Name	Id
Gasser ahmed mohamed	20190150
Youssef salah fathy	20190636

Source code:

```
# Import Libraries
import tensorflow as tf
import tensorflow_datasets as tfds
import numpy as np
from matplotlib import pyplot as plt
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score


#get train & test data
mnist = tf.keras.datasets.mnist

(train_images, train_labels), (test_images, test_labels) = mnist.load_data()

print("Training Data{}".format(train_images.shape ))
print("Test Data{} ".format(test_images.shape ))

def display_img(mnist_index):

    image = mnist_index
    image = np.array(image, dtype='float')
    pixels = image.reshape((28, 28))


display_img(train_images[10])
train_labels[10]
display_img(test_images[3])

#cutting image into images
def imaged_grid(img , row , col ):

    x , y= img.shape
    assert x % row == 0, x % row .format(x, row)
    assert y % col == 0, y % col.format(y, col)
    return (img.reshape ( x //row, row, -1, col)
                .swapaxes(1,2)
                .reshape(-1, row, col))


imaged_grid(test_images[1] , 7 , 7 )

#get centroid of each image(extract features)
def get_centroid(img):

    feature_vector = []

    for grid in imaged_grid(img , 7 , 7 ) :

        X_center = 0
        Y_center = 0
        summtion = 0

        for index, x in np.ndenumerate(grid):
          summtion+= x
          X_center += x * index[0]
          Y_center += x * index[1]

        if summtion == 0 :
            feature_vector.append(0)
            feature_vector.append(0)
        else :
          feature_vector.append( X_center/ summtion )
          feature_vector.append(Y_center/ summtion )

    return np.array(feature_vector)

train_features = [get_centroid(img)  for img in train_images  ]

train_features = np.array(train_features)

train_features.shape

train_features[:2]

test_features = [get_centroid(img)  for img in test_images  ]


test_features = np.array(test_features)

test_features.shape

test_features[:2]
```

```
#classify featuers by KNN
def KNN(train_features, test_features, train_labels):

    knn = KNeighborsClassifier(50, metric='euclidean')

    #fitting data
    knn.fit(train_features, train_labels)
    prediction = knn.predict(test_features)
    return prediction

Knn_prediction = KNN(train_features, test_features , train_labels )

print("Accuracy=", accuracy_score(test_labels, Knn_prediction) * 100, "%")
```

Report:

KNN with getting centroid of each square after dividing the image into squares 28*28:
i note that accuracy changing by changing of the value of hyper parameter k.
at k=1 ---> Accuracy = 91.7
at k=2 ---> Accuracy = 90.9
at k=3 ---> Accuracy = 91.4
at k=4 ---> Accuracy = 91.6
at k=5 ---> Accuracy = 91.3
at k=6 ---> Accuracy = 91.26
at k=7 ---> Accuracy = 91.16
at k=8 ---> Accuracy = 90.92
at k=9 ---> Accuracy = 90.76