

RESEARCH

Open Access



Writer verification based on a single handwriting word samples

Ameur Bensefia^{1*}  and Thierry Paquet²

Abstract

The writer recognition task has received a lot of interests during the last decade due to its wide range of applications. This task includes writer identification and/or writer verification. However, all the researches assumed that they dispose of a large amount of text to identify or authenticate the writer, which is never the case in real-life applications. In this paper, we present an original approach for the writer authentication task based on the analysis of a unique sample of a handwriting word. We used the Levenshtein edit distance based on Fisher-Wagner algorithm to estimate the cost of transforming one handwritten word into another. Such method has been successfully applied for signature authentication and voice recognition. In order to apply it to handwriting words, we developed a segmentation module to generate the graphemes; considered as elementary components for each word. We evaluated this approach on part of the IAM database (100 writers), where half of them provided three samples only of the same word. The obtained results are very promising since we succeeded to accept correctly in 87 % of cases when we used the whole database (100 writers) and up to 92 % when we used 40 writers.

Keywords: Verification, Authentication, Writer recognition, Edition distance, Levenshtein distance, Wagner-Fisher algorithm, Graphemes, IAM database, ROC curves

1 Introduction

Handwriting has always taken an important place in human lives. Even with the emergence of sophisticated devices (Ipad, smart phones, etc.), people still prefer writing. As a result, the amount of handwritten documents surrounding us is in continuous increase day after day. Researchers are trying to provide solutions to manage automatically this huge quantity of documents by developing specific tools based on the analysis of each specific need (see Fig. 1). The type of analysis depends also on the handwriting document acquisition mode (online or offline). In this paper, we focus in the offline mode only.

Figure 1 shows various handwriting analysis, where researchers have concentrated a lot of efforts for solving specific tasks among which are handwriting recognition [1, 2] and writer identification [3–5].

Forensic handwriting analysis has always been subject to controversies, since forensic experts have been using

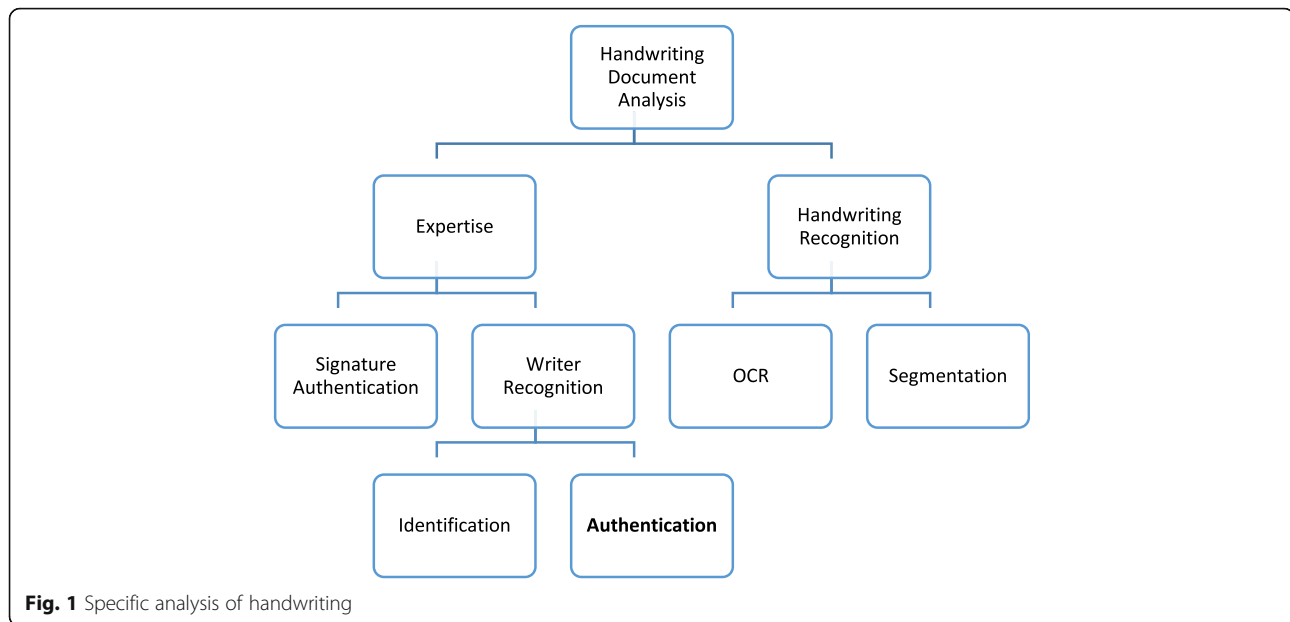
their subjective analysis by a visual inspection of the handwriting. Most of the time, their analysis was based on the inspection of specific character shapes or character ligatures, thus making the analysis text dependent, e.g., textual content is used during the analysis. The need of a scientific and an objective tool has emerged, and the scientists began proposing different approaches, to recognize the writer automatically and efficiently. Indeed, signature authentication is now mature enough to be embedded in industrial applications [6]. However, writer recognition began focusing interests recently due to emerging potential applications: in court of justice when there is a need to authenticate a will, in criminal investigations [7], or even when there is a need to reveal a genesis of some old manuscripts [8, 9].

