```
import tensorflow as tf
from tensorflow.keras.utils import to_categorical
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense, Dropout
# https://en.wikipedia.org/wiki/MNIST_database
mnist = tf.keras.datasets.mnist
# Read the data
(x_train, y_train), (x_test, y_test) = mnist.load_data()
# Scale it to values 0 - 1
x_train = x_train / 255.0
x_test = x_test / 255.0
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
x_train = x_train.reshape(x_train.shape[0], x_train.shape[1], x_train.shape[2], 1)
x_test = x_test.reshape(x_test.shape[0], x_test.shape[1], x_test.shape[2], 1)
# Creating a model
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(.5))
model.add(Dense(10, activation='softmax'))
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
model.fit(x_train, y_train, epochs=10)
model.evaluate(x_test, y_test)
```

```
Epoch 1/10
1875/1875 [==============================] - 39s 21ms/step - loss: 0.2595 - accuracy: 0.9217
Epoch 2/10
1875/1875 [==============================] - 39s 21ms/step - loss: 0.1064 - accuracy: 0.9681
Epoch 3/10
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0829 - accuracy: 0.9754
Epoch 4/10
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0669 - accuracy: 0.9794
Epoch 5/10
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0556 - accuracy: 0.9828
Epoch 6/10
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0462 - accuracy: 0.9851
Epoch 7/10
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0429 - accuracy: 0.9865
Epoch 8/10
1875/1875 [==============================] - 39s 21ms/step - loss: 0.0371 - accuracy: 0.9879
Epoch 9/10
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0322 - accuracy: 0.9897
Epoch 10/10
1875/1875 [==============================] - 38s 20ms/step - loss: 0.0295 - accuracy: 0.9900
313/313 [==============================] - 2s 5ms/step - loss: 0.0384 - accuracy: 0.9886
[0.03842239826917648, 0.9886000156402588]
```

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

```
Test loss: 0.03842239826917648
Test accuracy: 0.9886000156402588
```

```
predict_x=model.predict(x_test)
classes_x=np.argmax(predict_x,axis=1)
```

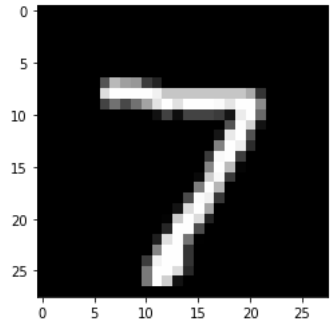
```
313/313 [==============================] - 3s 9ms/step
```

```
actual_x = classes_x=np.argmax(y_test,axis=1)
```

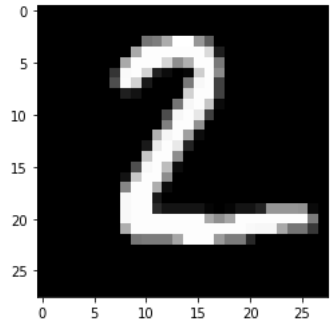
Show code

Show code

```
Predicted :  7 Actual :  7
```



```
Predicted :  2 Actual :  2
```



```
Predicted :  1 Actual :  1
```