

```
In [1]: import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: import tensorflow as tf
from tensorflow.keras.datasets import mnist
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from tensorflow.keras.utils import to_categorical
```

## load and split dataset for train test


```
In [3]: (X_train, y_train), (X_test, y_test) = mnist.load_data()
X_train = X_train.reshape(-1, 28, 28, 1) / 255.0
X_test = X_test.reshape(-1, 28, 28, 1) / 255.0
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```


## Model Training


```
In [4]: model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D((2, 2)),
    Conv2D(64, (3, 3), activation='relu'),
    Flatten(),
    Dense(64, activation='relu'),
    Dense(10, activation='softmax')
])


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


model.fit(X_train, y_train, batch_size=64, epochs=10, verbose=1)
```


Epoch 1/10  
938/938  15s 13ms/step - accuracy: 0.8667 - loss: 0.4304


Epoch 2/10  
938/938  14s 15ms/step - accuracy: 0.9825 - loss: 0.0567


Epoch 3/10  
938/938  16s 17ms/step - accuracy: 0.9876 - loss: 0.0379


Epoch 4/10  
938/938  16s 17ms/step - accuracy: 0.9923 - loss: 0.0258


Epoch 5/10  
938/938  16s 17ms/step - accuracy: 0.9941 - loss: 0.0205

Epoch 6/10  
938/938  16s 17ms/step - accuracy: 0.9947 - loss: 0.0177

Epoch 7/10  
938/938  16s 17ms/step - accuracy: 0.9955 - loss: 0.0144

Epoch 8/10  
938/938  15s 16ms/step - accuracy: 0.9966 - loss: 0.0116

Epoch 9/10  
938/938  16s 17ms/step - accuracy: 0.9970 - loss: 0.0090

Epoch 10/10  
938/938  16s 17ms/step - accuracy: 0.9972 - loss: 0.0089

Out[4]: <keras.src.callbacks.history.History at 0x190a9af0dc0>

## Evaluation of Model

```
In [5]: loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Loss: {loss}")
print(f"Test Accuracy: {accuracy}")
```

313/313 ————— 1s 4ms/step - accuracy: 0.9899 - loss: 0.0359  
Test Loss: 0.026699405163526535  
Test Accuracy: 0.9922999739646912

In [ ]: