Information Processing and Management 56 (2019) 354–366

Contents lists available at ScienceDirect

Information Processing and Management

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

Writer identification approach based on bag of words with OBI features

Amal Durou[a,*], Ibrahim Aref[a], Somaya Al-Maadeed[b], Ahmed Bouridane[a], Elhadj Benkhelifa[c]

[a] Department of Computer and Information Sciences, Northumbria University, Newcastle Upon Tyne, NE1 9ST, UK
[b] Department of Computer Science and Engineering, Collage of Engineering, Qatar University, Doha, 2713, Qatar
[c] School of Computing, Staffordshire University, Stoke on Trent, UK

ARTICLE INFO

Keywords:
Writer identification
Oriented basic image
Kernel principal component analysis
Graphemes
Text independent classification

ABSTRACT

Handwriter identification aims to simplify the task of forensic experts by providing them with semi-automated tools in order to enable them to narrow down the search to determine the final identification of an unknown handwritten sample. An identification algorithm aims to produce a list of predicted writers of the unknown handwritten sample ranked in terms of confidence measure metrics for use by the forensic expert will make the final decision.

Most existing handwriter identification systems use either statistical or model-based approaches. To further improve the performances this paper proposes to deploy a combination of both approaches using Oriented Basic Image features and the concept of graphemes codebook. To reduce the resulting high dimensionality of the feature vector a Kernel Principal Component Analysis has been used. To gauge the effectiveness of the proposed method a performance analysis, using IAM dataset for English handwriting and ICFHR 2012 dataset for Arabic handwriting, has been carried out. The results obtained achieved an accuracy of 96% thus demonstrating its superiority when compared against similar techniques.

1. Introduction

Writer identification remains a challenging biometric recognition application. It is carried out as a pattern recognition problem to allocate an unknown written sample/pattern to one class (e.g., a writer) out of a set of classes (writers) (Schlapbach, 2007). Therefore, the process of writer identification can be defined as an algorithm/tool to assign a handwriting sample to one author/writer (Awaida & Mahmoud, 2012; Sreeraj & Idicula, 2011). This problem has received significant interest by the research community and various methods have been proposed (Awaida & Mahmoud, 2012; Sreeraj & Idicula, 2011; Tan, Viard-Gaudin, & Kot, 2009; Yang, Jin, & Liu, 2016). However, a number of issues are still unsolved including an insufficiency of datasets and handwriting material in different languages. Currently, there exists a number of writer identification systems developed for various applications including forensic science, document analysis, investigation of the historical documents (Abdi & Khemakhem, 2012; Wu, Tang, & Bu, 2014).

A typical writer identification system operates in two methods: offline and online. The writing behaviour in the online mode is taken from the writer in real-time by converting it into list of signals and directions via a writing tablet. However, for offline mode, the scanned and digitized handwritings images are used to identify the writer (Awaida & Mahmoud, 2012; Raj & Chaudhary, 2016).

* Corresponding author.
E-mail address: amal.durou@northumbria.ac.uk (A. Durou).

http://dx.doi.org/10.1016/j.ipm.2017.09.005
Received 1 November 2016; Received in revised form 22 August 2017; Accepted 29 September 2017
Available online 06 October 2017

Furthermore, writer identification can be classified into two approaches: text-dependent and text-independent (Siddiqi &Vincent, 2010). In the text-dependent method, all writers write the same known text (Al-Maadeed, 2012; Wu et al., 2014). In the text-independent method, the dataset contains different text and sometimes different languages. The writer features can be captured using statistical features taken from different handwritten page to generate the set of features which are insensitive to the texts (Bulacu & Schomaker, 2007).

Moreover, there are two approaches of writer identification systems: statistical and model-based approaches. A statistical approach system analyses statically the set of extracted features from the handwritten text (Bulacu, Schomaker, & Vuurpijl, 2003), on the other hand, the model-based approach uses graphemes which are limited samples of handwritten strokes (Bulacu & Schomaker, 2007; Kam, Fielding, & Conn, 1997). Both approaches consist of two phases: a feature-extraction phase and a classification phase. In the feature extraction phase, the features are extracted from the handwriting scripts and the generated features are then analysed for their distinctive power before stored as a single feature vector. In the classification phase, the resulting feature vector is matched and assigned into different classes that best represent the authors.

The main objectives of this paper are twofold: (i) to develop an effective feature extraction that best distinguish handwriting patterns. To achieve this, a combination of a multiscale feature extraction with the concept of grapheme is judiciously carried out to capture and extract the several discriminating features such as handwriting curvature, direction, wrinkliness and various edge based features and (ii) develop a novel text-independent writer identification system for offline Arabic and English writings captured as scanned images.

The remainder of this paper is organized as follows: Section 2 discusses previous approaches relating to handwriting identification. The proposed approach is then detailed in Section 3. Section 4 discusses the experiments carried out to obtain results and their validation including a comparative study against some previous works. Finally, Section 5 concludes the paper.

2. Related works

As mentioned previously, a writer identification system allocates an unknown script to specific writer from a group of possible writers. To achieve this, a one-to many-search needs to be carried out on a large database containing handwriting samples of known writers to return the result as a list of candidates (Awaida & Mahmoud, 2012; Sreeraj & Idicula, 2011). The recognition process is still a very challenging task due to the fact that each person has different handwriting qualities and styles that are subject to inter-writer and intra-writer variations (Moubtahij, Halli, & Satori, 2014). During the last few years, significant research efforts have been devoted to tackle the problems associated with handwriting identification resulting in a plethora of papers; however, the topic still remains an active research area having many useful applications including forensic and historical document analysis (Sreeraj & Idicula, 2011). Early works on writer identification were mainly aimed for the English language. Recently, the research community has targeted other languages, such as Chinese, Dutch, Arabic and Greek, resulting in a significant contribution to this field (Awaida, Mahmoud, 2013). To the best of our knowledge, although there are many researchers working in writer identification of Arabic language, there are many issues still active and under investigation.

