0.9803 - 6s/epoch - 7ms/step
Epoch 2/15
844/844 - 5s - loss: 0.0539 - accuracy: 0.9830 - val_loss: 0.0519 -
val_accuracy: 0.9855 - 5s/epoch - 6ms/step
Epoch 3/15
844/844 - 5s - loss: 0.0345 - accuracy: 0.9892 - val_loss: 0.0472 -
val_accuracy: 0.9887 - 5s/epoch - 6ms/step
Epoch 4/15
844/844 - 5s - loss: 0.0208 - accuracy: 0.9933 - val_loss: 0.0448 -
val_accuracy: 0.9875 - 5s/epoch - 6ms/step
Epoch 5/15
844/844 - 4s - loss: 0.0151 - accuracy: 0.9954 - val_loss: 0.0478 -
val_accuracy: 0.9878 - 4s/epoch - 5ms/step
Epoch 6/15
844/844 - 4s - loss: 0.0108 - accuracy: 0.9965 - val_loss: 0.0441 -
val_accuracy: 0.9890 - 4s/epoch - 5ms/step
Epoch 7/15
```

```
844/844 - 5s - loss: 0.0110 - accuracy: 0.9964 - val_loss: 0.0627 -  
val_accuracy: 0.9852 - 5s/epoch - 5ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.0053 - accuracy: 0.9986 - val_loss: 0.0502 -  
val_accuracy: 0.9880 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.0078 - accuracy: 0.9974 - val_loss: 0.0676 -  
val_accuracy: 0.9867 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.0049 - accuracy: 0.9985 - val_loss: 0.0636 -  
val_accuracy: 0.9878 - 5s/epoch - 5ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.0034 - accuracy: 0.9991 - val_loss: 0.0602 -  
val_accuracy: 0.9873 - 5s/epoch - 5ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.0057 - accuracy: 0.9981 - val_loss: 0.0593 -  
val_accuracy: 0.9890 - 5s/epoch - 5ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.0032 - accuracy: 0.9990 - val_loss: 0.0640 -  
val_accuracy: 0.9880 - 5s/epoch - 5ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.0040 - accuracy: 0.9986 - val_loss: 0.0600 -  
val_accuracy: 0.9875 - 5s/epoch - 5ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.0021 - accuracy: 0.9993 - val_loss: 0.0682 -  
val_accuracy: 0.9878 - 5s/epoch - 5ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.001, 64  
batch size and 15 epochs:  
[[ 970 1 2 0 1 0 3 1 1 1 ]  
[ 0 1125 3 3 0 0 1 1 2 0 ]  
[ 1 2 1019 0 2 0 1 6 0 1 ]  
[ 0 0 3 1002 1 2 0 2 0 0 ]  
[ 0 0 0 0 973 0 1 0 0 8 ]  
[ 1 0 1 19 0 865 4 0 1 1 ]  
[ 4 1 1 1 4 3 941 0 3 0 ]  
[ 0 1 8 0 1 0 0 1011 2 5 ]  
[ 2 1 2 6 1 0 0 2 956 4 ]  
[ 1 2 2 6 6 3 0 4 2 983 ]]  
Precision: 0.9846  
Recall: 0.9845
```

Confusion Matrix for adam optimizer and 0.001 learning rate, 64 batch size and 15 epochs										
	0	1	2	3	4	5	6	7	8	9
0	970	1	2	0	1	0	3	1	1	1
1	0	1125	3	3	0	0	1	1	2	0
2	1	2	1019	0	2	0	1	6	0	1
3	0	0	3	1002	1	2	0	2	0	0
4	0	0	0	0	973	0	1	0	0	8
5	1	0	1	19	0	865	4	0	1	1
6	4	1	1	1	4	3	941	0	3	0
7	0	1	8	0	1	0	0	1011	2	5
8	2	1	2	6	1	0	0	2	956	4
9	1	2	2	6	6	3	0	4	2	983
	0	1	2	3	4	5	6	7	8	9
True Labels	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

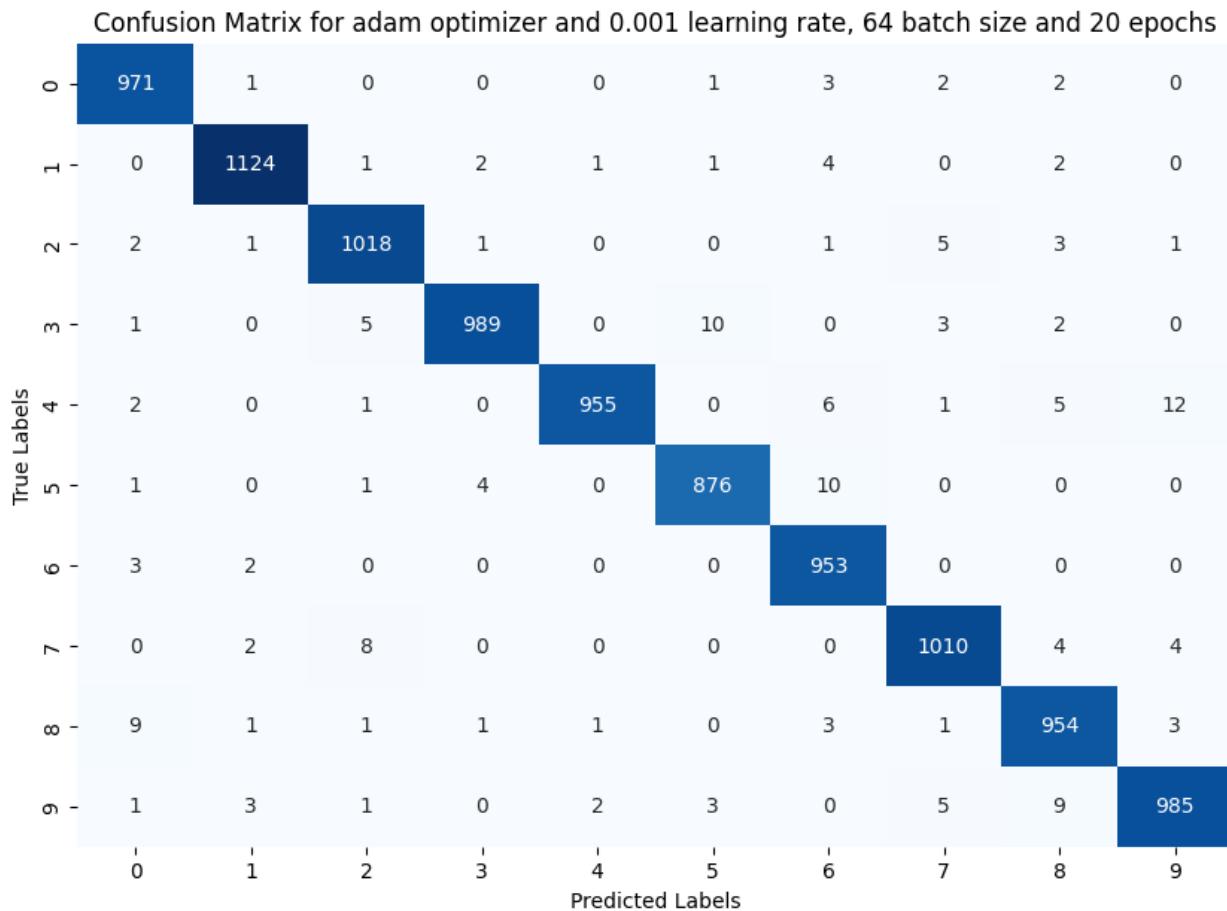
```
Training with adam optimizer and the learning_rate is 0.001, 64 batch size and 20 epochs...
Epoch 1/20
844/844 - 5s - loss: 0.1690 - accuracy: 0.9481 - val_loss: 0.0644 -
val_accuracy: 0.9807 - 5s/epoch - 6ms/step
Epoch 2/20
844/844 - 5s - loss: 0.0534 - accuracy: 0.9834 - val_loss: 0.0605 -
val_accuracy: 0.9833 - 5s/epoch - 5ms/step
Epoch 3/20
844/844 - 5s - loss: 0.0333 - accuracy: 0.9895 - val_loss: 0.0509 -
val_accuracy: 0.9875 - 5s/epoch - 5ms/step
Epoch 4/20
844/844 - 5s - loss: 0.0211 - accuracy: 0.9933 - val_loss: 0.0512 -
val_accuracy: 0.9865 - 5s/epoch - 5ms/step
Epoch 5/20
844/844 - 5s - loss: 0.0168 - accuracy: 0.9948 - val_loss: 0.0548 -
val_accuracy: 0.9865 - 5s/epoch - 5ms/step
Epoch 6/20
844/844 - 5s - loss: 0.0118 - accuracy: 0.9960 - val_loss: 0.0604 -
val_accuracy: 0.9875 - 5s/epoch - 5ms/step
Epoch 7/20
```

```
844/844 - 4s - loss: 0.0094 - accuracy: 0.9972 - val_loss: 0.0571 -  
val_accuracy: 0.9868 - 4s/epoch - 5ms/step  
Epoch 8/20  
844/844 - 4s - loss: 0.0067 - accuracy: 0.9980 - val_loss: 0.0565 -  
val_accuracy: 0.9883 - 4s/epoch - 5ms/step  
Epoch 9/20  
844/844 - 5s - loss: 0.0056 - accuracy: 0.9984 - val_loss: 0.0597 -  
val_accuracy: 0.9875 - 5s/epoch - 5ms/step  
Epoch 10/20  
844/844 - 5s - loss: 0.0059 - accuracy: 0.9981 - val_loss: 0.0719 -  
val_accuracy: 0.9852 - 5s/epoch - 5ms/step  
Epoch 11/20  
844/844 - 5s - loss: 0.0033 - accuracy: 0.9991 - val_loss: 0.0797 -  
val_accuracy: 0.9858 - 5s/epoch - 5ms/step  
Epoch 12/20  
844/844 - 5s - loss: 0.0054 - accuracy: 0.9982 - val_loss: 0.0660 -  
val_accuracy: 0.9875 - 5s/epoch - 5ms/step  
Epoch 13/20  
844/844 - 5s - loss: 0.0024 - accuracy: 0.9994 - val_loss: 0.0682 -  
val_accuracy: 0.9883 - 5s/epoch - 5ms/step  
Epoch 14/20  
844/844 - 5s - loss: 0.0017 - accuracy: 0.9995 - val_loss: 0.0709 -  
val_accuracy: 0.9882 - 5s/epoch - 5ms/step  
Epoch 15/20  
844/844 - 5s - loss: 0.0041 - accuracy: 0.9987 - val_loss: 0.0731 -  
val_accuracy: 0.9872 - 5s/epoch - 5ms/step  
Epoch 16/20  
844/844 - 5s - loss: 0.0030 - accuracy: 0.9991 - val_loss: 0.0830 -  
val_accuracy: 0.9877 - 5s/epoch - 5ms/step  
Epoch 17/20  
844/844 - 5s - loss: 0.0035 - accuracy: 0.9988 - val_loss: 0.0633 -  
val_accuracy: 0.9902 - 5s/epoch - 5ms/step  
Epoch 18/20  
844/844 - 5s - loss: 0.0021 - accuracy: 0.9995 - val_loss: 0.0810 -  
val_accuracy: 0.9880 - 5s/epoch - 5ms/step  
Epoch 19/20  
844/844 - 5s - loss: 5.7770e-04 - accuracy: 0.9998 - val_loss: 0.0696  
- val_accuracy: 0.9885 - 5s/epoch - 5ms/step  
Epoch 20/20  
844/844 - 5s - loss: 0.0020 - accuracy: 0.9992 - val_loss: 0.0929 -  
val_accuracy: 0.9863 - 5s/epoch - 5ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.001, 64  
batch size and 20 epochs:  
[[ 971 1 0 0 0 1 3 2 2 0]  
 [ 0 1124 1 2 1 1 4 0 2 0]  
 [ 2 1 1018 1 0 0 1 5 3 1]  
 [ 1 0 5 989 0 10 0 3 2 0]  
 [ 2 0 1 0 955 0 6 1 5 12]]
```

```
[ 1 0 1 4 0 876 10 0 0 0]
[ 3 2 0 0 0 953 0 0 0 0]
[ 0 2 8 0 0 0 1010 4 4 4]
[ 9 1 1 1 1 0 3 1 954 3]
[ 1 3 1 0 2 3 0 5 9 985]]
```

Precision: 0.9836

Recall: 0.9835



Training with adam optimizer and the learning_rate is 0.001, 128 batch size and 5 epochs...

Epoch 1/5

422/422 - 4s - loss: 0.1908 - accuracy: 0.9414 - val_loss: 0.0864 - val_accuracy: 0.9757 - 4s/epoch - 10ms/step

Epoch 2/5

422/422 - 3s - loss: 0.0620 - accuracy: 0.9811 - val_loss: 0.0578 - val_accuracy: 0.9848 - 3s/epoch - 7ms/step

Epoch 3/5

422/422 - 3s - loss: 0.0368 - accuracy: 0.9888 - val_loss: 0.0524 - val_accuracy: 0.9855 - 3s/epoch - 7ms/step

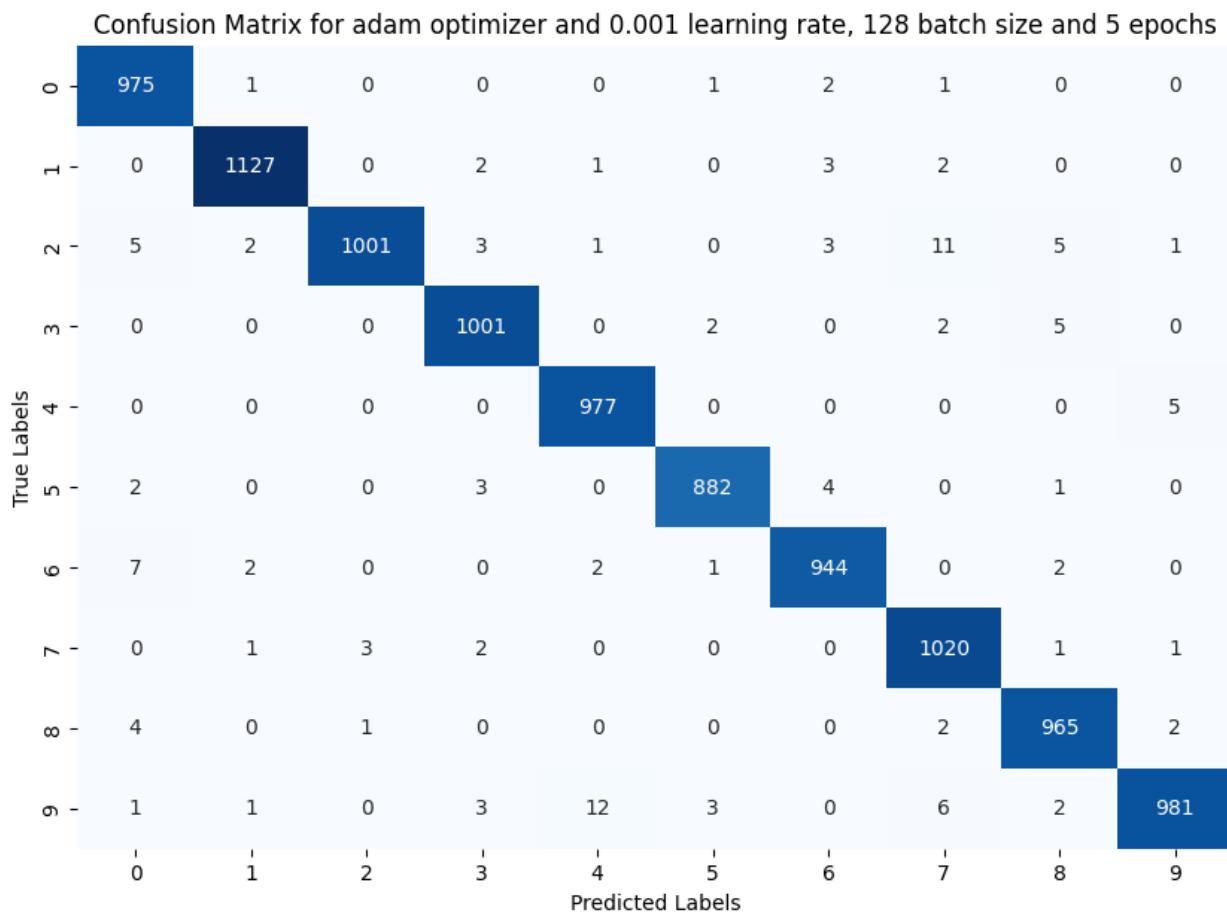
Epoch 4/5

422/422 - 3s - loss: 0.0258 - accuracy: 0.9920 - val_loss: 0.0545 -

```

val_accuracy: 0.9858 - 3s/epoch - 7ms/step
Epoch 5/5
422/422 - 3s - loss: 0.0177 - accuracy: 0.9944 - val_loss: 0.0484 -
val_accuracy: 0.9883 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.001, 128
batch size and 5 epochs:
[[ 975   1   0   0   0   1   2   1   0   0]
 [  0 1127   0   2   1   0   3   2   0   0]
 [  5   2 1001   3   1   0   3   11   5   1]
 [  0   0   0 1001   0   2   0   2   5   0]
 [  0   0   0   0 977   0   0   0   0   5]
 [  2   0   0   3   0 882   4   0   1   0]
 [  7   2   0   0   2   1 944   0   2   0]
 [  0   1   3   2   0   0   0 1020   1   1]
 [  4   0   1   0   0   0   0   2 965   2]
 [  1   1   0   3 12   3   0   6   2 981]]
Precision: 0.9874
Recall: 0.9873

```

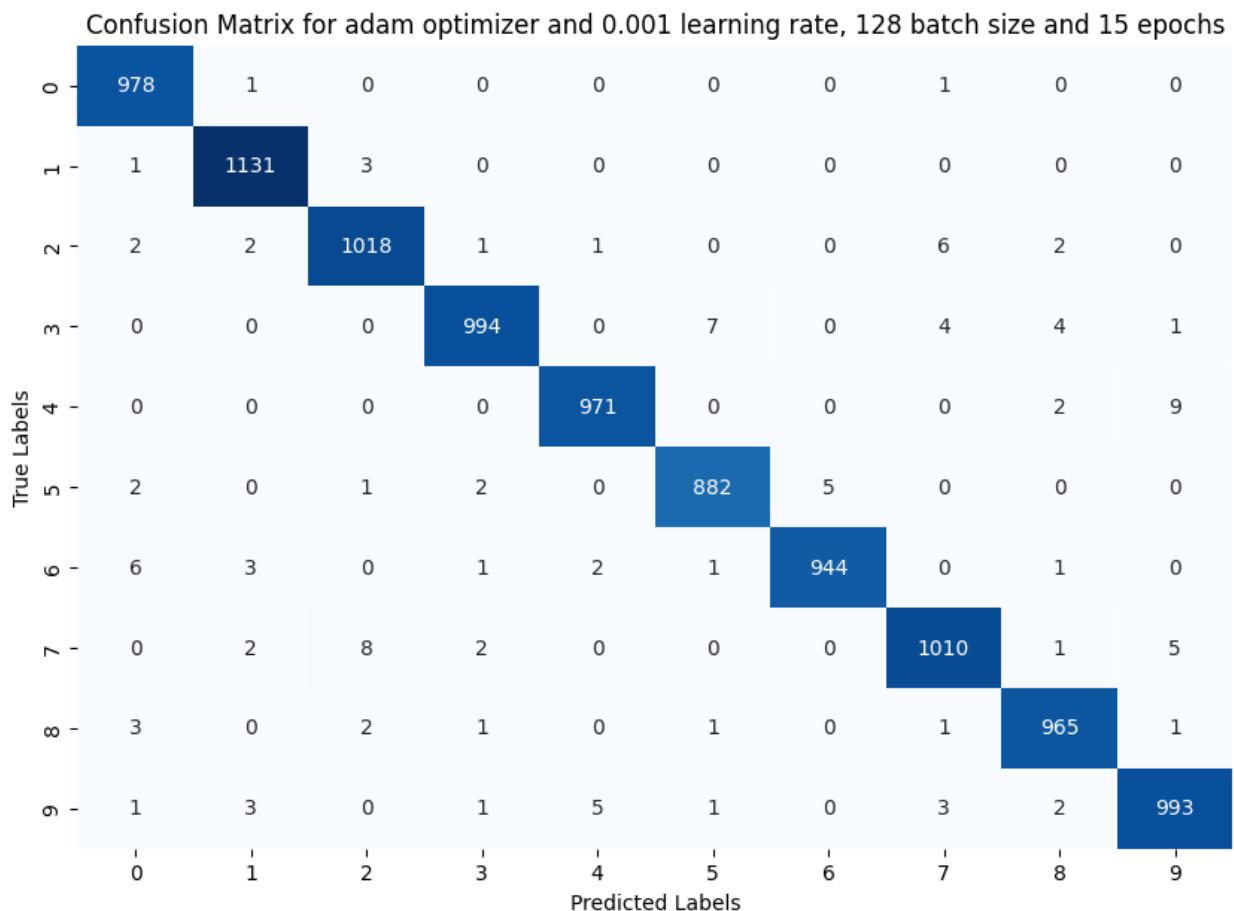


```
Training with adam optimizer and the learning_rate is 0.001, 128 batch size and 15 epochs...
Epoch 1/15
422/422 - 4s - loss: 0.1834 - accuracy: 0.9443 - val_loss: 0.0700 -
val_accuracy: 0.9798 - 4s/epoch - 9ms/step
Epoch 2/15
422/422 - 3s - loss: 0.0604 - accuracy: 0.9815 - val_loss: 0.0666 -
val_accuracy: 0.9823 - 3s/epoch - 7ms/step
Epoch 3/15
422/422 - 3s - loss: 0.0366 - accuracy: 0.9891 - val_loss: 0.0517 -
val_accuracy: 0.9865 - 3s/epoch - 7ms/step
Epoch 4/15
422/422 - 3s - loss: 0.0235 - accuracy: 0.9931 - val_loss: 0.0487 -
val_accuracy: 0.9870 - 3s/epoch - 7ms/step
Epoch 5/15
422/422 - 3s - loss: 0.0173 - accuracy: 0.9948 - val_loss: 0.0576 -
val_accuracy: 0.9857 - 3s/epoch - 7ms/step
Epoch 6/15
422/422 - 3s - loss: 0.0114 - accuracy: 0.9964 - val_loss: 0.0530 -
val_accuracy: 0.9870 - 3s/epoch - 8ms/step
Epoch 7/15
422/422 - 3s - loss: 0.0091 - accuracy: 0.9974 - val_loss: 0.0690 -
val_accuracy: 0.9837 - 3s/epoch - 8ms/step
Epoch 8/15
422/422 - 3s - loss: 0.0080 - accuracy: 0.9978 - val_loss: 0.0603 -
val_accuracy: 0.9863 - 3s/epoch - 8ms/step
Epoch 9/15
422/422 - 3s - loss: 0.0068 - accuracy: 0.9980 - val_loss: 0.0607 -
val_accuracy: 0.9858 - 3s/epoch - 7ms/step
Epoch 10/15
422/422 - 3s - loss: 0.0041 - accuracy: 0.9988 - val_loss: 0.0589 -
val_accuracy: 0.9887 - 3s/epoch - 7ms/step
Epoch 11/15
422/422 - 3s - loss: 0.0027 - accuracy: 0.9994 - val_loss: 0.0685 -
val_accuracy: 0.9860 - 3s/epoch - 7ms/step
Epoch 12/15
422/422 - 3s - loss: 0.0055 - accuracy: 0.9982 - val_loss: 0.0698 -
val_accuracy: 0.9872 - 3s/epoch - 7ms/step
Epoch 13/15
422/422 - 3s - loss: 0.0045 - accuracy: 0.9985 - val_loss: 0.0683 -
val_accuracy: 0.9880 - 3s/epoch - 7ms/step
Epoch 14/15
422/422 - 3s - loss: 0.0056 - accuracy: 0.9982 - val_loss: 0.0676 -
val_accuracy: 0.9867 - 3s/epoch - 7ms/step
Epoch 15/15
422/422 - 3s - loss: 0.0030 - accuracy: 0.9990 - val_loss: 0.0660 -
val_accuracy: 0.9882 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.001, 128 batch size and 15 epochs:
```

```
[[ 978   1   0   0   0   0   0   1   0   0]
 [ 1 1131   3   0   0   0   0   0   0   0]
 [ 2   2 1018   1   1   0   0   6   2   0]
 [ 0   0   0 994   0   7   0   4   4   1]
 [ 0   0   0   0 971   0   0   0   2   9]
 [ 2   0   1   2   0 882   5   0   0   0]
 [ 6   3   0   1   2   1 944   0   1   0]
 [ 0   2   8   2   0   0   0 1010   1   5]
 [ 3   0   2   1   0   1   0   1 965   1]
 [ 1   3   0   1   5   1   0   3   2 993]]
```

Precision: 0.9886

Recall: 0.9886



Training with adam optimizer and the learning_rate is 0.001, 128 batch size and 20 epochs...

Epoch 1/20

422/422 - 4s - loss: 0.1956 - accuracy: 0.9410 - val_loss: 0.0708 - val_accuracy: 0.9788 - 4s/epoch - 9ms/step

Epoch 2/20

422/422 - 3s - loss: 0.0661 - accuracy: 0.9800 - val_loss: 0.0540 - val_accuracy: 0.9843 - 3s/epoch - 7ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.0418 - accuracy: 0.9874 - val_loss: 0.0535 -
val_accuracy: 0.9867 - 3s/epoch - 7ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0297 - accuracy: 0.9908 - val_loss: 0.0453 -
val_accuracy: 0.9888 - 3s/epoch - 8ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0197 - accuracy: 0.9942 - val_loss: 0.0434 -
val_accuracy: 0.9878 - 3s/epoch - 8ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0151 - accuracy: 0.9955 - val_loss: 0.0499 -
val_accuracy: 0.9873 - 3s/epoch - 7ms/step
Epoch 7/20
422/422 - 3s - loss: 0.0111 - accuracy: 0.9968 - val_loss: 0.0513 -
val_accuracy: 0.9872 - 3s/epoch - 7ms/step
Epoch 8/20
422/422 - 3s - loss: 0.0094 - accuracy: 0.9973 - val_loss: 0.0451 -
val_accuracy: 0.9892 - 3s/epoch - 7ms/step
Epoch 9/20
422/422 - 3s - loss: 0.0074 - accuracy: 0.9978 - val_loss: 0.0482 -
val_accuracy: 0.9887 - 3s/epoch - 7ms/step
Epoch 10/20
422/422 - 3s - loss: 0.0050 - accuracy: 0.9985 - val_loss: 0.0529 -
val_accuracy: 0.9873 - 3s/epoch - 7ms/step
Epoch 11/20
422/422 - 3s - loss: 0.0054 - accuracy: 0.9985 - val_loss: 0.0526 -
val_accuracy: 0.9873 - 3s/epoch - 7ms/step
Epoch 12/20
422/422 - 3s - loss: 0.0035 - accuracy: 0.9991 - val_loss: 0.0583 -
val_accuracy: 0.9867 - 3s/epoch - 7ms/step
Epoch 13/20
422/422 - 3s - loss: 0.0038 - accuracy: 0.9990 - val_loss: 0.0575 -
val_accuracy: 0.9870 - 3s/epoch - 7ms/step
Epoch 14/20
422/422 - 3s - loss: 0.0071 - accuracy: 0.9977 - val_loss: 0.0579 -
val_accuracy: 0.9887 - 3s/epoch - 7ms/step
Epoch 15/20
422/422 - 3s - loss: 0.0034 - accuracy: 0.9991 - val_loss: 0.0663 -
val_accuracy: 0.9870 - 3s/epoch - 7ms/step
Epoch 16/20
422/422 - 3s - loss: 0.0013 - accuracy: 0.9997 - val_loss: 0.0574 -
val_accuracy: 0.9907 - 3s/epoch - 7ms/step
Epoch 17/20
422/422 - 3s - loss: 0.0012 - accuracy: 0.9998 - val_loss: 0.0663 -
val_accuracy: 0.9882 - 3s/epoch - 7ms/step
Epoch 18/20
422/422 - 3s - loss: 0.0060 - accuracy: 0.9981 - val_loss: 0.0736 -
val_accuracy: 0.9868 - 3s/epoch - 7ms/step
Epoch 19/20
```

```
422/422 - 3s - loss: 0.0061 - accuracy: 0.9980 - val_loss: 0.0776 -  
val_accuracy: 0.9867 - 3s/epoch - 7ms/step
```

```
Epoch 20/20
```

```
422/422 - 3s - loss: 0.0036 - accuracy: 0.9988 - val_loss: 0.0648 -  
val_accuracy: 0.9878 - 3s/epoch - 7ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

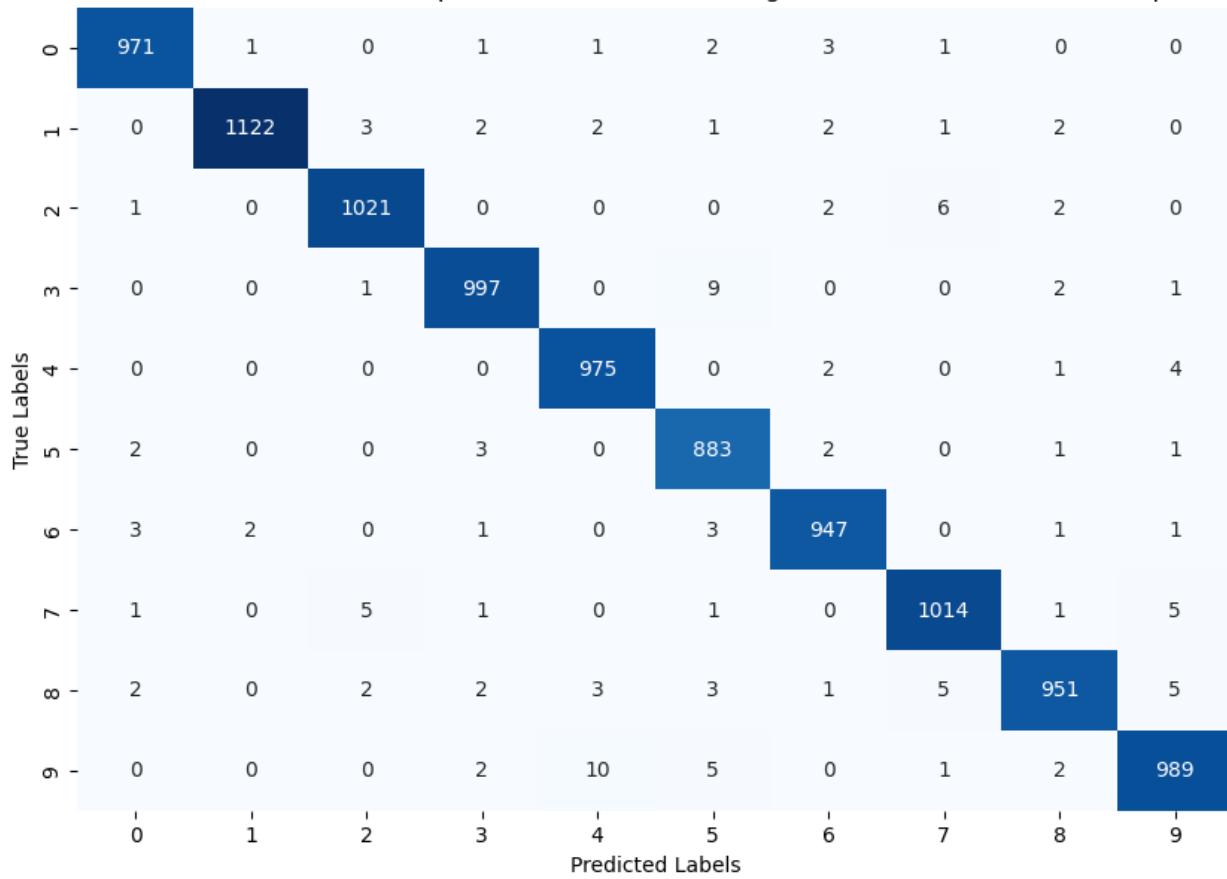
```
Confusion Matrix adam optimizer and the learning_rate is 0.001, 128  
batch size and 20 epochs:
```

```
[[ 971   1   0   1   1   2   3   1   0   0]  
 [ 0 1122   3   2   2   1   2   1   2   0]  
 [ 1   0 1021   0   0   0   2   6   2   0]  
 [ 0   0   1 997   0   9   0   0   2   1]  
 [ 0   0   0   0 975   0   2   0   1   4]  
 [ 2   0   0   3   0 883   2   0   1   1]  
 [ 3   2   0   1   0   3 947   0   1   1]  
 [ 1   0   5   1   0   1   0 1014   1   5]  
 [ 2   0   2   2   3   3   1   5 951   5]  
 [ 0   0   0   2   10  5   0   1   2 989]]
```

```
Precision: 0.9870
```

```
Recall: 0.9870
```

Confusion Matrix for adam optimizer and 0.001 learning rate, 128 batch size and 20 epochs



```
Training with adam optimizer and the learning_rate is 0.001, 256 batch size and 5 epochs...
Epoch 1/5
211/211 - 3s - loss: 0.2405 - accuracy: 0.9278 - val_loss: 0.0755 -
val_accuracy: 0.9800 - 3s/epoch - 15ms/step
Epoch 2/5
211/211 - 2s - loss: 0.0742 - accuracy: 0.9778 - val_loss: 0.0623 -
val_accuracy: 0.9830 - 2s/epoch - 12ms/step
Epoch 3/5
211/211 - 2s - loss: 0.0454 - accuracy: 0.9871 - val_loss: 0.0498 -
val_accuracy: 0.9863 - 2s/epoch - 12ms/step
Epoch 4/5
211/211 - 2s - loss: 0.0334 - accuracy: 0.9899 - val_loss: 0.0482 -
val_accuracy: 0.9870 - 2s/epoch - 12ms/step
Epoch 5/5
211/211 - 2s - loss: 0.0232 - accuracy: 0.9935 - val_loss: 0.0474 -
val_accuracy: 0.9875 - 2s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.001, 256 batch size and 5 epochs:
[[ 973    1    1    0    0    0    3    1    1    0]
 [  0 1130    2    0    0    0    1    1    1    0]
 [  0    2 1015    0    1    0    2    7    5    0]
 [  0    0    2  989    0    7    0    5    5    2]
 [  0    0    0    0  972    0    5    0    1    4]
 [  1    0    1    3    0  882    4    0    1    0]
 [  4    2    1    0    1    2  946    0    2    0]
 [  1    3    7    0    0    0    0 1012    2    3]
 [  2    0    2    2    0    1    0    2  964    1]
 [  2    4    0    1    9    4    0    4    6  979]]
```

Precision: 0.9862
Recall: 0.9862

Confusion Matrix for adam optimizer and 0.001 learning rate, 256 batch size and 5 epochs										
	0	1	2	3	4	5	6	7	8	9
True Labels	973	1	1	0	0	0	3	1	1	0
0	973	1	1	0	0	0	3	1	1	0
1	0	1130	2	0	0	0	1	1	1	0
2	0	2	1015	0	1	0	2	7	5	0
3	0	0	2	989	0	7	0	5	5	2
4	0	0	0	0	972	0	5	0	1	4
5	1	0	1	3	0	882	4	0	1	0
6	4	2	1	0	1	2	946	0	2	0
7	1	3	7	0	0	0	0	1012	2	3
8	2	0	2	2	0	1	0	2	964	1
9	2	4	0	1	9	4	0	4	6	979
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training with adam optimizer and the learning_rate is 0.001, 256 batch size and 15 epochs...
Epoch 1/15
211/211 - 3s - loss: 0.2176 - accuracy: 0.9350 - val_loss: 0.0813 - val_accuracy: 0.9775 - 3s/epoch - 15ms/step
Epoch 2/15
211/211 - 2s - loss: 0.0660 - accuracy: 0.9809 - val_loss: 0.0521 - val_accuracy: 0.9860 - 2s/epoch - 12ms/step
Epoch 3/15
211/211 - 2s - loss: 0.0427 - accuracy: 0.9873 - val_loss: 0.0539 - val_accuracy: 0.9853 - 2s/epoch - 12ms/step
Epoch 4/15
211/211 - 2s - loss: 0.0303 - accuracy: 0.9910 - val_loss: 0.0496 - val_accuracy: 0.9862 - 2s/epoch - 12ms/step
Epoch 5/15
211/211 - 2s - loss: 0.0212 - accuracy: 0.9935 - val_loss: 0.0465 - val_accuracy: 0.9880 - 2s/epoch - 12ms/step
Epoch 6/15
211/211 - 2s - loss: 0.0151 - accuracy: 0.9957 - val_loss: 0.0533 - val_accuracy: 0.9878 - 2s/epoch - 12ms/step
Epoch 7/15
```

```
211/211 - 2s - loss: 0.0119 - accuracy: 0.9968 - val_loss: 0.0448 -  
val_accuracy: 0.9885 - 2s/epoch - 12ms/step  
Epoch 8/15  
211/211 - 2s - loss: 0.0082 - accuracy: 0.9980 - val_loss: 0.0482 -  
val_accuracy: 0.9900 - 2s/epoch - 12ms/step  
Epoch 9/15  
211/211 - 2s - loss: 0.0065 - accuracy: 0.9987 - val_loss: 0.0468 -  
val_accuracy: 0.9893 - 2s/epoch - 11ms/step  
Epoch 10/15  
211/211 - 2s - loss: 0.0050 - accuracy: 0.9990 - val_loss: 0.0476 -  
val_accuracy: 0.9898 - 2s/epoch - 12ms/step  
Epoch 11/15  
211/211 - 2s - loss: 0.0038 - accuracy: 0.9994 - val_loss: 0.0482 -  
val_accuracy: 0.9887 - 2s/epoch - 11ms/step  
Epoch 12/15  
211/211 - 2s - loss: 0.0037 - accuracy: 0.9993 - val_loss: 0.0572 -  
val_accuracy: 0.9878 - 2s/epoch - 11ms/step  
Epoch 13/15  
211/211 - 2s - loss: 0.0017 - accuracy: 0.9998 - val_loss: 0.0508 -  
val_accuracy: 0.9907 - 2s/epoch - 11ms/step  
Epoch 14/15  
211/211 - 2s - loss: 9.6274e-04 - accuracy: 1.0000 - val_loss: 0.0548  
- val_accuracy: 0.9880 - 2s/epoch - 11ms/step  
Epoch 15/15  
211/211 - 2s - loss: 6.0453e-04 - accuracy: 1.0000 - val_loss: 0.0529  
- val_accuracy: 0.9900 - 2s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adam optimizer and the learning_rate is 0.001, 256  
batch size and 15 epochs:  
[[ 977  0  1  0  0  0  1  1  0  0]  
[  0 1129  1  1  0  0  3  0  1  0]  
[  4  2 1016  0  0  0  1  7  1  1]  
[  0  0  0 1000  0  4  0  2  4  0]  
[  0  0  0  0 973  0  3  0  1  5]  
[  2  0  0   8  0 878  3  0  1  0]  
[  7  2  1   0  1  3 944  0  0  0]  
[  0  1  7   3  0  0  0 1015  1  1]  
[  6  0  4   1  0  0  1  3 957  2]  
[  1  3  0   3  9  3  0  4  2 984]]  
Precision: 0.9873  
Recall: 0.9873
```

Confusion Matrix for adam optimizer and 0.001 learning rate, 256 batch size and 15 epochs										
	0	1	2	3	4	5	6	7	8	9
0	977	0	1	0	0	0	1	1	0	0
1	0	1129	1	1	0	0	3	0	1	0
2	4	2	1016	0	0	0	1	7	1	1
3	0	0	0	1000	0	4	0	2	4	0
4	0	0	0	0	973	0	3	0	1	5
5	2	0	0	8	0	878	3	0	1	0
6	7	2	1	0	1	3	944	0	0	0
7	0	1	7	3	0	0	0	1015	1	1
8	6	0	4	1	0	0	1	3	957	2
9	1	3	0	3	9	3	0	4	2	984
	0	1	2	3	4	5	6	7	8	9
	Predicted Labels									

```

Training with adam optimizer and the learning_rate is 0.001, 256 batch
size and 20 epochs...
Epoch 1/20
211/211 - 3s - loss: 0.2228 - accuracy: 0.9330 - val_loss: 0.0812 -
val_accuracy: 0.9772 - 3s/epoch - 15ms/step
Epoch 2/20
211/211 - 2s - loss: 0.0701 - accuracy: 0.9798 - val_loss: 0.0587 -
val_accuracy: 0.9823 - 2s/epoch - 11ms/step
Epoch 3/20
211/211 - 2s - loss: 0.0429 - accuracy: 0.9879 - val_loss: 0.0513 -
val_accuracy: 0.9870 - 2s/epoch - 11ms/step
Epoch 4/20
211/211 - 2s - loss: 0.0299 - accuracy: 0.9918 - val_loss: 0.0464 -
val_accuracy: 0.9880 - 2s/epoch - 11ms/step
Epoch 5/20
211/211 - 2s - loss: 0.0225 - accuracy: 0.9937 - val_loss: 0.0490 -
val_accuracy: 0.9865 - 2s/epoch - 11ms/step
Epoch 6/20
211/211 - 2s - loss: 0.0155 - accuracy: 0.9957 - val_loss: 0.0450 -
val_accuracy: 0.9877 - 2s/epoch - 11ms/step
Epoch 7/20

```

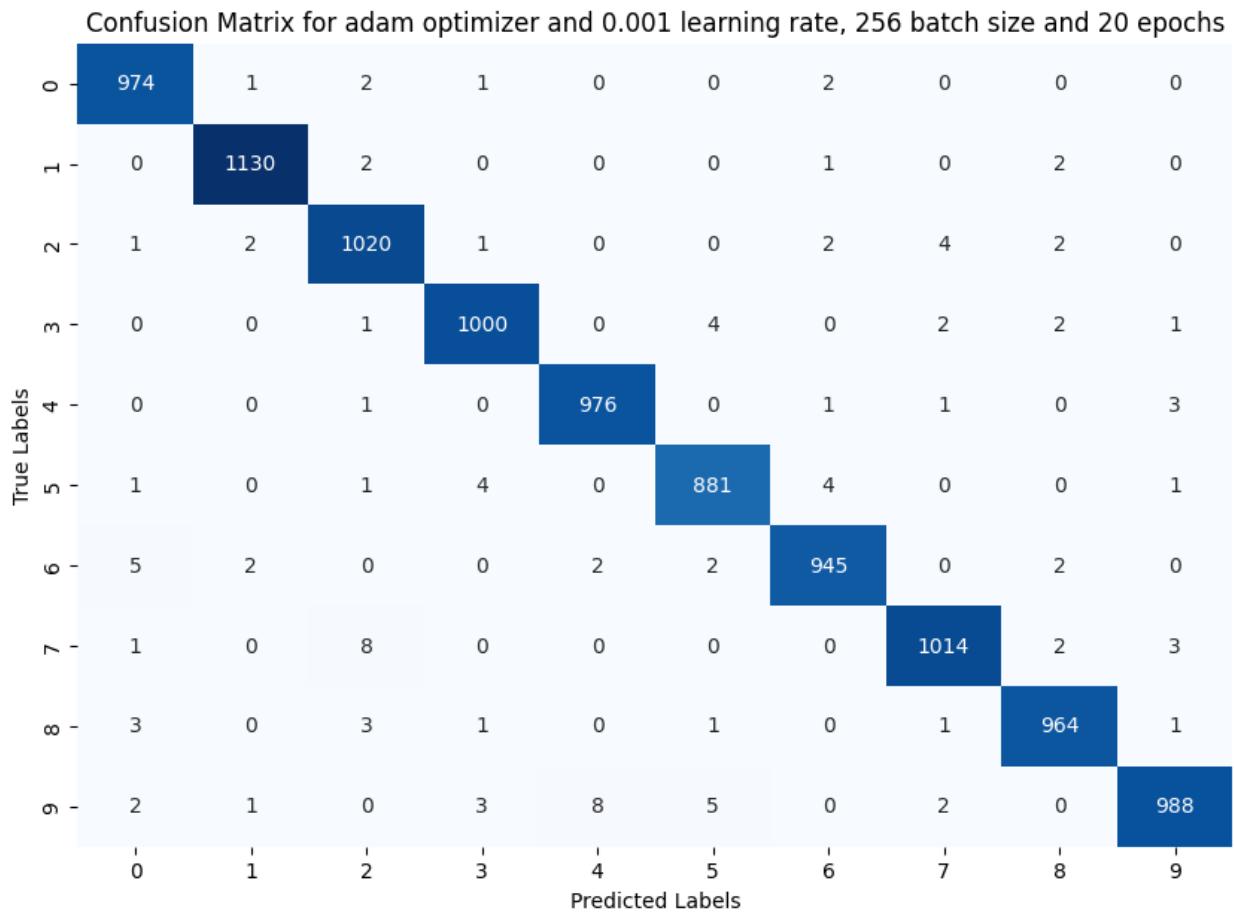
```

211/211 - 2s - loss: 0.0127 - accuracy: 0.9966 - val_loss: 0.0431 -
val_accuracy: 0.9890 - 2s/epoch - 12ms/step
Epoch 8/20
211/211 - 2s - loss: 0.0085 - accuracy: 0.9980 - val_loss: 0.0395 -
val_accuracy: 0.9903 - 2s/epoch - 12ms/step
Epoch 9/20
211/211 - 3s - loss: 0.0065 - accuracy: 0.9984 - val_loss: 0.0424 -
val_accuracy: 0.9888 - 3s/epoch - 12ms/step
Epoch 10/20
211/211 - 3s - loss: 0.0049 - accuracy: 0.9991 - val_loss: 0.0439 -
val_accuracy: 0.9905 - 3s/epoch - 12ms/step
Epoch 11/20
211/211 - 2s - loss: 0.0033 - accuracy: 0.9994 - val_loss: 0.0460 -
val_accuracy: 0.9903 - 2s/epoch - 11ms/step
Epoch 12/20
211/211 - 2s - loss: 0.0021 - accuracy: 0.9998 - val_loss: 0.0423 -
val_accuracy: 0.9903 - 2s/epoch - 12ms/step
Epoch 13/20
211/211 - 2s - loss: 0.0018 - accuracy: 0.9998 - val_loss: 0.0467 -
val_accuracy: 0.9903 - 2s/epoch - 12ms/step
Epoch 14/20
211/211 - 2s - loss: 0.0013 - accuracy: 0.9999 - val_loss: 0.0484 -
val_accuracy: 0.9902 - 2s/epoch - 12ms/step
Epoch 15/20
211/211 - 2s - loss: 7.5103e-04 - accuracy: 1.0000 - val_loss: 0.0474
- val_accuracy: 0.9905 - 2s/epoch - 12ms/step
Epoch 16/20
211/211 - 2s - loss: 5.3845e-04 - accuracy: 1.0000 - val_loss: 0.0487
- val_accuracy: 0.9908 - 2s/epoch - 11ms/step
Epoch 17/20
211/211 - 2s - loss: 4.2885e-04 - accuracy: 1.0000 - val_loss: 0.0479
- val_accuracy: 0.9910 - 2s/epoch - 11ms/step
Epoch 18/20
211/211 - 2s - loss: 3.2959e-04 - accuracy: 1.0000 - val_loss: 0.0485
- val_accuracy: 0.9908 - 2s/epoch - 11ms/step
Epoch 19/20
211/211 - 2s - loss: 2.6287e-04 - accuracy: 1.0000 - val_loss: 0.0499
- val_accuracy: 0.9910 - 2s/epoch - 12ms/step
Epoch 20/20
211/211 - 2s - loss: 2.3193e-04 - accuracy: 1.0000 - val_loss: 0.0509
- val_accuracy: 0.9907 - 2s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.001, 256
batch size and 20 epochs:
[[ 974    1    2    1    0    0    2    0    0    0]
 [  0  1130    2    0    0    0    1    0    2    0]
 [  1    2  1020    1    0    0    2    4    2    0]
 [  0    0    1 1000    0    4    0    2    2    1]
 [  0    0    1    0   976    0    1    1    0    3]]
```

```
[ 1 0 1 4 0 881 4 0 0 1]
[ 5 2 0 0 2 2 945 0 2 0]
[ 1 0 8 0 0 0 0 1014 2 3]
[ 3 0 3 1 0 1 0 1 964 1]
[ 2 1 0 3 8 5 0 2 0 988]]
```

Precision: 0.9892

Recall: 0.9892



Training with adam optimizer and the learning_rate is 0.001, 16 batch size and 5 epochs...

Epoch 1/5

3375/3375 - 14s - loss: 0.1363 - accuracy: 0.9580 - val_loss: 0.0746 - val_accuracy: 0.9783 - 14s/epoch - 4ms/step

Epoch 2/5

3375/3375 - 13s - loss: 0.0493 - accuracy: 0.9848 - val_loss: 0.0574 - val_accuracy: 0.9823 - 13s/epoch - 4ms/step

Epoch 3/5

3375/3375 - 14s - loss: 0.0267 - accuracy: 0.9916 - val_loss: 0.0498 - val_accuracy: 0.9872 - 14s/epoch - 4ms/step

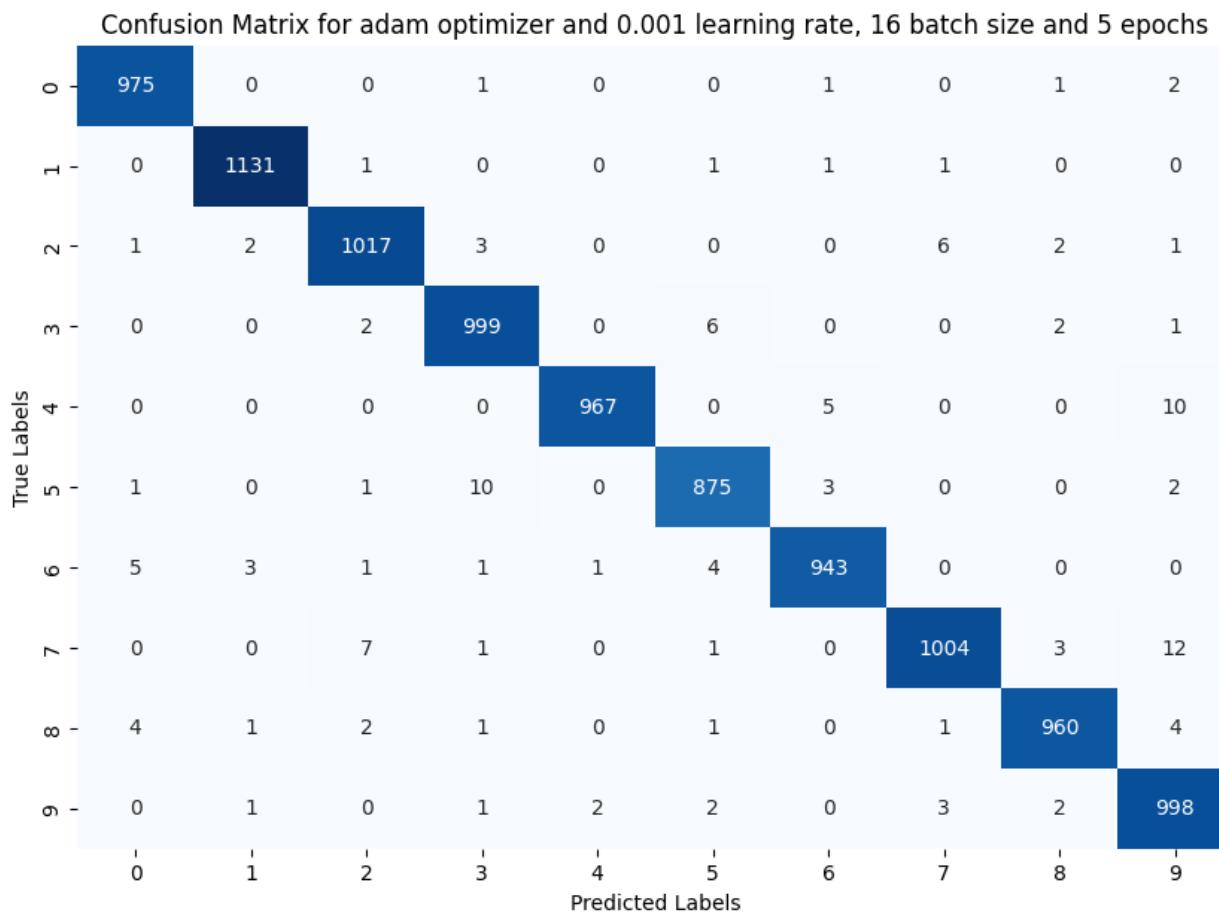
Epoch 4/5

3375/3375 - 14s - loss: 0.0184 - accuracy: 0.9939 - val_loss: 0.0542 -

```

val_accuracy: 0.9860 - 14s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 14s - loss: 0.0105 - accuracy: 0.9966 - val_loss: 0.0523 -
val_accuracy: 0.9893 - 14s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adam optimizer and the learning_rate is 0.001, 16
batch size and 5 epochs:
[[ 975 0 0 1 0 0 1 0 1 2]
 [ 0 1131 1 0 0 1 1 1 0 0]
 [ 1 2 1017 3 0 0 0 6 2 1]
 [ 0 0 2 999 0 6 0 0 2 1]
 [ 0 0 0 0 967 0 5 0 0 10]
 [ 1 0 1 10 0 875 3 0 0 2]
 [ 5 3 1 1 1 4 943 0 0 0]
 [ 0 0 7 1 0 1 0 1004 3 12]
 [ 4 1 2 1 0 1 0 1 960 4]
 [ 0 1 0 1 2 2 0 3 2 998]]
Precision: 0.9870
Recall: 0.9869

```



Training with adam optimizer and the learning_rate is 0.001, 16 batch size and 15 epochs...

Epoch 1/15
3375/3375 - 15s - loss: 0.1379 - accuracy: 0.9575 - val_loss: 0.0572 - val_accuracy: 0.9827 - 15s/epoch - 4ms/step

Epoch 2/15
3375/3375 - 14s - loss: 0.0473 - accuracy: 0.9849 - val_loss: 0.0537 - val_accuracy: 0.9868 - 14s/epoch - 4ms/step

Epoch 3/15
3375/3375 - 14s - loss: 0.0262 - accuracy: 0.9917 - val_loss: 0.0486 - val_accuracy: 0.9865 - 14s/epoch - 4ms/step

Epoch 4/15
3375/3375 - 14s - loss: 0.0161 - accuracy: 0.9945 - val_loss: 0.0609 - val_accuracy: 0.9818 - 14s/epoch - 4ms/step

Epoch 5/15
3375/3375 - 14s - loss: 0.0115 - accuracy: 0.9963 - val_loss: 0.0523 - val_accuracy: 0.9863 - 14s/epoch - 4ms/step

Epoch 6/15
3375/3375 - 14s - loss: 0.0075 - accuracy: 0.9975 - val_loss: 0.0619 - val_accuracy: 0.9868 - 14s/epoch - 4ms/step

Epoch 7/15
3375/3375 - 14s - loss: 0.0064 - accuracy: 0.9978 - val_loss: 0.0673 - val_accuracy: 0.9862 - 14s/epoch - 4ms/step

Epoch 8/15
3375/3375 - 14s - loss: 0.0054 - accuracy: 0.9982 - val_loss: 0.0639 - val_accuracy: 0.9880 - 14s/epoch - 4ms/step

Epoch 9/15
3375/3375 - 13s - loss: 0.0047 - accuracy: 0.9984 - val_loss: 0.0664 - val_accuracy: 0.9888 - 13s/epoch - 4ms/step

Epoch 10/15
3375/3375 - 14s - loss: 0.0028 - accuracy: 0.9990 - val_loss: 0.0780 - val_accuracy: 0.9887 - 14s/epoch - 4ms/step

Epoch 11/15
3375/3375 - 14s - loss: 0.0041 - accuracy: 0.9986 - val_loss: 0.0857 - val_accuracy: 0.9870 - 14s/epoch - 4ms/step

Epoch 12/15
3375/3375 - 14s - loss: 0.0035 - accuracy: 0.9989 - val_loss: 0.0756 - val_accuracy: 0.9883 - 14s/epoch - 4ms/step

Epoch 13/15
3375/3375 - 14s - loss: 0.0024 - accuracy: 0.9993 - val_loss: 0.1078 - val_accuracy: 0.9840 - 14s/epoch - 4ms/step

Epoch 14/15
3375/3375 - 14s - loss: 0.0020 - accuracy: 0.9994 - val_loss: 0.0754 - val_accuracy: 0.9903 - 14s/epoch - 4ms/step

Epoch 15/15
3375/3375 - 13s - loss: 0.0024 - accuracy: 0.9993 - val_loss: 0.0733 - val_accuracy: 0.9892 - 13s/epoch - 4ms/step

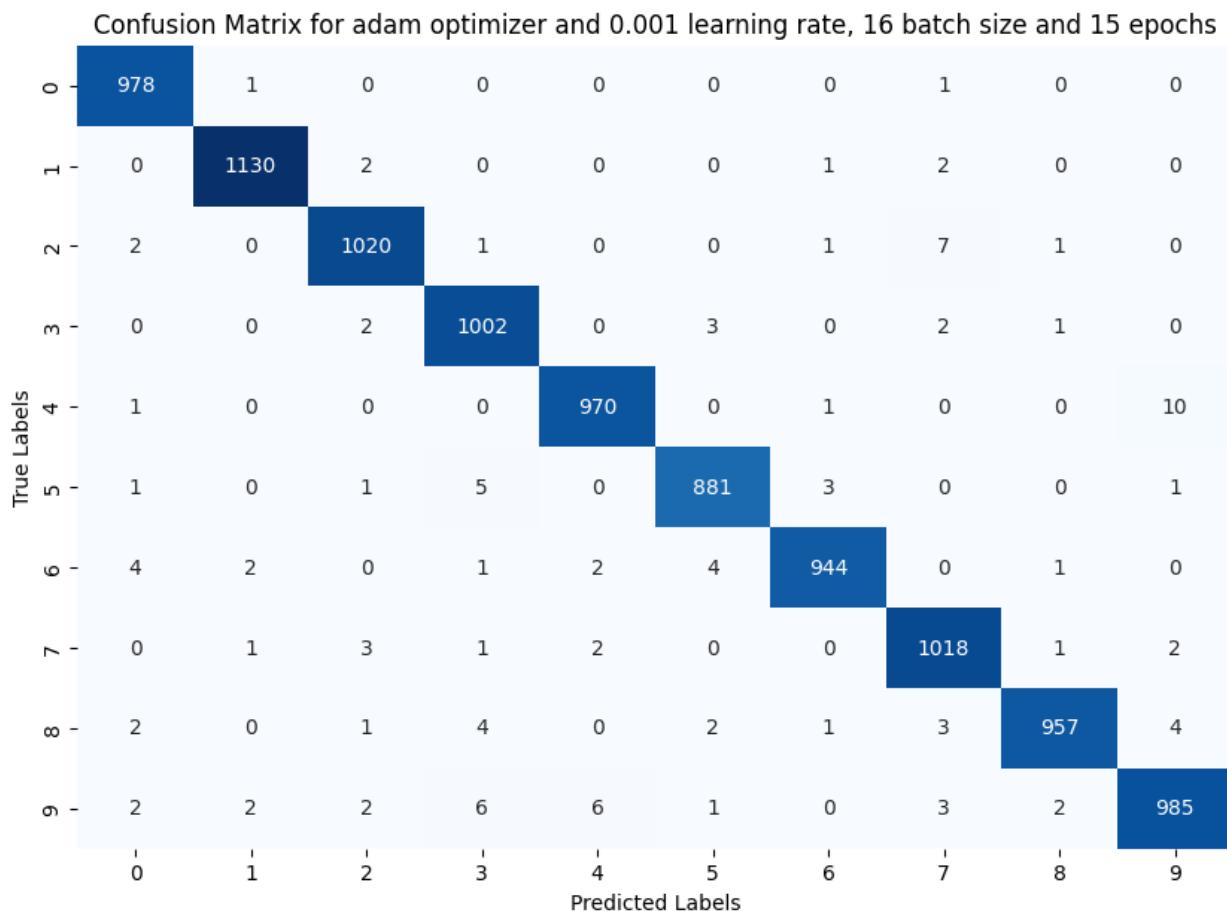
313/313 [=====] - 1s 2ms/step

Confusion Matrix adam optimizer and the learning_rate is 0.001, 16 batch size and 15 epochs:

```
[[ 978  1  0  0  0  0  0  1  0  0]
 [ 0 1130  2  0  0  0  1  2  0  0]
 [ 2  0 1020  1  0  0  1  7  1  0]
 [ 0  0  2 1002  0  3  0  2  1  0]
 [ 1  0  0  0  970  0  1  0  0 10]
 [ 1  0  1  5  0  881  3  0  0  1]
 [ 4  2  0  1  2  4  944  0  1  0]
 [ 0  1  3  1  2  0  0 1018  1  2]
 [ 2  0  1  4  0  2  1  3  957  4]
 [ 2  2  2  6  6  1  0  3  2  985]]
```

Precision: 0.9885

Recall: 0.9885



Training with adam optimizer and the learning_rate is 0.001, 16 batch size and 20 epochs...