Writer recognition is the process of revealing the identity of one individual based on the analysis of a sample of his handwriting (behavioral feature). In this sense, writer recognition can be presented as a biometric process that can be performed according to two different modes: identification and/or verification.

* Correspondence: abensefia@hct.ac.ae

¹Higher Colleges of Technology, Khalifa City Women's College, Abu Dhabi 41012, United Arab Emirates

Full list of author information is available at the end of the article



- In the identification mode, the system takes one unknown sample as input and must identify the author among a set of writers stored and known by the system.
- In the verification (authentication) mode, the system takes two unknown samples as inputs and has to decide whether these two samples have been written by the same hand or not.

In this paper, we are interested in the verification problem. Most studies have dealt with this issue by using large handwriting samples, composed of many lines of text (pages, one page or paragraph). Such approaches operate by being text independent, e.g., verification of the writer can be achieved whatever the textual content of the samples provided. In this paper, we propose a new approach for writer verification based on the analysis of a single handwriting word sample only. Such experimental setup is closely related to real-life application problems, where experts must take their decision by the analysis of a small sample of text of the writers: one word or even a part of a word to authenticate the writer rather than having a whole page or a paragraph. However, it is obvious that the performance of any writer recognition system is totally sample size dependent (small size sample means lack of information). This question has not been studied in the literature where the majority of studies have been designed considering large sample size. In this study, we have concentrated on the examination of very small samples of writing but considering similar text, thus designing a text-dependent approach for writer verification. In other words, the two samples taken as input by our verification system have to be textually similar.

This paper is organized around four sections: we begin by giving a brief state of the art about the main topics in the writer recognition task. In the second part, we present our system by introducing the notion of Levenstein edition distance and the Wagner-Fisher algorithm [10]. In the third part, we develop the experimentations by presenting the dataset used and the results. Finally, we draw some perspectives to this work in the conclusion.

2 Literature review

The number of studies related to writer recognition has significantly increased during the last decade. Some very detailed states of the art have been proposed in [3–5]. In the following section, we provide a framework that can be used to compare the multiple studies.

2.1 Identification vs. verification

As mentioned before, the writer recognition problem can be seen as a biometric problem. Two modes are available to operate such recognition: the identification and the verification. The identification system will always provide a ranked list of possible writers known in the system's database, even if the writer to be identified does not exist in this database. In this sense, the verification task provides a way of re-weighting this ranked list, allowing some potential candidates to be rejected during the second stage.

In the literature, we can distinguish two major schools: those who consider the writer recognition as an identification problem only [11–14] and those who consider it to be split into two complementary problems: identification and verification [15, 16]. The application domain has always driven the motivations of such choices. However,

writer verification has received much less interest compared to writer identification [17–20].

2.2 Samples: nature and size

In any handwriting analysis system, the handwriting samples play an essential role. They are used in both training and test. Consequently, the system will be based on their nature and on the quantity of information they may contain.

Samples from different natures have been used. Even if most of the studies used contemporary handwriting, some of them have used samples of ancient inscriptions and Byzantine codices [9]. In [16] the authors have studied the fourteenth century Dutch charters for writer identification. The use of handwritten musical scores [13] has also been a subject of interest.

In terms of sample size, different approaches have been considered. The authors in [12, 18, 21] used one to several pages; authors in [11] used paragraphs. However, only few studies [20, 22] have tried to follow an experimental setup where only small samples are available to identify or authenticate the writer, facing similar conditions than forensic examiners may be confronted to. This is mainly due to the fact that sample size affects mainly the system's performance.

2.3 Script

Even if the first writer recognition systems have been evaluated mainly with Latin languages, different systems have emerged later and focused to other scripts. The authors in [21] developed a Persian writer identification system, [15] and [11] proposed a system for Arabic writers, [23] for Chinese writers, [19] for Japanese writers, and [24] for Telugu writers.

Unlike the handwriting recognition systems, where each approach needs to be adapted and tailored to a given language to improve the segmentation and the features extraction process, the writer recognition systems can be independent, to a certain limit, of any script barrier. Indeed, most of the approaches did not take any advantage from the particularity or the textual content of the script, since they process the samples as simple images [12, 16, 20], thus showing that the systems might be adapted to different scripts provided some training data are available.

2.4 Features

We already stated in our previous works [8] that the features used in the writer recognition systems could be classified into structural and statistical features. In this section, we will focus on the features used in the verification systems only.

The authors in [17] built their verification system on the effect of the slant of handwriting. In order to extract

this feature, they used the probabilities of ink distribution on the contours, the Fraglets features (probabilities of distribution of graphemes with pre-computed codebook), and the probabilities distribution of angles combination of the ink at boundaries. These features have been evaluated using the IAM dataset using sentences and they provided very good results of 98 % on average.

The authors in [19] based their approach on the Japanese script. Their main feature was the pen pressure that they extracted by analyzing the ink distribution. They used two types of images, resulting from their multiband image scanner, which generates visible and infrared images (IR): two approaches have been used to extract the features. With the visible images, they use the LBP (8-b binary code) assigned to each pixel of their input grey level images. For the IR images, they extract the pen pressure using the first and the second order for some statistical measures such as the following: variance, mean, skewness, kurtosis, energy, and entropy.

