```python
from keras.datasets import mnist


(train_data,train_target),(test_data,test_target)=mnist.load_data()


from keras.utils import np_utils
num_classes=10
new_train_target=np_utils.to_categorical(train_target,num_classes)
new_test_target=np_utils.to_categorical(test_target,num_classes)


train_data=train_data.reshape(train_data.shape[0],28,28,1)
test_data=test_data.reshape(test_data.shape[0],28,28,1)


input_shape=(28,28,1)


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


model = Sequential()
model.add(Conv2D(32,(3,3),activation='relu', input_shape=input_shape))
model.add(Conv2D(64,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(256,activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes,activation='softmax'))


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


 model.fit(train_data,new_train_target,epochs=10,validation_data=(test_data,new_test_target),
 batch_size=128)


model.evaluate(new_test_data,new_test_target)


import numpy as np
pred = model.predict(test_data[:5])
print(np.argmax(pred,axis=1))
```

```python
from matplotlib import pyplot as plt
for i in range (0,5):
  im=new_test_data[i].reshape(28,28)
  plt.imshow(im,cmap='gray')
  plt.show()
```