Epoch 1/20

3375/3375 - 15s - loss: 0.1400 - accuracy: 0.9570 - val_loss: 0.0665 - val_accuracy: 0.9810 - 15s/epoch - 4ms/step

Epoch 2/20

3375/3375 - 14s - loss: 0.0491 - accuracy: 0.9847 - val_loss: 0.0449 - val_accuracy: 0.9882 - 14s/epoch - 4ms/step

```
Epoch 3/20
3375/3375 - 14s - loss: 0.0277 - accuracy: 0.9906 - val_loss: 0.0552 -
val_accuracy: 0.9845 - 14s/epoch - 4ms/step
Epoch 4/20
3375/3375 - 14s - loss: 0.0185 - accuracy: 0.9941 - val_loss: 0.0614 -
val_accuracy: 0.9872 - 14s/epoch - 4ms/step
Epoch 5/20
3375/3375 - 14s - loss: 0.0126 - accuracy: 0.9958 - val_loss: 0.0566 -
val_accuracy: 0.9865 - 14s/epoch - 4ms/step
Epoch 6/20
3375/3375 - 14s - loss: 0.0088 - accuracy: 0.9972 - val_loss: 0.0557 -
val_accuracy: 0.9872 - 14s/epoch - 4ms/step
Epoch 7/20
3375/3375 - 14s - loss: 0.0070 - accuracy: 0.9977 - val_loss: 0.0681 -
val_accuracy: 0.9870 - 14s/epoch - 4ms/step
Epoch 8/20
3375/3375 - 14s - loss: 0.0052 - accuracy: 0.9981 - val_loss: 0.0704 -
val_accuracy: 0.9848 - 14s/epoch - 4ms/step
Epoch 9/20
3375/3375 - 14s - loss: 0.0057 - accuracy: 0.9982 - val_loss: 0.0782 -
val_accuracy: 0.9848 - 14s/epoch - 4ms/step
Epoch 10/20
3375/3375 - 14s - loss: 0.0038 - accuracy: 0.9988 - val_loss: 0.0982 -
val_accuracy: 0.9850 - 14s/epoch - 4ms/step
Epoch 11/20
3375/3375 - 14s - loss: 0.0046 - accuracy: 0.9987 - val_loss: 0.0721 -
val_accuracy: 0.9863 - 14s/epoch - 4ms/step
Epoch 12/20
3375/3375 - 14s - loss: 0.0046 - accuracy: 0.9985 - val_loss: 0.0914 -
val_accuracy: 0.9848 - 14s/epoch - 4ms/step
Epoch 13/20
3375/3375 - 14s - loss: 0.0018 - accuracy: 0.9992 - val_loss: 0.0855 -
val_accuracy: 0.9885 - 14s/epoch - 4ms/step
Epoch 14/20
3375/3375 - 14s - loss: 0.0028 - accuracy: 0.9992 - val_loss: 0.0883 -
val_accuracy: 0.9890 - 14s/epoch - 4ms/step
Epoch 15/20
3375/3375 - 14s - loss: 0.0022 - accuracy: 0.9994 - val_loss: 0.0802 -
val_accuracy: 0.9898 - 14s/epoch - 4ms/step
Epoch 16/20
3375/3375 - 14s - loss: 0.0014 - accuracy: 0.9996 - val_loss: 0.0933 -
val_accuracy: 0.9870 - 14s/epoch - 4ms/step
Epoch 17/20
3375/3375 - 14s - loss: 0.0019 - accuracy: 0.9994 - val_loss: 0.0990 -
val_accuracy: 0.9885 - 14s/epoch - 4ms/step
Epoch 18/20
3375/3375 - 14s - loss: 0.0024 - accuracy: 0.9993 - val_loss: 0.0937 -
val_accuracy: 0.9887 - 14s/epoch - 4ms/step
Epoch 19/20
```

3375/3375 - 14s - loss: 0.0017 - accuracy: 0.9993 - val_loss: 0.0955 - val_accuracy: 0.9882 - 14s/epoch - 4ms/step

Epoch 20/20

3375/3375 - 14s - loss: 0.0018 - accuracy: 0.9993 - val_loss: 0.0880 - val_accuracy: 0.9882 - 14s/epoch - 4ms/step

313/313 [=====] - 1s 2ms/step

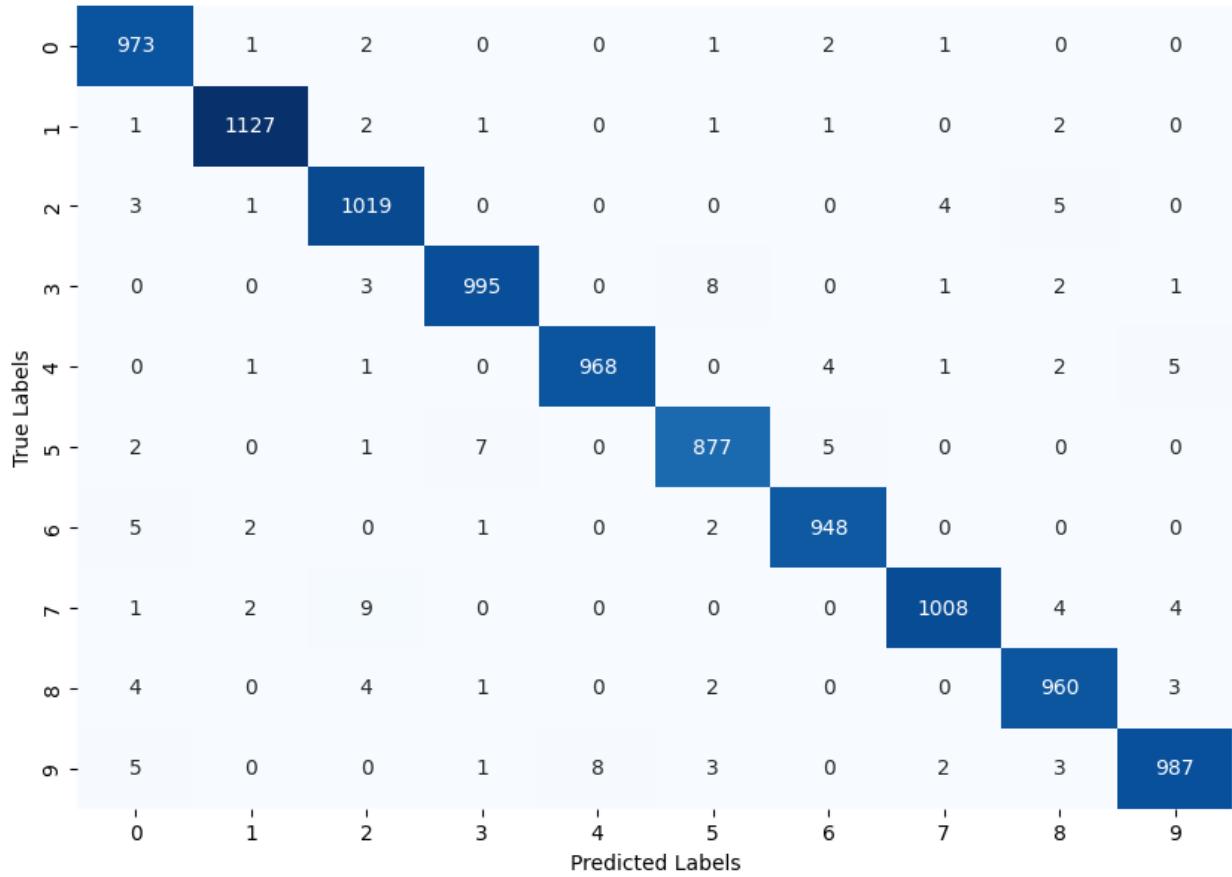
Confusion Matrix adam optimizer and the learning_rate is 0.001, 16 batch size and 20 epochs:

```
[[ 973   1   2   0   0   1   2   1   0   0]
 [ 1 1127   2   1   0   1   1   0   2   0]
 [ 3   1 1019   0   0   0   0   4   5   0]
 [ 0   0   3 995   0   8   0   1   2   1]
 [ 0   1   1   0 968   0   4   1   2   5]
 [ 2   0   1   7   0 877   5   0   0   0]
 [ 5   2   0   1   0   2 948   0   0   0]
 [ 1   2   9   0   0   0   0 1008   4   4]
 [ 4   0   4   1   0   2   0   0 960   3]
 [ 5   0   0   1   8   3   0   2   3 987]]
```

Precision: 0.9862

Recall: 0.9862

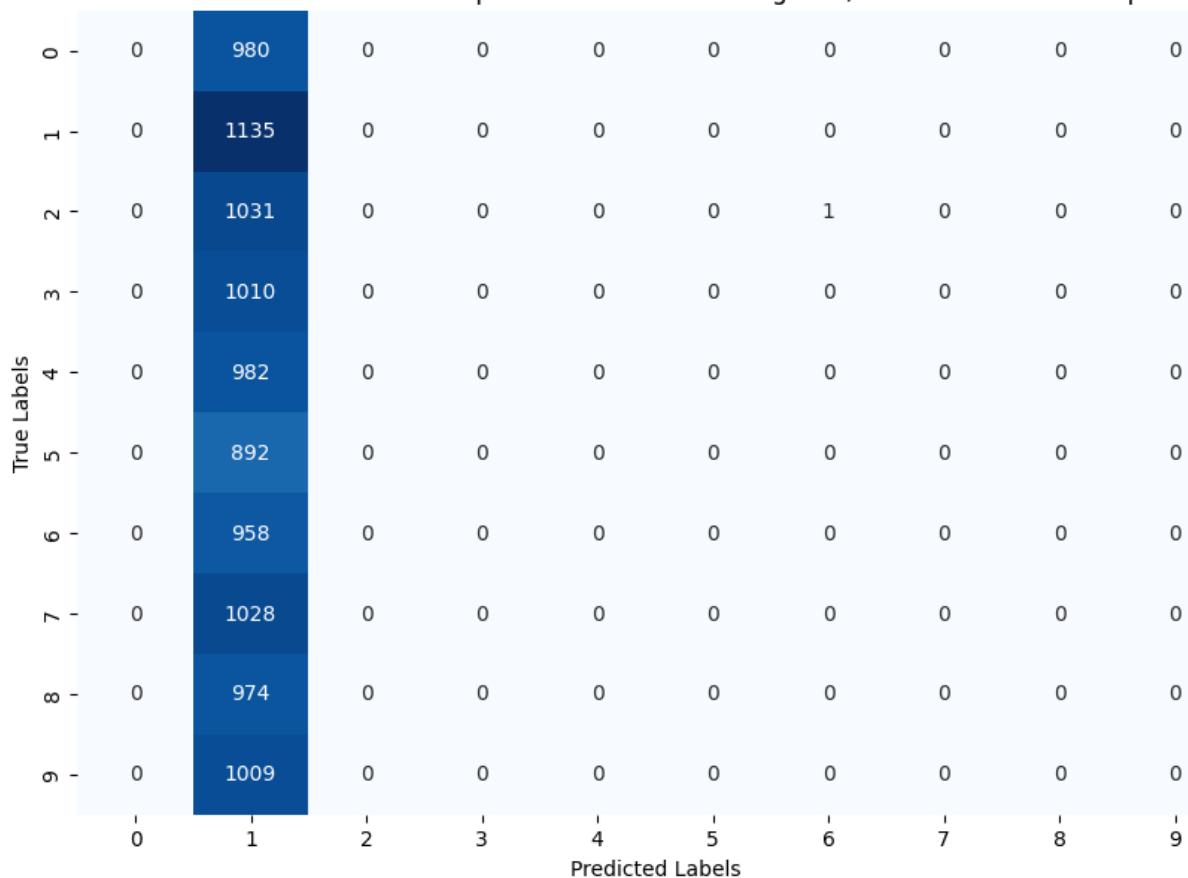
Confusion Matrix for adam optimizer and 0.001 learning rate, 16 batch size and 20 epochs



```
Training with nadam optimizer and the learning_rate is 0.1, 64 batch size and 5 epochs...
Epoch 1/5
844/844 - 6s - loss: 0.5331 - accuracy: 0.8922 - val_loss: 0.2280 -
val_accuracy: 0.9395 - 6s/epoch - 7ms/step
Epoch 2/5
844/844 - 5s - loss: 0.6707 - accuracy: 0.7943 - val_loss: 2.3090 -
val_accuracy: 0.0992 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 2.3113 - accuracy: 0.1034 - val_loss: 2.3124 -
val_accuracy: 0.1113 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 2.3091 - accuracy: 0.1054 - val_loss: 2.3114 -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 2.3089 - accuracy: 0.1046 - val_loss: 2.3103 -
val_accuracy: 0.1050 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 64 batch size and 5 epochs:
[[ 0 980 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0]
 [ 0 1031 0 0 0 0 1 0 0]
 [ 0 1010 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for nadam optimizer and 0.1 learning rate, 64 batch size and 5 epochs



```
Training with nadam optimizer and the learning_rate is 0.1, 64 batch size and 15 epochs...
Epoch 1/15
844/844 - 6s - loss: 0.4628 - accuracy: 0.9094 - val_loss: 0.1909 -
val_accuracy: 0.9515 - 6s/epoch - 7ms/step
Epoch 2/15
844/844 - 4s - loss: 0.3606 - accuracy: 0.9149 - val_loss: 0.2507 -
val_accuracy: 0.9317 - 4s/epoch - 5ms/step
Epoch 3/15
844/844 - 5s - loss: 0.2900 - accuracy: 0.9310 - val_loss: 0.2169 -
val_accuracy: 0.9450 - 5s/epoch - 5ms/step
Epoch 4/15
844/844 - 5s - loss: 0.3271 - accuracy: 0.9255 - val_loss: 0.2811 -
val_accuracy: 0.9437 - 5s/epoch - 5ms/step
Epoch 5/15
844/844 - 5s - loss: 0.4089 - accuracy: 0.9074 - val_loss: 0.3641 -
val_accuracy: 0.9277 - 5s/epoch - 5ms/step
Epoch 6/15
844/844 - 5s - loss: 0.7753 - accuracy: 0.8090 - val_loss: 0.5118 -
val_accuracy: 0.8885 - 5s/epoch - 5ms/step
Epoch 7/15
```

```
844/844 - 5s - loss: 0.4878 - accuracy: 0.8837 - val_loss: 0.4560 -  
val_accuracy: 0.8875 - 5s/epoch - 5ms/step  
Epoch 8/15  
844/844 - 5s - loss: 1.3860 - accuracy: 0.5164 - val_loss: 2.3116 -  
val_accuracy: 0.1053 - 5s/epoch - 5ms/step  
Epoch 9/15  
844/844 - 5s - loss: 2.3128 - accuracy: 0.1069 - val_loss: 2.3138 -  
val_accuracy: 0.0960 - 5s/epoch - 5ms/step  
Epoch 10/15  
844/844 - 5s - loss: 2.3090 - accuracy: 0.1021 - val_loss: 2.3143 -  
val_accuracy: 0.1000 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 2.3091 - accuracy: 0.1043 - val_loss: 2.3075 -  
val_accuracy: 0.0978 - 5s/epoch - 5ms/step  
Epoch 12/15  
844/844 - 5s - loss: 2.3089 - accuracy: 0.1049 - val_loss: 2.3049 -  
val_accuracy: 0.1113 - 5s/epoch - 5ms/step  
Epoch 13/15  
844/844 - 5s - loss: 2.3091 - accuracy: 0.1046 - val_loss: 2.3112 -  
val_accuracy: 0.0992 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 2.3079 - accuracy: 0.1044 - val_loss: 2.3095 -  
val_accuracy: 0.1113 - 5s/epoch - 5ms/step  
Epoch 15/15  
844/844 - 5s - loss: 2.3085 - accuracy: 0.1054 - val_loss: 2.3174 -  
val_accuracy: 0.0960 - 5s/epoch - 5ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 64  
batch size and 15 epochs:  
[[ 0  0  0  0  0  0  0  0  0  980]  
 [ 0  0  0  0  0  0  0  0  0  1135]  
 [ 0  0  0  0  0  0  0  0  0  1032]  
 [ 0  0  0  0  0  0  0  0  0  1010]  
 [ 0  0  0  0  0  0  0  0  0  982]  
 [ 0  0  0  0  0  3  0  0  0  889]  
 [ 0  0  0  0  0  1  0  0  0  957]  
 [ 0  0  0  0  0  0  0  0  0  1028]  
 [ 0  0  0  0  0  0  0  0  0  974]  
 [ 0  0  0  0  0  0  0  0  0  1009]]  
Precision: 0.0771  
Recall: 0.1012  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for nadam optimizer and 0.1 learning rate, 64 batch size and 15 epochs

	0 -	1	2	3	4	5	6	7	8	9	Total
True Labels	0 -	0	0	0	0	0	0	0	0	0	980
1 -	0	0	0	0	0	0	0	0	0	0	1135
2 -	0	0	0	0	0	0	0	0	0	0	1032
3 -	0	0	0	0	0	0	0	0	0	0	1010
4 -	0	0	0	0	0	0	0	0	0	0	982
5 -	0	0	0	0	0	3	0	0	0	0	889
6 -	0	0	0	0	0	1	0	0	0	0	957
7 -	0	0	0	0	0	0	0	0	0	0	1028
8 -	0	0	0	0	0	0	0	0	0	0	974
9 -	0	0	0	0	0	0	0	0	0	0	1009
	0	1	2	3	4	5	6	7	8	9	
	Predicted Labels										

```
Training with nadam optimizer and the learning_rate is 0.1, 64 batch size and 20 epochs...
Epoch 1/20
844/844 - 6s - loss: 0.4683 - accuracy: 0.9220 - val_loss: 0.2276 -
val_accuracy: 0.9308 - 6s/epoch - 7ms/step
Epoch 2/20
844/844 - 5s - loss: 0.2542 - accuracy: 0.9332 - val_loss: 0.1881 -
val_accuracy: 0.9488 - 5s/epoch - 5ms/step
Epoch 3/20
844/844 - 4s - loss: 0.2819 - accuracy: 0.9322 - val_loss: 0.1664 -
val_accuracy: 0.9578 - 4s/epoch - 5ms/step
Epoch 4/20
844/844 - 4s - loss: 0.3266 - accuracy: 0.9270 - val_loss: 0.2617 -
val_accuracy: 0.9478 - 4s/epoch - 5ms/step
Epoch 5/20
844/844 - 5s - loss: 0.7877 - accuracy: 0.8032 - val_loss: 2.3093 -
val_accuracy: 0.1000 - 5s/epoch - 5ms/step
Epoch 6/20
844/844 - 5s - loss: 2.4165 - accuracy: 0.1049 - val_loss: 2.3022 -
val_accuracy: 0.1005 - 5s/epoch - 5ms/step
Epoch 7/20
```

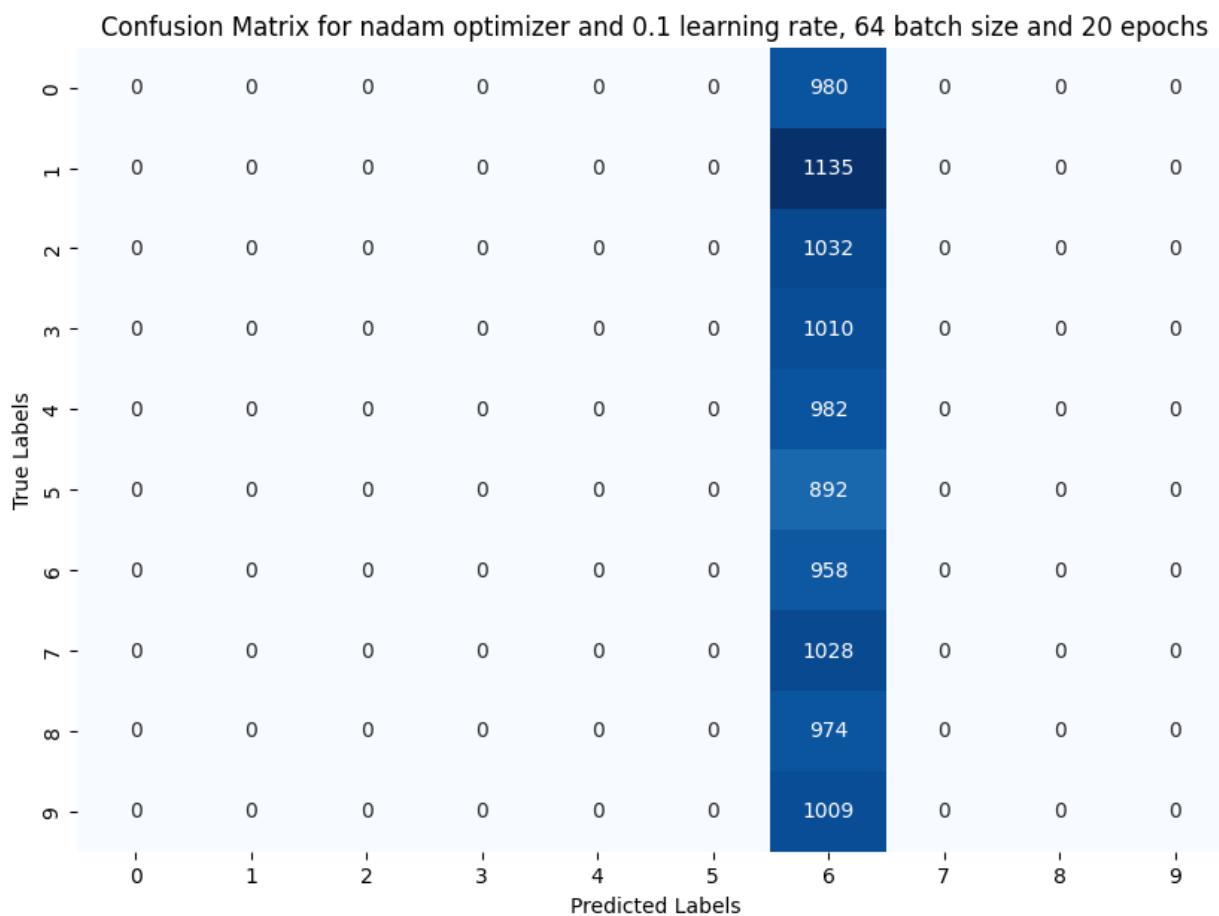
```
844/844 - 5s - loss: 2.3357 - accuracy: 0.1051 - val_loss: 2.3096 -  
val_accuracy: 0.1000 - 5s/epoch - 5ms/step  
Epoch 8/20  
844/844 - 5s - loss: 2.3089 - accuracy: 0.1017 - val_loss: 2.3123 -  
val_accuracy: 0.0978 - 5s/epoch - 5ms/step  
Epoch 9/20  
844/844 - 5s - loss: 2.3091 - accuracy: 0.1052 - val_loss: 2.3086 -  
val_accuracy: 0.1050 - 5s/epoch - 5ms/step  
Epoch 10/20  
844/844 - 5s - loss: 2.3080 - accuracy: 0.1046 - val_loss: 2.3057 -  
val_accuracy: 0.1045 - 5s/epoch - 5ms/step  
Epoch 11/20  
844/844 - 5s - loss: 2.3092 - accuracy: 0.1060 - val_loss: 2.3109 -  
val_accuracy: 0.0992 - 5s/epoch - 5ms/step  
Epoch 12/20  
844/844 - 5s - loss: 2.3086 - accuracy: 0.1056 - val_loss: 2.3116 -  
val_accuracy: 0.0992 - 5s/epoch - 5ms/step  
Epoch 13/20  
844/844 - 5s - loss: 2.3087 - accuracy: 0.1052 - val_loss: 2.3059 -  
val_accuracy: 0.1050 - 5s/epoch - 5ms/step  
Epoch 14/20  
844/844 - 5s - loss: 2.3093 - accuracy: 0.1025 - val_loss: 2.3110 -  
val_accuracy: 0.1050 - 5s/epoch - 5ms/step  
Epoch 15/20  
844/844 - 5s - loss: 2.3086 - accuracy: 0.1063 - val_loss: 2.3129 -  
val_accuracy: 0.0960 - 5s/epoch - 5ms/step  
Epoch 16/20  
844/844 - 5s - loss: 2.3083 - accuracy: 0.1054 - val_loss: 2.3096 -  
val_accuracy: 0.0960 - 5s/epoch - 5ms/step  
Epoch 17/20  
844/844 - 5s - loss: 2.3087 - accuracy: 0.1022 - val_loss: 2.3113 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: 2.3087 - accuracy: 0.1025 - val_loss: 2.3170 -  
val_accuracy: 0.1050 - 5s/epoch - 5ms/step  
Epoch 19/20  
844/844 - 5s - loss: 2.3089 - accuracy: 0.1044 - val_loss: 2.3099 -  
val_accuracy: 0.1050 - 5s/epoch - 5ms/step  
Epoch 20/20  
844/844 - 5s - loss: 2.3085 - accuracy: 0.1055 - val_loss: 2.3125 -  
val_accuracy: 0.0952 - 5s/epoch - 5ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 64  
batch size and 20 epochs:  
[[ 0  0  0  0  0  0  980  0  0  0]  
 [ 0  0  0  0  0  0  1135  0  0  0]  
 [ 0  0  0  0  0  0  1032  0  0  0]  
 [ 0  0  0  0  0  0  1010  0  0  0]  
 [ 0  0  0  0  0  0  982  0  0  0]]
```

```
[ 0  0  0  0  0  0  892  0  0  0]
[ 0  0  0  0  0  0  958  0  0  0]
[ 0  0  0  0  0  0 1028  0  0  0]
[ 0  0  0  0  0  0  974  0  0  0]
[ 0  0  0  0  0  0 1009  0  0  0]]
```

Precision: 0.0092

Recall: 0.0958

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



Training with nadam optimizer and the learning_rate is 0.1, 128 batch size and 5 epochs...

Epoch 1/5

422/422 - 4s - loss: 0.8460 - accuracy: 0.9214 - val_loss: 0.1916 - val_accuracy: 0.9447 - 4s/epoch - 10ms/step

Epoch 2/5

422/422 - 3s - loss: 0.1606 - accuracy: 0.9549 - val_loss: 0.3670 -

```
val_accuracy: 0.8962 - 3s/epoch - 7ms/step
Epoch 3/5
422/422 - 3s - loss: 0.1614 - accuracy: 0.9546 - val_loss: 0.1383 -
val_accuracy: 0.9615 - 3s/epoch - 7ms/step
Epoch 4/5
422/422 - 3s - loss: 0.1729 - accuracy: 0.9534 - val_loss: 0.1491 -
val_accuracy: 0.9588 - 3s/epoch - 7ms/step
Epoch 5/5
422/422 - 3s - loss: 0.1725 - accuracy: 0.9568 - val_loss: 0.1677 -
val_accuracy: 0.9652 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 128
batch size and 5 epochs:
[[ 961     0     4     0     1     1     5     3     5     0]
 [  0 1119     3     0     0     3     2     4     4     0]
 [  2     9   972     7     1     0     1     4    31     5]
 [  1     0   10  929     1    16     0    15    21    17]
 [  0     1     0     0  962     0     3     3     4     9]
 [  2     0     0     5     0  863    12     0     7     3]
 [ 11     5     0     1     7     5  929     0     0     0]
 [  2     4    15     0     3     0     0  994     5     5]
 [  4     4     6     7     1     0     8     8  922    14]
 [  1     4     3     1    14     2     0    37     6  941]]
```

Precision: 0.9595
Recall: 0.9592

Confusion Matrix for nadam optimizer and 0.1 learning rate, 128 batch size and 5 epochs

	0	1	2	3	4	5	6	7	8	9	
True Labels	961	0	4	0	1	1	5	3	5	0	0
0	961	0	4	0	1	1	5	3	5	0	0
1	0	1119	3	0	0	3	2	4	4	0	0
2	2	9	972	7	1	0	1	4	31	5	0
3	1	0	10	929	1	16	0	15	21	17	0
4	0	1	0	0	962	0	3	3	4	9	0
5	2	0	0	5	0	863	12	0	7	3	0
6	11	5	0	1	7	5	929	0	0	0	0
7	2	4	15	0	3	0	0	994	5	5	0
8	4	4	6	7	1	0	8	8	922	14	0
9	1	4	3	1	14	2	0	37	6	941	0
0	1	2	3	4	5	6	7	8	9	9	0
Predicted Labels											

```
Training with nadam optimizer and the learning_rate is 0.1, 128 batch size and 15 epochs...
```

```
Epoch 1/15
```

```
422/422 - 5s - loss: 0.7152 - accuracy: 0.9162 - val_loss: 0.1307 - val_accuracy: 0.9625 - 5s/epoch - 11ms/step
```

```
Epoch 2/15
```

```
422/422 - 3s - loss: 0.1612 - accuracy: 0.9527 - val_loss: 0.2122 - val_accuracy: 0.9378 - 3s/epoch - 7ms/step
```

```
Epoch 3/15
```

```
422/422 - 3s - loss: 0.1723 - accuracy: 0.9533 - val_loss: 0.1497 - val_accuracy: 0.9597 - 3s/epoch - 7ms/step
```

```
Epoch 4/15
```

```
422/422 - 3s - loss: 0.1698 - accuracy: 0.9553 - val_loss: 0.2082 - val_accuracy: 0.9503 - 3s/epoch - 7ms/step
```

```
Epoch 5/15
```

```
422/422 - 3s - loss: 0.1707 - accuracy: 0.9574 - val_loss: 0.2304 - val_accuracy: 0.9508 - 3s/epoch - 7ms/step
```

```
Epoch 6/15
```

```
422/422 - 3s - loss: 0.2072 - accuracy: 0.9516 - val_loss: 0.2533 - val_accuracy: 0.9527 - 3s/epoch - 7ms/step
```

```
Epoch 7/15
```

```
422/422 - 3s - loss: 0.2369 - accuracy: 0.9485 - val_loss: 0.4061 -  
val_accuracy: 0.9355 - 3s/epoch - 7ms/step  
Epoch 8/15  
422/422 - 3s - loss: 0.2541 - accuracy: 0.9459 - val_loss: 0.2726 -  
val_accuracy: 0.9470 - 3s/epoch - 8ms/step  
Epoch 9/15  
422/422 - 3s - loss: 0.2780 - accuracy: 0.9429 - val_loss: 0.2911 -  
val_accuracy: 0.9403 - 3s/epoch - 8ms/step  
Epoch 10/15  
422/422 - 3s - loss: 0.3783 - accuracy: 0.9275 - val_loss: 0.3818 -  
val_accuracy: 0.9258 - 3s/epoch - 7ms/step  
Epoch 11/15  
422/422 - 3s - loss: 0.5730 - accuracy: 0.8812 - val_loss: 1.2257 -  
val_accuracy: 0.7950 - 3s/epoch - 7ms/step  
Epoch 12/15  
422/422 - 3s - loss: 0.9405 - accuracy: 0.7712 - val_loss: 0.6984 -  
val_accuracy: 0.8005 - 3s/epoch - 8ms/step  
Epoch 13/15  
422/422 - 3s - loss: 0.8218 - accuracy: 0.7963 - val_loss: 0.7074 -  
val_accuracy: 0.8025 - 3s/epoch - 8ms/step  
Epoch 14/15  
422/422 - 3s - loss: 0.7936 - accuracy: 0.8274 - val_loss: 0.6797 -  
val_accuracy: 0.8328 - 3s/epoch - 7ms/step  
Epoch 15/15  
422/422 - 3s - loss: 0.7057 - accuracy: 0.8218 - val_loss: 0.7267 -  
val_accuracy: 0.8280 - 3s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 128  
batch size and 15 epochs:  
[[ 886  0  0  0  0  0  1  1  0  92]  
[  0 1084  1  0  1  0  1  0  0  48]  
[  1  0 809  0  0  0  0  1  0 221]  
[  0  0  0 702  0  0  0  3  0 305]  
[  0  1  0  0 863  0  0  0  1 117]  
[  0  0  0  1  0 622  1  0  0 268]  
[  1  2  0  0  5  1 849  0  0 100]  
[  0  1  1  0  1  0  0 796  1 228]  
[  2  0  3  0  4  1  0  0 512 452]  
[  0  1  0  0  2  0  0  3  0 1003]]  
Precision: 0.9296  
Recall: 0.8126
```

Confusion Matrix for nadam optimizer and 0.1 learning rate, 128 batch size and 15 epochs

	0	1	2	3	4	5	6	7	8	9	
True Labels	886	0	0	0	0	0	1	1	0	92	92
0	0	1084	1	0	1	0	1	0	0	48	48
1	1	0	809	0	0	0	0	1	0	221	221
2	0	0	0	702	0	0	0	3	0	305	305
3	0	1	0	0	863	0	0	0	1	117	117
4	0	0	0	1	0	622	1	0	0	268	268
5	1	2	0	0	5	1	849	0	0	100	100
6	0	1	1	0	1	0	0	0	1	228	228
7	2	0	3	0	4	1	0	0	512	452	452
8	0	1	0	0	2	0	0	3	0	1003	1003
9	0	1	2	3	4	5	6	7	8	9	9
	0	1	2	3	4	5	6	7	8	9	

Training with nadam optimizer and the learning_rate is 0.1, 128 batch size and 20 epochs...

Epoch 1/20

422/422 - 5s - loss: 0.7335 - accuracy: 0.9184 - val_loss: 0.1705 - val_accuracy: 0.9525 - 5s/epoch - 11ms/step

Epoch 2/20

422/422 - 3s - loss: 0.1991 - accuracy: 0.9484 - val_loss: 0.2923 - val_accuracy: 0.9270 - 3s/epoch - 7ms/step

Epoch 3/20

422/422 - 3s - loss: 0.1849 - accuracy: 0.9488 - val_loss: 0.1284 - val_accuracy: 0.9658 - 3s/epoch - 7ms/step

Epoch 4/20

422/422 - 3s - loss: 0.1584 - accuracy: 0.9570 - val_loss: 0.1637 - val_accuracy: 0.9602 - 3s/epoch - 7ms/step

Epoch 5/20

422/422 - 3s - loss: 0.1568 - accuracy: 0.9600 - val_loss: 0.1777 - val_accuracy: 0.9585 - 3s/epoch - 7ms/step

Epoch 6/20

422/422 - 3s - loss: 0.1968 - accuracy: 0.9528 - val_loss: 0.1431 - val_accuracy: 0.9612 - 3s/epoch - 7ms/step

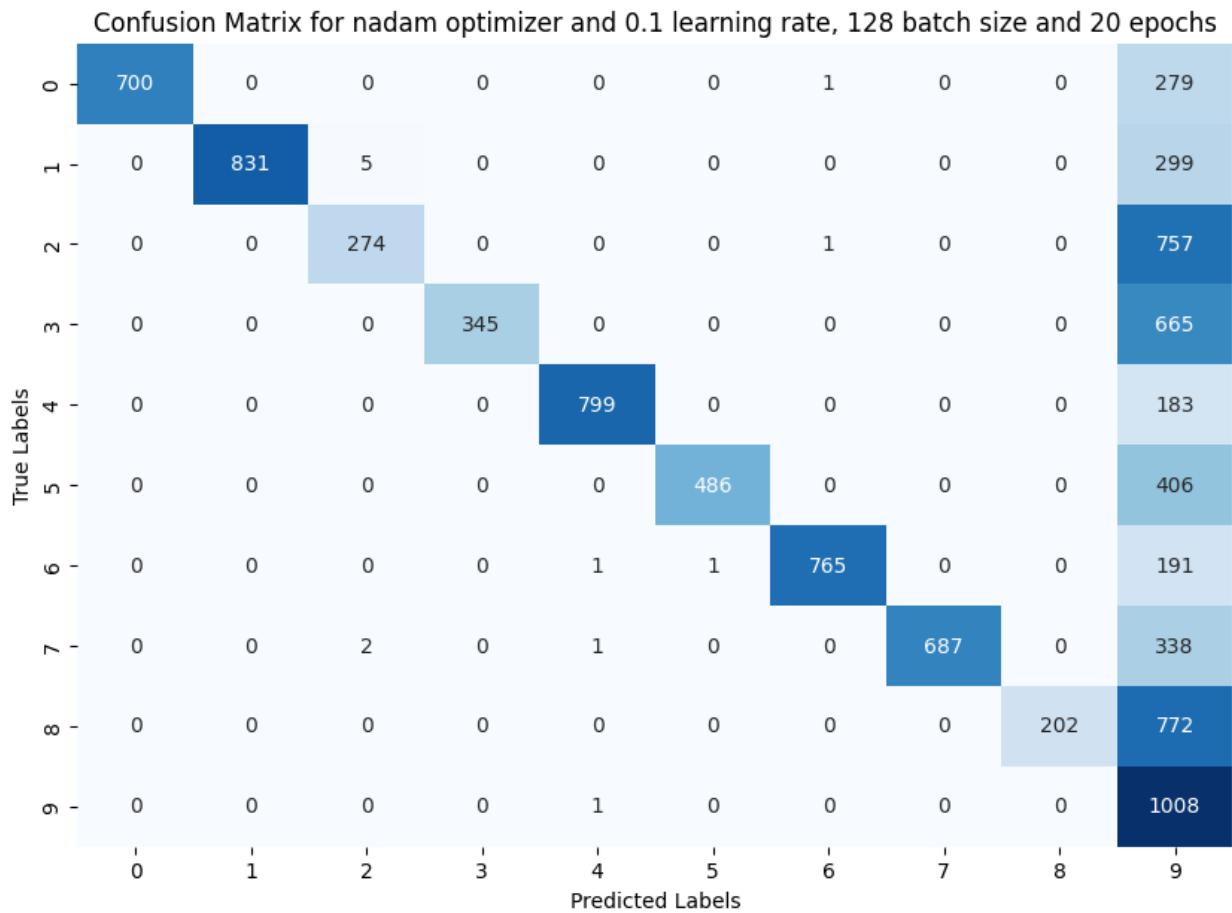
Epoch 7/20

```
422/422 - 3s - loss: 0.1771 - accuracy: 0.9586 - val_loss: 0.2384 -  
val_accuracy: 0.9527 - 3s/epoch - 7ms/step  
Epoch 8/20  
422/422 - 3s - loss: 0.2042 - accuracy: 0.9546 - val_loss: 0.2668 -  
val_accuracy: 0.9520 - 3s/epoch - 7ms/step  
Epoch 9/20  
422/422 - 3s - loss: 0.2689 - accuracy: 0.9449 - val_loss: 0.3211 -  
val_accuracy: 0.9350 - 3s/epoch - 7ms/step  
Epoch 10/20  
422/422 - 3s - loss: 0.3147 - accuracy: 0.9335 - val_loss: 0.2903 -  
val_accuracy: 0.9460 - 3s/epoch - 7ms/step  
Epoch 11/20  
422/422 - 3s - loss: 0.3814 - accuracy: 0.9200 - val_loss: 0.8617 -  
val_accuracy: 0.9053 - 3s/epoch - 7ms/step  
Epoch 12/20  
422/422 - 3s - loss: 0.8368 - accuracy: 0.7855 - val_loss: 0.6228 -  
val_accuracy: 0.8213 - 3s/epoch - 8ms/step  
Epoch 13/20  
422/422 - 3s - loss: 0.8932 - accuracy: 0.7802 - val_loss: 0.9381 -  
val_accuracy: 0.7777 - 3s/epoch - 8ms/step  
Epoch 14/20  
422/422 - 3s - loss: 1.0891 - accuracy: 0.6990 - val_loss: 1.1618 -  
val_accuracy: 0.6398 - 3s/epoch - 8ms/step  
Epoch 15/20  
422/422 - 3s - loss: 1.0404 - accuracy: 0.7053 - val_loss: 0.7876 -  
val_accuracy: 0.7508 - 3s/epoch - 8ms/step  
Epoch 16/20  
422/422 - 3s - loss: 1.1005 - accuracy: 0.6787 - val_loss: 0.9353 -  
val_accuracy: 0.7832 - 3s/epoch - 8ms/step  
Epoch 17/20  
422/422 - 3s - loss: 1.4168 - accuracy: 0.6509 - val_loss: 1.1931 -  
val_accuracy: 0.6917 - 3s/epoch - 8ms/step  
Epoch 18/20  
422/422 - 3s - loss: 0.9875 - accuracy: 0.7011 - val_loss: 1.2970 -  
val_accuracy: 0.7185 - 3s/epoch - 8ms/step  
Epoch 19/20  
422/422 - 3s - loss: 1.0451 - accuracy: 0.7125 - val_loss: 1.6376 -  
val_accuracy: 0.7363 - 3s/epoch - 8ms/step  
Epoch 20/20  
422/422 - 3s - loss: 1.0103 - accuracy: 0.6987 - val_loss: 1.8655 -  
val_accuracy: 0.6278 - 3s/epoch - 8ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 128  
batch size and 20 epochs:  
[[ 700 0 0 0 0 0 1 0 0 279]  
[ 0 831 5 0 0 0 0 0 0 299]  
[ 0 0 274 0 0 0 1 0 0 757]  
[ 0 0 0 345 0 0 0 0 0 665]  
[ 0 0 0 0 799 0 0 0 0 183]]
```

```
[ 0  0  0  0  0 486  0  0  0 406]
[ 0  0  0  0  1  1 765  0  0 191]
[ 0  0  2  0  1  0  0 687  0 338]
[ 0  0  0  0  0  0  0  0 202 772]
[ 0  0  0  0  1  0  0  0  0 1008]]
```

Precision: 0.9165

Recall: 0.6097



Training with nadam optimizer and the learning_rate is 0.1, 256 batch size and 5 epochs...

Epoch 1/5

211/211 - 4s - loss: 1.5026 - accuracy: 0.8569 - val_loss: 0.1419 - val_accuracy: 0.9568 - 4s/epoch - 18ms/step

Epoch 2/5

211/211 - 3s - loss: 0.1535 - accuracy: 0.9531 - val_loss: 0.1184 - val_accuracy: 0.9668 - 3s/epoch - 12ms/step

Epoch 3/5

211/211 - 2s - loss: 0.1323 - accuracy: 0.9592 - val_loss: 0.1241 - val_accuracy: 0.9648 - 2s/epoch - 12ms/step

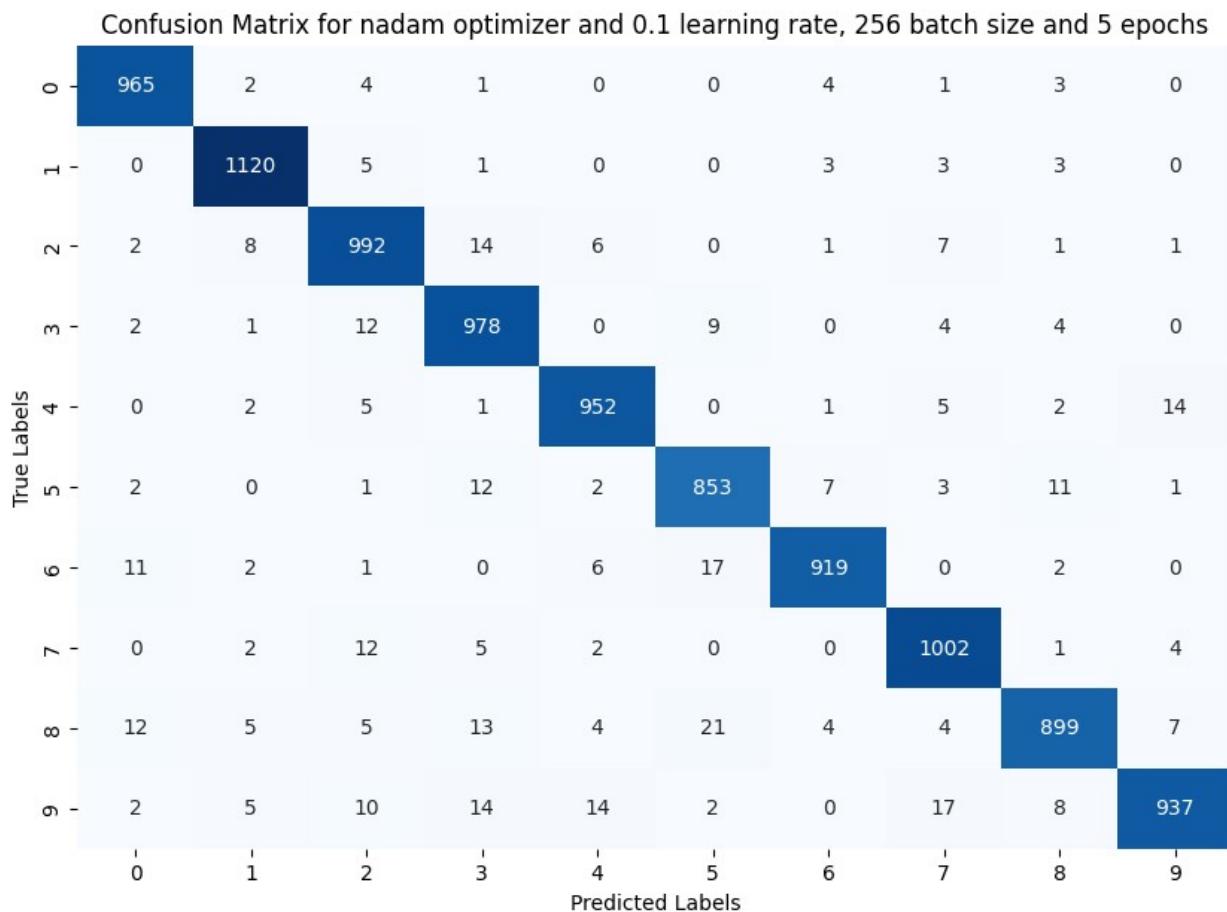
Epoch 4/5

211/211 - 2s - loss: 0.1373 - accuracy: 0.9579 - val_loss: 0.1609 -

```

val_accuracy: 0.9593 - 2s/epoch - 12ms/step
Epoch 5/5
211/211 - 3s - loss: 0.1272 - accuracy: 0.9605 - val_loss: 0.1070 -
val_accuracy: 0.9710 - 3s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 256
batch size and 5 epochs:
[[ 965   2   4   1   0   0   4   1   3   0]
 [  0 1120   5   1   0   0   3   3   3   0]
 [  2   8 992  14   6   0   1   7   1   1]
 [  2   1 12 978   0   9   0   4   4   0]
 [  0   2   5   1 952   0   1   5   2 14]
 [  2   0   1 12   2 853   7   3 11   1]
 [ 11   2   1   0   6 17 919   0   2   0]
 [  0   2 12   5   2   0   0 1002   1   4]
 [ 12   5   5 13   4 21   4   4 899   7]
 [  2   5 10 14 14   2   0 17   8 937]]
Precision: 0.9619
Recall: 0.9617

```

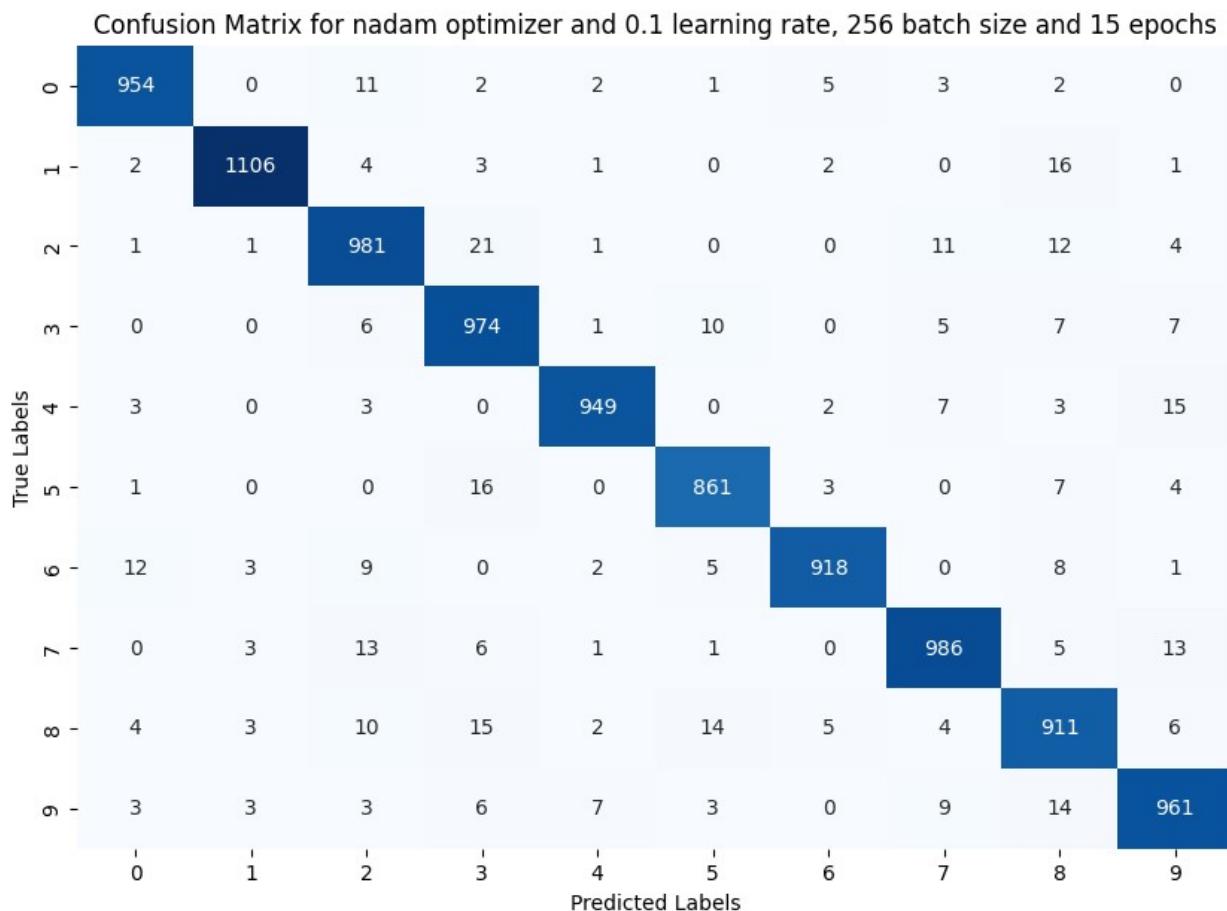


```
Training with nadam optimizer and the learning_rate is 0.1, 256 batch size and 15 epochs...
Epoch 1/15
211/211 - 4s - loss: 0.8220 - accuracy: 0.9196 - val_loss: 0.1043 -
val_accuracy: 0.9682 - 4s/epoch - 18ms/step
Epoch 2/15
211/211 - 2s - loss: 0.1063 - accuracy: 0.9677 - val_loss: 0.0934 -
val_accuracy: 0.9730 - 2s/epoch - 12ms/step
Epoch 3/15
211/211 - 2s - loss: 0.0929 - accuracy: 0.9721 - val_loss: 0.0933 -
val_accuracy: 0.9740 - 2s/epoch - 11ms/step
Epoch 4/15
211/211 - 2s - loss: 0.0906 - accuracy: 0.9735 - val_loss: 0.1118 -
val_accuracy: 0.9728 - 2s/epoch - 11ms/step
Epoch 5/15
211/211 - 2s - loss: 0.0856 - accuracy: 0.9750 - val_loss: 0.1313 -
val_accuracy: 0.9667 - 2s/epoch - 12ms/step
Epoch 6/15
211/211 - 2s - loss: 0.0978 - accuracy: 0.9736 - val_loss: 0.1501 -
val_accuracy: 0.9672 - 2s/epoch - 11ms/step
Epoch 7/15
211/211 - 2s - loss: 0.1001 - accuracy: 0.9736 - val_loss: 0.1285 -
val_accuracy: 0.9723 - 2s/epoch - 12ms/step
Epoch 8/15
211/211 - 2s - loss: 0.1084 - accuracy: 0.9735 - val_loss: 0.1706 -
val_accuracy: 0.9625 - 2s/epoch - 12ms/step
Epoch 9/15
211/211 - 2s - loss: 0.1076 - accuracy: 0.9725 - val_loss: 0.1368 -
val_accuracy: 0.9670 - 2s/epoch - 12ms/step
Epoch 10/15
211/211 - 2s - loss: 0.1093 - accuracy: 0.9732 - val_loss: 0.1349 -
val_accuracy: 0.9748 - 2s/epoch - 12ms/step
Epoch 11/15
211/211 - 2s - loss: 0.1445 - accuracy: 0.9696 - val_loss: 0.1977 -
val_accuracy: 0.9550 - 2s/epoch - 12ms/step
Epoch 12/15
211/211 - 3s - loss: 0.1446 - accuracy: 0.9641 - val_loss: 0.1531 -
val_accuracy: 0.9672 - 3s/epoch - 12ms/step
Epoch 13/15
211/211 - 2s - loss: 0.1011 - accuracy: 0.9737 - val_loss: 0.1541 -
val_accuracy: 0.9698 - 2s/epoch - 12ms/step
Epoch 14/15
211/211 - 2s - loss: 0.1163 - accuracy: 0.9722 - val_loss: 0.1911 -
val_accuracy: 0.9595 - 2s/epoch - 12ms/step
Epoch 15/15
211/211 - 2s - loss: 0.1517 - accuracy: 0.9662 - val_loss: 0.1865 -
val_accuracy: 0.9650 - 2s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 256 batch size and 15 epochs:
```

```
[[ 954  0  11  2  2  1  5  3  2  0]
 [ 2 1106  4  3  1  0  2  0  16  1]
 [ 1  1 981 21  1  0  0 11 12  4]
 [ 0  0  6 974  1 10  0  5  7  7]
 [ 3  0  3  0 949  0  2  7  3 15]
 [ 1  0  0 16  0 861  3  0  7  4]
 [ 12 3  9  0  2  5 918  0  8  1]
 [ 0  3 13  6  1  1  0 986  5 13]
 [ 4  3 10 15  2 14  5  4 911  6]
 [ 3  3  3  6  7  3  0  9 14 961]]
```

Precision: 0.9604

Recall: 0.9601



Training with nadam optimizer and the learning_rate is 0.1, 256 batch size and 20 epochs...

Epoch 1/20

211/211 - 4s - loss: 0.7917 - accuracy: 0.9106 - val_loss: 0.1348 - val_accuracy: 0.9628 - 4s/epoch - 18ms/step

Epoch 2/20

211/211 - 2s - loss: 0.1267 - accuracy: 0.9617 - val_loss: 0.1130 - val_accuracy: 0.9690 - 2s/epoch - 12ms/step

```
Epoch 3/20
211/211 - 2s - loss: 0.1109 - accuracy: 0.9663 - val_loss: 0.1135 -
val_accuracy: 0.9710 - 2s/epoch - 12ms/step
Epoch 4/20
211/211 - 2s - loss: 0.0969 - accuracy: 0.9706 - val_loss: 0.1156 -
val_accuracy: 0.9713 - 2s/epoch - 12ms/step
Epoch 5/20
211/211 - 3s - loss: 0.2240 - accuracy: 0.9637 - val_loss: 0.8735 -
val_accuracy: 0.8368 - 3s/epoch - 12ms/step
Epoch 6/20
211/211 - 2s - loss: 0.3574 - accuracy: 0.9020 - val_loss: 0.1843 -
val_accuracy: 0.9482 - 2s/epoch - 12ms/step
Epoch 7/20
211/211 - 3s - loss: 0.1828 - accuracy: 0.9460 - val_loss: 0.1554 -
val_accuracy: 0.9578 - 3s/epoch - 12ms/step
Epoch 8/20
211/211 - 3s - loss: 0.1519 - accuracy: 0.9549 - val_loss: 0.1601 -
val_accuracy: 0.9557 - 3s/epoch - 12ms/step
Epoch 9/20
211/211 - 3s - loss: 0.1392 - accuracy: 0.9586 - val_loss: 0.1734 -
val_accuracy: 0.9590 - 3s/epoch - 12ms/step
Epoch 10/20
211/211 - 3s - loss: 0.1368 - accuracy: 0.9590 - val_loss: 0.1597 -
val_accuracy: 0.9618 - 3s/epoch - 12ms/step
Epoch 11/20
211/211 - 3s - loss: 0.1331 - accuracy: 0.9618 - val_loss: 0.1876 -
val_accuracy: 0.9533 - 3s/epoch - 12ms/step
Epoch 12/20
211/211 - 2s - loss: 0.1334 - accuracy: 0.9602 - val_loss: 0.1541 -
val_accuracy: 0.9623 - 2s/epoch - 12ms/step
Epoch 13/20
211/211 - 3s - loss: 0.1180 - accuracy: 0.9656 - val_loss: 0.1537 -
val_accuracy: 0.9593 - 3s/epoch - 12ms/step
Epoch 14/20
211/211 - 3s - loss: 0.1201 - accuracy: 0.9655 - val_loss: 0.1824 -
val_accuracy: 0.9522 - 3s/epoch - 12ms/step
Epoch 15/20
211/211 - 3s - loss: 0.1292 - accuracy: 0.9629 - val_loss: 0.1753 -
val_accuracy: 0.9565 - 3s/epoch - 12ms/step
Epoch 16/20
211/211 - 3s - loss: 0.1212 - accuracy: 0.9660 - val_loss: 0.1843 -
val_accuracy: 0.9580 - 3s/epoch - 12ms/step
Epoch 17/20
211/211 - 3s - loss: 0.1152 - accuracy: 0.9665 - val_loss: 0.1549 -
val_accuracy: 0.9635 - 3s/epoch - 12ms/step
Epoch 18/20
211/211 - 2s - loss: 0.1133 - accuracy: 0.9678 - val_loss: 0.1436 -
val_accuracy: 0.9663 - 2s/epoch - 12ms/step
Epoch 19/20
```

```
211/211 - 2s - loss: 0.1992 - accuracy: 0.9511 - val_loss: 0.1912 -  
val_accuracy: 0.9607 - 2s/epoch - 12ms/step
```

```
Epoch 20/20
```

```
211/211 - 2s - loss: 0.1230 - accuracy: 0.9653 - val_loss: 0.1894 -  
val_accuracy: 0.9588 - 2s/epoch - 12ms/step
```

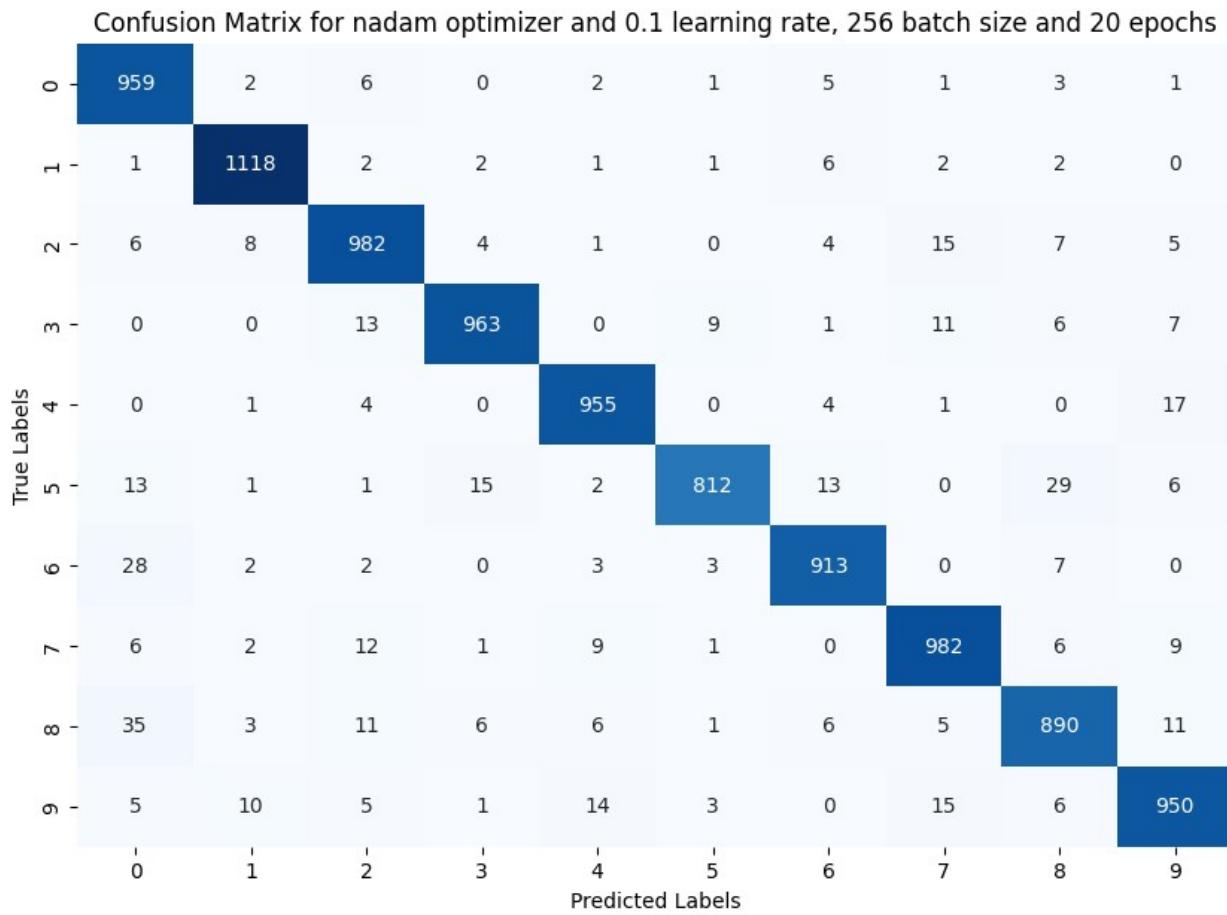
```
313/313 [=====] - 1s 2ms/step
```

```
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 256  
batch size and 20 epochs:
```

```
[[ 959   2   6   0   2   1   5   1   3   1]  
 [ 1 1118   2   2   1   1   6   2   2   0]  
 [ 6   8 982   4   1   0   4 15   7   5]  
 [ 0   0 13 963   0   9   1 11   6   7]  
 [ 0   1   4   0 955   0   4   1   0 17]  
 [ 13  1   1 15   2 812 13   0 29   6]  
 [ 28   2   2   0   3   3 913   0   7   0]  
 [ 6   2 12   1   9   1   0 982   6   9]  
 [ 35   3 11   6   6   1   6   5 890 11]  
 [ 5 10   5   1 14   3   0 15   6 950]]
```

```
Precision: 0.9528
```

```
Recall: 0.9524
```



```
Training with nadam optimizer and the learning_rate is 0.1, 16 batch size and 5 epochs...
Epoch 1/5
3375/3375 - 16s - loss: 1.6776 - accuracy: 0.4309 - val_loss: 2.3126 -
val_accuracy: 0.0952 - 16s/epoch - 5ms/step
Epoch 2/5
3375/3375 - 14s - loss: 2.3176 - accuracy: 0.1027 - val_loss: 2.3156 -
val_accuracy: 0.1113 - 14s/epoch - 4ms/step
Epoch 3/5
3375/3375 - 14s - loss: 2.3178 - accuracy: 0.1036 - val_loss: 2.3153 -
val_accuracy: 0.1050 - 14s/epoch - 4ms/step
Epoch 4/5
3375/3375 - 15s - loss: 2.3178 - accuracy: 0.1009 - val_loss: 2.3200 -
val_accuracy: 0.1050 - 15s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 15s - loss: 2.3170 - accuracy: 0.1034 - val_loss: 2.3160 -
val_accuracy: 0.1050 - 15s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 16 batch size and 5 epochs:
[[ 0 980 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0]]]
Precision: 0.0129
Recall: 0.1135

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for nadam optimizer and 0.1 learning rate, 16 batch size and 5 epochs

		0	1	2	3	4	5	6	7	8	9
True Labels	0	980	0	0	0	0	0	0	0	0	0
	1	1135	0	0	0	0	0	0	0	0	0
	2	1032	0	0	0	0	0	0	0	0	0
	3	1010	0	0	0	0	0	0	0	0	0
	4	982	0	0	0	0	0	0	0	0	0
	5	892	0	0	0	0	0	0	0	0	0
	6	958	0	0	0	0	0	0	0	0	0
	7	1028	0	0	0	0	0	0	0	0	0
	8	974	0	0	0	0	0	0	0	0	0
	9	1009	0	0	0	0	0	0	0	0	0

Training with nadam optimizer and the learning_rate is 0.1, 16 batch size and 15 epochs...

Epoch 1/15

3375/3375 - 15s - loss: 2.0435 - accuracy: 0.2796 - val_loss: 2.3182 - val_accuracy: 0.0960 - 15s/epoch - 5ms/step

Epoch 2/15

3375/3375 - 14s - loss: 2.3194 - accuracy: 0.1032 - val_loss: 2.3316 - val_accuracy: 0.1050 - 14s/epoch - 4ms/step

Epoch 3/15

3375/3375 - 15s - loss: 2.3178 - accuracy: 0.1014 - val_loss: 2.3245 - val_accuracy: 0.0915 - 15s/epoch - 4ms/step

Epoch 4/15

3375/3375 - 15s - loss: 2.3173 - accuracy: 0.1032 - val_loss: 2.3130 - val_accuracy: 0.0992 - 15s/epoch - 4ms/step

Epoch 5/15

3375/3375 - 15s - loss: 2.3180 - accuracy: 0.1031 - val_loss: 2.3150 - val_accuracy: 0.1045 - 15s/epoch - 4ms/step

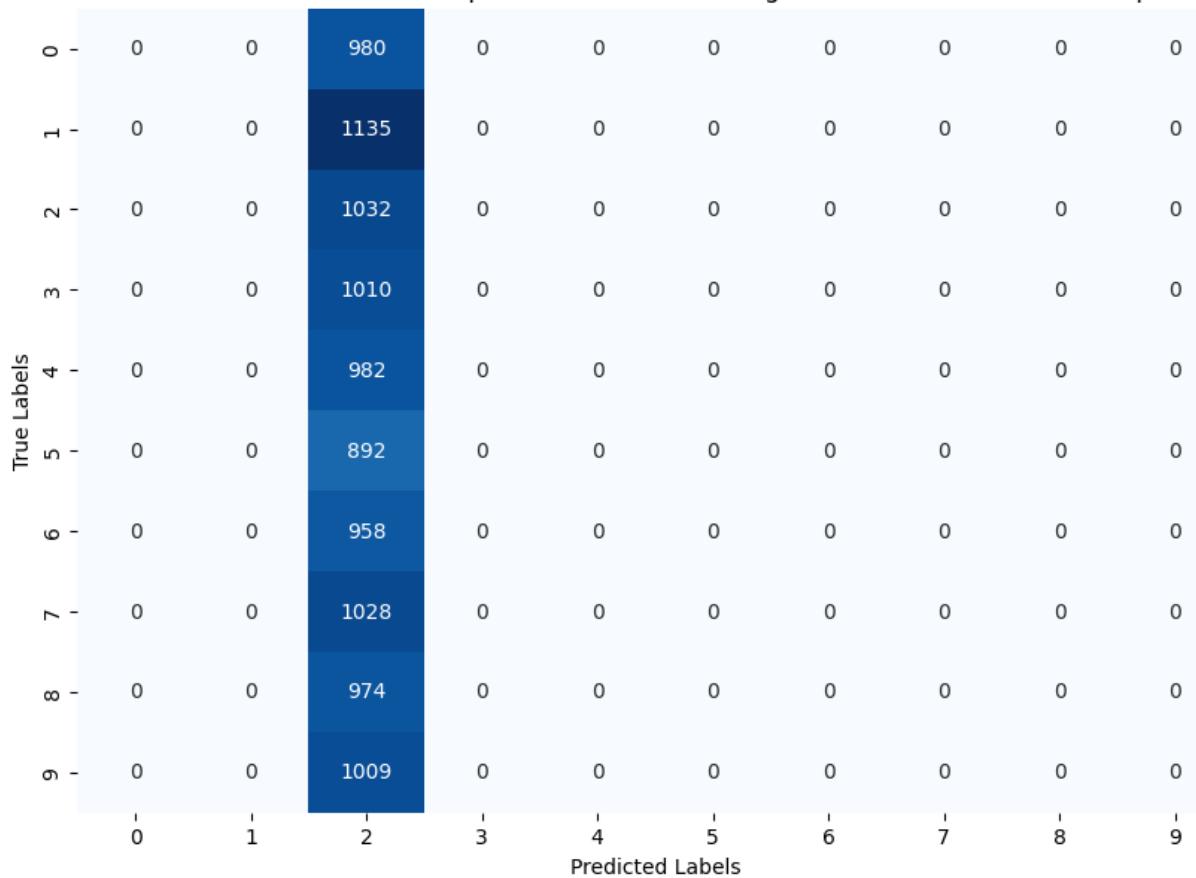
Epoch 6/15

3375/3375 - 14s - loss: 2.3180 - accuracy: 0.1018 - val_loss: 2.3081 - val_accuracy: 0.0960 - 14s/epoch - 4ms/step

Epoch 7/15

```
3375/3375 - 14s - loss: 2.3177 - accuracy: 0.1035 - val_loss: 2.3248 -  
val_accuracy: 0.0960 - 14s/epoch - 4ms/step  
Epoch 8/15  
3375/3375 - 15s - loss: 2.3180 - accuracy: 0.1039 - val_loss: 2.3123 -  
val_accuracy: 0.1113 - 15s/epoch - 4ms/step  
Epoch 9/15  
3375/3375 - 14s - loss: 2.3178 - accuracy: 0.1047 - val_loss: 2.3102 -  
val_accuracy: 0.0978 - 14s/epoch - 4ms/step  
Epoch 10/15  
3375/3375 - 15s - loss: 2.3169 - accuracy: 0.1053 - val_loss: 2.3305 -  
val_accuracy: 0.1050 - 15s/epoch - 4ms/step  
Epoch 11/15  
3375/3375 - 15s - loss: 2.3184 - accuracy: 0.1046 - val_loss: 2.3138 -  
val_accuracy: 0.1113 - 15s/epoch - 4ms/step  
Epoch 12/15  
3375/3375 - 15s - loss: 2.3170 - accuracy: 0.1044 - val_loss: 2.3189 -  
val_accuracy: 0.0952 - 15s/epoch - 4ms/step  
Epoch 13/15  
3375/3375 - 15s - loss: 2.3175 - accuracy: 0.1045 - val_loss: 2.3071 -  
val_accuracy: 0.1113 - 15s/epoch - 4ms/step  
Epoch 14/15  
3375/3375 - 15s - loss: 2.3179 - accuracy: 0.1026 - val_loss: 2.3125 -  
val_accuracy: 0.0995 - 15s/epoch - 4ms/step  
Epoch 15/15  
3375/3375 - 15s - loss: 2.3180 - accuracy: 0.1007 - val_loss: 2.3256 -  
val_accuracy: 0.1000 - 15s/epoch - 4ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 16  
batch size and 15 epochs:  
[[ 0  0 980  0  0  0  0  0  0  0]  
[ 0  0 1135  0  0  0  0  0  0  0]  
[ 0  0 1032  0  0  0  0  0  0  0]  
[ 0  0 1010  0  0  0  0  0  0  0]  
[ 0  0 982  0  0  0  0  0  0  0]  
[ 0  0 892  0  0  0  0  0  0  0]  
[ 0  0 958  0  0  0  0  0  0  0]  
[ 0  0 1028  0  0  0  0  0  0  0]  
[ 0  0 974  0  0  0  0  0  0  0]  
[ 0  0 1009  0  0  0  0  0  0  0]]  
Precision: 0.0107  
Recall: 0.1032  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for nadam optimizer and 0.1 learning rate, 16 batch size and 15 epochs



Training with nadam optimizer and the learning_rate is 0.1, 16 batch size and 20 epochs...