Several statistical and model based features were proposed in Paraskevas, Stefanos, and Ergina (2014) where the authors proposed an approach to improve the statistical feature extraction using an edge hinge distribution. Moreover, the authors explored a combination of this feature extraction approach with a codebook of the graphemes. The system was evaluated using the Firemaker database, which consists of 250 writers with 4 pages per writer. Newell and Griffin (2014) used a writer identification system using the concept of oriented Basic Image Feature Columns (oBIF Columns) and the authors proposed how a texture-based scheme can be enhanced by encoding a writer's style as the deviation from the mean encoding for a population of writers. In Hannad, Siddiqib, and El-Kettan (2016), a system was presented using a texture based approach for the identification of writers from offline handwritten images. The proposed method was implemented by dividing a handwriting script into small fragments where each fragment was processed individually as a texture. The authors used both Arabic and English text from IFN/ENIT and IAM databases to evaluate the performances. Fiel and Sablatnig (2012) proposed an approach for writer retrieval and writer identification based on texture features. In both cases, a codebook was generated using the Scale Invariant Feature Transform (SIFT) from different pages of the handwriting. Then, a histogram is generated and used to identify a writer or retrieve the documents of one particular writer. The IAM dataset was used for the evaluation resulting in an identification rate of 90.8%. Tang, Wu, and Bu (2013) proposed two feature extraction methods: the stroke fragment histogram (SFH) based on a codebook and a local contour pattern histogram (LCPH) generated by tracking the points on the contours of the handwriting images. Identification rates of 91.3% and 85.4% were obtained for SFH and LCPH, respectively. Another approach to evaluate the identification performance of five highly discriminating features was proposed (Daniels & Baird, 2013). The five classes of features investigated are: slant and slant energy, skew, pixel distribution, curvature, and entropy. Vasquez, Travieso, and Alonso (2013) presented a writer identification system using graphometrical and forensic features using an Artificial Neural Network (ANN) for the classification task. A database of 100 users with 10 samples per subject was constructed and the system achieved an identification performance of 94.6%.

Ghiasi and Safabakhsh (2013) proposed a text-independent writer identification system using the histogram of the codebook shapes to generate a feature vector for each manuscript. Furthermore, the technique uses cursive handwritings with a rich content of dissimilar shapes present in the handwriting connected components. Only part of the connected components were used to avoid complex patterns. Two approaches were used to extract codes from the contours. First, using the actual pixel coordinates of contour fragments. Second, using a linear piece-wise approximation using the lengths and angles of the segment. The two methods were evaluated using two English and three Farsi handwriting databases. The authors concluded that both methods are promising. However, the performance of later method is better than the first method.

Furthermore, a writer identification system for offline text-independent Arabic language was proposed in Abdi and Khemakhem (2015). The main idea of this method uses a beta-elliptic model in order to generate a synthetic codebook. In this algorithm, a feature selection was proposed to reduce the codebook's size where the feature extraction is performed using a template matching approach. The authors in Djeddi, Meslati, Siddiqi, Ennaji, El Abed, and Gattal (2014) proposed a handwriting-based identification system for Arabic handwritten documents. Their proposed method consists of two main stages: first, the system processes each handwritten image and extracts features such as edge-direction, edge-hinge, and run lengths features. Then, these features are fed to a Multiclass SVM (Support Vector Machine) for classification. The method was trained and tested on a large database of Arabic handwritings written by 1000 different writers. The authors reported that the best result was achieved when combining run-length and edge-hinge features achieving a classification rate of 84.10%.

Finally, the authors in Khalifa, Al-maadeed, Tahir, and Bouridane (2015) proposed a writer identification method using codebook extension model with an ensemble of codebooks in which a kernel discriminant analysis using spectral regression (SR-KDA) was deployed as a dimensionality reduction technique to avoid over-fitting problem. Two datasets were used in evaluation with single codebook and using multiple codebooks sizes. Furthermore, the authors conclude that the proposed method is competent when compared against existing methods.

As discussed previously most existing hand writer identification systems either use a statistical extraction method or a model based approaches with efforts made to the feature selection and dimensionality reduction using robust classifiers. It is also known that both feature extraction approaches have useful advantages including some limitations on their own. This paper proposes to combine the two approaches and develop a novel statistical and model based feature extraction approach in order to improve the recognition performances.

3. Proposed methodology

As mentioned previously, most existing handwriting identification systems are based on two approaches: a statistical approach or a model-based approach. These approaches have some limitations that lower the performance of the handwriting identification system. Therefore, this paper proposes an automatic handwriting identification system by combining both approaches. This is achieved by fusing Oriented Basic Image (OBI) features with a codebook of graphemes in order to improve the recognition performances of the work described in (Khalifa et al., 2015). The IAM (English) and ICFHR-2012 (Arabic) databases have been used for evaluation especially that both have several discriminating features such as handwriting curvature, direction, wrinkliness and some edge based features which require an efficient feature extraction strategy. In this paper and due to the uniqueness of the features, we have investigated various methods including SIFT, Speed Up Robust Features (SURF) and OBIs. An initial investigation has shown that OBI method outperforms others, therefore this method has been used as a base. We extract OBI features using a multi-scale approach with local symmetry and orientation. We use different orders and directions of multi scale Gaussian derivative filter with to generate a number of features. Then, orientations and scales of the features a histogram is generated based on the symmetry, which, once normalized, generates the final feature vector. The generated feature vector is then combined with a grapheme based codebook to investigate the system identification performances.

