Pattern Recognition 43 (2010) 2199–2209

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A text-independent Persian writer identification based on feature relation graph (FRG)

Behzad Helli, Mohsen Ebrahimi Moghaddam *

Electrical and Computer Engineering Department, Shahid Beheshti University, G.C, Tehran, Iran

ARTICLE INFO

Article history:
Received 22 October 2008
Received in revised form
21 October 2009
Accepted 27 November 2009

Keywords:
Persian writer identification
Fuzzy method
Graph similarity

ABSTRACT

The style of people's handwriting is a biometric feature that is used in person authentication. In this paper, we have proposed a text independent method for Persian writer identification. In the proposed method, pattern based features are extracted from data using Gabor and XGabor filter. The extracted features are represented for each person by using a graph that is called FRG (feature relation graph). This graph is constructed using relations between extracted features by employing a fuzzy method. The fuzzy method determines the similarity between features extracted from different handwritten instances of each person. In the identification phase, a graph similarity approach is employed to determine the similarity of the FRG generated from the test data and the FRGs generated by training data. The experimental results were satisfactory and the proposed method got about 100% accuracy on a dataset with 100 writers when enough training data was used. However, this method has been applied on Persian handwritings but we believe it can be extended on other languages especially in data representation and classification parts.

1. Introduction

Writer identification and verification is a behavioral biometric system. Despite other biometrics such as finger print and DNA which have physical or biophysical basis; handwriting analysis requires an immense knowledge at multiple levels of observation [33].

Writer identification and verification systems are divided into two main classes: online and offline [2,3]. The on-line handwritten data contains more information about the writing style of a person such as speed, angle, or pressure that is not available in the off-line data. Thus, the on-line classification task is considered to be less difficult than the off-line ones [2,10]. In another classification manner, writer identification and verification systems are categorized by their dependency on the used texts in these methods. In this manner, the identification and verification methods are categorized into two groups; text-independent and text-dependent [1,2,8]. The text-independent methods are able to identify writers independent of the text content but the results of text-dependent methods are dependent to the text content.

However, there are many strong writer identification methods in the literature, but most of them are not in Persian language and they are not extendable for this language, because Persian language has some specific characteristic and several writing styles. The main characteristics of the Persian language is that it is a connected language and letters are pasted together to make words. Also, there are different writing styles in Persian that makes the feature extraction procedure different. Therefore, it is necessary to provide an accurate writer identification method for Persian language. In this paper, we have proposed an offline, text-independent, and precise method to identify Persian writers using their handwritten documents. The feature extraction, data representation, classification, and identification procedures of the proposed method are novels and have not been presented in state-of-art since now, that is, in the classification phase of the proposed method, instead of straight comparison of features, an FRG (feature relation graph) is created and identification is done by using a graph similarity algorithm. FRG is constructed using the fuzzy relations among the features. Also, a new filter that is called XGabor filter has been presented here to extract the features. This filter is based on Gabor filter that has been used in literature frequently. However, the feature extraction method has been designed for Persian language but we believe it can be used for other languages by changing the direction of filters that should be identified by experiments. In order to test the proposed method, we asked 100 people to write texts on five A5 pages. The authors were authorized to write arbitrary text, because the proposed method is a text independent one. The proposed method was tested on this data set in several phases. In each

* Corresponding author.
E-mail addresses: Be.helli@mail.sbu.ac.ir (B. Helli), m_moghadam@sbu.ac.ir (M.E. Moghaddam).

0031-3203/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2009.11.026

phase, different numbers of test and training data were used. The proposed method outperformed the related works and its experimental results were satisfactory. We could get 100% accuracy in best case by increasing number of the training data versus the test ones.

The rest of the paper is organized as follows: in the next section related works are surveyed. Section 3 describes the proposed method. In Section 3.1 feature extraction phase of proposed method is presented. Section 3.2 explains the fuzzy approach which was taken to generate FRG. Section 3.3 explains the methods which were used to calculate the similarity between FRGs. Setups and results of our experiments are presented and discussed in Section 4. Finally, Section 5 concludes the paper and proposes future work.

2. Related works

In this section, some of the recent writer identification systems are surveyed. Because there are a lot of researches in this field, the most important of them are presented here in two classes: online and offline systems. Also, offline systems are categorized based on the languages: Persian, Arabic, and English.

2.1. Online systems

Because the proposed method is an offline identification system, in this sub-section, we have only a brief look on two important online methods that maybe extended to offline ones.

Schlapbach et al. have presented an online writer identification method on whiteboard texts in Ref. [2]. They used a database that was gathered from 200 writers. In this method, different feature sets have been tested by them. These feature sets are classified as point-based (speed, writing direction, curvature), stroke-based (duration, time to next stroke, number of points, number of up strokes, etc.), extended point-based (speed, acceleration, vicinity linearity, vicinity slope, etc.), and offline point-based features. In the classification phase, Gaussian mixture model (GMM) is used. The models of the writers have been obtained from a universal background model (UBM) and its basic idea is to derive the writer's model by updating the well-trained parameters from the UBM. In the first step, all data from all writers are used to train a single writer independent UBM. In the second step, for each writer a writer dependent model is built by updating the parameters in the UBM via adaptation and using all the training data from this individual writer. The UBM has been trained using the expectation–maximization (EM) algorithm based on the maximum a posteriori (MAP) principle. The authors got 98.56% accuracy in paragraph level and 88.96% in the line level test among 200 writers out of IAM-DB.[1] Also, they extended their work for an offline English handwritten identification [4] and satisfactory results were obtained.

In another work, Chapran et al. proposed a method for dynamic writer identification method which used the relation between static and dynamic information in a handwritten text [43,44]. The correlation between length, direction, pressure, altitude, and azimuth of handwriting segments between two sample points are used to identify the writer.

There are some other works which have tried to create semi online information from offline data [16,30,37], but these methods are more useful in writer verification and forensic analysis systems.

[1] The IAM-OnDB: www.iam.unibe.ch/~fki/iamondb.

2.2. Offline systems

There are more presented offline works versus online ones. Therefore, in this sub-section the most important and recent ones are described. At first we have surveyed the methods that have been presented since now for Persian writer identification. Because of similarity between Persian and Arabic languages, Arabic methods have been reviewed in second sub-section and finally methods in English and other languages have been studied. However, there are some other methods in other languages; the most important and recent ones are mentioned here.

2.2.1. Persian

In Ref. [20], Shahabi et al. have presented a Gabor based system for Persian writer identification and the accuracy of their work was reported about 92% in top-3 and 88% in top-1. It seems they did not use proper way of testing; because in the test phase, there was only one page per person such that $\frac{3}{4}$ of it was used in training and the rest of page used in test phase. To verify these results in more general way, we have implemented and tested their method; when 5 pages for each writer were used in training phase and another separate page was used in test phase; the method accuracy was $<60\%$ in 80 people.

In Ref. [1], we have presented a Persian handwritten identification system that was based on a new generation of Gabor filter that was called XGabor filter. Feature extraction was done by using Gabor and XGabor filters; in the classification phase, weighted Euclidian distance (WED) classifier was used. In order to test the system, we organized a data set of 100 people's handwritings which it has been used in some other works also. This data set is called PD100 and it is referenced by this word in present paper. The proposed method in Ref. [1] got 77% accuracy using the PD100.

In another recent work, we proposed an LCS (longest common subsequence) based classifier to classify features that are extracted by Gabor and XGabor filters [45,46]. This classifier improved the system accuracy up to 95% on PD100. However, the features extracted by XGabor filter could model the characteristic of written documents but the accuracy of these methods was not proper because of problems in data classification and representation. Therefore, in the present paper, we used XGabor filter together with Gabor filter with different data representation, classification, and identification schemes.