Epoch 1/20

3375/3375 - 17s - loss: 1.5590 - accuracy: 0.4733 - val_loss: 2.3149 - val_accuracy: 0.1113 - 17s/epoch - 5ms/step

Epoch 2/20

3375/3375 - 14s - loss: 2.3178 - accuracy: 0.1039 - val_loss: 2.3110 - val_accuracy: 0.0995 - 14s/epoch - 4ms/step

Epoch 3/20

3375/3375 - 15s - loss: 2.3173 - accuracy: 0.1054 - val_loss: 2.3104 - val_accuracy: 0.1113 - 15s/epoch - 4ms/step

Epoch 4/20

3375/3375 - 15s - loss: 2.3181 - accuracy: 0.1033 - val_loss: 2.3191 - val_accuracy: 0.0952 - 15s/epoch - 4ms/step

Epoch 5/20

3375/3375 - 15s - loss: 2.3181 - accuracy: 0.1031 - val_loss: 2.3152 - val_accuracy: 0.0915 - 15s/epoch - 4ms/step

Epoch 6/20

3375/3375 - 14s - loss: 2.3177 - accuracy: 0.1038 - val_loss: 2.3139 - val_accuracy: 0.0960 - 14s/epoch - 4ms/step

Epoch 7/20

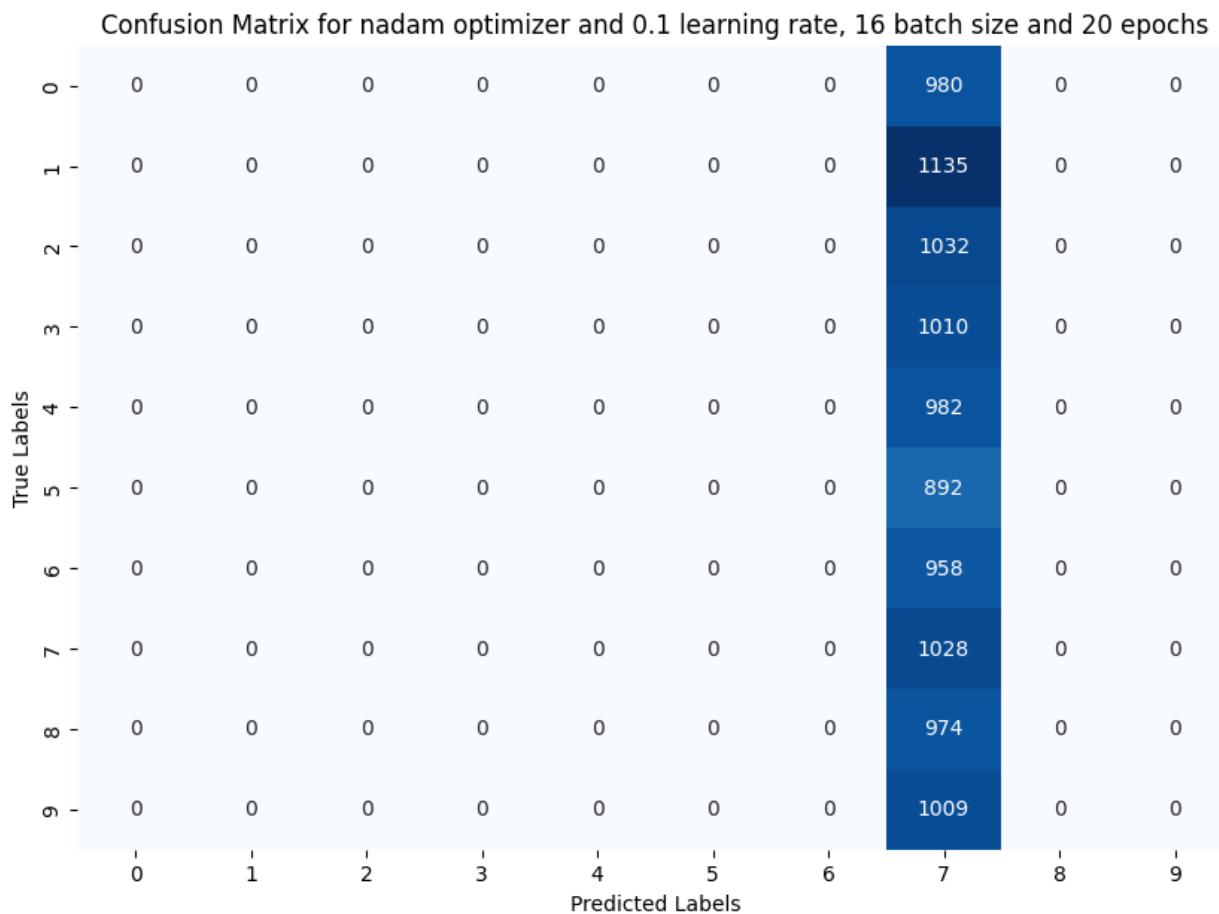
```
3375/3375 - 14s - loss: 2.3178 - accuracy: 0.0999 - val_loss: 2.3076 -  
val_accuracy: 0.1113 - 14s/epoch - 4ms/step  
Epoch 8/20  
3375/3375 - 14s - loss: 2.3186 - accuracy: 0.1013 - val_loss: 2.3061 -  
val_accuracy: 0.0978 - 14s/epoch - 4ms/step  
Epoch 9/20  
3375/3375 - 15s - loss: 2.3171 - accuracy: 0.1037 - val_loss: 2.3298 -  
val_accuracy: 0.0960 - 15s/epoch - 4ms/step  
Epoch 10/20  
3375/3375 - 15s - loss: 2.3177 - accuracy: 0.1039 - val_loss: 2.3204 -  
val_accuracy: 0.1050 - 15s/epoch - 4ms/step  
Epoch 11/20  
3375/3375 - 15s - loss: 2.3182 - accuracy: 0.1033 - val_loss: 2.3207 -  
val_accuracy: 0.1050 - 15s/epoch - 4ms/step  
Epoch 12/20  
3375/3375 - 15s - loss: 2.3183 - accuracy: 0.1050 - val_loss: 2.3119 -  
val_accuracy: 0.0960 - 15s/epoch - 4ms/step  
Epoch 13/20  
3375/3375 - 14s - loss: 2.3179 - accuracy: 0.1029 - val_loss: 2.3257 -  
val_accuracy: 0.1000 - 14s/epoch - 4ms/step  
Epoch 14/20  
3375/3375 - 15s - loss: 2.3184 - accuracy: 0.1031 - val_loss: 2.3155 -  
val_accuracy: 0.1050 - 15s/epoch - 4ms/step  
Epoch 15/20  
3375/3375 - 15s - loss: 2.3172 - accuracy: 0.1035 - val_loss: 2.3281 -  
val_accuracy: 0.0978 - 15s/epoch - 4ms/step  
Epoch 16/20  
3375/3375 - 15s - loss: 2.3186 - accuracy: 0.1013 - val_loss: 2.3325 -  
val_accuracy: 0.1050 - 15s/epoch - 4ms/step  
Epoch 17/20  
3375/3375 - 15s - loss: 2.3182 - accuracy: 0.1032 - val_loss: 2.3126 -  
val_accuracy: 0.1045 - 15s/epoch - 5ms/step  
Epoch 18/20  
3375/3375 - 15s - loss: 2.3185 - accuracy: 0.1033 - val_loss: 2.3166 -  
val_accuracy: 0.1113 - 15s/epoch - 4ms/step  
Epoch 19/20  
3375/3375 - 15s - loss: 2.3175 - accuracy: 0.1025 - val_loss: 2.3150 -  
val_accuracy: 0.0978 - 15s/epoch - 4ms/step  
Epoch 20/20  
3375/3375 - 15s - loss: 2.3176 - accuracy: 0.1020 - val_loss: 2.3108 -  
val_accuracy: 0.1113 - 15s/epoch - 4ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.1, 16  
batch size and 20 epochs:  
[[ 0  0  0  0  0  0  0  980  0  0]  
 [ 0  0  0  0  0  0  0  1135  0  0]  
 [ 0  0  0  0  0  0  0  1032  0  0]  
 [ 0  0  0  0  0  0  0  1010  0  0]  
 [ 0  0  0  0  0  0  0  982  0  0]]
```

```
[ 0  0  0  0  0  0  0  892  0  0]
[ 0  0  0  0  0  0  0  958  0  0]
[ 0  0  0  0  0  0  0 1028  0  0]
[ 0  0  0  0  0  0  0  974  0  0]
[ 0  0  0  0  0  0  0 1009  0  0]]
```

Precision: 0.0106

Recall: 0.1028

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



Training with nadam optimizer and the learning_rate is 0.01, 64 batch size and 5 epochs...

Epoch 1/5

844/844 - 6s - loss: 0.1976 - accuracy: 0.9487 - val_loss: 0.0681 - val_accuracy: 0.9815 - 6s/epoch - 7ms/step

Epoch 2/5

844/844 - 5s - loss: 0.0593 - accuracy: 0.9811 - val_loss: 0.0796 -

```
val_accuracy: 0.9773 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 0.0447 - accuracy: 0.9866 - val_loss: 0.0620 -
val_accuracy: 0.9818 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 0.0364 - accuracy: 0.9882 - val_loss: 0.0796 -
val_accuracy: 0.9818 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.0341 - accuracy: 0.9892 - val_loss: 0.0687 -
val_accuracy: 0.9853 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 64
batch size and 5 epochs:
[[ 973    0    2    0    0    0    2    1    2    0]
 [  0 1130    1    1    0    0    2    0    1    0]
 [  2    2 1007    4    1    0    0    9    6    1]
 [  1    1    2  989    0    4    0    8    3    2]
 [  2    2    0    0  955    0    4    2    2   15]
 [  4    3    0   11    0  860    5    0    2    7]
 [  9    2    0    0    1    6  929    0    9    2]
 [  0    1    4    0    1    0    0 1013    1    8]
 [  5    0    0    1    1    1    1    3  958    4]
 [  3    1    1    1    2    1    0    3    3  994]]
```

Precision: 0.9809
Recall: 0.9808

Confusion Matrix for nadam optimizer and 0.01 learning rate, 64 batch size and 5 epochs

	0	1	2	3	4	5	6	7	8	9	
True Labels	973	0	2	0	0	0	2	1	2	0	
0	973	0	2	0	0	0	2	1	2	0	
1	0	1130	1	1	0	0	2	0	1	0	
2	2	2	1007	4	1	0	0	9	6	1	
3	1	1	2	989	0	4	0	8	3	2	
4	2	2	0	0	955	0	4	2	2	15	
5	4	3	0	11	0	860	5	0	2	7	
6	9	2	0	0	1	6	929	0	9	2	
7	0	1	4	0	1	0	0	1013	1	8	
8	5	0	0	1	1	1	1	3	958	4	
9	3	1	1	1	2	1	0	3	3	994	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```
Training with nadam optimizer and the learning_rate is 0.01, 64 batch size and 15 epochs...
```

```
Epoch 1/15
```

```
844/844 - 6s - loss: 0.1872 - accuracy: 0.9519 - val_loss: 0.0735 - val_accuracy: 0.9797 - 6s/epoch - 7ms/step
```

```
Epoch 2/15
```

```
844/844 - 5s - loss: 0.0566 - accuracy: 0.9824 - val_loss: 0.0776 - val_accuracy: 0.9775 - 5s/epoch - 6ms/step
```

```
Epoch 3/15
```

```
844/844 - 5s - loss: 0.0374 - accuracy: 0.9881 - val_loss: 0.0659 - val_accuracy: 0.9827 - 5s/epoch - 6ms/step
```

```
Epoch 4/15
```

```
844/844 - 5s - loss: 0.0347 - accuracy: 0.9891 - val_loss: 0.0768 - val_accuracy: 0.9818 - 5s/epoch - 6ms/step
```

```
Epoch 5/15
```

```
844/844 - 5s - loss: 0.0323 - accuracy: 0.9902 - val_loss: 0.0901 - val_accuracy: 0.9807 - 5s/epoch - 5ms/step
```

```
Epoch 6/15
```

```
844/844 - 5s - loss: 0.0311 - accuracy: 0.9909 - val_loss: 0.1015 - val_accuracy: 0.9805 - 5s/epoch - 5ms/step
```

```
Epoch 7/15
```

```
844/844 - 5s - loss: 0.0261 - accuracy: 0.9927 - val_loss: 0.0913 -  
val_accuracy: 0.9857 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.0248 - accuracy: 0.9928 - val_loss: 0.1145 -  
val_accuracy: 0.9815 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.0266 - accuracy: 0.9926 - val_loss: 0.1030 -  
val_accuracy: 0.9812 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.0224 - accuracy: 0.9942 - val_loss: 0.1558 -  
val_accuracy: 0.9787 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.0210 - accuracy: 0.9946 - val_loss: 0.1358 -  
val_accuracy: 0.9813 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.0249 - accuracy: 0.9944 - val_loss: 0.1450 -  
val_accuracy: 0.9805 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.0228 - accuracy: 0.9952 - val_loss: 0.1444 -  
val_accuracy: 0.9830 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.0180 - accuracy: 0.9962 - val_loss: 0.2023 -  
val_accuracy: 0.9807 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.0265 - accuracy: 0.9946 - val_loss: 0.1821 -  
val_accuracy: 0.9837 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 64  
batch size and 15 epochs:  
[[ 977 1 1 0 0 0 0 1 0 0]  
[ 4 1121 2 1 4 0 1 2 0 0]  
[ 4 1 1017 1 2 0 0 4 1 2]  
[ 1 0 1 998 0 5 0 2 1 2]  
[ 1 0 0 0 975 0 2 1 1 2]  
[ 1 0 1 8 0 873 6 0 0 3]  
[ 7 2 1 1 1 5 939 1 1 0]  
[ 0 2 8 4 1 0 0 1009 0 4]  
[ 12 0 2 3 5 1 1 2 942 6]  
[ 0 2 0 2 10 6 0 7 2 980]]  
Precision: 0.9832  
Recall: 0.9831
```

Confusion Matrix for nadam optimizer and 0.01 learning rate, 64 batch size and 15 epochs										
	0	1	2	3	4	5	6	7	8	9
True Labels	977	1	1	0	0	0	0	1	0	0
0	977	1	1	0	0	0	0	1	0	0
1	4	1121	2	1	4	0	1	2	0	0
2	4	1	1017	1	2	0	0	4	1	2
3	1	0	1	998	0	5	0	2	1	2
4	1	0	0	0	975	0	2	1	1	2
5	1	0	1	8	0	873	6	0	0	3
6	7	2	1	1	1	5	939	1	1	0
7	0	2	8	4	1	0	0	1009	0	4
8	12	0	2	3	5	1	1	2	942	6
9	0	2	0	2	10	6	0	7	2	980
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```

Training with nadam optimizer and the learning_rate is 0.01, 64 batch
size and 20 epochs...
Epoch 1/20
844/844 - 6s - loss: 0.1874 - accuracy: 0.9507 - val_loss: 0.0619 -
val_accuracy: 0.9813 - 6s/epoch - 7ms/step
Epoch 2/20
844/844 - 5s - loss: 0.0572 - accuracy: 0.9826 - val_loss: 0.0669 -
val_accuracy: 0.9825 - 5s/epoch - 6ms/step
Epoch 3/20
844/844 - 5s - loss: 0.0442 - accuracy: 0.9864 - val_loss: 0.0669 -
val_accuracy: 0.9820 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: 0.0370 - accuracy: 0.9882 - val_loss: 0.0791 -
val_accuracy: 0.9818 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: 0.0313 - accuracy: 0.9904 - val_loss: 0.0940 -
val_accuracy: 0.9810 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 0.0320 - accuracy: 0.9901 - val_loss: 0.1034 -
val_accuracy: 0.9817 - 5s/epoch - 6ms/step
Epoch 7/20

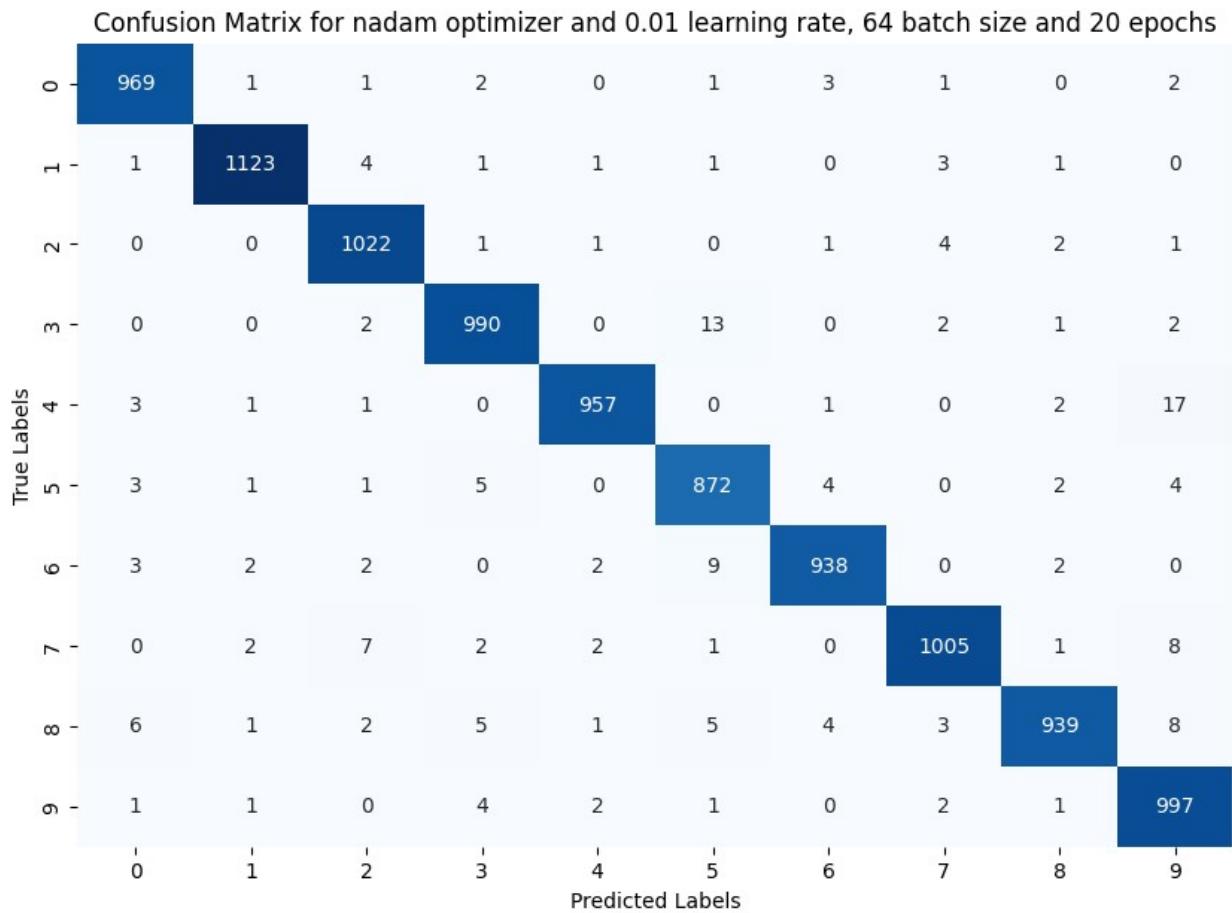
```

```
844/844 - 5s - loss: 0.0249 - accuracy: 0.9923 - val_loss: 0.0930 -  
val_accuracy: 0.9835 - 5s/epoch - 6ms/step  
Epoch 8/20  
844/844 - 5s - loss: 0.0231 - accuracy: 0.9928 - val_loss: 0.0895 -  
val_accuracy: 0.9833 - 5s/epoch - 6ms/step  
Epoch 9/20  
844/844 - 5s - loss: 0.0241 - accuracy: 0.9929 - val_loss: 0.1063 -  
val_accuracy: 0.9835 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: 0.0214 - accuracy: 0.9938 - val_loss: 0.1210 -  
val_accuracy: 0.9818 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: 0.0237 - accuracy: 0.9934 - val_loss: 0.1343 -  
val_accuracy: 0.9848 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 5s - loss: 0.0218 - accuracy: 0.9946 - val_loss: 0.1187 -  
val_accuracy: 0.9823 - 5s/epoch - 6ms/step  
Epoch 13/20  
844/844 - 5s - loss: 0.0191 - accuracy: 0.9951 - val_loss: 0.1824 -  
val_accuracy: 0.9772 - 5s/epoch - 6ms/step  
Epoch 14/20  
844/844 - 5s - loss: 0.0252 - accuracy: 0.9942 - val_loss: 0.1584 -  
val_accuracy: 0.9815 - 5s/epoch - 6ms/step  
Epoch 15/20  
844/844 - 5s - loss: 0.0200 - accuracy: 0.9952 - val_loss: 0.1894 -  
val_accuracy: 0.9820 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: 0.0189 - accuracy: 0.9958 - val_loss: 0.1690 -  
val_accuracy: 0.9810 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: 0.0176 - accuracy: 0.9960 - val_loss: 0.1747 -  
val_accuracy: 0.9830 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: 0.0203 - accuracy: 0.9959 - val_loss: 0.1757 -  
val_accuracy: 0.9827 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: 0.0253 - accuracy: 0.9953 - val_loss: 0.2110 -  
val_accuracy: 0.9835 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: 0.0258 - accuracy: 0.9956 - val_loss: 0.1678 -  
val_accuracy: 0.9840 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 64  
batch size and 20 epochs:  
[[ 969 1 1 2 0 1 3 1 0 2]  
 [ 1 1123 4 1 1 1 0 3 1 0]  
 [ 0 0 1022 1 1 0 1 4 2 1]  
 [ 0 0 2 990 0 13 0 2 1 2]  
 [ 3 1 1 0 957 0 1 0 2 17]]
```

```
[ 3 1 1 5 0 872 4 0 2 4]
[ 3 2 2 0 2 9 938 0 2 0]
[ 0 2 7 2 2 1 0 1005 1 8]
[ 6 1 2 5 1 5 4 3 939 8]
[ 1 1 0 4 2 1 0 2 1 997]]
```

Precision: 0.9813

Recall: 0.9812



Training with nadam optimizer and the learning_rate is 0.01, 128 batch size and 5 epochs...

Epoch 1/5

422/422 - 5s - loss: 0.2437 - accuracy: 0.9417 - val_loss: 0.0705 - val_accuracy: 0.9812 - 5s/epoch - 11ms/step

Epoch 2/5

422/422 - 3s - loss: 0.0559 - accuracy: 0.9828 - val_loss: 0.0588 - val_accuracy: 0.9850 - 3s/epoch - 8ms/step

Epoch 3/5

422/422 - 3s - loss: 0.0352 - accuracy: 0.9890 - val_loss: 0.0602 - val_accuracy: 0.9843 - 3s/epoch - 8ms/step

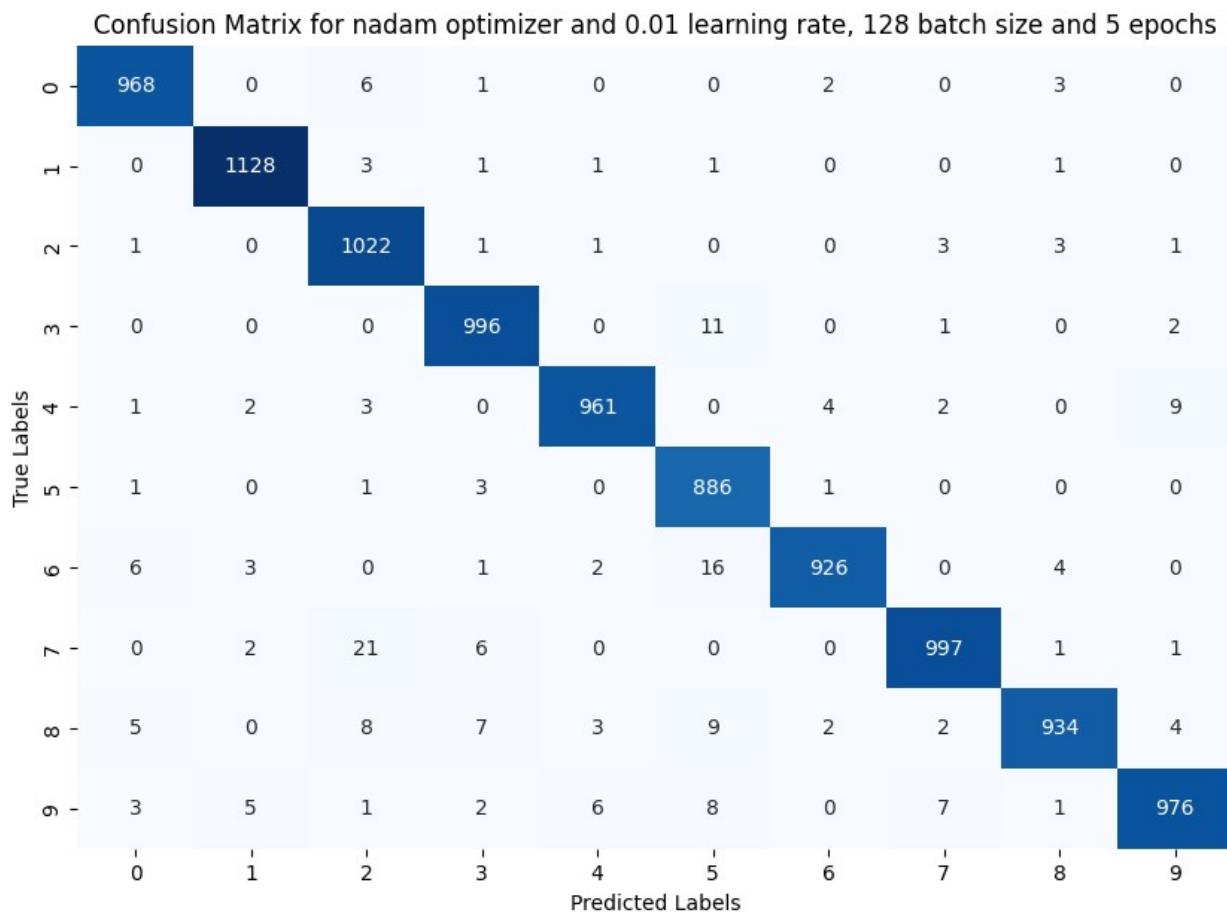
Epoch 4/5

422/422 - 3s - loss: 0.0284 - accuracy: 0.9902 - val_loss: 0.0634 -

```

val_accuracy: 0.9830 - 3s/epoch - 8ms/step
Epoch 5/5
422/422 - 3s - loss: 0.0227 - accuracy: 0.9924 - val_loss: 0.0559 -
val_accuracy: 0.9843 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 128
batch size and 5 epochs:
[[ 968 0 6 1 0 0 2 0 3 0]
 [ 0 1128 3 1 1 1 0 0 1 0]
 [ 1 0 1022 1 1 0 0 3 3 1]
 [ 0 0 0 996 0 11 0 1 0 2]
 [ 1 2 3 0 961 0 4 2 0 9]
 [ 1 0 1 3 0 886 1 0 0 0]
 [ 6 3 0 1 2 16 926 0 4 0]
 [ 0 2 21 6 0 0 0 997 1 1]
 [ 5 0 8 7 3 9 2 2 934 4]
 [ 3 5 1 2 6 8 0 7 1 976]]
Precision: 0.9796
Recall: 0.9794

```

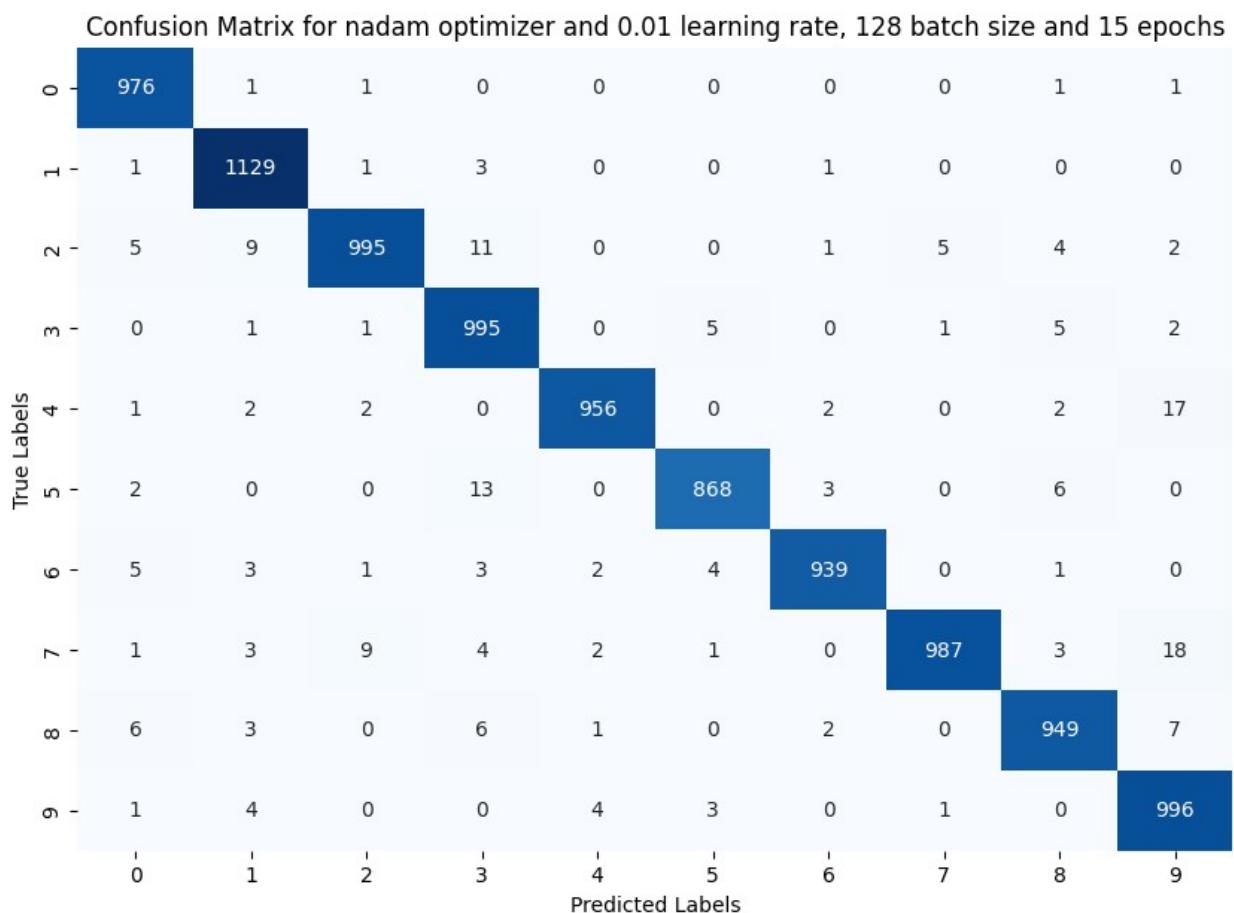


```
Training with nadam optimizer and the learning_rate is 0.01, 128 batch size and 15 epochs...
Epoch 1/15
422/422 - 5s - loss: 0.2669 - accuracy: 0.9410 - val_loss: 0.0755 -
val_accuracy: 0.9775 - 5s/epoch - 11ms/step
Epoch 2/15
422/422 - 3s - loss: 0.0563 - accuracy: 0.9822 - val_loss: 0.0508 -
val_accuracy: 0.9868 - 3s/epoch - 8ms/step
Epoch 3/15
422/422 - 3s - loss: 0.0367 - accuracy: 0.9884 - val_loss: 0.0593 -
val_accuracy: 0.9842 - 3s/epoch - 8ms/step
Epoch 4/15
422/422 - 3s - loss: 0.0257 - accuracy: 0.9914 - val_loss: 0.0631 -
val_accuracy: 0.9852 - 3s/epoch - 8ms/step
Epoch 5/15
422/422 - 3s - loss: 0.0209 - accuracy: 0.9929 - val_loss: 0.0850 -
val_accuracy: 0.9820 - 3s/epoch - 8ms/step
Epoch 6/15
422/422 - 3s - loss: 0.0213 - accuracy: 0.9928 - val_loss: 0.0841 -
val_accuracy: 0.9832 - 3s/epoch - 8ms/step
Epoch 7/15
422/422 - 3s - loss: 0.0181 - accuracy: 0.9940 - val_loss: 0.0928 -
val_accuracy: 0.9822 - 3s/epoch - 8ms/step
Epoch 8/15
422/422 - 3s - loss: 0.0180 - accuracy: 0.9941 - val_loss: 0.0781 -
val_accuracy: 0.9860 - 3s/epoch - 8ms/step
Epoch 9/15
422/422 - 3s - loss: 0.0175 - accuracy: 0.9942 - val_loss: 0.0818 -
val_accuracy: 0.9843 - 3s/epoch - 8ms/step
Epoch 10/15
422/422 - 3s - loss: 0.0173 - accuracy: 0.9945 - val_loss: 0.0781 -
val_accuracy: 0.9850 - 3s/epoch - 8ms/step
Epoch 11/15
422/422 - 3s - loss: 0.0136 - accuracy: 0.9954 - val_loss: 0.0940 -
val_accuracy: 0.9860 - 3s/epoch - 8ms/step
Epoch 12/15
422/422 - 3s - loss: 0.0155 - accuracy: 0.9954 - val_loss: 0.1147 -
val_accuracy: 0.9802 - 3s/epoch - 8ms/step
Epoch 13/15
422/422 - 3s - loss: 0.0169 - accuracy: 0.9949 - val_loss: 0.1553 -
val_accuracy: 0.9788 - 3s/epoch - 8ms/step
Epoch 14/15
422/422 - 3s - loss: 0.0176 - accuracy: 0.9951 - val_loss: 0.1221 -
val_accuracy: 0.9838 - 3s/epoch - 8ms/step
Epoch 15/15
422/422 - 3s - loss: 0.0162 - accuracy: 0.9952 - val_loss: 0.1456 -
val_accuracy: 0.9813 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 128 batch size and 15 epochs:
```

```
[[ 976  1   1   0   0   0   0   0   1   1
  [ 1 1129  1   3   0   0   1   0   0   0
  [ 5   9 995  11  0   0   1   5   4   2
  [ 0   1 1   995  0   5   0   1   5   2
  [ 1   2 2   0 956  0   2   0   2   17
  [ 2   0 0   13  0 868  3   0   6   0
  [ 5   3 1   3  2   4 939  0   1   0
  [ 1   3 9   4  2   1 0   987  3   18
  [ 6   3 0   6  1   0 2   0 949  7
  [ 1   4 0   0  4   3 0   1 0 996]]
```

Precision: 0.9792

Recall: 0.9790



Training with nadam optimizer and the learning_rate is 0.01, 128 batch size and 20 epochs...

Epoch 1/20

422/422 - 5s - loss: 0.1579 - accuracy: 0.9544 - val_loss: 0.0686 - val_accuracy: 0.9795 - 5s/epoch - 11ms/step

Epoch 2/20

422/422 - 3s - loss: 0.0449 - accuracy: 0.9861 - val_loss: 0.0661 - val_accuracy: 0.9822 - 3s/epoch - 8ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.0314 - accuracy: 0.9900 - val_loss: 0.0585 -
val_accuracy: 0.9848 - 3s/epoch - 8ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0247 - accuracy: 0.9923 - val_loss: 0.0711 -
val_accuracy: 0.9835 - 3s/epoch - 8ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0202 - accuracy: 0.9936 - val_loss: 0.0951 -
val_accuracy: 0.9793 - 3s/epoch - 8ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0218 - accuracy: 0.9930 - val_loss: 0.0749 -
val_accuracy: 0.9835 - 3s/epoch - 8ms/step
Epoch 7/20
422/422 - 3s - loss: 0.0174 - accuracy: 0.9947 - val_loss: 0.0909 -
val_accuracy: 0.9842 - 3s/epoch - 7ms/step
Epoch 8/20
422/422 - 3s - loss: 0.0216 - accuracy: 0.9934 - val_loss: 0.0895 -
val_accuracy: 0.9828 - 3s/epoch - 7ms/step
Epoch 9/20
422/422 - 3s - loss: 0.0206 - accuracy: 0.9936 - val_loss: 0.0925 -
val_accuracy: 0.9813 - 3s/epoch - 8ms/step
Epoch 10/20
422/422 - 3s - loss: 0.0164 - accuracy: 0.9948 - val_loss: 0.0870 -
val_accuracy: 0.9850 - 3s/epoch - 8ms/step
Epoch 11/20
422/422 - 3s - loss: 0.0136 - accuracy: 0.9961 - val_loss: 0.1085 -
val_accuracy: 0.9847 - 3s/epoch - 8ms/step
Epoch 12/20
422/422 - 3s - loss: 0.0181 - accuracy: 0.9951 - val_loss: 0.1034 -
val_accuracy: 0.9860 - 3s/epoch - 8ms/step
Epoch 13/20
422/422 - 3s - loss: 0.0125 - accuracy: 0.9964 - val_loss: 0.1620 -
val_accuracy: 0.9792 - 3s/epoch - 8ms/step
Epoch 14/20
422/422 - 3s - loss: 0.0181 - accuracy: 0.9957 - val_loss: 0.1558 -
val_accuracy: 0.9820 - 3s/epoch - 8ms/step
Epoch 15/20
422/422 - 3s - loss: 0.0206 - accuracy: 0.9954 - val_loss: 0.1340 -
val_accuracy: 0.9835 - 3s/epoch - 8ms/step
Epoch 16/20
422/422 - 3s - loss: 0.0156 - accuracy: 0.9961 - val_loss: 0.1437 -
val_accuracy: 0.9828 - 3s/epoch - 8ms/step
Epoch 17/20
422/422 - 3s - loss: 0.0141 - accuracy: 0.9968 - val_loss: 0.1544 -
val_accuracy: 0.9850 - 3s/epoch - 8ms/step
Epoch 18/20
422/422 - 3s - loss: 0.0187 - accuracy: 0.9961 - val_loss: 0.1513 -
val_accuracy: 0.9840 - 3s/epoch - 8ms/step
Epoch 19/20
```

```
422/422 - 3s - loss: 0.0191 - accuracy: 0.9960 - val_loss: 0.1634 -  
val_accuracy: 0.9843 - 3s/epoch - 8ms/step
```

```
Epoch 20/20
```

```
422/422 - 3s - loss: 0.0125 - accuracy: 0.9972 - val_loss: 0.1756 -  
val_accuracy: 0.9837 - 3s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

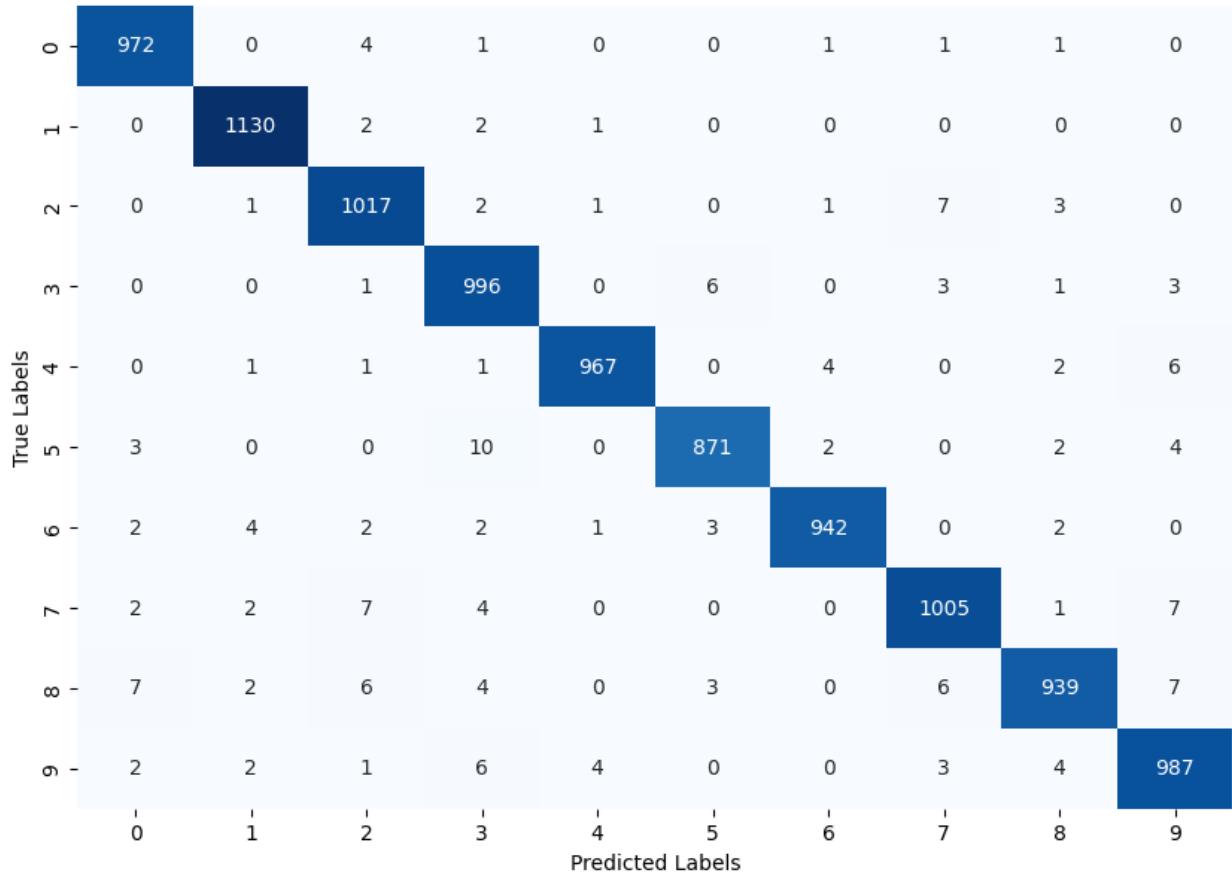
```
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 128  
batch size and 20 epochs:
```

```
[[ 972   0   4   1   0   0   1   1   1   0]  
[  0 1130   2   2   1   0   0   0   0   0]  
[  0   1 1017   2   1   0   1   7   3   0]  
[  0   0   1 996   0   6   0   3   1   3]  
[  0   1   1   1 967   0   4   0   2   6]  
[  3   0   0   10   0 871   2   0   2   4]  
[  2   4   2   2   1   3 942   0   2   0]  
[  2   2   7   4   0   0   0 1005   1   7]  
[  7   2   6   4   0   3   0   6 939   7]  
[  2   2   1   6   4   0   0   3   4 987]]
```

```
Precision: 0.9826
```

```
Recall: 0.9826
```

Confusion Matrix for nadam optimizer and 0.01 learning rate, 128 batch size and 20 epochs



```
Training with nadam optimizer and the learning_rate is 0.01, 256 batch size and 5 epochs...
Epoch 1/5
211/211 - 4s - loss: 0.2466 - accuracy: 0.9364 - val_loss: 0.0732 -
val_accuracy: 0.9790 - 4s/epoch - 18ms/step
Epoch 2/5
211/211 - 3s - loss: 0.0557 - accuracy: 0.9836 - val_loss: 0.0572 -
val_accuracy: 0.9840 - 3s/epoch - 12ms/step
Epoch 3/5
211/211 - 3s - loss: 0.0348 - accuracy: 0.9887 - val_loss: 0.0566 -
val_accuracy: 0.9852 - 3s/epoch - 12ms/step
Epoch 4/5
211/211 - 3s - loss: 0.0232 - accuracy: 0.9926 - val_loss: 0.0585 -
val_accuracy: 0.9858 - 3s/epoch - 12ms/step
Epoch 5/5
211/211 - 3s - loss: 0.0175 - accuracy: 0.9939 - val_loss: 0.0611 -
val_accuracy: 0.9865 - 3s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 256 batch size and 5 epochs:
[[ 973    0    2    0    0    0    2    1    2    0]
 [  0 1126    2    3    1    1    2    0    0    0]
 [  3    2 1014    3    1    0    1    2    5    1]
 [  0    0    1  996    0    5    0    1    1    6]
 [  0    0    0    0  970    0    3    0    1    8]
 [  1    0    1    4    0  881    4    0    0    1]
 [  6    3    1    2    2    1  941    0    2    0]
 [  0    6   17    2    2    0    0  991    5    5]
 [  5    0    1    2    2    1    1    1  957    4]
 [  2    1    0    3    6    3    0    2    1  991]]
```

Precision: 0.9840
Recall: 0.9840

Confusion Matrix for nadam optimizer and 0.01 learning rate, 256 batch size and 5 epochs

	0	1	2	3	4	5	6	7	8	9	
True Labels	973	0	2	0	0	0	2	1	2	0	0
0	973	0	2	0	0	0	2	1	2	0	0
1	0	1126	2	3	1	1	2	0	0	0	0
2	3	2	1014	3	1	0	1	2	5	1	
3	0	0	1	996	0	5	0	1	1	6	
4	0	0	0	0	970	0	3	0	1	8	
5	1	0	1	4	0	881	4	0	0	1	
6	6	3	1	2	2	1	941	0	2	0	
7	0	6	17	2	2	0	0	991	5	5	
8	5	0	1	2	2	1	1	1	957	4	
9	2	1	0	3	6	3	0	2	1	991	
	0	1	2	3	4	5	6	7	8	9	
Predicted Labels											

```
Training with nadam optimizer and the learning_rate is 0.01, 256 batch size and 15 epochs...
```

```
Epoch 1/15
```

```
211/211 - 4s - loss: 0.4191 - accuracy: 0.9163 - val_loss: 0.0802 - val_accuracy: 0.9803 - 4s/epoch - 18ms/step
```

```
Epoch 2/15
```

```
211/211 - 3s - loss: 0.0669 - accuracy: 0.9791 - val_loss: 0.0737 - val_accuracy: 0.9787 - 3s/epoch - 12ms/step
```

```
Epoch 3/15
```

```
211/211 - 3s - loss: 0.0455 - accuracy: 0.9860 - val_loss: 0.0615 - val_accuracy: 0.9845 - 3s/epoch - 12ms/step
```

```
Epoch 4/15
```

```
211/211 - 3s - loss: 0.0339 - accuracy: 0.9891 - val_loss: 0.0554 - val_accuracy: 0.9840 - 3s/epoch - 12ms/step
```

```
Epoch 5/15
```

```
211/211 - 3s - loss: 0.0240 - accuracy: 0.9918 - val_loss: 0.0600 - val_accuracy: 0.9840 - 3s/epoch - 12ms/step
```

```
Epoch 6/15
```

```
211/211 - 3s - loss: 0.0200 - accuracy: 0.9936 - val_loss: 0.0760 - val_accuracy: 0.9808 - 3s/epoch - 12ms/step
```

```
Epoch 7/15
```

```
211/211 - 2s - loss: 0.0156 - accuracy: 0.9948 - val_loss: 0.0672 -  
val_accuracy: 0.9837 - 2s/epoch - 12ms/step  
Epoch 8/15  
211/211 - 3s - loss: 0.0124 - accuracy: 0.9958 - val_loss: 0.0759 -  
val_accuracy: 0.9812 - 3s/epoch - 12ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.0129 - accuracy: 0.9955 - val_loss: 0.0753 -  
val_accuracy: 0.9827 - 3s/epoch - 12ms/step  
Epoch 10/15  
211/211 - 3s - loss: 0.0120 - accuracy: 0.9958 - val_loss: 0.0772 -  
val_accuracy: 0.9845 - 3s/epoch - 12ms/step  
Epoch 11/15  
211/211 - 3s - loss: 0.0117 - accuracy: 0.9961 - val_loss: 0.0911 -  
val_accuracy: 0.9800 - 3s/epoch - 12ms/step  
Epoch 12/15  
211/211 - 3s - loss: 0.0128 - accuracy: 0.9954 - val_loss: 0.0691 -  
val_accuracy: 0.9850 - 3s/epoch - 12ms/step  
Epoch 13/15  
211/211 - 3s - loss: 0.0091 - accuracy: 0.9969 - val_loss: 0.1033 -  
val_accuracy: 0.9820 - 3s/epoch - 13ms/step  
Epoch 14/15  
211/211 - 3s - loss: 0.0097 - accuracy: 0.9969 - val_loss: 0.0880 -  
val_accuracy: 0.9842 - 3s/epoch - 12ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.0091 - accuracy: 0.9967 - val_loss: 0.0952 -  
val_accuracy: 0.9840 - 3s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 256  
batch size and 15 epochs:  
[[ 974 1 2 0 0 0 1 1 1 0]  
[ 0 1128 2 1 1 0 3 0 0 0]  
[ 3 2 994 5 1 0 1 21 5 0]  
[ 0 0 1 1002 0 1 0 3 3 0]  
[ 0 0 0 0 969 1 3 0 2 7]  
[ 2 0 0 9 0 871 5 0 3 2]  
[ 5 2 0 2 3 1 945 0 0 0]  
[ 0 4 4 2 0 0 0 1017 1 0]  
[ 3 0 4 1 4 0 2 7 950 3]  
[ 0 7 1 8 7 9 0 11 1 965]]  
Precision: 0.9816  
Recall: 0.9815
```

Confusion Matrix for nadam optimizer and 0.01 learning rate, 256 batch size and 15 epochs

	0	1	2	3	4	5	6	7	8	9	
True Labels	974	1	2	0	0	0	1	1	1	0	0
0	0	1128	2	1	1	0	3	0	0	0	0
1	3	2	994	5	1	0	1	21	5	0	0
2	0	0	1	1002	0	1	0	3	3	0	0
3	0	0	0	0	969	1	3	0	2	7	0
4	2	0	0	9	0	871	5	0	3	2	0
5	5	2	0	2	3	1	945	0	0	0	0
6	0	4	4	2	0	0	0	0	1017	1	0
7	3	0	4	1	4	0	2	7	950	3	0
8	0	7	1	8	7	9	0	11	1	965	0
9	0	1	2	3	4	5	6	7	8	9	0
Predicted Labels											

```
Training with nadam optimizer and the learning_rate is 0.01, 256 batch size and 20 epochs...
Epoch 1/20
211/211 - 4s - loss: 0.3169 - accuracy: 0.9279 - val_loss: 0.0708 -
val_accuracy: 0.9792 - 4s/epoch - 18ms/step
Epoch 2/20
211/211 - 3s - loss: 0.0579 - accuracy: 0.9814 - val_loss: 0.0606 -
val_accuracy: 0.9838 - 3s/epoch - 12ms/step
Epoch 3/20
211/211 - 3s - loss: 0.0338 - accuracy: 0.9892 - val_loss: 0.0546 -
val_accuracy: 0.9840 - 3s/epoch - 12ms/step
Epoch 4/20
211/211 - 3s - loss: 0.0244 - accuracy: 0.9921 - val_loss: 0.0553 -
val_accuracy: 0.9847 - 3s/epoch - 12ms/step
Epoch 5/20
211/211 - 3s - loss: 0.0169 - accuracy: 0.9943 - val_loss: 0.0709 -
val_accuracy: 0.9837 - 3s/epoch - 12ms/step
Epoch 6/20
211/211 - 3s - loss: 0.0167 - accuracy: 0.9941 - val_loss: 0.0702 -
val_accuracy: 0.9835 - 3s/epoch - 12ms/step
Epoch 7/20
```

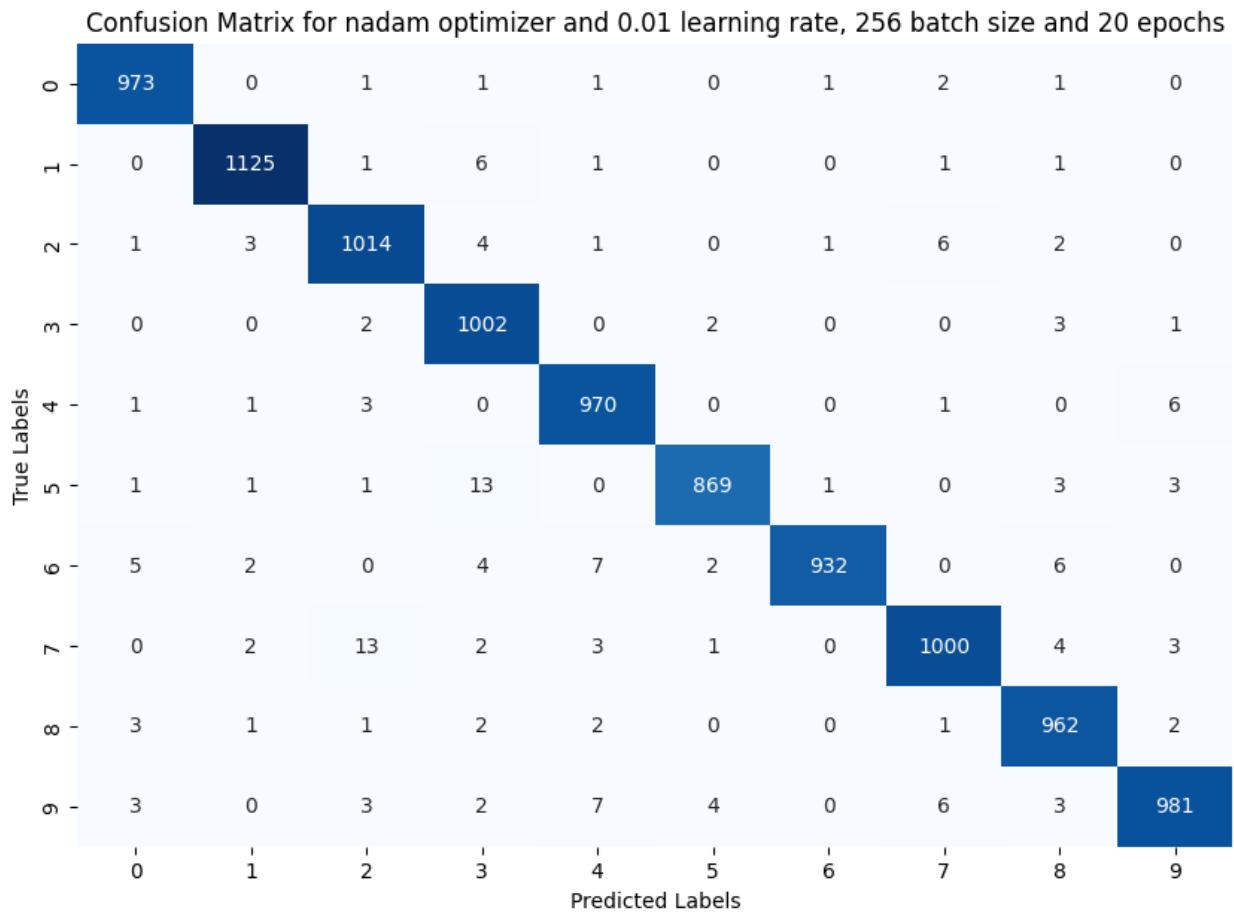
```

211/211 - 3s - loss: 0.0130 - accuracy: 0.9960 - val_loss: 0.0762 -
val_accuracy: 0.9842 - 3s/epoch - 12ms/step
Epoch 8/20
211/211 - 3s - loss: 0.0119 - accuracy: 0.9957 - val_loss: 0.0811 -
val_accuracy: 0.9845 - 3s/epoch - 12ms/step
Epoch 9/20
211/211 - 3s - loss: 0.0127 - accuracy: 0.9958 - val_loss: 0.0912 -
val_accuracy: 0.9842 - 3s/epoch - 12ms/step
Epoch 10/20
211/211 - 3s - loss: 0.0109 - accuracy: 0.9962 - val_loss: 0.1156 -
val_accuracy: 0.9807 - 3s/epoch - 12ms/step
Epoch 11/20
211/211 - 3s - loss: 0.0110 - accuracy: 0.9961 - val_loss: 0.0766 -
val_accuracy: 0.9863 - 3s/epoch - 12ms/step
Epoch 12/20
211/211 - 3s - loss: 0.0065 - accuracy: 0.9974 - val_loss: 0.0962 -
val_accuracy: 0.9867 - 3s/epoch - 12ms/step
Epoch 13/20
211/211 - 3s - loss: 0.0090 - accuracy: 0.9972 - val_loss: 0.0908 -
val_accuracy: 0.9855 - 3s/epoch - 12ms/step
Epoch 14/20
211/211 - 3s - loss: 0.0094 - accuracy: 0.9968 - val_loss: 0.1145 -
val_accuracy: 0.9850 - 3s/epoch - 12ms/step
Epoch 15/20
211/211 - 3s - loss: 0.0100 - accuracy: 0.9965 - val_loss: 0.1069 -
val_accuracy: 0.9855 - 3s/epoch - 12ms/step
Epoch 16/20
211/211 - 3s - loss: 0.0141 - accuracy: 0.9955 - val_loss: 0.1214 -
val_accuracy: 0.9830 - 3s/epoch - 12ms/step
Epoch 17/20
211/211 - 3s - loss: 0.0102 - accuracy: 0.9964 - val_loss: 0.1084 -
val_accuracy: 0.9857 - 3s/epoch - 12ms/step
Epoch 18/20
211/211 - 3s - loss: 0.0061 - accuracy: 0.9976 - val_loss: 0.1185 -
val_accuracy: 0.9845 - 3s/epoch - 12ms/step
Epoch 19/20
211/211 - 3s - loss: 0.0070 - accuracy: 0.9977 - val_loss: 0.1516 -
val_accuracy: 0.9845 - 3s/epoch - 12ms/step
Epoch 20/20
211/211 - 3s - loss: 0.0106 - accuracy: 0.9969 - val_loss: 0.1315 -
val_accuracy: 0.9842 - 3s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 256
batch size and 20 epochs:
[[ 973   0   1   1   1   0   1   2   1   0]
 [  0 1125   1   6   1   0   0   1   1   0]
 [  1   3 1014   4   1   0   1   6   2   0]
 [  0   0   2 1002   0   2   0   0   3   1]
 [  1   1   3   0  970   0   0   1   0   6]]
```

```
[ 1  1  1 13  0 869  1  0  3  3]
[ 5  2  0  4  7  2 932  0  6  0]
[ 0  2 13  2  3  1  0 1000  4  3]
[ 3  1  1  2  2  0  0  1 962  2]
[ 3  0  3  2  7  4  0  6  3 981]]
```

Precision: 0.9829

Recall: 0.9828



Training with nadam optimizer and the learning_rate is 0.01, 16 batch size and 5 epochs...