Furthermore, to maintain the system resources and increase the system operation speed, one needs to decrease the length of feature vectors. The Kernel Principal Component Analysis (KPCA), reduction technique, has been used because of its simplicity and effectiveness compared to other methods. For classification, various classifiers were initially tested including K-Neighrest Neighbour, Support Vector Machine (SVM), and Neural Networks. The 1-NN with Euclidian distance has been adopted as it provides the most effective results. The experimentation has been carried out using the two datasets (English and Arabic) using a single codebook first. However, the results obtained depicted below have shown that the performance was marginally similar to that of proposed in Khalifa et al. (2015). Therefore, to further improve the system performance, a multiple codebook approach has been investigated. In this paper we propose a writer identification system based on combining different OBI features with different graphemes codebook. The system overall stages are illustrated in Fig. 1 and will be discussed in the next sections.

3.1. Datasets

To evaluate the performances of the proposed approach experiments were carried out using two datasets: the IAM (Marti & Bunke, 2002) English dataset and the ICFHR-2012 (Hassaine & Al-Madeed, 2012) Arabic dataset.

3.1.1. IAM dataset

The IAM English handwritten dataset (Marti & Bunke, 2002) is one of the most widely used datasets for the evaluation and validation of handwriter identification and verification systems. The dataset comprises of 1539 English handwriting documents generated from 657 writers and saved as digital images having a resolution of 300 dpi. To ensure a fair evaluation of the proposed technique a similar environment, as used by Khalifa et al. (2015) and Bulacu, Schomaker, and Brink (2007), has been maintained and considered in this paper. Therefore, for the testing phase the IAM dataset comprises of a total of 1314 handwritten samples with two samples per writer.

On the other hand, the training process consists of the third and the fourth samples from the 127 writers who provided more than four samples. The data has been used for testing process is gathered from the first and second samples of all the 657 writers. In addition, care was taken in order that the training part and the testing part of the data are separated. Fig. 2 illustrates some samples from IAM dataset.

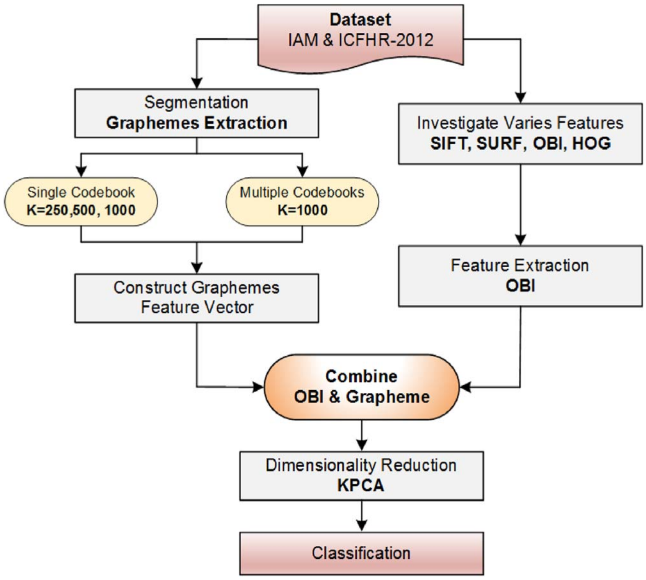

Fig. 1. The system overall diagram flow.

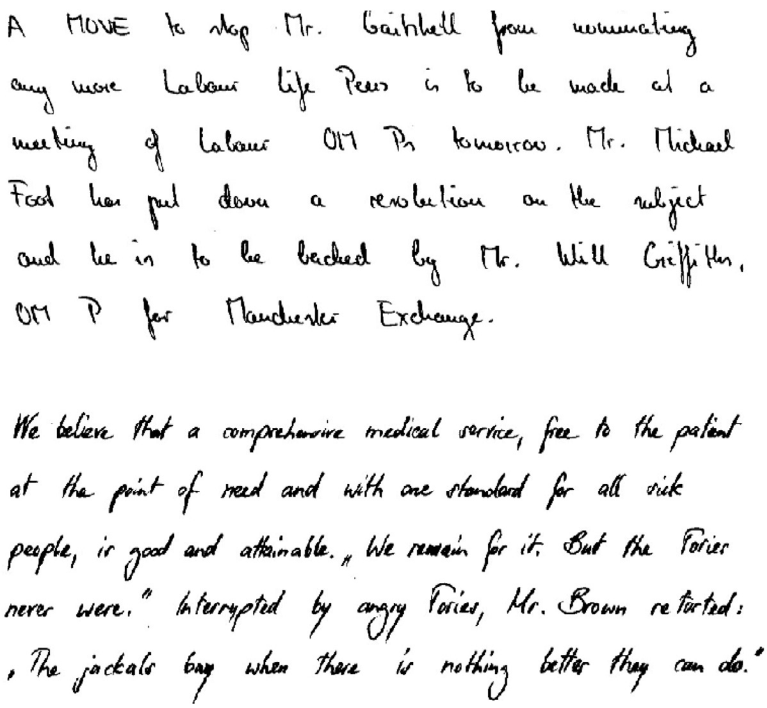

Fig. 2. Two samples from IAM dataset.

