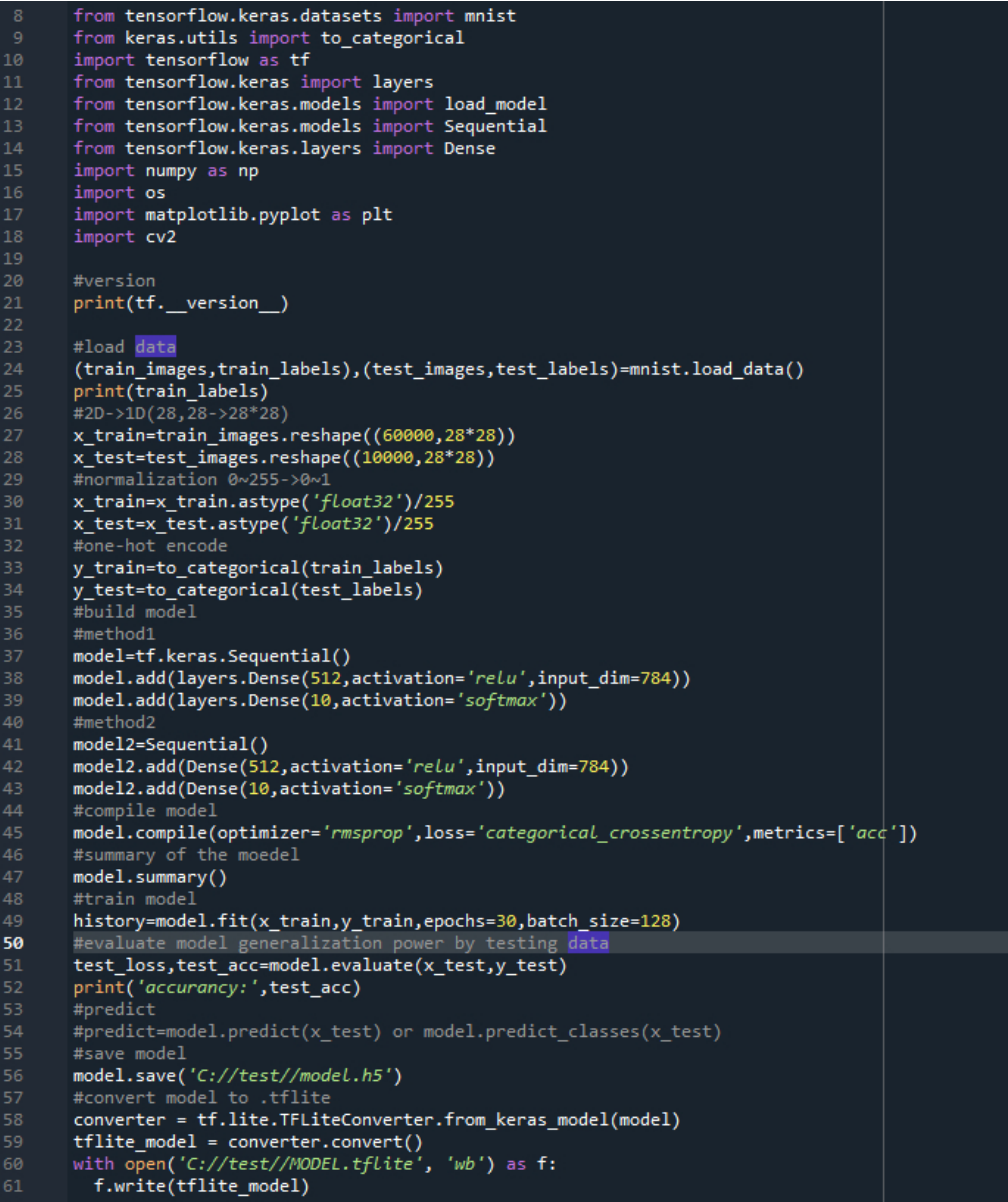
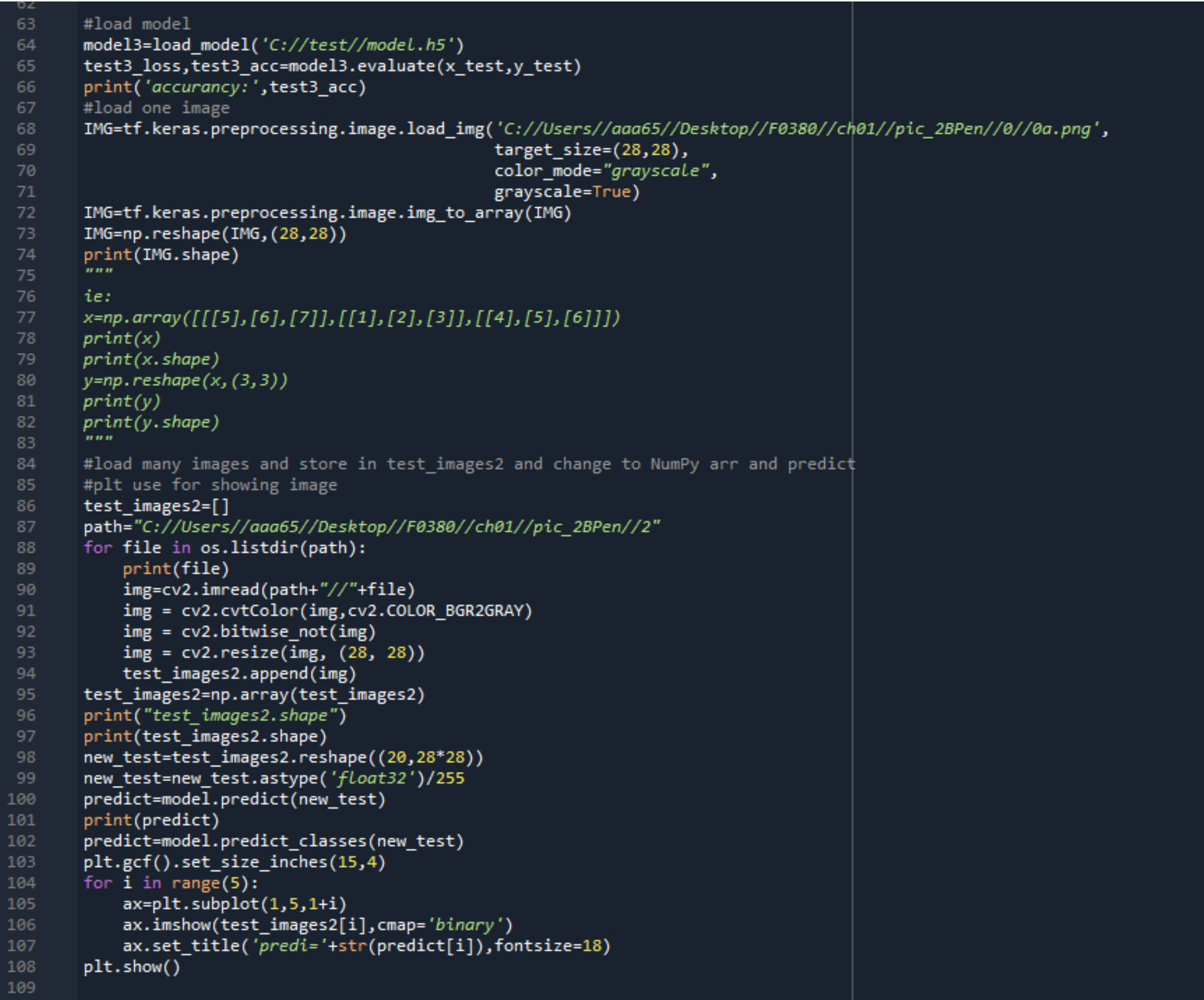


```python
from tensorflow.keras.datasets import mnist
from keras.utils import to_categorical
import tensorflow as tf
from tensorflow.keras import layers
from tensorflow.keras.models import load_model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
import numpy as np
import os
import matplotlib.pyplot as plt
import cv2

#version
print(tf.__version__)

#load data
(train_images,train_labels),(test_images,test_labels)=mnist.load_data()
print(train_labels)
#2D->1D(28,28->28*28)
x_train=train_images.reshape((60000,28*28))
x_test=test_images.reshape((10000,28*28))
#normalization 0~255->0~1
x_train=x_train.astype('float32')/255
x_test=x_test.astype('float32')/255
#one-hot encode
y_train=to_categorical(train_labels)
y_test=to_categorical(test_labels)
#build model
#method1
model=tf.keras.Sequential()
model.add(layers.Dense(512,activation='relu',input_dim=784))
model.add(layers.Dense(10,activation='softmax'))
#method2
model2=Sequential()
model2.add(Dense(512,activation='relu',input_dim=784))
model2.add(Dense(10,activation='softmax'))
#compile model
model.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['acc'])
#summary of the moedel
model.summary()
#train model
history=model.fit(x_train,y_train,epochs=30,batch_size=128)
#evaluate model generalization power by testing data
test_loss,test_acc=model.evaluate(x_test,y_test)
print('accurancy:',test_acc)
#predict
#predict=model.predict(x_test) or model.predict_classes(x_test)
#save model
model.save('C://test//model.h5')
#convert model to .tflite
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()
