# Assignment_7_NN

March 10, 2021

Assignment 7: Neural Networks using Keras and Tensorflow Please see the associated document for questions

If you have problems with Keras and Tensorflow on your local installation please make sure they are updated. On Google Colab this notebook runs.

**Group 6:**



| Name | Contribution |
|---|---|
| 1. Himanshu Chuphal (guschuhi@student.gu.se) | 12 H |
| 2. Claudio Aguilar Aguilar(claagu@student.chalmers.se) | 12 H |



---

[ ]:
```
# imports
from __future__ import print_function
import keras
from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K
import tensorflow as tf
from matplotlib import pyplot as plt
import numpy as np
```

[ ]:
```
# Hyper-parameters data-loading and formatting

batch_size = 128
num_classes = 10
epochs = 10

img_rows, img_cols = 28, 28

(x_train, lbl_train), (x_test, lbl_test) = mnist.load_data()

if K.image_data_format() == 'channels_first':
    x_train = x_train.reshape(x_train.shape[0], 1, img_rows, img_cols)
    x_test = x_test.reshape(x_test.shape[0], 1, img_rows, img_cols)
```

```
    input_shape = (1, img_rows, img_cols)
else:
    x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
    x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
    input_shape = (img_rows, img_cols, 1)

print('Train: X=%s, y=%s' % (x_train.shape, lbl_train.shape))
print('Test: X=%s, y=%s' % (x_test.shape, lbl_test.shape))
```

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-
datasets/mnist.npz
11493376/11490434 [==============================] - 0s 0us/step
Train: X=(60000, 28, 28, 1), y=(60000,)
Test: X=(10000, 28, 28, 1), y=(10000,)
```

**Preprocessing**

[ ]:
```
x_train = x_train.astype('float32')
x_test = x_test.astype('float32')

x_train /= 255
x_test /= 255

y_train = keras.utils.to_categorical(lbl_train, num_classes)
y_test = keras.utils.to_categorical(lbl_test, num_classes)
```

**1) Preprocessing.In the notebook, the data is downloaded from an external server imported into the notebook environment using the mnist.load_data()function call. Explain the data pre-processing high-lighted in the notebook.**

[**Answer**] After loading dataset into varaibles using mnist.load_data() from the previous cell, which gives about 60000 examples in the training dataset and 10000 in the test dataset and the images are square with 28×28pixel.

Since the pixel values for each image in the dataset are unsigned integers in the range between black and white, or 0 and 255 and there is no known best way to scale the pixel values for modeling, but some scaling is required.

Therefore, we normalize the pixel values of grayscale images, e.g. rescale them to the range [0,1], which is done by converting the data_type from unsigned integers to floats, then dividing the pixel values by the maximum value (255).

There are 10 classes (num_classes) and the classes are represented as unique integers. we use an one hot encoding for the class element of each sample, transforming the integer into a 10 element binary vector with a 1 for the index of the class value. We do this using to_categorical() utility function.

[ ]:
```
## Define model ##
model = Sequential()
```

```python
model.add(Flatten())
model.add(Dense(64, activation = 'relu'))
model.add(Dense(64, activation = 'relu'))
model.add(Dense(num_classes, activation='softmax'))


model.compile(loss=keras.losses.categorical_crossentropy,
               optimizer=keras.optimizers.SGD(lr = 0.1),
        metrics=['accuracy'],)

fit_info = model.fit(x_train, y_train,
           batch_size=batch_size,
           epochs=epochs,
           verbose=1,
           validation_data=(x_test, y_test))

score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss: {}, Test accuracy {}'.format(score[0], score[1]))
```

```
Epoch 1/10
469/469 [==============================] - 2s 4ms/step - loss: 0.8092 -
accuracy: 0.7631 - val_loss: 0.2581 - val_accuracy: 0.9245
Epoch 2/10
469/469 [==============================] - 1s 3ms/step - loss: 0.2450 -
accuracy: 0.9289 - val_loss: 0.1932 - val_accuracy: 0.9427
Epoch 3/10
469/469 [==============================] - 1s 3ms/step - loss: 0.1923 -
accuracy: 0.9443 - val_loss: 0.1658 - val_accuracy: 0.9504
Epoch 4/10
469/469 [==============================] - 1s 3ms/step - loss: 0.1569 -
accuracy: 0.9548 - val_loss: 0.1373 - val_accuracy: 0.9589
Epoch 5/10
469/469 [==============================] - 1s 3ms/step - loss: 0.1312 -
accuracy: 0.9630 - val_loss: 0.1291 - val_accuracy: 0.9618
Epoch 6/10
469/469 [==============================] - 1s 3ms/step - loss: 0.1152 -
accuracy: 0.9662 - val_loss: 0.1176 - val_accuracy: 0.9642
Epoch 7/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0966 -
accuracy: 0.9717 - val_loss: 0.1129 - val_accuracy: 0.9672
Epoch 8/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0892 -
accuracy: 0.9728 - val_loss: 0.1068 - val_accuracy: 0.9663
Epoch 9/10
469/469 [==============================] - 1s 3ms/step - loss: 0.0777 -
accuracy: 0.9779 - val_loss: 0.0980 - val_accuracy: 0.9687
Epoch 10/10
```

```
469/469 [==============================] - 1s 3ms/step - loss: 0.0749 -
accuracy: 0.9782 - val_loss: 0.0945 - val_accuracy: 0.9698
Test loss: 0.09454242885112762, Test accuracy 0.9697999954223633
```

[ ]:
```
# get the summary
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
flatten (Flatten)            (None, 784)               0
_________________________________________________________________
dense (Dense)                (None, 64)                50240
_________________________________________________________________
dense_1 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_2 (Dense)              (None, 10)                650
=================================================================
Total params: 55,050
Trainable params: 55,050
Non-trainable params: 0
_________________________________________________________________
```

#2)4points. Networkmodel, training, and changing hyper-parameters.

A) How many layers does the network in the notebook have? How many neurons does each layer have? What activation functions and why are these appropriate for this application? What is thetotal number of parameters for the network?Why does the input and output layers have the dimensions they have?

[**Answer**] - Network Layers and neurons in each layer (in bold)

Layer (type) :: Output Shape :: Param

**Input** :flatten (Flatten) :: (None, **784**) :: 0

---

**1st:** dense (Dense) :: (None, **64**) :: 50240

---

**2nd:** dense_1 (Dense) :: (None, **64**) :: 4160

---

**3rd:** dense_2 (Dense) :: (None, **10**) :: 650

- Activation functions - There are 2 uses here :

1.) *ReLU* - The rectified linear activation function or ReLU for short is a piecewise linear function that will output the input directly if it is positive, otherwise, it will output zero. It has become the default activation function for many types of neural networks because a model that uses it is easier to train and often achieves better performance. ReLu returns the element-wise maximum of (0, x), x is the input tensor. The advantage of using this function is that the neurons those get a negative value don't get activated and their output is not considered in the next layer.

2.) *Softmax*- It is alogistic regression function that normalizes values to be between (0,1) and is needed to convert the values to probabilities.

- Total Number of params :: **Total params: 55,050** For each neuron layer : we can calculate the total number of params using ((neurons*previous layer neurons)+ 1* neurons) The input_layer:: 0

First dense_layer:: 64 * 784 + 1 * 64 = 50176

2nd dense_layer_1:: 64 * 64 +64 = 4160

3rd dense_layer_2:: 10 * 64 + 64 = 704

Total :: 50176 + 4160 + 704 = 55,050 (Total params: 55,050 andTrainable params: 55,050)

- *Why does the input and output layers have the dimensions they have?*

The mnist images are square with big 28×28 pixel. The first flatten (Flatten) layer with (784, 1) neurons probably flattens the big 28×28 squares to a 1D array.

As there are a total of 10 classes, thereby we have 10 neurons in the output layer (as is explained before).

**B) What loss-function is used to train the network? What is the functional form(mathematical expression) of the loss function? and how should we interpret it?Why is it appropriate for the problem at hand?**

[**Answer**]

The loss-function used to train the network is the **categorical crossentropy loss** from the keras API. The functional form (mathematical expression) is: Loss $= -\sum_{i=1}^{outputsize} y_i \cdot log\hat{y}_i$

Where $\hat{y}_i$ is the i-th scalar value in the model output, $y_i$ the target value and output size the number of scalar values in the model output.

This loss function measures the distinguishability between two discrete probability distributions. $y_i$ is the probability that event i occurs and the sum of all $y_i$ is 1 which means that only one event may occur. A minus sign is present to ensure that the loss gets reduced when the distributions get closer to each other.

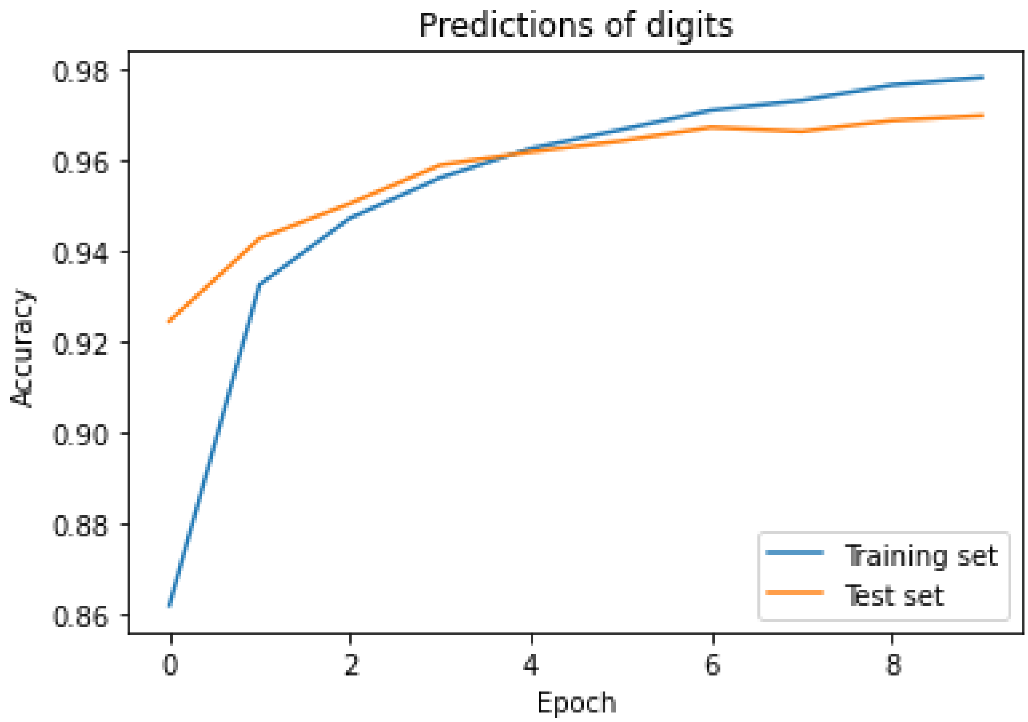
The loss function considers only one example belonging to a specific category with probability 1 and the others 0.

Which is well suited for our problem at hand because we want to learn the model to give a high probability to the correct digit and a low to the other digits.

**C) Train the network for 10 epochs and plot the training and validation accuracy for each epoch.**

[ ]:
```
plt.plot(fit_info.history['accuracy'])
plt.plot(fit_info.history['val_accuracy'])
plt.title('Predictions of digits')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
```

```python
plt.legend(['Training set', 'Test set'], loc='lower right')
plt.show()
```




Here we can see that the model is a little bit off but does quite a good job on predicting the values. In this example we only run for 40 epochs, training the model for more epochs can make the training set more accurate to the test set or also make the model overfit/underfit which is when a validation set comes in handy because then we will know when to stop training, which is when performance on the validation data stops improving.

**D)Update model to implement a three-layer neural network where the hidden-layers has 500 and 300 hidden units respectively. Train for 40epochs. What is the best validation accuracy you can achieve? –Geoff Hinton (a co-pioneer of Deep learning) claimed this network could reachavalidation accuracy of 0.9847 (http://yann.lecun.com/exdb/mnist/) using weight decay(L2 regularization of weights (kernels): https://keras.io/api/layers/regularizers/). Implement weight decay on hidden unitsand train and select 5 regularization factors from 0.000001 to 0.001. Train 3 replicates networks for each regularization factor. Plot the final validation accuracy with standard deviation (computed from the replicates) as a function of the regularization factor. How close do you get to Hintons result? –If you do not get the same results, what factors may influence this? (hint: What information is not given by Hinton on the MNIST database that may influence Model training)**

**[Answer]**

[ ]:
```python
def update_model(reg_factor, epochs):
  #make network wider
  model2 = Sequential()
  model2.add(Flatten())
  model2.add(Dense(500, activation='relu', activity_regularizer=keras.
 ↪regularizers.l2(reg_factor)))
  # 500 hidden units + l2 weight decay on the regularization factors from 0.
 ↪000001 - 0.001
  model2.add(Dense(300, activation='relu', activity_regularizer=keras.
 ↪regularizers.l2(reg_factor)))
  # 300 hidden units + l2 weight decay on the regularization factors from 0.
 ↪000001 - 0.001
  model2.add(Dense(num_classes, activation='softmax'))
  # convert vector of numbers into vector of probabilites

  #define the model
  model2.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.SGD(lr = 0.1),
              metrics=['accuracy'])
  #train the model
  fit_info2 = model2.fit(x_train, y_train,
                       batch_size=batch_size,
                       epochs=epochs,
                       verbose=1,
                       validation_data=(x_test, y_test))

  return fit_info2.history['val_accuracy'][-1]
  # return the last element which is the best validation accuracy we get.

val_accuracys = []
reg_factors = [0.001, 0.0001, 0.00005, 0.00001, 0.000001]
#[0.000001, 0.00005, 0.00001, 0.0001, 0.001] #5 regularization factors from 0.
 ↪000001 to 0.001

#5 regularization factors from 0.000001 to 0.001 and 3 replicas
for reg_f in reg_factors:
  temp=[]
  for rep_network in range(0,3):
    temp.append(update_model(reg_factor=reg_f, epochs=40))
  val_accuracys.append(temp)
```

```
Epoch 1/40
469/469 [==============================] - 6s 12ms/step - loss: 0.7652 -
accuracy: 0.8335 - val_loss: 0.2776 - val_accuracy: 0.9414
Epoch 2/40
469/469 [==============================] - 5s 11ms/step - loss: 0.2550 -
accuracy: 0.9485 - val_loss: 0.1974 - val_accuracy: 0.9605
```