3.1.2. ICFHR-2012 dataset

ICFHR 2012 dataset for Arabic language is a large dataset (Hassane & Al-Madeed, 2012). The documents have been digitized and saved as grey scale PNG images having various text contents. In this dataset, more than two hundred writers were asked to write three different content Arabic paragraphs. The pages have been divided into a training set and a testing set. The first two paragraphs from each page/writer are segmented as training set while the third paragraph is stored in the testing set. Fig. 3 illustrates some samples taken from ICFHR 2012 dataset.

3.2. OBI feature extraction

Various feature extraction methods have investigated as discussed previously and the findings have shown that OBI method

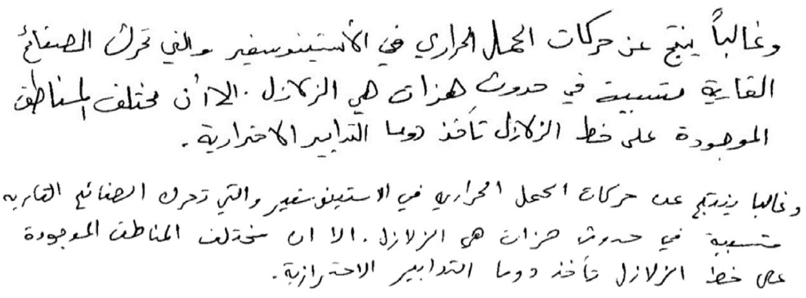

Fig. 3. Two samples from ICFHR 2012 dataset.

outperforms others thus method has been adopted in this paper. Fig. 1 shows the structure of our method. We start by describing the OBI method since it is an important component of the algorithm. Basic Image Features (BIFs) consist of texture based patterns encoded as images as follows. In an image each location is classified into one of seven types using an approximate local symmetry type as described in Newell and Griffin (2014) and Crosier and Griffin, 2008). The local symmetry types are flat, dark and light rotational, dark and light line, slop and saddle-like patterns. To classify the patterns, a bank of six derivative-of-Gaussian filters are used: one 0th order filter, two 1st order filters and three 2nd order filters. To compute the BIFs two tuneable parameters are necessary: a filter scale parameter σ and a threshold value ϵ. These will effectively classify the locality as flat, or as one of the other symmetry types. For example, by increasing the value of ϵ to a high value, the image will be classified as flat (Newell & Griffin, 2011).

A modification and extension to the BIF algorithm has been proposed through a combination of local orientation with local symmetry type resulting in the generation of oriented BIFSs (oBIFs). In this case, the value of n, which represents the orientation quantisation level, will enable the extraction of the possible orientations values depending on the local symmetry type (Newell & Griffin, 2014). For example, by setting the location to dark or light or a flat type there will be no orientation exhibited. On the other hand, by setting the location to a dark line, light line or saddle-like types will lead to the specification of n possible orientations. Finally, a slope type location will specify 2n orientations since a slope has also a further directional feature. Consequently, this results in a set of $(5n+3)$ features (Newell & Griffin, 2011). The oBIF calculation is described in Fig. 4.

3.3. Codebook extraction

The grapheme codebook method has been shown to be a useful technique in various pattern recognition problems including writer identification. It works by first extracting the graphemes, which can be defined as small pieces or segments of a character. One simple and effective method to extract the graphemes can be done by splitting the connected components of the written text. This can be carried out by using a suitable algorithm such as the ink-trace minima heuristic method shown in Fig. 5 (Siddiqi &Vincent, 2010).

The generated graphemes of an image would appear as an unordered bag of patterns and will be used to extract the codebook which will act as a reference set of graphemes. This descriptor will be used to determine a ‘shape alphabet’ with which to describe each image. There exist a number of codebook generation methods in the literature based on various criteria (Abdi & Khemakhem, 2015; Bulacu & Schomaker, 2007; Ghiasi & Safabakhsh, 2013). Various codebook selection methods can be used extract codebooks depending on the application at hand.

In this paper, we proposed to generate a codebook of the dataset to efficiently represent the data being tested so that the shapes to be recognised are closely tuned to the shapes used by the authors of the scripts. To achieve this, a selection approach to collect the graphemes by a shape-based similarity approach using a Kohonen Self-Organising Feature Map (SOFM) proposed in Bulacu and Schomaker (2006) has been used by specifying the number of clusters to be related to the size of the generated codebook. Furthermore, we propose to use the cluster centres for the codebook where each one is chosen as a representative of its cluster of similar graphemes. An extensive training is required by the SOFM on order to ensure convergence to a layout so as to generated the most effective codebook. Once the creation of the codebook is done, a feature extraction is then required.

As mentioned above, the proposed approach presented in this paper can be considered as a further development of the previous work of Khalifa et al. (2015). For a fair comparative study, the codebook generation steps of their work were followed. Therefore, one of the essential steps is to measure and investigate the effect of combining multiple codebook features on the identification performance rates. The representation of a multiple codebook can be defined as:

$$\mathrm{Y} = \sum_{j=1}^{n} P_j \tag{1}$$

where:

Y: the number of graphemes extracted from the whole training set.
n: the number of the partitions of the graphemes.
$\boldsymbol{P_j}$: one partition of the graphemes.

