



Faculty of Engineering

Credit Hour System

CMPN450



Cairo University

IAM Writer Identification

Computer and Communication Engineering, Cairo University

Abdelrahman Mohamed Emam 1170386

Monica Ehab 1170038

Roaa Magdy 4180298

Ziad Kamal El-Din Shebl 1170397

Submission date : 19/1/2021

Abstract. This paper targets a writer identification system. Firstly, the system identifies the handwritten text and segment it into individual lines. Next it extracts a set of features containing the LBP histogram. Finally, this set of features is used along with a feature vector extracted from a previously given labeled input and a classifying system to identify the writer. The proposed method has been tested on the IAM database. The data is defined as seven handwritten paragraphs, six of them are labeled (two for each writer), and the last paragraph is the one we desire to know its writer. On this database a recognition rate of about average 98% has been achieved with a minimum of 96% and a maximum of 100% was reported according to 22-fold cross-validation.

1. Introduction

The handwriting of a person is considered one of the unique characteristics of a person, similar to other biometrics such as finger prints, retina patterns and voice. It even went beyond that, as the psychological state of the writer at the time of the writing can be indicated using Graphology. As a result of the enormous characteristics that can be identified from just a piece of paper, analyzing patterns of handwriting is used in a variety of fields, such as forensics. In this paper we consider the problem of identifying a writer from three different writers using samples of handwritten text. The method proposed in this paper is completely text-independent, which means that meaning of the written paragraphs is not known. Moreover, it is an offline mode, which means that it depends only on images of the handwritten paragraphs.

2. Segmentation

2.1 Image Preprocessing

It is the first stage of the text segmentation algorithm used in the system. The goal of this stage is to enhance the image quality to provide the next stages a preprocessed image where it would be much easier to segment then. To reach this image the process consists of 3 main steps. The first step is converting the input color image to a grayscale image. Next, a gaussian blur filter is applied to filter the noise. Finally, inverse binarization is applied as it will facilitate things in the next stages.

After the enhancing steps, the image is passed through a cropping algorithm to crop the handwritten paragraph only. As shown in figure 2.1 below, the IAM samples have a constant layout, where there are two lines in the top half of the page, bounding the printed text, and a line at the end of the page in the bottom half. The area desired is usually lying between the second and the third line. To get this area, the algorithm starts by finding the contours of the binarized image and search for the three main lines by their predefined width. Next, it discards most of the spaces around the paragraph to avoid irrelevant data.

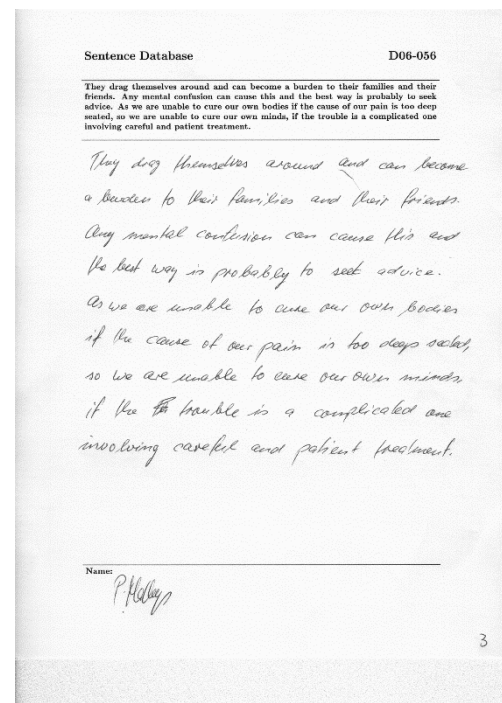
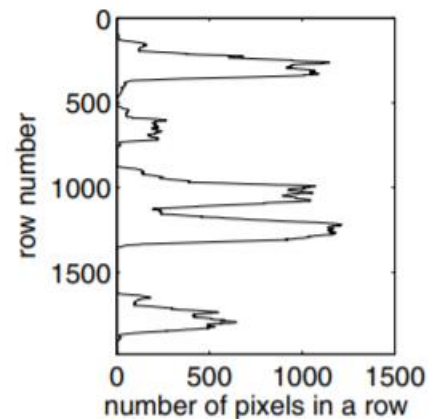


Fig 2.1

2.2 Calculating horizontal histogram

After enhancing the image quality, and discarding the unwanted areas, the target is to find the places of the lines to segment it. As a start, a horizontal projection profile is extracted to represent the density distribution of the handwriting. For example, long and dense lines will have higher peaks than short and less dense

Julian's mother performed
act of
motherly love-she put her
children's education ahead
of her own life.



ones.

