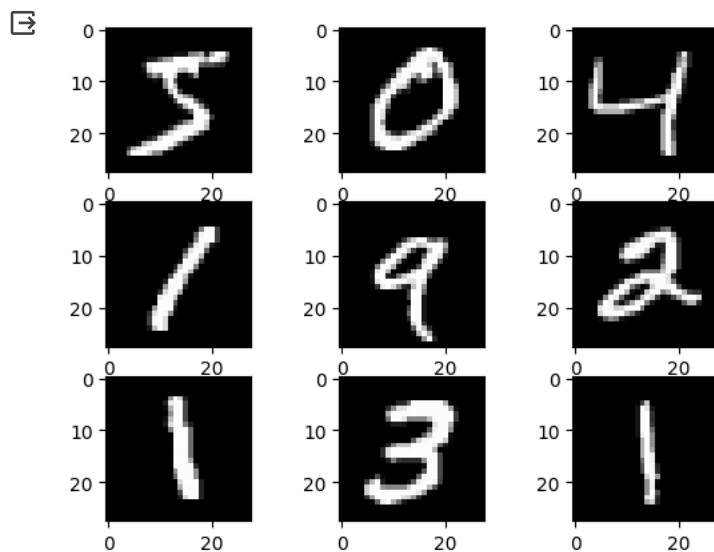


```
import numpy as np
import os
```

```
from tensorflow.keras.datasets import mnist
from matplotlib import pyplot as plt
(trainX, trainy), (testX, testy) = mnist.load_data()
print('Train: X=%s, y=%s' % (trainX.shape, trainy.shape))
print('Test: X=%s, y=%s' % (testX.shape, testy.shape))
```

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

```
for i in range(9):
    plt.subplot(330 + 1 + i)
    plt.imshow(trainX[i], cmap=plt.get_cmap('gray'))
data = ...
```



```
import matplotlib.pyplot as plt
import seaborn as sns
#import cv2
from PIL import Image
import tensorflow as tf
tf.random.set_seed(3)
from tensorflow import keras
from keras.datasets import mnist
from tensorflow.math import confusion_matrix
import random

(X_train,Y_train),(X_test,Y_test) = mnist.load_data()

type(X_train)

numpy.ndarray

print(X_train.shape,Y_train.shape,X_test.shape,Y_test.shape)

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

print(X_train[9])
```

```
[[ 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  189 190  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 0 143 247 153 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 136 247 242 86 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 192 252 187 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 62 185 18 0 0 0
0 89 236 217 47 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 216 253 60 0 0 0
0 212 255 81 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 0 206 252 68 0 0 0
48 242 253 89 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 0 0 131 251 212 21 0 0 11
167 252 197 5 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 29 232 247 63 0 0 0 153
252 226 0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 45 219 252 143 0 0 0 116 249
252 103 0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 4 96 253 255 253 200 122 7 25 201 250
158 0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 92 252 252 253 217 252 252 200 227 252 231
0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 87 251 247 231 65 48 189 252 252 253 252 251
227 35 0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 190 221 98 0 0 0 42 196 252 253 252 252
162 0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 111 29 0 0 0 62 239 252 86 42 42
14 0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 15 148 253 218 0 0 0
0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 0 121 252 231 28 0 0 0
0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 31 221 251 129 0 0 0 0
0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 218 252 160 0 0 0 0 0
0 0 0 0 0 0 0 0]
[ 0 0 0 0 0 0 0 0 0 0 122 252 82 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]

```

```
print(X_train[9].shape)
```

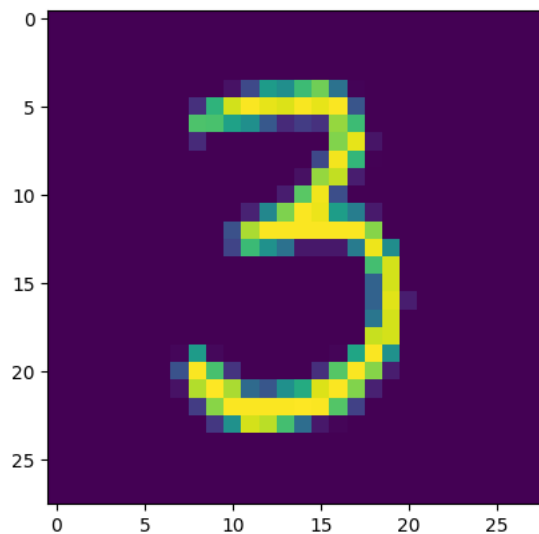
```
(28, 28)
```

```
plt.imshow(X_train[50])
```

```
plt.show
```

```
print(Y_train[50])
```

```
3
```



```
print(Y_train.shape,Y_test.shape)
```

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



```
model.fit(X_train, Y_train, epochs = 10)
```

```
Epoch 1/10
1875/1875 [=====] - 6s 3ms/step - loss: 0.2876 - accuracy: 0.9163
Epoch 2/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.1363 - accuracy: 0.9594
Epoch 3/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.1021 - accuracy: 0.9693
Epoch 4/10
1875/1875 [=====] - 5s 3ms/step - loss: 0.0837 - accuracy: 0.9743
Epoch 5/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.0678 - accuracy: 0.9790
Epoch 6/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.0594 - accuracy: 0.9812
Epoch 7/10
1875/1875 [=====] - 5s 3ms/step - loss: 0.0510 - accuracy: 0.9836
Epoch 8/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.0470 - accuracy: 0.9848
Epoch 9/10
1875/1875 [=====] - 4s 2ms/step - loss: 0.0401 - accuracy: 0.9868
Epoch 10/10
1875/1875 [=====] - 5s 3ms/step - loss: 0.0339 - accuracy: 0.9889
<keras.src.callbacks.History at 0x7deb6d454340>
```

```
INDEX = random.randint(0,Y_test.shape[0])
actual = Y_test[INDEX]
pred = np.argmax(model.predict(np.array([X_test[INDEX]])))[0]
pred,actual
plt.imshow(X_test[INDEX])
plt.title(f'actual label : {actual}, predict label : {pred}')
plt.show()
```

```
1/1 [=====] - 0s 96ms/step
```