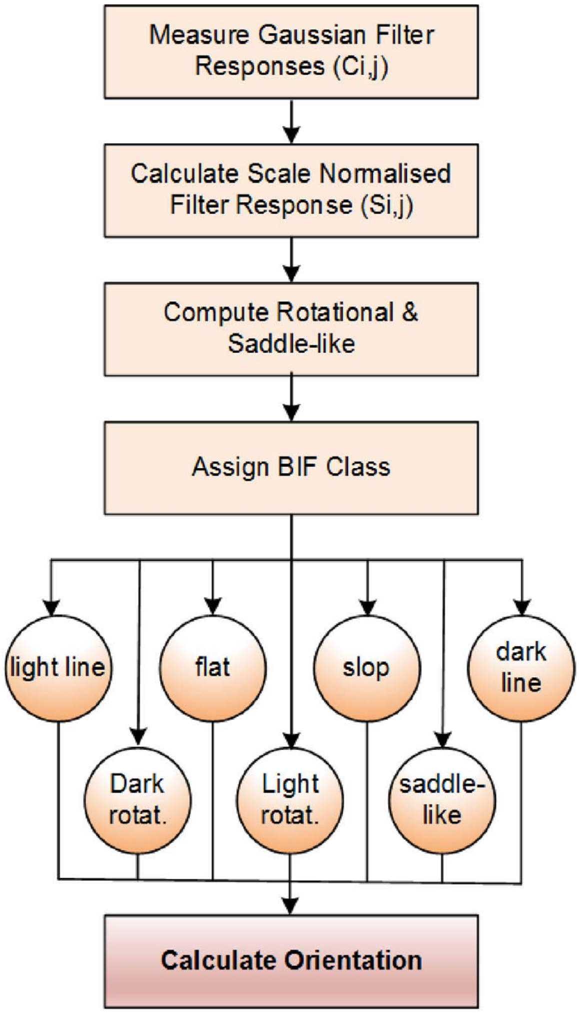

Fig. 4. Method to calculate oBIF.

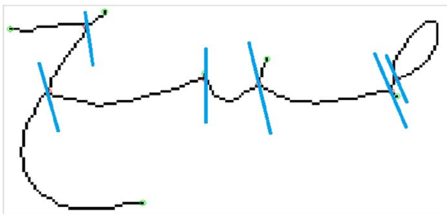

Fig. 5. Grapheme splitting points.

From Eq. (1), the grapheme features are divided into $\boldsymbol{n}$ partitions $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, $\boldsymbol{P}_3$…… $\boldsymbol{P}_n$ each of size $\boldsymbol{W}$. A popular tool of creating a codebook can be achieved by using k-means clustering, which is an unsupervised learning algorithm (Ghiasi & Safabakhsh, 2013; Khalifa et al., 2015). Therefore, the features of each grapheme partitions $\boldsymbol{P_j}$ have been clustered using k-means clustering algorithm. Once done, one needs to find the centres $\boldsymbol{C_k}$ of each cluster so that each writer sample of the training dataset can be assigned into a codebook (cluster) of size $\boldsymbol{Q_k}$, which can be done by mapping the grapheme features of the sample image to the nearest centre $\boldsymbol{C_k}$. In this work, at the first stage, the proposed approach investigates the identification performances based on single codebook grapheme. In the second stage, the effect of combining multiple codebook grapheme features is investigated and the findings discussed.

3.4. Combining OBI and grapheme features

The main idea of the proposed identification system relates to the fusion of OBIs and Grapheme features resulting in a large feature vector. Therefore, a reduction of the resulting high dimensionality vector is crucial in order to select those features with high discriminative power while at the same time speeding up the recognition process. The following presents the proposed approach used to address this issue.

3.5. Dimensionality reduction

Dimensionality reduction in a process to extract the most discriminative from a high-dimensional data set. The concept is to compact the raw data into a more condensed form so as to reduce both the high dimensionality of the feature vector and the computational complexity while still keeping intact the accuracy of the recognition (Arunasakthi & Priya, 2014). However, the performance will be affected if a noisy or faulty input data are considered since the removal of redundant data should not degrade the performances. This process can be divided into feature selection and feature extraction. In the first approach one attempts to detect a subset of features of the data using three strategies: filter, wrapper and embedded methods. The main idea of the second approach is to reduce the high dimensional data into a space of fewer dimensions and both linear or nonlinear techniques can be deployed.

Linear dimensionality reduction techniques are useful in many pattern recognition problems as a tool to support the analysis of high dimensional datasets (Cunningham & Ghahramani, 2015). It is to ne noted that linear methods may not be appropriate for use directly for the identification of handwriter since data in non-linear. However, they are simple and can be modified and tuned for nonlinear problems. In this paper, we have adopted such as approach as described in the following. Linear dimensionality reduction methods work by generating a low-dimensional linear data of the original high-dimensional data while maintaining the most discriminative features of the data. Principal Component Analysis (PCA), which is a very popular linear technique for dimensionality reduction, implements a linear mapping of the data to a lower-dimensional space ensuring that variance of the new data in the low-dimensional space is maximized (Arunasakthi & Priya, 2014; Cunningham & Ghahramani, 2015). PCA can be used in a nonlinear approach through the kernel trick. The output method can be employed to construct nonlinear mappings that maximize the variance of the data. The resulting approach, termed kernel PCA (KPCA), operates in a similar fashion as in conventional approach with the main difference being the use of a nonlinear mapping which maps each given data point onto an abstract function. In other words, KPCA technique implements PCA with some extra functionality of the kernel trick (Arunasakthi & Priya, 2014; Prufungsarbeit, 2011; Ross, 2008). At the first step, PCA starts by calculating the covariance matrix of the image matrix as shown by Eq. (2).

