IMPORTING THE PACKAGES

```
import numpy
import matplotlib.pyplot as plt
from keras.utils import np_utils
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, Flatten
from tensorflow.keras.optimizers import Adam
```

LOADING THE DATA

```
(X_train, y_train), (X_test, y_test) = mnist.load_data()
```

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

ANALYZING THE DATA

```
print(X_train.shape)
print(X_test.shape)
```

```
(60000, 28, 28)
(10000, 28, 28)
```

```
X_train[0]
```

```
array([[  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
```

```
          0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
3,
         18,  18,  18, 126, 136, 175,  26, 166, 255, 247, 127,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  30,  36,  94, 154,
170,
        253, 253, 253, 253, 253, 225, 172, 253, 242, 195,  64,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  49, 238, 253, 253, 253,
253,
        253, 253, 253, 253, 251,  93,  82,  82,  56,  39,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,  18, 219, 253, 253, 253,
253,
        253, 198, 182, 247, 241,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,  80, 156, 107, 253,
253,
        205,  11,   0,  43, 154,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,  14,   1, 154,
253,
         90,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0, 139,
253,
        190,   2,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  11,
190,
        253,  70,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
35,
        241, 225, 160, 108,   1,   0,   0,   0,   0,   0,   0,   0,
0,
          0,   0],
       [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
```

```
          81, 240, 253, 253, 119,  25,   0,   0,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,  45, 186, 253, 253, 150,  27,   0,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0,  16,  93, 252, 253, 187,   0,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0,   0,   0, 249, 253, 249,  64,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,  46, 130, 183, 253, 253, 207,   2,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
39,
         148, 229, 253, 253, 253, 250, 182,   0,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,  24, 114,
221,
         253, 253, 253, 253, 201,  78,   0,   0,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,   0,   0,   0,   0,  23,  66, 213, 253,
253,
         253, 253, 198,  81,   2,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,   0,   0,  18, 171, 219, 253, 253, 253,
253,
         195,  80,   9,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0,  55, 172, 226, 253, 253, 253, 253, 244,
133,
          11,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0],
        [  0,   0,   0,   0, 136, 253, 253, 253, 212, 135, 132,  16,
0,
```

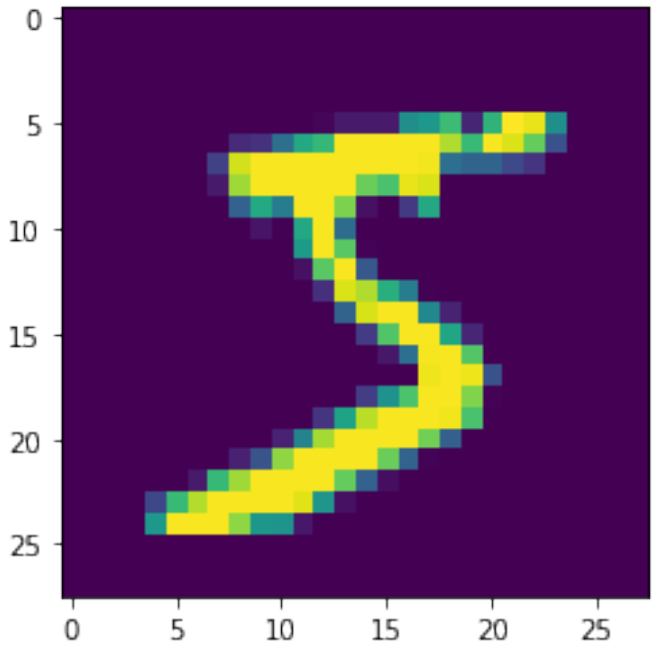
```
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0],
       [   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0],
       [   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0],
       [   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
0,
           0,   0]], dtype=uint8)
```

```
y_train[0]
```

```
5
```

```
plt.imshow(X_train[0])
```

```
<matplotlib.image.AxesImage at 0x7ff3c19c1ad0>
```



DATA PREPROCESSING

```
X_train = X_train.reshape(60000, 28, 28, 1).astype('float32')
X_test = X_test.reshape(10000, 28, 28, 1).astype('float32')

number_of_classes = 10
Y_train = np_utils.to_categorical(y_train, number_of_classes)
Y_test = np_utils.to_categorical(y_test, number_of_classes)


Y_train[0]

array([0., 0., 0., 0., 0., 1., 0., 0., 0., 0.], dtype=float32)
```

MODEL CREATING

```
model = Sequential()
model.add(Conv2D(64, (3, 3), input_shape=(28, 28, 1),
activation="relu"))
model.add(Conv2D(32, (3, 3), activation="relu"))
model.add(Flatten())
model.add(Dense(number_of_classes, activation="softmax"))


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

TRAIN THE MODEL

```
model.fit(X_train, Y_train, batch_size=32, epochs=5,
validation_data=(X_test,Y_test))

Epoch 1/5
1875/1875 [==============================] - 118s 63ms/step - loss:
0.2368 - accuracy: 0.9489 - val_loss: 0.1134 - val_accuracy: 0.9674
Epoch 2/5
1875/1875 [==============================] - 113s 60ms/step - loss:
0.0715 - accuracy: 0.9783 - val_loss: 0.0835 - val_accuracy: 0.9736
Epoch 3/5
1875/1875 [==============================] - 117s 62ms/step - loss:
0.0495 - accuracy: 0.9841 - val_loss: 0.0809 - val_accuracy: 0.9746
Epoch 4/5
1875/1875 [==============================] - 115s 61ms/step - loss:
0.0350 - accuracy: 0.9891 - val_loss: 0.0947 - val_accuracy: 0.9729
Epoch 5/5
1875/1875 [==============================] - 114s 61ms/step - loss:
0.0303 - accuracy: 0.9905 - val_loss: 0.0976 - val_accuracy: 0.9768

<keras.callbacks.History at 0x7ff3bd2365d0>
```

TEST THE MODEL

```
metrics = model.evaluate(X_test, Y_test, verbose=0)
print("Metrics (Test Loss & Test Accuracy): ")
print(metrics)
```

```
Metrics (Test Loss & Test Accuracy):
[0.09761594235897064, 0.9768000245094299]
```

```
prediction = model.predict(X_test[:4])
print(prediction)
```

```
1/1 [==============================] - 0s 109ms/step
[[1.6083676e-07 5.0151581e-22 4.0418258e-10 8.5343224e-08 2.2995244e-
15
  4.4694977e-15 1.0798898e-21 9.9999976e-01 1.8214679e-09 4.3404907e-
10]
 [8.1183879e-11 2.2536082e-11 1.0000000e+00 2.8008044e-16 2.5865686e-
16
  2.1337134e-19 1.6760615e-12 1.4822324e-18 2.7226044e-11 1.4262518e-
21]
 [3.7191185e-06 9.9621898e-01 1.6017430e-03 5.7848634e-08 1.8739940e-
03
  3.7856381e-08 5.8028988e-08 2.4127127e-07 3.0112482e-04 2.3934070e-
09]
 [1.0000000e+00 2.5093286e-19 8.1620119e-12 1.0207498e-13 6.5025548e-
17
  1.5716776e-17 7.4558651e-11 6.5080705e-15 1.0112380e-11 8.3561949e-
11]]
```

```
print(numpy.argmax(prediction, axis=1))
print(Y_test[:4])
```

```
[7 2 1 0]
[[0. 0. 0. 0. 0. 0. 0. 1. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 1. 0. 0. 0. 0. 0. 0. 0. 0.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 0.]]
```