Epoch 1/5

3375/3375 - 16s - loss: 0.1706 - accuracy: 0.9520 - val_loss: 0.1067 - val_accuracy: 0.9710 - 16s/epoch - 5ms/step

Epoch 2/5

3375/3375 - 15s - loss: 0.0922 - accuracy: 0.9731 - val_loss: 0.1016 - val_accuracy: 0.9770 - 15s/epoch - 4ms/step

Epoch 3/5

3375/3375 - 15s - loss: 0.0777 - accuracy: 0.9788 - val_loss: 0.1001 - val_accuracy: 0.9743 - 15s/epoch - 4ms/step

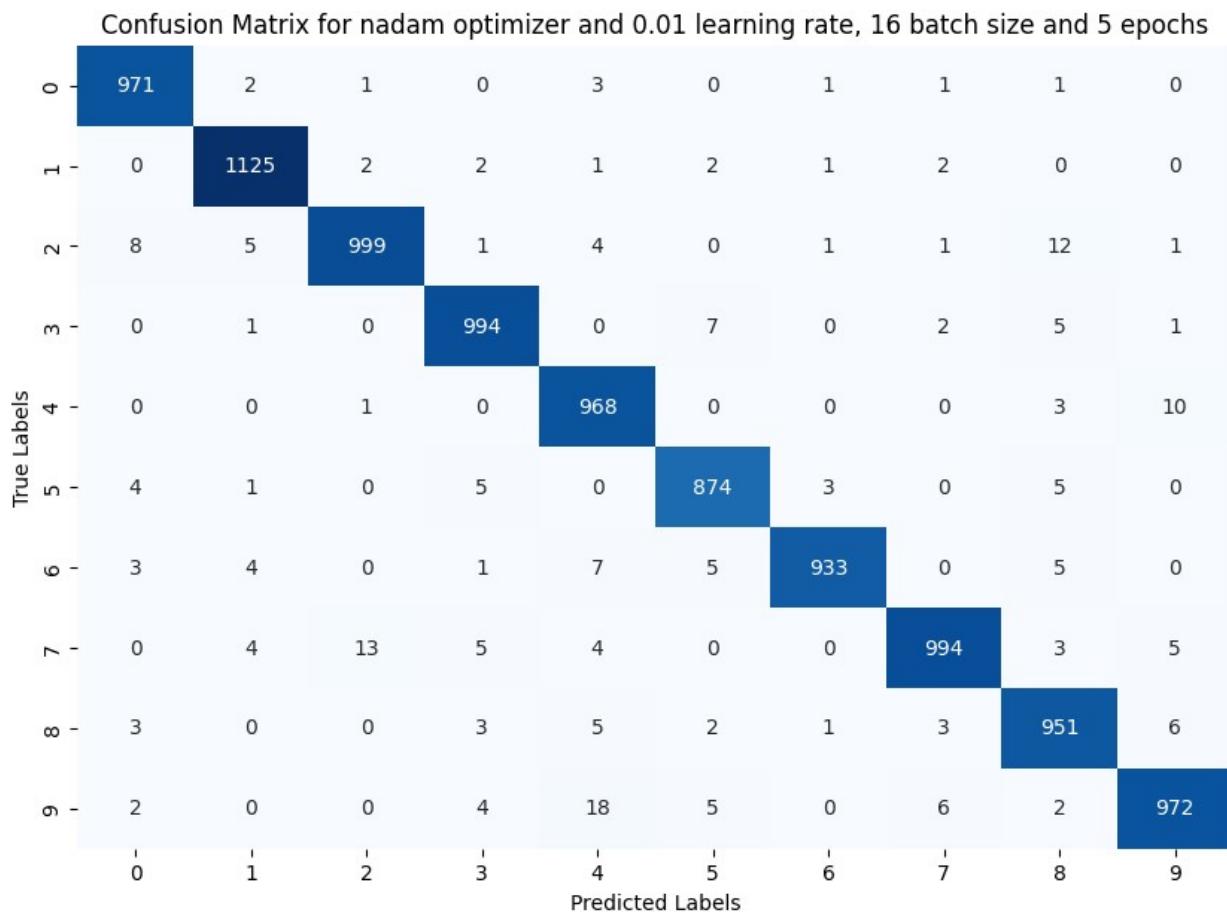
Epoch 4/5

3375/3375 - 15s - loss: 0.0704 - accuracy: 0.9816 - val_loss: 0.1138 -

```

val_accuracy: 0.9758 - 15s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 15s - loss: 0.0626 - accuracy: 0.9849 - val_loss: 0.1062 -
val_accuracy: 0.9787 - 15s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.01, 16
batch size and 5 epochs:
[[ 971   2   1   0   3   0   1   1   1   0]
 [ 0 1125   2   2   1   2   1   2   0   0]
 [ 8   5 999   1   4   0   1   1 12   1]
 [ 0   1   0 994   0   7   0   2   5   1]
 [ 0   0   1   0 968   0   0   0   3 10]
 [ 4   1   0   5   0 874   3   0   5   0]
 [ 3   4   0   1   7   5 933   0   5   0]
 [ 0   4 13   5   4   0   0 994   3   5]
 [ 3   0   0   3   5   2   1   3 951   6]
 [ 2   0   0   4 18   5   0   6   2 972]]
Precision: 0.9782
Recall: 0.9781

```



Training with nadam optimizer and the learning_rate is 0.01, 16 batch size and 15 epochs...

Epoch 1/15
3375/3375 - 16s - loss: 0.1709 - accuracy: 0.9523 - val_loss: 0.1078 - val_accuracy: 0.9703 - 16s/epoch - 5ms/step

Epoch 2/15
3375/3375 - 15s - loss: 0.0902 - accuracy: 0.9743 - val_loss: 0.0849 - val_accuracy: 0.9777 - 15s/epoch - 5ms/step

Epoch 3/15
3375/3375 - 15s - loss: 0.0754 - accuracy: 0.9798 - val_loss: 0.1201 - val_accuracy: 0.9772 - 15s/epoch - 4ms/step

Epoch 4/15
3375/3375 - 15s - loss: 0.0620 - accuracy: 0.9833 - val_loss: 0.1058 - val_accuracy: 0.9792 - 15s/epoch - 4ms/step

Epoch 5/15
3375/3375 - 15s - loss: 0.0591 - accuracy: 0.9855 - val_loss: 0.1849 - val_accuracy: 0.9762 - 15s/epoch - 4ms/step

Epoch 6/15
3375/3375 - 15s - loss: 0.0638 - accuracy: 0.9860 - val_loss: 0.1578 - val_accuracy: 0.9763 - 15s/epoch - 4ms/step

Epoch 7/15
3375/3375 - 15s - loss: 0.0567 - accuracy: 0.9878 - val_loss: 0.1754 - val_accuracy: 0.9778 - 15s/epoch - 4ms/step

Epoch 8/15
3375/3375 - 15s - loss: 0.0578 - accuracy: 0.9886 - val_loss: 0.1819 - val_accuracy: 0.9775 - 15s/epoch - 4ms/step

Epoch 9/15
3375/3375 - 15s - loss: 0.0564 - accuracy: 0.9902 - val_loss: 0.1678 - val_accuracy: 0.9800 - 15s/epoch - 4ms/step

Epoch 10/15
3375/3375 - 15s - loss: 0.0513 - accuracy: 0.9915 - val_loss: 0.2446 - val_accuracy: 0.9742 - 15s/epoch - 5ms/step

Epoch 11/15
3375/3375 - 15s - loss: 0.0645 - accuracy: 0.9912 - val_loss: 0.2466 - val_accuracy: 0.9827 - 15s/epoch - 4ms/step

Epoch 12/15
3375/3375 - 15s - loss: 0.0596 - accuracy: 0.9915 - val_loss: 0.2752 - val_accuracy: 0.9827 - 15s/epoch - 4ms/step

Epoch 13/15
3375/3375 - 15s - loss: 0.0505 - accuracy: 0.9926 - val_loss: 0.2695 - val_accuracy: 0.9800 - 15s/epoch - 4ms/step

Epoch 14/15
3375/3375 - 15s - loss: 0.0663 - accuracy: 0.9921 - val_loss: 0.3273 - val_accuracy: 0.9777 - 15s/epoch - 4ms/step

Epoch 15/15
3375/3375 - 15s - loss: 0.0546 - accuracy: 0.9939 - val_loss: 0.3222 - val_accuracy: 0.9797 - 15s/epoch - 4ms/step

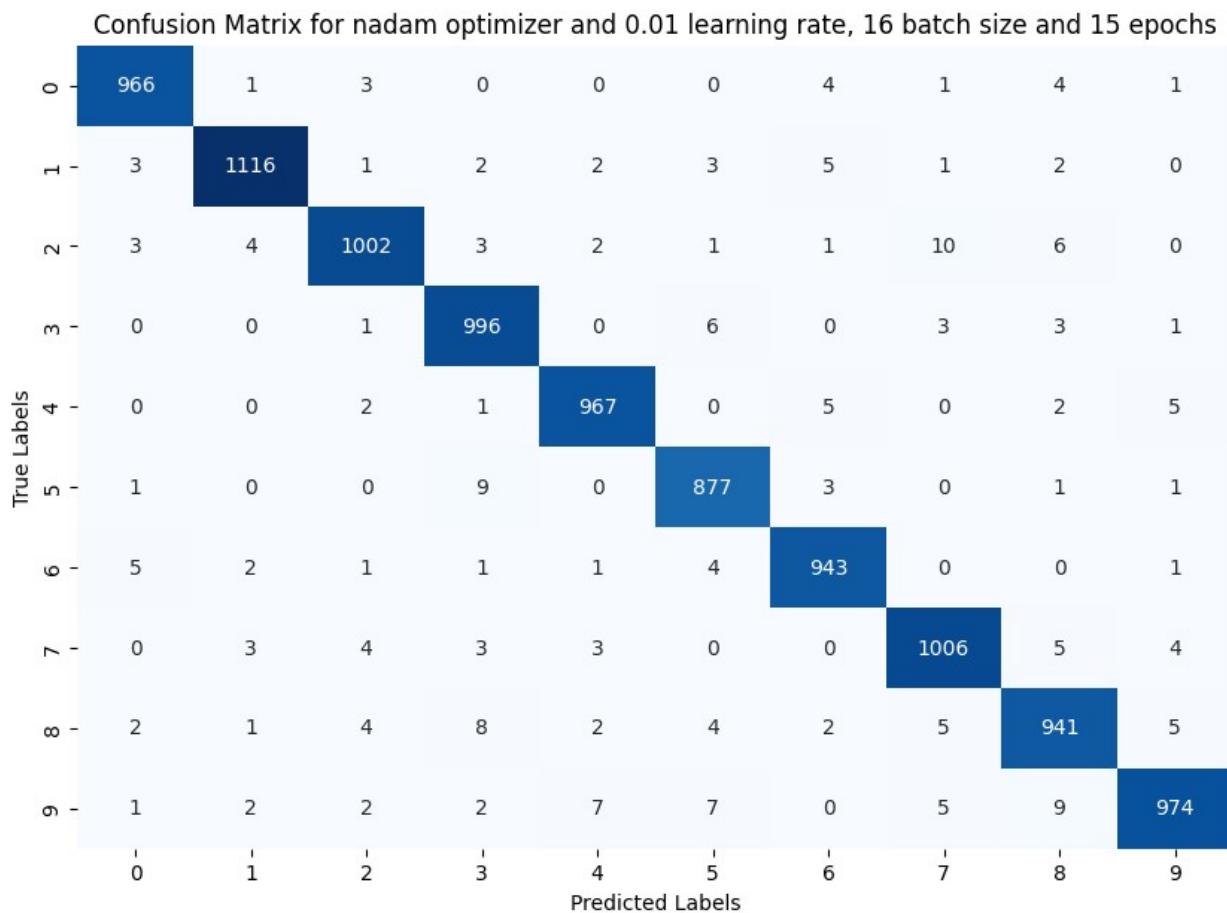
313/313 [=====] - 1s 2ms/step

Confusion Matrix nadam optimizer and the learning_rate is 0.01, 16 batch size and 15 epochs:

```
[[ 966  1   3   0   0   0   4   1   4   1]
 [ 3 1116  1   2   2   3   5   1   2   0]
 [ 3   4 1002  3   2   1   1   10  6   0]
 [ 0   0   1 996  0   6   0   3   3   1]
 [ 0   0   2   1 967  0   5   0   2   5]
 [ 1   0   0   9   0 877  3   0   1   1]
 [ 5   2   1   1   1   4 943  0   0   1]
 [ 0   3   4   3   3   0   0 1006  5   4]
 [ 2   1   4   8   2   4   2   5 941  5]
 [ 1   2   2   2   7   7   0   5   9 974]]
```

Precision: 0.9788

Recall: 0.9788



Training with nadam optimizer and the learning_rate is 0.01, 16 batch size and 20 epochs...

Epoch 1/20

3375/3375 - 16s - loss: 0.1733 - accuracy: 0.9505 - val_loss: 0.0847 - val_accuracy: 0.9757 - 16s/epoch - 5ms/step

Epoch 2/20

3375/3375 - 14s - loss: 0.0917 - accuracy: 0.9729 - val_loss: 0.1174 - val_accuracy: 0.9708 - 14s/epoch - 4ms/step

```
Epoch 3/20
3375/3375 - 15s - loss: 0.0809 - accuracy: 0.9770 - val_loss: 0.0877 -
val_accuracy: 0.9790 - 15s/epoch - 4ms/step
Epoch 4/20
3375/3375 - 15s - loss: 0.0676 - accuracy: 0.9813 - val_loss: 0.0965 -
val_accuracy: 0.9787 - 15s/epoch - 4ms/step
Epoch 5/20
3375/3375 - 15s - loss: 0.0578 - accuracy: 0.9843 - val_loss: 0.0974 -
val_accuracy: 0.9783 - 15s/epoch - 4ms/step
Epoch 6/20
3375/3375 - 15s - loss: 0.0583 - accuracy: 0.9856 - val_loss: 0.1284 -
val_accuracy: 0.9785 - 15s/epoch - 4ms/step
Epoch 7/20
3375/3375 - 15s - loss: 0.0550 - accuracy: 0.9867 - val_loss: 0.1301 -
val_accuracy: 0.9803 - 15s/epoch - 4ms/step
Epoch 8/20
3375/3375 - 14s - loss: 0.0507 - accuracy: 0.9888 - val_loss: 0.1653 -
val_accuracy: 0.9795 - 14s/epoch - 4ms/step
Epoch 9/20
3375/3375 - 15s - loss: 0.0524 - accuracy: 0.9886 - val_loss: 0.1763 -
val_accuracy: 0.9808 - 15s/epoch - 4ms/step
Epoch 10/20
3375/3375 - 14s - loss: 0.0520 - accuracy: 0.9896 - val_loss: 0.2153 -
val_accuracy: 0.9770 - 14s/epoch - 4ms/step
Epoch 11/20
3375/3375 - 15s - loss: 0.0553 - accuracy: 0.9904 - val_loss: 0.2216 -
val_accuracy: 0.9763 - 15s/epoch - 4ms/step
Epoch 12/20
3375/3375 - 15s - loss: 0.0518 - accuracy: 0.9909 - val_loss: 0.2444 -
val_accuracy: 0.9795 - 15s/epoch - 4ms/step
Epoch 13/20
3375/3375 - 14s - loss: 0.0507 - accuracy: 0.9913 - val_loss: 0.2231 -
val_accuracy: 0.9818 - 14s/epoch - 4ms/step
Epoch 14/20
3375/3375 - 15s - loss: 0.0500 - accuracy: 0.9916 - val_loss: 0.2827 -
val_accuracy: 0.9808 - 15s/epoch - 4ms/step
Epoch 15/20
3375/3375 - 15s - loss: 0.0514 - accuracy: 0.9921 - val_loss: 0.2601 -
val_accuracy: 0.9830 - 15s/epoch - 4ms/step
Epoch 16/20
3375/3375 - 15s - loss: 0.0513 - accuracy: 0.9928 - val_loss: 0.2643 -
val_accuracy: 0.9808 - 15s/epoch - 4ms/step
Epoch 17/20
3375/3375 - 15s - loss: 0.0476 - accuracy: 0.9929 - val_loss: 0.2777 -
val_accuracy: 0.9797 - 15s/epoch - 4ms/step
Epoch 18/20
3375/3375 - 15s - loss: 0.0444 - accuracy: 0.9940 - val_loss: 0.4782 -
val_accuracy: 0.9775 - 15s/epoch - 4ms/step
Epoch 19/20
```

3375/3375 - 15s - loss: 0.0528 - accuracy: 0.9938 - val_loss: 0.3533 - val_accuracy: 0.9812 - 15s/epoch - 4ms/step

Epoch 20/20

3375/3375 - 15s - loss: 0.0475 - accuracy: 0.9945 - val_loss: 0.3949 - val_accuracy: 0.9818 - 15s/epoch - 4ms/step

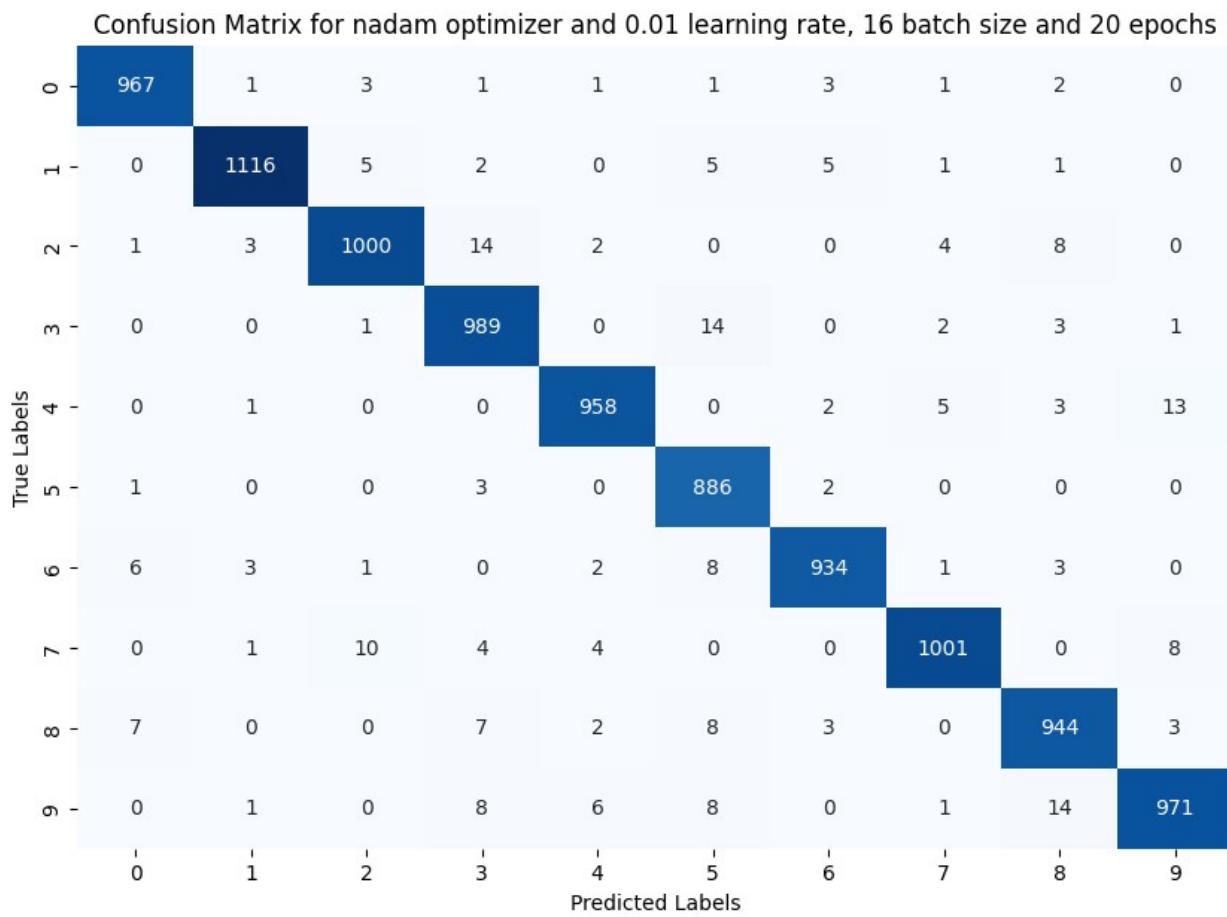
313/313 [=====] - 1s 2ms/step

Confusion Matrix nadam optimizer and the learning_rate is 0.01, 16 batch size and 20 epochs:

```
[[ 967   1   3   1   1   1   3   1   2   0]
 [ 0 1116   5   2   0   5   5   1   1   0]
 [ 1   3 1000  14   2   0   0   4   8   0]
 [ 0   0   1 989   0  14   0   2   3   1]
 [ 0   1   0   0 958   0   2   5   3 13]
 [ 1   0   0   3   0 886   2   0   0   0]
 [ 6   3   1   0   2   8 934   1   3   0]
 [ 0   1  10   4   4   0   0 1001   0   8]
 [ 7   0   0   7   2   8   3   0 944   3]
 [ 0   1   0   8   6   8   0   1 14 971]]
```

Precision: 0.9768

Recall: 0.9766



```
Training with nadam optimizer and the learning_rate is 0.001, 64 batch size and 5 epochs...
Epoch 1/5
844/844 - 6s - loss: 0.1793 - accuracy: 0.9449 - val_loss: 0.0702 -
val_accuracy: 0.9813 - 6s/epoch - 7ms/step
Epoch 2/5
844/844 - 5s - loss: 0.0577 - accuracy: 0.9825 - val_loss: 0.0506 -
val_accuracy: 0.9850 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 0.0353 - accuracy: 0.9896 - val_loss: 0.0493 -
val_accuracy: 0.9878 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 0.0232 - accuracy: 0.9929 - val_loss: 0.0548 -
val_accuracy: 0.9853 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.0166 - accuracy: 0.9946 - val_loss: 0.0440 -
val_accuracy: 0.9878 - 5s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 64 batch size and 5 epochs:
[[ 972    1    2    1    0    1    2    0    1    0]
 [  0 1130    2    0    1    1    1    0    0    0]
 [  1    1 1021    1    1    0    1    4    2    0]
 [  0    0    3 994    0    3    0    5    3    2]
 [  0    0    0    0 979    0    0    0    0    3]
 [  1    0    0    4    0 883    3    0    0    1]
 [  5    2    1    2    6    1 940    0    1    0]
 [  0    1    7    0    2    0    0 1015    1    2]
 [  5    1    2    1    5    1    4    1 950    4]
 [  1    2    0    5   19    2    0    4    1 975]]
```

Precision: 0.9860
Recall: 0.9859

Confusion Matrix for nadam optimizer and 0.001 learning rate, 64 batch size and 5 epochs										
	0	1	2	3	4	5	6	7	8	9
True Labels	972	1	2	1	0	1	2	0	1	0
0	972	1	2	1	0	1	2	0	1	0
1	0	1130	2	0	1	1	1	0	0	0
2	1	1	1021	1	1	0	1	4	2	0
3	0	0	3	994	0	3	0	5	3	2
4	0	0	0	0	979	0	0	0	0	3
5	1	0	0	4	0	883	3	0	0	1
6	5	2	1	2	6	1	940	0	1	0
7	0	1	7	0	2	0	0	1015	1	2
8	5	1	2	1	5	1	4	1	950	4
9	1	2	0	5	19	2	0	4	1	975
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```

Training with nadam optimizer and the learning_rate is 0.001, 64 batch
size and 15 epochs...
Epoch 1/15
844/844 - 6s - loss: 0.1591 - accuracy: 0.9524 - val_loss: 0.0766 -
val_accuracy: 0.9785 - 6s/epoch - 7ms/step
Epoch 2/15
844/844 - 5s - loss: 0.0500 - accuracy: 0.9843 - val_loss: 0.0507 -
val_accuracy: 0.9850 - 5s/epoch - 6ms/step
Epoch 3/15
844/844 - 5s - loss: 0.0298 - accuracy: 0.9911 - val_loss: 0.0550 -
val_accuracy: 0.9845 - 5s/epoch - 6ms/step
Epoch 4/15
844/844 - 5s - loss: 0.0199 - accuracy: 0.9937 - val_loss: 0.0522 -
val_accuracy: 0.9862 - 5s/epoch - 6ms/step
Epoch 5/15
844/844 - 5s - loss: 0.0128 - accuracy: 0.9961 - val_loss: 0.0595 -
val_accuracy: 0.9865 - 5s/epoch - 5ms/step
Epoch 6/15
844/844 - 5s - loss: 0.0099 - accuracy: 0.9970 - val_loss: 0.0473 -
val_accuracy: 0.9875 - 5s/epoch - 5ms/step
Epoch 7/15

```

```
844/844 - 5s - loss: 0.0064 - accuracy: 0.9983 - val_loss: 0.0484 -  
val_accuracy: 0.9883 - 5s/epoch - 5ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.0075 - accuracy: 0.9973 - val_loss: 0.0581 -  
val_accuracy: 0.9875 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.0054 - accuracy: 0.9984 - val_loss: 0.0625 -  
val_accuracy: 0.9875 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.0050 - accuracy: 0.9985 - val_loss: 0.0596 -  
val_accuracy: 0.9878 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.0020 - accuracy: 0.9996 - val_loss: 0.0625 -  
val_accuracy: 0.9865 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.0035 - accuracy: 0.9989 - val_loss: 0.0526 -  
val_accuracy: 0.9897 - 5s/epoch - 5ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.0037 - accuracy: 0.9988 - val_loss: 0.0631 -  
val_accuracy: 0.9885 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.0019 - accuracy: 0.9994 - val_loss: 0.0598 -  
val_accuracy: 0.9892 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.0037 - accuracy: 0.9989 - val_loss: 0.0657 -  
val_accuracy: 0.9887 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 64  
batch size and 15 epochs:  
[[ 976 1 1 1 0 0 1 0 0 0]  
[ 0 1130 2 1 0 0 1 1 0 0]  
[ 2 1 1020 1 0 0 0 6 1 1]  
[ 0 0 1 997 0 6 0 2 2 2]  
[ 0 0 1 0 973 0 1 0 0 7]  
[ 0 0 0 2 0 886 4 0 0 0]  
[ 2 3 1 1 1 2 948 0 0 0]  
[ 0 0 9 1 0 0 0 1014 1 3]  
[ 5 2 6 5 1 1 2 4 946 2]  
[ 0 4 1 4 9 4 0 7 2 978]]  
Precision: 0.9868  
Recall: 0.9868
```

Confusion Matrix for nadam optimizer and 0.001 learning rate, 64 batch size and 15 epochs										
	0	1	2	3	4	5	6	7	8	9
True Labels	976	1	1	1	0	0	1	0	0	0
0	976	1	1	1	0	0	1	0	0	0
1	0	1130	2	1	0	0	1	1	0	0
2	2	1	1020	1	0	0	0	6	1	1
3	0	0	1	997	0	6	0	2	2	2
4	0	0	1	0	973	0	1	0	0	7
5	0	0	0	2	0	886	4	0	0	0
6	2	3	1	1	1	2	948	0	0	0
7	0	0	9	1	0	0	0	1014	1	3
8	5	2	6	5	1	1	2	4	946	2
9	0	4	1	4	9	4	0	7	2	978
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```

Training with nadam optimizer and the learning_rate is 0.001, 64 batch
size and 20 epochs...
Epoch 1/20
844/844 - 7s - loss: 0.1721 - accuracy: 0.9474 - val_loss: 0.0723 -
val_accuracy: 0.9787 - 7s/epoch - 8ms/step
Epoch 2/20
844/844 - 5s - loss: 0.0548 - accuracy: 0.9836 - val_loss: 0.0534 -
val_accuracy: 0.9838 - 5s/epoch - 6ms/step
Epoch 3/20
844/844 - 5s - loss: 0.0331 - accuracy: 0.9899 - val_loss: 0.0508 -
val_accuracy: 0.9848 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: 0.0229 - accuracy: 0.9929 - val_loss: 0.0529 -
val_accuracy: 0.9858 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: 0.0163 - accuracy: 0.9950 - val_loss: 0.0486 -
val_accuracy: 0.9860 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 0.0119 - accuracy: 0.9963 - val_loss: 0.0543 -
val_accuracy: 0.9868 - 5s/epoch - 6ms/step
Epoch 7/20

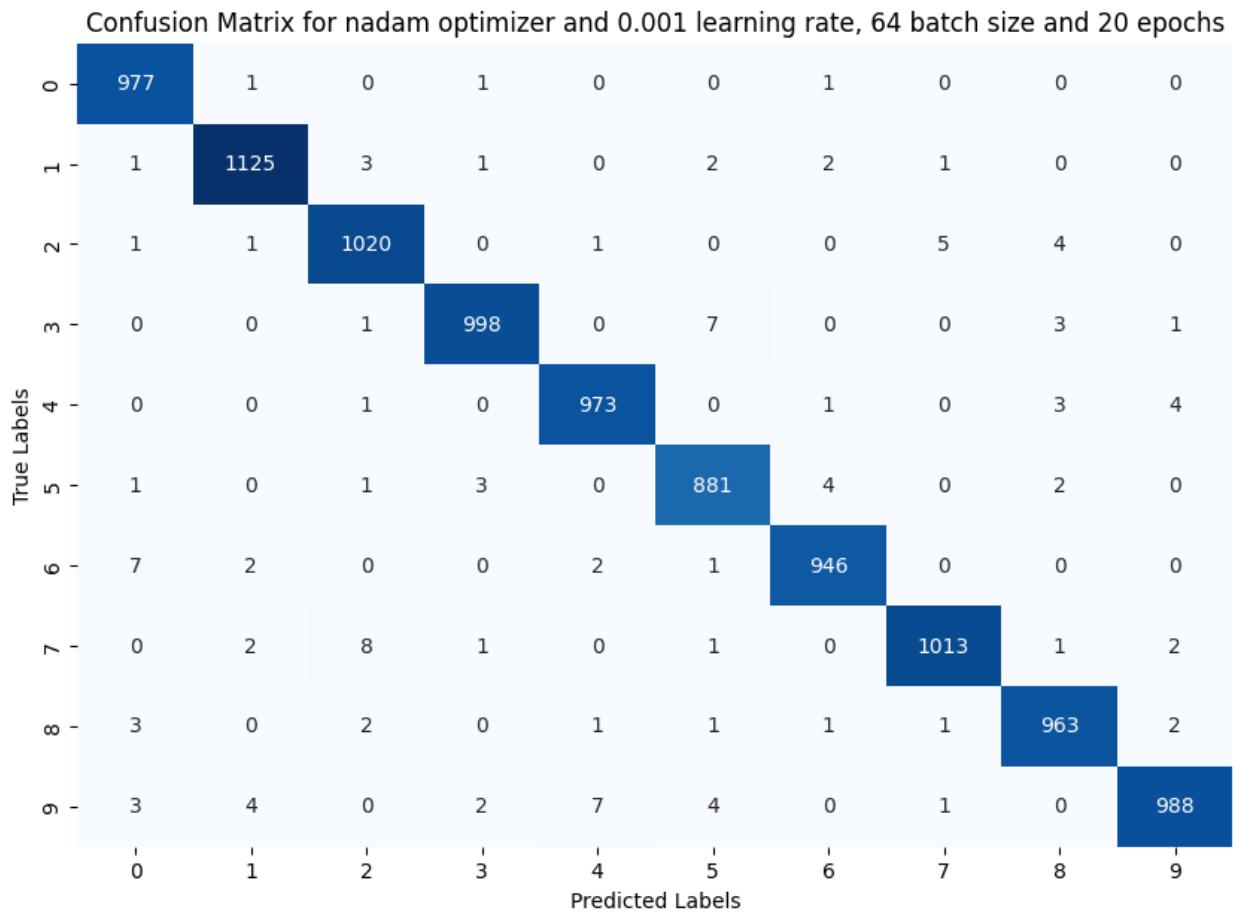
```

```
844/844 - 5s - loss: 0.0099 - accuracy: 0.9969 - val_loss: 0.0575 -  
val_accuracy: 0.9852 - 5s/epoch - 5ms/step  
Epoch 8/20  
844/844 - 5s - loss: 0.0078 - accuracy: 0.9975 - val_loss: 0.0573 -  
val_accuracy: 0.9860 - 5s/epoch - 5ms/step  
Epoch 9/20  
844/844 - 5s - loss: 0.0054 - accuracy: 0.9985 - val_loss: 0.0563 -  
val_accuracy: 0.9875 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: 0.0052 - accuracy: 0.9982 - val_loss: 0.0630 -  
val_accuracy: 0.9882 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: 0.0045 - accuracy: 0.9986 - val_loss: 0.0733 -  
val_accuracy: 0.9827 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 5s - loss: 0.0039 - accuracy: 0.9987 - val_loss: 0.0701 -  
val_accuracy: 0.9870 - 5s/epoch - 6ms/step  
Epoch 13/20  
844/844 - 5s - loss: 0.0038 - accuracy: 0.9987 - val_loss: 0.0666 -  
val_accuracy: 0.9882 - 5s/epoch - 6ms/step  
Epoch 14/20  
844/844 - 5s - loss: 0.0026 - accuracy: 0.9992 - val_loss: 0.0709 -  
val_accuracy: 0.9863 - 5s/epoch - 6ms/step  
Epoch 15/20  
844/844 - 5s - loss: 0.0039 - accuracy: 0.9988 - val_loss: 0.0700 -  
val_accuracy: 0.9858 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: 0.0027 - accuracy: 0.9992 - val_loss: 0.0807 -  
val_accuracy: 0.9872 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: 0.0027 - accuracy: 0.9991 - val_loss: 0.0699 -  
val_accuracy: 0.9877 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: 0.0036 - accuracy: 0.9987 - val_loss: 0.0753 -  
val_accuracy: 0.9877 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: 0.0019 - accuracy: 0.9994 - val_loss: 0.0719 -  
val_accuracy: 0.9878 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: 0.0016 - accuracy: 0.9995 - val_loss: 0.0705 -  
val_accuracy: 0.9878 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 64  
batch size and 20 epochs:  
[[ 977 1 0 1 0 0 1 0 0 0 ]]  
[ 1 1125 3 1 0 2 2 1 0 0 ]  
[ 1 1 1020 0 1 0 0 5 4 0 ]  
[ 0 0 1 998 0 7 0 0 3 1 ]  
[ 0 0 1 0 973 0 1 0 3 4 ]
```

```
[ 1 0 1 3 0 881 4 0 2 0]
[ 7 2 0 0 2 1 946 0 0 0]
[ 0 2 8 1 0 1 0 1013 1 2]
[ 3 0 2 0 1 1 1 1 963 2]
[ 3 4 0 2 7 4 0 1 0 988]]
```

Precision: 0.9884

Recall: 0.9884



Training with nadam optimizer and the learning_rate is 0.001, 128 batch size and 5 epochs...

Epoch 1/5

422/422 - 5s - loss: 0.1907 - accuracy: 0.9417 - val_loss: 0.0750 - val_accuracy: 0.9793 - 5s/epoch - 11ms/step

Epoch 2/5

422/422 - 3s - loss: 0.0613 - accuracy: 0.9813 - val_loss: 0.0541 - val_accuracy: 0.9858 - 3s/epoch - 8ms/step

Epoch 3/5

422/422 - 3s - loss: 0.0379 - accuracy: 0.9884 - val_loss: 0.0541 - val_accuracy: 0.9855 - 3s/epoch - 8ms/step

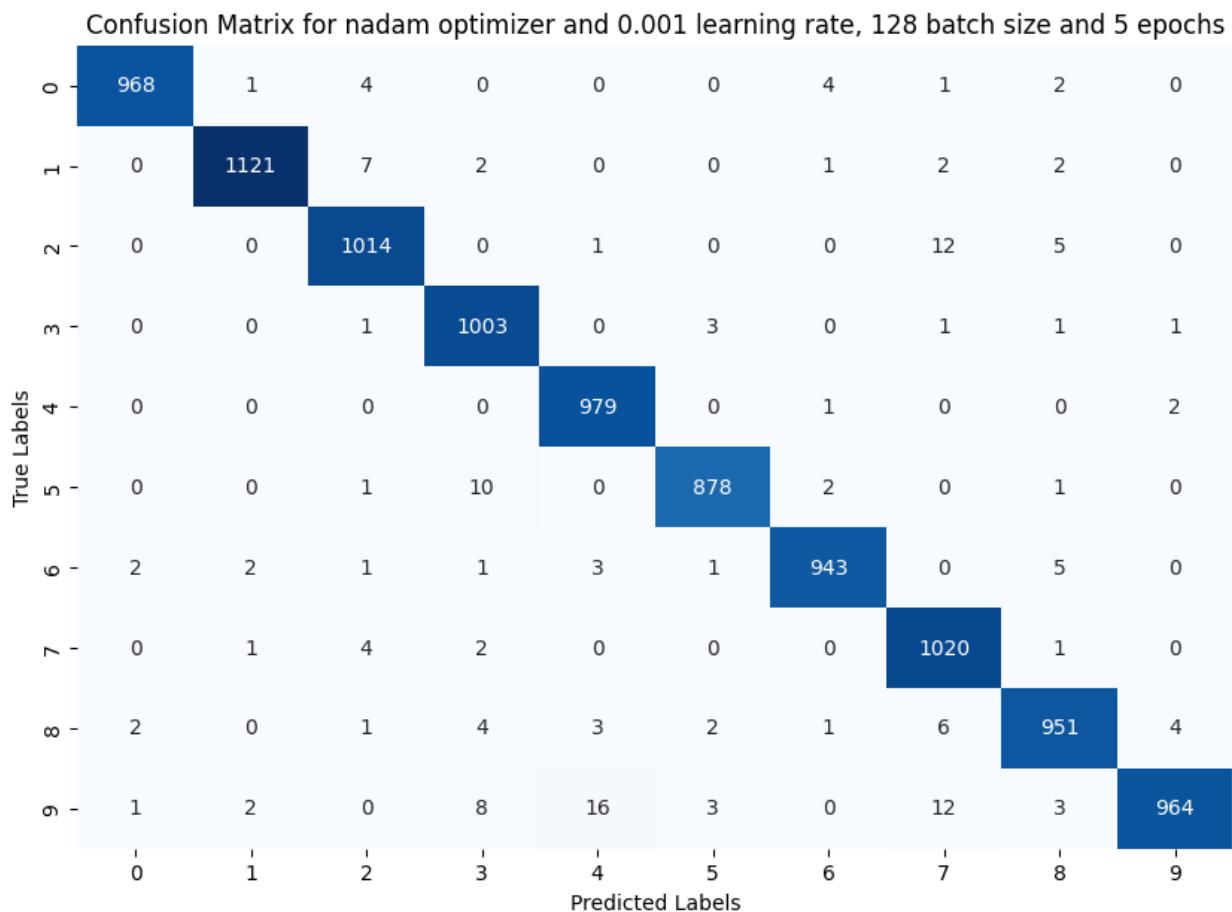
Epoch 4/5

422/422 - 3s - loss: 0.0248 - accuracy: 0.9924 - val_loss: 0.0581 -

```

val_accuracy: 0.9862 - 3s/epoch - 7ms/step
Epoch 5/5
422/422 - 3s - loss: 0.0173 - accuracy: 0.9953 - val_loss: 0.0574 -
val_accuracy: 0.9872 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 128
batch size and 5 epochs:
[[ 968   1   4   0   0   0   4   1   2   0]
 [ 0 1121   7   2   0   0   1   2   2   0]
 [ 0   0 1014   0   1   0   0 12   5   0]
 [ 0   0   1 1003   0   3   0   1   1   1]
 [ 0   0   0   0 979   0   1   0   0   2]
 [ 0   0   1   10   0 878   2   0   1   0]
 [ 2   2   1   1   3   1 943   0   5   0]
 [ 0   1   4   2   0   0   0 1020   1   0]
 [ 2   0   1   4   3   2   1   6 951   4]
 [ 1   2   0   8   16   3   0 12   3 964]]
Precision: 0.9842
Recall: 0.9841

```

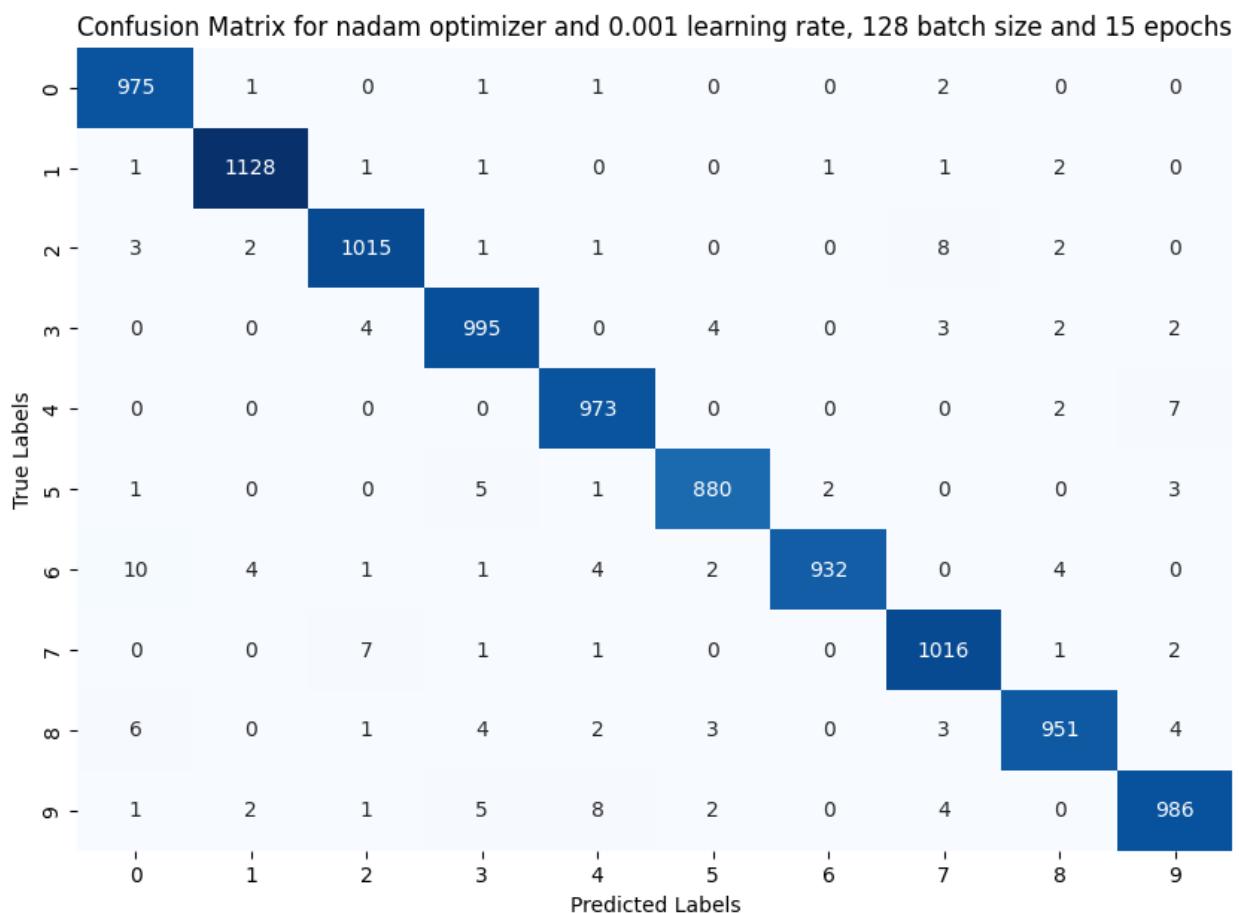


```
Training with nadam optimizer and the learning_rate is 0.001, 128
batch size and 15 epochs...
Epoch 1/15
422/422 - 5s - loss: 0.1778 - accuracy: 0.9467 - val_loss: 0.0658 -
val_accuracy: 0.9818 - 5s/epoch - 11ms/step
Epoch 2/15
422/422 - 3s - loss: 0.0566 - accuracy: 0.9829 - val_loss: 0.0596 -
val_accuracy: 0.9848 - 3s/epoch - 8ms/step
Epoch 3/15
422/422 - 3s - loss: 0.0351 - accuracy: 0.9893 - val_loss: 0.0577 -
val_accuracy: 0.9845 - 3s/epoch - 8ms/step
Epoch 4/15
422/422 - 3s - loss: 0.0243 - accuracy: 0.9925 - val_loss: 0.0494 -
val_accuracy: 0.9868 - 3s/epoch - 8ms/step
Epoch 5/15
422/422 - 3s - loss: 0.0159 - accuracy: 0.9953 - val_loss: 0.0522 -
val_accuracy: 0.9885 - 3s/epoch - 8ms/step
Epoch 6/15
422/422 - 3s - loss: 0.0114 - accuracy: 0.9965 - val_loss: 0.0548 -
val_accuracy: 0.9880 - 3s/epoch - 8ms/step
Epoch 7/15
422/422 - 3s - loss: 0.0076 - accuracy: 0.9979 - val_loss: 0.0502 -
val_accuracy: 0.9890 - 3s/epoch - 8ms/step
Epoch 8/15
422/422 - 3s - loss: 0.0053 - accuracy: 0.9989 - val_loss: 0.0504 -
val_accuracy: 0.9898 - 3s/epoch - 8ms/step
Epoch 9/15
422/422 - 3s - loss: 0.0055 - accuracy: 0.9984 - val_loss: 0.0608 -
val_accuracy: 0.9862 - 3s/epoch - 8ms/step
Epoch 10/15
422/422 - 3s - loss: 0.0098 - accuracy: 0.9967 - val_loss: 0.0771 -
val_accuracy: 0.9825 - 3s/epoch - 8ms/step
Epoch 11/15
422/422 - 3s - loss: 0.0053 - accuracy: 0.9983 - val_loss: 0.0572 -
val_accuracy: 0.9880 - 3s/epoch - 8ms/step
Epoch 12/15
422/422 - 3s - loss: 0.0043 - accuracy: 0.9986 - val_loss: 0.0555 -
val_accuracy: 0.9902 - 3s/epoch - 8ms/step
Epoch 13/15
422/422 - 3s - loss: 0.0037 - accuracy: 0.9990 - val_loss: 0.0745 -
val_accuracy: 0.9870 - 3s/epoch - 8ms/step
Epoch 14/15
422/422 - 3s - loss: 0.0044 - accuracy: 0.9986 - val_loss: 0.0696 -
val_accuracy: 0.9860 - 3s/epoch - 8ms/step
Epoch 15/15
422/422 - 3s - loss: 0.0033 - accuracy: 0.9989 - val_loss: 0.0651 -
val_accuracy: 0.9880 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 128
batch size and 15 epochs:
```

```
[[ 975  1  0  1  1  0  0  2  0  0]
 [ 1 1128  1  1  0  0  1  1  2  0]
 [ 3  2 1015  1  1  0  0  8  2  0]
 [ 0  0  4 995  0  4  0  3  2  2]
 [ 0  0  0  0 973  0  0  0  2  7]
 [ 1  0  0  5  1 880  2  0  0  3]
 [ 10  4  1  1  4  2 932  0  4  0]
 [ 0  0  7  1  1  0  0 1016  1  2]
 [ 6  0  1  4  2  3  0  3 951  4]
 [ 1  2  1  5  8  2  0  4  0 986]]
```

Precision: 0.9851

Recall: 0.9851



Training with nadam optimizer and the learning_rate is 0.001, 128 batch size and 20 epochs...

Epoch 1/20

422/422 - 5s - loss: 0.1816 - accuracy: 0.9446 - val_loss: 0.0750 - val_accuracy: 0.9783 - 5s/epoch - 11ms/step

Epoch 2/20

422/422 - 3s - loss: 0.0557 - accuracy: 0.9827 - val_loss: 0.0524 - val_accuracy: 0.9835 - 3s/epoch - 8ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.0353 - accuracy: 0.9895 - val_loss: 0.0497 -
val_accuracy: 0.9862 - 3s/epoch - 8ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0229 - accuracy: 0.9929 - val_loss: 0.0487 -
val_accuracy: 0.9870 - 3s/epoch - 8ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0152 - accuracy: 0.9955 - val_loss: 0.0468 -
val_accuracy: 0.9873 - 3s/epoch - 8ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0122 - accuracy: 0.9963 - val_loss: 0.0487 -
val_accuracy: 0.9902 - 3s/epoch - 8ms/step
Epoch 7/20
422/422 - 3s - loss: 0.0082 - accuracy: 0.9976 - val_loss: 0.0503 -
val_accuracy: 0.9890 - 3s/epoch - 8ms/step
Epoch 8/20
422/422 - 3s - loss: 0.0059 - accuracy: 0.9984 - val_loss: 0.0548 -
val_accuracy: 0.9883 - 3s/epoch - 8ms/step
Epoch 9/20
422/422 - 3s - loss: 0.0058 - accuracy: 0.9985 - val_loss: 0.0507 -
val_accuracy: 0.9892 - 3s/epoch - 7ms/step
Epoch 10/20
422/422 - 3s - loss: 0.0060 - accuracy: 0.9981 - val_loss: 0.0570 -
val_accuracy: 0.9862 - 3s/epoch - 7ms/step
Epoch 11/20
422/422 - 3s - loss: 0.0062 - accuracy: 0.9980 - val_loss: 0.0548 -
val_accuracy: 0.9888 - 3s/epoch - 8ms/step
Epoch 12/20
422/422 - 3s - loss: 0.0046 - accuracy: 0.9985 - val_loss: 0.0556 -
val_accuracy: 0.9892 - 3s/epoch - 8ms/step
Epoch 13/20
422/422 - 3s - loss: 0.0027 - accuracy: 0.9993 - val_loss: 0.0612 -
val_accuracy: 0.9887 - 3s/epoch - 8ms/step
Epoch 14/20
422/422 - 3s - loss: 0.0034 - accuracy: 0.9989 - val_loss: 0.0655 -
val_accuracy: 0.9868 - 3s/epoch - 8ms/step
Epoch 15/20
422/422 - 3s - loss: 0.0049 - accuracy: 0.9985 - val_loss: 0.0562 -
val_accuracy: 0.9898 - 3s/epoch - 8ms/step
Epoch 16/20
422/422 - 3s - loss: 0.0044 - accuracy: 0.9986 - val_loss: 0.0560 -
val_accuracy: 0.9875 - 3s/epoch - 8ms/step
Epoch 17/20
422/422 - 3s - loss: 0.0015 - accuracy: 0.9996 - val_loss: 0.0604 -
val_accuracy: 0.9877 - 3s/epoch - 8ms/step
Epoch 18/20
422/422 - 3s - loss: 4.8328e-04 - accuracy: 0.9999 - val_loss: 0.0609 -
val_accuracy: 0.9885 - 3s/epoch - 8ms/step
Epoch 19/20
```

```
422/422 - 3s - loss: 2.1071e-04 - accuracy: 1.0000 - val_loss: 0.0569  
- val_accuracy: 0.9892 - 3s/epoch - 8ms/step
```

```
Epoch 20/20
```

```
422/422 - 3s - loss: 6.1584e-05 - accuracy: 1.0000 - val_loss: 0.0586  
- val_accuracy: 0.9887 - 3s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

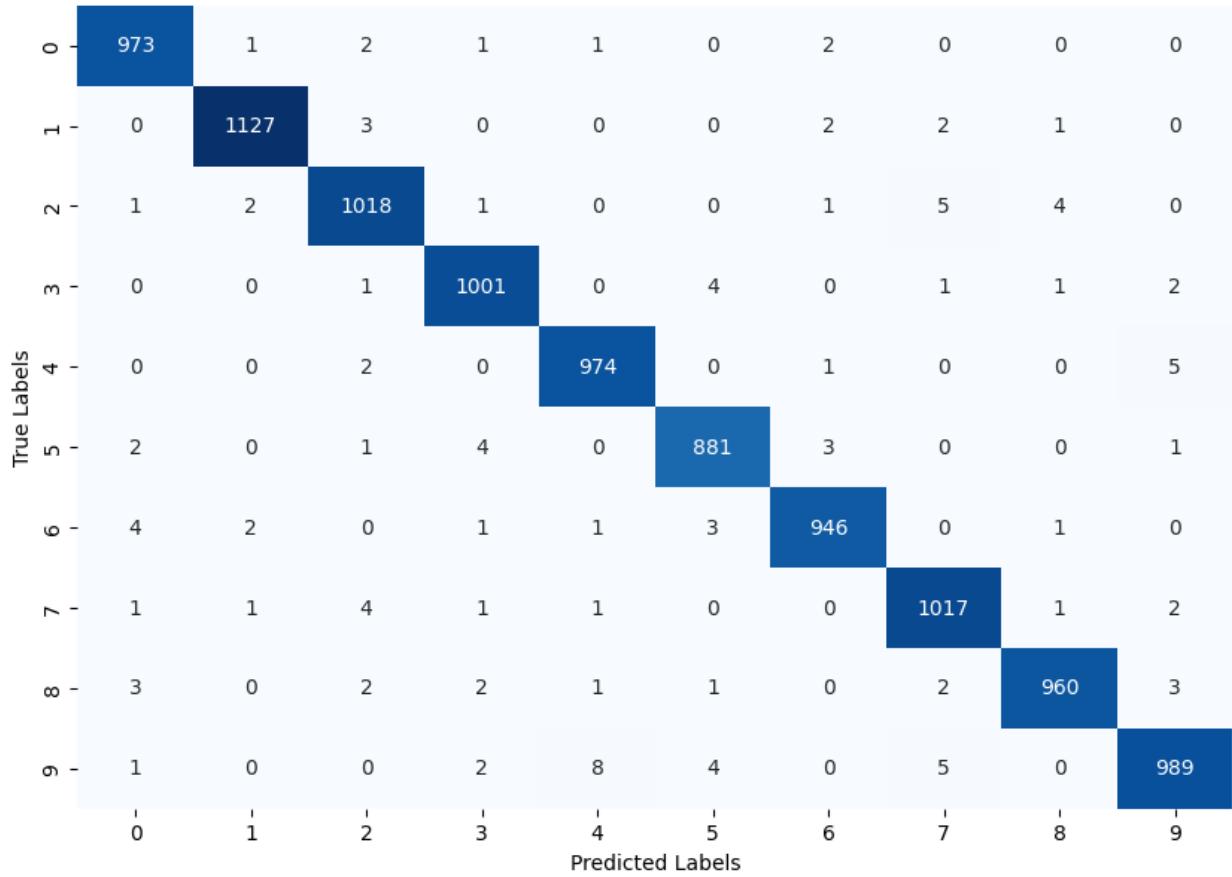
```
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 128  
batch size and 20 epochs:
```

```
[[ 973   1   2   1   1   0   2   0   0   0 ]  
[  0 1127   3   0   0   0   2   2   1   0 ]  
[  1   2 1018   1   0   0   1   5   4   0 ]  
[  0   0   1 1001   0   4   0   1   1   2 ]  
[  0   0   2   0  974   0   1   0   0   5 ]  
[  2   0   1   4   0  881   3   0   0   1 ]  
[  4   2   0   1   1   3  946   0   1   0 ]  
[  1   1   4   1   1   0   0 1017   1   2 ]  
[  3   0   2   2   1   1   0   2  960   3 ]  
[  1   0   0   2   8   4   0   5   0  989 ]]
```

```
Precision: 0.9886
```

```
Recall: 0.9886
```

Confusion Matrix for nadam optimizer and 0.001 learning rate, 128 batch size and 20 epochs



```
Training with nadam optimizer and the learning_rate is 0.001, 256
batch size and 5 epochs...
Epoch 1/5
211/211 - 4s - loss: 0.2311 - accuracy: 0.9308 - val_loss: 0.0765 -
val_accuracy: 0.9793 - 4s/epoch - 18ms/step
Epoch 2/5
211/211 - 3s - loss: 0.0633 - accuracy: 0.9822 - val_loss: 0.0538 -
val_accuracy: 0.9862 - 3s/epoch - 12ms/step
Epoch 3/5
211/211 - 3s - loss: 0.0397 - accuracy: 0.9884 - val_loss: 0.0447 -
val_accuracy: 0.9882 - 3s/epoch - 12ms/step
Epoch 4/5
211/211 - 3s - loss: 0.0271 - accuracy: 0.9922 - val_loss: 0.0451 -
val_accuracy: 0.9880 - 3s/epoch - 13ms/step
Epoch 5/5
211/211 - 3s - loss: 0.0191 - accuracy: 0.9950 - val_loss: 0.0481 -
val_accuracy: 0.9875 - 3s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 256
batch size and 5 epochs:
[[ 978    0    0    1    0    1    0    0    0    0]
 [  2 1126    2    2    0    0    1    2    0    0]
 [  7    2 1015    1    1    0    1    3    2    0]
 [  1    0    2 998    0    3    0    2    4    0]
 [  2    0    0    0 974    0    0    0    1    5]
 [  4    0    0    5    0 879    3    0    1    0]
 [  9    3    0    1    1    4 938    0    2    0]
 [  1    2   11    3    1    0    0 992    4   14]
 [  8    0    7    0    0    0    1    2 952    4]
 [  7    2    0    7    9    3    0    2    2 977]]
```

Precision: 0.9830
Recall: 0.9829

Confusion Matrix for nadam optimizer and 0.001 learning rate, 256 batch size and 5 epochs										
	0	1	2	3	4	5	6	7	8	9
0	978	0	0	1	0	1	0	0	0	0
1	2	1126	2	2	0	0	1	2	0	0
2	7	2	1015	1	1	0	1	3	2	0
3	1	0	2	998	0	3	0	2	4	0
4	2	0	0	0	974	0	0	0	1	5
5	4	0	0	5	0	879	3	0	1	0
6	9	3	0	1	1	4	938	0	2	0
7	1	2	11	3	1	0	0	992	4	14
8	8	0	7	0	0	0	1	2	952	4
9	7	2	0	7	9	3	0	2	2	977
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

Training with nadam optimizer and the learning_rate is 0.001, 256 batch size and 15 epochs...

Epoch 1/15

211/211 - 4s - loss: 0.2493 - accuracy: 0.9262 - val_loss: 0.0857 - val_accuracy: 0.9770 - 4s/epoch - 18ms/step

Epoch 2/15

211/211 - 3s - loss: 0.0721 - accuracy: 0.9782 - val_loss: 0.0612 - val_accuracy: 0.9820 - 3s/epoch - 12ms/step

Epoch 3/15

211/211 - 3s - loss: 0.0445 - accuracy: 0.9873 - val_loss: 0.0517 - val_accuracy: 0.9843 - 3s/epoch - 12ms/step

Epoch 4/15

211/211 - 3s - loss: 0.0314 - accuracy: 0.9904 - val_loss: 0.0502 - val_accuracy: 0.9870 - 3s/epoch - 12ms/step

Epoch 5/15

211/211 - 3s - loss: 0.0220 - accuracy: 0.9936 - val_loss: 0.0528 - val_accuracy: 0.9860 - 3s/epoch - 12ms/step

Epoch 6/15

211/211 - 2s - loss: 0.0163 - accuracy: 0.9954 - val_loss: 0.0475 - val_accuracy: 0.9872 - 2s/epoch - 12ms/step

Epoch 7/15

```
211/211 - 3s - loss: 0.0127 - accuracy: 0.9964 - val_loss: 0.0462 -  
val_accuracy: 0.9877 - 3s/epoch - 12ms/step  
Epoch 8/15  
211/211 - 2s - loss: 0.0091 - accuracy: 0.9978 - val_loss: 0.0461 -  
val_accuracy: 0.9888 - 2s/epoch - 12ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.0070 - accuracy: 0.9984 - val_loss: 0.0505 -  
val_accuracy: 0.9887 - 3s/epoch - 12ms/step  
Epoch 10/15  
211/211 - 2s - loss: 0.0047 - accuracy: 0.9991 - val_loss: 0.0514 -  
val_accuracy: 0.9885 - 2s/epoch - 12ms/step  
Epoch 11/15  
211/211 - 2s - loss: 0.0033 - accuracy: 0.9995 - val_loss: 0.0529 -  
val_accuracy: 0.9883 - 2s/epoch - 12ms/step  
Epoch 12/15  
211/211 - 2s - loss: 0.0042 - accuracy: 0.9989 - val_loss: 0.0541 -  
val_accuracy: 0.9885 - 2s/epoch - 12ms/step  
Epoch 13/15  
211/211 - 3s - loss: 0.0057 - accuracy: 0.9984 - val_loss: 0.0556 -  
val_accuracy: 0.9873 - 3s/epoch - 12ms/step  
Epoch 14/15  
211/211 - 3s - loss: 0.0039 - accuracy: 0.9991 - val_loss: 0.0626 -  
val_accuracy: 0.9862 - 3s/epoch - 12ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.0027 - accuracy: 0.9994 - val_loss: 0.0580 -  
val_accuracy: 0.9887 - 3s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 256  
batch size and 15 epochs:  
[[ 974  0  1  0  1  1  2  1  0  0]  
[ 0 1125  4  0  0  1  2  1  2  0]  
[ 1  1 1021  0  1  0  1  6  1  0]  
[ 0  0  4 990  0  7  0  4  1  4]  
[ 0  0  4  0 966  0  1  2  3  6]  
[ 1  0  0  5  0 881  3  0  0  2]  
[ 7  2  0  1  1  3 943  0  1  0]  
[ 0  1  6  2  0  0  0 1015  1  3]  
[ 6  0  5  2  1  2  0  2 952  4]  
[ 1  2  0  1  4  4  0  4  0 993]]  
Precision: 0.9860  
Recall: 0.9860
```

Confusion Matrix for nadam optimizer and 0.001 learning rate, 256 batch size and 15 epochs

	0	1	2	3	4	5	6	7	8	9
True Labels	974	0	1	0	1	1	2	1	0	0
0	974	0	1	0	1	1	2	1	0	0
1	0	1125	4	0	0	1	2	1	2	0
2	1	1	1021	0	1	0	1	6	1	0
3	0	0	4	990	0	7	0	4	1	4
4	0	0	4	0	966	0	1	2	3	6
5	1	0	0	5	0	881	3	0	0	2
6	7	2	0	1	1	3	943	0	1	0
7	0	1	6	2	0	0	0	1015	1	3
8	6	0	5	2	1	2	0	2	952	4
9	1	2	0	1	4	4	0	4	0	993
	0	1	2	3	4	5	6	7	8	9
Predicted Labels										

```
Training with nadam optimizer and the learning_rate is 0.001, 256
batch size and 20 epochs...
```

```
Epoch 1/20
```

```
211/211 - 4s - loss: 0.2275 - accuracy: 0.9319 - val_loss: 0.0867 -
val_accuracy: 0.9747 - 4s/epoch - 20ms/step
```

```
Epoch 2/20
```

```
211/211 - 3s - loss: 0.0689 - accuracy: 0.9794 - val_loss: 0.0588 -
val_accuracy: 0.9822 - 3s/epoch - 13ms/step
```

```
Epoch 3/20
```

```
211/211 - 3s - loss: 0.0434 - accuracy: 0.9877 - val_loss: 0.0532 -
val_accuracy: 0.9852 - 3s/epoch - 12ms/step
```

```
Epoch 4/20
```

```
211/211 - 3s - loss: 0.0307 - accuracy: 0.9907 - val_loss: 0.0473 -
val_accuracy: 0.9890 - 3s/epoch - 12ms/step
```

```
Epoch 5/20
```

```
211/211 - 3s - loss: 0.0218 - accuracy: 0.9938 - val_loss: 0.0525 -
val_accuracy: 0.9863 - 3s/epoch - 13ms/step
```

```
Epoch 6/20
```

```
211/211 - 3s - loss: 0.0155 - accuracy: 0.9959 - val_loss: 0.0553 -
val_accuracy: 0.9860 - 3s/epoch - 12ms/step
```

```
Epoch 7/20
```

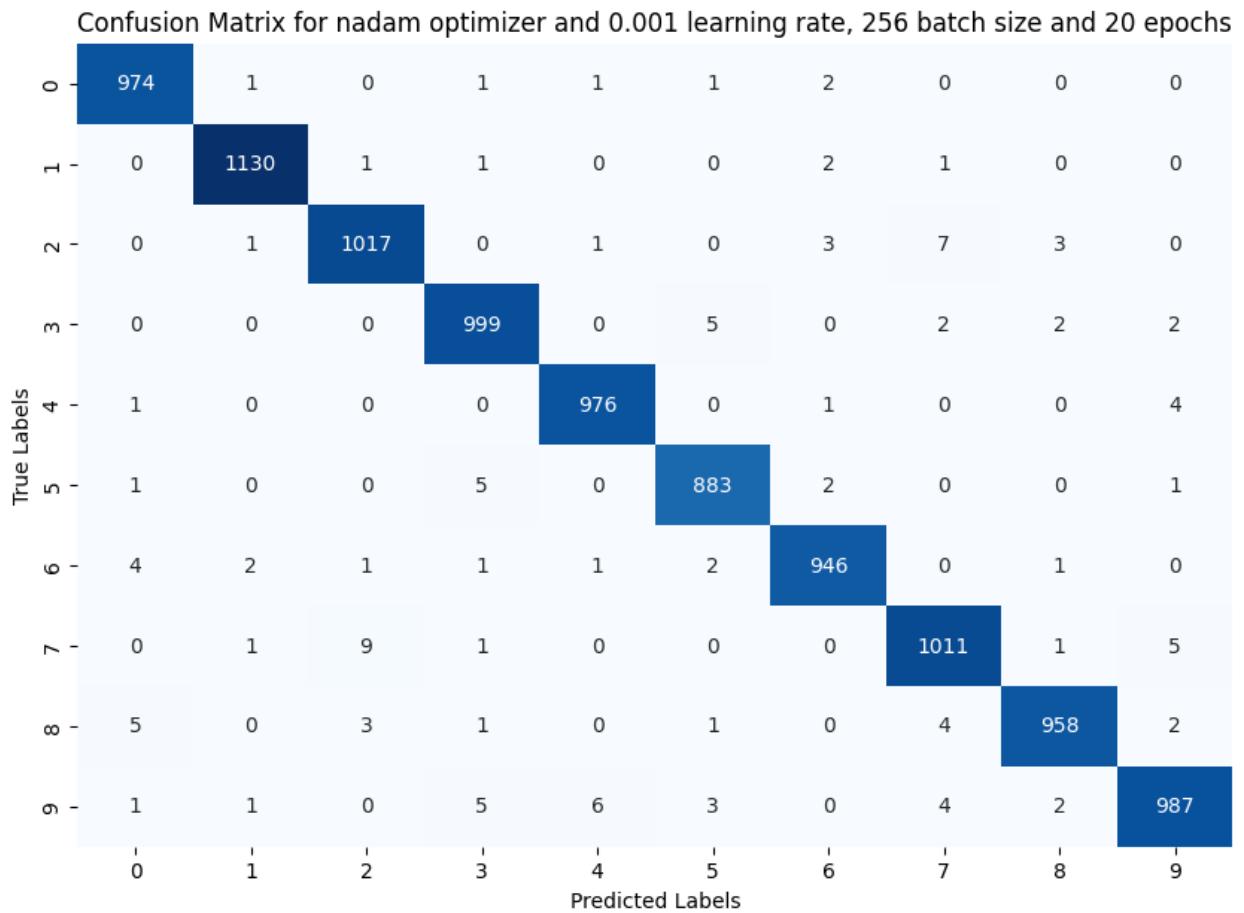
```

211/211 - 3s - loss: 0.0111 - accuracy: 0.9973 - val_loss: 0.0525 -
val_accuracy: 0.9878 - 3s/epoch - 13ms/step
Epoch 8/20
211/211 - 3s - loss: 0.0088 - accuracy: 0.9981 - val_loss: 0.0484 -
val_accuracy: 0.9877 - 3s/epoch - 12ms/step
Epoch 9/20
211/211 - 3s - loss: 0.0072 - accuracy: 0.9982 - val_loss: 0.0573 -
val_accuracy: 0.9867 - 3s/epoch - 12ms/step
Epoch 10/20
211/211 - 3s - loss: 0.0042 - accuracy: 0.9994 - val_loss: 0.0514 -
val_accuracy: 0.9885 - 3s/epoch - 12ms/step
Epoch 11/20
211/211 - 3s - loss: 0.0036 - accuracy: 0.9994 - val_loss: 0.0464 -
val_accuracy: 0.9890 - 3s/epoch - 12ms/step
Epoch 12/20
211/211 - 3s - loss: 0.0027 - accuracy: 0.9996 - val_loss: 0.0586 -
val_accuracy: 0.9872 - 3s/epoch - 12ms/step
Epoch 13/20
211/211 - 3s - loss: 0.0037 - accuracy: 0.9990 - val_loss: 0.0681 -
val_accuracy: 0.9840 - 3s/epoch - 12ms/step
Epoch 14/20
211/211 - 3s - loss: 0.0085 - accuracy: 0.9972 - val_loss: 0.0755 -
val_accuracy: 0.9843 - 3s/epoch - 12ms/step
Epoch 15/20
211/211 - 3s - loss: 0.0051 - accuracy: 0.9984 - val_loss: 0.0551 -
val_accuracy: 0.9872 - 3s/epoch - 12ms/step
Epoch 16/20
211/211 - 3s - loss: 0.0040 - accuracy: 0.9988 - val_loss: 0.0674 -
val_accuracy: 0.9873 - 3s/epoch - 12ms/step
Epoch 17/20
211/211 - 3s - loss: 0.0040 - accuracy: 0.9988 - val_loss: 0.0607 -
val_accuracy: 0.9890 - 3s/epoch - 12ms/step
Epoch 18/20
211/211 - 3s - loss: 0.0019 - accuracy: 0.9996 - val_loss: 0.0605 -
val_accuracy: 0.9883 - 3s/epoch - 12ms/step
Epoch 19/20
211/211 - 3s - loss: 5.5998e-04 - accuracy: 1.0000 - val_loss: 0.0617
- val_accuracy: 0.9890 - 3s/epoch - 12ms/step
Epoch 20/20
211/211 - 3s - loss: 2.3703e-04 - accuracy: 1.0000 - val_loss: 0.0596
- val_accuracy: 0.9893 - 3s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 256
batch size and 20 epochs:
[[ 974   1   0   1   1   1   2   0   0   0]
 [  0 1130   1   1   0   0   2   1   0   0]
 [  0   1 1017   0   1   0   3   7   3   0]
 [  0   0   0  999   0   5   0   2   2   2]
 [  1   0   0   0  976   0   1   0   0   4]]
```

```
[ 1 0 0 5 0 883 2 0 0 1]
[ 4 2 1 1 1 2 946 0 1 0]
[ 0 1 9 1 0 0 0 1011 1 5]
[ 5 0 3 1 0 1 0 4 958 2]
[ 1 1 0 5 6 3 0 4 2 987]]
```

Precision: 0.9881

Recall: 0.9881



Training with nadam optimizer and the learning_rate is 0.001, 16 batch size and 5 epochs...