$$\mathrm{C} = \frac{1}{\boldsymbol{m}} \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \mathbf{x_i}\, \mathbf{x_i^T} \tag{2}$$

KPCA starts by calculating the covariance matrix of the data after being converted into a higher-dimensional space,

$$\mathrm{C} = \frac{1}{\boldsymbol{m}} \sum_{\boldsymbol{i}=1}^{\boldsymbol{m}} \Phi(\mathbf{x_i})\, \Phi(\mathbf{x_i})^{\mathbf{T}} \tag{3}$$

It then maps the converted data from the previous step onto the first **k** eigenvectors. It utilizes the kernel trick to factor away much of the computation, such that the whole procedure can be executed without evaluating $\Phi(\mathrm{x})$. Obviously, Φ has to be selected such that it has a known corresponding kernel.

3.6. Classification

The classification, which is the final stage of the system, uses K-nearest neighbour (k-NN) classifier. This classifier has been utilized in many pattern classification problems and is very useful for measuring the distances between the test data and each of the training data in order to determine the final classification result (Gilliam, 2011). Moreover, this algorithm is a simple yet effective classifier because it can use different distance measures such as Euclidean distance, Chi-square distances and Manhattan distance. In this work, we have investigated these methods and the results obtained are as follows: Manhattan distance = 80%, Euclidian distance = 96.05% and Chi-square distance = 73%. This has allowed us to adopt the Euclidian distance in our work since it outperforms the other metrics. For experiments in the statistical approach, we have used 650 writers from the IAM dataset. In our combined model, matching is carried out using Eq. (4) as follows:

$$\boldsymbol{E_i} = \sqrt{\sum_{\boldsymbol{j}=1}^{\boldsymbol{k}} (\boldsymbol{M_{ij}} - \boldsymbol{V_i})^2} \tag{4}$$

where:

E_i: final distance between the input sample and model$i$
k: is the number of features in the features vector
M_{ij}: is the jth feature of model$i$,
V_j: is the jth feature of the input sample feature vector.

Table 1
Comparison of system's performance with previous work (Khalifa et al., 2015).

Codebook size	Top-1		Top-5		Top-10	
	Our work	previous work	Our work	previous work	Our work	previous work
250	87.56%	80.00%	90.13%	88.00%	93.34%	93.00%
500	87.96%	80.00%	91.34%	89.00%	94.28%	94.00%
1000	88.01%	81.00%	94.16%	89.00%	96.45%	94.00%

4. Experimental results

To evaluate the performances of the proposed approach, a set of experiments have been conducted. As mentioned above, the experimentation has been carried out using the IAM dataset for English handwriting and ICFHR-2012 dataset for Arabic handwriting. The codebook is generated and used in the system for both English and Arabic datasets. The evaluation was carried out first using a single codebook followed by a multiple codebook method. The results obtained were compared against the results reported in a similar approach (Khalifa et al., 2015).

4.1. Single codebook

In this experiment, KPCA has been used as a nonlinear dimensionality reduction technique to produce a low dimensional data in order to overcome the over-fitting problem and save the system resources while speeding up the execution time. This experiment was conducted on the IAM English dataset. To provide similar evaluation conditions in Khalifa et al. (2015), their experimental steps have been followed for the sake of a fair analysis. Initially, 188k graphemes were generated from the handwriting training set of 127 writers. Then, a codebook of size 250 using k-means clustering ($k = 250$) was created. For each input sample from the testing set, the writer descriptor for that sample was extracted from the generated codebook. The handwritten target sample is then compared against 1313 other samples. The classification process using a 1-nearest neighbour classifier (using Euclidian distance) is used to evaluate the writer identification performance. Table 1 depicts the performance results obtained under different codebook sizes ranging from 250 up to 1000.

As shown in Table 1, with a codebook size of (250-500- 1000) using k-means clustering ($k = 250$, 500 and 1000), one can notice that the best result with a recognition rate of 88.01%, which is achieved when the codebook size is 1000. On the other hand, this result is better than the result of 81% with a codebook of 1000 as in Khalifa et al. (2015). The results obtained are shown in Fig. 6 and clearly demonstrate that our results are attractive in the case of a single codebook.

4.2. Multiple codebooks

In the second part of the experiment, we have investigated the effect of combining multiple codebook features on the identification performance rate of the system. In the first step, the total graphemes size extracted from the whole training data set (English and Arabic) is split randomly into $\boldsymbol{n}$ partitions (as shown above by Eq. (1)) without an overlapping of the graphemes (where $n=$ 3,4,6,8,10,12,14,16). Therefore, $\boldsymbol{Y}$ multiple reference codebooks are generated from $\boldsymbol{P}$ for use to investigate and determine the effect of $\boldsymbol{Y}$ codebooks on the system performance. The total size of the graphemes has been randomly partitioned into $\boldsymbol{n}$ parts. Likewise, the experiments have used both English and Arabic datasets and the results will be illustrated and discussed in the next sections.

