Research Article

Text-Dependent Writer Identification for Arabic Handwriting

Somaya Al-Maadeed

Department of Computer Science and Engineering, College of Engineering, Qatar University, P.O. Box 2713, Doha, Qatar

Correspondence should be addressed to Somaya Al-Maadeed, s_alali@qu.edu.qa

Received 1 September 2011; Revised 7 November 2011; Accepted 28 November 2011

Academic Editor: Fouad Khelifi

Copyright © 2012 Somaya Al-Maadeed. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a system for text-dependent writer identification based on Arabic handwriting. First, a database of words was assembled and used as a test base. Next, features vectors were extracted from writers' word images. Prior to the feature extraction process, normalization operations were applied to the word or text line under analysis. In this work, we studied the feature extraction and recognition operations of Arabic text on the identification rate of writers. Because there is no well-known database containing Arabic handwritten words for researchers to test, we have built a new database of offline Arabic handwriting text to be used by the writer identification research community. The database of Arabic handwritten words collected from 100 writers is intended to provide training and testing sets for Arabic writer identification research. We evaluated the performance of edge-based directional probability distributions as features, among other characteristics, in Arabic writer identification. Results suggest that longer Arabic words and phrases have higher impact on writer identification.

1. Introduction

Two fundamental concepts are considered to be critical to writer identification: no two people write exactly alike, and no one person writes exactly the same way twice. These two principles, although oversimplified and disputable, clearly highlight two factors that directly conflict when attempting to identify a person based on handwriting samples. Figure 1 shows how Arabic handwriting differs from writer to writer. Our goal in this work was to automate the process of writer identification using scanned images of handwriting and thus to provide a computerized analysis of individual handwriting.

The task of writer identification is equivalent to answering one question: "who wrote this sample?" A writer identification system performs a one-to-many search in a large database with samples of known authorship and returns a likely list of candidates that have handwritings most similar to the sample in question. Considered within the general context of biometrics, automatic writer identification and verification is presently a thriving research topic.

Writer identification is used in forensic and biometric applications, in which the writer of a document can be identified based on handwriting samples. The identification

of writers of handwritten documents has great importance in the criminal justice system and has been widely explored in forensic handwriting analysis. The relationships between characters and the shape and style of writing will vary for different people. Handwriting, however, is a personal skill with individual characteristics. Therefore, it can be challenging to determine the best method for correctly identifying a writer.

This paper presents the problem of automatic writer identification using scanned images of Arabic handwriting. The objective was to identify the writer of one, or several, lines of handwritten text.

This paper is structured as follows: the following section summarizes previous publications related to writer identification; our proposed writer identification system is described in Section 3; in Section 4, we present our method for collecting and using handwriting data as a test case for our writer identification system for handwritten text lines using a set of confidence measures; in Section 5, the preprocessing operations are introduced in detail; the edge-based features are introduced in Section 6; the underlying database and the results of our experiments are presented in Section 7; finally, conclusions based on this study are presented in Section 8.

FIGURE 1: Handwriting by eight different writers.

2. Related Work

The identification of persons based on biometric measurements is currently a very active area of research [1–3]. Many different biometric modalities, including facial images, fingerprints, retina patterns, voice, and signatures, have been investigated. Writer identification has a long history, perhaps dating to the origins of handwriting itself. For example, many textbooks and research papers have been published that describe the methodologies employed by forensic document examiners [4–7].

As in related fields in forensic science, classic forensic handwriting examination is primarily based upon the knowledge and experience of the forensic expert. Due to problems associated with nonobjective measurements and nonreproducible decisions, recent attempts have been made to support traditional methods with computerized semiautomated and interactive systems.

For a survey covering work in automatic writer identification and signature verification until the end of the 1980's see [8]. An extension including work until 1993 has been published in [9].

According to Ameur Bensefia et al. [10], handwritten documents generally exhibit two kinds of use, corresponding with two different types of requests.

- (i) Handwritten documents can be analyzed for their textual content. In this case, the query of a handwritten document database would require one to resort to a transcription phase of the handwritten texts prior to the indexing of their textual content using standard techniques dedicated to information retrieval. Unfortunately, the state of the art in handwriting recognition does not allow the application of such an approach. Handwriting recognition remains poorly controlled in omniwriter applications when calling upon large lexicons.
- (ii) Handwritten documents can also be considered for their graphical content. In this case, queries of handwritten document databases can be carried out using graphical requests. For example, one can seek to retrieve documents from the database that contain calligraphy corresponding to specific writers. Other

possible applications involve the detection of the various handwritings present in a document or the dating of documents compared to the chronology of the author's work.