Epoch 1/5

3375/3375 - 16s - loss: 0.1251 - accuracy: 0.9618 - val_loss: 0.0589 - val_accuracy: 0.9838 - 16s/epoch - 5ms/step

Epoch 2/5

3375/3375 - 15s - loss: 0.0452 - accuracy: 0.9863 - val_loss: 0.0479 - val_accuracy: 0.9863 - 15s/epoch - 4ms/step

Epoch 3/5

3375/3375 - 15s - loss: 0.0262 - accuracy: 0.9915 - val_loss: 0.0583 - val_accuracy: 0.9848 - 15s/epoch - 4ms/step

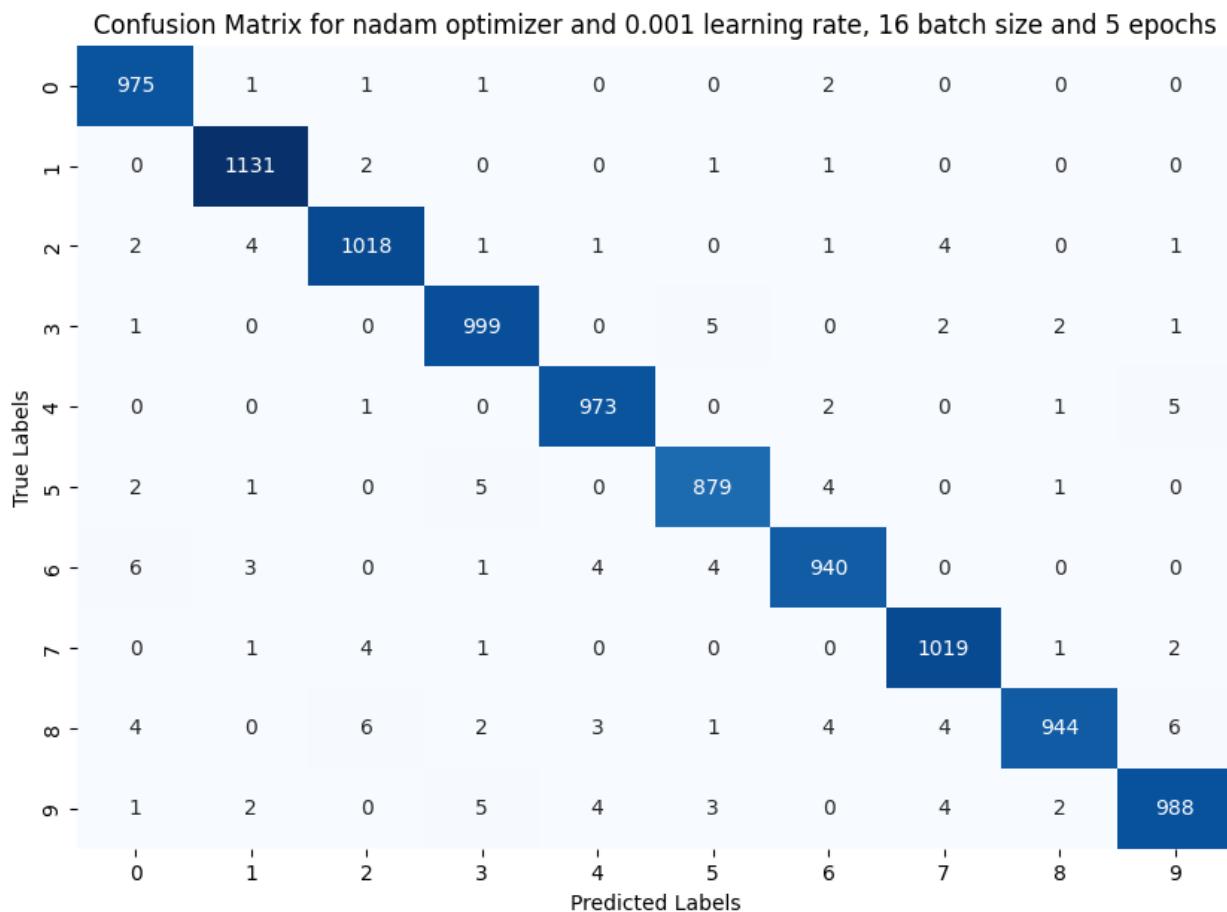
Epoch 4/5

3375/3375 - 15s - loss: 0.0155 - accuracy: 0.9947 - val_loss: 0.0567 -

```

val_accuracy: 0.9868 - 15s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 15s - loss: 0.0110 - accuracy: 0.9961 - val_loss: 0.0543 -
val_accuracy: 0.9877 - 15s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 16
batch size and 5 epochs:
[[ 975    1    1    1    0    0    2    0    0    0]
 [ 0 1131    2    0    0    1    1    0    0    0]
 [ 2    4 1018    1    1    0    1    4    0    1]
 [ 1    0    0 999    0    5    0    2    2    1]
 [ 0    0    1    0 973    0    2    0    1    5]
 [ 2    1    0    5    0 879    4    0    1    0]
 [ 6    3    0    1    4    4 940    0    0    0]
 [ 0    1    4    1    0    0    0 1019    1    2]
 [ 4    0    6    2    3    1    4    4    944    6]
 [ 1    2    0    5    4    3    0    4    2    988]]
Precision: 0.9866
Recall: 0.9866

```

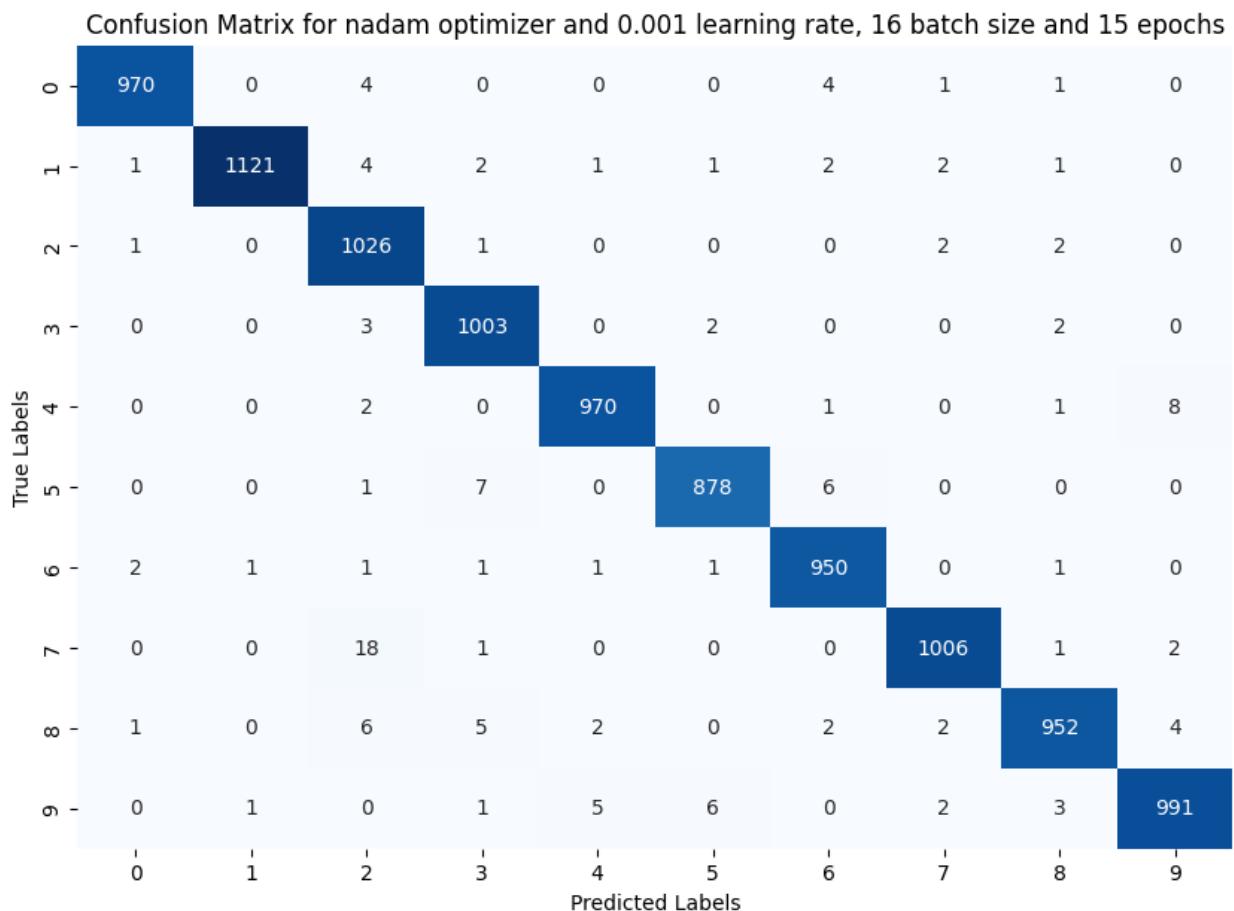


```
Training with nadam optimizer and the learning_rate is 0.001, 16 batch size and 15 epochs...
Epoch 1/15
3375/3375 - 16s - loss: 0.1399 - accuracy: 0.9561 - val_loss: 0.0642 -
val_accuracy: 0.9817 - 16s/epoch - 5ms/step
Epoch 2/15
3375/3375 - 15s - loss: 0.0474 - accuracy: 0.9848 - val_loss: 0.0573 -
val_accuracy: 0.9848 - 15s/epoch - 4ms/step
Epoch 3/15
3375/3375 - 15s - loss: 0.0286 - accuracy: 0.9905 - val_loss: 0.0577 -
val_accuracy: 0.9847 - 15s/epoch - 5ms/step
Epoch 4/15
3375/3375 - 15s - loss: 0.0165 - accuracy: 0.9943 - val_loss: 0.0541 -
val_accuracy: 0.9855 - 15s/epoch - 5ms/step
Epoch 5/15
3375/3375 - 16s - loss: 0.0115 - accuracy: 0.9961 - val_loss: 0.0674 -
val_accuracy: 0.9850 - 16s/epoch - 5ms/step
Epoch 6/15
3375/3375 - 15s - loss: 0.0087 - accuracy: 0.9970 - val_loss: 0.0609 -
val_accuracy: 0.9880 - 15s/epoch - 5ms/step
Epoch 7/15
3375/3375 - 15s - loss: 0.0065 - accuracy: 0.9977 - val_loss: 0.0748 -
val_accuracy: 0.9868 - 15s/epoch - 4ms/step
Epoch 8/15
3375/3375 - 16s - loss: 0.0054 - accuracy: 0.9981 - val_loss: 0.0682 -
val_accuracy: 0.9870 - 16s/epoch - 5ms/step
Epoch 9/15
3375/3375 - 16s - loss: 0.0045 - accuracy: 0.9984 - val_loss: 0.0797 -
val_accuracy: 0.9880 - 16s/epoch - 5ms/step
Epoch 10/15
3375/3375 - 15s - loss: 0.0034 - accuracy: 0.9989 - val_loss: 0.0929 -
val_accuracy: 0.9847 - 15s/epoch - 5ms/step
Epoch 11/15
3375/3375 - 16s - loss: 0.0037 - accuracy: 0.9988 - val_loss: 0.0893 -
val_accuracy: 0.9865 - 16s/epoch - 5ms/step
Epoch 12/15
3375/3375 - 15s - loss: 0.0035 - accuracy: 0.9990 - val_loss: 0.0779 -
val_accuracy: 0.9883 - 15s/epoch - 4ms/step
Epoch 13/15
3375/3375 - 16s - loss: 0.0030 - accuracy: 0.9990 - val_loss: 0.0798 -
val_accuracy: 0.9873 - 16s/epoch - 5ms/step
Epoch 14/15
3375/3375 - 16s - loss: 0.0023 - accuracy: 0.9993 - val_loss: 0.0899 -
val_accuracy: 0.9890 - 16s/epoch - 5ms/step
Epoch 15/15
3375/3375 - 16s - loss: 0.0019 - accuracy: 0.9994 - val_loss: 0.0937 -
val_accuracy: 0.9870 - 16s/epoch - 5ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 16 batch size and 15 epochs:
```

```
[[ 970  0   4   0   0   0   4   1   1   0]
 [ 1 1121  4   2   1   1   2   2   1   0]
 [ 1  0 1026  1   0   0   0   2   2   0]
 [ 0  0  3 1003  0   2   0   0   2   0]
 [ 0  0   2   0 970  0   1   0   1   8]
 [ 0  0   1   7   0 878  6   0   0   0]
 [ 2  1   1   1   1 950  0   1   0   0]
 [ 0  0  18   1   0   0   0 1006  1   2]
 [ 1  0   6   5   2   0   2   2 952  4]
 [ 0  1   0   1   5   6   0   2   3 991]]
```

Precision: 0.9868

Recall: 0.9867



Training with nadam optimizer and the learning_rate is 0.001, 16 batch size and 20 epochs...

Epoch 1/20

3375/3375 - 18s - loss: 0.1467 - accuracy: 0.9559 - val_loss: 0.0639 - val_accuracy: 0.9815 - 18s/epoch - 5ms/step

Epoch 2/20

3375/3375 - 16s - loss: 0.0501 - accuracy: 0.9838 - val_loss: 0.0628 - val_accuracy: 0.9817 - 16s/epoch - 5ms/step

```
Epoch 3/20
3375/3375 - 17s - loss: 0.0297 - accuracy: 0.9904 - val_loss: 0.0443 -
val_accuracy: 0.9878 - 17s/epoch - 5ms/step
Epoch 4/20
3375/3375 - 17s - loss: 0.0165 - accuracy: 0.9948 - val_loss: 0.0498 -
val_accuracy: 0.9867 - 17s/epoch - 5ms/step
Epoch 5/20
3375/3375 - 17s - loss: 0.0128 - accuracy: 0.9958 - val_loss: 0.0537 -
val_accuracy: 0.9885 - 17s/epoch - 5ms/step
Epoch 6/20
3375/3375 - 16s - loss: 0.0085 - accuracy: 0.9971 - val_loss: 0.0587 -
val_accuracy: 0.9885 - 16s/epoch - 5ms/step
Epoch 7/20
3375/3375 - 16s - loss: 0.0065 - accuracy: 0.9979 - val_loss: 0.0648 -
val_accuracy: 0.9858 - 16s/epoch - 5ms/step
Epoch 8/20
3375/3375 - 17s - loss: 0.0045 - accuracy: 0.9986 - val_loss: 0.0775 -
val_accuracy: 0.9850 - 17s/epoch - 5ms/step
Epoch 9/20
3375/3375 - 16s - loss: 0.0049 - accuracy: 0.9984 - val_loss: 0.0781 -
val_accuracy: 0.9873 - 16s/epoch - 5ms/step
Epoch 10/20
3375/3375 - 17s - loss: 0.0042 - accuracy: 0.9986 - val_loss: 0.0664 -
val_accuracy: 0.9873 - 17s/epoch - 5ms/step
Epoch 11/20
3375/3375 - 16s - loss: 0.0028 - accuracy: 0.9990 - val_loss: 0.0842 -
val_accuracy: 0.9868 - 16s/epoch - 5ms/step
Epoch 12/20
3375/3375 - 16s - loss: 0.0029 - accuracy: 0.9990 - val_loss: 0.0672 -
val_accuracy: 0.9892 - 16s/epoch - 5ms/step
Epoch 13/20
3375/3375 - 16s - loss: 0.0023 - accuracy: 0.9992 - val_loss: 0.1193 -
val_accuracy: 0.9822 - 16s/epoch - 5ms/step
Epoch 14/20
3375/3375 - 16s - loss: 0.0021 - accuracy: 0.9994 - val_loss: 0.0799 -
val_accuracy: 0.9892 - 16s/epoch - 5ms/step
Epoch 15/20
3375/3375 - 16s - loss: 0.0034 - accuracy: 0.9990 - val_loss: 0.0898 -
val_accuracy: 0.9868 - 16s/epoch - 5ms/step
Epoch 16/20
3375/3375 - 16s - loss: 0.0025 - accuracy: 0.9994 - val_loss: 0.0916 -
val_accuracy: 0.9857 - 16s/epoch - 5ms/step
Epoch 17/20
3375/3375 - 16s - loss: 0.0033 - accuracy: 0.9988 - val_loss: 0.0976 -
val_accuracy: 0.9877 - 16s/epoch - 5ms/step
Epoch 18/20
3375/3375 - 16s - loss: 0.0014 - accuracy: 0.9995 - val_loss: 0.0923 -
val_accuracy: 0.9890 - 16s/epoch - 5ms/step
Epoch 19/20
```

```
3375/3375 - 16s - loss: 0.0019 - accuracy: 0.9994 - val_loss: 0.1028 -  
val_accuracy: 0.9878 - 16s/epoch - 5ms/step
```

```
Epoch 20/20
```

```
3375/3375 - 16s - loss: 0.0021 - accuracy: 0.9994 - val_loss: 0.0944 -  
val_accuracy: 0.9883 - 16s/epoch - 5ms/step
```

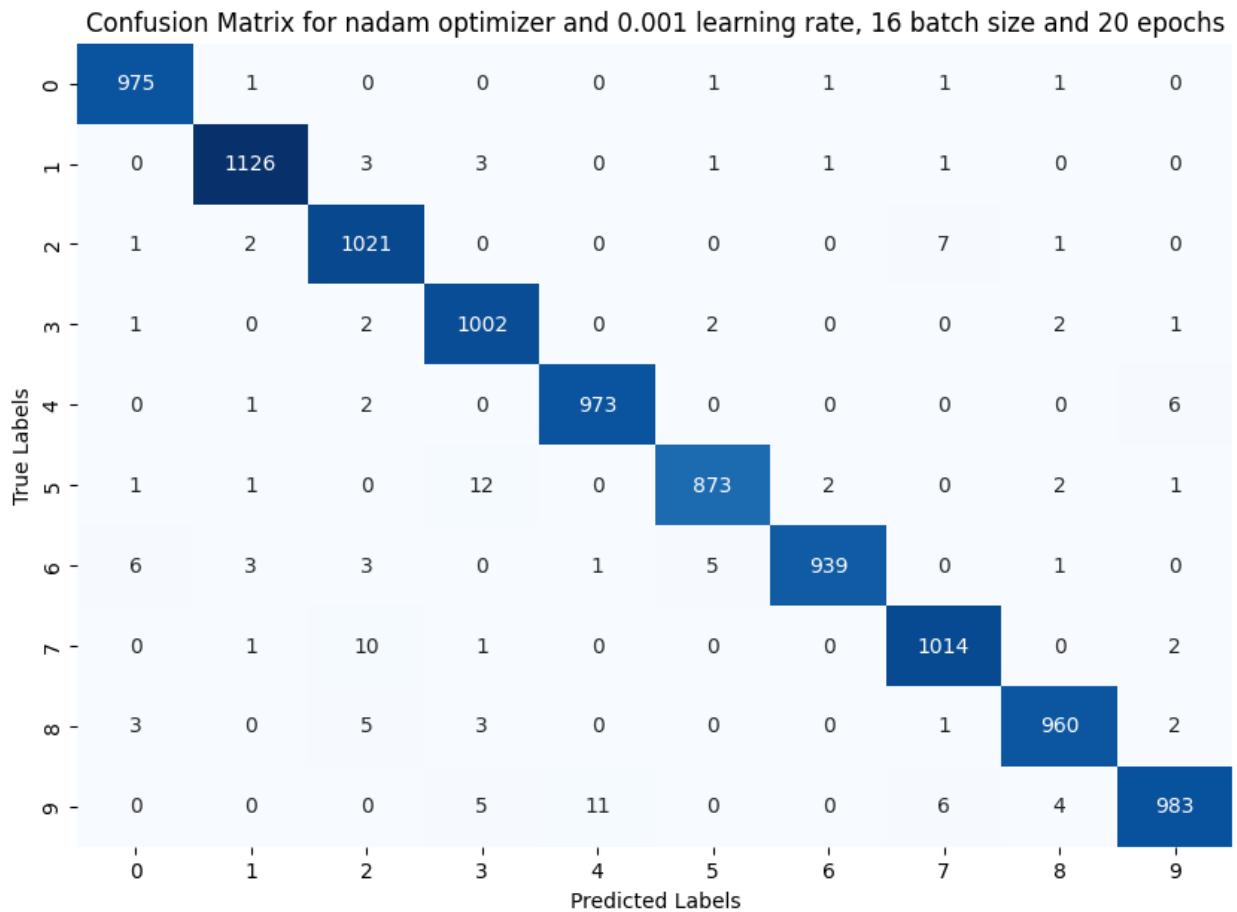
```
313/313 [=====] - 1s 2ms/step
```

```
Confusion Matrix nadam optimizer and the learning_rate is 0.001, 16  
batch size and 20 epochs:
```

```
[[ 975   1   0   0   0   1   1   1   1   0]  
 [  0 1126   3   3   0   1   1   1   0   0]  
 [  1   2 1021   0   0   0   0   7   1   0]  
 [  1   0   2 1002   0   2   0   0   2   1]  
 [  0   1   2   0 973   0   0   0   0   6]  
 [  1   1   0   12   0 873   2   0   2   1]  
 [  6   3   3   0   1   5 939   0   1   0]  
 [  0   1   10   1   0   0   0 1014   0   2]  
 [  3   0   5   3   0   0   0   1 960   2]  
 [  0   0   0   5   11   0   0   6   4 983]]
```

```
Precision: 0.9866
```

```
Recall: 0.9866
```

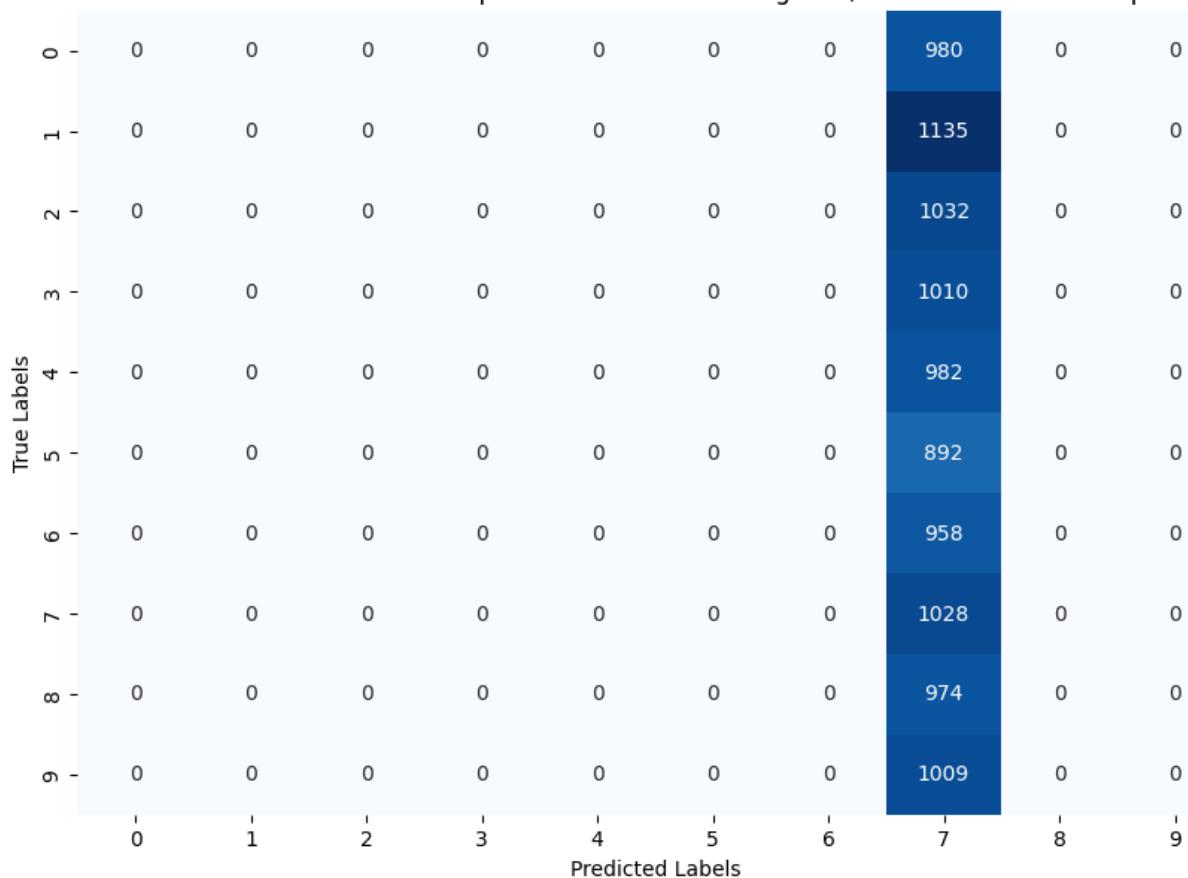


```
Training with adamw optimizer and the learning_rate is 0.1, 64 batch size and 5 epochs...
Epoch 1/5
844/844 - 6s - loss: 2.5301 - accuracy: 0.1047 - val_loss: 2.3100 -
val_accuracy: 0.0952 - 6s/epoch - 7ms/step
Epoch 2/5
844/844 - 5s - loss: 2.3092 - accuracy: 0.1051 - val_loss: 2.3081 -
val_accuracy: 0.0995 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 2.3092 - accuracy: 0.1068 - val_loss: 2.3050 -
val_accuracy: 0.1050 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 2.3098 - accuracy: 0.1021 - val_loss: 2.3151 -
val_accuracy: 0.1050 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 2.3086 - accuracy: 0.1053 - val_loss: 2.3042 -
val_accuracy: 0.1113 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 64 batch size and 5 epochs:
[[ 0  0  0  0  0  0  980  0  0]
 [ 0  0  0  0  0  0  1135  0  0]
 [ 0  0  0  0  0  0  1032  0  0]
 [ 0  0  0  0  0  0  1010  0  0]
 [ 0  0  0  0  0  0  982  0  0]
 [ 0  0  0  0  0  0  892  0  0]
 [ 0  0  0  0  0  0  958  0  0]
 [ 0  0  0  0  0  0  1028  0  0]
 [ 0  0  0  0  0  0  974  0  0]
 [ 0  0  0  0  0  0  1009  0  0]]
```

Precision: 0.0106
Recall: 0.1028

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

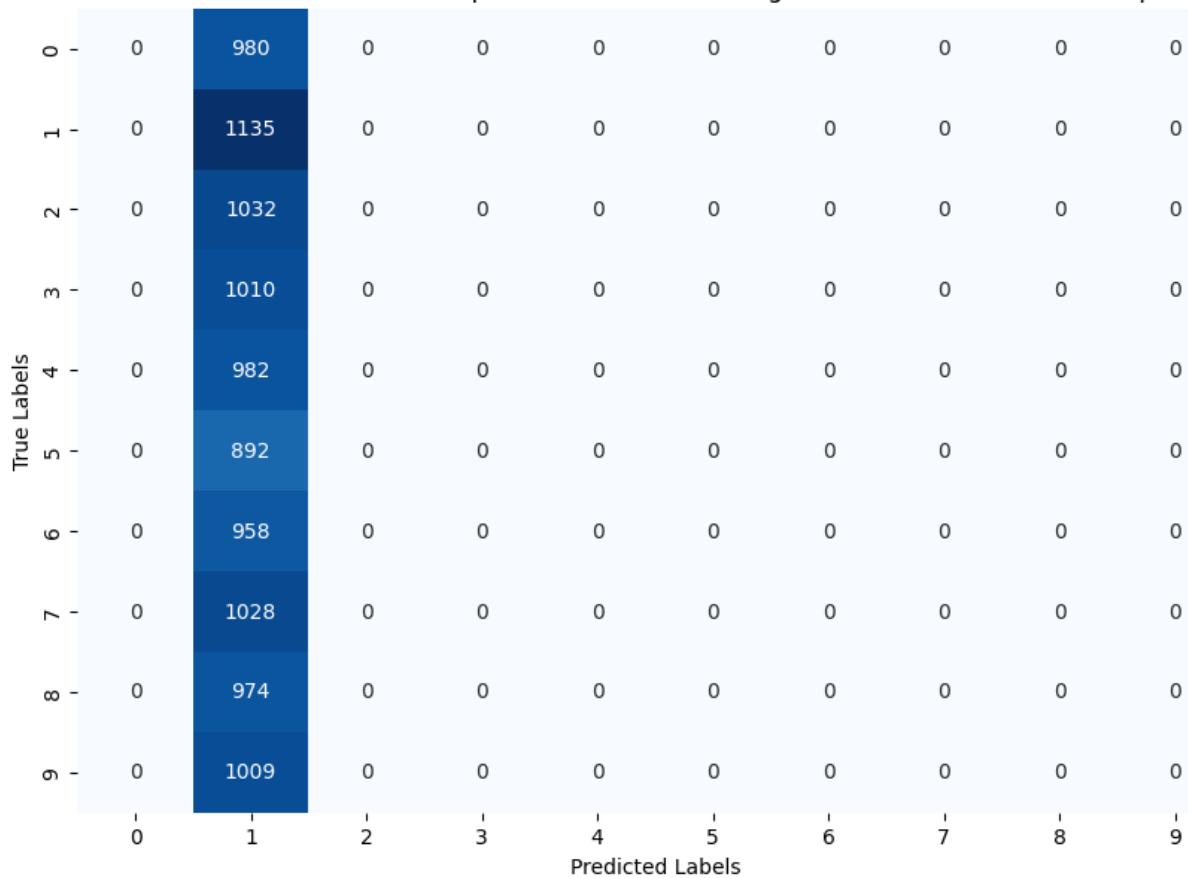
Confusion Matrix for adamw optimizer and 0.1 learning rate, 64 batch size and 5 epochs



```
Training with adamw optimizer and the learning_rate is 0.1, 64 batch size and 15 epochs...
Epoch 1/15
844/844 - 5s - loss: 2.6451 - accuracy: 0.1059 - val_loss: 2.3144 -
val_accuracy: 0.1113 - 5s/epoch - 6ms/step
Epoch 2/15
844/844 - 5s - loss: 2.3099 - accuracy: 0.1051 - val_loss: 2.3051 -
val_accuracy: 0.1050 - 5s/epoch - 6ms/step
Epoch 3/15
844/844 - 5s - loss: 2.3097 - accuracy: 0.1012 - val_loss: 2.3048 -
val_accuracy: 0.1113 - 5s/epoch - 6ms/step
Epoch 4/15
844/844 - 5s - loss: 2.3101 - accuracy: 0.1020 - val_loss: 2.3033 -
val_accuracy: 0.1050 - 5s/epoch - 6ms/step
Epoch 5/15
844/844 - 5s - loss: 2.3098 - accuracy: 0.1045 - val_loss: 2.3230 -
val_accuracy: 0.0995 - 5s/epoch - 6ms/step
Epoch 6/15
844/844 - 5s - loss: 2.3088 - accuracy: 0.1057 - val_loss: 2.3249 -
val_accuracy: 0.0978 - 5s/epoch - 6ms/step
Epoch 7/15
```

```
844/844 - 5s - loss: 2.3103 - accuracy: 0.1041 - val_loss: 2.3099 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 2.3101 - accuracy: 0.1044 - val_loss: 2.3112 -  
val_accuracy: 0.1045 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 2.3100 - accuracy: 0.1044 - val_loss: 2.3096 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 2.3102 - accuracy: 0.1030 - val_loss: 2.3065 -  
val_accuracy: 0.1045 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 2.3105 - accuracy: 0.1032 - val_loss: 2.3160 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 2.3098 - accuracy: 0.1023 - val_loss: 2.3102 -  
val_accuracy: 0.1113 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 2.3096 - accuracy: 0.1063 - val_loss: 2.3074 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 2.3094 - accuracy: 0.1045 - val_loss: 2.3121 -  
val_accuracy: 0.1113 - 5s/epoch - 5ms/step  
Epoch 15/15  
844/844 - 5s - loss: 2.3095 - accuracy: 0.1025 - val_loss: 2.3140 -  
val_accuracy: 0.1050 - 5s/epoch - 5ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 64  
batch size and 15 epochs:  
[[ 0 980 0 0 0 0 0 0 0 ]  
[ 0 1135 0 0 0 0 0 0 0 ]  
[ 0 1032 0 0 0 0 0 0 0 ]  
[ 0 1010 0 0 0 0 0 0 0 ]  
[ 0 982 0 0 0 0 0 0 0 ]  
[ 0 892 0 0 0 0 0 0 0 ]  
[ 0 958 0 0 0 0 0 0 0 ]  
[ 0 1028 0 0 0 0 0 0 0 ]  
[ 0 974 0 0 0 0 0 0 0 ]  
[ 0 1009 0 0 0 0 0 0 0 ]]  
Precision: 0.0129  
Recall: 0.1135  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adamw optimizer and 0.1 learning rate, 64 batch size and 15 epochs



```
Training with adamw optimizer and the learning_rate is 0.1, 64 batch size and 20 epochs...
Epoch 1/20
844/844 - 6s - loss: 2.5491 - accuracy: 0.1050 - val_loss: 2.3110 -
val_accuracy: 0.1050 - 6s/epoch - 7ms/step
Epoch 2/20
844/844 - 5s - loss: 2.3098 - accuracy: 0.1040 - val_loss: 2.3049 -
val_accuracy: 0.1113 - 5s/epoch - 6ms/step
Epoch 3/20
844/844 - 5s - loss: 2.3092 - accuracy: 0.1060 - val_loss: 2.3064 -
val_accuracy: 0.1045 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: 2.3100 - accuracy: 0.1035 - val_loss: 2.3163 -
val_accuracy: 0.0952 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: 2.3104 - accuracy: 0.1034 - val_loss: 2.3114 -
val_accuracy: 0.0995 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 2.3100 - accuracy: 0.1034 - val_loss: 2.3073 -
val_accuracy: 0.1113 - 5s/epoch - 6ms/step
Epoch 7/20
```

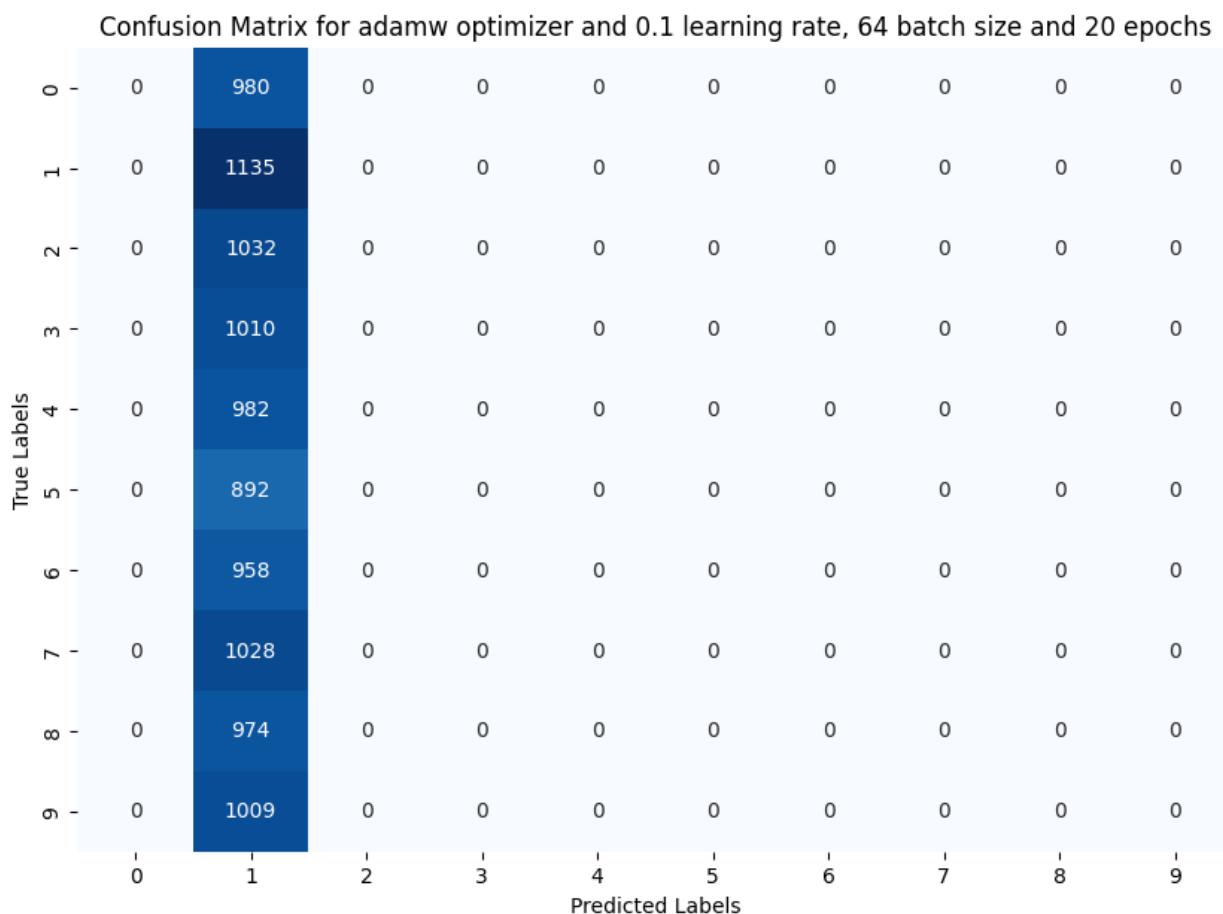
```
844/844 - 5s - loss: 2.3095 - accuracy: 0.1045 - val_loss: 2.3098 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 8/20  
844/844 - 5s - loss: 2.3100 - accuracy: 0.1030 - val_loss: 2.3070 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 9/20  
844/844 - 5s - loss: 2.3090 - accuracy: 0.1051 - val_loss: 2.3222 -  
val_accuracy: 0.1113 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: 2.3106 - accuracy: 0.1032 - val_loss: 2.3044 -  
val_accuracy: 0.0992 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: 2.3086 - accuracy: 0.1045 - val_loss: 2.3117 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 5s - loss: 2.3093 - accuracy: 0.1045 - val_loss: 2.3078 -  
val_accuracy: 0.1113 - 5s/epoch - 6ms/step  
Epoch 13/20  
844/844 - 5s - loss: 2.3097 - accuracy: 0.1041 - val_loss: 2.3133 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 14/20  
844/844 - 5s - loss: 2.3100 - accuracy: 0.1029 - val_loss: 2.3077 -  
val_accuracy: 0.0995 - 5s/epoch - 6ms/step  
Epoch 15/20  
844/844 - 5s - loss: 2.3090 - accuracy: 0.1046 - val_loss: 2.3203 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: 2.3102 - accuracy: 0.1050 - val_loss: 2.3080 -  
val_accuracy: 0.0978 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: 2.3091 - accuracy: 0.1033 - val_loss: 2.3162 -  
val_accuracy: 0.0992 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: 2.3094 - accuracy: 0.1059 - val_loss: 2.3163 -  
val_accuracy: 0.1000 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: 2.3096 - accuracy: 0.1022 - val_loss: 2.3210 -  
val_accuracy: 0.0960 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: 2.3097 - accuracy: 0.1032 - val_loss: 2.3254 -  
val_accuracy: 0.1050 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 64  
batch size and 20 epochs:  
[[ 0 980 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0]]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0]
[ 0 958 0 0 0 0 0 0 0 0]
[ 0 1028 0 0 0 0 0 0 0 0]
[ 0 974 0 0 0 0 0 0 0 0]
[ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training with adamw optimizer and the learning_rate is 0.1, 128 batch size and 5 epochs...

Epoch 1/5

422/422 - 4s - loss: 3.1859 - accuracy: 0.1087 - val_loss: 2.3064 -
val_accuracy: 0.1045 - 4s/epoch - 10ms/step

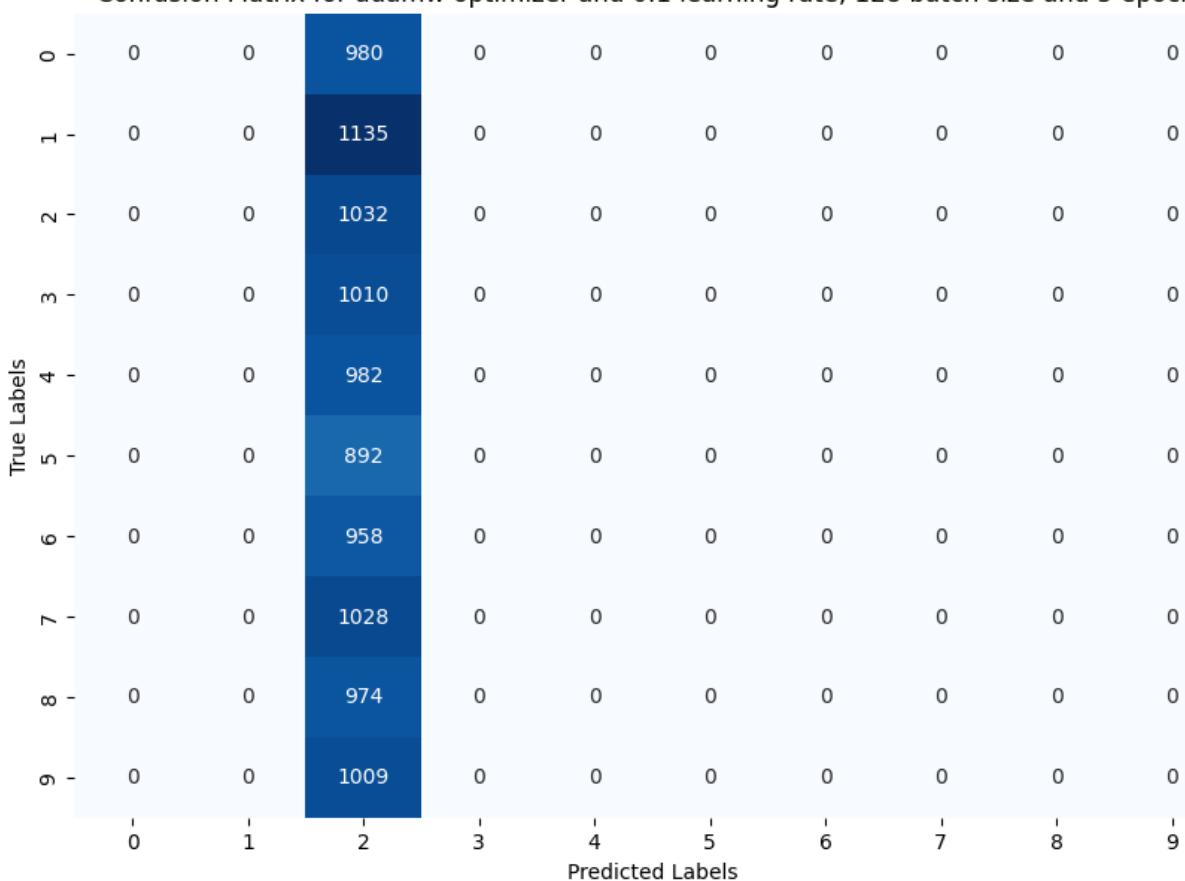
Epoch 2/5

422/422 - 3s - loss: 2.3061 - accuracy: 0.1041 - val_loss: 2.3050 -

```
val_accuracy: 0.1113 - 3s/epoch - 8ms/step
Epoch 3/5
422/422 - 3s - loss: 2.3070 - accuracy: 0.1031 - val_loss: 2.3037 -
val_accuracy: 0.1113 - 3s/epoch - 8ms/step
Epoch 4/5
422/422 - 3s - loss: 2.3073 - accuracy: 0.1048 - val_loss: 2.3091 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 5/5
422/422 - 3s - loss: 2.3072 - accuracy: 0.1035 - val_loss: 2.3093 -
val_accuracy: 0.1000 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 128
batch size and 5 epochs:
[[ 0  0 980  0  0  0  0  0  0  0]
 [ 0  0 1135  0  0  0  0  0  0  0]
 [ 0  0 1032  0  0  0  0  0  0  0]
 [ 0  0 1010  0  0  0  0  0  0  0]
 [ 0  0 982  0  0  0  0  0  0  0]
 [ 0  0 892  0  0  0  0  0  0  0]
 [ 0  0 958  0  0  0  0  0  0  0]
 [ 0  0 1028  0  0  0  0  0  0  0]
 [ 0  0 974  0  0  0  0  0  0  0]
 [ 0  0 1009  0  0  0  0  0  0  0]]]
Precision: 0.0107
Recall: 0.1032

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adamw optimizer and 0.1 learning rate, 128 batch size and 5 epochs



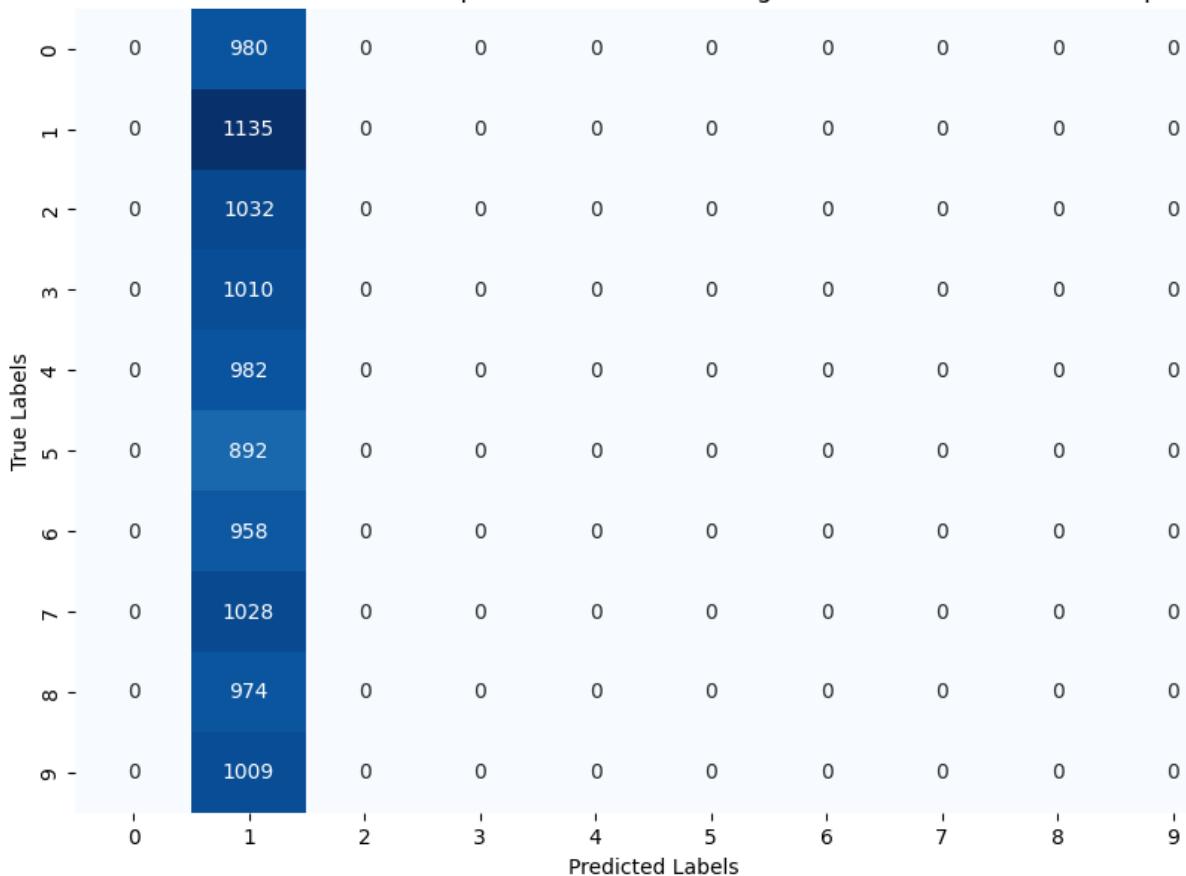
```
Training with adamw optimizer and the learning_rate is 0.1, 128 batch size and 15 epochs...
Epoch 1/15
422/422 - 4s - loss: 2.9569 - accuracy: 0.1091 - val_loss: 2.3116 -
val_accuracy: 0.1045 - 4s/epoch - 10ms/step
Epoch 2/15
422/422 - 3s - loss: 2.3066 - accuracy: 0.1059 - val_loss: 2.3134 -
val_accuracy: 0.0952 - 3s/epoch - 8ms/step
Epoch 3/15
422/422 - 3s - loss: 2.3067 - accuracy: 0.1066 - val_loss: 2.3087 -
val_accuracy: 0.0978 - 3s/epoch - 8ms/step
Epoch 4/15
422/422 - 3s - loss: 2.3069 - accuracy: 0.1054 - val_loss: 2.3044 -
val_accuracy: 0.1045 - 3s/epoch - 8ms/step
Epoch 5/15
422/422 - 3s - loss: 2.3070 - accuracy: 0.1048 - val_loss: 2.3129 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 6/15
422/422 - 3s - loss: 2.3079 - accuracy: 0.1053 - val_loss: 2.3078 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 7/15
```

```
422/422 - 3s - loss: 2.3078 - accuracy: 0.1044 - val_loss: 2.3069 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 8/15
422/422 - 3s - loss: 2.3067 - accuracy: 0.1052 - val_loss: 2.3069 -
val_accuracy: 0.1000 - 3s/epoch - 8ms/step
Epoch 9/15
422/422 - 3s - loss: 2.3072 - accuracy: 0.1073 - val_loss: 2.3069 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 10/15
422/422 - 3s - loss: 2.3072 - accuracy: 0.1045 - val_loss: 2.3072 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 11/15
422/422 - 3s - loss: 2.3069 - accuracy: 0.1037 - val_loss: 2.3038 -
val_accuracy: 0.1000 - 3s/epoch - 7ms/step
Epoch 12/15
422/422 - 3s - loss: 2.3069 - accuracy: 0.1044 - val_loss: 2.3076 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 13/15
422/422 - 3s - loss: 2.3069 - accuracy: 0.1056 - val_loss: 2.3139 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
Epoch 14/15
422/422 - 3s - loss: 2.3076 - accuracy: 0.1041 - val_loss: 2.3096 -
val_accuracy: 0.0952 - 3s/epoch - 8ms/step
Epoch 15/15
422/422 - 3s - loss: 2.3069 - accuracy: 0.1064 - val_loss: 2.3073 -
val_accuracy: 0.1050 - 3s/epoch - 8ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 128
batch size and 15 epochs:
[[ 0 980 0 0 0 0 0 0 0 0]
 [ 0 1135 0 0 0 0 0 0 0 0]
 [ 0 1032 0 0 0 0 0 0 0 0]
 [ 0 1010 0 0 0 0 0 0 0 0]
 [ 0 982 0 0 0 0 0 0 0 0]
 [ 0 892 0 0 0 0 0 0 0 0]
 [ 0 958 0 0 0 0 0 0 0 0]
 [ 0 1028 0 0 0 0 0 0 0 0]
 [ 0 974 0 0 0 0 0 0 0 0]
 [ 0 1009 0 0 0 0 0 0 0 0]]
```

Precision: 0.0129
Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adamw optimizer and 0.1 learning rate, 128 batch size and 15 epochs



```
Training with adamw optimizer and the learning_rate is 0.1, 128 batch size and 20 epochs...
Epoch 1/20
422/422 - 4s - loss: 2.6056 - accuracy: 0.1102 - val_loss: 2.3178 - val_accuracy: 0.0952 - 4s/epoch - 10ms/step
Epoch 2/20
422/422 - 3s - loss: 2.3072 - accuracy: 0.1039 - val_loss: 2.3111 - val_accuracy: 0.0995 - 3s/epoch - 8ms/step
Epoch 3/20
422/422 - 3s - loss: 2.3068 - accuracy: 0.1049 - val_loss: 2.3072 - val_accuracy: 0.0992 - 3s/epoch - 8ms/step
Epoch 4/20
422/422 - 3s - loss: 2.3070 - accuracy: 0.1046 - val_loss: 2.3076 - val_accuracy: 0.0952 - 3s/epoch - 8ms/step
Epoch 5/20
422/422 - 3s - loss: 2.3069 - accuracy: 0.1033 - val_loss: 2.3141 - val_accuracy: 0.0960 - 3s/epoch - 8ms/step
Epoch 6/20
422/422 - 3s - loss: 2.3078 - accuracy: 0.1055 - val_loss: 2.3127 - val_accuracy: 0.1000 - 3s/epoch - 8ms/step
Epoch 7/20
```

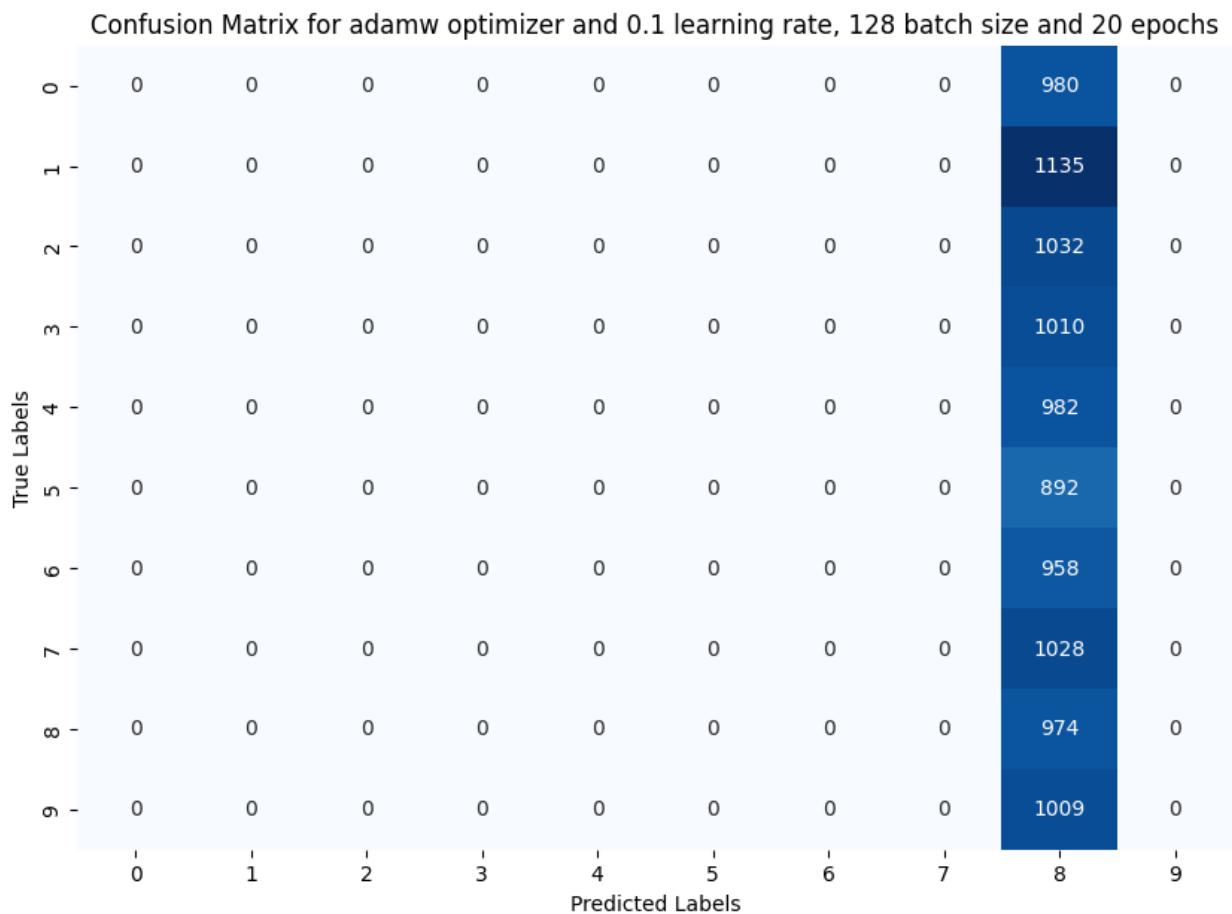
```
422/422 - 3s - loss: 2.3067 - accuracy: 0.1065 - val_loss: 2.3065 -  
val_accuracy: 0.1113 - 3s/epoch - 8ms/step  
Epoch 8/20  
422/422 - 3s - loss: 2.3074 - accuracy: 0.1025 - val_loss: 2.3041 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 9/20  
422/422 - 3s - loss: 2.3061 - accuracy: 0.1063 - val_loss: 2.3047 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 10/20  
422/422 - 3s - loss: 2.3073 - accuracy: 0.1052 - val_loss: 2.3037 -  
val_accuracy: 0.0992 - 3s/epoch - 8ms/step  
Epoch 11/20  
422/422 - 3s - loss: 2.3074 - accuracy: 0.1057 - val_loss: 2.3105 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 12/20  
422/422 - 3s - loss: 2.3067 - accuracy: 0.1052 - val_loss: 2.3053 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 13/20  
422/422 - 3s - loss: 2.3075 - accuracy: 0.1032 - val_loss: 2.3079 -  
val_accuracy: 0.1000 - 3s/epoch - 8ms/step  
Epoch 14/20  
422/422 - 3s - loss: 2.3068 - accuracy: 0.1062 - val_loss: 2.3105 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 15/20  
422/422 - 3s - loss: 2.3078 - accuracy: 0.1060 - val_loss: 2.3090 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 16/20  
422/422 - 3s - loss: 2.3075 - accuracy: 0.1053 - val_loss: 2.3060 -  
val_accuracy: 0.1045 - 3s/epoch - 8ms/step  
Epoch 17/20  
422/422 - 3s - loss: 2.3078 - accuracy: 0.1051 - val_loss: 2.3110 -  
val_accuracy: 0.0915 - 3s/epoch - 8ms/step  
Epoch 18/20  
422/422 - 3s - loss: 2.3066 - accuracy: 0.1027 - val_loss: 2.3045 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 19/20  
422/422 - 3s - loss: 2.3074 - accuracy: 0.1039 - val_loss: 2.3076 -  
val_accuracy: 0.1050 - 3s/epoch - 8ms/step  
Epoch 20/20  
422/422 - 3s - loss: 2.3070 - accuracy: 0.1042 - val_loss: 2.3111 -  
val_accuracy: 0.0995 - 3s/epoch - 7ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 128  
batch size and 20 epochs:  
[[ 0  0  0  0  0  0  0  0  980  0]  
 [ 0  0  0  0  0  0  0  0  1135  0]  
 [ 0  0  0  0  0  0  0  0  1032  0]  
 [ 0  0  0  0  0  0  0  0  1010  0]  
 [ 0  0  0  0  0  0  0  0  982  0]]
```

```
[ 0  0  0  0  0  0  0  0  892  0]
[ 0  0  0  0  0  0  0  958  0]
[ 0  0  0  0  0  0  0 1028  0]
[ 0  0  0  0  0  0  0  974  0]
[ 0  0  0  0  0  0  0 1009  0]]
```

Precision: 0.0095

Recall: 0.0974

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



Training with adamw optimizer and the learning_rate is 0.1, 256 batch size and 5 epochs...

Epoch 1/5

211/211 - 3s - loss: 1.3060 - accuracy: 0.8513 - val_loss: 0.1927 - val_accuracy: 0.9425 - 3s/epoch - 15ms/step

Epoch 2/5

211/211 - 2s - loss: 0.1630 - accuracy: 0.9492 - val_loss: 0.1220 -