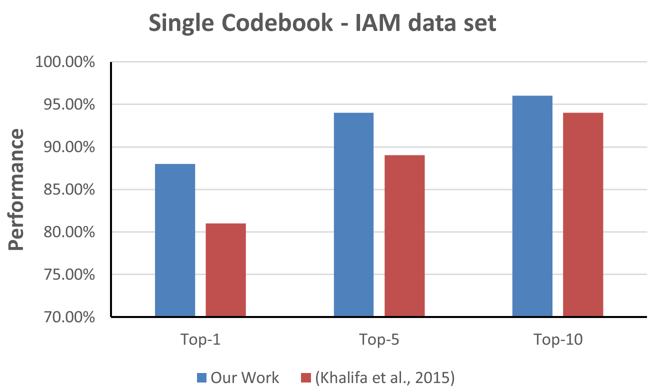

Fig. 6. System performance for codebook size = 1000.

Table 2
System performance (Top-1) versus execution time.

n	Top-1 Our work		Top-1 Previous work (Khalifa et al., 2015)	
	Performance (Based on KPCA)	Execution time (s)	Performance (Based on KDA)	Execution time (s)
3	86.34%	1.13	84.00%	10.64
4	87.46%	1.16	85.00%	13.59
6	89.34%	1.25	87.00%	19.24
8	89.95%	1.35	89.00%	25.23
10	92.03%	1.48	90.00%	31.5
12	89.57%	1.60	92.00%	36.9
14	90.25%	1.72	90.00%	42.5
16	90.41%	1.79	88.00%	48.6

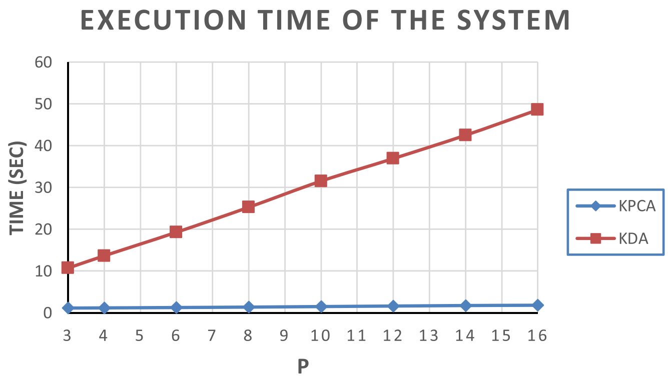

Fig. 7. Execution time for KPCA vs KDA (English dataset).

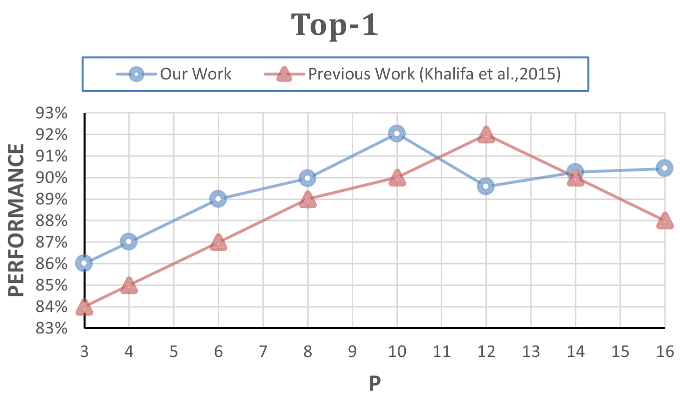

Fig. 8. Comparison of the performance for Top-1 identification.

Table 3
System performance for Top-5 and Top-10 identification versus to the performance in (Khalifa et al., 2015).

n	Top-5		Top-10	
	Our work (Based on KPCA)	Previous work* (Based on KDA)	Our work (Based on KPCA)	Previous work* (Based on KDA)
3	93.90%	89.00%	94.90%	92.00%
4	94.06%	89.00%	95.43%	93.00%
6	95.19%	91.00%	96.42%	94.00%
8	95.48%	92.00%	96.67%	95.00%
10	95.28%	92.00%	96.57%	96.00%
12	95.28%	93.00%	97.34%	97.00%
14	95.13%	91.00%	96.27%	95.00%
16	94.90%	91.00%	95.89%	94.00%

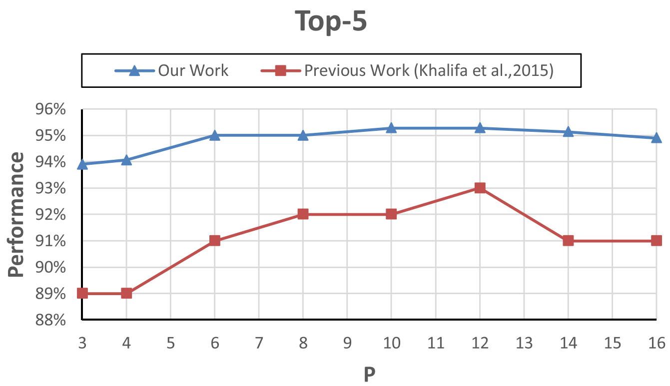

Fig. 9. Comparison of the performance for Top-5 (English dataset).

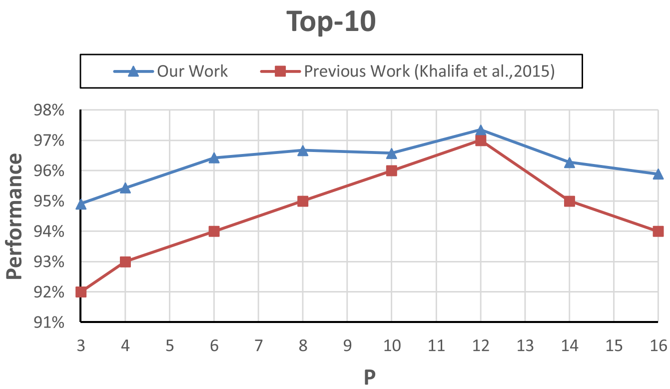

Fig. 10. Comparison of the performance for Top-10 (English dataset).