In another research, a mixture of some different methods has been used by Sadeghi ram et al. In this method, grapheme based features are clustered by fuzzy clustering method and after selecting some clusters, final decision is made based on gradient features. The authors got about 90% accuracy in average on 50 people that were selected randomly from PD100 [47].The same authors also used a three layer MLP(multi layer perceptron) to classify the gradient based features, and they got about 94% average accuracy on same data set [48].

In another method, Soleymani Baghshah et al. have presented a fuzzy approach for Persian writer identification [19]. In this method, they have calculated some fuzzy directional features and the fuzzy learning vector quantization (FLVQ) have been trained in order to recognize the writers. The weakness of this method is that it only works on disjoint Persian characters that are not conventional in Persian language. This system was tested using 128 writers and results were around 90%–95% in different situations of test.

To the best of our knowledge, there is no any other reported method in Persian writer identification.

2.2.2. Arabic

In Ref. [7], Somaya Al-Ma'adeed et al. have presented a text-dependent writer identification method in Arabic using only 16 words. They have extracted some edge-based directional features such as height, area, length, and three edge-direction distributions with different sizes andWED has been used as classifier. They gathered 32 000 Arabic text images from 100 people, trained their system with 75% of the data and tested it by using other 25%. They did not mention the top-1 accuracy of the method, but the best result in top-10 was 90% when 3 words were used. The main concern of this method is its dependency to text and small dataset that were used in experiments.

Bulacu et al. presented text-independent Arabic writer identification by combining some textural and allographic features [8,12]. After extracting textural features (mostly relations between different angles in each written pixel) a probability distribution function was generated and the nearest neighborhood classifier using the χ^2 as a distance measure was used. For the allographic features, a codebook of 400 allographs was generated from the handwritings of 61 writers and the similarity of these allographs was used as another feature. The used database in experiments consisted of 350 writers with 5 samples per writer (each sample consisted of 2 lines (about 9 words)). The best accuracy of method in experiments was 88% in top-1 and 99% in top-10. Also, a simpler definition of this method was presented by M. Bulacu et al. earlier [29].

Also, Ayman Al-Dmour et al. have presented an Arabic writer identification system in Ref. [11]. They have tested several feature extraction methods such as hybrid spectral-statistical measures (SSMs), multiple-channel (Gabor) filters, and the grey-level co-occurrence matrix (GLCM). To determine which subset of these features is the best, they first used a support vector machine (SVM) to rank the features and then used a GA (which its fitness function was a linear discriminant classifier (LDC)) to get the best one. They also tried several classification methods such as LDC, SVM, weighted Euclidean distance (WED), and the K nearest neighbors (KNN). The KNN-5, WED, SVM, and LDC results after feature selection per sub-images were reported as 57.0%, 47.0%, 69.0% and 90.0%, respectively. The results were better when the whole image was used such that the LDC result was increased to 100% (with no rotation). The database they used was gathered from 20 writers; each writer was asked to copy 2 A4 documents, one for training and the other one for testing. The used documents for each writer were different from the others and the sub-images were generated by dividing each document into $3 \times 3 = 9$ non-overlapping images. However, this method has good accuracy when LDC was used, but it seems the test database and samples per writer was small and it needs to be tested on more popular dataset.

Although, Arabic language is similar to Persian in character set and some writing styles, the Arabic methods cannot be extended to Persian language completely because of some special symbols that exists in Arabic language.

2.2.3. English and other languages

Zhenyu He et al. have presented an offline Chinese writer identification method which used Gabor filter to extract features from the text. Also, they have used a Hidden Markov Tree (HMT) in wavelet domain. They tested their system by a database containing 1000 documents written by 500 writers. Each sample contained 64 Chinese characters. The top-1, top-15, and top-30 results had 40%, 82.4%, and 100% accuracy, respectively [3]. Also, these authors have used a combination of general Gaussian model (GGD) and wavelet transform on Chinese handwriting in Ref. [5]. They tested the method on a database gathered from 500 people. This database consisted of 2 handwriting images per person. In the experiments, top-1, top-15 and top-30 results had 39.2%, 84.8% and 100% accuracy, respectively. As, the authors reported the accuracy of proposed methods was low especially in top-1.

In Ref. [13], Vladimir Pervouchine et al. have presented a method based on high frequent characters. In their method, at first they find those characters ('f','d','y','th') then according to the similarity of those characters, the writer is selected. The similarity is calculated according to the several features (such as height, width, slant, etc.) that are variable in different characters (e.g. 'f' had 7 features while 'th' had 10 ones). A simple Manhattan distance was used in the classification phase by them. In order to select the best subset of the features, they also used a GA which evaluated about 5000 of the subsets out of 2^{31} possible subsets. The system was tested in a database with 165 writers (between 15 to 30 patterns per writer), and the system accuracy was more than 95%. However, this method is simple and has good results, but the main concern of this method is that if a writer knows the procedure of method, he/she can write a text in test phase that its characters are totally different with trained one and therefore the method cannot identify him/her.

Schomaker et al. has presented a method based on fragmented connected-component contours (FCO^3) [14,27]. They used the χ^2 method in the classification phase to calculate distance. Also, they tested their method in an English data set with 150 writers. The top-1 of the method results had 72% and the top-10 had 93% accuracy. However, the top-10 results are satisfactory but its top-1 is not.

Schlapbach et al. have presented a HMM based writer identification and verification method [17,23]. They designed and trained an individual HMM for each writer's handwriting. To determine which writer has written an unknown text, the text is given to all the HMMs. The one with biggest result is assumed to be the writer. The identification method was tested by using documents gathered from 650 writers. Their method accuracy was 97%. Also, this method was tested as a writer verification method. For that reason they gathered writings from 100 people and asked 20 unskilled imposters and 20 skilled ones to forge them and they got about 96% overall accuracy in verification. It seems this method can be extended to other languages by applying some changes on feature extraction phase.

The method proposed in Ref. [18] is same as presented method in Ref. [8]. The difference between these two methods is that former was used in English handwriting and got about 80% accuracy in top-1 results and about 92% in top-10 results while the Ref. [8] supported Arabic handwritten and its accuracy was 88% in top-1 and 99% in top-10 results.

In Ref. [21,22] Ameur Bensefia et al. have developed a probability based approach using a codebook of graphemes in the IAM and PSI databases. Their system accuracy was 95% in IAM database and 86% in PSI database. Also, Laurens van der Maaten et al. have used a combination of simple directional features and codebook of graphemes [24]. The method was tested on 150 writers and the system accuracy was 97%.

Vladimir Pervouchine et al. only focused on letters "t" and "h" on their English identification system. After detecting these shapes in the image, their skeletons are extracted. After that a cost function along the curve is calculated. The similarity of cost functions shows the writer [26]. It is obvious that this method cannot be extended for other languages.

In another research, Graham Leeham et al. have presented a method to identify the writer of numbers [28]. They have extracted features such as height, width, area, center of gravity, slant, number of loops, etc. Their system was tested among 15 people and the accuracy was 95%. This method should be tested on more popular and larger database to find its precise accuracy.

3. Proposed method

The proposed method consists of three main phases: Feature extraction, Graph generation, and classification. The sub-sections of this section present these phases.

3.1. Feature extraction phase

Two groups of features are extracted in this phase of proposed method. Gabor filter and XGabor filter are the tools that are used to extract the features. Gabor filter is a well-known filter which has been used to extract textural features in literature [1,2,3,15,20,25,35]. XGabor filter, is a mutation of the Gabor filter that we have designed in Ref. [1] to extract textural curve-based features. After explaining Gabor and XGabor filters in next two subsections, Section 3.1.3 describes how these two filters are employed on the handwriting Images.