One can consider these two applications from the perspective of either textual or graphical information retrieval problems. These two tasks have been extensively studied in the electronic document retrieval and image processing fields. The task of automatic handwriting analysis falls into the writer identification paradigm.

The identification of the writer based on a piece of handwriting is a challenging task for pattern recognition. Automatic handwriting analysis techniques allow us to consider specific applications, as described in the following papers.

In [11], a system for writer identification using textural features derived from the grey-level cooccurrence matrix and Gabor filters is described. For this method, whole pages of handwritten text are required.

Similarly, in [12, 13], a system for writer verification is described. This system takes two pages of handwritten text as input and determines whether they were produced by the same writer. The features used to characterize a page of text include writing slant and skew, character height, stroke width, and frequency of loops and blobs. Morphological features obtained by transforming the projection of the thinned writing have been computed in [14].

Franke et al. [15] developed a computer system, known as the FISH system, for retrieving a small set of documents from a larger set. However, this system does not use suitable graphical user interfaces or wizards.

A complete handwritten document management system known as CEDAR-FOX has recently been developed by Srihari et al. [16] for writer verification. As a document management system for forensic analysis, CEDAR-FOX provides users with three major functionalities: a document analysis system; a system for creating a digital library; and a database management system for document retrieval and writer identification. The software, however, is currently undergoing beta testing.

Ameur Bensefia et al. [10] have presented two complementary approaches to writer recognition. They have adapted and applied an information retrieval approach to handwritten documents that has traditionally been used on electronic documents. In addition, they have proposed a hypothesis test that allows for the verification of the compatibility between the handwritings of two different documents. The major drawback of the proposed methodology is that it requires a sufficient amount of handwritten material to become independent of the textual contents.

A number of different systems have been used in Europe and the United States for writer verification and identification, as described above. However, most of these systems have been implemented for handwriting using only Latin-derived alphabets, and these systems are increasingly becoming outdated. Because common standards for data are lacking, a significant improvement can be expected if

FIGURE 2: Training and testing phases in the text-dependent writer identification system.

state-of-the-art methods in pattern recognition in the new millennium are employed.

Most work in the field of writer recognition has concentrated on signature verification because signatures typically present more individuality. However, in many cases, only words or characters rather than signatures are available for analysis. Word-based analysis began with the work of Steinke [17]. Zois and Anastassapoulous [14] used features obtained from the morphological transformation of thinned word images to answer different writer/same writer questions using a single word. In Srihari et al. [18], eleven global macro (document level) and microfeatures (character level) are employed; however, the performance based on macrofeatures extracted from words was very low.

Features used for the writer identification task mainly include global features based on statistical measurements extracted from the entire block of text to be identified. These features can be broadly classified into two families.

- (i) *Features extracted from textures*: in this case, the document image is seen simply as an image, not as handwriting. For example, the application of Gabor filters and cooccurrence matrices was considered in [11].
- (ii) *Structural features*: in this case, the structural properties of the handwriting are described based on extracted features. One can determine, for example, the average height, width, slope, and legibility of characters [19].

It is worth noting that it is also possible to combine these two families of features [12]. The nature of the statistical features extracted from a block of text has allowed for interesting performances; however, the results are difficult to compare due to a lack of common references.

One can also categorize previous studies according to the number of writers and the nature of training samples used by the system [8]. On the one hand, the system is required to deal with as many writers as possible. On the other hand, training samples of each handwriting may include several lines of text or only a few words. The work suggested in

[11], for example, makes it possible to identify 95% of the 40 writers that the system can handle through the analysis of text lines of handwriting. The work presented in [14] reports a correct writer identification performance of 92.48% among 50 writers using 45 samples of the same word that the participants were asked to write. It should be noted that the work presented in [12] had 1,000 writers and used the same text written three times by each writer.

