

## In-Lab:

### Lab Task 1:

```
# Import necessary libraries
from keras.models import Sequential
from keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from keras.datasets import mnist
from keras.utils import to_categorical
```

### Lab Task 2:

```
# Load and preprocess the MNIST dataset
(train_images, train_labels), (test_images, test_labels) =
mnist.load_data()
# Reshape and normalize the images
train_images = train_images.reshape((60000, 28, 28,
1)).astype('float32') / 255
test_images = test_images.reshape((10000, 28, 28,
1)).astype('float32') / 255
```

### Lab Task 3:

```
# One-hot encode the labels
train_labels = to_categorical (train_labels)
test_labels = to_categorical (test_labels)
```

### Lab Task 4:

```
# Build the CNN model
model = Sequential()
# Step 1: Convolutional Layer with ReLU activation
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28,
1)))
# Step 2: Max Pooling Layer
model.add(MaxPooling2D((2, 2)))
# Step 3: Convolutional Layer with ReLU activation
model.add(Conv2D(64, (3, 3), activation='relu'))
# Step 4: Max Pooling Layer
model.add(MaxPooling2D((2, 2)))
# Step 5: Flatten Layer
model.add(Flatten())
# Step 6: Dense (Fully Connected) Layer with ReLU activation
model.add(Dense (64, activation='relu'))
# Step 7: Output Layer with Softmax activation (for multi-class
classification)
model.add(Dense (10, activation='softmax'))
```

## Lab Task 5:

```
model.compile(optimizer='adam', loss='categorical_crossentropy',  
metrics=['accuracy'])
```

## Lab Task 6:

```
# Train the model  
model.fit(train_images, train_labels, epochs=5, batch_size=64,  
validation_data=(test_images,  
test_labels))  
  
test_loss, test_acc = model.evaluate(test_images, test_labels)  
print(f'Test Accuracy: {round(test_acc,4)}')
```

## Output:

```
Downloading data from https://storage.googleapis.com/tensorflow/tf-keras-datasets/mnist.npz  
11490434/11490434 [=====] - 0s 0us/step  
Epoch 1/5  
938/938 [=====] - 59s 62ms/step - loss: 0.1709 - accuracy: 0.9492 - val_loss: 0.0608 - val_accuracy: 0.9811  
Epoch 2/5  
938/938 [=====] - 57s 60ms/step - loss: 0.0520 - accuracy: 0.9839 - val_loss: 0.0424 - val_accuracy: 0.9853  
Epoch 3/5  
938/938 [=====] - 53s 57ms/step - loss: 0.0371 - accuracy: 0.9886 - val_loss: 0.0306 - val_accuracy: 0.9896  
Epoch 4/5  
938/938 [=====] - 56s 60ms/step - loss: 0.0284 - accuracy: 0.9911 - val_loss: 0.0323 - val_accuracy: 0.9898  
Epoch 5/5  
938/938 [=====] - 63s 67ms/step - loss: 0.0232 - accuracy: 0.9928 - val_loss: 0.0276 - val_accuracy: 0.9897  
313/313 [=====] - 3s 9ms/step - loss: 0.0276 - accuracy: 0.9897
```

```
313/313 [=====] - 3s 9ms/step - loss: 0.0276 - accuracy: 0.9897  
Test Accuracy: 0.9897
```