3.1.1. Gabor filter

A 2D Gabor filter (Fig. 1c) is obtained by modulating a 2D sinusoid (Fig. 1a) with a 2D Gaussian (Fig. 1b) [25]. Let $g(x,y,\theta,\Phi)$ be the function defining a 2D Gabor filter centered at the origin

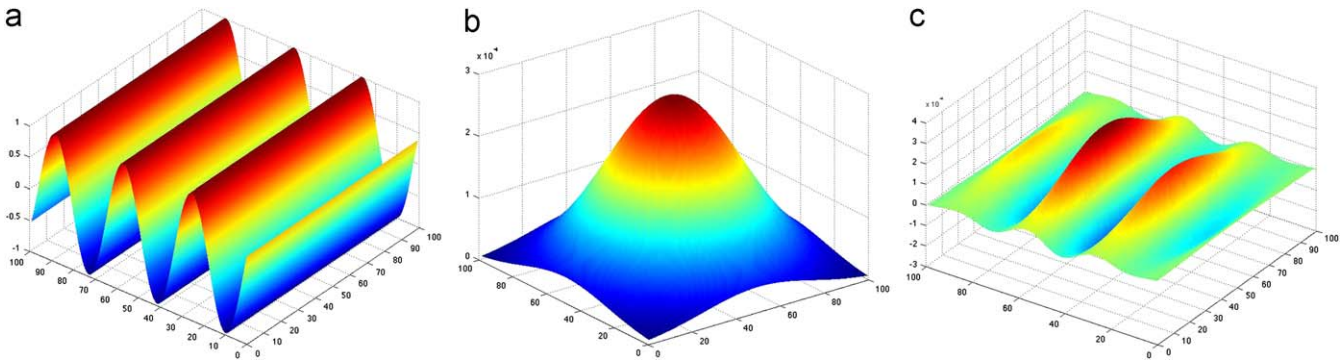

Fig. 1. (a) 2D sinusoid function, (b) 2D Gaussian function and (c) 2D Gabor filter which is obtained by modulating (a) and (b).

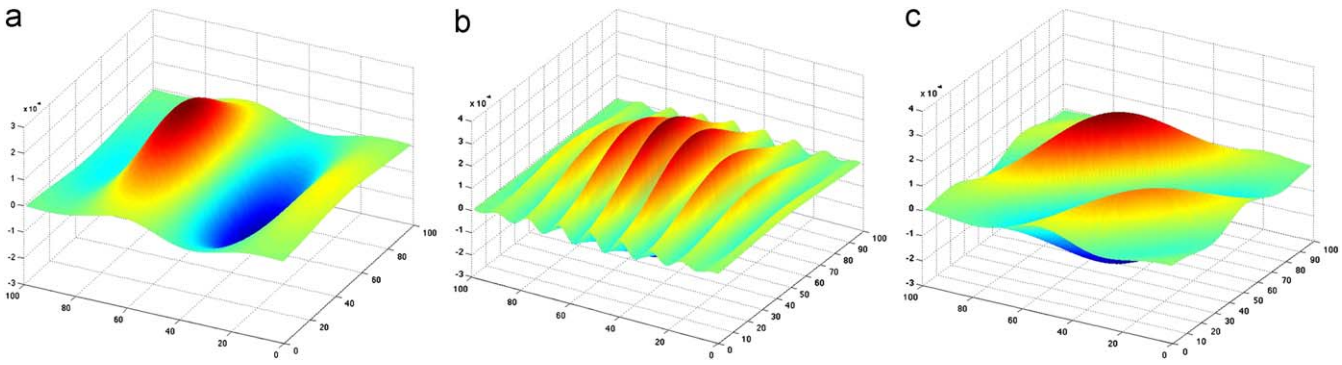

Fig. 2. (a) Gabor filter with θ=0, Φ=0.5, (b) Gabor Filter with θ=0, Φ=2 and (c) Gabor filter with θ=π/4, Φ=0.2.

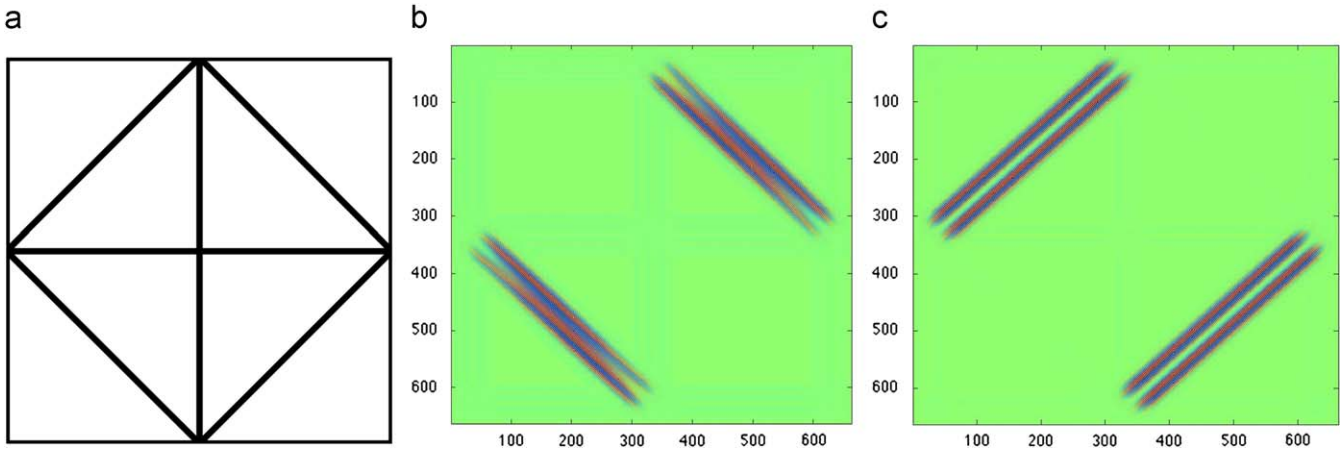

Fig. 3. (a) A sample image, (b) response of (a) to a Gabor filter with θ =π/4 and (c) response of (a) to a Gabor filter with θ =3π/4.

with Φ as the spatial frequency and θ as the orientation. Eq. (1) defines a Gabor filter:

$$g(x,y,\theta,\phi)=\exp\left(-\frac{x^2+y^2}{\sigma^2}\right)\exp(2\cdot\pi\cdot\phi\cdot i(x\cos\theta+y\sin\theta)) \quad (1)$$

It has been shown in Ref. [25] that σ, the standard deviation of the Gaussian kernel depends on the spatial frequency to be measured, i.e Φ. Fig. 2, shows 3D plots of some different Gabor filters.

The response of a Gabor filter to an image is obtained by a 2D convolution operation [31]. Let $I(x,y)$ denote the image and $G(x,y,\theta,\Phi)$ denote the response of a Gabor filter with frequency Φ and orientation θ to an image at point (x,y) on the image plane; G is obtained by Eq. (2)

$$G(x,y,\theta,\phi)=\int\int I(p,q)g(x-p,y-q,\theta,\phi)\,dp\,dq \quad (2)$$

The Gabor filter is mostly used to recognize frequent patterns. Because Gabor filter reacts to single lines and only depends on the gradient of the line, it has been used in writer identification feature extraction phase. Fig. 3 shows an example of this reaction. As it is shown in this figure, $G(x,y,\theta,\Phi)$ depends on how many lines with angle $(\pi/2-\theta)$ the writing has. The value of Φ should be adjusted according to the thickness of the lines the Gabor filter should detect.