```
val_accuracy: 0.9672 - 2s/epoch - 12ms/step
Epoch 3/5
211/211 - 2s - loss: 0.1606 - accuracy: 0.9501 - val_loss: 0.1226 -
val_accuracy: 0.9635 - 2s/epoch - 11ms/step
Epoch 4/5
211/211 - 2s - loss: 0.1502 - accuracy: 0.9533 - val_loss: 0.1317 -
val_accuracy: 0.9583 - 2s/epoch - 11ms/step
Epoch 5/5
211/211 - 2s - loss: 0.1611 - accuracy: 0.9514 - val_loss: 0.2041 -
val_accuracy: 0.9350 - 2s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 256
batch size and 5 epochs:
[[ 915    0    6    3    0   25   21    1    8    1]
 [  0 1069    4    1    0    3    2    0   55    1]
 [  1    8  928   53    1    6    5   12   18    0]
 [  0    0    2  991    0    2    0    3   11    1]
 [  0    0    4    0  882    1   18    1    5   71]
 [  1    0    0   20    0  860    6    0    5    0]
 [  7    2    1    1    1   16  929    0    1    0]
 [  1    5    6   28    1    3    0  890    8   86]
 [  4    0    9   30    0   18    2    0  897   14]
 [  2    3    3   12    6   21    1    1    9  951]]
```

Precision: 0.9352
Recall: 0.9312

Confusion Matrix for adamw optimizer and 0.1 learning rate, 256 batch size and 5 epochs

	0	1	2	3	4	5	6	7	8	9	
0	915	0	6	3	0	25	21	1	8	1	
1	0	1069	4	1	0	3	2	0	55	1	
2	1	8	928	53	1	6	5	12	18	0	
3	0	0	2	991	0	2	0	3	11	1	
4	0	0	4	0	882	1	18	1	5	71	
5	1	0	0	20	0	860	6	0	5	0	
6	7	2	1	1	1	16	929	0	1	0	
7	1	5	6	28	1	3	0	890	8	86	
8	4	0	9	30	0	18	2	0	897	14	
9	2	3	3	12	6	21	1	1	9	951	
	0	1	2	3	4	5	6	7	8	9	
	True Labels										Predicted Labels

Training with adamw optimizer and the learning_rate is 0.1, 256 batch size and 15 epochs...

Epoch 1/15

211/211 - 3s - loss: 3.0429 - accuracy: 0.1118 - val_loss: 2.3034 - val_accuracy: 0.1050 - 3s/epoch - 15ms/step

Epoch 2/15

211/211 - 2s - loss: 2.3040 - accuracy: 0.1070 - val_loss: 2.3021 - val_accuracy: 0.1050 - 2s/epoch - 12ms/step

Epoch 3/15

211/211 - 2s - loss: 2.3050 - accuracy: 0.1046 - val_loss: 2.3058 - val_accuracy: 0.1113 - 2s/epoch - 12ms/step

Epoch 4/15

211/211 - 2s - loss: 2.3049 - accuracy: 0.1049 - val_loss: 2.3078 - val_accuracy: 0.1050 - 2s/epoch - 11ms/step

Epoch 5/15

211/211 - 2s - loss: 2.3045 - accuracy: 0.1059 - val_loss: 2.3052 - val_accuracy: 0.0952 - 2s/epoch - 11ms/step

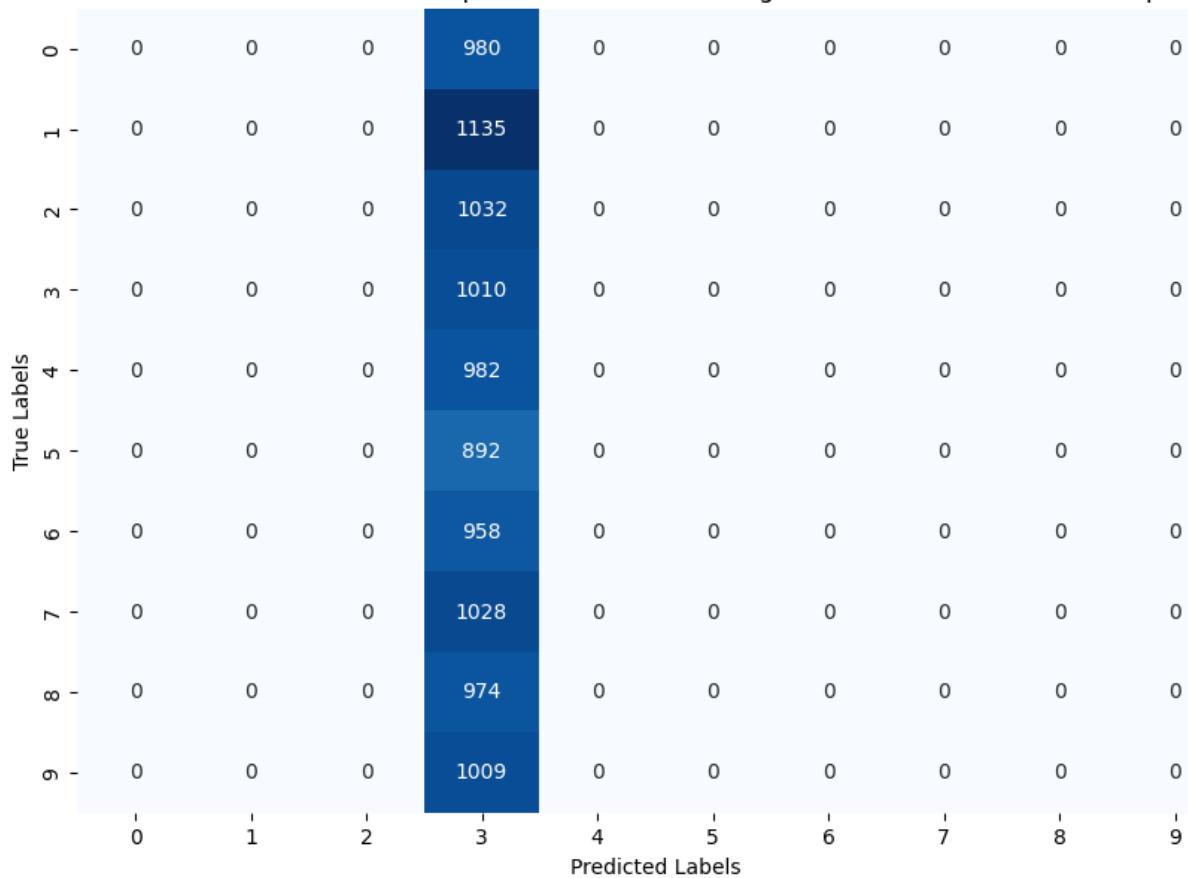
Epoch 6/15

211/211 - 3s - loss: 2.3060 - accuracy: 0.1053 - val_loss: 2.3057 - val_accuracy: 0.0960 - 3s/epoch - 12ms/step

Epoch 7/15

```
211/211 - 2s - loss: 2.3061 - accuracy: 0.1058 - val_loss: 2.3036 -  
val_accuracy: 0.1050 - 2s/epoch - 11ms/step  
Epoch 8/15  
211/211 - 2s - loss: 2.3056 - accuracy: 0.1066 - val_loss: 2.3072 -  
val_accuracy: 0.1000 - 2s/epoch - 11ms/step  
Epoch 9/15  
211/211 - 2s - loss: 2.3051 - accuracy: 0.1051 - val_loss: 2.3074 -  
val_accuracy: 0.1050 - 2s/epoch - 11ms/step  
Epoch 10/15  
211/211 - 2s - loss: 2.3055 - accuracy: 0.1050 - val_loss: 2.3044 -  
val_accuracy: 0.0978 - 2s/epoch - 11ms/step  
Epoch 11/15  
211/211 - 2s - loss: 2.3051 - accuracy: 0.1077 - val_loss: 2.3044 -  
val_accuracy: 0.1050 - 2s/epoch - 11ms/step  
Epoch 12/15  
211/211 - 2s - loss: 2.3052 - accuracy: 0.1042 - val_loss: 2.3077 -  
val_accuracy: 0.1050 - 2s/epoch - 12ms/step  
Epoch 13/15  
211/211 - 2s - loss: 2.3049 - accuracy: 0.1082 - val_loss: 2.3071 -  
val_accuracy: 0.1050 - 2s/epoch - 12ms/step  
Epoch 14/15  
211/211 - 2s - loss: 2.3051 - accuracy: 0.1070 - val_loss: 2.3071 -  
val_accuracy: 0.1050 - 2s/epoch - 12ms/step  
Epoch 15/15  
211/211 - 2s - loss: 2.3054 - accuracy: 0.1071 - val_loss: 2.3052 -  
val_accuracy: 0.1045 - 2s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 256  
batch size and 15 epochs:  
[[ 0  0  0 980  0  0  0  0  0  0]  
[ 0  0  0 1135  0  0  0  0  0  0]  
[ 0  0  0 1032  0  0  0  0  0  0]  
[ 0  0  0 1010  0  0  0  0  0  0]  
[ 0  0  0 982  0  0  0  0  0  0]  
[ 0  0  0 892  0  0  0  0  0  0]  
[ 0  0  0 958  0  0  0  0  0  0]  
[ 0  0  0 1028  0  0  0  0  0  0]  
[ 0  0  0 974  0  0  0  0  0  0]  
[ 0  0  0 1009  0  0  0  0  0  0]]  
Precision: 0.0102  
Recall: 0.1010  
  
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/  
_classification.py:1344: UndefinedMetricWarning: Precision is ill-  
defined and being set to 0.0 in labels with no predicted samples. Use  
'zero_division' parameter to control this behavior.  
_warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adamw optimizer and 0.1 learning rate, 256 batch size and 15 epochs



```
Training with adamw optimizer and the learning_rate is 0.1, 256 batch size and 20 epochs...
Epoch 1/20
211/211 - 3s - loss: 3.7845 - accuracy: 0.1090 - val_loss: 2.3059 - val_accuracy: 0.0960 - 3s/epoch - 15ms/step
Epoch 2/20
211/211 - 2s - loss: 2.3040 - accuracy: 0.1062 - val_loss: 2.3034 - val_accuracy: 0.1050 - 2s/epoch - 12ms/step
Epoch 3/20
211/211 - 2s - loss: 2.3045 - accuracy: 0.1067 - val_loss: 2.3046 - val_accuracy: 0.1113 - 2s/epoch - 11ms/step
Epoch 4/20
211/211 - 2s - loss: 2.3045 - accuracy: 0.1063 - val_loss: 2.3046 - val_accuracy: 0.1113 - 2s/epoch - 12ms/step
Epoch 5/20
211/211 - 2s - loss: 2.3052 - accuracy: 0.1034 - val_loss: 2.3050 - val_accuracy: 0.1045 - 2s/epoch - 12ms/step
Epoch 6/20
211/211 - 2s - loss: 2.3046 - accuracy: 0.1079 - val_loss: 2.3032 - val_accuracy: 0.1050 - 2s/epoch - 11ms/step
Epoch 7/20
```

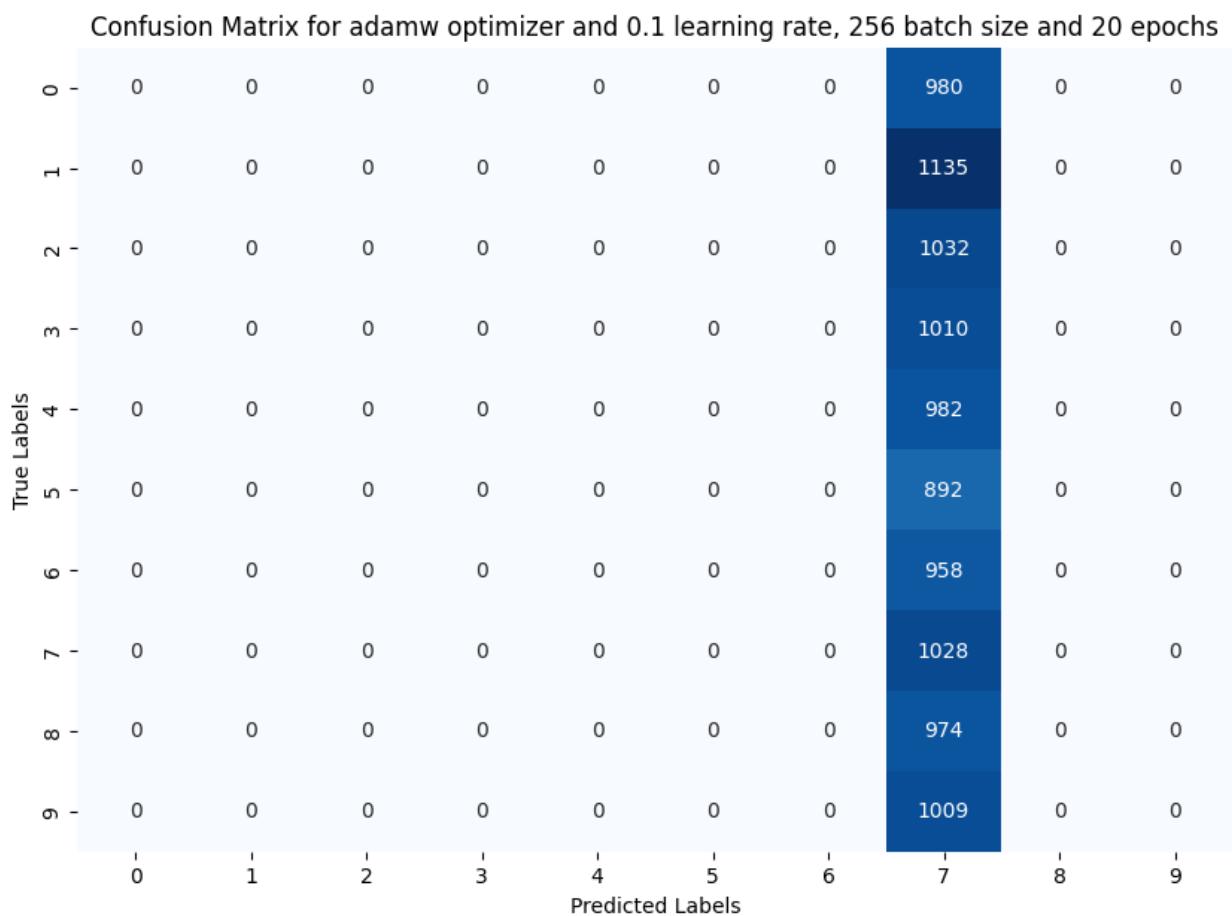
```
211/211 - 2s - loss: 2.3045 - accuracy: 0.1077 - val_loss: 2.3047 -  
val_accuracy: 0.1050 - 2s/epoch - 11ms/step  
Epoch 8/20  
211/211 - 2s - loss: 2.3048 - accuracy: 0.1078 - val_loss: 2.3048 -  
val_accuracy: 0.1045 - 2s/epoch - 12ms/step  
Epoch 9/20  
211/211 - 2s - loss: 2.3051 - accuracy: 0.1059 - val_loss: 2.3065 -  
val_accuracy: 0.1000 - 2s/epoch - 11ms/step  
Epoch 10/20  
211/211 - 2s - loss: 2.3059 - accuracy: 0.1042 - val_loss: 2.3028 -  
val_accuracy: 0.1113 - 2s/epoch - 11ms/step  
Epoch 11/20  
211/211 - 2s - loss: 2.3045 - accuracy: 0.1066 - val_loss: 2.3062 -  
val_accuracy: 0.0978 - 2s/epoch - 11ms/step  
Epoch 12/20  
211/211 - 2s - loss: 2.3051 - accuracy: 0.1074 - val_loss: 2.3047 -  
val_accuracy: 0.1000 - 2s/epoch - 12ms/step  
Epoch 13/20  
211/211 - 2s - loss: 2.3051 - accuracy: 0.1034 - val_loss: 2.3033 -  
val_accuracy: 0.1000 - 2s/epoch - 11ms/step  
Epoch 14/20  
211/211 - 2s - loss: 2.3049 - accuracy: 0.1058 - val_loss: 2.3044 -  
val_accuracy: 0.1113 - 2s/epoch - 11ms/step  
Epoch 15/20  
211/211 - 2s - loss: 2.3049 - accuracy: 0.1070 - val_loss: 2.3058 -  
val_accuracy: 0.1050 - 2s/epoch - 11ms/step  
Epoch 16/20  
211/211 - 2s - loss: 2.3051 - accuracy: 0.1072 - val_loss: 2.3089 -  
val_accuracy: 0.0978 - 2s/epoch - 11ms/step  
Epoch 17/20  
211/211 - 2s - loss: 2.3052 - accuracy: 0.1078 - val_loss: 2.3058 -  
val_accuracy: 0.1050 - 2s/epoch - 11ms/step  
Epoch 18/20  
211/211 - 2s - loss: 2.3049 - accuracy: 0.1074 - val_loss: 2.3031 -  
val_accuracy: 0.1113 - 2s/epoch - 11ms/step  
Epoch 19/20  
211/211 - 2s - loss: 2.3046 - accuracy: 0.1051 - val_loss: 2.3053 -  
val_accuracy: 0.0960 - 2s/epoch - 11ms/step  
Epoch 20/20  
211/211 - 2s - loss: 2.3047 - accuracy: 0.1070 - val_loss: 2.3019 -  
val_accuracy: 0.1113 - 2s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 256  
batch size and 20 epochs:  
[[ 0  0  0  0  0  0  0  980  0  0]  
 [ 0  0  0  0  0  0  0  1135  0  0]  
 [ 0  0  0  0  0  0  0  1032  0  0]  
 [ 0  0  0  0  0  0  0  1010  0  0]  
 [ 0  0  0  0  0  0  0  982  0  0]]
```

```
[ 0  0  0  0  0  0  0  892  0  0]
[ 0  0  0  0  0  0  0  958  0  0]
[ 0  0  0  0  0  0  0 1028  0  0]
[ 0  0  0  0  0  0  0  974  0  0]
[ 0  0  0  0  0  0  0 1009  0  0]]
```

Precision: 0.0106

Recall: 0.1028

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```



Training with adamw optimizer and the learning_rate is 0.1, 16 batch size and 5 epochs...

Epoch 1/5

3375/3375 - 15s - loss: 2.4262 - accuracy: 0.1056 - val_loss: 2.3252 - val_accuracy: 0.1050 - 15s/epoch - 4ms/step

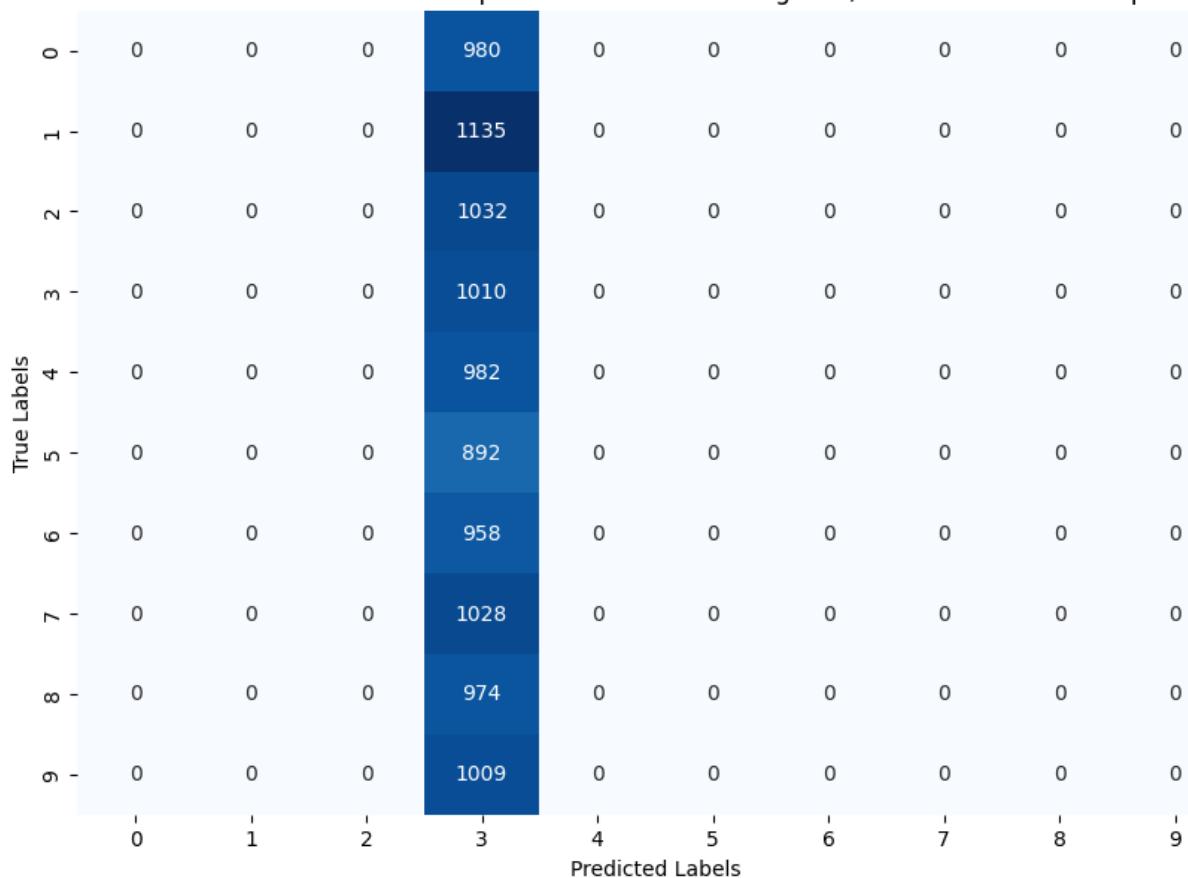
Epoch 2/5

3375/3375 - 14s - loss: 2.3184 - accuracy: 0.1038 - val_loss: 2.3235 -

```
val_accuracy: 0.1050 - 14s/epoch - 4ms/step
Epoch 3/5
3375/3375 - 14s - loss: 2.3176 - accuracy: 0.1019 - val_loss: 2.3144 -
val_accuracy: 0.1000 - 14s/epoch - 4ms/step
Epoch 4/5
3375/3375 - 14s - loss: 2.3185 - accuracy: 0.1036 - val_loss: 2.3283 -
val_accuracy: 0.0915 - 14s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 14s - loss: 2.3179 - accuracy: 0.1029 - val_loss: 2.3094 -
val_accuracy: 0.1045 - 14s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 16
batch size and 5 epochs:
[[ 0  0  0 980  0  0  0  0  0  0]
 [ 0  0  0 1135  0  0  0  0  0  0]
 [ 0  0  0 1032  0  0  0  0  0  0]
 [ 0  0  0 1010  0  0  0  0  0  0]
 [ 0  0  0 982  0  0  0  0  0  0]
 [ 0  0  0 892  0  0  0  0  0  0]
 [ 0  0  0 958  0  0  0  0  0  0]
 [ 0  0  0 1028  0  0  0  0  0  0]
 [ 0  0  0 974  0  0  0  0  0  0]
 [ 0  0  0 1009  0  0  0  0  0  0]]]
Precision: 0.0102
Recall: 0.1010

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
    _warn_prf(average, modifier, msg_start, len(result))
```

Confusion Matrix for adamw optimizer and 0.1 learning rate, 16 batch size and 5 epochs



Training with adamw optimizer and the learning_rate is 0.1, 16 batch size and 15 epochs...

Epoch 1/15

3375/3375 - 15s - loss: 2.4883 - accuracy: 0.1026 - val_loss: 2.3095 - val_accuracy: 0.0992 - 15s/epoch - 4ms/step

Epoch 2/15

3375/3375 - 14s - loss: 2.3180 - accuracy: 0.1053 - val_loss: 2.3091 - val_accuracy: 0.1050 - 14s/epoch - 4ms/step

Epoch 3/15

3375/3375 - 14s - loss: 2.3188 - accuracy: 0.1034 - val_loss: 2.3202 - val_accuracy: 0.1113 - 14s/epoch - 4ms/step

Epoch 4/15

3375/3375 - 14s - loss: 2.3179 - accuracy: 0.1031 - val_loss: 2.3140 - val_accuracy: 0.1000 - 14s/epoch - 4ms/step

Epoch 5/15

3375/3375 - 14s - loss: 2.3180 - accuracy: 0.1047 - val_loss: 2.3321 - val_accuracy: 0.0952 - 14s/epoch - 4ms/step

Epoch 6/15

3375/3375 - 14s - loss: 2.3184 - accuracy: 0.1045 - val_loss: 2.3202 - val_accuracy: 0.1045 - 14s/epoch - 4ms/step

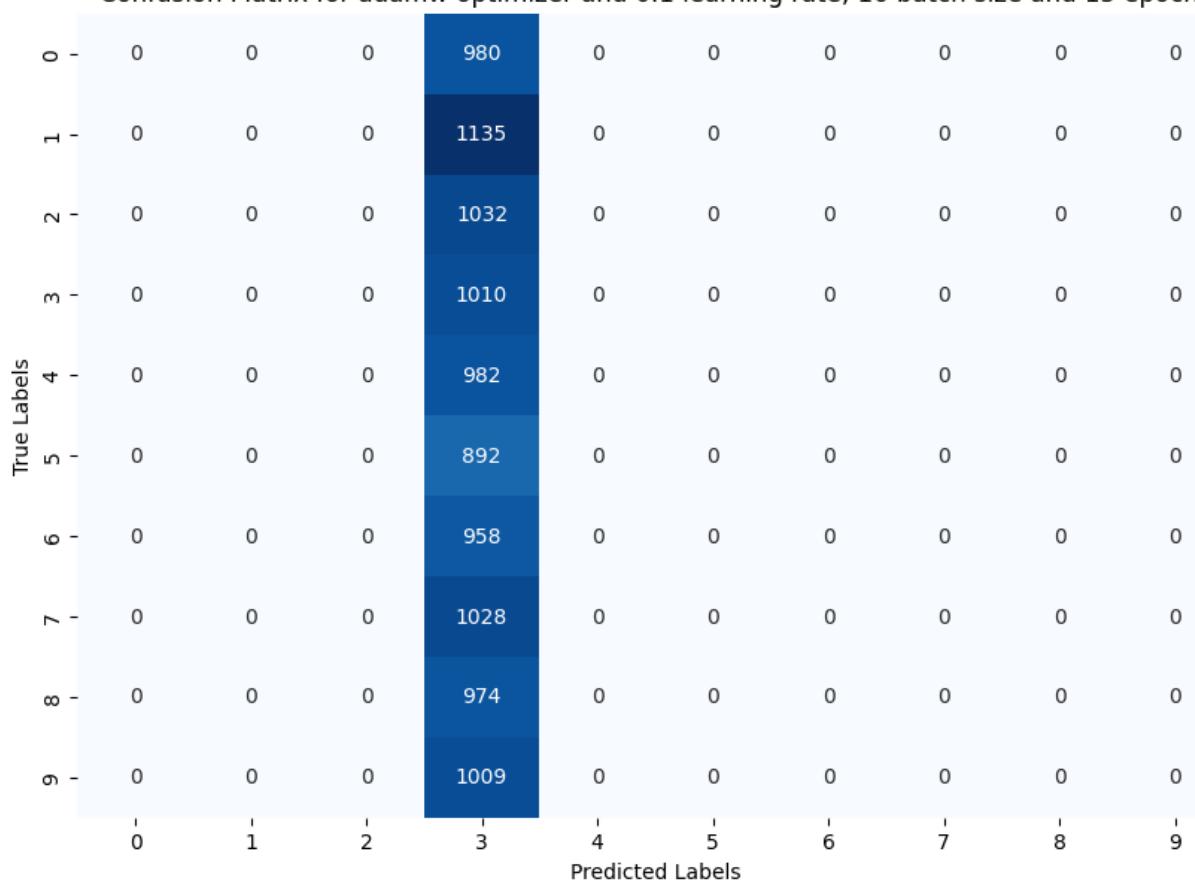
Epoch 7/15

```
3375/3375 - 14s - loss: 2.3179 - accuracy: 0.1019 - val_loss: 2.3112 -  
val_accuracy: 0.1000 - 14s/epoch - 4ms/step  
Epoch 8/15  
3375/3375 - 14s - loss: 2.3191 - accuracy: 0.1021 - val_loss: 2.3220 -  
val_accuracy: 0.0952 - 14s/epoch - 4ms/step  
Epoch 9/15  
3375/3375 - 14s - loss: 2.3181 - accuracy: 0.1037 - val_loss: 2.3104 -  
val_accuracy: 0.0960 - 14s/epoch - 4ms/step  
Epoch 10/15  
3375/3375 - 14s - loss: 2.3185 - accuracy: 0.1033 - val_loss: 2.3097 -  
val_accuracy: 0.1113 - 14s/epoch - 4ms/step  
Epoch 11/15  
3375/3375 - 13s - loss: 2.3183 - accuracy: 0.1030 - val_loss: 2.3159 -  
val_accuracy: 0.1000 - 13s/epoch - 4ms/step  
Epoch 12/15  
3375/3375 - 14s - loss: 2.3179 - accuracy: 0.1004 - val_loss: 2.3063 -  
val_accuracy: 0.0992 - 14s/epoch - 4ms/step  
Epoch 13/15  
3375/3375 - 14s - loss: 2.3193 - accuracy: 0.1021 - val_loss: 2.3173 -  
val_accuracy: 0.1113 - 14s/epoch - 4ms/step  
Epoch 14/15  
3375/3375 - 14s - loss: 2.3176 - accuracy: 0.1032 - val_loss: 2.3177 -  
val_accuracy: 0.0915 - 14s/epoch - 4ms/step  
Epoch 15/15  
3375/3375 - 14s - loss: 2.3185 - accuracy: 0.1016 - val_loss: 2.3153 -  
val_accuracy: 0.1045 - 14s/epoch - 4ms/step  
313/313 [=====] - 2s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 16  
batch size and 15 epochs:  
[[ 0  0  0 980  0  0  0  0  0  0]  
[ 0  0  0 1135  0  0  0  0  0  0]  
[ 0  0  0 1032  0  0  0  0  0  0]  
[ 0  0  0 1010  0  0  0  0  0  0]  
[ 0  0  0 982  0  0  0  0  0  0]  
[ 0  0  0 892  0  0  0  0  0  0]  
[ 0  0  0 958  0  0  0  0  0  0]  
[ 0  0  0 1028  0  0  0  0  0  0]  
[ 0  0  0 974  0  0  0  0  0  0]  
[ 0  0  0 1009  0  0  0  0  0  0]]  
Precision: 0.0102  
Recall: 0.1010  


```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
'zero_division' parameter to control this behavior.
 _warn_prf(average, modifier, msg_start, len(result))
```


```

Confusion Matrix for adamw optimizer and 0.1 learning rate, 16 batch size and 15 epochs



Training with adamw optimizer and the learning_rate is 0.1, 16 batch size and 20 epochs...

Epoch 1/20

3375/3375 - 15s - loss: 2.6165 - accuracy: 0.1022 - val_loss: 2.3273 - val_accuracy: 0.0995 - 15s/epoch - 4ms/step

Epoch 2/20

3375/3375 - 14s - loss: 2.3182 - accuracy: 0.1026 - val_loss: 2.3491 - val_accuracy: 0.0995 - 14s/epoch - 4ms/step

Epoch 3/20

3375/3375 - 14s - loss: 2.3177 - accuracy: 0.1038 - val_loss: 2.3063 - val_accuracy: 0.0960 - 14s/epoch - 4ms/step

Epoch 4/20

3375/3375 - 14s - loss: 2.3179 - accuracy: 0.1028 - val_loss: 2.3282 - val_accuracy: 0.1113 - 14s/epoch - 4ms/step

Epoch 5/20

3375/3375 - 14s - loss: 2.3177 - accuracy: 0.1032 - val_loss: 2.3127 - val_accuracy: 0.1050 - 14s/epoch - 4ms/step

Epoch 6/20

3375/3375 - 14s - loss: 2.3185 - accuracy: 0.1035 - val_loss: 2.3153 - val_accuracy: 0.1113 - 14s/epoch - 4ms/step

Epoch 7/20

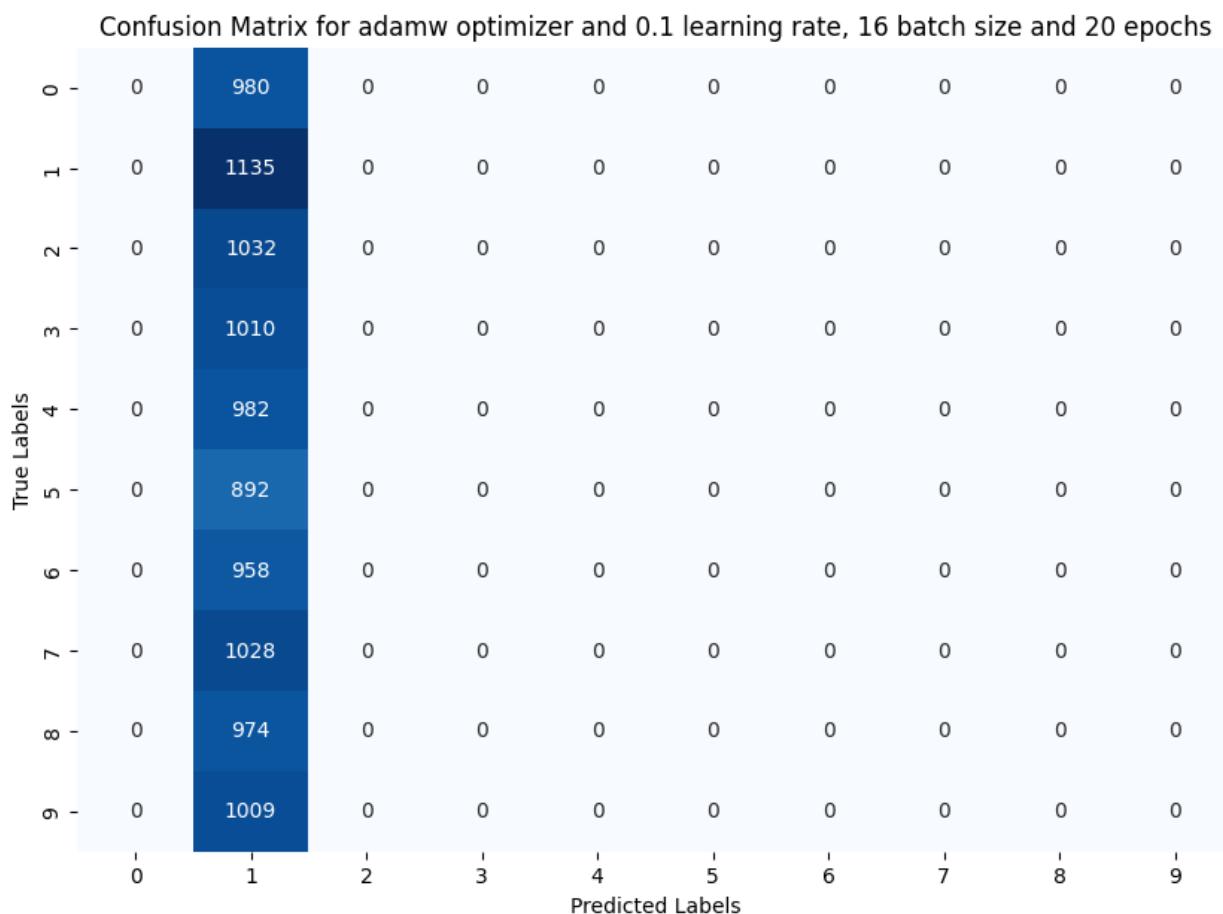
```
3375/3375 - 14s - loss: 2.3184 - accuracy: 0.1019 - val_loss: 2.3219 -  
val_accuracy: 0.1050 - 14s/epoch - 4ms/step  
Epoch 8/20  
3375/3375 - 14s - loss: 2.3183 - accuracy: 0.1025 - val_loss: 2.3129 -  
val_accuracy: 0.0978 - 14s/epoch - 4ms/step  
Epoch 9/20  
3375/3375 - 14s - loss: 2.3177 - accuracy: 0.1026 - val_loss: 2.3208 -  
val_accuracy: 0.1045 - 14s/epoch - 4ms/step  
Epoch 10/20  
3375/3375 - 14s - loss: 2.3183 - accuracy: 0.1029 - val_loss: 2.3195 -  
val_accuracy: 0.1000 - 14s/epoch - 4ms/step  
Epoch 11/20  
3375/3375 - 14s - loss: 2.3188 - accuracy: 0.1025 - val_loss: 2.3236 -  
val_accuracy: 0.0915 - 14s/epoch - 4ms/step  
Epoch 12/20  
3375/3375 - 14s - loss: 2.3178 - accuracy: 0.1013 - val_loss: 2.3236 -  
val_accuracy: 0.0992 - 14s/epoch - 4ms/step  
Epoch 13/20  
3375/3375 - 14s - loss: 2.3184 - accuracy: 0.1039 - val_loss: 2.3131 -  
val_accuracy: 0.0952 - 14s/epoch - 4ms/step  
Epoch 14/20  
3375/3375 - 14s - loss: 2.3177 - accuracy: 0.1028 - val_loss: 2.3211 -  
val_accuracy: 0.1050 - 14s/epoch - 4ms/step  
Epoch 15/20  
3375/3375 - 14s - loss: 2.3183 - accuracy: 0.1040 - val_loss: 2.3139 -  
val_accuracy: 0.0978 - 14s/epoch - 4ms/step  
Epoch 16/20  
3375/3375 - 14s - loss: 2.3178 - accuracy: 0.1040 - val_loss: 2.3107 -  
val_accuracy: 0.0992 - 14s/epoch - 4ms/step  
Epoch 17/20  
3375/3375 - 15s - loss: 2.3180 - accuracy: 0.1007 - val_loss: 2.3199 -  
val_accuracy: 0.1045 - 15s/epoch - 4ms/step  
Epoch 18/20  
3375/3375 - 14s - loss: 2.3181 - accuracy: 0.1048 - val_loss: 2.3139 -  
val_accuracy: 0.0978 - 14s/epoch - 4ms/step  
Epoch 19/20  
3375/3375 - 13s - loss: 2.3183 - accuracy: 0.1021 - val_loss: 2.3252 -  
val_accuracy: 0.0995 - 13s/epoch - 4ms/step  
Epoch 20/20  
3375/3375 - 14s - loss: 2.3190 - accuracy: 0.1025 - val_loss: 2.3109 -  
val_accuracy: 0.1050 - 14s/epoch - 4ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.1, 16  
batch size and 20 epochs:  
[[ 0 980 0 0 0 0 0 0 0]  
[ 0 1135 0 0 0 0 0 0 0]  
[ 0 1032 0 0 0 0 0 0 0]  
[ 0 1010 0 0 0 0 0 0 0]  
[ 0 982 0 0 0 0 0 0 0]]
```

```
[ [ 0 892 0 0 0 0 0 0 0 0 ]
[ 0 958 0 0 0 0 0 0 0 0 ]
[ 0 1028 0 0 0 0 0 0 0 0 ]
[ 0 974 0 0 0 0 0 0 0 0 ]
[ 0 1009 0 0 0 0 0 0 0 0 ] ]
```

Precision: 0.0129

Recall: 0.1135

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_classification.py:1344: UndefinedMetricWarning: Precision is ill-
defined and being set to 0.0 in labels with no predicted samples. Use
`zero_division` parameter to control this behavior.
_warn_prf(average, modifier, msg_start, len(result))
```



Training with adamw optimizer and the learning_rate is 0.01, 64 batch size and 5 epochs...

Epoch 1/5

844/844 - 6s - loss: 0.2878 - accuracy: 0.9345 - val_loss: 0.1257 - val_accuracy: 0.9682 - 6s/epoch - 7ms/step

Epoch 2/5

844/844 - 5s - loss: 0.1412 - accuracy: 0.9583 - val_loss: 0.1198 -

```
val_accuracy: 0.9630 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 0.1372 - accuracy: 0.9583 - val_loss: 0.0857 -
val_accuracy: 0.9752 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 0.1340 - accuracy: 0.9594 - val_loss: 0.1092 -
val_accuracy: 0.9687 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.1308 - accuracy: 0.9611 - val_loss: 0.1273 -
val_accuracy: 0.9637 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 64
batch size and 5 epochs:
[[ 946    0    1    0    6    0   18    3    1    5]
 [  0 1118    7    0    0    0    5    0    5    0]
 [  5    1 998    1    4    0    1   16    6    0]
 [  3    0   26 927    0    2    6   23   14    9]
 [  0    1    3    0 962    0    5    1    1    9]
 [  3    1    1   25    2 785   27    1   32   15]
 [  2    2    3    0    2    1 946    0    2    0]
 [  1    3   13    0    0    0    0 1006    1    4]
 [ 10    1   11    6   14    0   12   14  901    5]
 [  2    5    1    2   21    0    0   14    4 960]]]
Precision: 0.9557
Recall: 0.9549
```

Confusion Matrix for adamw optimizer and 0.01 learning rate, 64 batch size and 5 epochs

	0	1	2	3	4	5	6	7	8	9	10
True Labels	946	0	1	0	6	0	18	3	1	5	0
0	0	1118	7	0	0	0	5	0	5	0	0
1	5	1	998	1	4	0	1	16	6	0	0
2	3	0	26	927	0	2	6	23	14	9	0
3	0	1	3	0	962	0	5	1	1	9	0
4	3	1	1	25	2	785	27	1	32	15	0
5	2	2	3	0	2	1	946	0	2	0	0
6	1	3	13	0	0	0	0	1006	1	4	0
7	10	1	11	6	14	0	12	14	901	5	0
8	2	5	1	2	21	0	0	14	4	960	0
9	1	2	3	4	5	6	7	8	9	0	0
10	0	1	2	3	4	5	6	7	8	9	0
Predicted Labels											

Training with adamw optimizer and the learning_rate is 0.01, 64 batch size and 15 epochs...

Epoch 1/15

844/844 - 6s - loss: 0.2406 - accuracy: 0.9366 - val_loss: 0.1176 - val_accuracy: 0.9660 - 6s/epoch - 7ms/step

Epoch 2/15

844/844 - 5s - loss: 0.1314 - accuracy: 0.9606 - val_loss: 0.1292 - val_accuracy: 0.9638 - 5s/epoch - 6ms/step

Epoch 3/15

844/844 - 5s - loss: 0.1288 - accuracy: 0.9619 - val_loss: 0.1222 - val_accuracy: 0.9655 - 5s/epoch - 6ms/step

Epoch 4/15

844/844 - 5s - loss: 0.1261 - accuracy: 0.9621 - val_loss: 0.0969 - val_accuracy: 0.9743 - 5s/epoch - 6ms/step

Epoch 5/15

844/844 - 5s - loss: 0.1314 - accuracy: 0.9605 - val_loss: 0.0947 - val_accuracy: 0.9720 - 5s/epoch - 6ms/step

Epoch 6/15

844/844 - 5s - loss: 0.1294 - accuracy: 0.9603 - val_loss: 0.0964 - val_accuracy: 0.9740 - 5s/epoch - 6ms/step

Epoch 7/15

```
844/844 - 4s - loss: 0.1272 - accuracy: 0.9618 - val_loss: 0.1134 -  
val_accuracy: 0.9703 - 4s/epoch - 5ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.1264 - accuracy: 0.9616 - val_loss: 0.0930 -  
val_accuracy: 0.9732 - 5s/epoch - 5ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.1252 - accuracy: 0.9630 - val_loss: 0.1099 -  
val_accuracy: 0.9695 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.1307 - accuracy: 0.9602 - val_loss: 0.0967 -  
val_accuracy: 0.9723 - 5s/epoch - 6ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.1257 - accuracy: 0.9629 - val_loss: 0.0956 -  
val_accuracy: 0.9738 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.1275 - accuracy: 0.9615 - val_loss: 0.0884 -  
val_accuracy: 0.9757 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.1280 - accuracy: 0.9619 - val_loss: 0.0987 -  
val_accuracy: 0.9732 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.1260 - accuracy: 0.9619 - val_loss: 0.0990 -  
val_accuracy: 0.9732 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.1273 - accuracy: 0.9622 - val_loss: 0.1072 -  
val_accuracy: 0.9672 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 64  
batch size and 15 epochs:  
[[ 961 0 2 0 4 0 6 0 1 6]  
[ 0 1125 1 3 0 1 3 0 2 0]  
[ 8 2 989 6 3 0 2 6 16 0]  
[ 0 0 4 969 1 26 0 3 5 2]  
[ 1 1 1 0 952 0 5 0 0 22]  
[ 2 0 0 1 0 883 3 0 2 1]  
[ 7 3 0 0 1 4 939 0 4 0]  
[ 1 6 18 6 0 1 0 933 3 60]  
[ 6 0 4 4 9 16 1 4 921 9]  
[ 4 1 0 6 5 16 0 1 2 974]]  
Precision: 0.9653  
Recall: 0.9646
```

Confusion Matrix for adamw optimizer and 0.01 learning rate, 64 batch size and 15 epochs											
	0	1	2	3	4	5	6	7	8	9	
0	961	0	2	0	4	0	6	0	1	6	
1	0	1125	1	3	0	1	3	0	2	0	
2	8	2	989	6	3	0	2	6	16	0	
3	0	0	4	969	1	26	0	3	5	2	
4	1	1	1	0	952	0	5	0	0	22	
5	2	0	0	1	0	883	3	0	2	1	
6	7	3	0	0	1	4	939	0	4	0	
7	1	6	18	6	0	1	0	933	3	60	
8	6	0	4	4	9	16	1	4	921	9	
9	4	1	0	6	5	16	0	1	2	974	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	

```
Training with adamw optimizer and the learning_rate is 0.01, 64 batch size and 20 epochs...
Epoch 1/20
844/844 - 5s - loss: 0.2895 - accuracy: 0.9321 - val_loss: 0.1170 - val_accuracy: 0.9678 - 5s/epoch - 6ms/step
Epoch 2/20
844/844 - 5s - loss: 0.1413 - accuracy: 0.9582 - val_loss: 0.1296 - val_accuracy: 0.9647 - 5s/epoch - 6ms/step
Epoch 3/20
844/844 - 5s - loss: 0.1378 - accuracy: 0.9588 - val_loss: 0.1063 - val_accuracy: 0.9688 - 5s/epoch - 6ms/step
Epoch 4/20
844/844 - 5s - loss: 0.1344 - accuracy: 0.9592 - val_loss: 0.1242 - val_accuracy: 0.9653 - 5s/epoch - 6ms/step
Epoch 5/20
844/844 - 5s - loss: 0.1321 - accuracy: 0.9593 - val_loss: 0.1088 - val_accuracy: 0.9693 - 5s/epoch - 6ms/step
Epoch 6/20
844/844 - 5s - loss: 0.1272 - accuracy: 0.9613 - val_loss: 0.0983 - val_accuracy: 0.9753 - 5s/epoch - 6ms/step
Epoch 7/20
```

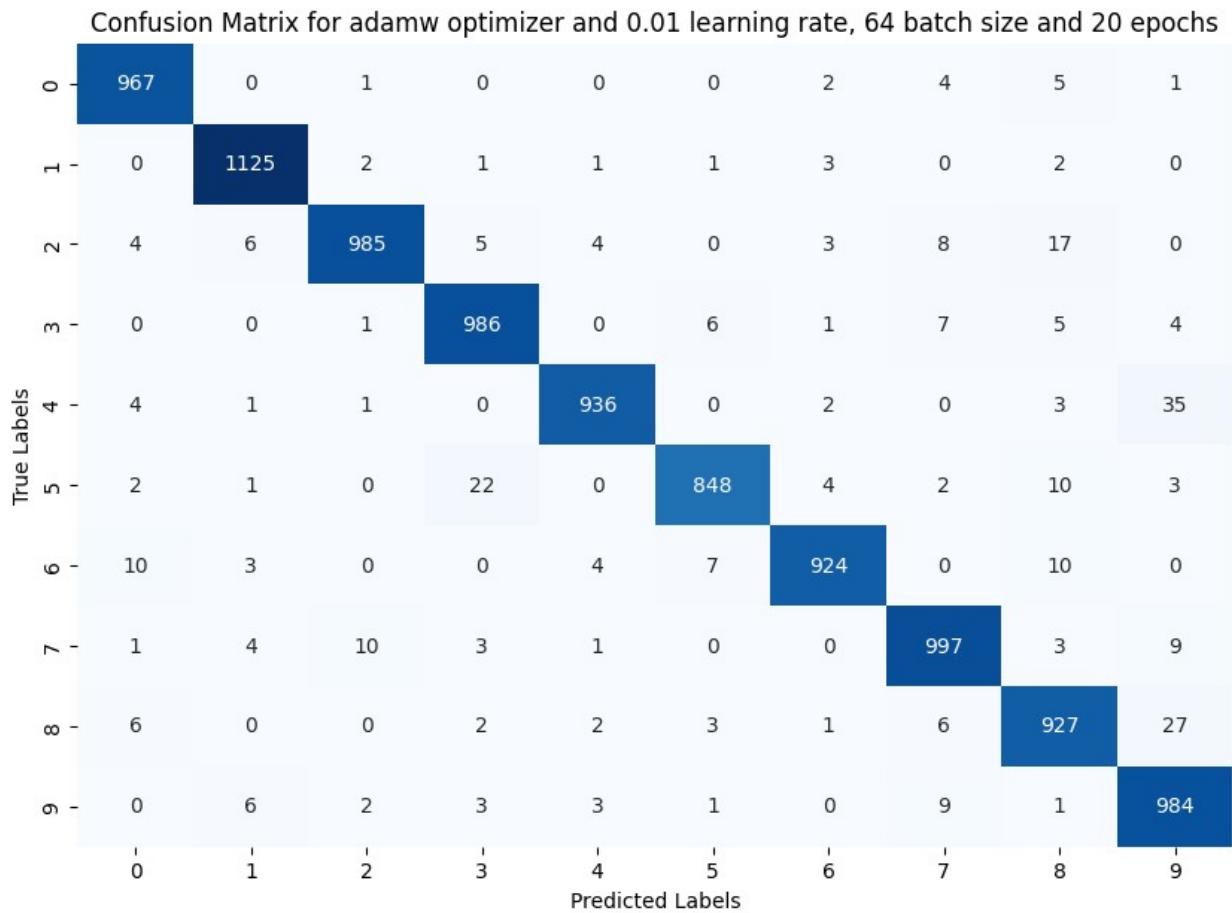
```

844/844 - 5s - loss: 0.1282 - accuracy: 0.9607 - val_loss: 0.1001 -
val_accuracy: 0.9698 - 5s/epoch - 6ms/step
Epoch 8/20
844/844 - 5s - loss: 0.1285 - accuracy: 0.9609 - val_loss: 0.1541 -
val_accuracy: 0.9598 - 5s/epoch - 6ms/step
Epoch 9/20
844/844 - 5s - loss: 0.1305 - accuracy: 0.9601 - val_loss: 0.0923 -
val_accuracy: 0.9737 - 5s/epoch - 5ms/step
Epoch 10/20
844/844 - 5s - loss: 0.1267 - accuracy: 0.9617 - val_loss: 0.1316 -
val_accuracy: 0.9602 - 5s/epoch - 6ms/step
Epoch 11/20
844/844 - 5s - loss: 0.1272 - accuracy: 0.9614 - val_loss: 0.1224 -
val_accuracy: 0.9657 - 5s/epoch - 6ms/step
Epoch 12/20
844/844 - 5s - loss: 0.1283 - accuracy: 0.9621 - val_loss: 0.1273 -
val_accuracy: 0.9607 - 5s/epoch - 6ms/step
Epoch 13/20
844/844 - 5s - loss: 0.1286 - accuracy: 0.9611 - val_loss: 0.1002 -
val_accuracy: 0.9728 - 5s/epoch - 6ms/step
Epoch 14/20
844/844 - 5s - loss: 0.1275 - accuracy: 0.9618 - val_loss: 0.1024 -
val_accuracy: 0.9708 - 5s/epoch - 6ms/step
Epoch 15/20
844/844 - 5s - loss: 0.1250 - accuracy: 0.9621 - val_loss: 0.1169 -
val_accuracy: 0.9680 - 5s/epoch - 6ms/step
Epoch 16/20
844/844 - 5s - loss: 0.1288 - accuracy: 0.9601 - val_loss: 0.0964 -
val_accuracy: 0.9707 - 5s/epoch - 6ms/step
Epoch 17/20
844/844 - 5s - loss: 0.1259 - accuracy: 0.9614 - val_loss: 0.1017 -
val_accuracy: 0.9688 - 5s/epoch - 6ms/step
Epoch 18/20
844/844 - 5s - loss: 0.1270 - accuracy: 0.9618 - val_loss: 0.1035 -
val_accuracy: 0.9710 - 5s/epoch - 6ms/step
Epoch 19/20
844/844 - 5s - loss: 0.1267 - accuracy: 0.9616 - val_loss: 0.1076 -
val_accuracy: 0.9702 - 5s/epoch - 6ms/step
Epoch 20/20
844/844 - 5s - loss: 0.1271 - accuracy: 0.9619 - val_loss: 0.1037 -
val_accuracy: 0.9702 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 64
batch size and 20 epochs:
[[ 967   0   1   0   0   0   2   4   5   1]
 [  0 1125   2   1   1   1   3   0   2   0]
 [  4   6  985   5   4   0   3   8  17   0]
 [  0   0   1  986   0   6   1   7   5   4]
 [  4   1   1   0  936   0   2   0   3  35]]
```

```
[ 2  1  0 22  0 848  4  2 10  3
[ 10 3  0  0  4  7 924  0 10  0]
[ 1  4 10  3  1  0  0 997  3  9]
[ 6  0  0  2  2  3  1  6 927 27]
[ 0  6  2  3  3  1  0  9  1 984]]
```

Precision: 0.9683

Recall: 0.9679



Training with adamw optimizer and the learning_rate is 0.01, 128 batch size and 5 epochs...

Epoch 1/5

422/422 - 4s - loss: 0.2797 - accuracy: 0.9351 - val_loss: 0.0961 - val_accuracy: 0.9717 - 4s/epoch - 9ms/step

Epoch 2/5

422/422 - 3s - loss: 0.0996 - accuracy: 0.9696 - val_loss: 0.1137 - val_accuracy: 0.9680 - 3s/epoch - 8ms/step

Epoch 3/5

422/422 - 3s - loss: 0.0918 - accuracy: 0.9723 - val_loss: 0.0969 - val_accuracy: 0.9708 - 3s/epoch - 8ms/step

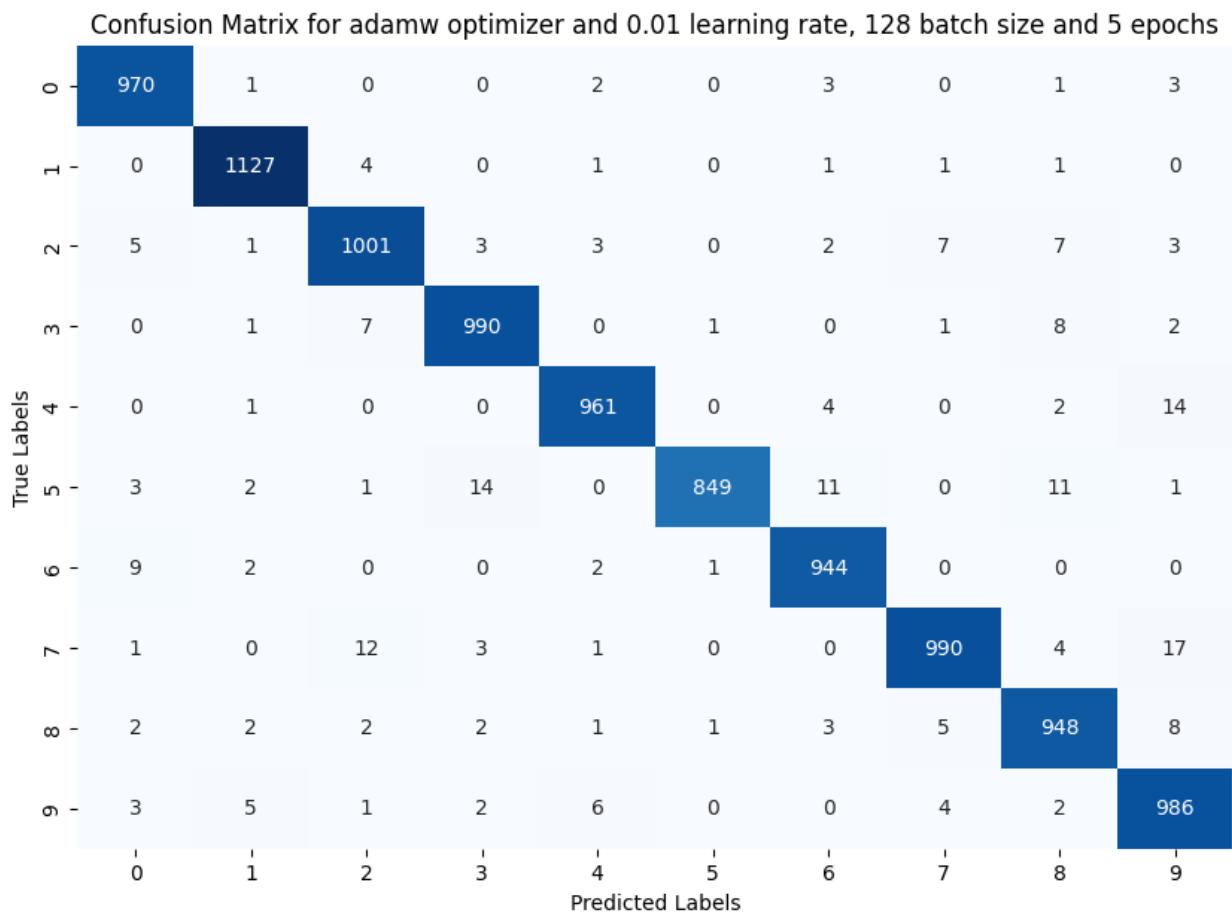
Epoch 4/5

422/422 - 3s - loss: 0.0905 - accuracy: 0.9727 - val_loss: 0.0710 -

```

val_accuracy: 0.9805 - 3s/epoch - 8ms/step
Epoch 5/5
422/422 - 3s - loss: 0.0885 - accuracy: 0.9727 - val_loss: 0.0706 -
val_accuracy: 0.9807 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 128
batch size and 5 epochs:
[[ 970   1   0   0   2   0   3   0   1   3]
 [ 0 1127   4   0   1   0   1   1   1   0]
 [ 5   1 1001   3   3   0   2   7   7   3]
 [ 0   1   7 990   0   1   0   1   8   2]
 [ 0   1   0   0 961   0   4   0   2 14]
 [ 3   2   1 14   0 849   11   0 11   1]
 [ 9   2   0   0   2   1 944   0   0   0]
 [ 1   0 12   3   1   0   0 990   4 17]
 [ 2   2   2   2   1   1   3   5 948   8]
 [ 3   5   1   2   6   0   0   4   2 986]]
Precision: 0.9768
Recall: 0.9766

```

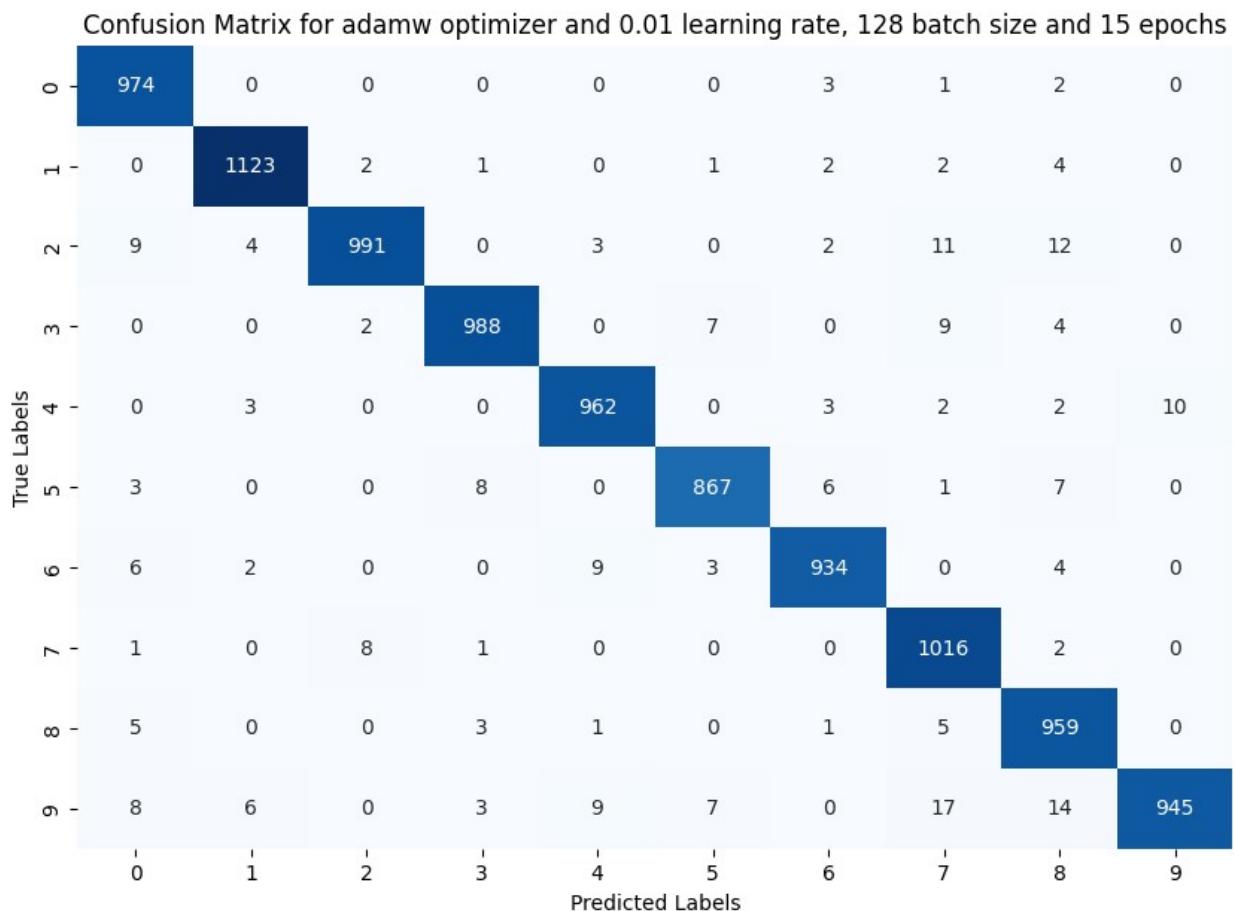


```
Training with adamw optimizer and the learning_rate is 0.01, 128 batch size and 15 epochs...
Epoch 1/15
422/422 - 4s - loss: 0.2175 - accuracy: 0.9415 - val_loss: 0.0774 -
val_accuracy: 0.9775 - 4s/epoch - 9ms/step
Epoch 2/15
422/422 - 3s - loss: 0.0971 - accuracy: 0.9705 - val_loss: 0.0965 -
val_accuracy: 0.9728 - 3s/epoch - 7ms/step
Epoch 3/15
422/422 - 3s - loss: 0.0904 - accuracy: 0.9734 - val_loss: 0.0810 -
val_accuracy: 0.9773 - 3s/epoch - 7ms/step
Epoch 4/15
422/422 - 3s - loss: 0.0901 - accuracy: 0.9725 - val_loss: 0.0759 -
val_accuracy: 0.9768 - 3s/epoch - 7ms/step
Epoch 5/15
422/422 - 3s - loss: 0.0886 - accuracy: 0.9739 - val_loss: 0.0898 -
val_accuracy: 0.9767 - 3s/epoch - 7ms/step
Epoch 6/15
422/422 - 3s - loss: 0.0878 - accuracy: 0.9736 - val_loss: 0.0763 -
val_accuracy: 0.9783 - 3s/epoch - 8ms/step
Epoch 7/15
422/422 - 3s - loss: 0.0874 - accuracy: 0.9730 - val_loss: 0.0861 -
val_accuracy: 0.9727 - 3s/epoch - 8ms/step
Epoch 8/15
422/422 - 3s - loss: 0.0857 - accuracy: 0.9737 - val_loss: 0.0668 -
val_accuracy: 0.9812 - 3s/epoch - 7ms/step
Epoch 9/15
422/422 - 3s - loss: 0.0833 - accuracy: 0.9744 - val_loss: 0.0767 -
val_accuracy: 0.9795 - 3s/epoch - 8ms/step
Epoch 10/15
422/422 - 3s - loss: 0.0856 - accuracy: 0.9743 - val_loss: 0.0733 -
val_accuracy: 0.9795 - 3s/epoch - 7ms/step
Epoch 11/15
422/422 - 3s - loss: 0.0817 - accuracy: 0.9755 - val_loss: 0.0827 -
val_accuracy: 0.9763 - 3s/epoch - 8ms/step
Epoch 12/15
422/422 - 3s - loss: 0.0863 - accuracy: 0.9731 - val_loss: 0.0672 -
val_accuracy: 0.9815 - 3s/epoch - 8ms/step
Epoch 13/15
422/422 - 3s - loss: 0.0822 - accuracy: 0.9748 - val_loss: 0.0738 -
val_accuracy: 0.9785 - 3s/epoch - 8ms/step
Epoch 14/15
422/422 - 3s - loss: 0.0819 - accuracy: 0.9751 - val_loss: 0.0599 -
val_accuracy: 0.9810 - 3s/epoch - 8ms/step
Epoch 15/15
422/422 - 3s - loss: 0.0841 - accuracy: 0.9745 - val_loss: 0.0637 -
val_accuracy: 0.9817 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 128 batch size and 15 epochs:
```

