```
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, Dropout
from tensorflow.keras.utils import to_categorical
import matplotlib.pyplot as plt
```

```

from tensorflow.keras.datasets import mnist
from tensorflow.keras.utils import to_categorical

(x_train, y_train), (x_test, y_test) = mnist.load_data()
x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0

x_train = x_train.reshape((x_train.shape[0], 28, 28, 1))
x_test = x_test.reshape((x_test.shape[0], 28, 28, 1))

y_train = to_categorical(y_train, 10)
y_test = to_categorical(y_test, 10)
print()
```

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dropout, Flatten, Dense

model = Sequential([
    Conv2D(32, kernel_size=(3, 3), activation='relu', input_shape=(28, 28, 1)),
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.25),
    Conv2D(64, kernel_size=(3, 3), activation='relu'),
    MaxPooling2D(pool_size=(2, 2)),
    Dropout(0.25),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(10, activation='softmax')
])

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

history = model.fit(x_train, y_train, validation_split=0.1,
                    epochs=5, batch_size=128, verbose=2)

test_loss, test_acc = model.evaluate(x_test, y_test, verbose=2)
print('Test accuracy: {:.4f}'.format(test_acc))
```

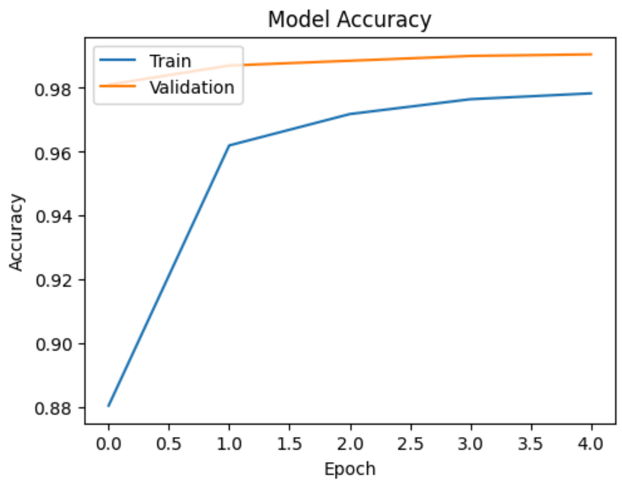
```
/usr/local/lib/python3.12/dist-packages/keras/src/layers/convolutional/base_conv.py:113: UserWarning: Do not pass an `input_
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
Epoch 1/5
422/422 - 43s - 102ms/step - accuracy: 0.8806 - loss: 0.3836 - val_accuracy: 0.9810 - val_loss: 0.0709
Epoch 2/5
422/422 - 81s - 192ms/step - accuracy: 0.9620 - loss: 0.1277 - val_accuracy: 0.9870 - val_loss: 0.0498
Epoch 3/5
422/422 - 45s - 106ms/step - accuracy: 0.9718 - loss: 0.0933 - val_accuracy: 0.9885 - val_loss: 0.0392
Epoch 4/5
422/422 - 42s - 100ms/step - accuracy: 0.9765 - loss: 0.0790 - val_accuracy: 0.9900 - val_loss: 0.0354
Epoch 5/5
422/422 - 80s - 190ms/step - accuracy: 0.9783 - loss: 0.0716 - val_accuracy: 0.9905 - val_loss: 0.0349
313/313 - 2s - 8ms/step - accuracy: 0.9899 - loss: 0.0314
Test accuracy: 0.9899
```

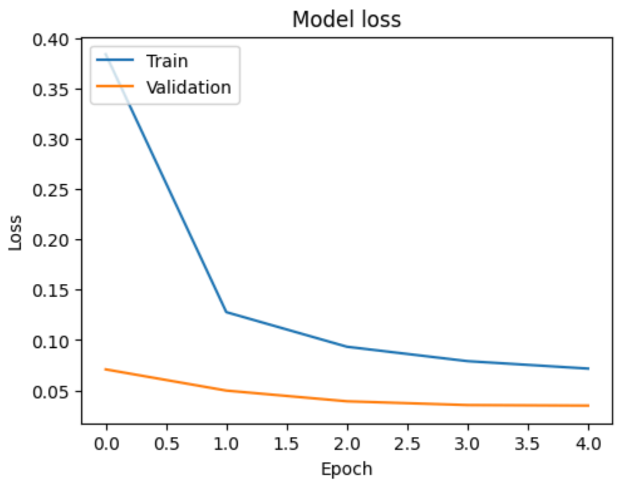
```

print("="*70)
print("MODEL")
model.summary()

plt.figure(figsize=(12,4))
plt.subplot(1,2,1)
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train','Validation'],loc='upper left')
plt.subplot(1,2,2)
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train','Validation'],loc='upper left')
plt.show()
```

```
======================================================================
MODEL
Model: "sequential"
```

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 26, 26, 32)	320
max_pooling2d (MaxPooling2D)	(None, 13, 13, 32)	0
dropout (Dropout)	(None, 13, 13, 32)	0
conv2d_1 (Conv2D)	(None, 11, 11, 64)	18,496
max_pooling2d_1 (MaxPooling2D)	(None, 5, 5, 64)	0
dropout_1 (Dropout)	(None, 5, 5, 64)	0
flatten (Flatten)	(None, 1600)	0
dense (Dense)	(None, 128)	204,928
dropout_2 (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 10)	1,290

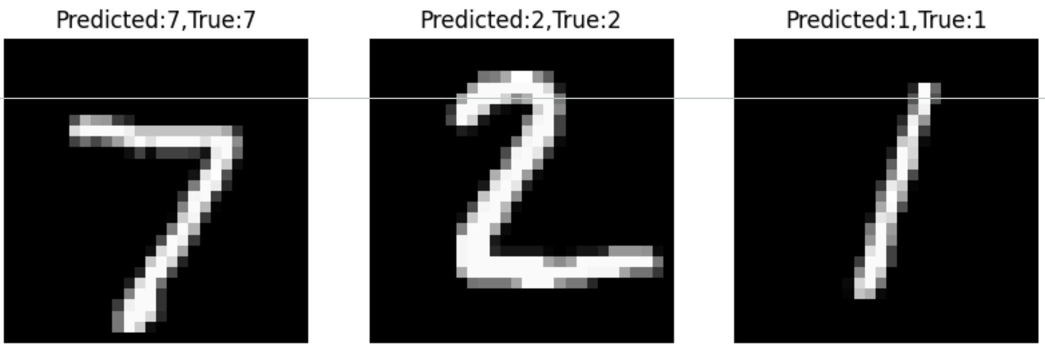
```
Total params: 675,104 (2.58 MB)
Trainable params: 225,034 (879.04 KB)
Non-trainable params: 0 (0.00 B)
Optimizer params: 450,070 (1.72 MB)
```





```
import numpy as np
predictions=model.predict(x_test)
predicted_labels=[tf.argmax(tf.convert_to_tensor(prediction)).numpy()
for prediction in predictions]
plt.figure(figsize=(10,4))
num_images=3
for i in range(num_images):
  plt.subplot(1,num_images,i+1)
  plt.imshow(x_test[i].reshape(28,28),cmap='gray')
  plt.title(f"Predicted:{predicted_labels[i]},True:{np.argmax(y_test[i])}")
  plt.axis('off')
plt.show()
```

```
313/313 ━━━━━━━━━━━━━━━━━━━━ 2s 7ms/step
```


Predicted:7,True:7
Predicted:2,True:2
Predicted:1,True:1