3.1.2. XGabor filter

XGabor filter is another form of Gabor filter that we designed to react to the curves [1]. A 2D XGabor filter is obtained by modulating a 2D centered sinusoid with a 2D Gaussian. Fig. 4c shows an example of 2D XGabor, Fig. 4a shows a 2D centered sinusoid function and Fig. 4b shows a 2D Gaussian function. Let $xg(x,y,\phi,r_x,r_y)$ be the function defining an extended Gabor filter centered at the origin with Φ as the spatial frequency and r_x, r_y as the growth rates in x,y axis. Eq. (3) shows this filter.

$$xg(x,y,\phi,r_x,r_y)=\exp\left(-\frac{x^2+y^2}{\sigma^2}\right)\sin\left(\phi\cdot\frac{r_x\cdot x^2+r_y\cdot y^2}{r_x+r_y}\right) \quad (3)$$

In this equation σ is standard deviation [25]. Fig. 5 shows some examples of XGabor with different parameters. The response of an XGabor filter to an image is obtained by a 2D convolution operation. Let $I(x,y)$ denote the image and $XG(x,y,\phi,r_x,r_y)$ denote the response of an XGabor filter with frequency Φ and r_x, r_y as growth rates to an image at point (x,y) on the image plane. XG is obtained by using the following equation:

$$XG(x,y,\phi,r_x,r_y)=\int\int I(p,q)xg(x-p,y-q,\phi,r_x,r_y)\,dp\,dq \quad (4)$$

Fig. 6, shows the results of applying a circular and an elliptic XGabor filter on an image. However, XGabor does not get the exact curves out, but as Fig. 6 shows, the results are in relation with the curves of the original image.

3.1.3. Extracting features

To extract features, at first each document image is divided into non-overlapping blocks such that each line fits into one block. But if a line is not written completely, it may affect the results of the method, so each line that more than 2% of its pixels are black, is considered as a valid line. Using this definition, more than half of the line has to be used. The value of this threshold has been obtained by experiments. Fig. 7, shows two documents, at first one all lines are valid and in second one there is one invalid line. The size of each block is 1580×240 pixels.

To extract features, Gabor and XGabor filters are applied on valid lines. To apply the Gabor filter, θ was set to 36 different angles as 0, 5, 10, 15…175 and Φ was set to 4 (because the

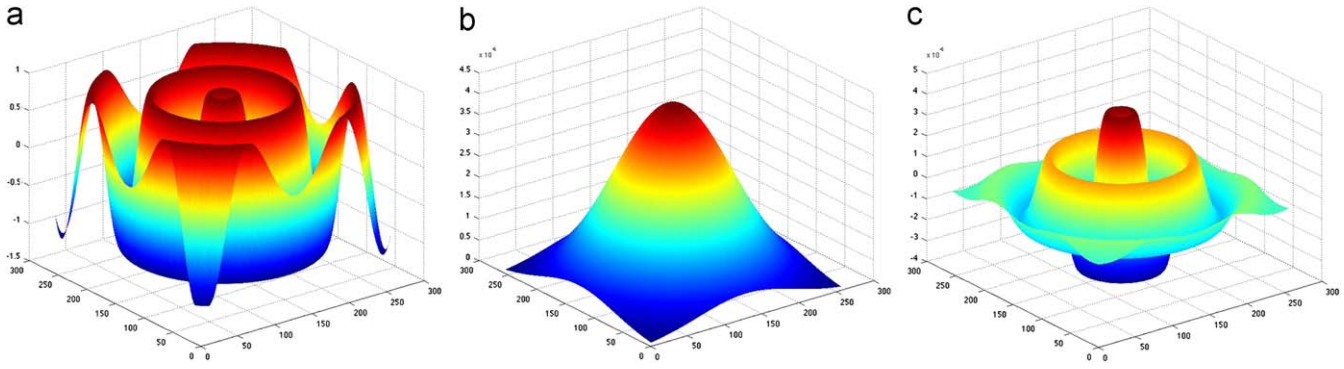

Fig. 4. (a) A 2D centered sinusoid, (b) a 2D Gaussian function and (c) the XGabor filter that is generated by modulating (a) and (b).

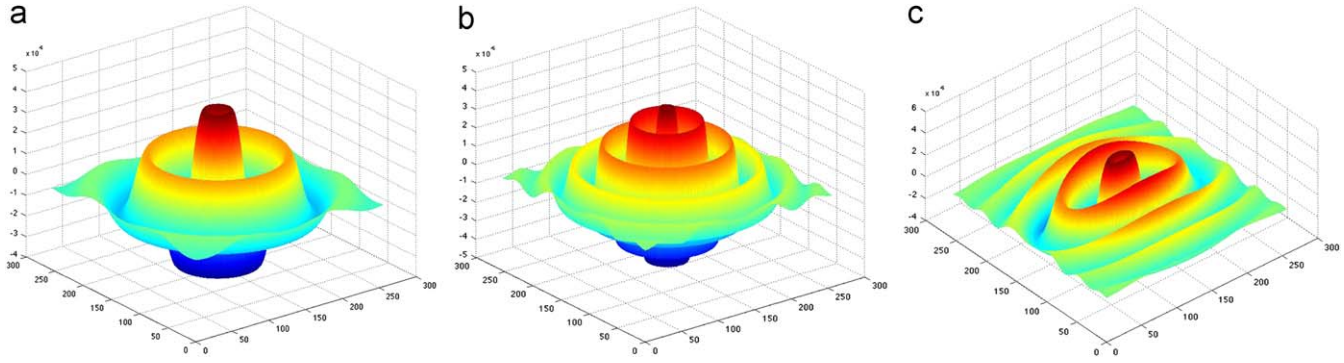

Fig. 5. (a) Circular XGabor with Φ=0.5, (b) circular XGabor with Φ=2 and (c) elliptic XGabor with Φ=2.

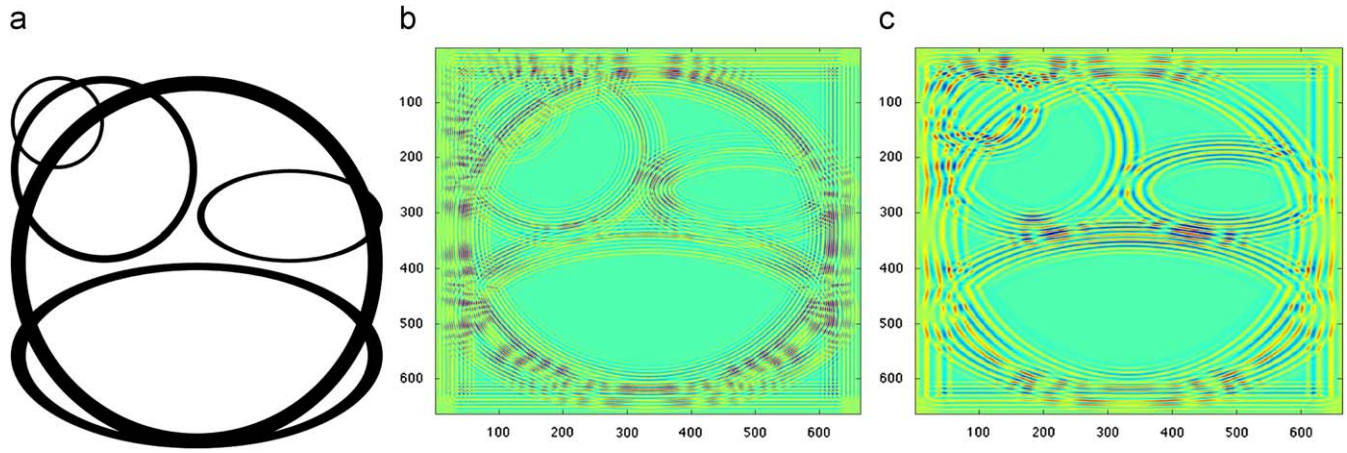

Fig. 6. (a) Sample image consists of 9 different elliptic shapes, (b) result of convolving of an elliptic XGabor with (a) and (c) result of convolving an circular XGabor with (a).