Writer identification based on a single handwritten word has not received enough interest despite its wide range of application. Few authors only [20, 22, 25] ventured to use one word for both training and testing. Indeed, the authors in [22] tried to authenticate writers based on a single word by cutting out the word and transforming it into a six-dimensional time series and compared it by means of DTW methods. The authors tried to use the IAM database; however, they need a large number of occurrences from each word, which pushed them to create their own dataset (in German) made of 104 writers and where each writer provided five samples for each word. The performances reached 66.3 % when they use one word; however, when they combine 12 words (authentication based on sentence), the performances are higher and reached 99.6 %.

The authors in [20] proposed an approach, to authenticate the writer, based on horizontal and vertical projections for the different regions of the handwritten word. They evaluated their approach on a small dataset (English and Greek) made of 20 different writers who provided each 120 samples from the same sentence. The writer is authenticated by the decision fusion on five different words. Their results display a discrimination error smaller than 1 % for a five-word sentence.

In [25], the authors proposed a text-dependent approach to authenticate writers based on their handwriting words. They exploited the images statistical pixels directions by counting the occurrence of their transitions, along predefined paths, within two pre-confined chessboard distances. A two stage classification scheme based on similarity measure and an SVM was used. In order to evaluate their approach, the authors used the same dataset in [20], but they added 20 other writers to reach 40 writers. When the authors used one word to authenticate the writer, the equal error rate is 15.5 % for

the English words (22.8 % for the Greek words). This rate decreases significantly when five words are used simultaneously and their decisions merged to reach 4.08 % (5.71 % for the Greek words).

From the studies reviewed in this section, it seems clear that the writer authentication based on handwritten words has received less interests comparing to authentication based on sentences or block of texts, this is mainly due to the lack of writer individuality embedded when the samples are small. In this paper, we propose an approach that authenticates the writer based on a single and unique handwritten word. To overcome the problem of the lack of information, we decided to be text dependent, and this allows us to examine, as the forensic examiners, the elementary forms of both samples: the graphemes.

3 Proposed system

Our verification problem can be formulated as a comparison between two strings or two words. This comparison should be done by comparing their elementary components two by two, ideally comparing their respective characters two by two. This is especially true if we know that the forensic examiners use this approach in authenticating the handwriting. Indeed, they identify two elementary shapes in both samples and they compare them in depth [26].

3.1 Edit distance

Defining our problem in such terms is similar to the definition of the edit distance. Indeed, if we consider X and Y as two distinct strings defined as follows:

$$X = x_0x_1\dots x_n$$

$$Y = y_0y_1\dots y_m$$

The comparison of these two strings requires the comparison of their elementary components, which can be done according to different ways [27]. One of the famous distances is the Levenshtein distance [28]. This distance has already been used in different application: in gesture recognition [29], in the online handwriting recognition [30], in image sorting [31], and even in plagiarism detection [32]. It consists in summarizing the number of the elementary operations required to transform the string X into the string Y using three main operations:

1. Substitution $x \rightarrow y$ cost : $\gamma(x, y)$
2. Insertion $\lambda \rightarrow y$ cost : $\gamma(x, y)$
3. Deletion $x \rightarrow \lambda$ cost : $\gamma(x, y)$

The Wagner-Fisher algorithm [33] is one among those used to transform one string into another. It defines an edit distance between strings, by eliminating the unrealistic constraint of equal lengths. This algorithm is widespread and has already been used in several applications: handwriting recognition [27], speech recognition [34], and signature verification [35].

However, to be efficient in our case, the two samples need to represent the same word. In other word, we should be text dependent. Since, if we use two different words, the cost of transforming one element of the string X into another one from the string Y will always be high.

3.2 Graphemes

Considering character strings, the elementary elements are characters. To be able to apply the edit distance to the handwriting words, the elementary element should be the character too. However, extracting characters from a handwritten word is a recurrent and unsolvable problem due to the segmentation complexity [33, 36]. To overcome this problem, we decided to use a grapheme decomposition of handwriting.

A grapheme is an elementary, graphical shape, resulting from a process of handwriting segmentation. This elementary shape can be a character, if the segmentation is perfectly done. However, in most cases, it can be just a part of a character (over-segmentation) or a combination of many characters (under-segmentation) (see Fig. 2).

In our approach, we are dealing with writer authentication, a perfect segmentation is not a barrier in our case, since we are looking to preserve the writer individuality in each grapheme [37], whatever this grapheme may be, a whole character (perfect situation), piece of a character or a combination of several characters.

We implemented a segmentation module that began with the images binarization, a skeletonization, and an upper contour image extraction. Our segmentation module is based on the following steps, performed in the binarized upper contour of the image:

- The baseline detection: we assume that the junction between characters is usually located in

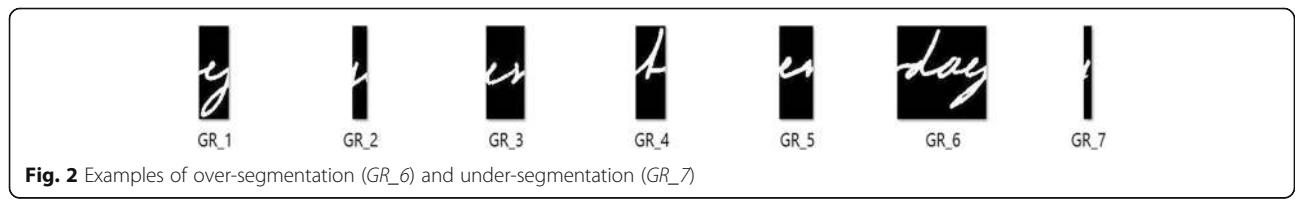


Fig. 2 Examples of over-segmentation (GR_6) and under-segmentation (GR_7)

the baseline, which is a fictive line used by writers to align their handwritings. We identify this area by performing a vertical histogram analysis; we choose the two highest values in the histogram to represent this area.