```
Epoch 3/40
469/469 [==============================] - 5s 11ms/step - loss: 0.1831 -
accuracy: 0.9650 - val_loss: 0.1614 - val_accuracy: 0.9682
Epoch 4/40
469/469 [==============================] - 5s 12ms/step - loss: 0.1468 -
accuracy: 0.9734 - val_loss: 0.1420 - val_accuracy: 0.9721
Epoch 5/40
469/469 [==============================] - 5s 12ms/step - loss: 0.1244 -
accuracy: 0.9784 - val_loss: 0.1313 - val_accuracy: 0.9741
Epoch 6/40
469/469 [==============================] - 5s 11ms/step - loss: 0.1058 -
accuracy: 0.9831 - val_loss: 0.1169 - val_accuracy: 0.9769
Epoch 7/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0916 -
accuracy: 0.9870 - val_loss: 0.1104 - val_accuracy: 0.9786
Epoch 8/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0834 -
accuracy: 0.9884 - val_loss: 0.1032 - val_accuracy: 0.9797
Epoch 9/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0743 -
accuracy: 0.9915 - val_loss: 0.0971 - val_accuracy: 0.9819
Epoch 10/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0654 -
accuracy: 0.9932 - val_loss: 0.0943 - val_accuracy: 0.9815
Epoch 11/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0597 -
accuracy: 0.9949 - val_loss: 0.0927 - val_accuracy: 0.9807
Epoch 12/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0540 -
accuracy: 0.9965 - val_loss: 0.0883 - val_accuracy: 0.9822
Epoch 13/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0501 -
accuracy: 0.9971 - val_loss: 0.0875 - val_accuracy: 0.9819
Epoch 14/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0470 -
accuracy: 0.9978 - val_loss: 0.0844 - val_accuracy: 0.9834
Epoch 15/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0436 -
accuracy: 0.9983 - val_loss: 0.0827 - val_accuracy: 0.9821
Epoch 16/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0404 -
accuracy: 0.9988 - val_loss: 0.0803 - val_accuracy: 0.9837
Epoch 17/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0384 -
accuracy: 0.9989 - val_loss: 0.0804 - val_accuracy: 0.9827
Epoch 18/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0362 -
accuracy: 0.9995 - val_loss: 0.0784 - val_accuracy: 0.9841
```

```
Epoch 19/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0339 -
accuracy: 0.9996 - val_loss: 0.0782 - val_accuracy: 0.9836
Epoch 20/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0324 -
accuracy: 0.9996 - val_loss: 0.0781 - val_accuracy: 0.9841
Epoch 21/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0311 -
accuracy: 0.9997 - val_loss: 0.0744 - val_accuracy: 0.9839
Epoch 22/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0300 -
accuracy: 0.9997 - val_loss: 0.0746 - val_accuracy: 0.9833
Epoch 23/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0286 -
accuracy: 0.9999 - val_loss: 0.0741 - val_accuracy: 0.9844
Epoch 24/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0278 -
accuracy: 0.9998 - val_loss: 0.0734 - val_accuracy: 0.9839
Epoch 25/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0265 -
accuracy: 0.9999 - val_loss: 0.0727 - val_accuracy: 0.9843
Epoch 26/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0258 -
accuracy: 1.0000 - val_loss: 0.0721 - val_accuracy: 0.9838
Epoch 27/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0247 -
accuracy: 0.9999 - val_loss: 0.0729 - val_accuracy: 0.9836
Epoch 28/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0241 -
accuracy: 1.0000 - val_loss: 0.0707 - val_accuracy: 0.9846
Epoch 29/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0235 -
accuracy: 1.0000 - val_loss: 0.0716 - val_accuracy: 0.9847
Epoch 30/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0229 -
accuracy: 0.9999 - val_loss: 0.0705 - val_accuracy: 0.9842
Epoch 31/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0222 -
accuracy: 0.9999 - val_loss: 0.0696 - val_accuracy: 0.9852
Epoch 32/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0218 -
accuracy: 1.0000 - val_loss: 0.0699 - val_accuracy: 0.9842
Epoch 33/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0212 -
accuracy: 1.0000 - val_loss: 0.0690 - val_accuracy: 0.9844
Epoch 34/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0209 -
accuracy: 1.0000 - val_loss: 0.0686 - val_accuracy: 0.9842
```

```
Epoch 35/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0203 -
accuracy: 1.0000 - val_loss: 0.0682 - val_accuracy: 0.9850
Epoch 36/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0199 -
accuracy: 1.0000 - val_loss: 0.0685 - val_accuracy: 0.9845
Epoch 37/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0196 -
accuracy: 1.0000 - val_loss: 0.0676 - val_accuracy: 0.9844
Epoch 38/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0192 -
accuracy: 1.0000 - val_loss: 0.0677 - val_accuracy: 0.9843
Epoch 39/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0189 -
accuracy: 1.0000 - val_loss: 0.0669 - val_accuracy: 0.9842
Epoch 40/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0184 -
accuracy: 1.0000 - val_loss: 0.0667 - val_accuracy: 0.9842
Epoch 1/40
469/469 [==============================] - 6s 12ms/step - loss: 0.7834 -
accuracy: 0.8321 - val_loss: 0.2770 - val_accuracy: 0.9455
Epoch 2/40
469/469 [==============================] - 5s 11ms/step - loss: 0.2579 -
accuracy: 0.9473 - val_loss: 0.1999 - val_accuracy: 0.9598
Epoch 3/40
469/469 [==============================] - 5s 11ms/step - loss: 0.1820 -
accuracy: 0.9646 - val_loss: 0.1621 - val_accuracy: 0.9682
Epoch 4/40
469/469 [==============================] - 5s 12ms/step - loss: 0.1480 -
accuracy: 0.9733 - val_loss: 0.1427 - val_accuracy: 0.9732
Epoch 5/40
469/469 [==============================] - 5s 11ms/step - loss: 0.1217 -
accuracy: 0.9791 - val_loss: 0.1268 - val_accuracy: 0.9761
Epoch 6/40
469/469 [==============================] - 5s 11ms/step - loss: 0.1043 -
accuracy: 0.9843 - val_loss: 0.1172 - val_accuracy: 0.9775
Epoch 7/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0891 -
accuracy: 0.9884 - val_loss: 0.1108 - val_accuracy: 0.9780
Epoch 8/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0800 -
accuracy: 0.9904 - val_loss: 0.1027 - val_accuracy: 0.9793
Epoch 9/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0714 -
accuracy: 0.9919 - val_loss: 0.0987 - val_accuracy: 0.9806
Epoch 10/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0645 -
accuracy: 0.9940 - val_loss: 0.0955 - val_accuracy: 0.9798
```

```
Epoch 11/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0588 -
accuracy: 0.9952 - val_loss: 0.0918 - val_accuracy: 0.9811
Epoch 12/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0539 -
accuracy: 0.9963 - val_loss: 0.0904 - val_accuracy: 0.9804
Epoch 13/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0488 -
accuracy: 0.9976 - val_loss: 0.0890 - val_accuracy: 0.9817
Epoch 14/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0461 -
accuracy: 0.9979 - val_loss: 0.0859 - val_accuracy: 0.9819
Epoch 15/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0430 -
accuracy: 0.9984 - val_loss: 0.0833 - val_accuracy: 0.9830
Epoch 16/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0396 -
accuracy: 0.9990 - val_loss: 0.0830 - val_accuracy: 0.9823
Epoch 17/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0371 -
accuracy: 0.9994 - val_loss: 0.0808 - val_accuracy: 0.9827
Epoch 18/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0353 -
accuracy: 0.9995 - val_loss: 0.0808 - val_accuracy: 0.9824
Epoch 19/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0341 -
accuracy: 0.9996 - val_loss: 0.0794 - val_accuracy: 0.9826
Epoch 20/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0321 -
accuracy: 0.9998 - val_loss: 0.0784 - val_accuracy: 0.9824
Epoch 21/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0307 -
accuracy: 0.9998 - val_loss: 0.0784 - val_accuracy: 0.9825
Epoch 22/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0295 -
accuracy: 0.9998 - val_loss: 0.0772 - val_accuracy: 0.9818
Epoch 23/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0286 -
accuracy: 0.9999 - val_loss: 0.0778 - val_accuracy: 0.9820
Epoch 24/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0275 -
accuracy: 0.9999 - val_loss: 0.0759 - val_accuracy: 0.9829
Epoch 25/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0261 -
accuracy: 1.0000 - val_loss: 0.0762 - val_accuracy: 0.9818
Epoch 26/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0255 -
accuracy: 0.9999 - val_loss: 0.0749 - val_accuracy: 0.9830
```

```
Epoch 27/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0246 -
accuracy: 1.0000 - val_loss: 0.0746 - val_accuracy: 0.9823
Epoch 28/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0241 -
accuracy: 1.0000 - val_loss: 0.0744 - val_accuracy: 0.9825
Epoch 29/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0233 -
accuracy: 1.0000 - val_loss: 0.0739 - val_accuracy: 0.9821
Epoch 30/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0228 -
accuracy: 1.0000 - val_loss: 0.0727 - val_accuracy: 0.9823
Epoch 31/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0222 -
accuracy: 1.0000 - val_loss: 0.0729 - val_accuracy: 0.9826
Epoch 32/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0216 -
accuracy: 1.0000 - val_loss: 0.0724 - val_accuracy: 0.9821
Epoch 33/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0212 -
accuracy: 1.0000 - val_loss: 0.0721 - val_accuracy: 0.9822
Epoch 34/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0207 -
accuracy: 1.0000 - val_loss: 0.0718 - val_accuracy: 0.9820
Epoch 35/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0204 -
accuracy: 1.0000 - val_loss: 0.0711 - val_accuracy: 0.9823
Epoch 36/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0199 -
accuracy: 1.0000 - val_loss: 0.0714 - val_accuracy: 0.9823
Epoch 37/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0195 -
accuracy: 1.0000 - val_loss: 0.0711 - val_accuracy: 0.9826
Epoch 38/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0192 -
accuracy: 1.0000 - val_loss: 0.0710 - val_accuracy: 0.9822
Epoch 39/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0188 -
accuracy: 1.0000 - val_loss: 0.0703 - val_accuracy: 0.9826
Epoch 40/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0184 -
accuracy: 1.0000 - val_loss: 0.0699 - val_accuracy: 0.9825
Epoch 1/40
469/469 [==============================] - 6s 12ms/step - loss: 0.7910 -
accuracy: 0.8232 - val_loss: 0.2805 - val_accuracy: 0.9424
Epoch 2/40
469/469 [==============================] - 5s 12ms/step - loss: 0.2557 -
accuracy: 0.9484 - val_loss: 0.1954 - val_accuracy: 0.9603
```

```
Epoch 3/40
469/469 [==============================] - 5s 12ms/step - loss: 0.1814 -
accuracy: 0.9647 - val_loss: 0.1642 - val_accuracy: 0.9661
Epoch 4/40
469/469 [==============================] - 5s 11ms/step - loss: 0.1441 -
accuracy: 0.9755 - val_loss: 0.1415 - val_accuracy: 0.9723
Epoch 5/40
469/469 [==============================] - 5s 12ms/step - loss: 0.1214 -
accuracy: 0.9803 - val_loss: 0.1281 - val_accuracy: 0.9752
Epoch 6/40
469/469 [==============================] - 5s 12ms/step - loss: 0.1038 -
accuracy: 0.9839 - val_loss: 0.1167 - val_accuracy: 0.9776
Epoch 7/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0913 -
accuracy: 0.9875 - val_loss: 0.1142 - val_accuracy: 0.9778
Epoch 8/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0799 -
accuracy: 0.9902 - val_loss: 0.1019 - val_accuracy: 0.9798
Epoch 9/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0718 -
accuracy: 0.9927 - val_loss: 0.0993 - val_accuracy: 0.9800
Epoch 10/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0648 -
accuracy: 0.9942 - val_loss: 0.0958 - val_accuracy: 0.9805
Epoch 11/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0582 -
accuracy: 0.9957 - val_loss: 0.0925 - val_accuracy: 0.9804
Epoch 12/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0534 -
accuracy: 0.9967 - val_loss: 0.0899 - val_accuracy: 0.9820
Epoch 13/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0487 -
accuracy: 0.9976 - val_loss: 0.0866 - val_accuracy: 0.9810
Epoch 14/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0458 -
accuracy: 0.9981 - val_loss: 0.0857 - val_accuracy: 0.9826
Epoch 15/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0423 -
accuracy: 0.9987 - val_loss: 0.0847 - val_accuracy: 0.9822
Epoch 16/40
469/469 [==============================] - 7s 14ms/step - loss: 0.0394 -
accuracy: 0.9990 - val_loss: 0.0846 - val_accuracy: 0.9816
Epoch 17/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0371 -
accuracy: 0.9994 - val_loss: 0.0811 - val_accuracy: 0.9826
Epoch 18/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0355 -
accuracy: 0.9994 - val_loss: 0.0805 - val_accuracy: 0.9825
```

```
Epoch 19/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0336 -
accuracy: 0.9996 - val_loss: 0.0789 - val_accuracy: 0.9833
Epoch 20/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0324 -
accuracy: 0.9996 - val_loss: 0.0808 - val_accuracy: 0.9821
Epoch 21/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0309 -
accuracy: 0.9998 - val_loss: 0.0773 - val_accuracy: 0.9832
Epoch 22/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0292 -
accuracy: 0.9999 - val_loss: 0.0770 - val_accuracy: 0.9832
Epoch 23/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0285 -
accuracy: 0.9998 - val_loss: 0.0761 - val_accuracy: 0.9829
Epoch 24/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0272 -
accuracy: 1.0000 - val_loss: 0.0762 - val_accuracy: 0.9832
Epoch 25/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0261 -
accuracy: 1.0000 - val_loss: 0.0751 - val_accuracy: 0.9832
Epoch 26/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0257 -
accuracy: 0.9999 - val_loss: 0.0750 - val_accuracy: 0.9833
Epoch 27/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0248 -
accuracy: 0.9999 - val_loss: 0.0748 - val_accuracy: 0.9829
Epoch 28/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0239 -
accuracy: 1.0000 - val_loss: 0.0740 - val_accuracy: 0.9839
Epoch 29/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0236 -
accuracy: 1.0000 - val_loss: 0.0734 - val_accuracy: 0.9828
Epoch 30/40
469/469 [==============================] - 5s 11ms/step - loss: 0.0228 -
accuracy: 1.0000 - val_loss: 0.0737 - val_accuracy: 0.9833
Epoch 31/40
469/469 [==============================] - 5s 12ms/step - loss: 0.0221 -
accuracy: 1.0000 - val_loss: 0.0725 - val_accuracy: 0.9835
Epoch 32/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0217 -
accuracy: 1.0000 - val_loss: 0.0728 - val_accuracy: 0.9832
Epoch 33/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0212 -
accuracy: 1.0000 - val_loss: 0.0718 - val_accuracy: 0.9834
Epoch 34/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0208 -
accuracy: 1.0000 - val_loss: 0.0715 - val_accuracy: 0.9835
```