with open('C://test//MODEL.tflite', 'wb') as f:
  f.write(tflite_model)
```



```
#load model
model3=load_model('C://test//model.h5')
test3_loss,test3_acc=model3.evaluate(x_test,y_test)
print('accurancy:',test3_acc)
#load one image
IMG=tf.keras.preprocessing.image.load_img('C://Users//aaa65//Desktop//F0380//ch01//pic_2BPen//0//0a.png',
                                          target_size=(28,28),
                                          color_mode="grayscale",
                                          grayscale=True)
IMG=tf.keras.preprocessing.image.img_to_array(IMG)
IMG=np.reshape(IMG,(28,28))
print(IMG.shape)
"""
ie:
x=np.array([[[5],[6],[7]],[[1],[2],[3]],[[4],[5],[6]]])
print(x)
print(x.shape)
y=np.reshape(x,(3,3))
print(y)
print(y.shape)
"""
#load many images and store in test_images2 and change to NumPy arr and predict
#plt use for showing image
test_images2=[]
path="C://Users//aaa65//Desktop//F0380//ch01//pic_2BPen//2"
for file in os.listdir(path):
    print(file)
    img=cv2.imread(path+"//"+file)
    img = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    img = cv2.bitwise_not(img)
    img = cv2.resize(img, (28, 28))
    test_images2.append(img)
test_images2=np.array(test_images2)
print("test_images2.shape")
print(test_images2.shape)
new_test=test_images2.reshape((20,28*28))
new_test=new_test.astype('float32')/255
predict=model.predict(new_test)
print(predict)
predict=model.predict_classes(new_test)
plt.gcf().set_size_inches(15,4)
for i in range(5):
    ax=plt.subplot(1,5,1+i)
    ax.imshow(test_images2[i],cmap='binary')
    ax.set_title('predi='+str(predict[i]),fontsize=18)
plt.show()
```