3. System Overview

Traditionally, there are four steps in any writer identification system: (i) a step in which samples of scanned handwriting are entered into the system; (ii) a preprocessing step, in which information is set up that will be used to correctly perform the writer identification; (iii) a feature extraction step, which is used to obtain a relevant representation for the last step; (iv) a classification process step, which is the final step of the system. In the classification step, we used a k-nearest neighbor classifier. The other steps are described in Sections 4, 5, and 6. Our algorithm is based on text-dependent writer identification and consists of a training phase and a testing phase; a conceptual illustration is presented in Figure 2. The training phase of the system consists of preprocessing, feature extraction, and storing of the feature vectors and class labels (writers) of the training samples. In the actual classification phase, the test sample (where the writer is not known) is represented as a vector in the feature space after the preprocessing and feature extraction processes. Distances from the new vector to all stored vectors are computed, and the k closest samples are selected.

4. Data Set

The evaluation of our project requires a suitable data set of handwriting images. To obtain a suitable data set, a form was distributed and completed by 100 people using the same pen. The form consisted of four common Arabic phrases and twelve common words that were to be copied twenty times by the writers into ruled empty squares. Table 1 contains some

TABLE 1: Arabic words and phrases most likely to appear in handwritten letters.

1	بسم الله الرحمن الرحيم
2	السلام عليكم ورحمة الله وبركاته
3	تحية طيبة وبعد
4	ولكم جزيل الشكر
5	بخير
6	السيد
7	لك
8	في
9	هو
10	هي
11	المحترم
12	التوقيع
13	إلى
14	على
15	من
16	عن

of the most popular Arabic words and phrases used in Arabic letters. The most common phrases were counted manually from one thousand handwritten Arabic letters, while the most common words were taken from previous study [20]. The process of creating the data set proceeded as follows.

- (i) Filling of the forms by 100 people using the same pen.
- (ii) Scanning the forms using the same scanner (300 dpi, millions of colors).
- (iii) Cropping and naming each word image using a program that we created for cropping.

After completing these steps, we collected a total of 32,000 JPEG text images related to 100 different writers. The volunteers who completed the forms came from various age groups and levels of education.

5. Preprocessing

The preprocessing step in this paper begins with page scanning, as detailed in Section 5.1. Next, we describe the document segmentation step, which segments a page into words and stores each word in a separate file. Section 5.3 describes the background removing process. Finally, the edge detection process is described in Section 5.4.

5.1. Page Scanning. The completed forms were scanned at a resolution of 200 by 200 pixels. The forms were stored using the writers' identification numbers. Because we had a total of 100 writers, each writer was assigned a number from 001–100. The forms were then segmented into words, as described in Section 6.

5.2. Document Segmentation. In the preprocessing stage and after scanning the forms, we needed to crop the text images from the pages of the forms. To achieve this, a MATLAB program was written. Because we knew the dimensions of the edge pixels of the forms and the edge pixels of the words, the program uses this information to segment pages into words. The program segments the page into words and stores each word in a separate file.

5.3. Removing Background. The colored squared borders of the words in each form were removed using a thresholding operation. We used Otsu's method, which maximizes the likelihood that the threshold will be chosen, to split the image between an object and its background. This is achieved by selecting a threshold that gives the best separation of classes for all pixels within an image as shown in Table 2.

5.4. Edge Detection. The Sobel edge detection technique was used in this work. This method gives more accurate results for Arabic handwriting.

6. Feature Extraction

The edge-direction distribution [19], moment invariants, and word measurements feature extraction methods were used in this study. Section 6.1 describes edge-direction distribution with different angles, Section 6.2 describes the moment invariants method as feature extraction, and Section 6.3 describes the word measurements, such as area, length, height, length from baseline to upper edge, and length from baseline to the lower edge.

6.1. Edge-Direction Distribution. We used edge-direction distributions for four, eight, twelve, and sixteen angles. To find the edge direction for all of these angles, the technique uses the Sobel edge detection method. The program then labels the connected component of the image 8-pixel connected neighborhood. Next, the number of rows and columns of the binary image is found using the function size. Our system then searches the image to find pixels (black pixels) and the direction of each of these pixels. In this paper, we have calculated the edge-distribution features for four, eight, twelve, and sixteen angles. The following subsections describe the method of finding the direction of each of the pixels in all of these angles.

6.2. 4-Angle Edge-Direction Distribution. After finding a black edge pixel, the program considers this pixel as the center of a 3×3 -square neighborhood. Then the black edge was checked using the logical AND operator, in all directions starting from the central pixel and ending at one of the edges of the 3×3 square. However, all of the pixels shall be in the same connected component. To avoid redundancy, our algorithm checks only the upper two quadrants in the neighborhood because without online information, we do not know which way the writer traveled along the found oriented-edge fragment, which will give us only 4 possible angles (see Figure 3). Next, all verified angles of each pixel are

TABLE 2: Otsu's thresholding for an image with pink border.