```
Epoch 35/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0203 -
accuracy: 1.0000 - val_loss: 0.0723 - val_accuracy: 0.9833
Epoch 36/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0199 -
accuracy: 1.0000 - val_loss: 0.0716 - val_accuracy: 0.9832
Epoch 37/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0194 -
accuracy: 1.0000 - val_loss: 0.0708 - val_accuracy: 0.9837
Epoch 38/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0191 -
accuracy: 1.0000 - val_loss: 0.0708 - val_accuracy: 0.9832
Epoch 39/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0188 -
accuracy: 1.0000 - val_loss: 0.0705 - val_accuracy: 0.9837
Epoch 40/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0184 -
accuracy: 1.0000 - val_loss: 0.0707 - val_accuracy: 0.9829
Epoch 1/40
469/469 [==============================] - 6s 13ms/step - loss: 0.7085 -
accuracy: 0.8134 - val_loss: 0.2311 - val_accuracy: 0.9384
Epoch 2/40
469/469 [==============================] - 6s 12ms/step - loss: 0.2171 -
accuracy: 0.9423 - val_loss: 0.1672 - val_accuracy: 0.9573
Epoch 3/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1552 -
accuracy: 0.9599 - val_loss: 0.1348 - val_accuracy: 0.9654
Epoch 4/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1222 -
accuracy: 0.9696 - val_loss: 0.1167 - val_accuracy: 0.9691
Epoch 5/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0999 -
accuracy: 0.9755 - val_loss: 0.1037 - val_accuracy: 0.9722
Epoch 6/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0822 -
accuracy: 0.9812 - val_loss: 0.0926 - val_accuracy: 0.9754
Epoch 7/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0724 -
accuracy: 0.9841 - val_loss: 0.0872 - val_accuracy: 0.9772
Epoch 8/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0608 -
accuracy: 0.9875 - val_loss: 0.0830 - val_accuracy: 0.9779
Epoch 9/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0545 -
accuracy: 0.9889 - val_loss: 0.0780 - val_accuracy: 0.9796
Epoch 10/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0475 -
accuracy: 0.9917 - val_loss: 0.0750 - val_accuracy: 0.9804
```

```
Epoch 11/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0422 -
accuracy: 0.9931 - val_loss: 0.0738 - val_accuracy: 0.9803
Epoch 12/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0370 -
accuracy: 0.9943 - val_loss: 0.0704 - val_accuracy: 0.9813
Epoch 13/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0333 -
accuracy: 0.9959 - val_loss: 0.0710 - val_accuracy: 0.9814
Epoch 14/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0305 -
accuracy: 0.9967 - val_loss: 0.0687 - val_accuracy: 0.9818
Epoch 15/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0283 -
accuracy: 0.9978 - val_loss: 0.0665 - val_accuracy: 0.9825
Epoch 16/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0252 -
accuracy: 0.9981 - val_loss: 0.0669 - val_accuracy: 0.9816
Epoch 17/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0240 -
accuracy: 0.9985 - val_loss: 0.0672 - val_accuracy: 0.9827
Epoch 18/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0215 -
accuracy: 0.9989 - val_loss: 0.0638 - val_accuracy: 0.9823
Epoch 19/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0206 -
accuracy: 0.9991 - val_loss: 0.0651 - val_accuracy: 0.9824
Epoch 20/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0188 -
accuracy: 0.9995 - val_loss: 0.0631 - val_accuracy: 0.9827
Epoch 21/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0179 -
accuracy: 0.9995 - val_loss: 0.0644 - val_accuracy: 0.9829
Epoch 22/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0172 -
accuracy: 0.9996 - val_loss: 0.0628 - val_accuracy: 0.9833
Epoch 23/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0165 -
accuracy: 0.9995 - val_loss: 0.0623 - val_accuracy: 0.9830
Epoch 24/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0156 -
accuracy: 0.9998 - val_loss: 0.0628 - val_accuracy: 0.9833
Epoch 25/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0147 -
accuracy: 0.9999 - val_loss: 0.0624 - val_accuracy: 0.9828
Epoch 26/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0145 -
accuracy: 0.9999 - val_loss: 0.0617 - val_accuracy: 0.9832
```

```
Epoch 27/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0138 -
accuracy: 0.9999 - val_loss: 0.0611 - val_accuracy: 0.9831
Epoch 28/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0133 -
accuracy: 0.9999 - val_loss: 0.0603 - val_accuracy: 0.9830
Epoch 29/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0127 -
accuracy: 0.9999 - val_loss: 0.0614 - val_accuracy: 0.9832
Epoch 30/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0123 -
accuracy: 1.0000 - val_loss: 0.0611 - val_accuracy: 0.9833
Epoch 31/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0119 -
accuracy: 1.0000 - val_loss: 0.0603 - val_accuracy: 0.9829
Epoch 32/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0116 -
accuracy: 1.0000 - val_loss: 0.0608 - val_accuracy: 0.9826
Epoch 33/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0114 -
accuracy: 1.0000 - val_loss: 0.0601 - val_accuracy: 0.9829
Epoch 34/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0109 -
accuracy: 1.0000 - val_loss: 0.0596 - val_accuracy: 0.9832
Epoch 35/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0108 -
accuracy: 1.0000 - val_loss: 0.0605 - val_accuracy: 0.9830
Epoch 36/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0105 -
accuracy: 1.0000 - val_loss: 0.0593 - val_accuracy: 0.9836
Epoch 37/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0103 -
accuracy: 1.0000 - val_loss: 0.0596 - val_accuracy: 0.9831
Epoch 38/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0100 -
accuracy: 1.0000 - val_loss: 0.0589 - val_accuracy: 0.9830
Epoch 39/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0099 -
accuracy: 1.0000 - val_loss: 0.0592 - val_accuracy: 0.9829
Epoch 40/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0096 -
accuracy: 1.0000 - val_loss: 0.0587 - val_accuracy: 0.9837
Epoch 1/40
469/469 [==============================] - 6s 12ms/step - loss: 0.7151 -
accuracy: 0.8115 - val_loss: 0.2396 - val_accuracy: 0.9330
Epoch 2/40
469/469 [==============================] - 6s 12ms/step - loss: 0.2193 -
accuracy: 0.9431 - val_loss: 0.1649 - val_accuracy: 0.9565
```

```
Epoch 3/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1543 -
accuracy: 0.9607 - val_loss: 0.1327 - val_accuracy: 0.9658
Epoch 4/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1194 -
accuracy: 0.9706 - val_loss: 0.1152 - val_accuracy: 0.9699
Epoch 5/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0975 -
accuracy: 0.9757 - val_loss: 0.1060 - val_accuracy: 0.9727
Epoch 6/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0840 -
accuracy: 0.9805 - val_loss: 0.0948 - val_accuracy: 0.9763
Epoch 7/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0706 -
accuracy: 0.9841 - val_loss: 0.0896 - val_accuracy: 0.9754
Epoch 8/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0614 -
accuracy: 0.9873 - val_loss: 0.0850 - val_accuracy: 0.9777
Epoch 9/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0536 -
accuracy: 0.9892 - val_loss: 0.0813 - val_accuracy: 0.9790
Epoch 10/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0487 -
accuracy: 0.9909 - val_loss: 0.0749 - val_accuracy: 0.9800
Epoch 11/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0428 -
accuracy: 0.9929 - val_loss: 0.0759 - val_accuracy: 0.9796
Epoch 12/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0372 -
accuracy: 0.9949 - val_loss: 0.0734 - val_accuracy: 0.9800
Epoch 13/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0341 -
accuracy: 0.9952 - val_loss: 0.0721 - val_accuracy: 0.9804
Epoch 14/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0299 -
accuracy: 0.9971 - val_loss: 0.0702 - val_accuracy: 0.9808
Epoch 15/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0286 -
accuracy: 0.9970 - val_loss: 0.0680 - val_accuracy: 0.9821
Epoch 16/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0258 -
accuracy: 0.9980 - val_loss: 0.0680 - val_accuracy: 0.9820
Epoch 17/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0242 -
accuracy: 0.9982 - val_loss: 0.0699 - val_accuracy: 0.9810
Epoch 18/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0221 -
accuracy: 0.9989 - val_loss: 0.0676 - val_accuracy: 0.9813
```

```
Epoch 19/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0206 -
accuracy: 0.9993 - val_loss: 0.0670 - val_accuracy: 0.9814
Epoch 20/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0187 -
accuracy: 0.9995 - val_loss: 0.0668 - val_accuracy: 0.9817
Epoch 21/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0179 -
accuracy: 0.9996 - val_loss: 0.0660 - val_accuracy: 0.9821
Epoch 22/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0172 -
accuracy: 0.9997 - val_loss: 0.0650 - val_accuracy: 0.9830
Epoch 23/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0164 -
accuracy: 0.9997 - val_loss: 0.0640 - val_accuracy: 0.9826
Epoch 24/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0155 -
accuracy: 0.9998 - val_loss: 0.0646 - val_accuracy: 0.9821
Epoch 25/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0149 -
accuracy: 0.9999 - val_loss: 0.0650 - val_accuracy: 0.9825
Epoch 26/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0142 -
accuracy: 0.9999 - val_loss: 0.0646 - val_accuracy: 0.9825
Epoch 27/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0135 -
accuracy: 0.9999 - val_loss: 0.0641 - val_accuracy: 0.9823
Epoch 28/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0131 -
accuracy: 1.0000 - val_loss: 0.0652 - val_accuracy: 0.9820
Epoch 29/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0129 -
accuracy: 1.0000 - val_loss: 0.0632 - val_accuracy: 0.9825
Epoch 30/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0122 -
accuracy: 0.9999 - val_loss: 0.0635 - val_accuracy: 0.9821
Epoch 31/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0120 -
accuracy: 0.9999 - val_loss: 0.0636 - val_accuracy: 0.9825
Epoch 32/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0116 -
accuracy: 0.9999 - val_loss: 0.0637 - val_accuracy: 0.9825
Epoch 33/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0113 -
accuracy: 1.0000 - val_loss: 0.0633 - val_accuracy: 0.9824
Epoch 34/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0110 -
accuracy: 1.0000 - val_loss: 0.0634 - val_accuracy: 0.9822
```

```
Epoch 35/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0108 -
accuracy: 1.0000 - val_loss: 0.0629 - val_accuracy: 0.9824
Epoch 36/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0105 -
accuracy: 1.0000 - val_loss: 0.0630 - val_accuracy: 0.9822
Epoch 37/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0103 -
accuracy: 1.0000 - val_loss: 0.0632 - val_accuracy: 0.9828
Epoch 38/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0101 -
accuracy: 1.0000 - val_loss: 0.0628 - val_accuracy: 0.9826
Epoch 39/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0098 -
accuracy: 1.0000 - val_loss: 0.0625 - val_accuracy: 0.9825
Epoch 40/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0096 -
accuracy: 1.0000 - val_loss: 0.0625 - val_accuracy: 0.9828
Epoch 1/40
469/469 [==============================] - 6s 13ms/step - loss: 0.7149 -
accuracy: 0.8163 - val_loss: 0.2383 - val_accuracy: 0.9371
Epoch 2/40
469/469 [==============================] - 6s 12ms/step - loss: 0.2190 -
accuracy: 0.9434 - val_loss: 0.1656 - val_accuracy: 0.9562
Epoch 3/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1572 -
accuracy: 0.9601 - val_loss: 0.1375 - val_accuracy: 0.9630
Epoch 4/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1231 -
accuracy: 0.9704 - val_loss: 0.1150 - val_accuracy: 0.9690
Epoch 5/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0994 -
accuracy: 0.9759 - val_loss: 0.1017 - val_accuracy: 0.9737
Epoch 6/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0849 -
accuracy: 0.9807 - val_loss: 0.0973 - val_accuracy: 0.9740
Epoch 7/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0726 -
accuracy: 0.9837 - val_loss: 0.0896 - val_accuracy: 0.9769
Epoch 8/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0623 -
accuracy: 0.9871 - val_loss: 0.0818 - val_accuracy: 0.9785
Epoch 9/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0542 -
accuracy: 0.9895 - val_loss: 0.0777 - val_accuracy: 0.9793
Epoch 10/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0499 -
accuracy: 0.9903 - val_loss: 0.0796 - val_accuracy: 0.9782
```

```
Epoch 11/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0427 -
accuracy: 0.9933 - val_loss: 0.0747 - val_accuracy: 0.9802
Epoch 12/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0385 -
accuracy: 0.9947 - val_loss: 0.0716 - val_accuracy: 0.9805
Epoch 13/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0335 -
accuracy: 0.9959 - val_loss: 0.0747 - val_accuracy: 0.9801
Epoch 14/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0315 -
accuracy: 0.9964 - val_loss: 0.0704 - val_accuracy: 0.9817
Epoch 15/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0280 -
accuracy: 0.9978 - val_loss: 0.0710 - val_accuracy: 0.9815
Epoch 16/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0260 -
accuracy: 0.9978 - val_loss: 0.0673 - val_accuracy: 0.9815
Epoch 17/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0232 -
accuracy: 0.9990 - val_loss: 0.0675 - val_accuracy: 0.9825
Epoch 18/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0224 -
accuracy: 0.9989 - val_loss: 0.0668 - val_accuracy: 0.9826
Epoch 19/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0208 -
accuracy: 0.9992 - val_loss: 0.0659 - val_accuracy: 0.9828
Epoch 20/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0191 -
accuracy: 0.9995 - val_loss: 0.0661 - val_accuracy: 0.9828
Epoch 21/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0183 -
accuracy: 0.9996 - val_loss: 0.0656 - val_accuracy: 0.9819
Epoch 22/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0172 -
accuracy: 0.9998 - val_loss: 0.0638 - val_accuracy: 0.9829
Epoch 23/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0166 -
accuracy: 0.9995 - val_loss: 0.0642 - val_accuracy: 0.9834
Epoch 24/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0157 -
accuracy: 0.9997 - val_loss: 0.0635 - val_accuracy: 0.9833
Epoch 25/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0148 -
accuracy: 0.9999 - val_loss: 0.0633 - val_accuracy: 0.9826
Epoch 26/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0144 -
accuracy: 0.9998 - val_loss: 0.0634 - val_accuracy: 0.9833
```

```
Epoch 27/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0139 -
accuracy: 0.9999 - val_loss: 0.0627 - val_accuracy: 0.9833
Epoch 28/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0133 -
accuracy: 1.0000 - val_loss: 0.0636 - val_accuracy: 0.9831
Epoch 29/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0129 -
accuracy: 1.0000 - val_loss: 0.0626 - val_accuracy: 0.9833
Epoch 30/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0125 -
accuracy: 0.9999 - val_loss: 0.0622 - val_accuracy: 0.9828
Epoch 31/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0121 -
accuracy: 1.0000 - val_loss: 0.0621 - val_accuracy: 0.9835
Epoch 32/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0118 -
accuracy: 0.9999 - val_loss: 0.0619 - val_accuracy: 0.9833
Epoch 33/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0115 -
accuracy: 1.0000 - val_loss: 0.0619 - val_accuracy: 0.9833
Epoch 34/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0111 -
accuracy: 1.0000 - val_loss: 0.0624 - val_accuracy: 0.9828
Epoch 35/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0109 -
accuracy: 1.0000 - val_loss: 0.0618 - val_accuracy: 0.9832
Epoch 36/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0107 -
accuracy: 1.0000 - val_loss: 0.0616 - val_accuracy: 0.9830
Epoch 37/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0105 -
accuracy: 1.0000 - val_loss: 0.0616 - val_accuracy: 0.9823
Epoch 38/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0101 -
accuracy: 1.0000 - val_loss: 0.0618 - val_accuracy: 0.9833
Epoch 39/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0100 -
accuracy: 1.0000 - val_loss: 0.0608 - val_accuracy: 0.9833
Epoch 40/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0098 -
accuracy: 1.0000 - val_loss: 0.0612 - val_accuracy: 0.9833
Epoch 1/40
469/469 [==============================] - 6s 13ms/step - loss: 0.6972 -
accuracy: 0.8161 - val_loss: 0.2785 - val_accuracy: 0.9197
Epoch 2/40
469/469 [==============================] - 6s 12ms/step - loss: 0.2165 -
accuracy: 0.9406 - val_loss: 0.1625 - val_accuracy: 0.9566
```