```
[[ 974  0  0  0  0  0  3  1  2  0]
 [ 0 1123  2  1  0  1  2  2  4  0]
 [ 9  4 991  0  3  0  2 11 12  0]
 [ 0  0  2 988  0  7  0  9  4  0]
 [ 0  3  0  0 962  0  3  2  2 10]
 [ 3  0  0  8  0 867  6  1  7  0]
 [ 6  2  0  0  9  3 934  0  4  0]
 [ 1  0  8  1  0  0  0 1016  2  0]
 [ 5  0  0  3  1  0  1  5 959  0]
 [ 8  6  0  3  9  7  0 17 14 945]]
```

Precision: 0.9762

Recall: 0.9759



Training with adamw optimizer and the learning_rate is 0.01, 128 batch size and 20 epochs...

Epoch 1/20

422/422 - 4s - loss: 0.2872 - accuracy: 0.9362 - val_loss: 0.0840 - val_accuracy: 0.9767 - 4s/epoch - 10ms/step

Epoch 2/20

422/422 - 3s - loss: 0.1053 - accuracy: 0.9691 - val_loss: 0.0921 - val_accuracy: 0.9735 - 3s/epoch - 8ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.0982 - accuracy: 0.9716 - val_loss: 0.0821 -
val_accuracy: 0.9777 - 3s/epoch - 8ms/step
Epoch 4/20
422/422 - 3s - loss: 0.0973 - accuracy: 0.9711 - val_loss: 0.0808 -
val_accuracy: 0.9773 - 3s/epoch - 8ms/step
Epoch 5/20
422/422 - 3s - loss: 0.0943 - accuracy: 0.9718 - val_loss: 0.0961 -
val_accuracy: 0.9720 - 3s/epoch - 8ms/step
Epoch 6/20
422/422 - 3s - loss: 0.0936 - accuracy: 0.9721 - val_loss: 0.0719 -
val_accuracy: 0.9815 - 3s/epoch - 8ms/step
Epoch 7/20
422/422 - 3s - loss: 0.0914 - accuracy: 0.9726 - val_loss: 0.0669 -
val_accuracy: 0.9825 - 3s/epoch - 8ms/step
Epoch 8/20
422/422 - 3s - loss: 0.0894 - accuracy: 0.9733 - val_loss: 0.0825 -
val_accuracy: 0.9792 - 3s/epoch - 8ms/step
Epoch 9/20
422/422 - 3s - loss: 0.0897 - accuracy: 0.9729 - val_loss: 0.0778 -
val_accuracy: 0.9765 - 3s/epoch - 8ms/step
Epoch 10/20
422/422 - 3s - loss: 0.0884 - accuracy: 0.9730 - val_loss: 0.0659 -
val_accuracy: 0.9812 - 3s/epoch - 8ms/step
Epoch 11/20
422/422 - 3s - loss: 0.0883 - accuracy: 0.9729 - val_loss: 0.1109 -
val_accuracy: 0.9652 - 3s/epoch - 8ms/step
Epoch 12/20
422/422 - 3s - loss: 0.0892 - accuracy: 0.9731 - val_loss: 0.0647 -
val_accuracy: 0.9822 - 3s/epoch - 8ms/step
Epoch 13/20
422/422 - 3s - loss: 0.0876 - accuracy: 0.9736 - val_loss: 0.0842 -
val_accuracy: 0.9750 - 3s/epoch - 8ms/step
Epoch 14/20
422/422 - 3s - loss: 0.0870 - accuracy: 0.9732 - val_loss: 0.0781 -
val_accuracy: 0.9788 - 3s/epoch - 8ms/step
Epoch 15/20
422/422 - 3s - loss: 0.0874 - accuracy: 0.9736 - val_loss: 0.0758 -
val_accuracy: 0.9780 - 3s/epoch - 8ms/step
Epoch 16/20
422/422 - 3s - loss: 0.0870 - accuracy: 0.9737 - val_loss: 0.0660 -
val_accuracy: 0.9825 - 3s/epoch - 8ms/step
Epoch 17/20
422/422 - 3s - loss: 0.0842 - accuracy: 0.9747 - val_loss: 0.0778 -
val_accuracy: 0.9765 - 3s/epoch - 8ms/step
Epoch 18/20
422/422 - 3s - loss: 0.0873 - accuracy: 0.9733 - val_loss: 0.0804 -
val_accuracy: 0.9760 - 3s/epoch - 8ms/step
Epoch 19/20
```

```
422/422 - 3s - loss: 0.0877 - accuracy: 0.9729 - val_loss: 0.0799 -  
val_accuracy: 0.9777 - 3s/epoch - 8ms/step
```

```
Epoch 20/20
```

```
422/422 - 3s - loss: 0.0859 - accuracy: 0.9735 - val_loss: 0.0602 -  
val_accuracy: 0.9828 - 3s/epoch - 8ms/step
```

```
313/313 [=====] - 1s 2ms/step
```

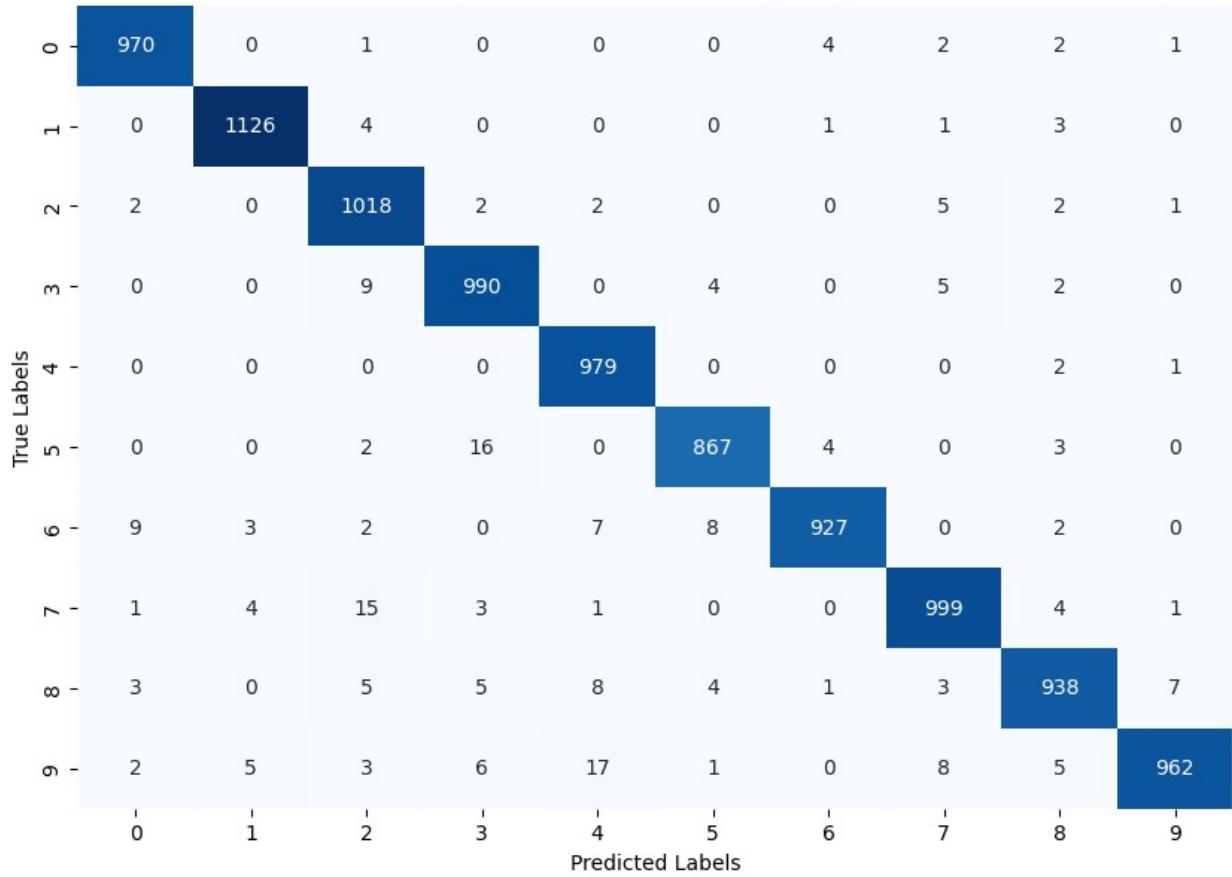
```
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 128  
batch size and 20 epochs:
```

```
[[ 970   0   1   0   0   0   4   2   2   1]  
[  0 1126   4   0   0   0   1   1   3   0]  
[  2   0 1018   2   2   0   0   5   2   1]  
[  0   0   9 990   0   4   0   5   2   0]  
[  0   0   0   0 979   0   0   0   2   1]  
[  0   0   2 16   0 867   4   0   3   0]  
[  9   3   2   0   7   8 927   0   2   0]  
[  1   4 15   3   1   0   0 999   4   1]  
[  3   0   5   5   8   4   1   3 938   7]  
[  2   5   3   6 17   1   0   8   5 962]]
```

```
Precision: 0.9777
```

```
Recall: 0.9776
```

Confusion Matrix for adamw optimizer and 0.01 learning rate, 128 batch size and 20 epochs



```
Training with adamw optimizer and the learning_rate is 0.01, 256 batch size and 5 epochs...
Epoch 1/5
211/211 - 4s - loss: 0.3072 - accuracy: 0.9222 - val_loss: 0.0801 -
val_accuracy: 0.9773 - 4s/epoch - 20ms/step
Epoch 2/5
211/211 - 3s - loss: 0.0778 - accuracy: 0.9775 - val_loss: 0.0690 -
val_accuracy: 0.9807 - 3s/epoch - 12ms/step
Epoch 3/5
211/211 - 3s - loss: 0.0712 - accuracy: 0.9787 - val_loss: 0.0700 -
val_accuracy: 0.9798 - 3s/epoch - 12ms/step
Epoch 4/5
211/211 - 3s - loss: 0.0676 - accuracy: 0.9799 - val_loss: 0.0618 -
val_accuracy: 0.9842 - 3s/epoch - 13ms/step
Epoch 5/5
211/211 - 3s - loss: 0.0645 - accuracy: 0.9808 - val_loss: 0.0641 -
val_accuracy: 0.9812 - 3s/epoch - 12ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 256 batch size and 5 epochs:
[[ 967    1    1    1    1    0    7    0    2    0]
 [  0 1130    1    1    1    0    1    0    1    0]
 [  0    1 1012    2    4    0    2    3    8    0]
 [  0    0    0 992    0   12    0    1    3    2]
 [  0    0    0    0 980    0    1    0    1    0]
 [  2    0    0    6    0 877    5    0    2    0]
 [  3    2    0    0    2    5 943    0    3    0]
 [  1    3   14    6    3    1    0 993    5    2]
 [  3    0    2    4    6    2    5    1 949    2]
 [  2    4    2    8   43    7    0    4   13 926]]
```

Precision: 0.9773
Recall: 0.9769

Confusion Matrix for adamw optimizer and 0.01 learning rate, 256 batch size and 5 epochs

	0	1	1	1	1	0	7	0	2	0
True Labels	967	1	1	1	1	0	7	0	2	0
0	967	1	1	1	1	0	7	0	2	0
1	0	1130	1	1	1	0	1	0	1	0
2	0	1	1012	2	4	0	2	3	8	0
3	0	0	0	992	0	12	0	1	3	2
4	0	0	0	0	980	0	1	0	1	0
5	2	0	0	6	0	877	5	0	2	0
6	3	2	0	0	2	5	943	0	3	0
7	1	3	14	6	3	1	0	993	5	2
8	3	0	2	4	6	2	5	1	949	2
9	2	4	2	8	43	7	0	4	13	926
0	1	2	3	4	5	6	7	8	9	9
Predicted Labels										

```
Training with adamw optimizer and the learning_rate is 0.01, 256 batch size and 15 epochs...
```

```
Epoch 1/15
```

```
211/211 - 3s - loss: 0.3638 - accuracy: 0.9209 - val_loss: 0.0750 - val_accuracy: 0.9803 - 3s/epoch - 16ms/step
```

```
Epoch 2/15
```

```
211/211 - 3s - loss: 0.0869 - accuracy: 0.9749 - val_loss: 0.0905 - val_accuracy: 0.9740 - 3s/epoch - 13ms/step
```

```
Epoch 3/15
```

```
211/211 - 3s - loss: 0.0770 - accuracy: 0.9773 - val_loss: 0.0704 - val_accuracy: 0.9798 - 3s/epoch - 12ms/step
```

```
Epoch 4/15
```

```
211/211 - 3s - loss: 0.0742 - accuracy: 0.9785 - val_loss: 0.0616 - val_accuracy: 0.9823 - 3s/epoch - 12ms/step
```

```
Epoch 5/15
```

```
211/211 - 3s - loss: 0.0699 - accuracy: 0.9795 - val_loss: 0.0684 - val_accuracy: 0.9808 - 3s/epoch - 12ms/step
```

```
Epoch 6/15
```

```
211/211 - 3s - loss: 0.0673 - accuracy: 0.9811 - val_loss: 0.0668 - val_accuracy: 0.9812 - 3s/epoch - 12ms/step
```

```
Epoch 7/15
```

```
211/211 - 2s - loss: 0.0682 - accuracy: 0.9795 - val_loss: 0.0664 -  
val_accuracy: 0.9822 - 2s/epoch - 12ms/step  
Epoch 8/15  
211/211 - 3s - loss: 0.0666 - accuracy: 0.9809 - val_loss: 0.0544 -  
val_accuracy: 0.9847 - 3s/epoch - 12ms/step  
Epoch 9/15  
211/211 - 3s - loss: 0.0651 - accuracy: 0.9809 - val_loss: 0.0674 -  
val_accuracy: 0.9780 - 3s/epoch - 12ms/step  
Epoch 10/15  
211/211 - 3s - loss: 0.0609 - accuracy: 0.9815 - val_loss: 0.0675 -  
val_accuracy: 0.9818 - 3s/epoch - 12ms/step  
Epoch 11/15  
211/211 - 2s - loss: 0.0646 - accuracy: 0.9802 - val_loss: 0.0722 -  
val_accuracy: 0.9797 - 2s/epoch - 12ms/step  
Epoch 12/15  
211/211 - 3s - loss: 0.0618 - accuracy: 0.9813 - val_loss: 0.0671 -  
val_accuracy: 0.9825 - 3s/epoch - 12ms/step  
Epoch 13/15  
211/211 - 2s - loss: 0.0600 - accuracy: 0.9817 - val_loss: 0.0583 -  
val_accuracy: 0.9847 - 2s/epoch - 12ms/step  
Epoch 14/15  
211/211 - 3s - loss: 0.0615 - accuracy: 0.9816 - val_loss: 0.0653 -  
val_accuracy: 0.9798 - 3s/epoch - 12ms/step  
Epoch 15/15  
211/211 - 3s - loss: 0.0579 - accuracy: 0.9826 - val_loss: 0.0583 -  
val_accuracy: 0.9848 - 3s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 256  
batch size and 15 epochs:  
[[ 972  0  1  0  0  0  2  3  2  0]  
[  0 1124  5  0  0  0  1  3  2  0]  
[  1  0 1016  0  0  0  0  10  4  1]  
[  0  0  3 985  0  6  0  10  5  1]  
[  0  0  2  0 975  0  0  1  0  4]  
[  1  0  0  4  0 882  3  0  2  0]  
[  9  2  1  0  7  5 930  0  4  0]  
[  1  0  6  0  0  1  0 1018  1  1]  
[  4  0  3  1  1  4  0  5 950  6]  
[  2  2  1  2 11  5  0  7  3 976]]  
Precision: 0.9829  
Recall: 0.9828
```

Confusion Matrix for adamw optimizer and 0.01 learning rate, 256 batch size and 15 epochs

	0	1	2	3	4	5	6	7	8	9	
0	972	0	1	0	0	0	2	3	2	0	
1	0	1124	5	0	0	0	1	3	2	0	
2	1	0	1016	0	0	0	0	10	4	1	
3	0	0	3	985	0	6	0	10	5	1	
4	0	0	2	0	975	0	0	1	0	4	
5	1	0	0	4	0	882	3	0	2	0	
6	9	2	1	0	7	5	930	0	4	0	
7	1	0	6	0	0	1	0	1018	1	1	
8	4	0	3	1	1	4	0	5	950	6	
9	2	2	1	2	11	5	0	7	3	976	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```
Training with adamw optimizer and the learning_rate is 0.01, 256 batch size and 20 epochs...
```

```
Epoch 1/20
```

```
211/211 - 3s - loss: 0.3846 - accuracy: 0.9207 - val_loss: 0.0871 - val_accuracy: 0.9762 - 3s/epoch - 15ms/step
```

```
Epoch 2/20
```

```
211/211 - 3s - loss: 0.0900 - accuracy: 0.9741 - val_loss: 0.0685 - val_accuracy: 0.9828 - 3s/epoch - 12ms/step
```

```
Epoch 3/20
```

```
211/211 - 3s - loss: 0.0802 - accuracy: 0.9764 - val_loss: 0.0638 - val_accuracy: 0.9830 - 3s/epoch - 12ms/step
```

```
Epoch 4/20
```

```
211/211 - 2s - loss: 0.0772 - accuracy: 0.9773 - val_loss: 0.0641 - val_accuracy: 0.9827 - 2s/epoch - 12ms/step
```

```
Epoch 5/20
```

```
211/211 - 3s - loss: 0.0738 - accuracy: 0.9779 - val_loss: 0.0771 - val_accuracy: 0.9790 - 3s/epoch - 12ms/step
```

```
Epoch 6/20
```

```
211/211 - 3s - loss: 0.0720 - accuracy: 0.9789 - val_loss: 0.0689 - val_accuracy: 0.9807 - 3s/epoch - 12ms/step
```

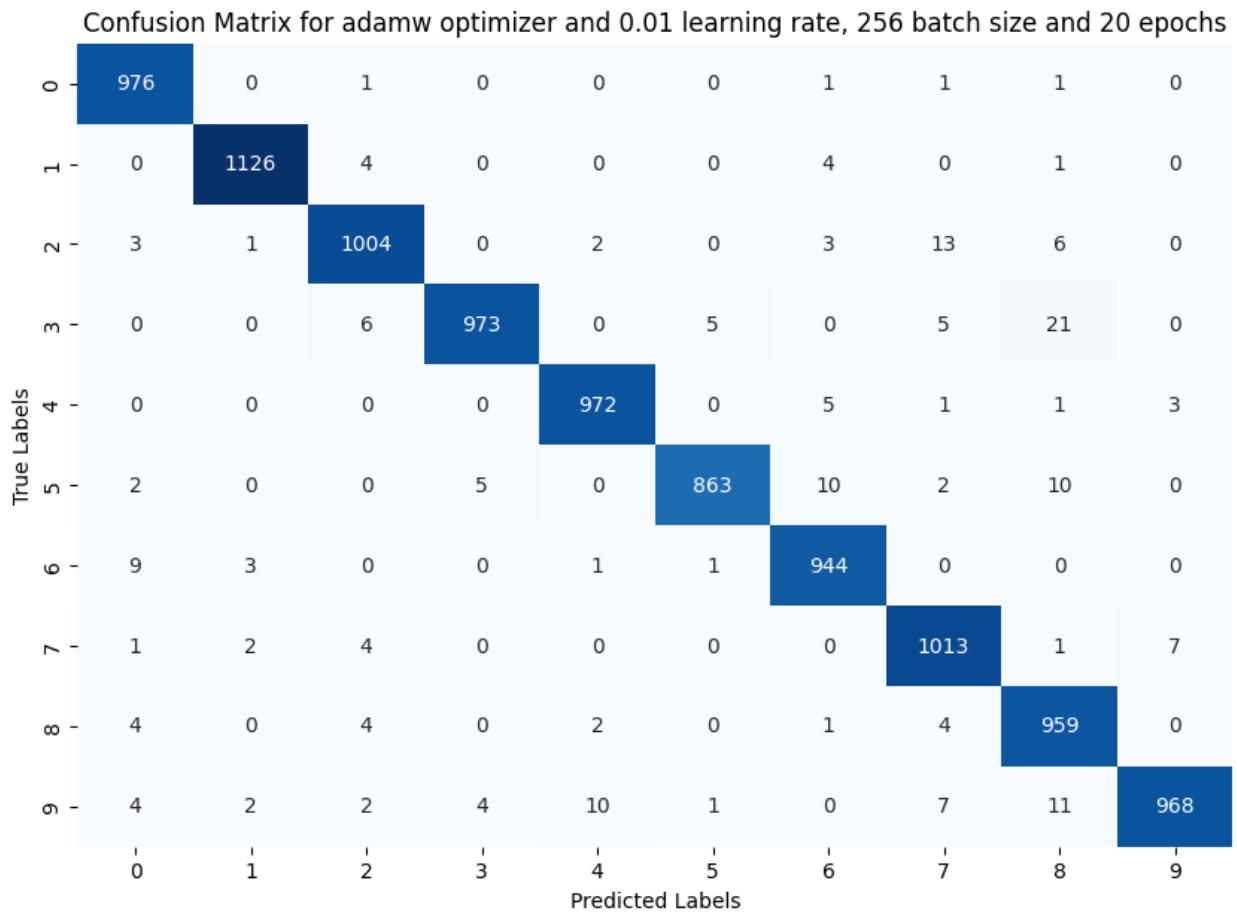
```
Epoch 7/20
```

```
211/211 - 3s - loss: 0.0726 - accuracy: 0.9786 - val_loss: 0.0695 -  
val_accuracy: 0.9802 - 3s/epoch - 12ms/step  
Epoch 8/20  
211/211 - 3s - loss: 0.0682 - accuracy: 0.9799 - val_loss: 0.0582 -  
val_accuracy: 0.9827 - 3s/epoch - 12ms/step  
Epoch 9/20  
211/211 - 2s - loss: 0.0667 - accuracy: 0.9800 - val_loss: 0.0689 -  
val_accuracy: 0.9828 - 2s/epoch - 12ms/step  
Epoch 10/20  
211/211 - 2s - loss: 0.0657 - accuracy: 0.9808 - val_loss: 0.0557 -  
val_accuracy: 0.9858 - 2s/epoch - 12ms/step  
Epoch 11/20  
211/211 - 2s - loss: 0.0654 - accuracy: 0.9803 - val_loss: 0.0545 -  
val_accuracy: 0.9855 - 2s/epoch - 12ms/step  
Epoch 12/20  
211/211 - 3s - loss: 0.0662 - accuracy: 0.9806 - val_loss: 0.0653 -  
val_accuracy: 0.9827 - 3s/epoch - 12ms/step  
Epoch 13/20  
211/211 - 3s - loss: 0.0638 - accuracy: 0.9807 - val_loss: 0.0656 -  
val_accuracy: 0.9815 - 3s/epoch - 12ms/step  
Epoch 14/20  
211/211 - 3s - loss: 0.0644 - accuracy: 0.9811 - val_loss: 0.0757 -  
val_accuracy: 0.9782 - 3s/epoch - 12ms/step  
Epoch 15/20  
211/211 - 3s - loss: 0.0634 - accuracy: 0.9810 - val_loss: 0.0615 -  
val_accuracy: 0.9820 - 3s/epoch - 12ms/step  
Epoch 16/20  
211/211 - 3s - loss: 0.0618 - accuracy: 0.9817 - val_loss: 0.0717 -  
val_accuracy: 0.9795 - 3s/epoch - 12ms/step  
Epoch 17/20  
211/211 - 3s - loss: 0.0604 - accuracy: 0.9819 - val_loss: 0.0553 -  
val_accuracy: 0.9833 - 3s/epoch - 13ms/step  
Epoch 18/20  
211/211 - 3s - loss: 0.0595 - accuracy: 0.9825 - val_loss: 0.0498 -  
val_accuracy: 0.9863 - 3s/epoch - 13ms/step  
Epoch 19/20  
211/211 - 3s - loss: 0.0614 - accuracy: 0.9814 - val_loss: 0.0640 -  
val_accuracy: 0.9808 - 3s/epoch - 12ms/step  
Epoch 20/20  
211/211 - 3s - loss: 0.0580 - accuracy: 0.9830 - val_loss: 0.0600 -  
val_accuracy: 0.9830 - 3s/epoch - 12ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 256  
batch size and 20 epochs:  
[[ 976   0   1   0   0   0   1   1   1   0]  
 [  0 1126   4   0   0   0   4   0   1   0]  
 [  3   1 1004   0   2   0   3  13   6   0]  
 [  0   0   6  973   0   5   0   5  21   0]  
 [  0   0   0   0  972   0   5   1   1   3]]
```

```
[ 2  0  0  5  0 863 10  2 10  0
[ 9  3  0  0  1  1 944  0  0  0
[ 1  2  4  0  0  0   0 1013  1  7
[ 4  0  4  0  2  0  1   4 959  0
[ 4  2  2  4 10  1  0   7 11 968]]
```

Precision: 0.9800

Recall: 0.9798



Training with adamw optimizer and the learning_rate is 0.01, 16 batch size and 5 epochs...

Epoch 1/5

3375/3375 - 15s - loss: 0.2597 - accuracy: 0.9239 - val_loss: 0.1526 - val_accuracy: 0.9575 - 15s/epoch - 5ms/step

Epoch 2/5

3375/3375 - 14s - loss: 0.2337 - accuracy: 0.9290 - val_loss: 0.2069 - val_accuracy: 0.9408 - 14s/epoch - 4ms/step

Epoch 3/5

3375/3375 - 15s - loss: 0.2388 - accuracy: 0.9267 - val_loss: 0.2450 - val_accuracy: 0.9233 - 15s/epoch - 4ms/step

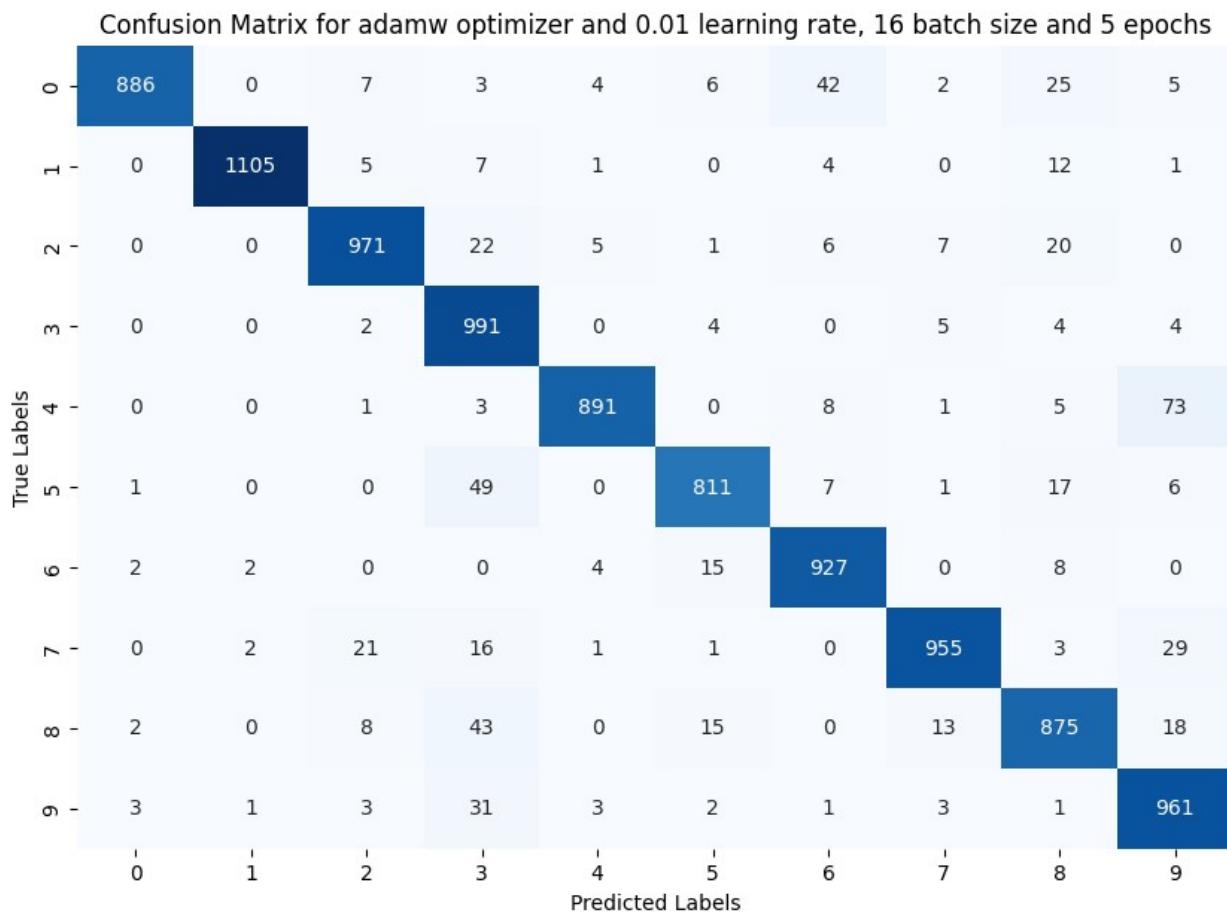
Epoch 4/5

3375/3375 - 15s - loss: 0.2406 - accuracy: 0.9276 - val_loss: 0.1783 -

```

val_accuracy: 0.9472 - 15s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 15s - loss: 0.2362 - accuracy: 0.9282 - val_loss: 0.1788 -
val_accuracy: 0.9457 - 15s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.01, 16
batch size and 5 epochs:
[[ 886   0   7   3   4   6   42   2   25   5]
 [ 0 1105   5   7   1   0   4   0   12   1]
 [ 0   0 971  22   5   1   6   7  20   0]
 [ 0   0   2 991   0   4   0   5   4   4]
 [ 0   0   1   3 891   0   8   1   5 73]
 [ 1   0   0  49   0 811   7   1  17   6]
 [ 2   2   0   0   4 15 927   0   8   0]
 [ 0   2   21  16   1   1   0 955   3 29]
 [ 2   0   8  43   0 15   0 13 875 18]
 [ 3   1   3 31   3   2   1   3   1 961]]
Precision: 0.9403
Recall: 0.9373

```



Training with adamw optimizer and the learning_rate is 0.01, 16 batch size and 15 epochs...

Epoch 1/15
3375/3375 - 15s - loss: 0.2602 - accuracy: 0.9236 - val_loss: 0.1627 - val_accuracy: 0.9527 - 15s/epoch - 5ms/step

Epoch 2/15
3375/3375 - 14s - loss: 0.2321 - accuracy: 0.9291 - val_loss: 0.2954 - val_accuracy: 0.9052 - 14s/epoch - 4ms/step

Epoch 3/15
3375/3375 - 15s - loss: 0.2331 - accuracy: 0.9297 - val_loss: 0.1543 - val_accuracy: 0.9527 - 15s/epoch - 4ms/step

Epoch 4/15
3375/3375 - 15s - loss: 0.2352 - accuracy: 0.9284 - val_loss: 0.1893 - val_accuracy: 0.9405 - 15s/epoch - 4ms/step

Epoch 5/15
3375/3375 - 15s - loss: 0.2359 - accuracy: 0.9284 - val_loss: 0.2838 - val_accuracy: 0.9113 - 15s/epoch - 4ms/step

Epoch 6/15
3375/3375 - 15s - loss: 0.2342 - accuracy: 0.9301 - val_loss: 0.1949 - val_accuracy: 0.9513 - 15s/epoch - 4ms/step

Epoch 7/15
3375/3375 - 15s - loss: 0.2358 - accuracy: 0.9284 - val_loss: 0.2101 - val_accuracy: 0.9372 - 15s/epoch - 4ms/step

Epoch 8/15
3375/3375 - 14s - loss: 0.2381 - accuracy: 0.9274 - val_loss: 0.2022 - val_accuracy: 0.9408 - 14s/epoch - 4ms/step

Epoch 9/15
3375/3375 - 15s - loss: 0.2352 - accuracy: 0.9282 - val_loss: 0.1673 - val_accuracy: 0.9495 - 15s/epoch - 4ms/step

Epoch 10/15
3375/3375 - 15s - loss: 0.2393 - accuracy: 0.9276 - val_loss: 0.2062 - val_accuracy: 0.9383 - 15s/epoch - 4ms/step

Epoch 11/15
3375/3375 - 15s - loss: 0.2431 - accuracy: 0.9266 - val_loss: 0.2704 - val_accuracy: 0.9183 - 15s/epoch - 5ms/step

Epoch 12/15
3375/3375 - 15s - loss: 0.2403 - accuracy: 0.9272 - val_loss: 0.1916 - val_accuracy: 0.9455 - 15s/epoch - 4ms/step

Epoch 13/15
3375/3375 - 14s - loss: 0.2417 - accuracy: 0.9259 - val_loss: 0.1688 - val_accuracy: 0.9488 - 14s/epoch - 4ms/step

Epoch 14/15
3375/3375 - 15s - loss: 0.2375 - accuracy: 0.9276 - val_loss: 0.2065 - val_accuracy: 0.9393 - 15s/epoch - 4ms/step

Epoch 15/15
3375/3375 - 15s - loss: 0.2365 - accuracy: 0.9271 - val_loss: 0.1576 - val_accuracy: 0.9537 - 15s/epoch - 4ms/step

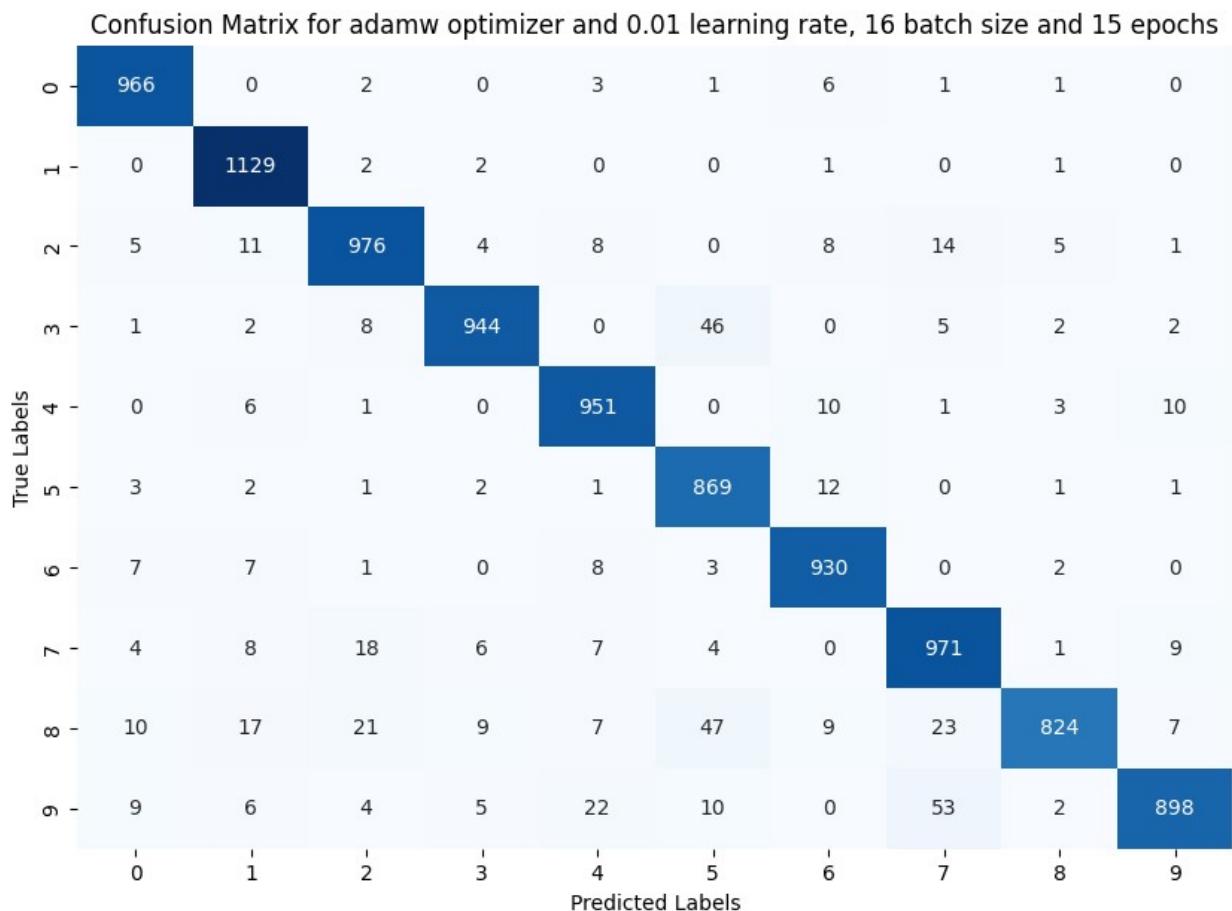
313/313 [=====] - 1s 2ms/step

Confusion Matrix adamw optimizer and the learning_rate is 0.01, 16 batch size and 15 epochs:

```
[[ 966  0   2   0   3   1   6   1   1   0]
 [ 0 1129  2   2   0   0   1   0   1   0]
 [ 5 11  976  4   8   0   8   14  5   1]
 [ 1 2   8 944  0   46  0   5   2   2]
 [ 0 6   1  0 951  0   10  1   3   10]
 [ 3 2   1   2  1 869  12  0   1   1]
 [ 7 7   1   0   8  3 930  0   2   0]
 [ 4 8   18  6   7  4  0 971  1   9]
 [ 10 17  21  9   7  47  9  23 824  7]
 [ 9 6   4   5  22  10  0 53  2 898]]
```

Precision: 0.9471

Recall: 0.9458



Training with adamw optimizer and the learning_rate is 0.01, 16 batch size and 20 epochs...

Epoch 1/20

3375/3375 - 15s - loss: 0.2621 - accuracy: 0.9226 - val_loss: 0.2231 - val_accuracy: 0.9325 - 15s/epoch - 5ms/step

Epoch 2/20

3375/3375 - 15s - loss: 0.2399 - accuracy: 0.9271 - val_loss: 0.2274 - val_accuracy: 0.9325 - 15s/epoch - 4ms/step

```
Epoch 3/20
3375/3375 - 15s - loss: 0.2353 - accuracy: 0.9288 - val_loss: 0.1656 -
val_accuracy: 0.9548 - 15s/epoch - 4ms/step
Epoch 4/20
3375/3375 - 15s - loss: 0.2409 - accuracy: 0.9272 - val_loss: 0.1645 -
val_accuracy: 0.9492 - 15s/epoch - 4ms/step
Epoch 5/20
3375/3375 - 15s - loss: 0.2389 - accuracy: 0.9283 - val_loss: 0.1465 -
val_accuracy: 0.9615 - 15s/epoch - 4ms/step
Epoch 6/20
3375/3375 - 15s - loss: 0.2347 - accuracy: 0.9284 - val_loss: 0.1434 -
val_accuracy: 0.9592 - 15s/epoch - 4ms/step
Epoch 7/20
3375/3375 - 15s - loss: 0.2402 - accuracy: 0.9275 - val_loss: 0.1792 -
val_accuracy: 0.9487 - 15s/epoch - 4ms/step
Epoch 8/20
3375/3375 - 15s - loss: 0.2408 - accuracy: 0.9267 - val_loss: 0.1718 -
val_accuracy: 0.9505 - 15s/epoch - 4ms/step
Epoch 9/20
3375/3375 - 14s - loss: 0.2375 - accuracy: 0.9280 - val_loss: 0.1669 -
val_accuracy: 0.9533 - 14s/epoch - 4ms/step
Epoch 10/20
3375/3375 - 14s - loss: 0.2397 - accuracy: 0.9264 - val_loss: 0.2166 -
val_accuracy: 0.9360 - 14s/epoch - 4ms/step
Epoch 11/20
3375/3375 - 14s - loss: 0.2377 - accuracy: 0.9287 - val_loss: 0.1612 -
val_accuracy: 0.9510 - 14s/epoch - 4ms/step
Epoch 12/20
3375/3375 - 14s - loss: 0.2430 - accuracy: 0.9261 - val_loss: 0.2192 -
val_accuracy: 0.9288 - 14s/epoch - 4ms/step
Epoch 13/20
3375/3375 - 14s - loss: 0.2367 - accuracy: 0.9280 - val_loss: 0.1503 -
val_accuracy: 0.9560 - 14s/epoch - 4ms/step
Epoch 14/20
3375/3375 - 14s - loss: 0.2406 - accuracy: 0.9278 - val_loss: 0.1624 -
val_accuracy: 0.9553 - 14s/epoch - 4ms/step
Epoch 15/20
3375/3375 - 13s - loss: 0.2376 - accuracy: 0.9279 - val_loss: 0.1797 -
val_accuracy: 0.9455 - 13s/epoch - 4ms/step
Epoch 16/20
3375/3375 - 14s - loss: 0.2415 - accuracy: 0.9269 - val_loss: 0.1836 -
val_accuracy: 0.9530 - 14s/epoch - 4ms/step
Epoch 17/20
3375/3375 - 14s - loss: 0.2403 - accuracy: 0.9276 - val_loss: 0.1668 -
val_accuracy: 0.9517 - 14s/epoch - 4ms/step
Epoch 18/20
3375/3375 - 15s - loss: 0.2394 - accuracy: 0.9277 - val_loss: 0.1772 -
val_accuracy: 0.9430 - 15s/epoch - 5ms/step
Epoch 19/20
```

3375/3375 - 15s - loss: 0.2443 - accuracy: 0.9263 - val_loss: 0.2017 - val_accuracy: 0.9402 - 15s/epoch - 4ms/step

Epoch 20/20

3375/3375 - 15s - loss: 0.2390 - accuracy: 0.9278 - val_loss: 0.2737 - val_accuracy: 0.9148 - 15s/epoch - 4ms/step

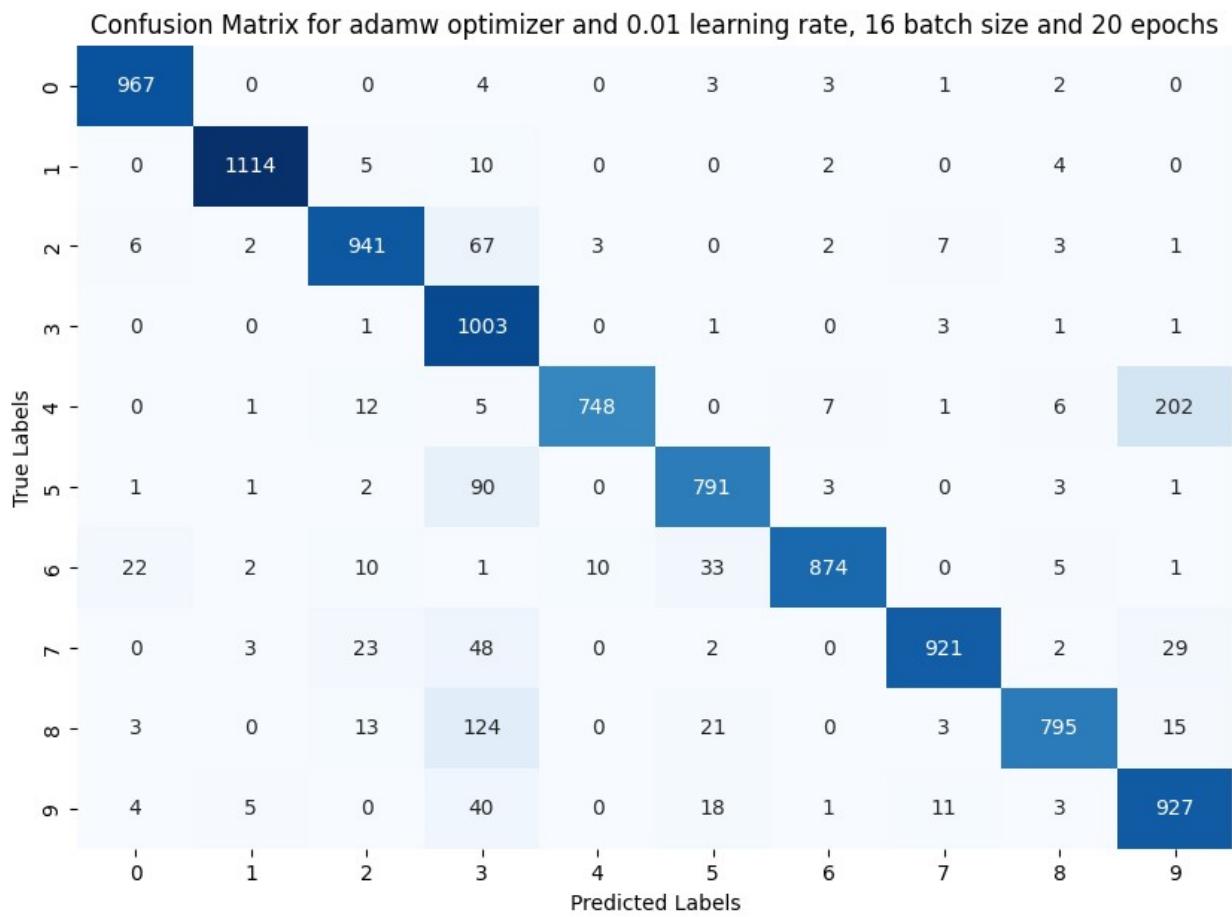
313/313 [=====] - 1s 2ms/step

Confusion Matrix adamw optimizer and the learning_rate is 0.01, 16 batch size and 20 epochs:

```
[[ 967   0   0   4   0   3   3   1   2   0]
 [ 0 1114   5  10   0   0   2   0   4   0]
 [ 6   2 941   67   3   0   2   7   3   1]
 [ 0   0   1 1003   0   1   0   3   1   1]
 [ 0   1   12   5 748   0   7   1   6 202]
 [ 1   1   2   90   0 791   3   0   3   1]
 [ 22   2   10   1 10   33 874   0   5   1]
 [ 0   3   23   48   0   2   0 921   2   29]
 [ 3   0   13 124   0   21   0   3 795   15]
 [ 4   5   0   40   0   18   1 11   3 927]]
```

Precision: 0.9208

Recall: 0.9081



```
Training with adamw optimizer and the learning_rate is 0.001, 64 batch size and 5 epochs...
Epoch 1/5
844/844 - 6s - loss: 0.3170 - accuracy: 0.9137 - val_loss: 0.2286 -
val_accuracy: 0.9460 - 6s/epoch - 7ms/step
Epoch 2/5
844/844 - 5s - loss: 0.2871 - accuracy: 0.9231 - val_loss: 0.2307 -
val_accuracy: 0.9382 - 5s/epoch - 6ms/step
Epoch 3/5
844/844 - 5s - loss: 0.2807 - accuracy: 0.9252 - val_loss: 0.2325 -
val_accuracy: 0.9425 - 5s/epoch - 6ms/step
Epoch 4/5
844/844 - 5s - loss: 0.2722 - accuracy: 0.9271 - val_loss: 0.2214 -
val_accuracy: 0.9442 - 5s/epoch - 6ms/step
Epoch 5/5
844/844 - 5s - loss: 0.2691 - accuracy: 0.9268 - val_loss: 0.2118 -
val_accuracy: 0.9473 - 5s/epoch - 6ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 64 batch size and 5 epochs:
[[ 972    0    0    0    0    1    2    1    3    1]
 [  0 1124    3    1    0    0    3    0    4    0]
 [ 26    4  946    9    3    0    7   13   21    3]
 [  6    1   12  931    0   12    0    9   29   10]
 [  3    3    6    0  891    0   10    2    2   65]
 [ 22    1    2   29    3  804    8    1   14    8]
 [ 33    5    3    1    4   15  889    1    7    0]
 [  4   11   17    3    5    0    0  948    2   38]
 [ 22    5    6   13   10   11    5    8  880   14]
 [ 13    6    1   13    9    0    0    9    3  955]]
```

Precision: 0.9355
Recall: 0.9340

Confusion Matrix for adamw optimizer and 0.001 learning rate, 64 batch size and 5 epochs

	0	1	2	3	4	5	6	7	8	9	10
0	972	0	0	0	0	1	2	1	3	1	
1	0	1124	3	1	0	0	3	0	4	0	
2	26	4	946	9	3	0	7	13	21	3	
3	6	1	12	931	0	12	0	9	29	10	
4	3	3	6	0	891	0	10	2	2	65	
5	22	1	2	29	3	804	8	1	14	8	
6	33	5	3	1	4	15	889	1	7	0	
7	4	11	17	3	5	0	0	948	2	38	
8	22	5	6	13	10	11	5	8	880	14	
9	13	6	1	13	9	0	0	9	3	955	
10	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	
True Labels											Predicted Labels

Training with adamw optimizer and the learning_rate is 0.001, 64 batch size and 15 epochs...

Epoch 1/15

844/844 - 6s - loss: 0.3099 - accuracy: 0.9187 - val_loss: 0.2565 - val_accuracy: 0.9365 - 6s/epoch - 7ms/step

Epoch 2/15

844/844 - 5s - loss: 0.2966 - accuracy: 0.9204 - val_loss: 0.2467 - val_accuracy: 0.9308 - 5s/epoch - 6ms/step

Epoch 3/15

844/844 - 5s - loss: 0.2944 - accuracy: 0.9205 - val_loss: 0.2484 - val_accuracy: 0.9395 - 5s/epoch - 6ms/step

Epoch 4/15

844/844 - 5s - loss: 0.2833 - accuracy: 0.9231 - val_loss: 0.2251 - val_accuracy: 0.9502 - 5s/epoch - 6ms/step

Epoch 5/15

844/844 - 5s - loss: 0.2797 - accuracy: 0.9255 - val_loss: 0.2268 - val_accuracy: 0.9452 - 5s/epoch - 6ms/step

Epoch 6/15

844/844 - 5s - loss: 0.2768 - accuracy: 0.9247 - val_loss: 0.2170 - val_accuracy: 0.9473 - 5s/epoch - 6ms/step

Epoch 7/15

```
844/844 - 5s - loss: 0.2752 - accuracy: 0.9253 - val_loss: 0.2227 -  
val_accuracy: 0.9450 - 5s/epoch - 6ms/step  
Epoch 8/15  
844/844 - 5s - loss: 0.2720 - accuracy: 0.9265 - val_loss: 0.2261 -  
val_accuracy: 0.9423 - 5s/epoch - 6ms/step  
Epoch 9/15  
844/844 - 5s - loss: 0.2749 - accuracy: 0.9248 - val_loss: 0.2217 -  
val_accuracy: 0.9448 - 5s/epoch - 6ms/step  
Epoch 10/15  
844/844 - 5s - loss: 0.2694 - accuracy: 0.9270 - val_loss: 0.2200 -  
val_accuracy: 0.9397 - 5s/epoch - 5ms/step  
Epoch 11/15  
844/844 - 5s - loss: 0.2717 - accuracy: 0.9259 - val_loss: 0.2295 -  
val_accuracy: 0.9420 - 5s/epoch - 6ms/step  
Epoch 12/15  
844/844 - 5s - loss: 0.2713 - accuracy: 0.9261 - val_loss: 0.2209 -  
val_accuracy: 0.9463 - 5s/epoch - 6ms/step  
Epoch 13/15  
844/844 - 5s - loss: 0.2702 - accuracy: 0.9265 - val_loss: 0.2368 -  
val_accuracy: 0.9358 - 5s/epoch - 6ms/step  
Epoch 14/15  
844/844 - 5s - loss: 0.2704 - accuracy: 0.9260 - val_loss: 0.2330 -  
val_accuracy: 0.9397 - 5s/epoch - 6ms/step  
Epoch 15/15  
844/844 - 5s - loss: 0.2725 - accuracy: 0.9256 - val_loss: 0.2552 -  
val_accuracy: 0.9325 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 64  
batch size and 15 epochs:  
[[ 971 0 0 2 0 1 0 1 4 1]  
[ 0 1118 0 5 0 0 3 1 7 1]  
[ 17 6 897 45 10 0 1 28 24 4]  
[ 1 1 2 966 0 0 0 18 12 10]  
[ 3 0 3 0 800 0 3 8 6 159]  
[ 12 2 0 84 3 739 3 5 15 29]  
[ 43 3 8 3 15 28 843 1 14 0]  
[ 1 4 10 6 2 0 0 966 1 38]  
[ 7 2 2 48 3 5 3 21 842 41]  
[ 10 5 0 18 3 0 0 10 0 963]]  
Precision: 0.9179  
Recall: 0.9105
```

Confusion Matrix for adamw optimizer and 0.001 learning rate, 64 batch size and 15 epochs

	0	1	2	3	4	5	6	7	8	9	
0	971	0	0	2	0	1	0	1	4	1	
1	0	1118	0	5	0	0	3	1	7	1	
2	17	6	897	45	10	0	1	28	24	4	
3	1	1	2	966	0	0	0	18	12	10	
4	3	0	3	0	800	0	3	8	6	159	
5	12	2	0	84	3	739	3	5	15	29	
6	43	3	8	3	15	28	843	1	14	0	
7	1	4	10	6	2	0	0	966	1	38	
8	7	2	2	48	3	5	3	21	842	41	
9	10	5	0	18	3	0	0	10	0	963	
	0	1	2	3	4	5	6	7	8	9	
	0	1	2	3	4	5	6	7	8	9	Predicted Labels
True Labels	0	1	2	3	4	5	6	7	8	9	

```
Training with adamw optimizer and the learning_rate is 0.001, 64 batch size and 20 epochs...
```

```
Epoch 1/20
```

```
844/844 - 6s - loss: 0.3225 - accuracy: 0.9124 - val_loss: 0.2498 - val_accuracy: 0.9382 - 6s/epoch - 7ms/step
```

```
Epoch 2/20
```

```
844/844 - 5s - loss: 0.2953 - accuracy: 0.9205 - val_loss: 0.2309 - val_accuracy: 0.9393 - 5s/epoch - 6ms/step
```

```
Epoch 3/20
```

```
844/844 - 5s - loss: 0.2902 - accuracy: 0.9207 - val_loss: 0.2269 - val_accuracy: 0.9417 - 5s/epoch - 6ms/step
```

```
Epoch 4/20
```

```
844/844 - 5s - loss: 0.2849 - accuracy: 0.9238 - val_loss: 0.2523 - val_accuracy: 0.9338 - 5s/epoch - 6ms/step
```

```
Epoch 5/20
```

```
844/844 - 5s - loss: 0.2800 - accuracy: 0.9236 - val_loss: 0.2249 - val_accuracy: 0.9510 - 5s/epoch - 6ms/step
```

```
Epoch 6/20
```

```
844/844 - 5s - loss: 0.2785 - accuracy: 0.9238 - val_loss: 0.2297 - val_accuracy: 0.9422 - 5s/epoch - 6ms/step
```

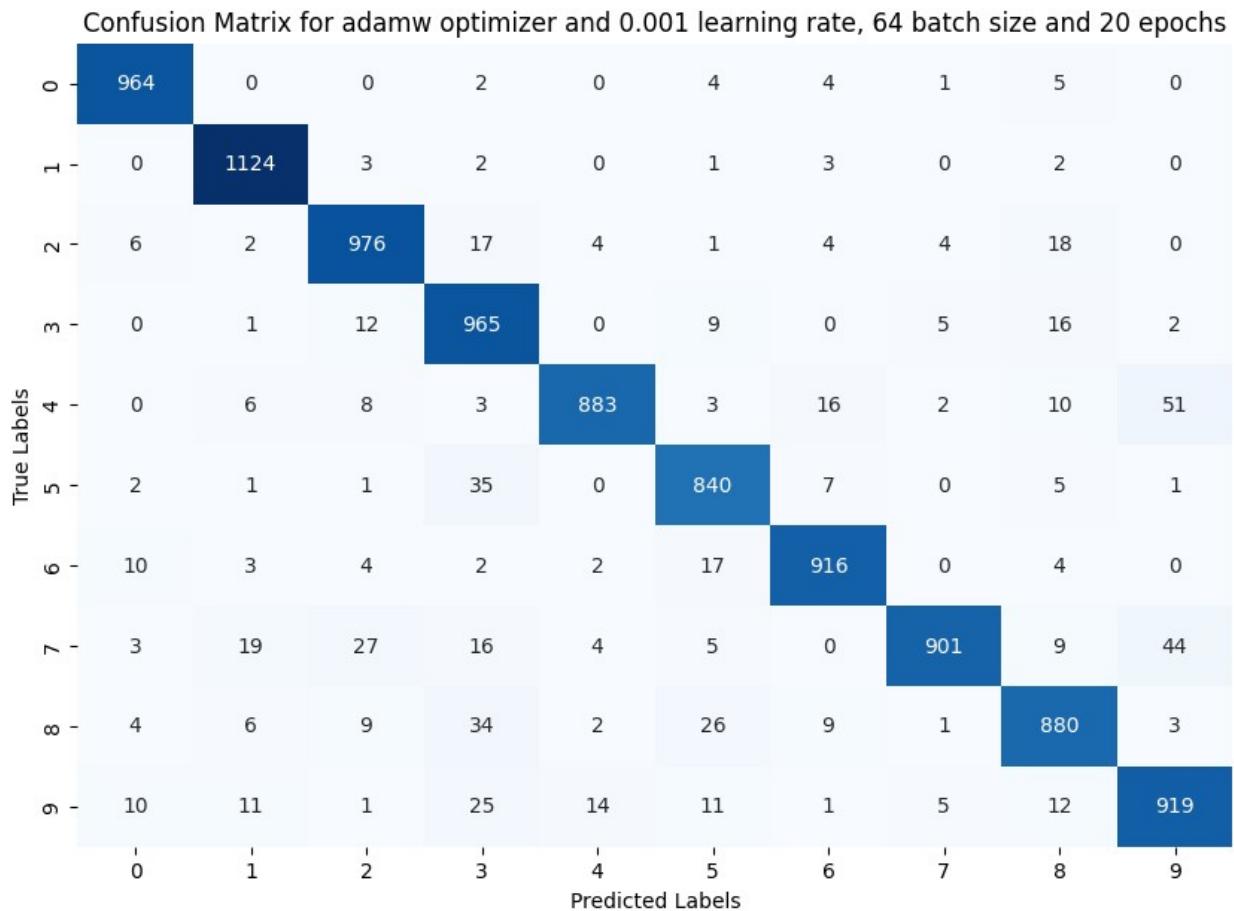
```
Epoch 7/20
```

```
844/844 - 5s - loss: 0.2781 - accuracy: 0.9250 - val_loss: 0.2263 -  
val_accuracy: 0.9463 - 5s/epoch - 6ms/step  
Epoch 8/20  
844/844 - 5s - loss: 0.2792 - accuracy: 0.9243 - val_loss: 0.2319 -  
val_accuracy: 0.9428 - 5s/epoch - 6ms/step  
Epoch 9/20  
844/844 - 5s - loss: 0.2753 - accuracy: 0.9257 - val_loss: 0.2121 -  
val_accuracy: 0.9463 - 5s/epoch - 6ms/step  
Epoch 10/20  
844/844 - 5s - loss: 0.2734 - accuracy: 0.9248 - val_loss: 0.2245 -  
val_accuracy: 0.9407 - 5s/epoch - 6ms/step  
Epoch 11/20  
844/844 - 5s - loss: 0.2743 - accuracy: 0.9250 - val_loss: 0.2200 -  
val_accuracy: 0.9465 - 5s/epoch - 6ms/step  
Epoch 12/20  
844/844 - 5s - loss: 0.2738 - accuracy: 0.9250 - val_loss: 0.2220 -  
val_accuracy: 0.9470 - 5s/epoch - 6ms/step  
Epoch 13/20  
844/844 - 5s - loss: 0.2748 - accuracy: 0.9253 - val_loss: 0.2770 -  
val_accuracy: 0.9250 - 5s/epoch - 6ms/step  
Epoch 14/20  
844/844 - 5s - loss: 0.2754 - accuracy: 0.9256 - val_loss: 0.2233 -  
val_accuracy: 0.9430 - 5s/epoch - 6ms/step  
Epoch 15/20  
844/844 - 5s - loss: 0.2766 - accuracy: 0.9232 - val_loss: 0.2448 -  
val_accuracy: 0.9345 - 5s/epoch - 6ms/step  
Epoch 16/20  
844/844 - 5s - loss: 0.2796 - accuracy: 0.9248 - val_loss: 0.2327 -  
val_accuracy: 0.9447 - 5s/epoch - 6ms/step  
Epoch 17/20  
844/844 - 5s - loss: 0.2783 - accuracy: 0.9243 - val_loss: 0.2321 -  
val_accuracy: 0.9460 - 5s/epoch - 6ms/step  
Epoch 18/20  
844/844 - 5s - loss: 0.2747 - accuracy: 0.9247 - val_loss: 0.2155 -  
val_accuracy: 0.9495 - 5s/epoch - 6ms/step  
Epoch 19/20  
844/844 - 5s - loss: 0.2748 - accuracy: 0.9246 - val_loss: 0.2094 -  
val_accuracy: 0.9447 - 5s/epoch - 6ms/step  
Epoch 20/20  
844/844 - 5s - loss: 0.2750 - accuracy: 0.9249 - val_loss: 0.2325 -  
val_accuracy: 0.9498 - 5s/epoch - 6ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 64  
batch size and 20 epochs:  
[[ 964 0 0 2 0 4 4 1 5 0]  
 [ 0 1124 3 2 0 1 3 0 2 0]  
 [ 6 2 976 17 4 1 4 4 18 0]  
 [ 0 1 12 965 0 9 0 5 16 2]  
 [ 0 6 8 3 883 3 16 2 10 51]]
```

```
[ 2  1  1  35  0  840  7  0  5  1]
[ 10  3  4  2  2  17  916  0  4  0]
[ 3  19  27  16  4  5  0  901  9  44]
[ 4  6  9  34  2  26  9  1  880  3]
[ 10  11  1  25  14  11  1  5  12  919]]
```

Precision: 0.9380

Recall: 0.9368



Training with adamw optimizer and the learning_rate is 0.001, 128 batch size and 5 epochs...

Epoch 1/5

422/422 - 4s - loss: 0.2689 - accuracy: 0.9287 - val_loss: 0.1973 - val_accuracy: 0.9500 - 4s/epoch - 9ms/step

Epoch 2/5

422/422 - 3s - loss: 0.2196 - accuracy: 0.9450 - val_loss: 0.1882 - val_accuracy: 0.9517 - 3s/epoch - 7ms/step

Epoch 3/5

422/422 - 3s - loss: 0.2264 - accuracy: 0.9413 - val_loss: 0.1781 - val_accuracy: 0.9582 - 3s/epoch - 7ms/step

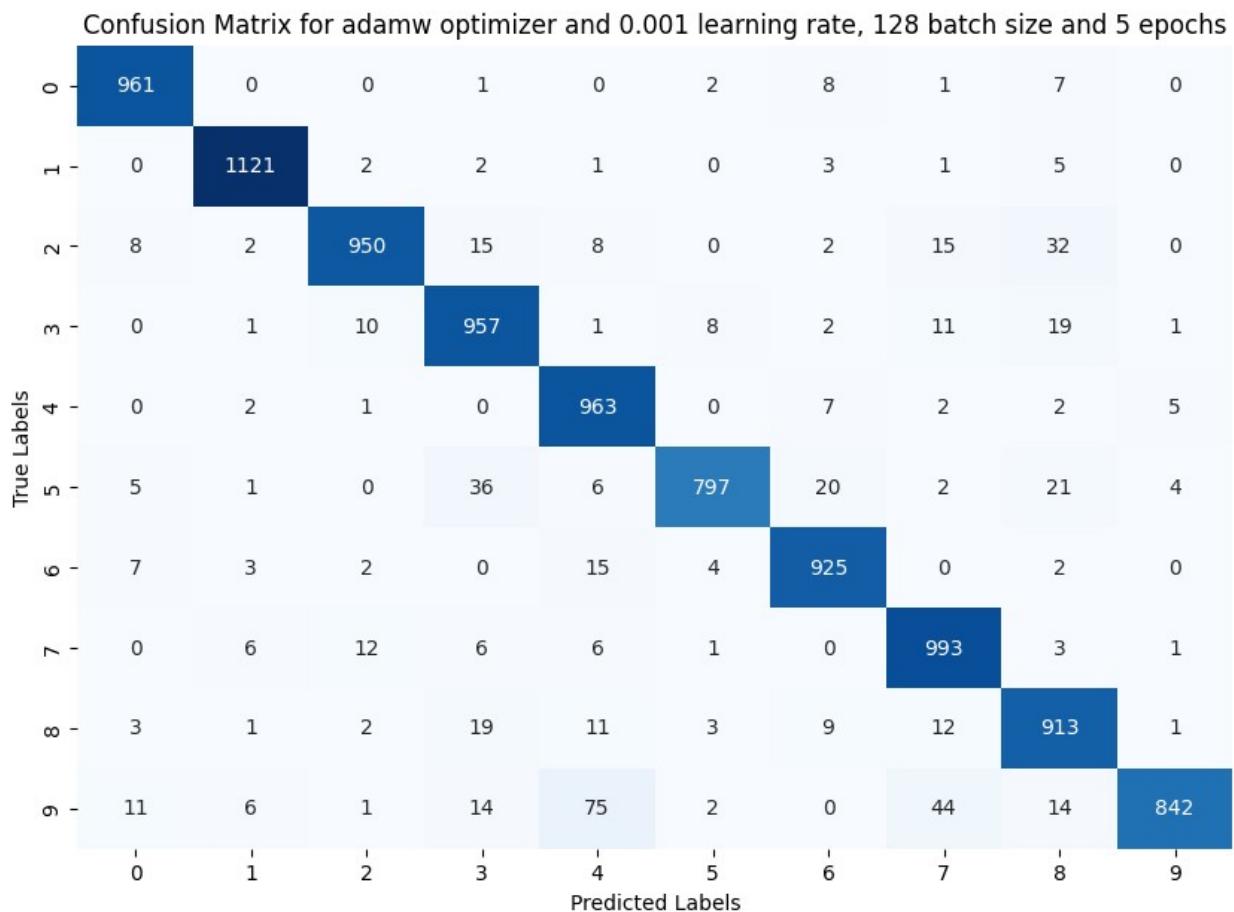
Epoch 4/5

422/422 - 3s - loss: 0.2255 - accuracy: 0.9406 - val_loss: 0.1826 -

