Assignment 3
Student:Irakli Kelbakiani

Deadline: Tuesday, March 15, 2022 (end of day)

Overview

This assignment is to write a Python program that can classify handwritten digits from the MNIST dataset: https://ilias.unibe.ch/goto_ilias3_unibe_file_2381417_download.html. Build the entire pipeline, including preprocessing, feature extraction, classification, and evaluation. Use a GitHub repository for your project.

Feature Extraction and Classifier

(a) Define (at least) 6 features that you consider useful to classify the digits in MNIST. Explain each feature briefly and why you chose them.

Pixel values, mean of all pixels, histogram of gradients, vertical Prewitt for edge detection, horizontal Prewitt for edge detection, compactness, center of gravity, central moments, hp-mean, hpd-stdev, hr-stdev, vr-mean, vr-stdev

(b) Implement (at least) two features with the help of a Python program.

Implemented calculation of mean pixel values (mean value for each digit), also classifier using distance between pixel values of all digits and tested image, calculated HOG ,vertical Prewitt, horizontal prewitt

(c) As a classifier, use a basic approach: create for each class a class representative, e.g., mean of the features over the whole class, and then calculate the distance between each sample and these class representatives. Assign the class of the closest representative to the sample.

Implemented aforementioned approaches.

(d) Add (at least) one additional classifier to your o your Python program: k-nearest neighbors classifier (k-NN), support vector machine classifier (SVM), and multi-layer perceptron classifier (MLP). We recommend sklearn.neighbors.KNeighborsClassifier, sklearn.svm.LinearSVC, and sklearn.neural_network.MLPClassifier.

Used K-NN

(e) Implement an evaluation measure: an accuracy metric which presents the correct predictions of your network divided by the total amount:

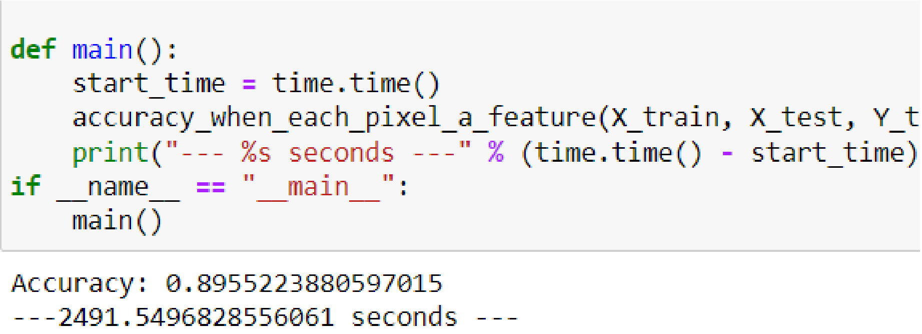
Implemented metric.

$$\text{accuracy} = \frac{\text{correct predication}}{\text{total samples}}$$

Submit the following items via ILIAS:

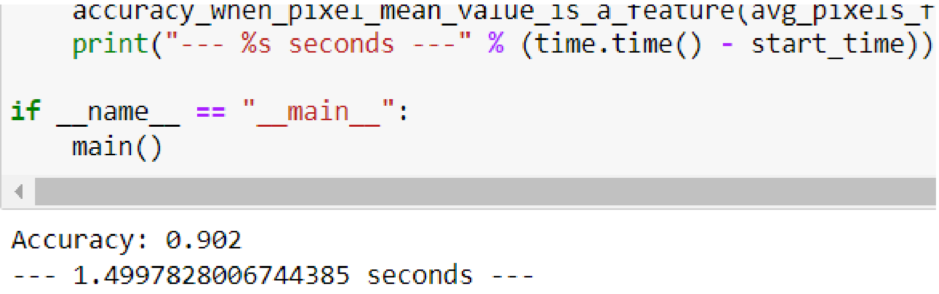
- link to your GitHub repository: https://github.com/KelbakianiIrakli/DIA_HW/tree/DIA-Task3
- brief descriptions of your features: Each pixel of the image, mean value of all pixels for each digit, calculated HOG ,vertical Prewitt, horizontal prewitt.
- table with your results

	Simple Classifier		KNN	
	Accuracy	Runtime	Accuracy	Runtime
Each Pixel Val from Grey Pic	89%	2491.5s	93.8%	15.6s
Pixel Mean Val	90.2%	1.4s		
Each Pixel Val from Prewitt horizontal	88%	9.9s		



```
def main():
    start_time = time.time()
    accuracy_when_each_pixel_a_feature(X_train, X_test, Y_t
    print("--- %s seconds ---" % (time.time() - start_time)
if __name__ == "__main__":
    main()
```

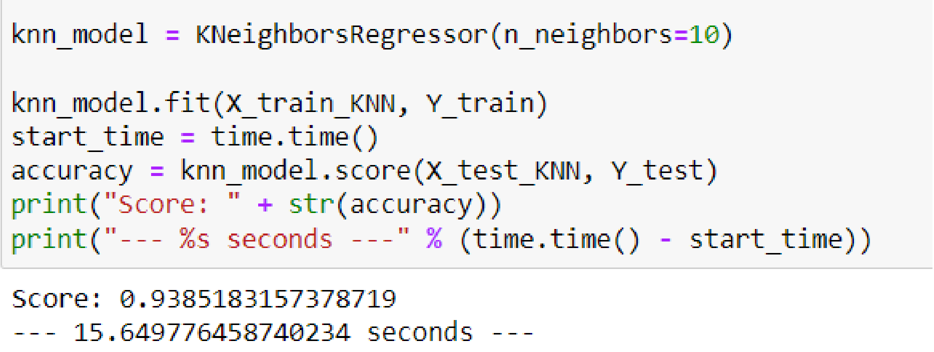
```
Accuracy: 0.8955223880597015
---2491.5496828556061 seconds ---
```



```
    accuracy_when_pixel_mean_value_is_a_feature(avg_pixels_f
    print("--- %s seconds ---" % (time.time() - start_time))

if __name__ == "__main__":
    main()
```

```
Accuracy: 0.902
--- 1.4997828006744385 seconds ---
```



```
knn_model = KNeighborsRegressor(n_neighbors=10)

knn_model.fit(X_train_KNN, Y_train)
start_time = time.time()
accuracy = knn_model.score(X_test_KNN, Y_test)
print("Score: " + str(accuracy))
print("--- %s seconds ---" % (time.time() - start_time))
```

```
Score: 0.9385183157378719
--- 15.649776458740234 seconds ---
```