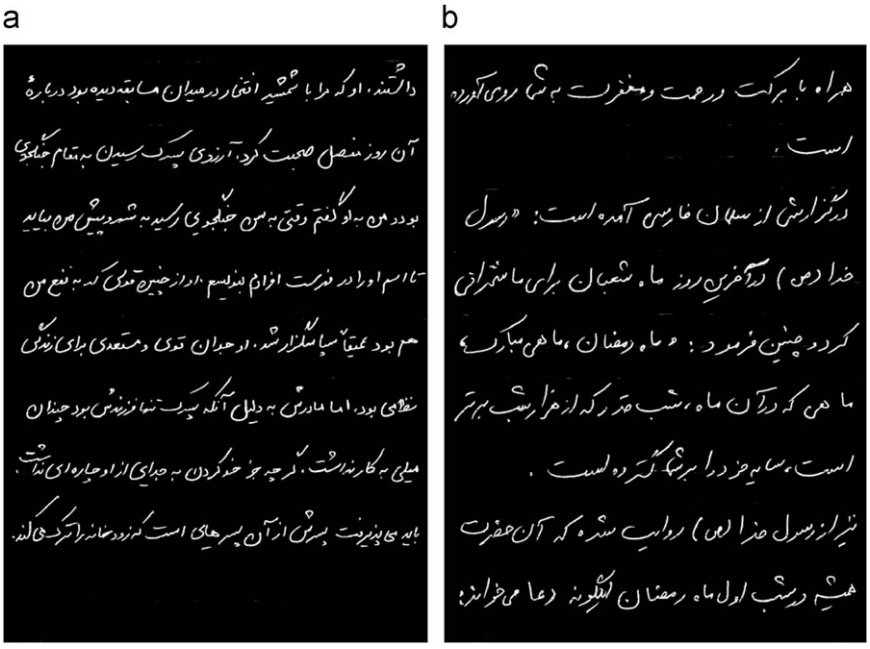

Fig. 7. (a) is a 8 lined document, which will be divided into 8 valid blocks (lines) (b) is a 9 lined document, which the second line does not fulfill the 2% threshold (the images are inversed to be more visible).

thickness of the written text was about 4 pixels and all texts in data set had been written by same pen), and the filter size was 128×128. After convolving the filters with lines of input images, the results are combined by using the following equation:

$$F_i = \sum_x \sum_y \left| G\left(x, y, \frac{\pi \cdot i}{36}, 4\right) \right| \tag{5}$$

where $1 \le i \le 36$ and x, y are pixels coordinates in valid image lines. Therefore, 36 features are extracted in this phase for each valid line of input image.

To implement the XGabor filter, 7 different ratios (r_x, r_y) along with 4 different filter sizes were used. The ratios were (1,1), (1,2), (1,3), (2,3), (2,1), (3,1) and (3,2). The filter sizes were 32×32, 64×64, 128×128 and 256×265. The results were combined by following equation to extract 28 features for each valid line of each input image:

$$F_{i+36} = \sum_x \sum_y |XG(x, y, 4, \alpha, \beta)| \tag{6}$$

Therefore, the feature vector of each valid line of each image instance (F) consists of 64 features. The feature vectors of valid lines are combined together using the following fuzzy approach.

3.2. Graph generation phase

Using the relations of the extracted features in each feature vector, a graph is generated. This graph is called feature relation graph (FRG). It is a directed, non-weighted graph which has exactly N nodes (N is the length of the feature vector). An edge (a,b) exists in the graph if feature "a" is always bigger than feature "b". The word "bigger" is defined by a Fuzzy approach in the following paragraph.

To define the relation between two extracted features, five fuzzy variables were defined. These variables were used to show the relationship between two features. These variables represent 5 different levels of relationship. Each one is called $\mu_{R_i}(a, b, j)$ which represents the i-th level relationship between feature "a"

and feature "b" calculated from the j-th line of document. Eqs. (7)–(11) show how these fuzzy variables are calculated:

$$\mu_{R_1}(a,b,j)=\begin{cases}1 & f_{ja}-f_{jb}<M_1\\ \dfrac{|M_1-(f_{ja}-f_{jb})|}{d} & M_1\le f_{ja}-f_{jb}\le M_1+d\\ 0 & f_{ja}-f_{jb}>M_1+d\end{cases} \tag{7}$$

$$\mu_{R_2}(a,b,j)=\begin{cases}\dfrac{|M_2-(f_{ja}-f_{jb})|}{d} & M_2-d\le f_{ja}-f_{jb}\le M_2+d\\ 0 & f_{ja}-f_{jb}<M_2-d \text{ OR } f_{ja}-f_{jb}>M_2+d\end{cases} \tag{8}$$

$$\mu_{R_3}(a,b,j)=\begin{cases}\dfrac{|M_3-(f_{ja}-f_{jb})|}{d} & M_3-d\le f_{ja}-f_{jb}\le M_3+d\\ 0 & f_{ja}-f_{jb}<M_3-d \text{ OR } f_{ja}-f_{jb}>M_3+d\end{cases} \tag{9}$$

$$\mu_{R_4}(a,b,j)=\begin{cases}\dfrac{|M_4-(f_{ja}-f_{jb})|}{d} & M_4-d\le f_{ja}-f_{jb}\le M_4+d\\ 0 & f_{ja}-f_{jb}<M_4-d \text{ OR } f_{ja}-f_{jb}>M_4+d\end{cases} \tag{10}$$

$$\mu_{R_5}(a,b,j)=\begin{cases}0 & f_{ja}-f_{jb}<M_5-d\\ \dfrac{|M_5-(f_{ja}-f_{jb})|}{d} & M_5-d\le f_{ja}-f_{jb}\le M_5\\ 1 & f_{ja}-f_{jb}>M_5\end{cases} \tag{11}$$

where f_{ja} is the a-th feature that is extracted from the j-th valid line for each person. In these equations, $M_1 \ldots M_5$ is defined based on maximum differences of the features. With regards to the experiments, the maximum difference between the features was detected as 0.012, therefore "M_1" was defined as -0.01, "M_2" as -0.005, "M_3" as 0, "M_4" as 0.005, and "M_5" as 0.01. In other words, $M_1 \ldots M_5$ are distributed uniformly in $[-M, M]$ where M is the maximum difference between features.

In Eqs. (7)–(11), d is the validity range of each variable. The value of d was defined as 0.005. Fig. 8 shows the structure of these variables.

Each variable corresponds with a linguistic relationship level. The value of first variable shows the concept of "*much lower*". The values of other variables show the concept of "*lower*", "*equal*", "*higher*", and "*much higher*", respectively.