Fig 2.2

2.3 Identifying the peaks

Finding the peaks of the histogram, means finding the places of the lines. One of the obstacles facing this method, is that there are fake peaks. To avoid the previous problem, a minimum threshold is defined, so any peaks found before this maximum threshold is discarded. If a peak is found after this threshold, it is compared to the neighbors until the continuous line drops under the threshold again; the maximum peak between rising above the threshold and dropping under it again is taken.

2.4 Identifying the valleys

To know how to identify a valley, a valley needs to be properly defined first. A valley is the point where a horizontal line can be drawn to separate two different lines with minimal details missing. It might seem easy, but the main obstacle here is the people who leave no space between the lines while writing as the image shown below. The algorithm used to detect the valley is much similar to the one used to detect peaks, the only difference is that it works on the minimum rather than maximum, so here a minimum threshold is used and the algorithm make sure to detect the minimum point that occurred in the range between the line going under this threshold then going up again.

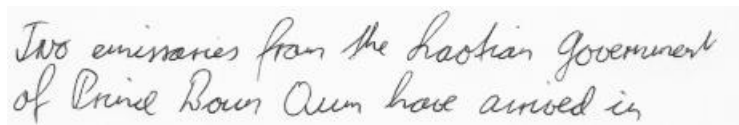


Fig 2.3

2.5 Identifying missing peaks

Unfortunately, some lines are short or not dense enough to project a peak that is high enough to overcome the threshold. To overcome this obstacle, distance between each two consecutive valleys is calculated and compared to the average distance between peaks, if the distance is almost the double, then there might be a missing line in that area, then the algorithm starts to find the peak in that area. Although the occurrence of short lines does not happen that much, adding this part to the algorithm resulted to increase in the accuracy up to 1%.

2.6 Detecting line boundaries and segmenting

This is the final stage of segmentation, the desired segments are always lying between every two consecutive valleys, so the algorithm hold the indices of every two consecutive valleys and try to minimize the distance between them if there is an empty area to avoid irrelevant data. Moreover, it does the same for the right and left sides of the segment to discard empty spaces at the beginning and the end. Next, a list of line boundaries consisting of the four boundaries of a segment (left, up, right, down) are provided so the segments could be extracted easily. Finally, each segment is considered as an independent data point.

3. Features for Writer Identification

Many methods were tried before settling on the proposed one as it provided the best results. Extracting the Local Binary Patterns from the whole paragraph was one method. Second method was to extract the slanting of the handwriting in eight different directions. Another method was to mix between extracting Local Binary Patterns from segments and the slanting feature vector. In contrast to all the previous methods, extracting the Local Binary Patterns from the segments and creating a 256 bins histogram was the method resulting the best accuracy.

3.1 LBP

Local Binary Patterns is a method used to describe the texture of a material. What differentiate it among other describers, is that it computes a local representation of the texture, without looking to other parts of the image. It calculates a number that represents the texture by comparing each pixel with its neighbors. Moreover, it has many hyperparameters such as how many neighbors should contribute, and the radius of the circle surrounding this pixel to search for neighbors at. In the method proposed, a radius of 1 and 8 neighbors are used.

3.2 Creating a window

According to research papers, it was found that resizing the width of the image, as to compress it, results in a better accuracy. Moreover, by trials it was figured that resizing the height too, results in reduction of the computation time and a bit better accuracy as it maintains the aspect ratio; as a result, for the previous statements, the image is resized to quarter its real size. Next, a sliding window of size 5 x 5 is passed over the resized image and check if it is not an empty window LBP histogram is calculated for it.

3.3 Calculating the LBP for a window

As the LBP used in this method has a radius of 1 and 8 neighbors a window of size 3 x 3 should be considered when calculating the LBP of a pixel. For each pixel in a 5 x 5 window, the LBP is calculated with taking its neighbors into consideration. To overcome that the corner and edges pixels do not have neighbors in specific directions, the 5 x 5 window is padded with zeros all around it, and the resulted window is of size 6 x 6. After calculating the LBP for the whole window, a new window of size 5 x 5 is provided containing the LBP values for the input window. Next, a histogram of 256 bins is calculated by counting the number of occurrences of that index.

