Cairo University
Faculty of Engineering
Credit Hour Programs

جـــــامعة القـــاهـرة
كـلـيـــــة الـهـنـدســـة
برامـــج الساعـات المعتمـدة

Pattern Recognition - CCEN

Writer Identification System

Project Report

Date:19/01/2021

Amr Afifi	1162377
Mostafa Sherif	1162153
Karim Ibrahim	1162328
Youssef Sayed	11617065

Workload:

Amr Afifi : Literature Survey ,Feature Selection ,Code Optimization

Mostafa Sherif : Preprocessing and Segmentation

Karim Ibrahim : Pipeline Construction and Integration

Youssef Sayed : Dataset Manipulation and Classification

Table Of Contents

(a)Project Pipeline:

1. Reading the images into a grayscale format
2. Binarizing the image using OTSU global threshold
3. Segmenting the image into handwritten and text blocks
4. Morphological operations on the lines to prepare it for line segmentation
5. Segmenting the lines of the handwritten paragraph
6. Extracting the features fo each line using LBP and computing average values
7. Using feature vectors to train the SVM model
8. Loading the test set and predicting the results
9. Calculating the accuracy and time and printing the results in output files

(b)Preprocessing Module:

Applied Deskewing and adjust rotation of image but then removed after examining the dataset and finding there is no need for it. And that the dataset is correctly oriented.

- Binarization of Images using OTSU global threshold
- Connected Component Analysis using scki-image region props to segment between text data and handwritten data

(We used Hough lines but it gave us a little bit less correct results in segmentation)

- Line Segmentation using morphological operations and finding bounding boxes with specific requirements calculations like average area and height of the line.
- Word Segmentation was applied but then removed due to the low contribution of its features in the accuracy while taking more time.

(c)Feature Extraction/Selection Module:

Our main feature that was recommended by many papers that results in a high accuracy is the LBP

We implemented the LBP on line components and we tried implementing it on the texture block for achieving better accuracy and faster execution time.

We also tried some features live average word length and height relative to the text data, Average blob size and pattern, Average gaps between lines and words, Slant angle, bounding lines, but none of them gave us the required accuracy relative to the performance of the LBP.

(d)Model Selection/Training Module:

- All papers recommended the SVM Model with the LBP combination.

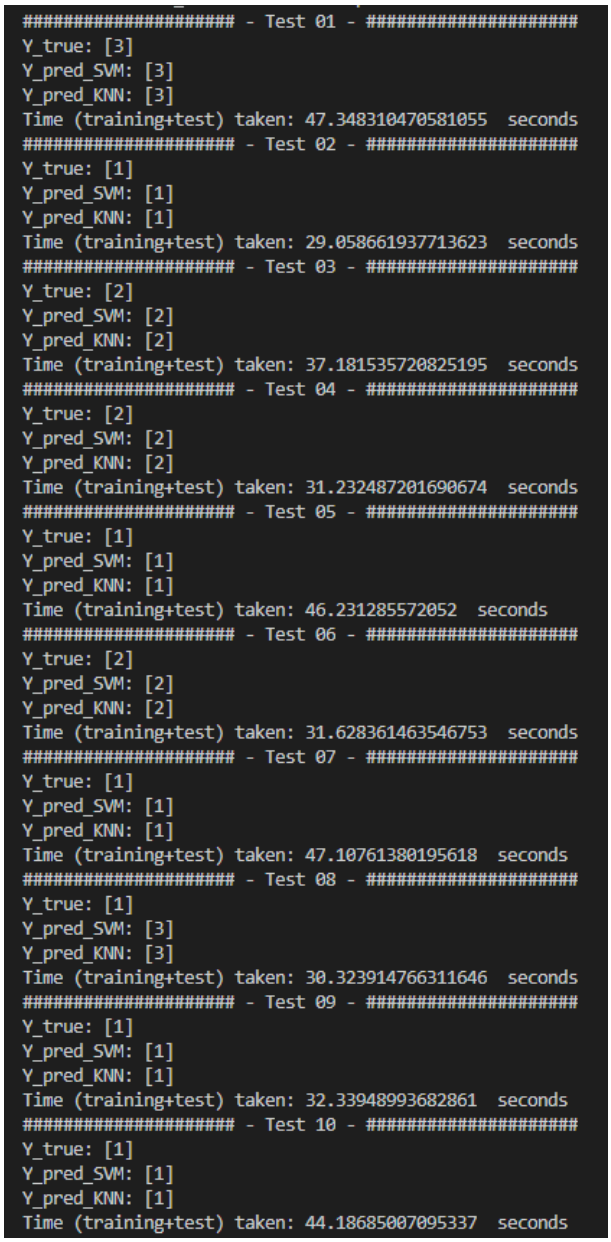
We tried the KNN but we see no use of KNN because we rarely have 3 examples or more for an author than eliminates KNN with K=3 so we should stick with KNN with K=1 but we were not convinced so we sticked to the recommendations of the papers using SVM.