After generating all the above variables, the relation between features a,b in the i-th level ($\mu_{R_i}(a,b)$) is defined by following equation where j is the line no of sample images of each writer:

$$\mu_{R_i}(a,b)=\underset{1\le j\le number\_of\_sample\_lines}{Average}\ \mu_{R_i}(a,b,j) \tag{12}$$

Eq. (13) shows if there is an edge from "a" to "b" in FRG:

$$(a,b)\in E \quad iff\ (-2\cdot\mu_{R_1}(a,b)-\mu_{R_2}(a,b)+\mu_{R_4}(a,b)+2\cdot\mu_{R_5}(a,b))\ge 1 \tag{13}$$

Eq. (13) is a simple weighted sum function. Also, the maximum value of this weighted sum may be equal to 2, because the sum of the five variable values is equal to 1. With regards to Eq. (13), an edge between a and b exists if "a" *is often* "*higher*" *than* "b". In other words:

$$(a,b)\in E \quad iff\ f_a>f_b\ most\_of\_the\_time \tag{14}$$

If (a,b) is an edge of the graph, that means feature a is greater than feature b most of the time. Therefore, this graph may not have a cycle because a cycle means that each one is greater than the other by transient rule. Such graphs have been called DAFRG (directed acyclic FRG). The created DAFRG is saved for each person in database in training phase.

3.3. Classification phase

In the classification phase, FRG of test data is generated and a method is needed to find out how much it is similar with trained ones. Such algorithms are called graph similarity scoring algorithms. There have been many researches in this field and many algorithms have been presented to solve this problem since now [38–42], but because these researches assumed the input data was a simple graph with no special constraints, the problem was an NP-Hard challenge. That means if the optimum result is needed, there will be no algorithm to achieve it in polynomial time but some polynomial algorithms exist if an approximate result is acceptable.

According to Section 3.2, the processed graphs are DAGs, it means FRG has some special properties. Therefore, a dynamic algorithm which has $\theta(ElgE)$ complexity would be used in classification phase to find optimum similarity of two DAFRGs.

3.3.1. Basics and definitions

The general idea of the algorithm is to find out how many paths exist that are common in both graphs. To present our algorithm, some definitions are needed which are defined as follows. These definitions are also shown in Fig. 9.

- source(root): a node that has no parent.
- sink(leaf): a node that has no child.
- depth of v: the length of the longest path from a source to vertex v in graph $G\cdot(D_G(V))$.
- height of v: the length of the longest path from a sink to vertex v in graph $G\cdot(H_G(v))$.

$T(i)$ is defined as the number of the common paths between graph G_1 and graph G_2 that starts from vertex i. Because the graphs are

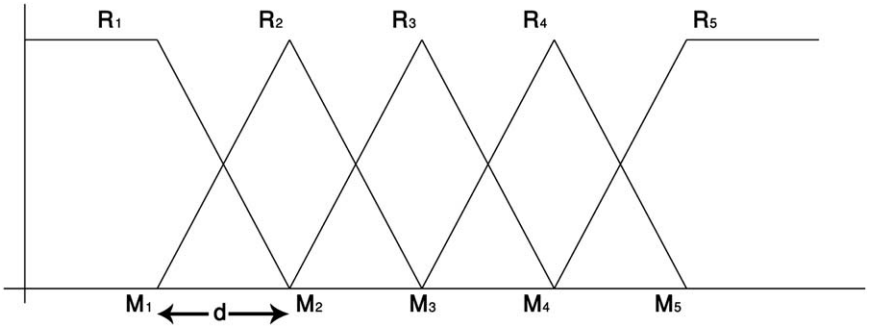

Fig. 8. Fuzzy sets of relation levels.

DAGs, the equation is written as

$$T_{G_1,G_2}(i) = \sum_j (T(j)+1) | (i,j) \in G_1 \& (i,j) \in G_2 \quad (15)$$

Fig. 10 shows how Eq. (15) works for two typical graphs.

For example the paths starring from node 1 that are common in both graphs are:

(1–2)
(1–3)
(1–2–5)
(1–3–5)

Therefore, with regards to definition of T, the similarity score is presented as

$$SimS(G_1, G_2) = \sum_i T_{G_1,G_2}(i) \quad (16)$$

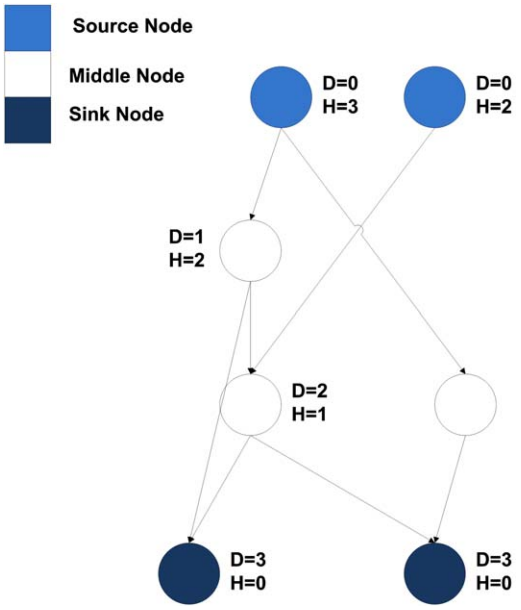

Fig. 9. A sample DAG: D represents depth, H represents height.

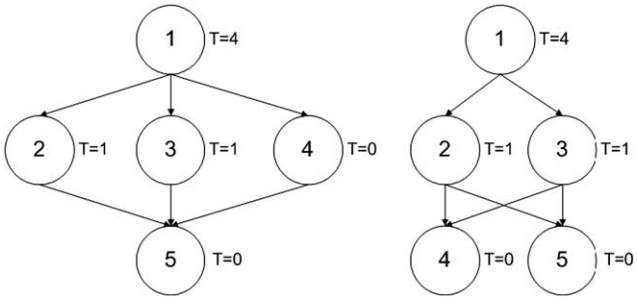

Fig. 10. Calculating T values according to Eq. (15).

```
H (node x , graph G )
        max_child_height = 0
        for all children of x in G as q
                if height(q) is not calculated
                        H( q , G )
                max_child_h = max ( max_child_h , height( q ))
        height ( x ) = max_child_h + 1
```

Fig. 11. A recursive (DFS based) algorithm, calculating the height of the nodes.

```
CalcT( X )
        for j = 1 to n
                T(j) = 0
        for i = 1 to E
                if ( X( i ) is in G1 & X( i ) is in G2 )
                        T( X( i , 1 ) ) += T( X( i , 2 ) ) + 1
```

Fig. 12. The dynamic implementation to calculate the T value

Table 1
Accuracy of proposed method on gathered database.

Training lines/testing lines	First approach (%)	Second approach (%)
5/35	58.40	37.18
10/30	71.23	55.41
15/25	87.56	80.20
20/20	94.62	93.76
25/15	98.61	97.23
30/10	99.04	98.57
35/5	100.00	99.41

First column shows the ratio of the training and testing data. Second column shows the result of the system when testing and training have been done by first approach and third columns shows results when second approach has been used in test and train.

It is obvious that the similarity score of same graphs is has maximum value.

3.3.2. *The implementation and time complexity*

The graph similarity algorithm has four steps:

1. *Height of each node is calculated*: To calculate the height, a recursive algorithm is used. This algorithm is shown in Fig. 11. This algorithm is a Depth First Search (DFS) [32] based algorithm, so its complexity is same as the complexity of the DFS which is $\theta(E)$ where E is the number of the edges in the graph.
2. *The edges are sorted according to the nodes height*: The edges that are lower than the others must be processed first. The relation between two edges is defined as follows:
$$(a,b) < (c,d) \quad \text{iff } H_{G_1}(a) < H_{G_1}(c) \text{ or } H_{G_2}(a) < H_{G_2}(c) \quad (17)$$
The quick sort algorithm is used to sort edges. The complexity of quick sort is $\theta(ElgE)$.
3. *Edges are processed in their descending sorted order to calculate* T: The pseudo code of algorithm that calculates T is shown in Fig. 12.
It's obvious that the complexity of this part is $\theta(E+V)$, where V is number of the nodes (vertices).
4. *With regards to* Eq. (16), *the graph similarity scores are calculated*: Therefore, the complexity of algorithm is $\theta(E)+\theta(ElgE)+\theta(E+V)=\theta(ElgE+V)$.