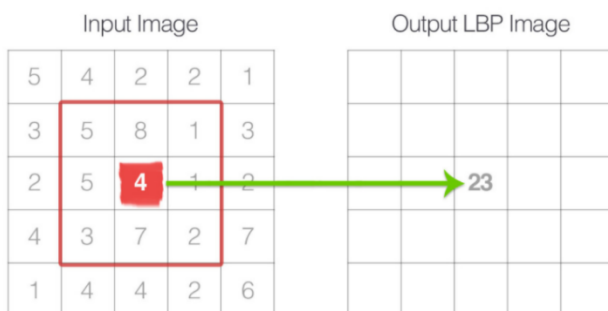


Fig 3.1

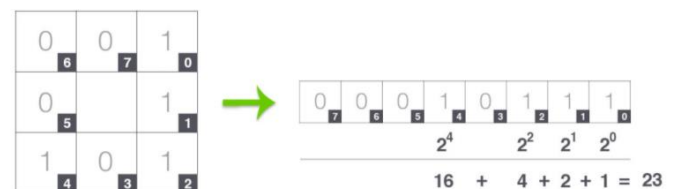


Fig 3.2

3.4 Creating the feature Vector

For each line segment one feature vector of size 256 is computed by accumulating all the LBP values calculated for all the windows in a line segment. This feature vector is describing the texture of the whole segment which is used in the following stages to train the models and test them.

4. Classification

In the method proposed, a voting system was used for classification. A variety of mix and match between different types of classifiers and different values for the hyper parameters were used. The classifiers used are in the following list:

- SVM
- Neural Networks (with two hidden layers)
- KNN (with K=3 and K=5)
- Random Forests
- Adaboost
- Gradient Boosting

After several trials, the combination that resulted for the best accuracy was SVM, KNN, and random forests as classification, in addition to a neural network in the post classification. The models are trained on all the segments extracted from the labelled input. Next, the SVM, KNN, and random forest take the segments of the test image as input and return a 2D array containing the probabilities of the belongingness of each segment to each writer. After that, the probabilities are added up so a score is assigned to each writer, and the writer with the highest score wins. If the ratio of the winners score compared to the total score is less than 0.4, a neural network is used as a post classifier to enhance the results; otherwise, the result is taken as it is.

5. Experimental Results

The experiments specified in this section are based on the IAM dataset, which contains around 1500 sample for over 650 writers. Each of the test cases specified in the following table consists of:

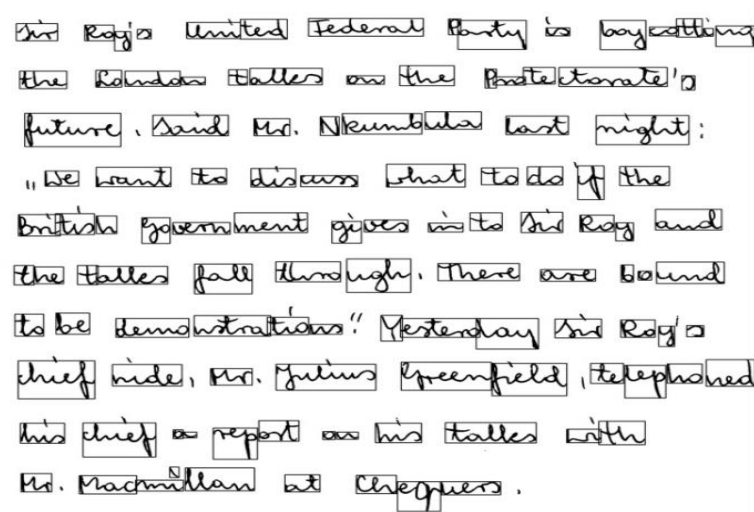
- 3 pairs of labelled paragraphs (A pair for each writer)
- 1 paragraph which is belonging to one of the 3 writers

The system takes the three pairs and train on them, then it takes the test paragraph and assign it to one of the three writers, if the paragraph was classified correctly a point is counted.

Batch Number	Number of Test Cases	Accuracy
1	100	100%
2	100	99%
3	100	99%
4	100	97%
5	100	97%
6	100	98%
7	100	99%
8	100	98%
9	100	100%
10	100	100%
11	100	98%
12	100	96%
13	100	99%
14	100	99%
15	100	99%
16	100	99%
17	100	97%
18	100	97%
19	100	99%
20	100	100%
21	100	100%
22	100	98%

6. Another Approach

Another approach was tried based on a research paper that uses connected component analysis. The approach was tried on the extracted gray paragraph from the samples of IAM dataset then inverse binarization was applied. After that connect components were extracted as separate images as shown in figure (6.1). Small components, such as periods, commas, strokes, and noise, are discarded at this time. These images were used to build a compressed version of the real paragraph .The objective behind this is to build a more representative texture of the paragraph minimizing blank space as shown in figure (6.2).