- Isolate the connected components: for each handwriting word (image), we identify the connected components, which represent masses of black pixels connected between each other.
- Segment only the biggest connected component: since we are processing words, we assume that only the biggest connected component, in terms of number of black pixels, should be segmented. Indeed, the rest of the components represent most of the time single characters.
- Candidate points: for the selected connected component Z of size (n, p) , we consider all the black horizontal points, at the baseline, as potential segmentation points. We reduce their number by eliminating the following:
 - Points which cross holes and curves: for every candidate point (x_i, y_j) , we check all the pixels (x_i, y_k) , where $k = 0..j - 1$ and $k = j + 1..n$. If we found at least two consecutive black pixels (x_i, y_k) and $(x_i, y_{\pm k})$ both belonging to the same connected component Z , we consider that we are crossing a hole or a curve and we eliminate the point (x_i, y_j) from the list of potential segmentation points.
 - Vertical lines: for each candidate point (x_i, y_j) , we check all the pixels (x_i, y_k) , where $k < j$ till the top

of the image. If we found a group of consecutive black pixels, we consider that we are crossing a line and we eliminate the point (x_i, y_j) from the list of potential segmentation points.

- Segmentation points: for the remaining points, and in order to reduce their number, we use a threshold to merge several neighbor points in a single and unique segmentation point. Some results are presented in Fig. 3.

The rules embedded in our segmentation module are specific to the Latin languages (English, French). It cannot be used to segment Arabic or Chinese handwritings due to their specificities, unless a deep modification.

3.3 Edit distance between handwriting words

The writer verification based on handwriting words can be formulated as a process of calculation of edit distances between two simples' handwriting strings. Let X and Y be the two handwriting words:

$$X = x_0x_1...x_n$$

$$Y = y_0y_1...y_m$$

where x_i and y_j represent, respectively, their graphemes.

We also define the substitution function, which associate a cost of transforming one grapheme x_i into another grapheme y_j by a similarity measure (correlation). The