4. Experimental setups and results

In order to test the designed system, PD100 data set was used. As it was mentioned, PD100 contains 100 people's handwriting such that each person was asked to write 5 different pages in Persian. Its pages are in size of A5 and each person has been asked to write about 8–9 lines on that. There were no constraints in what people should write, so some of them wrote novels and some poems. Because in the test procedure, different number of lines were used as training set, the authors should had enough lines for training otherwise the results of different test were not comparable, therefore, if someone wrote <8 lines on each page,

Table 2
The precision of applying different feature extraction and classification techniques on PD100 data set.

Ref.	Features				Classifier					Precision (%)
	Gabor	XGabor	Grapheme based	Gradient based	WED	Neural network	LCS based	Fuzzy KNN	FRG	
[1]	•				•					55
[1]		•			•					40
[1]	•	•			•					77
[45]	•	•					•			89
[46]	•	•			•		•			95
[48]				•		•				94
[47]			•	•				•		96
–	•	•				•				97
–	•	•							•	**100**

Table 3
Comparing recent writer identification published papers in Persian and Arabic.

Ref.	Language	Dataset properties			Reported result	Method
		Number of writers	Samples per. writer	Train/test ratio		
This	Persian	80[a]	40 lines (5 8-lined pages)	3/2	Top-1:98%	Off-line, texture based
[1]	Persian	80[a]	5 A5 pages	3/2	Top-1: 77%	Off-line, texture based
[7]	Arabic	100	320 words (16 different types)	3/1	Top-10: 90%	Off-line, model based
[8]	Arabic	350	5 images	4/1	Top-1: 88%	Off-line, texture+model based
[19]	Persian	128	(32 disjoint characters)	Not mentioned	Top-1: 90%	Off-line, character based
[20]	Persian	25	1 page (divided in two)	3/1	Top-1: 92%	Off-line, texture based
[45]	Persian	100[a]	5 A5 pages	3/2	Top-1: 95%	Off-line, texture based
[46]	Persian	50[a]	5 lines	3/2	Top-1: 94%	Off-line, texture based
[47]	Persian	50[a]	5 lines	3/2	Top-1:90%	Off-line, texture based

[a] The database is the same (PD100).

Table 4
Comparing recent writer identification published papers in other languages.

Ref.	Language	Dataset properties			Reported result	Method
		Number of writers	Samples per. writer	Train/test ratio		
This	Persian	80	40 lines	3/2	Top-1:98%	Off-line, texture based
[2]	English	200	8 paragraph of about 8 lines	6/2	Top-1: 98.5%	On-line
[3]	Chinese	500	2 images	1/1	Top-1: 40%	Off-line, texture based
[5]	Chinese	500	2 images	1/1	Top-1: 39.2	Off-line, texture based
[13]	English	165	15–30 patterns	Not mentioned	Top-1: 95%	Off-line, character based
[14,27]	English	150	One paragraph (divided in two)	1/1	Top-1: 72%	Off-line, character based
[17,23]	English	100 of IAM	5 pages	Not mentioned	Top-1: 97%	Off-line, character based
[21,22]	English	650 of IAM	5 pages	Not mentioned	Top-1: 95%	Off-line, model based
[24]	English	150	Not mentioned	Not mentioned	Top-1: 97%	Off-line, texture+model based
[28]	English	15	(10 digits)	Not mentioned	Top-1: 95%	Off-line, character based

some random lines were divided to create enough lines (see Table 1). Also, in order to have fewer difficulties in preprocessing of the system and having more focus on the feature extraction and the classification phase, the writers were given the same pen to write. The pen was blue, so a simple color filter could extract the written lines.

The system was tested in two different approaches. In the first approach, the lines of data were divided in two groups randomly. Then the first group of lines was given to the system as the training data and the second group was given to the system as the test data. The sizes of the groups were variant in different tests. As presented in Table 1, the sizes are all the numbers dividable by $5 \leq 35$. It is worth to mention that because different number of lines were used in training phase of each test.

In the second approach, two groups of lines were generated for training and testing. In this approach, the lines were not selected randomly and they were selected sequentially from the first line that was gathered from the writer.

The results of these two approaches in the different sizes of training and testing data are presented in Table 1.

However, it was possible to use pages instead of lines in training, but using lines makes the test procedure more real and increases its precision, because different pages do not have same amount of information necessarily, that is it is possible to have only two words in a page or a plenty of words on.

Also, in another test, we were going to find the best combination of feature selection and classification methods when it was applied in PD100 data set. Table 2 presents the result of applying different ones. As it is presented in Table 2, the best precision occurred when Gabor and XGabor were used as feature extraction method and FRG was used as classifier approach. It is worth mentioning that some rows of Table 2 are the result of

some other researches that were published by us, also, we tried to use almost same data for train and test.

Recent writer identification approaches in Arabic and Persian Languages have been compared with proposed method in Table 3. In Table 3, the best results of methods have been reported. Also, the methods that used the same data set with us have been marked by an asterisk. As it is presented in Table 3, the proposed method has outperformed the other ones.

Table 4 compares the proposed method with some well known methods in other languages. To make the proposed method comparable with related methods, the Train/Test ratio has been mentioned based on pages. As it is presented in Table 4, the proposed method has better results than other offline method, however, because the data set of methods are not same, only the structure comparison of methods is reasonable.

5. Conclusion

In this paper, we have proposed a method to identify Persian writers. In the feature extraction part, Gabor and XGabor filter are used with different directions. The proposed method is based on new concept that is called FRG. FRG is a graph that is created based on some fuzzy variables. In the classification phase, FRG graphs are compared together using graph similarity measures. To test the presented method; we gathered a database of Persian handwritten instances. It consisted of 80 writes. Each writes wrote 5 handwritten pages. Two different approaches were used to test the method. At first one, some random lines of input data for each writer were used to train and other lines were used to test. At the second approach, lines were selected sequentially for training. In both case, the accuracy of the method was great when number of training data was enough and it was about 98% in average. The proposed method is text independent and its result was better than many related works. In the future, we are going to extend this method to develop a language independent identification approach. However, the feature extraction phase of proposed method has been designed for Persian language, but based on the filters characteristics; they can be used for other languages if proper directions use.

References

[1] B. Helli, M.E. Moghaddam, Persian writer identification using extended Gabor filter, in: International Conference on Image Analysis and Recognition (ICIAR), 2008.

[2] A. Schlapbach, L. Marcus, H. Bunke, A writer identification system for on-line whiteboard data, Pattern Recognition Journal 41 (2008) 23821–23897.

[3] Z. He, X. You, Y.Y. Tang, Writer identification of Chinese handwriting documents using hidden Markov tree model, Pattern Recognition Journal 41 (2008) 1295–1307 2008-06-15.

[4] A. Schlapbach, H. Bunke, Off-line writer identification and verification using Gaussian mixture models, studies in computational intelligence, vol. 90, Springer, Berlin, 2008, pp. 409–428.

[5] H. Zhenyu, Y. Xinge, Y.Y. Tang, Writer Identification using global wavelet-based features, Neurocomputing 71 (2008) 1832–1841.

[7] S. Al-Ma'adeed, E. Mohammed, D. Al Kassis, F. Al-Muslih, Writer identification using edge-based directional probability distribution features for arabic words, in: IEEE/ACS International Conference on Computer Systems and Applications (AICCSA), 2008.

[8] M. Bulacu, L. Schomaker, A. Brink, Text-independent writer identification and verification on offline arabic handwriting, in: Ninth Conference on Document Analysis and Recognition (ICDAR), 2007.