Sir Roy's United Federal Party is boycotting the London talks on the Protectorate's future. Said Mr. Nkumbula last night: "We want to discuss what to do if the British government gives in to Sir Roy and the talks fall through. There are bound to be demonstrations." Yesterday Sir Roy's chief aide, Mr. Julius Greenfield, telephoned his chief a report on his talks with Mr. Macmillan at Chequers.

Fig 6.1

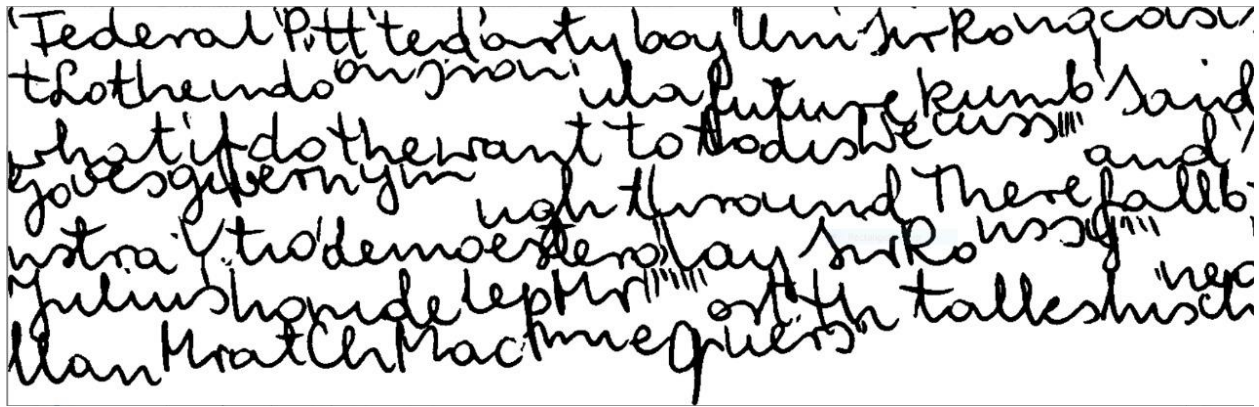


Fig 6.2

Then the LBP features were extracted from the compressed image and the same voting system of classification mentioned in 4 was applied in this approach. But it didn't achieve a good accuracy.

7. Conclusion

In this paper we proposed a previously used method but with some modifications as resizing the image to quarter of its width and height then applying the Local Binary Pattern descriptor on a window with size 5 x 5 sliding over the image. Also, we used a combination of classifiers with a voting system to get the best accuracy. Finally, after training the model on different random test case batches we obtained a min accuracy of 96%, maximum accuracy of 100%, and an average of 98.27% with an average time per test case approximately 5.4 second.

8. Future Work

For the second approach we believe it could result in a better accuracy if we build the compressed image by aligning the connected components using their centroids.

We can also use with the first approach another set of features which are the HOG (Histogram of Oriented Gradients) features.

9. References

- Sulaiman, K. Omar, M. F. Nasrudin and A. Arram, "Length Independent Writer Identification Based on the Fusion of Deep and Hand-Crafted Descriptors," in IEEE Access, vol. 7, pp. 91772-91784, 2019, doi: 10.1109/ACCESS.2019.2927286. Retrieved from: <https://ieeexplore.ieee.org/document/8756225>
- U. -. Marti, R. Messerli and H. Bunke, "Writer identification using text line based features," Proceedings of Sixth International Conference on Document Analysis and Recognition, Seattle, WA, USA, 2001, pp. 101-105, doi: 10.1109/ICDAR.2001.953763. Retrieved from: <https://ieeexplore.ieee.org/document/953763>
- Singh, P., Roy, P. P., & Raman, B. (2018). Writer identification using texture features: A comparative study. Computers & Electrical Engineering, 71, 1-12.]
- Bertolini, D., Oliveira, L. S., Justino, E., & Sabourin, R. (2013). Texture-based descriptors for writer identification and verification. Expert Systems with Applications, 40(6), 2069-2080.

Credits

Abdelrahman Mohamed Emam	(Worked on the first approach)
Ziad Kamal El-Din Shebl	(Worked on the first approach)
Monica Ehab	(Worked on the second approach)
Roaa Magdy	(Worked on the second approach)