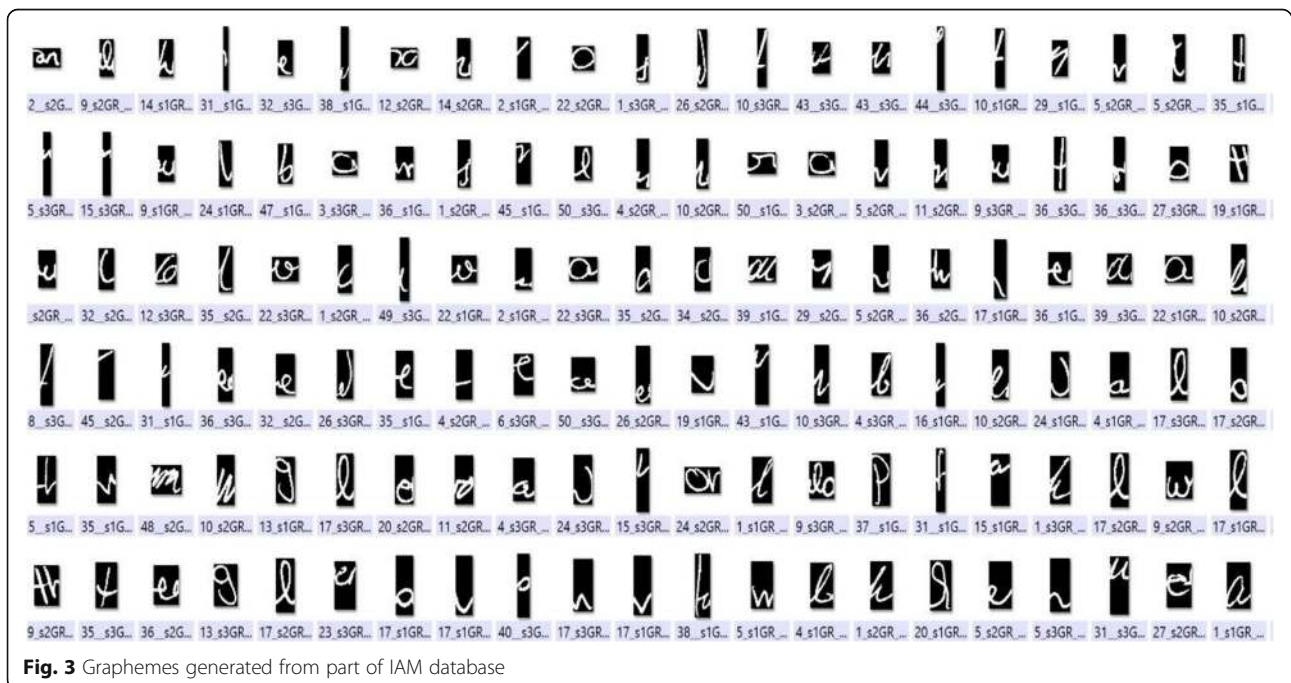


Fig. 3 Graphemes generated from part of IAM database

algorithm for computing the edit distance between handwriting words is defined as follows:

```

D(0,0) := sim(x0, y0)
for i := 1 to n Do
    D(i, 0) := D(i-1, 0) + sim(xi, y0)
for j := 1 to m Do
    D(0, j) := D(0, j-1) + sim(x0, yj)
for i := 1 to n Do
    for j := 1 to m Do
        M1 := D(i-1, j-1) + sim(xi, yj)
        M2 := D(i-1, j) + sim(xi, yj)
        M3 := D(i, j-1) + sim(xi, yj)
        D(i, j) = Max(M1, M2, M3)
Distance(X,Y) := D(n, m) / (n+m)

```

where $\text{sim}(x_i, y_j)$ is the correlation measure between the two normalized graphemes x_i and y_j . The graphemes have been normalized to 50×50 pixels.

Indeed, high values of the transformation cost of X into Y reflects a low variability intra-words, which implies that the two words may have been written by the same hand, whereas low values of cost transformation reflects a great variability which implies that the two words may have been written by different writers.

4 Experimentation and results

4.1 Dataset

In order to evaluate our approach and to make our performances comparable with other approaches, we decided to use one of the famous handwriting databases available: the IAM database [38]. This English handwriting dataset is made of 657 writers. It has been created initially for handwriting recognition purposes to evolve later on to allow writer recognition purposes. Each writer has been asked to write at least one form that has been scanned in gray level at a 300-dpi resolution. These forms have been segmented into 115,320 words stored in separate files.

As mentioned before, our system requires the two words in input to be textually the same. In this regard, we investigated the IAM database to look for the words written by the maximum number of writers and, at the same time, written several times by every single writer, but no word satisfied such criteria. Then, we made the choice to identify for each writer the longest word written at least three times.

Forty-four different words have been identified, written by 50 different writers generating 150 images (see Fig. 4a)

that have been used for the intra-writer distance computation. These 44 words have different sizes varying from five characters, for the shortest words (like, “should,” “added,” “piece”...), up to 13 characters for the longest words (like, “advertisement”).

Four other words have been written by 50 other writers generating 50 different images (see Fig. 4b) that have been used for the inter-writer distance calculations. The words used in this dataset were as follows: “Government,” “Yesterday,” “Tradition,” and “Younger”.

In summary, we used 100 writers from the IAM dataset, 50 writers were used to generate the intra-writer distances, and the rest was used to generate the inter-writer distances.

4.2 Discussion and results

We evaluated our system progressively, with 20 writers only at the beginning, then 40, 60, 80, and finally 100 writers (the whole database). No fusion decision is made, and the distance calculated is based on a unique handwritten word.

Since our approach can be formulated as a two class decision, we generated a ROC (receiver operating characteristic) curve for each size of the dataset used (see Fig. 5). The ROC curve shows the false positive rate error (deciding that the two handwriting words belong to different writers, when it is wrong) against the true positive rate (deciding that the two handwriting words belong to the same writer, when it is true) and this by varying the threshold decision values. The perfect point in any ROC curve should be in the top left corner.

It seems clear that the performances are high when the probability of confusion is low; when the size of the database is small (20 writers). In this case, we display results of correct authentication in around 98 % of the cases; however, we rejected wrongly in 17 % of the cases. These results are lower compared to those obtained in [20] where the error rate was less than 1 %; however, they used a decision fusion of five words at the same time. Our rates decrease naturally when the size of the database increases to accept correctly in 92 % with a dataset of 40 and reject wrongly in 15 %. Our results seems better compared to those obtained in [25] with the same dataset size, where the authors display results of 74.5 % of good acceptance.

The most relevant results are those obtained with the whole dataset (100 writers). In this case, the best performance we achieved was 87 % of good acceptance, where the authors in [22] achieved 66.7 % with the same number of writers, and 27 % of false rejection.

Compared to other approaches, our method seems displaying a very good results, despite a high rate of false rejection (27 %) in the case of 100 writers. The first reason that can explain these results is for sure the lack of