```
Epoch 3/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1480 -
accuracy: 0.9598 - val_loss: 0.1320 - val_accuracy: 0.9635
Epoch 4/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1151 -
accuracy: 0.9695 - val_loss: 0.1169 - val_accuracy: 0.9686
Epoch 5/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0982 -
accuracy: 0.9748 - val_loss: 0.1034 - val_accuracy: 0.9715
Epoch 6/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0811 -
accuracy: 0.9791 - val_loss: 0.0910 - val_accuracy: 0.9742
Epoch 7/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0682 -
accuracy: 0.9836 - val_loss: 0.0838 - val_accuracy: 0.9763
Epoch 8/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0602 -
accuracy: 0.9853 - val_loss: 0.0799 - val_accuracy: 0.9783
Epoch 9/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0516 -
accuracy: 0.9889 - val_loss: 0.0760 - val_accuracy: 0.9797
Epoch 10/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0456 -
accuracy: 0.9909 - val_loss: 0.0807 - val_accuracy: 0.9769
Epoch 11/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0397 -
accuracy: 0.9924 - val_loss: 0.0742 - val_accuracy: 0.9800
Epoch 12/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0364 -
accuracy: 0.9936 - val_loss: 0.0708 - val_accuracy: 0.9800
Epoch 13/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0327 -
accuracy: 0.9949 - val_loss: 0.0665 - val_accuracy: 0.9813
Epoch 14/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0294 -
accuracy: 0.9961 - val_loss: 0.0661 - val_accuracy: 0.9825
Epoch 15/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0257 -
accuracy: 0.9968 - val_loss: 0.0655 - val_accuracy: 0.9815
Epoch 16/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0238 -
accuracy: 0.9977 - val_loss: 0.0655 - val_accuracy: 0.9815
Epoch 17/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0210 -
accuracy: 0.9984 - val_loss: 0.0663 - val_accuracy: 0.9814
Epoch 18/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0197 -
accuracy: 0.9987 - val_loss: 0.0628 - val_accuracy: 0.9825
```

```
Epoch 19/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0184 -
accuracy: 0.9989 - val_loss: 0.0633 - val_accuracy: 0.9817
Epoch 20/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0169 -
accuracy: 0.9992 - val_loss: 0.0624 - val_accuracy: 0.9819
Epoch 21/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0163 -
accuracy: 0.9993 - val_loss: 0.0613 - val_accuracy: 0.9823
Epoch 22/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0147 -
accuracy: 0.9995 - val_loss: 0.0613 - val_accuracy: 0.9823
Epoch 23/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0140 -
accuracy: 0.9996 - val_loss: 0.0604 - val_accuracy: 0.9824
Epoch 24/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0133 -
accuracy: 0.9996 - val_loss: 0.0607 - val_accuracy: 0.9832
Epoch 25/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0127 -
accuracy: 0.9998 - val_loss: 0.0603 - val_accuracy: 0.9830
Epoch 26/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0120 -
accuracy: 0.9998 - val_loss: 0.0601 - val_accuracy: 0.9827
Epoch 27/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0114 -
accuracy: 0.9998 - val_loss: 0.0609 - val_accuracy: 0.9833
Epoch 28/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0111 -
accuracy: 0.9999 - val_loss: 0.0592 - val_accuracy: 0.9833
Epoch 29/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0108 -
accuracy: 1.0000 - val_loss: 0.0597 - val_accuracy: 0.9832
Epoch 30/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0102 -
accuracy: 0.9999 - val_loss: 0.0597 - val_accuracy: 0.9832
Epoch 31/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0098 -
accuracy: 1.0000 - val_loss: 0.0591 - val_accuracy: 0.9832
Epoch 32/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0097 -
accuracy: 1.0000 - val_loss: 0.0599 - val_accuracy: 0.9828
Epoch 33/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0093 -
accuracy: 1.0000 - val_loss: 0.0590 - val_accuracy: 0.9833
Epoch 34/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0093 -
accuracy: 1.0000 - val_loss: 0.0594 - val_accuracy: 0.9832
```

```
Epoch 35/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0088 -
accuracy: 1.0000 - val_loss: 0.0588 - val_accuracy: 0.9833
Epoch 36/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0086 -
accuracy: 1.0000 - val_loss: 0.0588 - val_accuracy: 0.9836
Epoch 37/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0084 -
accuracy: 1.0000 - val_loss: 0.0590 - val_accuracy: 0.9830
Epoch 38/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0082 -
accuracy: 1.0000 - val_loss: 0.0585 - val_accuracy: 0.9834
Epoch 39/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0081 -
accuracy: 1.0000 - val_loss: 0.0586 - val_accuracy: 0.9833
Epoch 40/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0079 -
accuracy: 1.0000 - val_loss: 0.0581 - val_accuracy: 0.9834
Epoch 1/40
469/469 [==============================] - 7s 13ms/step - loss: 0.6769 -
accuracy: 0.8220 - val_loss: 0.2277 - val_accuracy: 0.9389
Epoch 2/40
469/469 [==============================] - 6s 12ms/step - loss: 0.2091 -
accuracy: 0.9434 - val_loss: 0.1616 - val_accuracy: 0.9554
Epoch 3/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1529 -
accuracy: 0.9588 - val_loss: 0.1333 - val_accuracy: 0.9635
Epoch 4/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1156 -
accuracy: 0.9699 - val_loss: 0.1147 - val_accuracy: 0.9681
Epoch 5/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0968 -
accuracy: 0.9748 - val_loss: 0.1011 - val_accuracy: 0.9714
Epoch 6/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0806 -
accuracy: 0.9796 - val_loss: 0.0937 - val_accuracy: 0.9728
Epoch 7/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0704 -
accuracy: 0.9833 - val_loss: 0.0885 - val_accuracy: 0.9745
Epoch 8/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0604 -
accuracy: 0.9874 - val_loss: 0.0846 - val_accuracy: 0.9753
Epoch 9/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0543 -
accuracy: 0.9877 - val_loss: 0.0824 - val_accuracy: 0.9763
Epoch 10/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0460 -
accuracy: 0.9908 - val_loss: 0.0774 - val_accuracy: 0.9786
```

```
Epoch 11/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0426 -
accuracy: 0.9919 - val_loss: 0.0756 - val_accuracy: 0.9782
Epoch 12/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0367 -
accuracy: 0.9940 - val_loss: 0.0746 - val_accuracy: 0.9774
Epoch 13/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0328 -
accuracy: 0.9952 - val_loss: 0.0746 - val_accuracy: 0.9770
Epoch 14/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0300 -
accuracy: 0.9958 - val_loss: 0.0713 - val_accuracy: 0.9790
Epoch 15/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0272 -
accuracy: 0.9962 - val_loss: 0.0717 - val_accuracy: 0.9785
Epoch 16/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0245 -
accuracy: 0.9975 - val_loss: 0.0696 - val_accuracy: 0.9794
Epoch 17/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0220 -
accuracy: 0.9981 - val_loss: 0.0691 - val_accuracy: 0.9804
Epoch 18/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0207 -
accuracy: 0.9983 - val_loss: 0.0677 - val_accuracy: 0.9795
Epoch 19/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0181 -
accuracy: 0.9990 - val_loss: 0.0670 - val_accuracy: 0.9806
Epoch 20/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0173 -
accuracy: 0.9992 - val_loss: 0.0677 - val_accuracy: 0.9793
Epoch 21/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0163 -
accuracy: 0.9994 - val_loss: 0.0667 - val_accuracy: 0.9804
Epoch 22/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0154 -
accuracy: 0.9995 - val_loss: 0.0685 - val_accuracy: 0.9799
Epoch 23/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0145 -
accuracy: 0.9995 - val_loss: 0.0669 - val_accuracy: 0.9811
Epoch 24/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0136 -
accuracy: 0.9997 - val_loss: 0.0649 - val_accuracy: 0.9808
Epoch 25/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0128 -
accuracy: 0.9997 - val_loss: 0.0647 - val_accuracy: 0.9809
Epoch 26/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0123 -
accuracy: 0.9997 - val_loss: 0.0647 - val_accuracy: 0.9813
```

```
Epoch 27/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0114 -
accuracy: 0.9999 - val_loss: 0.0643 - val_accuracy: 0.9815
Epoch 28/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0111 -
accuracy: 1.0000 - val_loss: 0.0647 - val_accuracy: 0.9811
Epoch 29/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0110 -
accuracy: 0.9999 - val_loss: 0.0636 - val_accuracy: 0.9819
Epoch 30/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0105 -
accuracy: 0.9999 - val_loss: 0.0641 - val_accuracy: 0.9814
Epoch 31/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0100 -
accuracy: 1.0000 - val_loss: 0.0634 - val_accuracy: 0.9816
Epoch 32/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0098 -
accuracy: 1.0000 - val_loss: 0.0645 - val_accuracy: 0.9812
Epoch 33/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0095 -
accuracy: 1.0000 - val_loss: 0.0635 - val_accuracy: 0.9818
Epoch 34/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0092 -
accuracy: 1.0000 - val_loss: 0.0636 - val_accuracy: 0.9816
Epoch 35/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0090 -
accuracy: 1.0000 - val_loss: 0.0631 - val_accuracy: 0.9816
Epoch 36/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0088 -
accuracy: 1.0000 - val_loss: 0.0629 - val_accuracy: 0.9821
Epoch 37/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0086 -
accuracy: 1.0000 - val_loss: 0.0627 - val_accuracy: 0.9819
Epoch 38/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0085 -
accuracy: 1.0000 - val_loss: 0.0634 - val_accuracy: 0.9821
Epoch 39/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0082 -
accuracy: 1.0000 - val_loss: 0.0630 - val_accuracy: 0.9819
Epoch 40/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0080 -
accuracy: 1.0000 - val_loss: 0.0627 - val_accuracy: 0.9822
Epoch 1/40
469/469 [==============================] - 6s 12ms/step - loss: 0.6931 -
accuracy: 0.8156 - val_loss: 0.2214 - val_accuracy: 0.9409
Epoch 2/40
469/469 [==============================] - 6s 12ms/step - loss: 0.2111 -
accuracy: 0.9425 - val_loss: 0.1656 - val_accuracy: 0.9539
```

```
Epoch 3/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1522 -
accuracy: 0.9596 - val_loss: 0.1296 - val_accuracy: 0.9632
Epoch 4/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1151 -
accuracy: 0.9699 - val_loss: 0.1129 - val_accuracy: 0.9682
Epoch 5/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0987 -
accuracy: 0.9747 - val_loss: 0.1027 - val_accuracy: 0.9713
Epoch 6/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0825 -
accuracy: 0.9788 - val_loss: 0.0926 - val_accuracy: 0.9733
Epoch 7/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0685 -
accuracy: 0.9841 - val_loss: 0.0901 - val_accuracy: 0.9751
Epoch 8/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0587 -
accuracy: 0.9870 - val_loss: 0.0835 - val_accuracy: 0.9760
Epoch 9/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0525 -
accuracy: 0.9884 - val_loss: 0.0755 - val_accuracy: 0.9790
Epoch 10/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0454 -
accuracy: 0.9907 - val_loss: 0.0791 - val_accuracy: 0.9768
Epoch 11/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0394 -
accuracy: 0.9926 - val_loss: 0.0701 - val_accuracy: 0.9793
Epoch 12/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0352 -
accuracy: 0.9943 - val_loss: 0.0716 - val_accuracy: 0.9791
Epoch 13/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0316 -
accuracy: 0.9951 - val_loss: 0.0674 - val_accuracy: 0.9808
Epoch 14/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0283 -
accuracy: 0.9960 - val_loss: 0.0661 - val_accuracy: 0.9805
Epoch 15/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0257 -
accuracy: 0.9972 - val_loss: 0.0663 - val_accuracy: 0.9800
Epoch 16/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0230 -
accuracy: 0.9978 - val_loss: 0.0671 - val_accuracy: 0.9800
Epoch 17/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0211 -
accuracy: 0.9985 - val_loss: 0.0652 - val_accuracy: 0.9804
Epoch 18/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0200 -
accuracy: 0.9988 - val_loss: 0.0653 - val_accuracy: 0.9805
```

```
Epoch 19/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0178 -
accuracy: 0.9990 - val_loss: 0.0649 - val_accuracy: 0.9808
Epoch 20/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0165 -
accuracy: 0.9993 - val_loss: 0.0646 - val_accuracy: 0.9813
Epoch 21/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0159 -
accuracy: 0.9994 - val_loss: 0.0633 - val_accuracy: 0.9816
Epoch 22/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0148 -
accuracy: 0.9995 - val_loss: 0.0640 - val_accuracy: 0.9812
Epoch 23/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0140 -
accuracy: 0.9997 - val_loss: 0.0623 - val_accuracy: 0.9823
Epoch 24/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0132 -
accuracy: 0.9998 - val_loss: 0.0639 - val_accuracy: 0.9815
Epoch 25/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0123 -
accuracy: 0.9999 - val_loss: 0.0624 - val_accuracy: 0.9824
Epoch 26/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0120 -
accuracy: 0.9998 - val_loss: 0.0640 - val_accuracy: 0.9814
Epoch 27/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0115 -
accuracy: 0.9998 - val_loss: 0.0630 - val_accuracy: 0.9818
Epoch 28/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0113 -
accuracy: 0.9999 - val_loss: 0.0635 - val_accuracy: 0.9819
Epoch 29/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0108 -
accuracy: 0.9999 - val_loss: 0.0633 - val_accuracy: 0.9818
Epoch 30/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0105 -
accuracy: 0.9999 - val_loss: 0.0624 - val_accuracy: 0.9821
Epoch 31/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0100 -
accuracy: 0.9999 - val_loss: 0.0625 - val_accuracy: 0.9824
Epoch 32/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0097 -
accuracy: 0.9999 - val_loss: 0.0624 - val_accuracy: 0.9818
Epoch 33/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0094 -
accuracy: 0.9999 - val_loss: 0.0619 - val_accuracy: 0.9827
Epoch 34/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0091 -
accuracy: 1.0000 - val_loss: 0.0622 - val_accuracy: 0.9827
```