Image before applying thresholding	Image after applying thresholding

FIGURE 3: Extraction of the edge-direction distribution using four angles.

FIGURE 4: Extraction of the edge-direction distribution using 8 angles.

counted into a four-bin histogram that is then normalized to a probability distribution that gives the probability of finding an edge fragment oriented in the image at the angle measured from the horizontal. In Figure 3, the pixels from center to pixel 1 are considered as long edge fragments.

6.3. 8-Angle Edge-Direction Distribution. As in the 4-angle edge-directional distribution, our program considers a pixel in the middle of a 5×5 -square neighborhood and checks in all directions starting from the central pixel and ending on one of the edges of the 5×5 square (Figure 4 shows the upper part of the squared window); all of the pixels shall be in the same connected component. Next, all of the verified angles of each pixel are counted into an eight-bin histogram that is then normalized to a probability distribution that gives the probability of finding an edge fragment oriented at the angle measured from the horizontal in the image.

FIGURE 5: Extraction of the edge-direction distribution using 12 angles.

FIGURE 6: Extraction of the edge-direction distribution using 16 angles.

6.4. 12-Angle Edge-Direction Distribution. Figure 5 shows the upper part of 7×7 squared window, to be used to extract the “12 edge-directional distribution features”.

All of the verified angles of each pixel are counted into a twelve-bin histogram that is then normalized to a probability distribution that gives the probability of finding an edge fragment oriented at the angle measured from the horizontal in the image.

6.5. 16-Angle Edge-Direction Distribution. The algorithm considers each edge pixel in the middle of a 9×9 -square neighborhood, as shown in Figure 6. The algorithm then checks in all directions using the logical AND operator starting from the central pixel and ending on one of the edges of the 9×9 square; all the pixels shall be in the same connected component. To avoid redundancy, this algorithm checks only the upper two quadrants in the neighborhood because without online information, we do not know which way the writer traveled along the found oriented edge fragment, which will give only 16 possible angles (see Figure 6).

Next, all the verified angles of each pixel are counted into a 16-bin histogram that is then normalized to a probability distribution that gives the probability of finding an edge fragment oriented at the angle measured from the horizontal in the image.

6.6. Moment Invariants. For text-dependent writer identification, we extracted the moment features. Pattern recognition using moment invariants uses a set of seven equations [21] as follows:

$$\begin{aligned}
 \Phi 1 &= \eta_{20} + \eta_{02}, \\
 \Phi 2 &= (\eta_{20} - \eta_{02})^2 + (2\eta_{11})^2, \\
 \Phi 3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2, \\
 \Phi 4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2, \\
 \Phi 5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \\
 &\quad \times [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
 &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
 &\quad \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2], \\
 \Phi 6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \\
 &\quad + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})], \\
 \Phi 7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \\
 &\quad \times [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
 &\quad - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03}) \\
 &\quad \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2].
 \end{aligned} \tag{1}$$

The seven moments were added in one feature vector, and we used standardization on this vector.

6.7. Word Measurements. We used word measurements, such as area, length, height, length from the baseline to the upper edge, and the length from the baseline to the lower edge, as features.

After computing each of these features, as discussed below, we used them together to create one feature vector and used standardization on this vector. We used standardization because features can have different scales, although they may refer to comparable objects. The following equation has been used for this purpose:

$$x'_i = \frac{(x_i - \mu_i)}{\sigma_i}, \tag{2}$$

where μ_i and σ_i are the mean and standard deviation, respectively, of feature x_i in the training examples [21]. The following sections describe in detail how we computed the area, length, height, length from the baseline to the upper edge, and the length from the baseline to the lower edge.

FIGURE 7: Measuring the length of the phrase.

FIGURE 8: Measuring the height feature.

6.8. Area. The area of the image was found by searching for any black pixels in the image. If the program finds a black pixel, it adds 1 to the total area value.

6.9. Length. In this paper, the length of a text segment or word is found by sequentially searching each column in the binary image to find the first and last pixels in the image and store their column numbers. Next, to find the length of the image, the algorithm subtracts the column number of the first pixel from the column number of the last pixel, as shown in Figure 7. The result gives us the length of the text.