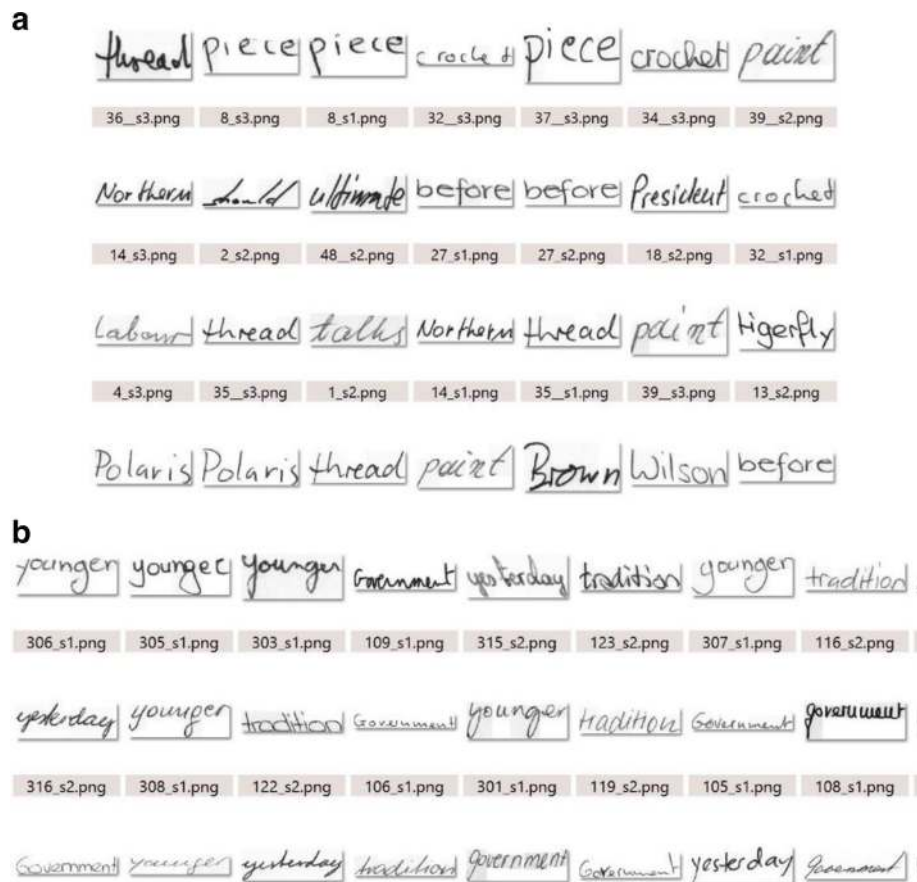


Fig. 4 Samples from the intra-database (a) and the inter-database (b) writers

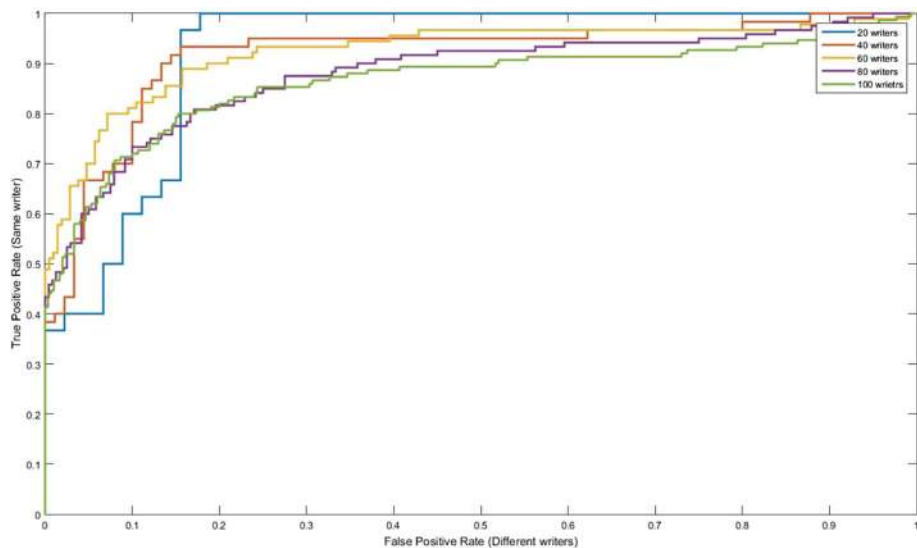


Fig. 5 ROC curves for writer verification with different database size

information: some writers have been represented with five-character words only which cannot embed all the variability and the specificity of the writer. In addition, we analyzed the cases where our system accept/reject wrongly, and it seems that even for our human eyes, the handwritten samples are too different (or too similar) to be considered belonging to the same writer (to different writers). Indeed, Fig. 6a shows a high handwriting variability for intra-writer (words written by the same writer, but different shape every time) where Fig. 6b shows a low handwriting variability for inter-writers. The consequence of such situation is a different segmentation and a generation of a different set of graphemes that cannot be matched.

Finally, if we compare these results with those obtained in [20, 22, 25], ours are very promising if we put in light the fact that the authentication is based really on a single word, no fusion or merge decisions are made with any other words. In addition, we evaluated our system on a 100 writer's dataset at the opposite of [20] who used 20 writers only. We used only three samples per writer where [20, 22, 25] used five samples and finally in our system no fusion decision was made.

5 Conclusions

In this paper, we dealt with the writer recognition problem. We focused in the verification task, where we proposed a