```
Epoch 35/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0089 -
accuracy: 1.0000 - val_loss: 0.0620 - val_accuracy: 0.9822
Epoch 36/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0086 -
accuracy: 1.0000 - val_loss: 0.0621 - val_accuracy: 0.9821
Epoch 37/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0085 -
accuracy: 1.0000 - val_loss: 0.0615 - val_accuracy: 0.9820
Epoch 38/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0083 -
accuracy: 1.0000 - val_loss: 0.0618 - val_accuracy: 0.9826
Epoch 39/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0080 -
accuracy: 1.0000 - val_loss: 0.0621 - val_accuracy: 0.9827
Epoch 40/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0079 -
accuracy: 1.0000 - val_loss: 0.0621 - val_accuracy: 0.9822
Epoch 1/40
469/469 [==============================] - 6s 12ms/step - loss: 0.6899 -
accuracy: 0.8133 - val_loss: 0.2392 - val_accuracy: 0.9304
Epoch 2/40
469/469 [==============================] - 6s 12ms/step - loss: 0.2033 -
accuracy: 0.9429 - val_loss: 0.1544 - val_accuracy: 0.9548
Epoch 3/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1361 -
accuracy: 0.9622 - val_loss: 0.1228 - val_accuracy: 0.9632
Epoch 4/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1117 -
accuracy: 0.9694 - val_loss: 0.1084 - val_accuracy: 0.9687
Epoch 5/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0886 -
accuracy: 0.9751 - val_loss: 0.0955 - val_accuracy: 0.9720
Epoch 6/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0768 -
accuracy: 0.9789 - val_loss: 0.0932 - val_accuracy: 0.9729
Epoch 7/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0622 -
accuracy: 0.9829 - val_loss: 0.0802 - val_accuracy: 0.9756
Epoch 8/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0540 -
accuracy: 0.9860 - val_loss: 0.0790 - val_accuracy: 0.9760
Epoch 9/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0469 -
accuracy: 0.9879 - val_loss: 0.0755 - val_accuracy: 0.9773
Epoch 10/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0408 -
accuracy: 0.9903 - val_loss: 0.0756 - val_accuracy: 0.9769
```

```
Epoch 11/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0360 -
accuracy: 0.9914 - val_loss: 0.0739 - val_accuracy: 0.9772
Epoch 12/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0308 -
accuracy: 0.9934 - val_loss: 0.0707 - val_accuracy: 0.9784
Epoch 13/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0263 -
accuracy: 0.9950 - val_loss: 0.0675 - val_accuracy: 0.9794
Epoch 14/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0254 -
accuracy: 0.9953 - val_loss: 0.0674 - val_accuracy: 0.9799
Epoch 15/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0215 -
accuracy: 0.9966 - val_loss: 0.0658 - val_accuracy: 0.9802
Epoch 16/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0189 -
accuracy: 0.9973 - val_loss: 0.0786 - val_accuracy: 0.9775
Epoch 17/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0162 -
accuracy: 0.9980 - val_loss: 0.0629 - val_accuracy: 0.9819
Epoch 18/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0155 -
accuracy: 0.9983 - val_loss: 0.0653 - val_accuracy: 0.9806
Epoch 19/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0137 -
accuracy: 0.9984 - val_loss: 0.0634 - val_accuracy: 0.9812
Epoch 20/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0121 -
accuracy: 0.9990 - val_loss: 0.0674 - val_accuracy: 0.9812
Epoch 21/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0116 -
accuracy: 0.9993 - val_loss: 0.0631 - val_accuracy: 0.9810
Epoch 22/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0102 -
accuracy: 0.9994 - val_loss: 0.0640 - val_accuracy: 0.9812
Epoch 23/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0089 -
accuracy: 0.9997 - val_loss: 0.0647 - val_accuracy: 0.9808
Epoch 24/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0088 -
accuracy: 0.9998 - val_loss: 0.0649 - val_accuracy: 0.9814
Epoch 25/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0082 -
accuracy: 0.9997 - val_loss: 0.0633 - val_accuracy: 0.9814
Epoch 26/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0075 -
accuracy: 0.9999 - val_loss: 0.0631 - val_accuracy: 0.9818
```

```
Epoch 27/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0071 -
accuracy: 0.9999 - val_loss: 0.0645 - val_accuracy: 0.9819
Epoch 28/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0070 -
accuracy: 0.9999 - val_loss: 0.0635 - val_accuracy: 0.9815
Epoch 29/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0065 -
accuracy: 0.9998 - val_loss: 0.0638 - val_accuracy: 0.9819
Epoch 30/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0063 -
accuracy: 0.9999 - val_loss: 0.0624 - val_accuracy: 0.9824
Epoch 31/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0059 -
accuracy: 0.9999 - val_loss: 0.0638 - val_accuracy: 0.9818
Epoch 32/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0056 -
accuracy: 1.0000 - val_loss: 0.0640 - val_accuracy: 0.9817
Epoch 33/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0054 -
accuracy: 1.0000 - val_loss: 0.0634 - val_accuracy: 0.9824
Epoch 34/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0053 -
accuracy: 1.0000 - val_loss: 0.0636 - val_accuracy: 0.9823
Epoch 35/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0052 -
accuracy: 1.0000 - val_loss: 0.0631 - val_accuracy: 0.9823
Epoch 36/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0050 -
accuracy: 1.0000 - val_loss: 0.0645 - val_accuracy: 0.9819
Epoch 37/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0049 -
accuracy: 1.0000 - val_loss: 0.0635 - val_accuracy: 0.9826
Epoch 38/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0047 -
accuracy: 1.0000 - val_loss: 0.0639 - val_accuracy: 0.9821
Epoch 39/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0047 -
accuracy: 1.0000 - val_loss: 0.0637 - val_accuracy: 0.9825
Epoch 40/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0045 -
accuracy: 1.0000 - val_loss: 0.0641 - val_accuracy: 0.9822
Epoch 1/40
469/469 [==============================] - 6s 12ms/step - loss: 0.7016 -
accuracy: 0.8064 - val_loss: 0.2393 - val_accuracy: 0.9318
Epoch 2/40
469/469 [==============================] - 6s 12ms/step - loss: 0.2144 -
accuracy: 0.9390 - val_loss: 0.1512 - val_accuracy: 0.9557
```

```
Epoch 3/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1442 -
accuracy: 0.9593 - val_loss: 0.1245 - val_accuracy: 0.9630
Epoch 4/40
469/469 [==============================] - 6s 12ms/step - loss: 0.1102 -
accuracy: 0.9685 - val_loss: 0.1109 - val_accuracy: 0.9678
Epoch 5/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0910 -
accuracy: 0.9739 - val_loss: 0.0924 - val_accuracy: 0.9724
Epoch 6/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0752 -
accuracy: 0.9798 - val_loss: 0.0885 - val_accuracy: 0.9730
Epoch 7/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0593 -
accuracy: 0.9850 - val_loss: 0.0802 - val_accuracy: 0.9757
Epoch 8/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0563 -
accuracy: 0.9853 - val_loss: 0.0753 - val_accuracy: 0.9760
Epoch 9/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0472 -
accuracy: 0.9874 - val_loss: 0.0751 - val_accuracy: 0.9771
Epoch 10/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0418 -
accuracy: 0.9900 - val_loss: 0.0703 - val_accuracy: 0.9784
Epoch 11/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0348 -
accuracy: 0.9923 - val_loss: 0.0734 - val_accuracy: 0.9781
Epoch 12/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0308 -
accuracy: 0.9928 - val_loss: 0.0724 - val_accuracy: 0.9779
Epoch 13/40
469/469 [==============================] - 6s 12ms/step - loss: 0.0268 -
accuracy: 0.9946 - val_loss: 0.0656 - val_accuracy: 0.9803
Epoch 14/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0229 -
accuracy: 0.9955 - val_loss: 0.0709 - val_accuracy: 0.9781
Epoch 15/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0208 -
accuracy: 0.9969 - val_loss: 0.0637 - val_accuracy: 0.9805
Epoch 16/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0178 -
accuracy: 0.9973 - val_loss: 0.0648 - val_accuracy: 0.9811
Epoch 17/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0159 -
accuracy: 0.9982 - val_loss: 0.0623 - val_accuracy: 0.9806
Epoch 18/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0146 -
accuracy: 0.9983 - val_loss: 0.0637 - val_accuracy: 0.9809
```

```
Epoch 19/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0130 -
accuracy: 0.9990 - val_loss: 0.0633 - val_accuracy: 0.9812
Epoch 20/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0120 -
accuracy: 0.9991 - val_loss: 0.0617 - val_accuracy: 0.9822
Epoch 21/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0111 -
accuracy: 0.9992 - val_loss: 0.0637 - val_accuracy: 0.9811
Epoch 22/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0097 -
accuracy: 0.9996 - val_loss: 0.0644 - val_accuracy: 0.9808
Epoch 23/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0091 -
accuracy: 0.9997 - val_loss: 0.0632 - val_accuracy: 0.9814
Epoch 24/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0089 -
accuracy: 0.9995 - val_loss: 0.0620 - val_accuracy: 0.9819
Epoch 25/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0080 -
accuracy: 0.9998 - val_loss: 0.0641 - val_accuracy: 0.9816
Epoch 26/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0074 -
accuracy: 0.9998 - val_loss: 0.0632 - val_accuracy: 0.9814
Epoch 27/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0070 -
accuracy: 0.9998 - val_loss: 0.0628 - val_accuracy: 0.9819
Epoch 28/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0068 -
accuracy: 0.9999 - val_loss: 0.0633 - val_accuracy: 0.9821
Epoch 29/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0061 -
accuracy: 0.9999 - val_loss: 0.0624 - val_accuracy: 0.9819
Epoch 30/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0060 -
accuracy: 0.9999 - val_loss: 0.0628 - val_accuracy: 0.9824
Epoch 31/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0058 -
accuracy: 0.9999 - val_loss: 0.0635 - val_accuracy: 0.9824
Epoch 32/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0056 -
accuracy: 1.0000 - val_loss: 0.0631 - val_accuracy: 0.9820
Epoch 33/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0053 -
accuracy: 1.0000 - val_loss: 0.0627 - val_accuracy: 0.9820
Epoch 34/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0052 -
accuracy: 1.0000 - val_loss: 0.0632 - val_accuracy: 0.9824
```

```
Epoch 35/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0050 -
accuracy: 1.0000 - val_loss: 0.0643 - val_accuracy: 0.9821
Epoch 36/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0049 -
accuracy: 1.0000 - val_loss: 0.0637 - val_accuracy: 0.9820
Epoch 37/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0048 -
accuracy: 1.0000 - val_loss: 0.0637 - val_accuracy: 0.9822
Epoch 38/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0046 -
accuracy: 1.0000 - val_loss: 0.0634 - val_accuracy: 0.9827
Epoch 39/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0046 -
accuracy: 1.0000 - val_loss: 0.0635 - val_accuracy: 0.9824
Epoch 40/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0044 -
accuracy: 1.0000 - val_loss: 0.0645 - val_accuracy: 0.9825
Epoch 1/40
469/469 [==============================] - 7s 13ms/step - loss: 0.6758 -
accuracy: 0.8157 - val_loss: 0.2539 - val_accuracy: 0.9203
Epoch 2/40
469/469 [==============================] - 6s 13ms/step - loss: 0.2049 -
accuracy: 0.9411 - val_loss: 0.1560 - val_accuracy: 0.9533
Epoch 3/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1446 -
accuracy: 0.9593 - val_loss: 0.1277 - val_accuracy: 0.9627
Epoch 4/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1113 -
accuracy: 0.9679 - val_loss: 0.0991 - val_accuracy: 0.9701
Epoch 5/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0894 -
accuracy: 0.9746 - val_loss: 0.0908 - val_accuracy: 0.9729
Epoch 6/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0731 -
accuracy: 0.9802 - val_loss: 0.0847 - val_accuracy: 0.9757
Epoch 7/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0636 -
accuracy: 0.9827 - val_loss: 0.0767 - val_accuracy: 0.9779
Epoch 8/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0530 -
accuracy: 0.9858 - val_loss: 0.0731 - val_accuracy: 0.9783
Epoch 9/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0451 -
accuracy: 0.9882 - val_loss: 0.0676 - val_accuracy: 0.9809
Epoch 10/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0431 -
accuracy: 0.9888 - val_loss: 0.0711 - val_accuracy: 0.9779
```

```
Epoch 11/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0357 -
accuracy: 0.9918 - val_loss: 0.0651 - val_accuracy: 0.9806
Epoch 12/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0294 -
accuracy: 0.9936 - val_loss: 0.0649 - val_accuracy: 0.9797
Epoch 13/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0278 -
accuracy: 0.9943 - val_loss: 0.0623 - val_accuracy: 0.9806
Epoch 14/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0237 -
accuracy: 0.9956 - val_loss: 0.0599 - val_accuracy: 0.9815
Epoch 15/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0214 -
accuracy: 0.9966 - val_loss: 0.0596 - val_accuracy: 0.9813
Epoch 16/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0188 -
accuracy: 0.9971 - val_loss: 0.0582 - val_accuracy: 0.9813
Epoch 17/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0160 -
accuracy: 0.9982 - val_loss: 0.0664 - val_accuracy: 0.9791
Epoch 18/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0146 -
accuracy: 0.9982 - val_loss: 0.0587 - val_accuracy: 0.9823
Epoch 19/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0129 -
accuracy: 0.9989 - val_loss: 0.0593 - val_accuracy: 0.9817
Epoch 20/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0126 -
accuracy: 0.9988 - val_loss: 0.0586 - val_accuracy: 0.9824
Epoch 21/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0109 -
accuracy: 0.9992 - val_loss: 0.0588 - val_accuracy: 0.9829
Epoch 22/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0099 -
accuracy: 0.9994 - val_loss: 0.0579 - val_accuracy: 0.9829
Epoch 23/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0091 -
accuracy: 0.9996 - val_loss: 0.0590 - val_accuracy: 0.9828
Epoch 24/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0085 -
accuracy: 0.9996 - val_loss: 0.0577 - val_accuracy: 0.9836
Epoch 25/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0082 -
accuracy: 0.9997 - val_loss: 0.0581 - val_accuracy: 0.9835
Epoch 26/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0078 -
accuracy: 0.9997 - val_loss: 0.0582 - val_accuracy: 0.9831
```

```
Epoch 27/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0070 -
accuracy: 0.9999 - val_loss: 0.0598 - val_accuracy: 0.9834
Epoch 28/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0068 -
accuracy: 0.9999 - val_loss: 0.0590 - val_accuracy: 0.9827
Epoch 29/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0062 -
accuracy: 0.9999 - val_loss: 0.0592 - val_accuracy: 0.9831
Epoch 30/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0062 -
accuracy: 0.9999 - val_loss: 0.0588 - val_accuracy: 0.9836
Epoch 31/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0057 -
accuracy: 1.0000 - val_loss: 0.0592 - val_accuracy: 0.9830
Epoch 32/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0056 -
accuracy: 1.0000 - val_loss: 0.0588 - val_accuracy: 0.9839
Epoch 33/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0055 -
accuracy: 1.0000 - val_loss: 0.0589 - val_accuracy: 0.9834
Epoch 34/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0052 -
accuracy: 1.0000 - val_loss: 0.0586 - val_accuracy: 0.9837
Epoch 35/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0050 -
accuracy: 1.0000 - val_loss: 0.0592 - val_accuracy: 0.9834
Epoch 36/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0048 -
accuracy: 1.0000 - val_loss: 0.0592 - val_accuracy: 0.9840
Epoch 37/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0048 -
accuracy: 1.0000 - val_loss: 0.0595 - val_accuracy: 0.9839
Epoch 38/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0047 -
accuracy: 1.0000 - val_loss: 0.0587 - val_accuracy: 0.9836
Epoch 39/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0045 -
accuracy: 1.0000 - val_loss: 0.0591 - val_accuracy: 0.9842
Epoch 40/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0043 -
accuracy: 1.0000 - val_loss: 0.0592 - val_accuracy: 0.9840
Epoch 1/40
469/469 [==============================] - 7s 14ms/step - loss: 0.6868 -
accuracy: 0.8156 - val_loss: 0.2120 - val_accuracy: 0.9365
Epoch 2/40
469/469 [==============================] - 6s 14ms/step - loss: 0.2054 -
accuracy: 0.9420 - val_loss: 0.1646 - val_accuracy: 0.9507
```