6.10. Height. The method for determining the height of the image is similar to the method for finding the length of the image. However, when searching the binary image, the algorithm scans each row of the image to find the first and last pixels in the image and store their row numbers. The height of the image is found by subtracting the row number of the first pixel from the row number of the last pixel (Figure 8).

6.11. Height from the Baseline to the Upper Edge. To find the height of the text from the baseline to the upper edge (see Figure 9), the algorithm first determines the baseline position in the image. This is performed by forming an array where the index is the row number in the image. Then, the technique calculates the number of black pixels in each row of the binary image and stores the result in the array. After completing the entire image, the program finds the maximum value of the array and stores the row number of this maximum value as the baseline. Then, the program searches for the first pixel in the image and stores its row number. Finally, the row number is subtracted from the baseline row number to yield the length of the image from the baseline to the upper edge.

6.12. Baseline to the Lower Edge. To compute the length of the binary image from the baseline to the lower edge, we first determine the baseline row number, as described in Section 6.11. Then, we compute the last pixel row number of the image. The index row number is then subtracted from the

TABLE 3: Identification rates for different sets of features: F1, F2, F3, and F4.

Feature set	Min/max	Word/phrase	Identification rate
F1	Min	Ho “هو”	57.2%
	Max	Bism “بسم الله الرحمن الرحيم”	90.8%
F2	Min	Ho “هو”	53.8%
	Max	Bism “بسم الله الرحمن الرحيم”	91.6%
F3	Min	An “أن”	53.4%
	Max	Bism “بسم الله الرحمن الرحيم”	93.8%
F4	Min	An “أن”	53.4%
	Max	Salam “عليكم السلام ورحمة الله وبركاته”	92.8%

FIGURE 9: Measuring the length from the baseline to the upper edge.

FIGURE 10: Measuring the length from the baseline to the lower edge.

last pixel row number; this results in the length of the image from the baseline to the lower edge as shown in Figure 10.

7. Experiments and Results

The implementation of this system is based on a dataset containing 32,000 Arabic text images corresponding to 16 different words repeated 20 times each and written by 100 people using the same pen. The K-nearest neighbor classifier was trained using 75% of the words, and the remaining 25% were used for testing. The performance measures used were the top 10 identification rates. The performance histogram of the edge-based directional features is presented in Figure 11. The histogram shows the results of applying the features mentioned in the feature extraction section. F1 represents the combination of edge-directional features using four angles, invariant moments, and word measurement functions; F2 represents the combination of edge-directional features using eight angles, invariant moments, and word measurement functions; F3 represents the combination of edge-directional features using twelve angles, invariant moments, and word measurement functions; F4 represents the combination of edge-directional features using sixteen angles, invariant moments, and word measurement functions.

Table 3 shows the minimum and maximum percentages obtained from applying features F1, F2, F3, and F4. We

FIGURE 11: Performance histogram of the edge-based directional features for different Arabic words.

observed that the combination of features in F3 led to a high recognition rate in most of the tested words, and longer words had a higher recognition rate than shorter ones.

8. Conclusions

In this paper, novel features for writers identification were contributed. The performance of the new edge-based directional probability distributions and other features in Arabic writer identification was evaluated. The recognition rate for top ten writers is greater than 90% for certain words.

Many previous studies have focused on handwriting identification for English writers, including both hand-printed and cursive script [11–16]. The study of Arabic handwriting identification, however, has been much more limited. The recognition of Arabic characters is also important for certain non-Arabic-speaking languages, such as Farsi, Kurd, Persian, and Urdu. These groups use Arabic characters in writing, although they have different pronunciations.

Comparison of the final results obtained in this study with other research is difficult because of differences in experimental details, the actual handwriting used, the method of data collection, and the use of real Arabic offline handwritten words. Compared to other research [11–16] on

the identification of writers who wrote handwritten Arabic, the study presented here is the first to use edge-direction features on Arabic words. The selection of a suitable dataset is a critical step for successfully comparing results. To our knowledge, there is no existing large database with a good collection of Arabic handwriting documents specifically designed for writer identification research and application. Therefore, a clear standard data set to be used in future research is currently being generated at Qatar University. This data set will allow us to compare our results with other research results. In future work, we will test our system on text-independent writer identification with an improved set of features and classification methods.