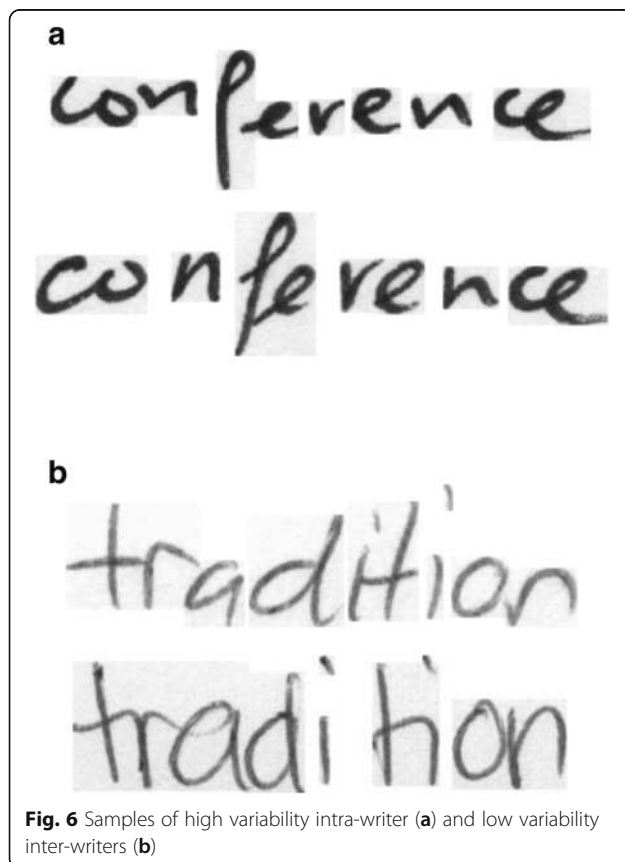


Fig. 6 Samples of high variability intra-writer (a) and low variability inter-writers (b)

new approach to authenticate writers based on a single word of their handwriting.

We considered the writer authentication problem similar to a signature authentication problem where the Levenshtein edit distance has been considered. Indeed, we used the Wagner-Fisher algorithm that allows estimating the cost of transforming one handwritten word into another one by evaluating the cost of transformation of their elementary components. As an elementary component of handwriting words (strings), we chose the grapheme. We generated the graphemes using a segmentation module that analyzes the upper images contours after binarization and a skeletonization process.

The approach has been evaluated on a part of the IAM database, where 100 writers were involved. Fifty writers were used to generate the intra-class distances and the rest to generate the inter-class distances. We used three samples per word for the intra-distances. The writers were represented through a single handwriting word made of 5 to 13 characters in length.

The obtained results are very promising regarding the small amount of information that we used. We succeeded to accept correctly in 87 % of cases when we used the whole database (100 writers) and up to 92 % when we used 40 writers. We analyzed the rejected cases of our system and the high variability for the intra-writer is one of the explanations. Indeed, when two samples written by the same hand are different, the segmentation module will behave differently producing different graphemes that cannot be matched.

As future works, we want to proceed to a selection process among the graphemes produced by choosing those who embed the most relevant variability information as the forensic examiners may do.

Authors' contributions

AB and TP designed the core methodology of the approach. AB carried out the implementation and the experiments. TP participated in helping to draft the manuscript. All authors approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Author details

¹Higher Colleges of Technology, Khalifa City Women's College, Abu Dhabi 41012, United Arab Emirates. ²Normandi Universite, UniRouen, UniLehavre, INSARouen, LITIS, Saint-Etienne du Rouvray, France.

Received: 2 March 2016 Accepted: 30 September 2016

Published online: 18 October 2016

References

1. A Graves, M Liwicki, S Fernandez, R Bertolami, H Bunke, J Schmidhuber, A novel connectionist system for unconstrained handwriting recognition. *Pattern Anal. Mach. Intell.* **31**(5), 855–868 (2009)
2. R Plamondon, SN Srihari, Online and off-line handwriting recognition: a comprehensive survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **22**(1), 63–84 (2000)
3. R Plamondon, G Lorette, Automatic signature verification and writer identification—the state of the art. *Pattern Recogn.* **22**, 107–131 (1989)