```
Epoch 3/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1512 -
accuracy: 0.9564 - val_loss: 0.1280 - val_accuracy: 0.9606
Epoch 4/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1150 -
accuracy: 0.9662 - val_loss: 0.1040 - val_accuracy: 0.9690
Epoch 5/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0906 -
accuracy: 0.9747 - val_loss: 0.1050 - val_accuracy: 0.9671
Epoch 6/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0741 -
accuracy: 0.9785 - val_loss: 0.0872 - val_accuracy: 0.9734
Epoch 7/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0671 -
accuracy: 0.9813 - val_loss: 0.0779 - val_accuracy: 0.9760
Epoch 8/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0538 -
accuracy: 0.9845 - val_loss: 0.0752 - val_accuracy: 0.9764
Epoch 9/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0456 -
accuracy: 0.9874 - val_loss: 0.0700 - val_accuracy: 0.9775
Epoch 10/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0406 -
accuracy: 0.9890 - val_loss: 0.0733 - val_accuracy: 0.9768
Epoch 11/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0354 -
accuracy: 0.9906 - val_loss: 0.0663 - val_accuracy: 0.9789
Epoch 12/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0320 -
accuracy: 0.9918 - val_loss: 0.0697 - val_accuracy: 0.9781
Epoch 13/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0277 -
accuracy: 0.9931 - val_loss: 0.0673 - val_accuracy: 0.9797
Epoch 14/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0230 -
accuracy: 0.9951 - val_loss: 0.0650 - val_accuracy: 0.9787
Epoch 15/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0200 -
accuracy: 0.9959 - val_loss: 0.0621 - val_accuracy: 0.9796
Epoch 16/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0172 -
accuracy: 0.9970 - val_loss: 0.0634 - val_accuracy: 0.9807
Epoch 17/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0151 -
accuracy: 0.9974 - val_loss: 0.0671 - val_accuracy: 0.9787
Epoch 18/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0141 -
accuracy: 0.9979 - val_loss: 0.0637 - val_accuracy: 0.9801
```

```
Epoch 19/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0122 -
accuracy: 0.9985 - val_loss: 0.0629 - val_accuracy: 0.9807
Epoch 20/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0107 -
accuracy: 0.9988 - val_loss: 0.0621 - val_accuracy: 0.9800
Epoch 21/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0090 -
accuracy: 0.9992 - val_loss: 0.0628 - val_accuracy: 0.9800
Epoch 22/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0090 -
accuracy: 0.9992 - val_loss: 0.0614 - val_accuracy: 0.9811
Epoch 23/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0074 -
accuracy: 0.9995 - val_loss: 0.0618 - val_accuracy: 0.9811
Epoch 24/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0069 -
accuracy: 0.9995 - val_loss: 0.0618 - val_accuracy: 0.9808
Epoch 25/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0061 -
accuracy: 0.9996 - val_loss: 0.0627 - val_accuracy: 0.9809
Epoch 26/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0054 -
accuracy: 0.9997 - val_loss: 0.0612 - val_accuracy: 0.9809
Epoch 27/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0049 -
accuracy: 0.9998 - val_loss: 0.0623 - val_accuracy: 0.9813
Epoch 28/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0047 -
accuracy: 0.9998 - val_loss: 0.0637 - val_accuracy: 0.9809
Epoch 29/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0044 -
accuracy: 0.9999 - val_loss: 0.0634 - val_accuracy: 0.9814
Epoch 30/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0040 -
accuracy: 1.0000 - val_loss: 0.0623 - val_accuracy: 0.9819
Epoch 31/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0036 -
accuracy: 1.0000 - val_loss: 0.0623 - val_accuracy: 0.9814
Epoch 32/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0034 -
accuracy: 1.0000 - val_loss: 0.0622 - val_accuracy: 0.9813
Epoch 33/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0032 -
accuracy: 1.0000 - val_loss: 0.0633 - val_accuracy: 0.9812
Epoch 34/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0030 -
accuracy: 1.0000 - val_loss: 0.0642 - val_accuracy: 0.9818
```

```
Epoch 35/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0030 -
accuracy: 1.0000 - val_loss: 0.0642 - val_accuracy: 0.9811
Epoch 36/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0026 -
accuracy: 1.0000 - val_loss: 0.0636 - val_accuracy: 0.9811
Epoch 37/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0026 -
accuracy: 1.0000 - val_loss: 0.0643 - val_accuracy: 0.9818
Epoch 38/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0025 -
accuracy: 1.0000 - val_loss: 0.0643 - val_accuracy: 0.9814
Epoch 39/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0024 -
accuracy: 1.0000 - val_loss: 0.0645 - val_accuracy: 0.9818
Epoch 40/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0022 -
accuracy: 1.0000 - val_loss: 0.0649 - val_accuracy: 0.9816
Epoch 1/40
469/469 [==============================] - 7s 14ms/step - loss: 0.6798 -
accuracy: 0.8154 - val_loss: 0.2246 - val_accuracy: 0.9312
Epoch 2/40
469/469 [==============================] - 6s 13ms/step - loss: 0.2040 -
accuracy: 0.9408 - val_loss: 0.1445 - val_accuracy: 0.9571
Epoch 3/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1420 -
accuracy: 0.9600 - val_loss: 0.1220 - val_accuracy: 0.9633
Epoch 4/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1110 -
accuracy: 0.9670 - val_loss: 0.1051 - val_accuracy: 0.9695
Epoch 5/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0895 -
accuracy: 0.9744 - val_loss: 0.0917 - val_accuracy: 0.9707
Epoch 6/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0732 -
accuracy: 0.9794 - val_loss: 0.0890 - val_accuracy: 0.9728
Epoch 7/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0601 -
accuracy: 0.9836 - val_loss: 0.0813 - val_accuracy: 0.9752
Epoch 8/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0500 -
accuracy: 0.9866 - val_loss: 0.0727 - val_accuracy: 0.9789
Epoch 9/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0467 -
accuracy: 0.9874 - val_loss: 0.0717 - val_accuracy: 0.9772
Epoch 10/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0376 -
accuracy: 0.9903 - val_loss: 0.0694 - val_accuracy: 0.9778
```

```
Epoch 11/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0317 -
accuracy: 0.9921 - val_loss: 0.0655 - val_accuracy: 0.9795
Epoch 12/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0278 -
accuracy: 0.9939 - val_loss: 0.0642 - val_accuracy: 0.9800
Epoch 13/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0245 -
accuracy: 0.9945 - val_loss: 0.0613 - val_accuracy: 0.9806
Epoch 14/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0209 -
accuracy: 0.9960 - val_loss: 0.0650 - val_accuracy: 0.9811
Epoch 15/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0182 -
accuracy: 0.9967 - val_loss: 0.0623 - val_accuracy: 0.9800
Epoch 16/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0149 -
accuracy: 0.9976 - val_loss: 0.0613 - val_accuracy: 0.9806
Epoch 17/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0132 -
accuracy: 0.9978 - val_loss: 0.0588 - val_accuracy: 0.9819
Epoch 18/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0118 -
accuracy: 0.9983 - val_loss: 0.0641 - val_accuracy: 0.9803
Epoch 19/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0105 -
accuracy: 0.9988 - val_loss: 0.0597 - val_accuracy: 0.9824
Epoch 20/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0097 -
accuracy: 0.9991 - val_loss: 0.0605 - val_accuracy: 0.9819
Epoch 21/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0084 -
accuracy: 0.9994 - val_loss: 0.0614 - val_accuracy: 0.9816
Epoch 22/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0077 -
accuracy: 0.9994 - val_loss: 0.0618 - val_accuracy: 0.9815
Epoch 23/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0067 -
accuracy: 0.9996 - val_loss: 0.0608 - val_accuracy: 0.9824
Epoch 24/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0061 -
accuracy: 0.9996 - val_loss: 0.0627 - val_accuracy: 0.9823
Epoch 25/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0053 -
accuracy: 0.9998 - val_loss: 0.0622 - val_accuracy: 0.9818
Epoch 26/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0048 -
accuracy: 0.9999 - val_loss: 0.0624 - val_accuracy: 0.9812
```

```
Epoch 27/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0043 -
accuracy: 0.9999 - val_loss: 0.0628 - val_accuracy: 0.9815
Epoch 28/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0041 -
accuracy: 0.9999 - val_loss: 0.0629 - val_accuracy: 0.9814
Epoch 29/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0038 -
accuracy: 0.9999 - val_loss: 0.0637 - val_accuracy: 0.9815
Epoch 30/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0036 -
accuracy: 0.9999 - val_loss: 0.0631 - val_accuracy: 0.9817
Epoch 31/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0033 -
accuracy: 1.0000 - val_loss: 0.0627 - val_accuracy: 0.9820
Epoch 32/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0031 -
accuracy: 1.0000 - val_loss: 0.0637 - val_accuracy: 0.9822
Epoch 33/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0030 -
accuracy: 1.0000 - val_loss: 0.0633 - val_accuracy: 0.9826
Epoch 34/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0028 -
accuracy: 1.0000 - val_loss: 0.0646 - val_accuracy: 0.9820
Epoch 35/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0027 -
accuracy: 1.0000 - val_loss: 0.0635 - val_accuracy: 0.9817
Epoch 36/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0025 -
accuracy: 1.0000 - val_loss: 0.0650 - val_accuracy: 0.9818
Epoch 37/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0024 -
accuracy: 1.0000 - val_loss: 0.0648 - val_accuracy: 0.9816
Epoch 38/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0023 -
accuracy: 1.0000 - val_loss: 0.0648 - val_accuracy: 0.9820
Epoch 39/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0023 -
accuracy: 0.9999 - val_loss: 0.0654 - val_accuracy: 0.9823
Epoch 40/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0021 -
accuracy: 1.0000 - val_loss: 0.0651 - val_accuracy: 0.9823
Epoch 1/40
469/469 [==============================] - 7s 14ms/step - loss: 0.6800 -
accuracy: 0.8162 - val_loss: 0.2206 - val_accuracy: 0.9369
Epoch 2/40
469/469 [==============================] - 6s 13ms/step - loss: 0.2095 -
accuracy: 0.9385 - val_loss: 0.1565 - val_accuracy: 0.9549
```

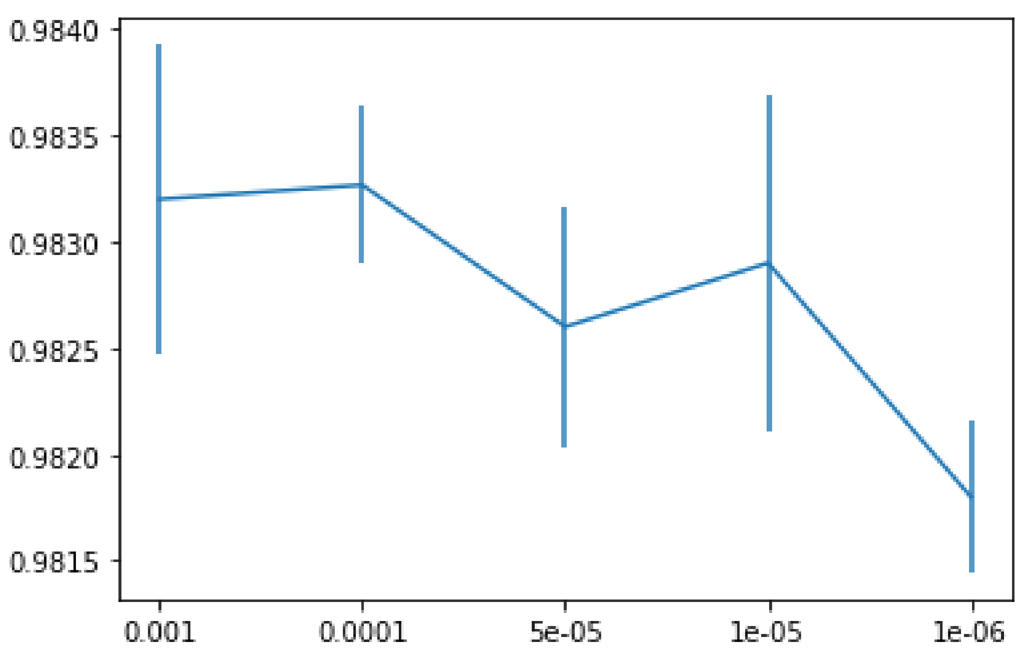
```
Epoch 3/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1456 -
accuracy: 0.9587 - val_loss: 0.1221 - val_accuracy: 0.9650
Epoch 4/40
469/469 [==============================] - 6s 13ms/step - loss: 0.1124 -
accuracy: 0.9688 - val_loss: 0.1100 - val_accuracy: 0.9661
Epoch 5/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0904 -
accuracy: 0.9738 - val_loss: 0.0884 - val_accuracy: 0.9730
Epoch 6/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0723 -
accuracy: 0.9802 - val_loss: 0.0830 - val_accuracy: 0.9741
Epoch 7/40
469/469 [==============================] - 7s 14ms/step - loss: 0.0632 -
accuracy: 0.9823 - val_loss: 0.0794 - val_accuracy: 0.9758
Epoch 8/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0541 -
accuracy: 0.9849 - val_loss: 0.0702 - val_accuracy: 0.9783
Epoch 9/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0465 -
accuracy: 0.9876 - val_loss: 0.0688 - val_accuracy: 0.9788
Epoch 10/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0399 -
accuracy: 0.9891 - val_loss: 0.0667 - val_accuracy: 0.9789
Epoch 11/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0339 -
accuracy: 0.9915 - val_loss: 0.0681 - val_accuracy: 0.9771
Epoch 12/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0305 -
accuracy: 0.9926 - val_loss: 0.0682 - val_accuracy: 0.9772
Epoch 13/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0251 -
accuracy: 0.9942 - val_loss: 0.0660 - val_accuracy: 0.9782
Epoch 14/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0224 -
accuracy: 0.9951 - val_loss: 0.0644 - val_accuracy: 0.9797
Epoch 15/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0198 -
accuracy: 0.9956 - val_loss: 0.0606 - val_accuracy: 0.9804
Epoch 16/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0168 -
accuracy: 0.9971 - val_loss: 0.0599 - val_accuracy: 0.9794
Epoch 17/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0146 -
accuracy: 0.9976 - val_loss: 0.0616 - val_accuracy: 0.9792
Epoch 18/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0134 -
accuracy: 0.9981 - val_loss: 0.0677 - val_accuracy: 0.9783
```