Acknowledgments

This publication was made possible by a grant from Qatar University and the Qatar National Research Fund. The contents of this publication are solely the responsibility of the authors and do not necessarily represent the official views of Qatar University or the Qatar National Research Fund.

References

- [1] A. K. Jain, R. Bolle, and S. Pankanti, Eds., *Biometrics. Personal Identification in Networked Society*, Kluwer Academic publisher, 1999.
- [2] A. K. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Communications of the ACM*, vol. 43, no. 2, pp. 91–98, 2000.
- [3] J. Bigun and I. Smeraldi, Eds., *Audio- and Video-Based Biometric Person Authentication*, Proceedings of the 3rd International Conference, AVBPA, Halmstadt, Sweden, 2001.
- [4] R. N. Morris, *Forensic Handwriting Identification- Fundamental Concepts and Principles*, Academic Press, New York, NY, USA, 2000.
- [5] J. E. Dines, *Document Examiner Textbook*, Pantex International, USA, 1998.
- [6] G. R. McMenamin, *Forensic Linguistics- Advances in Forensic Stylistics*, CRC press, USA, 2002.
- [7] R. A. Huber and A. M. Headrick, *Handwriting Identification- Facts and Fundamentals*, CRC Press, USA, 1999.
- [8] R. Plamondon and G. Lorette, "Automatic signature verification and writer identification—the state of the art," *Pattern Recognition*, vol. 22, no. 2, pp. 107–131, 1989.
- [9] F. Leclerc and R. Plamondon, "Automatic signature verification: the state of the art 1989–1993," in *Progress in Automatic Signature Verification*, R. Plamondon, Ed., pp. 13–19, World Scientific, 1994.
- [10] A. Bensefia, T. Paquet, and L. Heutte, "Handwritten document analysis for automatic writer recognition," *Electronic Letters on Computer Vision and Image Analysis*, vol. 5, no. 2, pp. 72–86, 2005.
- [11] H. E. S. Said, T. N. Tan, and K. D. Baker, "Personal identification based on handwriting," *Pattern Recognition*, vol. 33, no. 1, pp. 149–160, 2000.
- [12] S.-H. Cha and S. Srihari, "Writer identification: statistical analysis and dichotomizer," in *SSPR and SPR 2000*, F. J. Ferrie et al., Ed., vol. 1876 of *Lecture Notes in Computer Science*, pp. 123–132, Springer, 2000.
- [13] S.-H. Cha and S. Srihari, "Multiple feature integration for writer verification," in *Proceedings of the 7th International Workshop Frontiers in Handwriting Recognition*, L. R. B. Schomaker and L. G. Vuurpijl, Eds., pp. 333–342, 2000.
- [14] E. N. Zois and V. Anastassopoulos, "Morphological waveform coding for writer identification," *Pattern Recognition*, vol. 33, no. 3, pp. 385–398, 2000.
- [15] K. Franke, L. Schomaker, L. Vuurpijl, and St. Giesler, "FISH-new: a common ground for computer based forensic writer identification," in *Proceedings of the 3rd European Academy of Forensic Science Triennial Meeting*, p. 84, Istanbul, Turkey, 2003.
- [16] S. N. Srihari, B. Zhang, C. Tomai, S.-J. Lee, Z. Shi, and Y. C. Shin, "A system for handwriting matching and recognition," in *Proceedings of the Symposium on Document Image Understanding Technology (SDIUT'03)*, Greenbelt, Md, USA, 2003.
- [17] K. Steinke, "Recognition of writers by handwriting images," *Pattern Recognition*, vol. 14, no. 1-6, pp. 357–364, 1981.
- [18] S. N. Srihari, S.-H. Cha, H. Arora, and S. Lee, "Individuality of handwriting," *Journal of Forensic Sciences*, vol. 47, no. 4, pp. 1–17, 2002.
- [19] U. V. Marti, R. Messerli, and H. Bunke, "Writer identification using text line based features," in *Proceedings of the Sixth International Conference on Document Analysis and Recognition (ICDAR'01)*, pp. 101–105, Seattle, Wash, USA, 2001.
- [20] S. Al-Ma'adeed, D. Elliman, and C. A. Higgins, "A data base for arabic handwritten text recognition research," in *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition (IWFHR'02)*, p. 485, 2002.
- [21] M. Nixon and A. Aguado, *Feature Extraction and Image Processing*, UK, 2002.