4. LRB Schomaker, Advances in writer identification and verification, in *Ninth International Conference on Document Analysis and Recognition*, vol. 2, IEEE, (Curitiba, Parana, Brazil, 2007), pp. 1268–1273
5. M Sreeraj, SM Idicula, A survey on writer identification schemes. *Int. J. Comp. Appl.* **26**(2), 23–33 (2011)
6. D Impedovo, G Pirlo, Automatic signature verification: the state of the art. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **38**(5), 609–635 (2008). doi:10.1109/TSMCC.2008.923866
7. L Jane, *Forensic Document Examination: Fundamentals and Current Trends*, 1st edn. (Academic, Norwell, MA, USA, 2014)
8. A Bensefia, T Paquet, L Heutte, *Information Retrieval Based Writer Identification*. 3rd International Workshop on Pattern Recognition in Information Systems, PRIS'2003, 2003, pp. 56–63
9. C Papaodysseus, P Rousopoulos, F Giannopoulos, F Zannos, D Arabadjis, M Panagopoulos, E Kalfa, C Blackwell, S Tracy, Identifying the writer of ancient inscriptions and Byzantine codices. A novel approach. *Comput. Vis. Image Underst.* **121**, 57–73 (2014)
10. RA Wagner, MJ Fisher, The string to string correction problem. *JACM* **21**(1), 168–173 (1974)
11. S Al-Maadeed, Text-dependent writer identification for Arabic handwriting. *J. Electrical Comput. Eng.* **2012**, 8 (2012). Article ID 794106. doi:10.1155/2012/794106
12. D Bertolini, LS Oliveira, E Justino, R Sabourin, Texture-based descriptors for writer identification and verification. *Expert Syst. Appl.* **40**(6), 2069–2080 (2013)
13. A Gordo, A Fornés, E Valveny, Writer identification in handwritten musical scores with bags of notes. *Pattern Recogn.* **46**(5), 1337–1345 (2013)
14. S Al-maadeed, A Hassaine, A Bouridane, MA Tahir, Novel geometric features for off-line writer identification. *Pattern. Anal. Applic.* **19**(3), 699–708 (2014)
15. MN Abdi, MA Khemakhem, Model-based approach to offline text-independent Arabic writer identification and verification. *Pattern Recogn.* **48**(5), 1890–1903 (2015)
16. M Bulacu, LRB Schomaker, Text-independent writer identification and verification using textual and allographic features. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 701–717 (2007)
17. AA Brink, RMJ Niels, RA Van Batenburg, CE Van den Heuvel, LRB Schomaker, Towards robust writer verification by correcting unnatural slant. *Pattern Recogn. Lett.* **32**(3), 449–457 (2011)
18. AA Brink, J Smit, ML Bulacu, LRB Schomaker, Writer identification using directional ink-trace width measurements. *Pattern Recogn.* **45**(1), 162–171 (2012)
19. M Okawa, K Yoshida, Text and user generic model for writer verification using combined pen pressure information from ink intensity and indented writing on paper. *IEEE Trans. Hum. Mach. Syst.* **45**(3), 339–349 (2015)
20. EN Zois, V Anastassopoulos, Fusion of correlated decisions for writer verification. *Pattern Recogn.* **34**(1), 47–61 (2001)
21. B Helli, ME Moghaddam, A text-independent Persian writer identification based on feature relation graph (FRG). *Pattern Recogn.* **43**(6), 2199–2209 (2010)
22. M Gehrke, KH Steinke, R Dzido, *Writer recognition by characters words and sentences*. 43rd Annual International Carnahan Conference in Security Technology, 2009, pp. 281–288
23. Z He, X You, Y Yan, Tang, Writer identification of Chinese handwriting documents using hidden Markov tree model. *Pattern Recogn.* **41**(4), 1295–1307 (2008)
24. P Purkait, R Kumar, B Chanda, Writer identification for handwritten Telugu documents using directional morphological features. 12th International Conference on Frontiers in Handwriting Recognition (ICFHR), Kolkata, 2010, pp. 658–663.
25. K Tselios, EN Zois, A Nassiopoulos, S Karabetsos, *Automated Off-Line Writer Verification Using Short Sentences and Grid Features*. 1st International Workshop on Automated Forensic Handwriting Analysis (AFHA), 2011, pp. 21–25
26. KM Koppenhaver, *Forensic Document Examination: Principles and Practice*, (Springer Science and Business Media, 2007): Humana Press; 2007.
27. M Parizeau, N Ghazzali, J-F Hébert, Optimizing the cost matrix for approximate string matching using genetic algorithms. *Pattern Recogn.* **31**(4), 431–440 (1998)
28. V Levenshtein, Binary codes capable of correcting deletions, insertions and reversals. *Sov. Phys. Dokl.* **10**, 707 (1966)
29. C Nyirarugira, C Hyo-Rim, K Jun Young, M Hayes, K Taeyong, *Modified Levenshtein distance for real-time gesture recognition*. 6th International Congress in Image and Signal Processing (CISP), vol. 2, 2013, pp. 974–979
30. SD Chowdhury, U Bhattacharya, SK Parui, *Online handwriting recognition using Levenshtein distance metric*. 12th International Conference in Document Analysis and Recognition, 2013, pp. 79–83
31. D Michel, AA Argyros, MIA Lourakis, *Localizing unordered panoramic images using the Levenshtein distance*. IEEE 11th International Conference in Computer Vision, ICCV, 2007, pp. 1–7
32. S Zhan, A Byung-Ryul, E Ki-Yol, K Min-Koo, K Jin-Pyung, K Moon-Kyun, *Plagiarism Detection Using the Levenshtein Distance and Smith-Waterman Algorithm*. 3rd International Conference in Innovative Computing Information and Control, 2008. ICICIC'08, 2008, pp. 569–569
33. M Vyas, K Verma, *A comprehensive survey of handwritten character segmentation*. International Conference in Advanced Communication Control and Computing Technologies (IACCCT), 2014, pp. 1462–1465
34. D Vaufraydaz, M Gery, *Internet evolution and progress in full automatic French language modeling*. IEEE Workshop in Automatic Speech Recognition and Understanding, ASRU'01, 2001, pp. 363–366
35. L Bovino, S Impedovo, G Pirlo, L Sarcinella, *Multi-expert verification of hand-written signatures*. Proceedings. Seventh International Conference in Document Analysis and Recognition, 2003, pp. 932–936
36. MA Abuzaraida, AM Zeki, AM Zeki, *Segmentation Techniques for Online Arabic Handwriting Recognition: a Survey*. International Conference In Information and Communication Technology for the Muslim World (ICT4M), 2010, pp. 37–40
37. A Nosary, L Heutte, T Paquet, Y Lecourtier, Defining writer's invariants to adapt the recognition task. in *Proc. ICDAR 99*, (Bangalore (India), 1999), pp. 765–768
38. U Marti, H Bunke, The IAM-database: an English sentence database for off-line handwriting recognition. *Int. J. Doc. Anal. Recogn.* **5**, 39–46 (2002)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com