```
Epoch 19/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0117 -
accuracy: 0.9985 - val_loss: 0.0605 - val_accuracy: 0.9805
Epoch 20/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0098 -
accuracy: 0.9989 - val_loss: 0.0579 - val_accuracy: 0.9825
Epoch 21/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0089 -
accuracy: 0.9991 - val_loss: 0.0611 - val_accuracy: 0.9802
Epoch 22/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0080 -
accuracy: 0.9992 - val_loss: 0.0592 - val_accuracy: 0.9807
Epoch 23/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0072 -
accuracy: 0.9995 - val_loss: 0.0607 - val_accuracy: 0.9811
Epoch 24/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0065 -
accuracy: 0.9996 - val_loss: 0.0588 - val_accuracy: 0.9807
Epoch 25/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0058 -
accuracy: 0.9997 - val_loss: 0.0585 - val_accuracy: 0.9809
Epoch 26/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0051 -
accuracy: 0.9997 - val_loss: 0.0604 - val_accuracy: 0.9811
Epoch 27/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0049 -
accuracy: 0.9998 - val_loss: 0.0595 - val_accuracy: 0.9814
Epoch 28/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0042 -
accuracy: 0.9999 - val_loss: 0.0604 - val_accuracy: 0.9809
Epoch 29/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0041 -
accuracy: 0.9999 - val_loss: 0.0598 - val_accuracy: 0.9809
Epoch 30/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0037 -
accuracy: 0.9999 - val_loss: 0.0598 - val_accuracy: 0.9813
Epoch 31/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0035 -
accuracy: 0.9999 - val_loss: 0.0608 - val_accuracy: 0.9817
Epoch 32/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0033 -
accuracy: 0.9999 - val_loss: 0.0602 - val_accuracy: 0.9815
Epoch 33/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0030 -
accuracy: 1.0000 - val_loss: 0.0608 - val_accuracy: 0.9809
Epoch 34/40
469/469 [==============================] - 6s 14ms/step - loss: 0.0029 -
accuracy: 1.0000 - val_loss: 0.0608 - val_accuracy: 0.9812
```

```
Epoch 35/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0030 -
accuracy: 1.0000 - val_loss: 0.0611 - val_accuracy: 0.9808
Epoch 36/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0027 -
accuracy: 1.0000 - val_loss: 0.0618 - val_accuracy: 0.9811
Epoch 37/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0024 -
accuracy: 1.0000 - val_loss: 0.0622 - val_accuracy: 0.9813
Epoch 38/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0023 -
accuracy: 1.0000 - val_loss: 0.0618 - val_accuracy: 0.9815
Epoch 39/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0023 -
accuracy: 1.0000 - val_loss: 0.0625 - val_accuracy: 0.9815
Epoch 40/40
469/469 [==============================] - 6s 13ms/step - loss: 0.0022 -
accuracy: 1.0000 - val_loss: 0.0622 - val_accuracy: 0.9815
```

[ ]:

```
#Plot the final validation accuracy with standard deviation
x = list(map(str, reg_factors))
y = list(map(np.mean, val_accuracys))
e = list(map(np.std, val_accuracys))

plt.errorbar(x, y, yerr=e)
plt.show()
```



**How close do you get to Hintons result? – If you do not get the same results, what factors may influence this? (hint: What information is not given by Hinton on the MNIST database that may influence Model training)**

[**Answer**]

As seen in the graph the highest accuracy we get is around 0.9838. We did not quite get the same result as Hinton, which we believe can be due to us not choosing the most proper regularization factors which is something that is not mentioned in the MNIST database. Maybe by modfying the regularization factors we can get closer or get the same result as Hinton, but since it takes such a long time we just let it be. Another thing that was different was the batch sizes, Hinton had a batch size of 100 while we have 128, maybe this could also be something related to the result. The nr of epochs could be a factor, in the science paper Hinton uses a maxepoch of 50 while we use 40.

#3) 2points. Convolutional layers.

**A) Design a model that makes use of at least one convolutional layer–how performant a model can you get? –According to the MNIST database it should be possible reach to 99% accuracy on the validation data. If you choose to use any layers apart from convolutional layers and layers that you used in previous questions, you must describe what they do. If you do not reach 99% accuracy, report your best performance and explain your attempts and thought process.**

[**Answer**]

[ ]:
```
#model with convolutional layer
convModel = Sequential()
convModel.add(Conv2D(32, kernel_size=(3, 3),
                 activation='relu',
                 input_shape=(28,28,1)))

convModel.add(Conv2D(64, (3, 3), activation='relu'))
convModel.add(MaxPooling2D(pool_size=(2, 2)))
convModel.add(Dropout(0.25))# helps to prevent overfitting
convModel.add(Flatten())
convModel.add(Dense(128, activation='relu'))
convModel.add(Dropout(0.5))# helps to prevent overfitting
convModel.add(Dense(num_classes, activation='softmax'))

convModel.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.SGD(learning_rate=0.1),
              metrics=['accuracy'])

fit_info_convModel = convModel.fit(x_train, y_train,
                     batch_size=batch_size,
                     epochs=40,
                     verbose=1,
                     validation_data=(x_test, y_test))

score = convModel.evaluate(x_test, y_test, verbose=0)
```

```python
print("Accuracy: ", score[1])
```

```
Epoch 1/40
469/469 [==============================] - 142s 302ms/step - loss: 0.7713 -
accuracy: 0.7521 - val_loss: 0.1057 - val_accuracy: 0.9681
Epoch 2/40
469/469 [==============================] - 140s 299ms/step - loss: 0.1659 -
accuracy: 0.9511 - val_loss: 0.0632 - val_accuracy: 0.9792
Epoch 3/40
469/469 [==============================] - 140s 298ms/step - loss: 0.1177 -
accuracy: 0.9653 - val_loss: 0.0501 - val_accuracy: 0.9839
Epoch 4/40
469/469 [==============================] - 139s 297ms/step - loss: 0.0922 -
accuracy: 0.9714 - val_loss: 0.0431 - val_accuracy: 0.9853
Epoch 5/40
469/469 [==============================] - 139s 297ms/step - loss: 0.0782 -
accuracy: 0.9768 - val_loss: 0.0401 - val_accuracy: 0.9864
Epoch 6/40
469/469 [==============================] - 139s 297ms/step - loss: 0.0704 -
accuracy: 0.9782 - val_loss: 0.0379 - val_accuracy: 0.9877
Epoch 7/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0605 -
accuracy: 0.9818 - val_loss: 0.0392 - val_accuracy: 0.9877
Epoch 8/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0571 -
accuracy: 0.9825 - val_loss: 0.0342 - val_accuracy: 0.9893
Epoch 9/40
469/469 [==============================] - 137s 292ms/step - loss: 0.0536 -
accuracy: 0.9832 - val_loss: 0.0335 - val_accuracy: 0.9894
Epoch 10/40
469/469 [==============================] - 137s 292ms/step - loss: 0.0513 -
accuracy: 0.9840 - val_loss: 0.0314 - val_accuracy: 0.9898
Epoch 11/40
469/469 [==============================] - 137s 292ms/step - loss: 0.0468 -
accuracy: 0.9857 - val_loss: 0.0302 - val_accuracy: 0.9889
Epoch 12/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0409 -
accuracy: 0.9871 - val_loss: 0.0307 - val_accuracy: 0.9898
Epoch 13/40
469/469 [==============================] - 139s 297ms/step - loss: 0.0380 -
accuracy: 0.9876 - val_loss: 0.0273 - val_accuracy: 0.9911
Epoch 14/40
469/469 [==============================] - 140s 299ms/step - loss: 0.0369 -
accuracy: 0.9878 - val_loss: 0.0284 - val_accuracy: 0.9911
Epoch 15/40
469/469 [==============================] - 140s 298ms/step - loss: 0.0348 -
accuracy: 0.9889 - val_loss: 0.0286 - val_accuracy: 0.9908
```

```
Epoch 16/40
469/469 [==============================] - 140s 300ms/step - loss: 0.0332 -
accuracy: 0.9891 - val_loss: 0.0285 - val_accuracy: 0.9903
Epoch 17/40
469/469 [==============================] - 140s 298ms/step - loss: 0.0327 -
accuracy: 0.9894 - val_loss: 0.0284 - val_accuracy: 0.9907
Epoch 18/40
469/469 [==============================] - 139s 296ms/step - loss: 0.0286 -
accuracy: 0.9907 - val_loss: 0.0267 - val_accuracy: 0.9923
Epoch 19/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0285 -
accuracy: 0.9911 - val_loss: 0.0266 - val_accuracy: 0.9910
Epoch 20/40
469/469 [==============================] - 138s 294ms/step - loss: 0.0276 -
accuracy: 0.9912 - val_loss: 0.0286 - val_accuracy: 0.9909
Epoch 21/40
469/469 [==============================] - 137s 292ms/step - loss: 0.0262 -
accuracy: 0.9916 - val_loss: 0.0289 - val_accuracy: 0.9914
Epoch 22/40
469/469 [==============================] - 140s 298ms/step - loss: 0.0256 -
accuracy: 0.9922 - val_loss: 0.0264 - val_accuracy: 0.9917
Epoch 23/40
469/469 [==============================] - 140s 298ms/step - loss: 0.0224 -
accuracy: 0.9923 - val_loss: 0.0292 - val_accuracy: 0.9907
Epoch 24/40
469/469 [==============================] - 141s 301ms/step - loss: 0.0220 -
accuracy: 0.9926 - val_loss: 0.0273 - val_accuracy: 0.9913
Epoch 25/40
469/469 [==============================] - 140s 299ms/step - loss: 0.0229 -
accuracy: 0.9923 - val_loss: 0.0261 - val_accuracy: 0.9920
Epoch 26/40
469/469 [==============================] - 140s 298ms/step - loss: 0.0210 -
accuracy: 0.9929 - val_loss: 0.0255 - val_accuracy: 0.9920
Epoch 27/40
469/469 [==============================] - 138s 295ms/step - loss: 0.0212 -
accuracy: 0.9928 - val_loss: 0.0243 - val_accuracy: 0.9923
Epoch 28/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0196 -
accuracy: 0.9932 - val_loss: 0.0259 - val_accuracy: 0.9918
Epoch 29/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0207 -
accuracy: 0.9932 - val_loss: 0.0248 - val_accuracy: 0.9918
Epoch 30/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0198 -
accuracy: 0.9933 - val_loss: 0.0266 - val_accuracy: 0.9920
Epoch 31/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0183 -
accuracy: 0.9939 - val_loss: 0.0270 - val_accuracy: 0.9926
```

```
Epoch 32/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0172 -
accuracy: 0.9939 - val_loss: 0.0275 - val_accuracy: 0.9911
Epoch 33/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0177 -
accuracy: 0.9944 - val_loss: 0.0268 - val_accuracy: 0.9922
Epoch 34/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0189 -
accuracy: 0.9937 - val_loss: 0.0271 - val_accuracy: 0.9920
Epoch 35/40
469/469 [==============================] - 137s 292ms/step - loss: 0.0159 -
accuracy: 0.9946 - val_loss: 0.0265 - val_accuracy: 0.9929
Epoch 36/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0165 -
accuracy: 0.9941 - val_loss: 0.0287 - val_accuracy: 0.9924
Epoch 37/40
469/469 [==============================] - 137s 293ms/step - loss: 0.0153 -
accuracy: 0.9947 - val_loss: 0.0279 - val_accuracy: 0.9924
Epoch 38/40
469/469 [==============================] - 140s 299ms/step - loss: 0.0148 -
accuracy: 0.9945 - val_loss: 0.0287 - val_accuracy: 0.9919
Epoch 39/40
469/469 [==============================] - 140s 298ms/step - loss: 0.0148 -
accuracy: 0.9950 - val_loss: 0.0288 - val_accuracy: 0.9930
Epoch 40/40
469/469 [==============================] - 140s 299ms/step - loss: 0.0153 -
accuracy: 0.9948 - val_loss: 0.0288 - val_accuracy: 0.9926
Accuracy:  0.9926000237464905
```

After creating our convolutional layer we apply a MaxPooling2D layer which reduces the dimensions of the feature maps, thus, reducing the number of parameters to learn and computational load. Maxpooling might also reduce overfitting.

Maxpooling will select the maximum element(higher valued pixels that are the most activated) from each region generated by the convolutional layer and preserve these values in a 2x2 matrix. This makes the model more robust to variations in the position of the features in the input image.

Then we add a dropout layer which is used to prevent overfitting since it drops neurons out of the network during training so other neurons can step in and make the predictions for the missing neurons, the network then becomes less sensitive to specific weights and more generalized.

We then flatten the network, converting a matrix into a single array.

We then add a Dense layer which is a neural network. Our dense layer has 128 units(neurons) which is our batch size.

Another dropout layer is added with probability of 0.5 in order to retain the output of each node in a hidden layer.

And finally at last we have the output layer with 10 units which is the number of possible outputs/predictions classes that we have(0-9).

**B)Discuss the differences and potential benefits of using convolutional layers over fully connected onesfor the particular application?**

[**Answer**]

In a fully connected layer each neuron is connected to every neuron in the previous layer where each connection has it's own weight. This makes it more of a general principle that does not make any assumptions about the features in the data. Due to the mass of connections it becomes very expensive in terms of computation and memory.

In a convolutional layer each neuron is instead only connected to a few near(local) neurons in the previous layer. Each neuron is applied the same set of weights.

The convolutional layer consists of feature maps that are connected directly to the inputs. The feature maps enables the network to detect different kinds of features of the image.

The weakness of a fully connected network is that if we shift a digit slightly to any direction, the network will no longer recognize the digit. This is something that convolutional networks can handle. Convolutional networks are most used today in networks for image recognition, which makes it a better option for our problem at hand.

Due to the reduction of connections, convolutional layers makes it cheaper in terms of computation and memory.

#### 0.0.1 Question 4) Auto-Encoder for denoising

[ ]:
```
import numpy as np
def salt_and_pepper(input, noise_level=0.5):
    """
    This applies salt and pepper noise to the input tensor - randomly setting␣
 ↪bits to 1 or 0.
    Parameters
    ----------
    input : tensor
        The tensor to apply salt and pepper noise to.
    noise_level : float
        The amount of salt and pepper noise to add.
    Returns
    -------
    tensor
        Tensor with salt and pepper noise applied.
    """
    # salt and pepper noise
    a = np.random.binomial(size=input.shape, n=1, p=(1 - noise_level))
    b = np.random.binomial(size=input.shape, n=1, p=0.5)
    c = (a==0) * b
    return input * a + c
```

```
#data preparation
flattened_x_train = x_train.reshape(-1,784)
flattened_x_train_seasoned = salt_and_pepper(flattened_x_train, noise_level=0.4)

flattened_x_test = x_test.reshape(-1,784)
flattened_x_test_seasoneed = salt_and_pepper(flattened_x_test, noise_level=0.4)
```

[ ]:
```
latent_dim = 96

input_image = keras.Input(shape=(784,))
encoded = Dense(128, activation='relu')(input_image)
encoded = Dense(latent_dim, activation='relu')(encoded)
decoded = Dense(128, activation='relu')(encoded)
decoded = Dense(784, activation='sigmoid')(decoded)

autoencoder = keras.Model(input_image, decoded)
encoder_only = keras.Model(input_image, encoded)

encoded_input = keras.Input(shape=(latent_dim,))
decoder_layer = Sequential(autoencoder.layers[-2:])
decoder = keras.Model(encoded_input, decoder_layer(encoded_input))

autoencoder.compile(optimizer='adam', loss='binary_crossentropy')
```

[ ]:
```
fit_info_AE = autoencoder.fit(flattened_x_train_seasoned, flattened_x_train,
                epochs=32,
                batch_size=64,
                shuffle=True,
                validation_data=(flattened_x_test_seasoneed, flattened_x_test))
```