```

val_accuracy: 0.9573 - 3s/epoch - 7ms/step
Epoch 5/5
422/422 - 3s - loss: 0.2263 - accuracy: 0.9395 - val_loss: 0.1953 -
val_accuracy: 0.9520 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 128
batch size and 5 epochs:
[[ 961 0 0 1 0 2 8 1 7 0]
 [ 0 1121 2 2 1 0 3 1 5 0]
 [ 8 2 950 15 8 0 2 15 32 0]
 [ 0 1 10 957 1 8 2 11 19 1]
 [ 0 2 1 0 963 0 7 2 2 5]
 [ 5 1 0 36 6 797 20 2 21 4]
 [ 7 3 2 0 15 4 925 0 2 0]
 [ 0 6 12 6 6 1 0 993 3 1]
 [ 3 1 2 19 11 3 9 12 913 1]
 [ 11 6 1 14 75 2 0 44 14 842]]
Precision: 0.9442
Recall: 0.9422

```

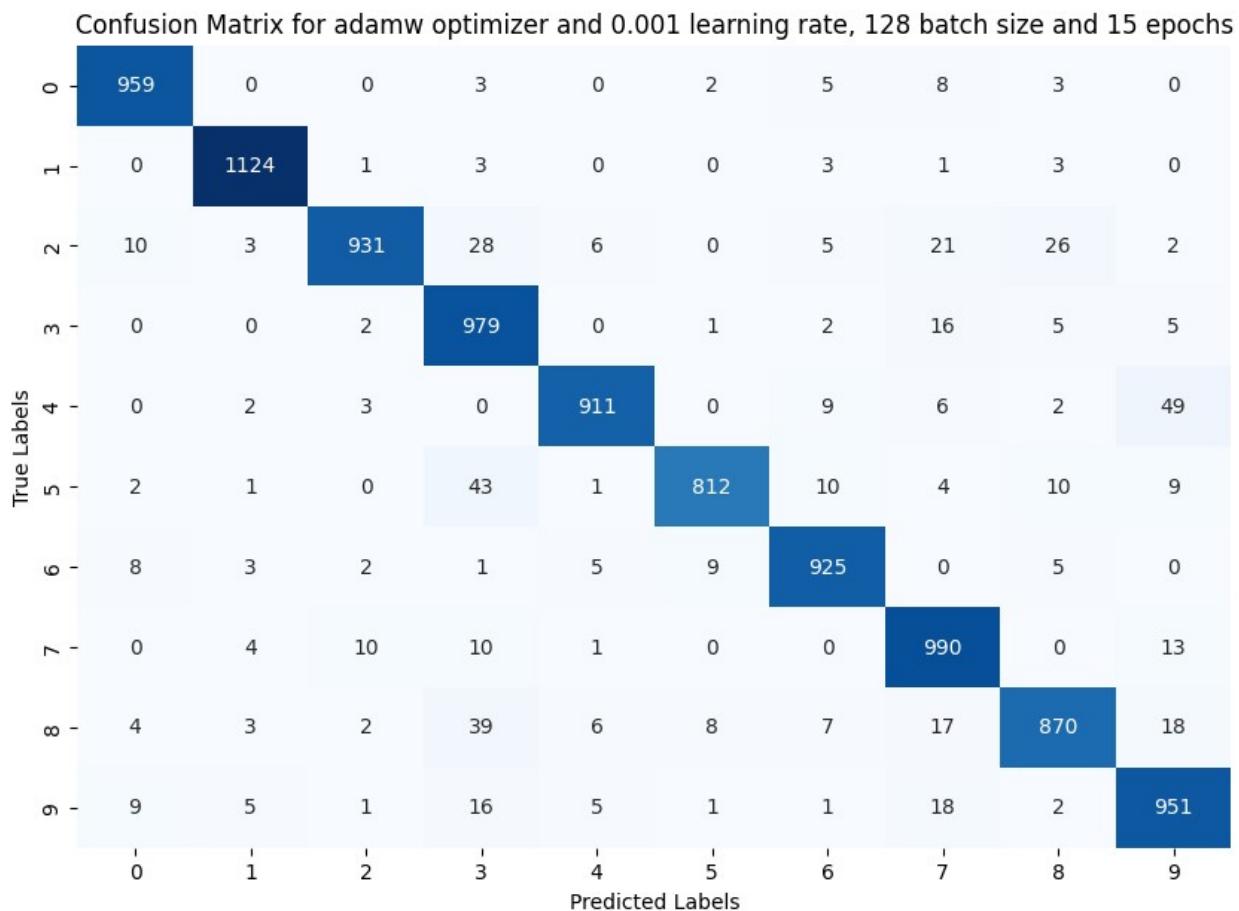


```
Training with adamw optimizer and the learning_rate is 0.001, 128
batch size and 15 epochs...
Epoch 1/15
422/422 - 4s - loss: 0.2686 - accuracy: 0.9303 - val_loss: 0.1828 -
val_accuracy: 0.9575 - 4s/epoch - 9ms/step
Epoch 2/15
422/422 - 3s - loss: 0.2179 - accuracy: 0.9451 - val_loss: 0.1963 -
val_accuracy: 0.9507 - 3s/epoch - 7ms/step
Epoch 3/15
422/422 - 3s - loss: 0.2271 - accuracy: 0.9404 - val_loss: 0.1869 -
val_accuracy: 0.9563 - 3s/epoch - 7ms/step
Epoch 4/15
422/422 - 3s - loss: 0.2314 - accuracy: 0.9385 - val_loss: 0.1801 -
val_accuracy: 0.9608 - 3s/epoch - 7ms/step
Epoch 5/15
422/422 - 3s - loss: 0.2286 - accuracy: 0.9409 - val_loss: 0.1819 -
val_accuracy: 0.9562 - 3s/epoch - 8ms/step
Epoch 6/15
422/422 - 3s - loss: 0.2245 - accuracy: 0.9415 - val_loss: 0.1781 -
val_accuracy: 0.9575 - 3s/epoch - 7ms/step
Epoch 7/15
422/422 - 3s - loss: 0.2219 - accuracy: 0.9412 - val_loss: 0.1734 -
val_accuracy: 0.9590 - 3s/epoch - 7ms/step
Epoch 8/15
422/422 - 3s - loss: 0.2188 - accuracy: 0.9422 - val_loss: 0.1731 -
val_accuracy: 0.9603 - 3s/epoch - 7ms/step
Epoch 9/15
422/422 - 3s - loss: 0.2149 - accuracy: 0.9448 - val_loss: 0.1742 -
val_accuracy: 0.9578 - 3s/epoch - 7ms/step
Epoch 10/15
422/422 - 3s - loss: 0.2122 - accuracy: 0.9444 - val_loss: 0.1679 -
val_accuracy: 0.9625 - 3s/epoch - 7ms/step
Epoch 11/15
422/422 - 3s - loss: 0.2077 - accuracy: 0.9469 - val_loss: 0.1788 -
val_accuracy: 0.9563 - 3s/epoch - 7ms/step
Epoch 12/15
422/422 - 3s - loss: 0.2116 - accuracy: 0.9444 - val_loss: 0.1701 -
val_accuracy: 0.9620 - 3s/epoch - 7ms/step
Epoch 13/15
422/422 - 3s - loss: 0.2117 - accuracy: 0.9449 - val_loss: 0.1859 -
val_accuracy: 0.9485 - 3s/epoch - 7ms/step
Epoch 14/15
422/422 - 3s - loss: 0.2131 - accuracy: 0.9444 - val_loss: 0.1692 -
val_accuracy: 0.9610 - 3s/epoch - 7ms/step
Epoch 15/15
422/422 - 3s - loss: 0.2119 - accuracy: 0.9441 - val_loss: 0.1757 -
val_accuracy: 0.9557 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 128
batch size and 15 epochs:
```

```
[[ 959  0  0  3  0  2  5  8  3  0]
 [ 0 1124  1  3  0  0  3  1  3  0]
 [ 10  3 931  28  6  0  5  21  26  2]
 [ 0  0  2 979  0  1  2  16  5  5]
 [ 0  2  3  0 911  0  9  6  2  49]
 [ 2  1  0  43  1 812  10  4  10  9]
 [ 8  3  2  1  5  9 925  0  5  0]
 [ 0  4  10  10  1  0  0 990  0  13]
 [ 4  3  2  39  6  8  7  17 870  18]
 [ 9  5  1  16  5  1  1  18  2 951]]
```

Precision: 0.9468

Recall: 0.9452



Training with adamw optimizer and the learning_rate is 0.001, 128 batch size and 20 epochs...

Epoch 1/20

422/422 - 4s - loss: 0.2876 - accuracy: 0.9235 - val_loss: 0.1896 - val_accuracy: 0.9575 - 4s/epoch - 9ms/step

Epoch 2/20

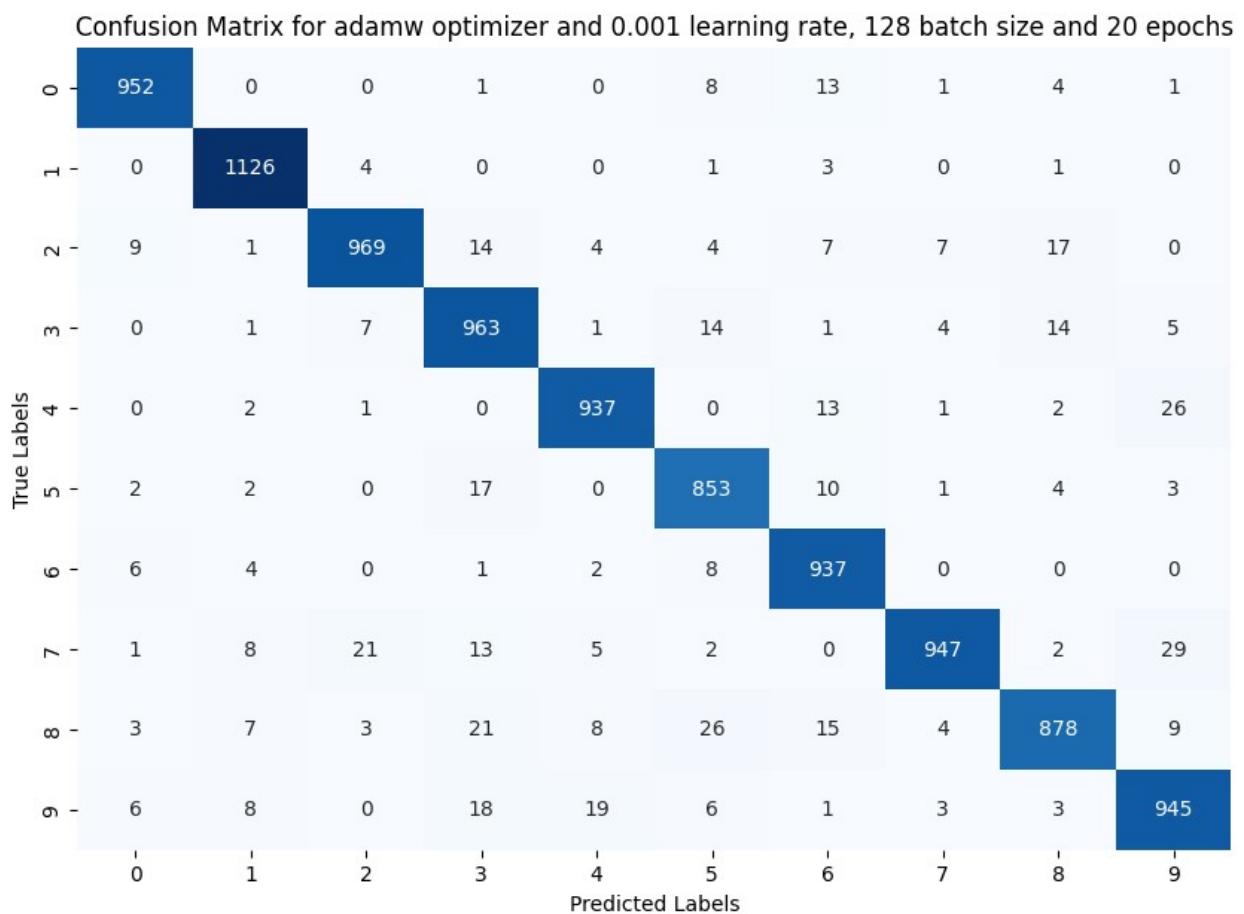
422/422 - 3s - loss: 0.2281 - accuracy: 0.9418 - val_loss: 0.2107 - val_accuracy: 0.9435 - 3s/epoch - 7ms/step

```
Epoch 3/20
422/422 - 3s - loss: 0.2355 - accuracy: 0.9387 - val_loss: 0.1900 -
val_accuracy: 0.9517 - 3s/epoch - 7ms/step
Epoch 4/20
422/422 - 3s - loss: 0.2393 - accuracy: 0.9356 - val_loss: 0.2113 -
val_accuracy: 0.9455 - 3s/epoch - 7ms/step
Epoch 5/20
422/422 - 3s - loss: 0.2372 - accuracy: 0.9358 - val_loss: 0.1901 -
val_accuracy: 0.9520 - 3s/epoch - 7ms/step
Epoch 6/20
422/422 - 3s - loss: 0.2334 - accuracy: 0.9379 - val_loss: 0.1936 -
val_accuracy: 0.9503 - 3s/epoch - 7ms/step
Epoch 7/20
422/422 - 3s - loss: 0.2268 - accuracy: 0.9400 - val_loss: 0.1851 -
val_accuracy: 0.9592 - 3s/epoch - 7ms/step
Epoch 8/20
422/422 - 3s - loss: 0.2222 - accuracy: 0.9412 - val_loss: 0.1866 -
val_accuracy: 0.9522 - 3s/epoch - 7ms/step
Epoch 9/20
422/422 - 3s - loss: 0.2189 - accuracy: 0.9424 - val_loss: 0.1787 -
val_accuracy: 0.9582 - 3s/epoch - 7ms/step
Epoch 10/20
422/422 - 3s - loss: 0.2168 - accuracy: 0.9424 - val_loss: 0.1851 -
val_accuracy: 0.9590 - 3s/epoch - 7ms/step
Epoch 11/20
422/422 - 3s - loss: 0.2163 - accuracy: 0.9429 - val_loss: 0.1722 -
val_accuracy: 0.9628 - 3s/epoch - 7ms/step
Epoch 12/20
422/422 - 3s - loss: 0.2157 - accuracy: 0.9430 - val_loss: 0.1839 -
val_accuracy: 0.9577 - 3s/epoch - 7ms/step
Epoch 13/20
422/422 - 3s - loss: 0.2143 - accuracy: 0.9437 - val_loss: 0.1735 -
val_accuracy: 0.9572 - 3s/epoch - 7ms/step
Epoch 14/20
422/422 - 3s - loss: 0.2151 - accuracy: 0.9431 - val_loss: 0.1846 -
val_accuracy: 0.9568 - 3s/epoch - 7ms/step
Epoch 15/20
422/422 - 3s - loss: 0.2136 - accuracy: 0.9426 - val_loss: 0.1778 -
val_accuracy: 0.9555 - 3s/epoch - 7ms/step
Epoch 16/20
422/422 - 3s - loss: 0.2132 - accuracy: 0.9432 - val_loss: 0.2022 -
val_accuracy: 0.9468 - 3s/epoch - 7ms/step
Epoch 17/20
422/422 - 3s - loss: 0.2125 - accuracy: 0.9440 - val_loss: 0.1781 -
val_accuracy: 0.9597 - 3s/epoch - 7ms/step
Epoch 18/20
422/422 - 3s - loss: 0.2120 - accuracy: 0.9430 - val_loss: 0.2021 -
val_accuracy: 0.9455 - 3s/epoch - 7ms/step
Epoch 19/20
```

```

422/422 - 3s - loss: 0.2126 - accuracy: 0.9429 - val_loss: 0.1752 -
val_accuracy: 0.9585 - 3s/epoch - 7ms/step
Epoch 20/20
422/422 - 3s - loss: 0.2107 - accuracy: 0.9448 - val_loss: 0.1658 -
val_accuracy: 0.9607 - 3s/epoch - 7ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 128
batch size and 20 epochs:
[[ 952    0    0    1    0    8   13    1    4    1]
 [  0 1126    4    0    0    1    3    0    1    0]
 [  9    1 969   14    4    4    7    7   17    0]
 [  0    1    7 963    1   14    1    4   14    5]
 [  0    2    1    0 937    0   13    1    2   26]
 [  2    2    0   17    0 853   10    1    4    3]
 [  6    4    0    1    2    8 937    0    0    0]
 [  1    8   21   13    5    2    0 947    2   29]
 [  3    7   31   21    8   26   15    4 878    9]
 [  6    8    0   18   19    6    1    3    3 945]]
Precision: 0.9511
Recall: 0.9507

```



```
Training with adamw optimizer and the learning_rate is 0.001, 256
batch size and 5 epochs...
Epoch 1/5
211/211 - 3s - loss: 0.2910 - accuracy: 0.9204 - val_loss: 0.1616 -
val_accuracy: 0.9632 - 3s/epoch - 14ms/step
Epoch 2/5
211/211 - 2s - loss: 0.1898 - accuracy: 0.9537 - val_loss: 0.1523 -
val_accuracy: 0.9662 - 2s/epoch - 11ms/step
Epoch 3/5
211/211 - 2s - loss: 0.1879 - accuracy: 0.9518 - val_loss: 0.1448 -
val_accuracy: 0.9675 - 2s/epoch - 11ms/step
Epoch 4/5
211/211 - 2s - loss: 0.1890 - accuracy: 0.9509 - val_loss: 0.1596 -
val_accuracy: 0.9632 - 2s/epoch - 11ms/step
Epoch 5/5
211/211 - 2s - loss: 0.1898 - accuracy: 0.9499 - val_loss: 0.1618 -
val_accuracy: 0.9585 - 2s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 256
batch size and 5 epochs:
[[ 962    0    2    1    1    2    7    2    3    0]
 [  0 1127    4    0    0    0    3    0    1    0]
 [  7   3  986    5    4    1    4   12   10    0]
 [  0   1   11  974    0    6    0    9    7    2]
 [  1   2    4    0  959    0    5    2    2    7]
 [  5   4    0   32    3  823   12    3    5    5]
 [  9   3    5    1    7    9  923    0    1    0]
 [  0   11   17    7    3    1    0  982    0    7]
 [  7   12    7   22   12   11    9   18  870    6]
 [  8   13    1   16   32    2    1   18    4  914]]
```

Precision: 0.9524
Recall: 0.9520

Confusion Matrix for adamw optimizer and 0.001 learning rate, 256 batch size and 5 epochs

		Predicted Labels									
		0	1	2	3	4	5	6	7	8	9
True Labels	0	962	0	2	1	1	2	7	2	3	0
	1	0	1127	4	0	0	0	3	0	1	0
2	7	3	986	5	4	1	4	12	10	0	0
3	0	1	11	974	0	6	0	9	7	2	0
4	1	2	4	0	959	0	5	2	2	7	0
5	5	4	0	32	3	823	12	3	5	5	5
6	9	3	5	1	7	9	923	0	1	0	0
7	0	11	17	7	3	1	0	982	0	7	0
8	7	12	7	22	12	11	9	18	870	6	0
9	8	13	1	16	32	2	1	18	4	914	9
		0	1	2	3	4	5	6	7	8	9

Training with adamw optimizer and the learning_rate is 0.001, 256 batch size and 15 epochs...

Epoch 1/15

211/211 - 3s - loss: 0.3040 - accuracy: 0.9162 - val_loss: 0.1693 - val_accuracy: 0.9622 - 3s/epoch - 14ms/step

Epoch 2/15

211/211 - 2s - loss: 0.1976 - accuracy: 0.9517 - val_loss: 0.1638 -
val_accuracy: 0.9632 - 2s/epoch - 11ms/step

Epoch 3/15

211/211 - 2s - loss: 0.1936 - accuracy: 0.9508 - val_loss: 0.1581 - val_accuracy: 0.9645 - 2s/epoch - 11ms/step

Epoch 4/15

211/211 - 2s - loss: 0.1942 - accuracy: 0.9495 - val_loss: 0.1656 - val_accuracy: 0.9615 - 2s/epoch - 11ms/step

Epoch 5/15

211/211 - 2s - loss: 0.1952 - accuracy: 0.9491 - val_loss: 0.1632 -
val_accuracy: 0.9600 - 2s/epoch - 11ms/step

Epoch 6/15
211/211

```
211/211 - 2s - loss: 0.1941 - accuracy: 0.9492 - val_loss: 0.1774 -  
val_accuracy: 0.9500 - 2s/epoch - 11ms/step  
Epoch 7/15
```

Epoch 77/15

```
211/211 - 2s - loss: 0.1932 - accuracy: 0.9503 - val_loss: 0.1656 -  
val_accuracy: 0.9618 - 2s/epoch - 11ms/step  
Epoch 8/15  
211/211 - 2s - loss: 0.1951 - accuracy: 0.9483 - val_loss: 0.1716 -  
val_accuracy: 0.9590 - 2s/epoch - 11ms/step  
Epoch 9/15  
211/211 - 2s - loss: 0.1917 - accuracy: 0.9495 - val_loss: 0.1552 -  
val_accuracy: 0.9627 - 2s/epoch - 11ms/step  
Epoch 10/15  
211/211 - 2s - loss: 0.1915 - accuracy: 0.9495 - val_loss: 0.1626 -  
val_accuracy: 0.9620 - 2s/epoch - 11ms/step  
Epoch 11/15  
211/211 - 2s - loss: 0.1863 - accuracy: 0.9513 - val_loss: 0.1630 -  
val_accuracy: 0.9585 - 2s/epoch - 11ms/step  
Epoch 12/15  
211/211 - 2s - loss: 0.1895 - accuracy: 0.9491 - val_loss: 0.1598 -  
val_accuracy: 0.9610 - 2s/epoch - 11ms/step  
Epoch 13/15  
211/211 - 2s - loss: 0.1843 - accuracy: 0.9518 - val_loss: 0.1548 -  
val_accuracy: 0.9640 - 2s/epoch - 11ms/step  
Epoch 14/15  
211/211 - 2s - loss: 0.1847 - accuracy: 0.9513 - val_loss: 0.1604 -  
val_accuracy: 0.9602 - 2s/epoch - 11ms/step  
Epoch 15/15  
211/211 - 2s - loss: 0.1809 - accuracy: 0.9531 - val_loss: 0.1559 -  
val_accuracy: 0.9645 - 2s/epoch - 11ms/step  
313/313 [=====] - 1s 2ms/step  
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 256  
batch size and 15 epochs:  
[[ 959  0  1  1  0  1  12  1  5  0]  
 [ 0 1123  3  0  0  0  4  0  5  0]  
 [ 4  1 991  7  3  0  4  7  15  0]  
 [ 0  1 11 965  0  6  1 10  16  0]  
 [ 0  0  2  0 953  0 12  1  4 10]  
 [ 2  3  0 24  1 826  19  1 14  2]  
 [ 6  2  1  0  2  2 943  0  2  0]  
 [ 3  9 20  7  4  0  0 973  5  7]  
 [ 3  2  6 13  6  4 11  5 922  2]  
 [ 9  6  1 16  41  4  1 12 12 907]]  
Precision: 0.9567  
Recall: 0.9562
```

Confusion Matrix for adamw optimizer and 0.001 learning rate, 256 batch size and 15 epochs

	0	1	2	3	4	5	6	7	8	9	10
0	959	0	1	1	0	1	12	1	5	0	
1	0	1123	3	0	0	0	4	0	5	0	
2	4	1	991	7	3	0	4	7	15	0	
3	0	1	11	965	0	6	1	10	16	0	
4	0	0	2	0	953	0	12	1	4	10	
5	2	3	0	24	1	826	19	1	14	2	
6	6	2	1	0	2	2	943	0	2	0	
7	3	9	20	7	4	0	0	973	5	7	
8	3	2	6	13	6	4	11	5	922	2	
9	9	6	1	16	41	4	1	12	12	907	
10	1	2	3	4	5	6	7	8	9	9	
	0	1	2	3	4	5	6	7	8	9	

Training with adamw optimizer and the learning_rate is 0.001, 256 batch size and 20 epochs...

Epoch 1/20

211/211 - 3s - loss: 0.3039 - accuracy: 0.9168 - val_loss: 0.1573 - val_accuracy: 0.9652 - 3s/epoch - 14ms/step

Epoch 2/20

211/211 - 2s - loss: 0.1796 - accuracy: 0.9562 - val_loss: 0.1451 - val_accuracy: 0.9698 - 2s/epoch - 11ms/step

Epoch 3/20

211/211 - 2s - loss: 0.1745 - accuracy: 0.9571 - val_loss: 0.1468 - val_accuracy: 0.9675 - 2s/epoch - 11ms/step

Epoch 4/20

211/211 - 2s - loss: 0.1830 - accuracy: 0.9540 - val_loss: 0.1603 - val_accuracy: 0.9597 - 2s/epoch - 11ms/step

Epoch 5/20

211/211 - 2s - loss: 0.1866 - accuracy: 0.9525 - val_loss: 0.1512 - val_accuracy: 0.9642 - 2s/epoch - 11ms/step

Epoch 6/20

211/211 - 2s - loss: 0.1910 - accuracy: 0.9497 - val_loss: 0.1547 - val_accuracy: 0.9617 - 2s/epoch - 11ms/step

Epoch 7/20

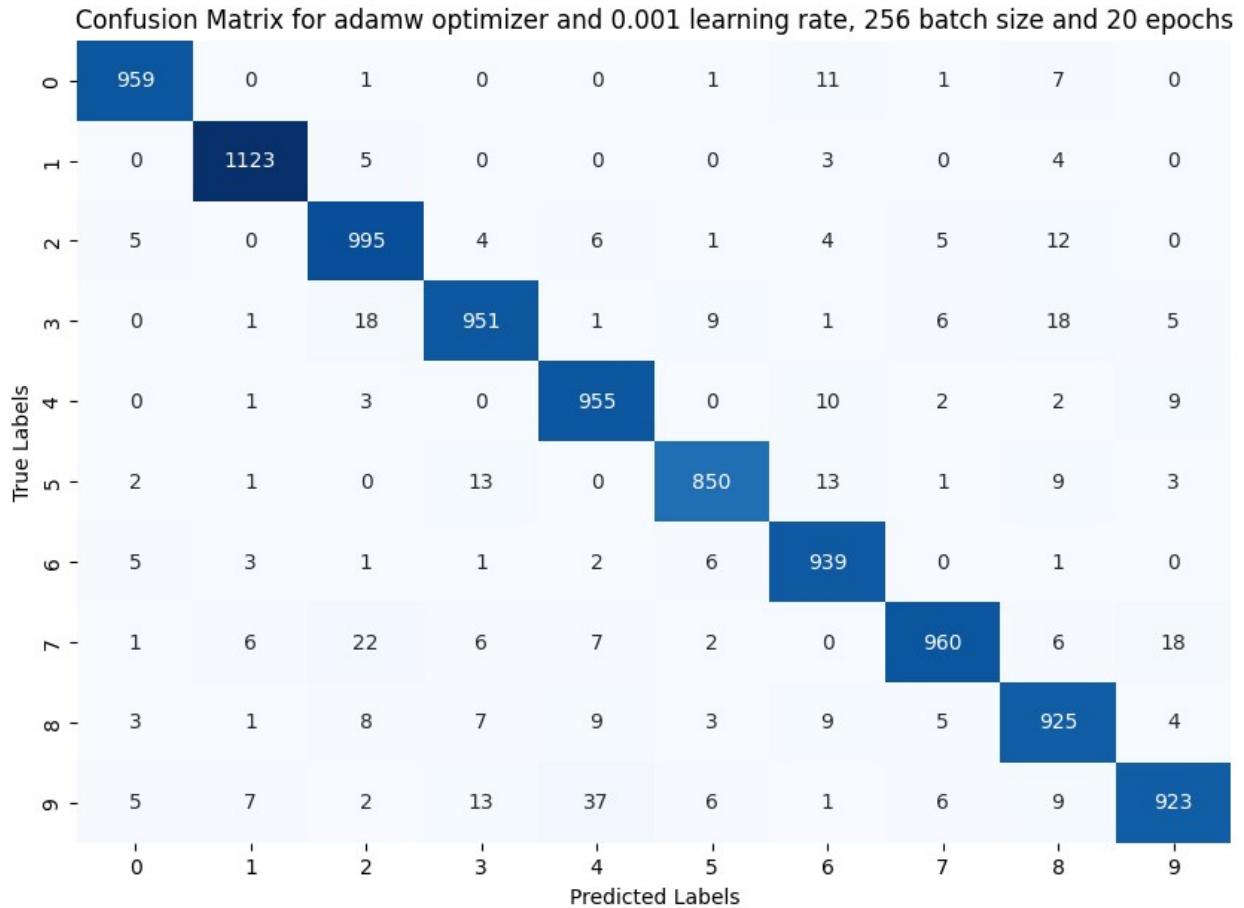
```

211/211 - 2s - loss: 0.1895 - accuracy: 0.9509 - val_loss: 0.1523 -
val_accuracy: 0.9648 - 2s/epoch - 11ms/step
Epoch 8/20
211/211 - 2s - loss: 0.1928 - accuracy: 0.9486 - val_loss: 0.1656 -
val_accuracy: 0.9610 - 2s/epoch - 11ms/step
Epoch 9/20
211/211 - 2s - loss: 0.1906 - accuracy: 0.9493 - val_loss: 0.1559 -
val_accuracy: 0.9625 - 2s/epoch - 11ms/step
Epoch 10/20
211/211 - 2s - loss: 0.1916 - accuracy: 0.9492 - val_loss: 0.1599 -
val_accuracy: 0.9592 - 2s/epoch - 11ms/step
Epoch 11/20
211/211 - 2s - loss: 0.1904 - accuracy: 0.9499 - val_loss: 0.1592 -
val_accuracy: 0.9630 - 2s/epoch - 11ms/step
Epoch 12/20
211/211 - 2s - loss: 0.1923 - accuracy: 0.9486 - val_loss: 0.1621 -
val_accuracy: 0.9612 - 2s/epoch - 11ms/step
Epoch 13/20
211/211 - 2s - loss: 0.1886 - accuracy: 0.9502 - val_loss: 0.1598 -
val_accuracy: 0.9633 - 2s/epoch - 11ms/step
Epoch 14/20
211/211 - 2s - loss: 0.1900 - accuracy: 0.9489 - val_loss: 0.1601 -
val_accuracy: 0.9607 - 2s/epoch - 11ms/step
Epoch 15/20
211/211 - 2s - loss: 0.1865 - accuracy: 0.9515 - val_loss: 0.1534 -
val_accuracy: 0.9633 - 2s/epoch - 11ms/step
Epoch 16/20
211/211 - 2s - loss: 0.1867 - accuracy: 0.9508 - val_loss: 0.1596 -
val_accuracy: 0.9640 - 2s/epoch - 11ms/step
Epoch 17/20
211/211 - 2s - loss: 0.1841 - accuracy: 0.9514 - val_loss: 0.1615 -
val_accuracy: 0.9640 - 2s/epoch - 11ms/step
Epoch 18/20
211/211 - 2s - loss: 0.1812 - accuracy: 0.9527 - val_loss: 0.1495 -
val_accuracy: 0.9662 - 2s/epoch - 11ms/step
Epoch 19/20
211/211 - 2s - loss: 0.1820 - accuracy: 0.9521 - val_loss: 0.1508 -
val_accuracy: 0.9653 - 2s/epoch - 11ms/step
Epoch 20/20
211/211 - 2s - loss: 0.1801 - accuracy: 0.9526 - val_loss: 0.1503 -
val_accuracy: 0.9642 - 2s/epoch - 11ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 256
batch size and 20 epochs:
[[ 959   0   1   0   0   1  11   1   7   0]
 [  0 1123   5   0   0   0   3   0   4   0]
 [  5   0 995   4   6   1   4   5  12   0]
 [  0   1  18  951   1   9   1   6  18   5]
 [  0   1   3   0  955   0  10   2   2   9]]
```

```
[ 2 1 0 13 0 850 13 1 9 3]
[ 5 3 1 1 2 6 939 0 1 0]
[ 1 6 22 6 7 2 0 960 6 18]
[ 3 1 8 7 9 3 9 5 925 4]
[ 5 7 2 13 37 6 1 6 9 923]]
```

Precision: 0.9582

Recall: 0.9580



Training with adamw optimizer and the learning_rate is 0.001, 16 batch size and 5 epochs...

Epoch 1/5

3375/3375 - 14s - loss: 0.4534 - accuracy: 0.8729 - val_loss: 0.4213 - val_accuracy: 0.8725 - 14s/epoch - 4ms/step

Epoch 2/5

3375/3375 - 13s - loss: 0.4414 - accuracy: 0.8741 - val_loss: 0.3546 - val_accuracy: 0.9135 - 13s/epoch - 4ms/step

Epoch 3/5

3375/3375 - 13s - loss: 0.4348 - accuracy: 0.8763 - val_loss: 0.3328 - val_accuracy: 0.9053 - 13s/epoch - 4ms/step

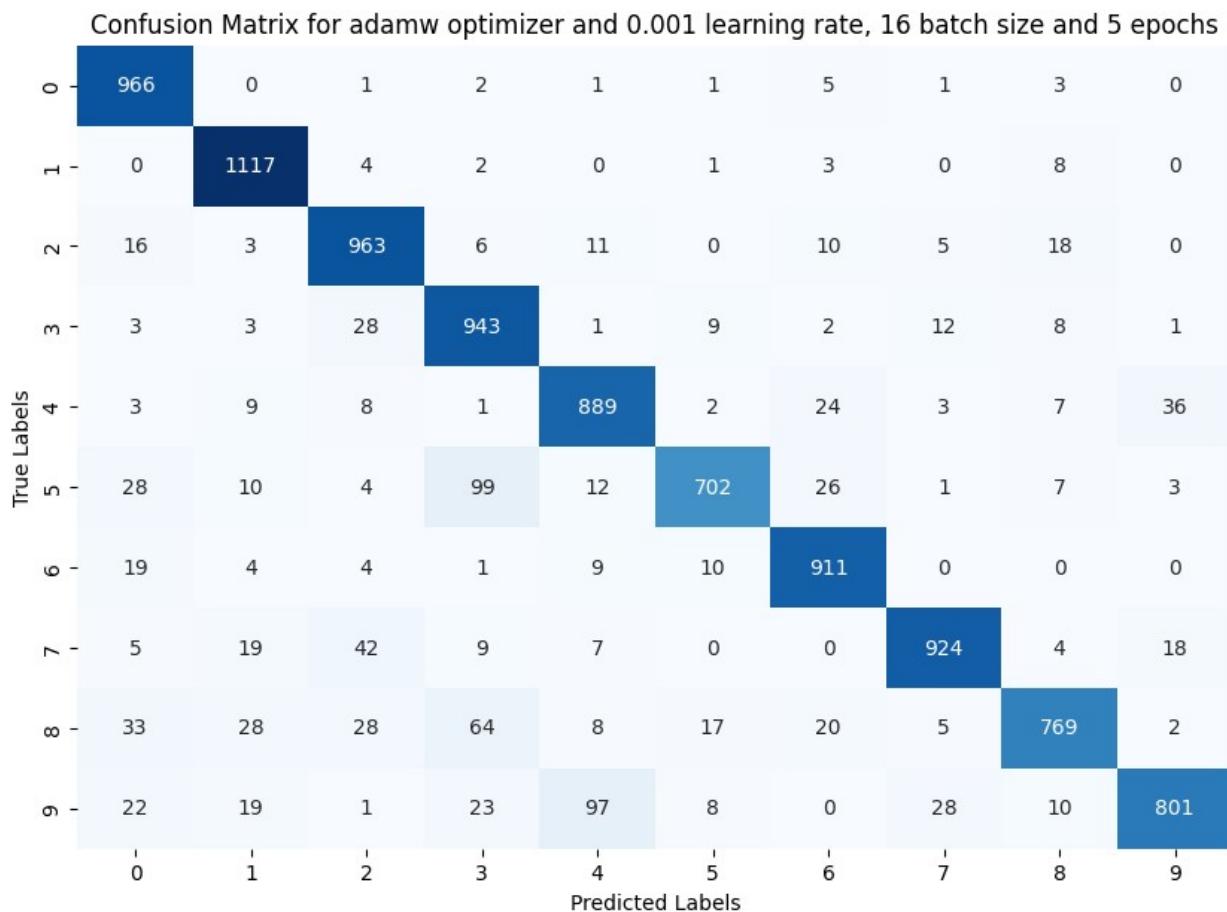
Epoch 4/5

3375/3375 - 13s - loss: 0.4319 - accuracy: 0.8763 - val_loss: 0.3994 -

```

val_accuracy: 0.8895 - 13s/epoch - 4ms/step
Epoch 5/5
3375/3375 - 13s - loss: 0.4366 - accuracy: 0.8755 - val_loss: 0.3581 -
val_accuracy: 0.9115 - 13s/epoch - 4ms/step
313/313 [=====] - 1s 2ms/step
Confusion Matrix adamw optimizer and the learning_rate is 0.001, 16
batch size and 5 epochs:
[[ 966 0 1 2 1 1 5 1 3 0]
 [ 0 1117 4 2 0 1 3 0 8 0]
 [ 16 3 963 6 11 0 10 5 18 0]
 [ 3 3 28 943 1 9 2 12 8 1]
 [ 3 9 8 1 889 2 24 3 7 36]
 [ 28 10 4 99 12 702 26 1 7 3]
 [ 19 4 4 1 9 10 911 0 0 0]
 [ 5 19 42 9 7 0 0 924 4 18]
 [ 33 28 28 64 8 17 20 5 769 2]
 [ 22 19 1 23 97 8 0 28 10 801]]
Precision: 0.9014
Recall: 0.8985

```



Training with adamw optimizer and the learning_rate is 0.001, 16 batch size and 15 epochs...

Epoch 1/15
3375/3375 - 14s - loss: 0.4567 - accuracy: 0.8738 - val_loss: 0.4210 - val_accuracy: 0.8708 - 14s/epoch - 4ms/step

Epoch 2/15
3375/3375 - 14s - loss: 0.4396 - accuracy: 0.8746 - val_loss: 0.3814 - val_accuracy: 0.9042 - 14s/epoch - 4ms/step

Epoch 3/15
3375/3375 - 14s - loss: 0.4363 - accuracy: 0.8751 - val_loss: 0.5014 - val_accuracy: 0.8418 - 14s/epoch - 4ms/step

Epoch 4/15
3375/3375 - 14s - loss: 0.4344 - accuracy: 0.8749 - val_loss: 0.3677 - val_accuracy: 0.8903 - 14s/epoch - 4ms/step

Epoch 5/15
3375/3375 - 14s - loss: 0.4305 - accuracy: 0.8762 - val_loss: 0.3703 - val_accuracy: 0.8998 - 14s/epoch - 4ms/step

Epoch 6/15
3375/3375 - 14s - loss: 0.4306 - accuracy: 0.8775 - val_loss: 0.3328 - val_accuracy: 0.9195 - 14s/epoch - 4ms/step

Epoch 7/15
3375/3375 - 14s - loss: 0.4281 - accuracy: 0.8785 - val_loss: 0.4185 - val_accuracy: 0.8747 - 14s/epoch - 4ms/step

Epoch 8/15
3375/3375 - 14s - loss: 0.4304 - accuracy: 0.8753 - val_loss: 0.3389 - val_accuracy: 0.9180 - 14s/epoch - 4ms/step

Epoch 9/15
3375/3375 - 14s - loss: 0.4259 - accuracy: 0.8783 - val_loss: 0.3284 - val_accuracy: 0.9127 - 14s/epoch - 4ms/step

Epoch 10/15
3375/3375 - 14s - loss: 0.4287 - accuracy: 0.8775 - val_loss: 0.3269 - val_accuracy: 0.9195 - 14s/epoch - 4ms/step

Epoch 11/15
3375/3375 - 14s - loss: 0.4314 - accuracy: 0.8756 - val_loss: 0.3373 - val_accuracy: 0.9108 - 14s/epoch - 4ms/step

Epoch 12/15
3375/3375 - 14s - loss: 0.4297 - accuracy: 0.8756 - val_loss: 0.3427 - val_accuracy: 0.9090 - 14s/epoch - 4ms/step

Epoch 13/15
3375/3375 - 14s - loss: 0.4266 - accuracy: 0.8774 - val_loss: 0.3245 - val_accuracy: 0.9227 - 14s/epoch - 4ms/step

Epoch 14/15
3375/3375 - 14s - loss: 0.4338 - accuracy: 0.8753 - val_loss: 0.3630 - val_accuracy: 0.9058 - 14s/epoch - 4ms/step

Epoch 15/15
3375/3375 - 14s - loss: 0.4311 - accuracy: 0.8780 - val_loss: 0.3493 - val_accuracy: 0.9082 - 14s/epoch - 4ms/step

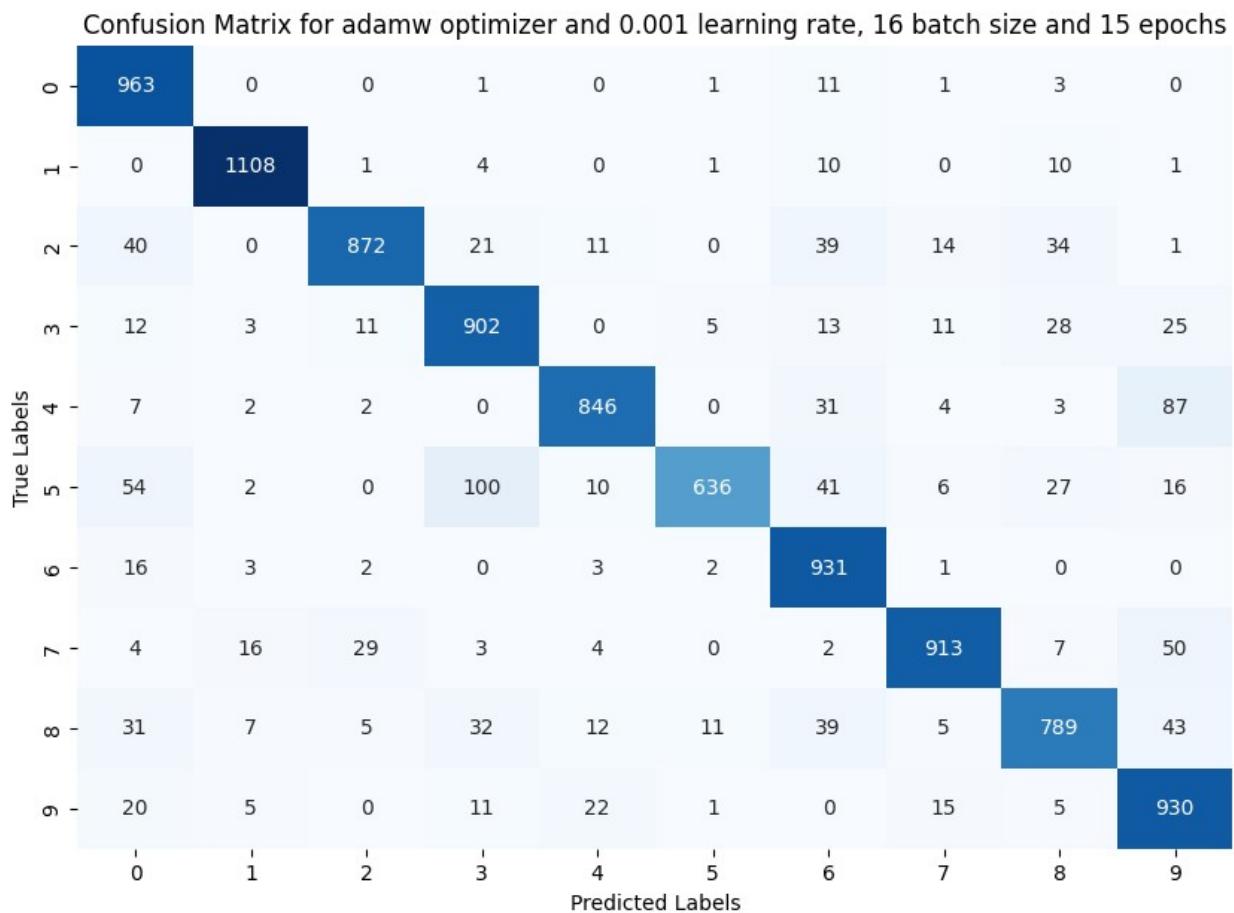
313/313 [=====] - 1s 2ms/step

Confusion Matrix adamw optimizer and the learning_rate is 0.001, 16 batch size and 15 epochs:

```
[[ 963  0  0  1  0  1  11  1  3  0]
 [ 0 1108  1  4  0  1  10  0 10  1]
 [ 40  0 872 21 11  0 39 14 34  1]
 [ 12  3 11 902  0  5 13 11 28 25]
 [ 7  2  2  0 846  0 31  4  3 87]
 [ 54  2  0 100 10 636 41  6 27 16]
 [ 16  3  2  0  3  2 931  1  0  0]
 [ 4  16 29  3  4  0  2 913  7 50]
 [ 31  7  5 32 12 11 39  5 789 43]
 [ 20  5  0 11 22  1  0 15  5 930]]
```

Precision: 0.8951

Recall: 0.8890



Training with adamw optimizer and the learning_rate is 0.001, 16 batch size and 20 epochs...

Epoch 1/20

3375/3375 - 14s - loss: 0.4509 - accuracy: 0.8735 - val_loss: 0.3924 - val_accuracy: 0.8955 - 14s/epoch - 4ms/step

Epoch 2/20

3375/3375 - 14s - loss: 0.4376 - accuracy: 0.8756 - val_loss: 0.3489 - val_accuracy: 0.9053 - 14s/epoch - 4ms/step

```
Epoch 3/20
3375/3375 - 14s - loss: 0.4345 - accuracy: 0.8756 - val_loss: 0.3117 -
val_accuracy: 0.9263 - 14s/epoch - 4ms/step
Epoch 4/20
3375/3375 - 14s - loss: 0.4284 - accuracy: 0.8786 - val_loss: 0.3840 -
val_accuracy: 0.8973 - 14s/epoch - 4ms/step
Epoch 5/20
3375/3375 - 14s - loss: 0.4272 - accuracy: 0.8774 - val_loss: 0.4048 -
val_accuracy: 0.8880 - 14s/epoch - 4ms/step
Epoch 6/20
3375/3375 - 14s - loss: 0.4244 - accuracy: 0.8789 - val_loss: 0.3346 -
val_accuracy: 0.9163 - 14s/epoch - 4ms/step
Epoch 7/20
3375/3375 - 14s - loss: 0.4298 - accuracy: 0.8770 - val_loss: 0.3609 -
val_accuracy: 0.8960 - 14s/epoch - 4ms/step
Epoch 8/20
3375/3375 - 14s - loss: 0.4320 - accuracy: 0.8750 - val_loss: 0.3534 -
val_accuracy: 0.9088 - 14s/epoch - 4ms/step
Epoch 9/20
3375/3375 - 14s - loss: 0.4293 - accuracy: 0.8766 - val_loss: 0.3815 -
val_accuracy: 0.8987 - 14s/epoch - 4ms/step
Epoch 10/20
3375/3375 - 15s - loss: 0.4269 - accuracy: 0.8764 - val_loss: 0.3515 -
val_accuracy: 0.9000 - 15s/epoch - 4ms/step
Epoch 11/20
3375/3375 - 15s - loss: 0.4297 - accuracy: 0.8752 - val_loss: 0.3234 -
val_accuracy: 0.9167 - 15s/epoch - 4ms/step
Epoch 12/20
3375/3375 - 15s - loss: 0.4284 - accuracy: 0.8781 - val_loss: 0.3499 -
val_accuracy: 0.9048 - 15s/epoch - 4ms/step
Epoch 13/20
3375/3375 - 14s - loss: 0.4258 - accuracy: 0.8785 - val_loss: 0.3511 -
val_accuracy: 0.9025 - 14s/epoch - 4ms/step
Epoch 14/20
3375/3375 - 14s - loss: 0.4246 - accuracy: 0.8777 - val_loss: 0.4174 -
val_accuracy: 0.8833 - 14s/epoch - 4ms/step
Epoch 15/20
3375/3375 - 14s - loss: 0.4255 - accuracy: 0.8770 - val_loss: 0.3693 -
val_accuracy: 0.8972 - 14s/epoch - 4ms/step
Epoch 16/20
3375/3375 - 14s - loss: 0.4232 - accuracy: 0.8803 - val_loss: 0.3535 -
val_accuracy: 0.8982 - 14s/epoch - 4ms/step
Epoch 17/20
3375/3375 - 14s - loss: 0.4247 - accuracy: 0.8777 - val_loss: 0.3165 -
val_accuracy: 0.9198 - 14s/epoch - 4ms/step
Epoch 18/20
3375/3375 - 14s - loss: 0.4260 - accuracy: 0.8771 - val_loss: 0.3728 -
val_accuracy: 0.8923 - 14s/epoch - 4ms/step
Epoch 19/20
```

3375/3375 - 14s - loss: 0.4292 - accuracy: 0.8760 - val_loss: 0.3553 - val_accuracy: 0.9112 - 14s/epoch - 4ms/step

Epoch 20/20

3375/3375 - 14s - loss: 0.4286 - accuracy: 0.8771 - val_loss: 0.3360 - val_accuracy: 0.9093 - 14s/epoch - 4ms/step

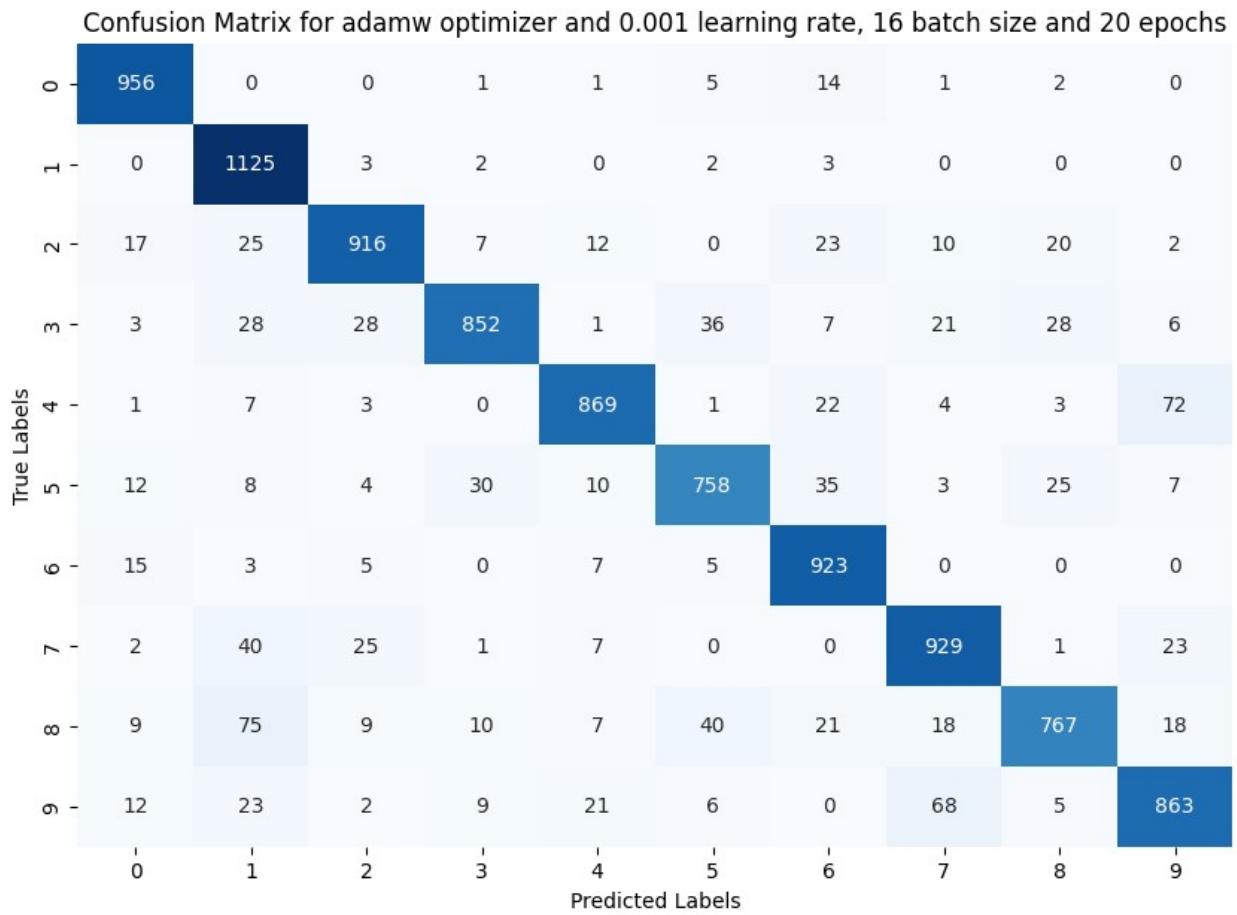
313/313 [=====] - 1s 2ms/step

Confusion Matrix adamw optimizer and the learning_rate is 0.001, 16 batch size and 20 epochs:

```
[[ 956   0   0   1   1   5  14   1   2   0]
 [  0 1125   3   2   0   2   3   0   0   0]
 [ 17  25 916   7  12   0  23  10  20  2]
 [  3  28  28 852   1  36   7  21  28  6]
 [  1   7   3   0 869   1  22   4   3 72]
 [ 12   8   4  30  10 758  35   3  25  7]
 [ 15   3   5   0   7   5  923   0   0   0]
 [  2  40  25   1   7   0   0 929   1 23]
 [  9  75   9  10   7  40   21  18 767  18]
 [ 12  23   2   9  21   6   0  68   5 863]]
```

Precision: 0.8975

Recall: 0.8958



```

def create_flexible_model_exp(conv_activation='relu',
dense_activation='relu', num_conv_layers=1, num_dense_layers=1,
num_neurons=100, optimizer_name="adam", lr=0.001):
    model = Sequential()

    # Add specified number of convolutional layers
    for i in range(num_conv_layers):
        if i == 0:
            model.add(Conv2D(32, (3, 3), activation=conv_activation,
kernel_initializer="he_uniform", input_shape=(28, 28, 1)))
        else:
            model.add(Conv2D(32, (3, 3), activation=conv_activation,
kernel_initializer="he_uniform"))
        model.add(MaxPooling2D((2, 2)))

    model.add(Flatten())

    # Add specified number of dense layers
    for _ in range(num_dense_layers):
        model.add(Dense(num_neurons, activation=dense_activation,
kernel_initializer="he_uniform"))

    model.add(Dense(10, activation='softmax')) # Output layer

    # Optimizer selection
    if optimizer_name == "adam":
        opt = Adam(learning_rate=lr)
    elif optimizer_name == "nadam":
        opt = Nadam(learning_rate=lr)
    elif optimizer_name == "adamw":
        opt = tfa.optimizers.AdamW(learning_rate=lr)
    #elif optimizer_name == "adadelta":
    #    opt = Adadelta(learning_rate=lr)
    #elif optimizer_name == "rmsprop":
    #    #opt = RMSprop(learning_rate=lr, momentum=0.9)
    else:
        raise ValueError("Unsupported Optimizer")

    model.compile(optimizer=opt, loss='categorical_crossentropy',
metrics=['accuracy'])

    return model

```

Key Observations:

Optimizer Sensitivity to Learning Rate: The optimizer's performance is highly sensitive to the learning rate. For instance, with an excessively high learning rate (e.g., 0.1), the model's accuracy significantly suffers, as evident from the adam optimizer experiments. This suggests that the optimizer overshoots the minima in the loss landscape, failing to converge to a solution that generalizes well.

Effect of Batch Size and Epochs:

Larger batch sizes and higher epochs do not necessarily translate to better model performance, especially when paired with suboptimal learning rates. For instance, larger batch sizes with a high learning rate of 0.1 consistently resulted in poor model accuracy and convergence issues, indicated by precision and recall scores close to random guessing. However, when the learning rate was adjusted to a more appropriate value (e.g., 0.01), the model performance improved dramatically, underscoring the importance of a balanced learning rate for stable training dynamics. When learning rate is set to 0.01, the precision and recall scores are 0.9811.

Model Performance Across Different Configurations:

It shows a stark contrast in model performance between experiments with a learning rate of 0.1 and 0.01. With a learning rate of 0.01, the model achieves high accuracy, precision, and recall across different batch sizes and epochs. This indicates that a lower learning rate allows the optimizer to make finer adjustments to the model weights, leading to better generalization on the validation set.

Precision and Recall Analysis:

In scenarios with a high learning rate of 0.1, the model failed to predict any samples correctly, as indicated by precision and recall scores close to 0. This is a clear indicator of model divergence, where the optimizer's step size is too large to find a meaningful direction for weight updates. Conversely, with a lower learning rate of 0.01, the model's precision and recall scores significantly improve, showcasing effective learning and generalization. This reflects the optimizer's ability to fine-tune the model weights towards optimizing the loss function more efficiently.

Optimization Challenges: The experiments highlight the challenges in optimizer configuration, particularly in selecting an appropriate learning rate. An optimally chosen learning rate, coupled with a suitable batch size and epoch number, can lead to vastly improved model performance, as seen with the adam optimizer at a learning rate of 0.01.

Experimenting without any CNN

```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.metrics import confusion_matrix, precision_score,
recall_score

model = tf.keras.models.Sequential([
    tf.keras.layers.Flatten(input_shape = (28,28,1)),
    tf.keras.layers.Dense(128, activation = "relu", kernel_initializer =
    "he_uniform"),
    tf.keras.layers.Dense(10, activation = "softmax")
])

opt = SGD(learning_rate = 0.01, momentum = 0.9)
```

```

model.compile(optimizer = opt, loss = "categorical_crossentropy",
metrics = ["accuracy"])

history = model.fit(train_norm, y_train, epochs= 10,
validation_split=0.1, verbose=2)

Epoch 1/10
1688/1688 - 4s - loss: 0.3210 - accuracy: 0.9074 - val_loss: 0.1612 -
val_accuracy: 0.9552 - 4s/epoch - 2ms/step
Epoch 2/10
1688/1688 - 3s - loss: 0.1544 - accuracy: 0.9542 - val_loss: 0.1080 -
val_accuracy: 0.9700 - 3s/epoch - 2ms/step
Epoch 3/10
1688/1688 - 3s - loss: 0.1106 - accuracy: 0.9677 - val_loss: 0.0941 -
val_accuracy: 0.9745 - 3s/epoch - 2ms/step
Epoch 4/10
1688/1688 - 3s - loss: 0.0860 - accuracy: 0.9751 - val_loss: 0.0857 -
val_accuracy: 0.9763 - 3s/epoch - 2ms/step
Epoch 5/10
1688/1688 - 3s - loss: 0.0696 - accuracy: 0.9804 - val_loss: 0.0793 -
val_accuracy: 0.9792 - 3s/epoch - 2ms/step
Epoch 6/10
1688/1688 - 3s - loss: 0.0585 - accuracy: 0.9827 - val_loss: 0.0751 -
val_accuracy: 0.9773 - 3s/epoch - 2ms/step
Epoch 7/10
1688/1688 - 3s - loss: 0.0496 - accuracy: 0.9863 - val_loss: 0.0743 -
val_accuracy: 0.9783 - 3s/epoch - 2ms/step
Epoch 8/10
1688/1688 - 3s - loss: 0.0423 - accuracy: 0.9878 - val_loss: 0.0889 -
val_accuracy: 0.9750 - 3s/epoch - 2ms/step
Epoch 9/10
1688/1688 - 3s - loss: 0.0374 - accuracy: 0.9892 - val_loss: 0.0708 -
val_accuracy: 0.9802 - 3s/epoch - 2ms/step
Epoch 10/10
1688/1688 - 3s - loss: 0.0323 - accuracy: 0.9913 - val_loss: 0.0707 -
val_accuracy: 0.9803 - 3s/epoch - 2ms/step

## prediction

y_proba = model.predict(test_norm)

y_test_pred = np.argmax(y_proba, axis = 1)

y_true = np.argmax(y_test, axis = 1) if y_test.ndim > 1 else y_test

### confusion matrix

conf = confusion_matrix(y_true, y_test_pred)

```

```

#precision

precision = precision_score(y_true, y_test_pred, average = "weighted")

### recall
recall = recall_score(y_true, y_test_pred, average = "weighted")

### printing the matrix and plotting the confusion matrix

print("Confusion Matrix is ....")

print(conf)

print(f"Precision score:{precision: .4f}")
print(f"Recall score: {recall: .4f}")

plt.figure(figsize = (10,6))
sns.heatmap(conf, annot=True, fmt="g", cmap="Blues", cbar=False)
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.title(f"Confusion Matrix for the model without CNN Layers ")
plt.show()

313/313 [=====] - 0s 1ms/step
Confusion Matrix is ....
[[ 967    0    1    1    1    3    1    2    3    1]
 [  0 1124    2    1    0    2    2    1    3    0]
 [  6    2  999    3    4    0    2    8    8    0]
 [  0    0    4  983    0    8    0    4    6    5]
 [  1    0    6    1  952    0    2    2    1   17]
 [  2    0    0    3    1  878    4    0    2    2]
 [  4    3    1    1    4    3  936    1    4    1]
 [  0    5    9    3    0    0    0 1008    0    3]
 [  4    0    2    3    4    1    1    4  951    4]
 [  2    3    0    7    7    2    0    5    1  982]]
Precision score: 0.9780
Recall score:  0.9780

```

Confusion Matrix for the model without CNN Layers										
	0	1	2	3	4	5	6	7	8	9
0	967	0	1	1	1	3	1	2	3	1
1	0	1124	2	1	0	2	2	1	3	0
2	6	2	999	3	4	0	2	8	8	0
3	0	0	4	983	0	8	0	4	6	5
4	1	0	6	1	952	0	2	2	1	17
5	2	0	0	3	1	878	4	0	2	2
6	4	3	1	1	4	3	936	1	4	1
7	0	5	9	3	0	0	0	1008	0	3
8	4	0	2	3	4	1	1	4	951	4
9	2	3	0	7	7	2	0	5	1	982
0	1	2	3	4	5	6	7	8	9	9

```
test_loss, test_acc = model.evaluate(test_norm, y_test, verbose = 2)
print(f"Test Accuracy: {test_acc:.4f}")

313/313 - 0s - loss: 0.0729 - accuracy: 0.9780 - 434ms/epoch - 1ms/step
Test Accuracy: 0.9780
```

In this experiment, we focused on neural network architectures excluding the convolutional layers, and we concentrated only on the flatten and dense layers. The SGD (Stochastic Gradient Descent) algorithm was employed, configured with a learning rate of 0.01 and a momentum of 0.9. The evaluation of model accuracy was conducted through precision and recall metrics. Additionally, the confusion matrix was plotted to provide a visual representation of the model's performance across different classes. This approach allowed for an assessment of the impact of excluding convolutional layers on the model's ability to classify images accurately. The test accuracy is 0.9780 and the recall and precision scores are 0.9780.

Therefore, the performance of the neural network decreases when the convolutional layers are excluded from the network.