Soft margin=2

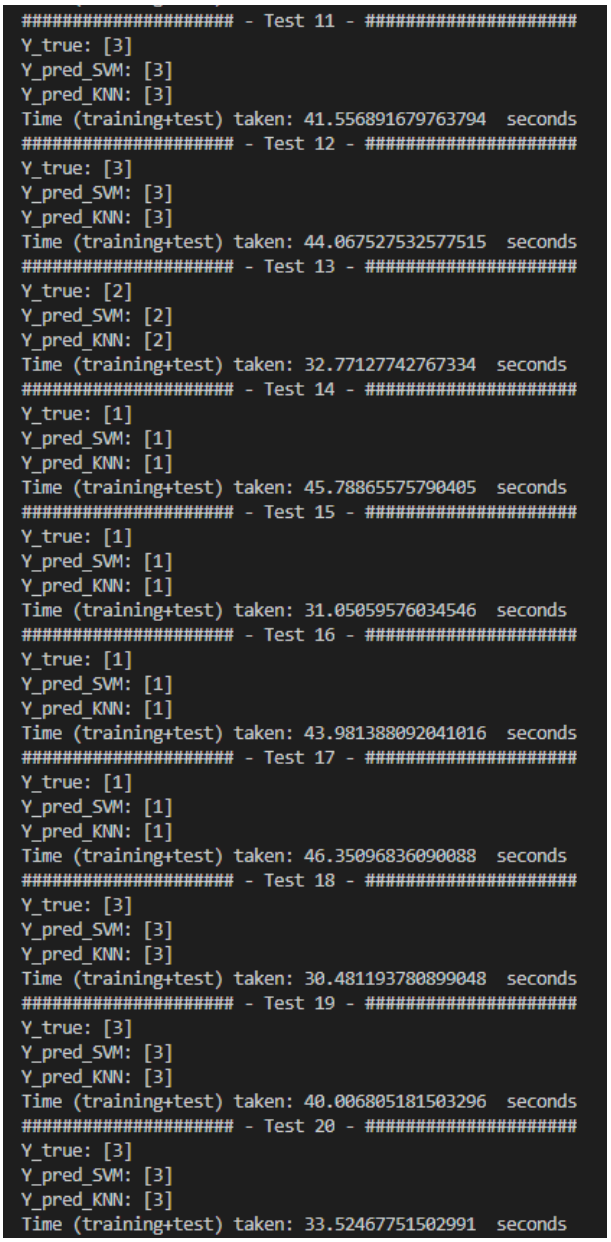
Max iterations by default: 1000

kernel:Linear

(e)Performance Analysis Module:



```
#################### - Test 01 - ####################
Y_true: [3]
Y_pred_SVM: [3]
Y_pred_KNN: [3]
Time (training+test) taken: 47.348310470581055  seconds
#################### - Test 02 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 29.058661937713623  seconds
#################### - Test 03 - ####################
Y_true: [2]
Y_pred_SVM: [2]
Y_pred_KNN: [2]
Time (training+test) taken: 37.181535720825195  seconds
#################### - Test 04 - ####################
Y_true: [2]
Y_pred_SVM: [2]
Y_pred_KNN: [2]
Time (training+test) taken: 31.232487201690674  seconds
#################### - Test 05 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 46.231285572052  seconds
#################### - Test 06 - ####################
Y_true: [2]
Y_pred_SVM: [2]
Y_pred_KNN: [2]
Time (training+test) taken: 31.628361463546753  seconds
#################### - Test 07 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 47.10761380195618  seconds
#################### - Test 08 - ####################
Y_true: [1]
Y_pred_SVM: [3]
Y_pred_KNN: [3]
Time (training+test) taken: 30.323914766311646  seconds
#################### - Test 09 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 32.33948993682861  seconds
#################### - Test 10 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 44.18685007095337  seconds
```



```
#################### - Test 11 - ####################
Y_true: [3]
Y_pred_SVM: [3]
Y_pred_KNN: [3]
Time (training+test) taken: 41.556891679763794  seconds
#################### - Test 12 - ####################
Y_true: [3]
Y_pred_SVM: [3]
Y_pred_KNN: [3]
Time (training+test) taken: 44.067527532577515  seconds
#################### - Test 13 - ####################
Y_true: [2]
Y_pred_SVM: [2]
Y_pred_KNN: [2]
Time (training+test) taken: 32.77127742767334  seconds
#################### - Test 14 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 45.78865575790405  seconds
#################### - Test 15 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 31.05059576034546  seconds
#################### - Test 16 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 43.981388092041016  seconds
#################### - Test 17 - ####################
Y_true: [1]
Y_pred_SVM: [1]
Y_pred_KNN: [1]
Time (training+test) taken: 46.35096836090088  seconds
#################### - Test 18 - ####################
Y_true: [3]
Y_pred_SVM: [3]
Y_pred_KNN: [3]
Time (training+test) taken: 30.481193780899048  seconds
#################### - Test 19 - ####################
Y_true: [3]
Y_pred_SVM: [3]
Y_pred_KNN: [3]
Time (training+test) taken: 40.006805181503296  seconds
#################### - Test 20 - ####################
Y_true: [3]
Y_pred_SVM: [3]
Y_pred_KNN: [3]
Time (training+test) taken: 33.52467751502991  seconds
```

Accuracy: 95+%

Average Execution time (training +test)= 35sec

(f)Enhancements and Future work:

- Better line segmentation for accurate performance in some typing fonts for some authors.
- More Optimization by using the already implemented LBP for faster execution.
- Calculation of slant angle for identification of smoke authors.
- Using texture blocks for LBP instead of line LBP.

References

- https://www.sciencedirect.com/science/article/abs/pii/S095741741201130X
- https://www.semanticscholar.org/paper/Efficient-Handwritten-Alphabet-Recognition-Using-Hirwani-Verma/07996293e7cf7bb84dc0ea7d081d33257eadb18c
- https://cvl.tuwien.ac.at/wp-content/uploads/2016/02/dissertation-fiel.pdf
- https://www.journals.elsevier.com/computers-and-electrical-engineering