```
Epoch 1/32
938/938 [==============================] - 7s 7ms/step - loss: 0.2441 -
val_loss: 0.1544
Epoch 2/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1508 -
val_loss: 0.1403
Epoch 3/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1386 -
val_loss: 0.1344
Epoch 4/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1330 -
val_loss: 0.1307
Epoch 5/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1295 -
val_loss: 0.1285
Epoch 6/32
938/938 [==============================] - 7s 7ms/step - loss: 0.1275 -
```

```
val_loss: 0.1272
Epoch 7/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1253 -
val_loss: 0.1264
Epoch 8/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1243 -
val_loss: 0.1260
Epoch 9/32
938/938 [==============================] - 7s 7ms/step - loss: 0.1232 -
val_loss: 0.1244
Epoch 10/32
938/938 [==============================] - 7s 7ms/step - loss: 0.1221 -
val_loss: 0.1238
Epoch 11/32
938/938 [==============================] - 7s 7ms/step - loss: 0.1211 -
val_loss: 0.1237
Epoch 12/32
938/938 [==============================] - 7s 7ms/step - loss: 0.1206 -
val_loss: 0.1228
Epoch 13/32
938/938 [==============================] - 7s 7ms/step - loss: 0.1201 -
val_loss: 0.1232
Epoch 14/32
938/938 [==============================] - 7s 7ms/step - loss: 0.1193 -
val_loss: 0.1220
Epoch 15/32
938/938 [==============================] - 7s 7ms/step - loss: 0.1188 -
val_loss: 0.1218
Epoch 16/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1187 -
val_loss: 0.1226
Epoch 17/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1183 -
val_loss: 0.1220
Epoch 18/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1176 -
val_loss: 0.1211
Epoch 19/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1175 -
val_loss: 0.1210
Epoch 20/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1169 -
val_loss: 0.1208
Epoch 21/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1170 -
val_loss: 0.1209
Epoch 22/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1167 -
```

```
val_loss: 0.1204
Epoch 23/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1167 -
val_loss: 0.1209
Epoch 24/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1161 -
val_loss: 0.1204
Epoch 25/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1162 -
val_loss: 0.1208
Epoch 26/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1158 -
val_loss: 0.1201
Epoch 27/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1158 -
val_loss: 0.1206
Epoch 28/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1157 -
val_loss: 0.1205
Epoch 29/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1154 -
val_loss: 0.1201
Epoch 30/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1151 -
val_loss: 0.1200
Epoch 31/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1152 -
val_loss: 0.1203
Epoch 32/32
938/938 [==============================] - 6s 7ms/step - loss: 0.1150 -
val_loss: 0.1201
```

#4)3points. Auto-Encodersfor denoising.

**A) The notebook implements a simple denoising deep autoencoder model. Explain what the model does: use the data-preparation and model definition code to explain how the goal of the model is achieved. Explain the role of the loss function? Draw a diagram of the model and include it in your report.Train the model with the settings given.**

[**Answer**]

The goal of this model is to be able to recreate the input images as accurate as possible by denoising the images after they have been compressed and added with noise.

The model is constructed by 4 layers(2 encoders & 2 decoders) and one bottleneck(code) layer in between.

In the encoder layers the images will be compressed and added with noise while trying to maintain sufficient information in order for the decoder to be able to recreate the images.

When entering the decoder layers the model tries to recreate the images with the information it received from the encoder. In this process the noise in the images will be removed and thus, the recreated images revealed.

In the first encoder layer the image have a 784 dimensional vector and then enters the second encoder layer where it reduces to a 128 dimensional vector. In between(bottleneck) the 2 types of layers(encoder & decoder) the image reduces to a 96 dimensional vector. When entering the decoder layers it first increases to a 128 dimensional vector and then enters the last layer where it increases to the original 784 dimensional vector(the recreated image).

The model trains by taking images from the mnist database as inputs and adding noise to them and then running them over the autoencoder to compare the output(recreated image) to the original input(original image).

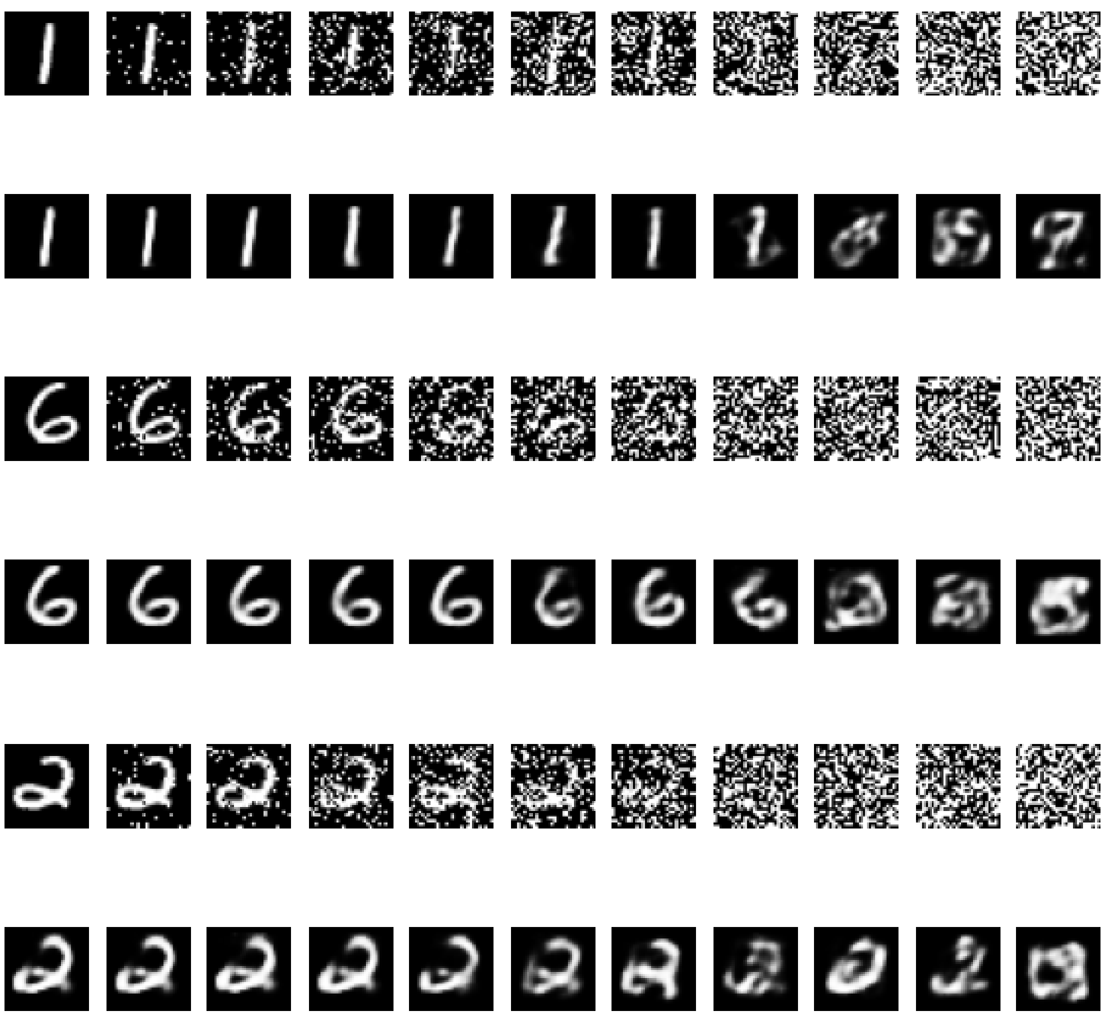
The loss function describes the amount of information loss between the compressed and decompressed image. Thus, comparing the recreated image(denoised image) to the original image(input) to observe the amount of accuracy of the autoencoder. This information helps to update the weights to improve future predictions when back propagating.

**B) Add increasing levels of noise to the test-set using the salt_and_pepper()-function (0 to 1).Use matplotlib to visualize a few examples (3-4) in the original, "seasoned" (noisy), and denoisedversions. (Hint: for visualization use imshow(), use the trained autoencoder from 4A to denoise the noisydigits). At what noise level does it become difficult to identify the seasoned digits for you? At what noise leveldoes the denoising stop working?**

[**Answer**]

[ ]:
```
#noises
noises = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]

#input images
num_images = 3
np.random.seed(42)
random_test_images = np.random.randint(flattened_x_test.shape[0],␣
 ↪size=num_images)

#encoded and decoded images
encoded_imgs = encoder_only.predict(flattened_x_test)
decoded_imgs = autoencoder.predict(flattened_x_test)

#in order to plot images in row/column format style
fig, ax = plt.subplots(nrows=2*len(random_test_images), ncols=len(noises),␣
 ↪figsize=(10,10))

for i, image_idx in enumerate(random_test_images):
  #plot original image
```

```
    ax[2*i][0].imshow(flattened_x_test[image_idx].reshape(28,28))
    ax[2*i][0].axis('off')
    for j, noise in enumerate(noises):
      #plot encoded image with varying noises
      ax[2*i][j].imshow(tf.reshape(salt_and_pepper(flattened_x_test[image_idx].reshape(1, -1),
                                                  noise_level=noise),(28,28)), cmap='gray')
      ax[2*i][j].axis('off')
      #plot reconstructed image
      ax[2*i+1][j].imshow(tf.reshape(autoencoder(salt_and_pepper(flattened_x_test[image_idx].reshape(1, -1),
                                                  noise_level=noise)),(28,28)), cmap='gray')
      ax[2*i+1][j].axis('off')
```



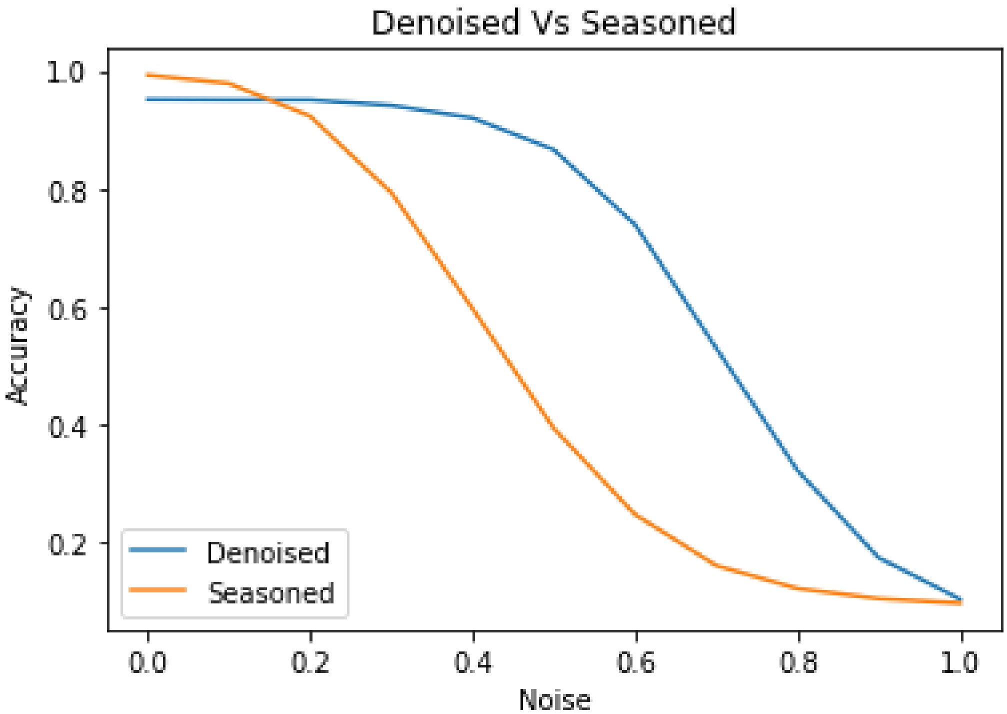
It becomes difficult to identify the digits at noise level 0.7(70%). At the noise level of 0.5(50%) the quality of the recreation images start to decline, so we would say that it is at this point that the denoising stops working.

**C)Test whether denoising improves the classification with the best performing model you obtained in questions 2 or 3. Plot the accuracy as a function of noise-level for the seasoned and denoised datasets. Discuss your results.**

[**Answer**]

[14]:
```
#score = convModel.evaluate(x_test, y_test, verbose=0)
#print("Accuracy: ", score[1])

#lists of accuracy for denoised and seasoned datasets
autoencoded_accuracy_list = []
seasoned_accuracy_list    = []

for i, noise in enumerate(noises):
  #denoised datasets
  x_autoencoded_test = tf.reshape(autoencoder(salt_and_pepper(flattened_x_test,␣
↪noise_level=noise)),(10000,28,28,1))
  #loss and accuracy data
  autoencoded_score    = convModel.evaluate(x_autoencoded_test, y_test,␣
↪verbose=0)
  autoencoded_loss     = autoencoded_score[0]
  autoencoded_accuracy = autoencoded_score[1]
  #add the accuracy to the list
  autoencoded_accuracy_list.append(autoencoded_accuracy)

  #seasoned datasets
  x_seasoned_test = tf.reshape(salt_and_pepper(flattened_x_test,␣
↪noise_level=noise),(10000,28,28,1))
  #loss and accuracy data
  seasoned_score    = convModel.evaluate(x_seasoned_test, y_test, verbose=0)
  seasoned_loss     = seasoned_score[0]
  seasoned_accuracy = seasoned_score[1]
  #add the accuracy to the list
  seasoned_accuracy_list.append(seasoned_accuracy)

#plot
plt.plot(noises, autoencoded_accuracy_list)
plt.plot(noises, seasoned_accuracy_list)
plt.title('Denoised Vs Seasoned')
plt.ylabel('Accuracy')
plt.xlabel('Noise')
plt.legend(['Denoised', 'Seasoned'], loc='lower left')
```

```
plt.show()
```




What we can see from the plot is that the convolutional neural network with denoised input improves the classification drastically compared to the convolutional neural network with seasoned(noised) input, after reaching the noise level of around 0.1(10%). What it also shows us is what we mentioned before, that at noise level 0.5(50%) or even before that the accuracy decreases which is an indication that the denoising has stopped working.

**D) Explain how you can use the decoder part of the denoising auto-encoder to generate synthetic"hand-written" digits? –Describe the procedure, implement it and show examples in your report.**

[**Answer**]

One way could be to train a Generative Adversarial Network(GAN) to generate "fake" hand-written digits. But how we did it here is that we instead of giving the autoencoder the flattened_x_test(original image of the digit) values we gave it the flattened_x_train_seasoned which are the "trained" attempts of recreating the digits, which will look more like handwritten, which in this case can be seen in the images plotted.

[16]:
```
import random
encoded_imgs = encoder_only.predict(flattened_x_test)
decoded_imgs_seasoned = autoencoder.predict(flattened_x_train_seasoned)

for i in range(10): #4
```

```
    plt.subplot(5,10,1+i*2) #2,4
    plt.imshow(flattened_x_train_seasoned[i].reshape(28,28), cmap='gray')
    plt.axis('off')

    plt.subplot(5,10,2+i*2) #2,4
    plt.imshow((decoded_imgs_seasoned[i].reshape(28,28)+1)/2, cmap='gray')
    plt.axis('off')
plt.show()
```