[10] L. Schomaker, Advances in Writer identification and verification, in: Ninth International Conference on Document Analysis and Recognition (ICDAR), 2007.

[11] A. Al-Dmour, R.A. Zitar, Arabic writer identification based on hybrid spectral-statistical measures, Journal of Experimental & Theoretical Artificial Intelligence 19 (4) (2007) 307–332.

[12] M. Bulacu, L. Schomaker, Text-independent writer identification and verification using textural and allographic features, IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI) 29 (4) (2007) 701–717 Special Issue—Biometrics: Progress and Directions.

[13] V. Pervouchine, G. Leedham, Extraction and analysis of forensic document examiner features used for writer identification, Patter Recognition Journal 40 (2007) 1004–1013.

[14] L. Schomaker, K. Franke, M. Bulacu, Using codebooks of fragmented connected-component contours in forensic and historic writer identification, Pattern Recognition Letter 28 (2007) 719–727.

[15] V. Eglin, S. Bres, C. Rivero, Hermit and Gabor transforms for noise reduction and handwriting classification in ancient manuscripts, International Journal of Document Analysis and Recognition, IJDAR 9 (2007) 101–122 doi: 10.1007/s10032-007-0039-z, Springer, 2007.

[16] R. Neils, L. Vuurpijl, L. Shomaker, Automatic allograph matching in forensic writer identification, International Journal of Pattern Recognition and Artificial Intelligence 21 (1) (2007) 61–81.

[17] A. Schlapbach, H. Bunke, A writer identification and verification system using HMM based recognizers, Pattern Analysis Application (Springer) 10 (2007) 33–43, doi:10.1007/s10044-006-0047-5.

[18] M. Bulacu, L. Schomaker, Combining multiple features for text-independent writer identification and verification, in: 10th international Workshop on Frontiers in Handwriting Recognition (IWFHR), 2006.

[19] M. Soleymani Baghshah, S. Bagheri Shouraki, S. Kasaei, A novel fuzzy classifier using fuzzy LVQ to recognize online persian handwriting, in: Second IEEE Conference on Information and Communication Technology (ICTTA), 2006.

[20] F. Shahabi, M. Rahmati, Comparison of Gabor-based features for writer identification of Farsi/Arabic handwriting, in: 10th International Workshop on Frontiers in Handwritten Recognition (IWFHR), 2006.

[21] A. Bensefia, T. Paquet, L. Heutte, A writer identification and verification system, Pattern Recognition Letters 26 (2005) 2080–2092.

[22] A. Bensefia, T. Paquet, L. Heutte, Handwriting document analysis for automatic writer recognition, Electronic Letters on Computer Vision and Image Analysis (ELCVIA), 2005.

[23] A. Schlapbach, H. Bunke, Writer identification using an HMM-based handwriting recognition system: to normalize the input or not?, in: 12th Conference of the International Graphonomics Society, Salerno, Italy, June 26–29, 2005, pp. 138–142.

[24] L. van der Maaten, E. Postma, Improving automatic writer identification, in: 17th Belgium-Netherland Conference on Artificial Intelligence, 2005.

[25] V. Shiv Naga Prasad, J. Domke, Gabor Filter Visualization, Technical Report, University of Maryland, 2005.

[26] V. Pervouchine, G. Leedham, K. Melikhov, Handwritten character skeletonisation for forensic document analysis, in: ACM Symposium on Applied Computing, 2005.

[27] M.B.L. Schomaker, Analysis of texture and connected-component contours for the automatic identification of writers, in: 16th Belgium–Netherland Conference on Artificial Intelligence (BNAIC), 2004.

[28] G. Leeham, S. Chachra, Writer identification using innovative binerized features of handwriting numerals, in: Seventh International Conference on Document Analysis and Recognition (ICDAR), 2003.

[29] M. Bulacu, L. Schomaker, L. Vuurpijl, Writer identification using edge-based directional features, in: Seventh International Conference on Document Analysis and Recognition (ICDAR), 2003.

[30] S.-H. Cha, C.C. Tappert, Automatic detection of handwriting forgery, in: Eighth International Workshop on Frontiers in Handwriting Recognition (IWFHR), 2002.

[31] R.C. Gonzalez, R.E. Woods, Digital Image Processing, second ed., Prentice Hall, Inc., Englewood Cliffs, NJ, 2002.

[32] T.H. Cormen, C.E. Leiserson, R.L. Rives, C. Stein, Introduction to Algorithms, second ed., The MIT Press, 2001.

[33] Y. Zhu, T. Tan, Y. Wang, Biometric personal identification based on handwriting, in: 15th International Conference on Pattern Recognition, 2000.

[35] H.E.S. Said, G.S. Peake1, T.N. Tan, K.D. Baker, Writer identification from non-uniformly skewed handwriting images, in: Ninth British Machine Vision Conference, 1999.

[37] K. Franke, G. Grube, The automatic extraction of pseudo-dynamic information from static images of handwriting based on marked gray value segmentation, Journal of Forensic Document Examination 11 (1998) 17–38.

[38] V.D. Blondel, A. Gajardo, M. Heymans, P. Senellart, P. Van Dooren, A measure of similarity between graph vertices: applications to synonym extraction and web searching, SIAM Review 46 (4) (2004) 647–666.

[39] S. Melnik, H. Garcia-Molina, E. Rahm, Similarity flooding: a versatile graph matching algorithm and its application to schema matching, in: Proceedings of the 18th ICDE Conference, 2002.

[40] L. Zager, Graph similarity and matching, Masters' thesis report, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, May 2005.

[41] O. Sokolsky, S. Kannan, I. Lee, Simulation-Based Graph Similarity, Departmental Papers (CIS), Department of Computer and Information Science, University of Pennsylvania, 2006.

[42] T. Törnfeldt, Graph Similarity, Parallel Texts, and Automatic Bilingual Lexicon Acquisition, Masters' thesis report, Department of Mathematics, Linköpings Universitet, April 2008.

[43] J. Chapran, Biometric writer identification: feature analysis and classification, International Journal of Pattern Recognition and Artificial Intelligence 20 (4) (2006) 483–503.

[44] J. Chapran, M.C. Fairhurst, Biometric writer identification based on the interdependency between static and dynamic features of handwriting, in:

Proceedings of the 10th International Workshop on Frontiers in Handwriting Recognition, 2006, pp. 505–510.
[45] B. Helli, M.E. Moghaddam, A text-independent Persian writer identification system using LCS based classifier, in: IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2008.
[46] B. Helli, M.E. Moghaddam, A writer identification method based on XGabor and LCS, IEICE Electronics Express 6 (10) (2009).
[47] S.S. Ram, M.E. Moghaddam, Text-independent Persian writer identification using fuzzy clustering approach, in: International Conference on Information Management and Engineering (ICIME), Malaysia, 2009.
[48] S.S. Ram, M.E. Moghaddam, A Persian writer identification method based on gradient features and neural networks, in: Second International Conference on Image and Signal Processing (CISP), China, 2009.

About the Author—BEHZAD HELLI is a M.Sc. student in computer engineering. He has done this work as his B.Sc. and M.Sc. thesis under supervision of Dr Ebrahimi Moghaddam. He was the first rank student in his B.Sc. and M.Sc. studies. His research interest is image processing and he work in image processing Lab of SBU. He is going to be a Ph.D. student in next semester.

About the Author—MOHSEN EBRAHIMI MOGHADDAM has Ph.D. degree in computer engineering. He received his degree from Sharif university of Technology. He is a faculty member of Electrical and computer engineering of Shahid Beheshti University. His research interests are Image processing, Multimedia operating systems and multimedia sensor networks.