Table 4
System performance (Top-1) versus execution time for Arabic dataset.

n	Top-1 Our work		Top-1 previous work (Khalifa et al., 2015)	
	Performance (Based on KPCA)	Execution time (s)	Performance (Based on KDA)	Execution time (s)
3	96.49%	1.15	88.00%	11.64
4	96.49%	1.18	89.00%	15.59
6	96.49%	1.29	90.00%	19.24
8	96.49%	1.37	92.00%	26.23
10	96.05%	1.49	93.00%	33.5
12	96.05%	1.62	95.00%	38.9
14	96.05%	1.77	92.00%	45.5
16	96.05%	1.80	91.00%	49.6

4.2.1. Evaluation of the performance using english dataset

The system performance for Top-1 identification is assessed and compared against the results reported in Khalifa et al. (2015) including the computational complexity. Table 2 depicts the results obtained when n is varied. As shown in Table 2, the average execution time of our proposed system is around 1.48 s compared to 31.5 s in the work of Khalifa et al. (2015) On the other hand, the maximum value of the system performance in their work was 92% compared to 90.41% in our work. Moreover, if we consider the execution time, one can conclude that the use of KPCA is a more effective reduction technique in terms of the execution time factor. This experiment proved that the identification system's performance as well as the execution time have been improved compared to the work of Khalifa et al. (2015).

Fig. 7 illustrates the execution time of our system based on KPCA technique versus KDA technique given in Khalifa et al. (2015). From the results obtained, one can notice that the KPCA technique is capable to improve the execution time of the system compared to KDA. Fig. 8 shows a comparison of the performance for Top-1 between our system and the work in Khalifa et al. (2015).

The results depicted in Table 3 present the performance of our developed system for Top-5 and Top-10 identification. In the case of Top-5 identification, it can be observed from the table that when n = 10 and 12, a performance of 95.28% is obtained demonstrating that the proposed technique outperforms the method proposed by Khalifa et al. (2015). In addition, in the case of Top-10, the

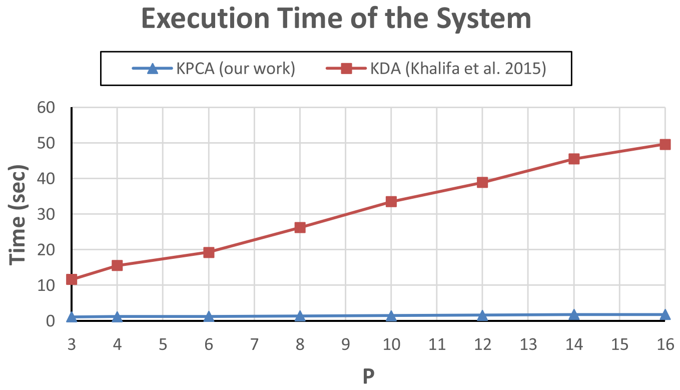

Fig. 11. Execution time for KPCA vs KDA (Arabic dataset).

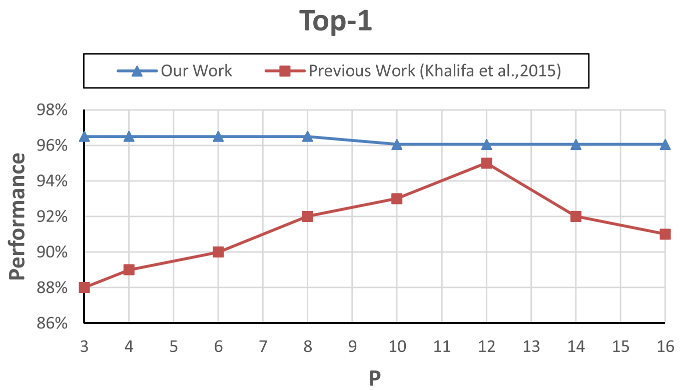

Fig. 12. Comparison of the performance for Top-1 (Arabic dataset).

Table 5
System performance for Top-5 and Top-10 for Arabic dataset versus to the performance in (Khalifa et al., 2015).

p	Top-5		Top-10	
	Our work (Based on KPCA)	Previous work (Based on KDA)	Our work (Based on KPCA)	Previous work (Based on KDA)
3	98.36%	90.00%	98.60%	92.00%
4	98.45%	90.00%	98.60%	92.00%
6	98.25%	91.00%	98.60%	94.00%
8	97.50%	93.00%	98.60%	95.00%
10	97.50%	94.00%	97.90%	96.00%
12	97.50%	96.00%	97.90%	97.00%
14	97.50%	93.00%	97.90%	96.00%
16	97.50%	93.00%	97.90%	96.00%

performance at $n = 12$ reaches 97.34% compared against 97% in Khalifa et al. (2015). Although the results are marginally better and the speed performance is significantly better as shown in Figs. 9 and 10.

4.2.2. Evaluation of the performance using arabic dataset

In this part of the experiment, the system performance is evaluated and compared against the results given in Khalifa et al. (2015) using the Arabic dataset. The results obtained indicate that the proposed method demonstrates an improvement of the identification rates when compared against the results of Khalifa et al. (2015). Table 4 illustrates the identification results for Top-1 and showing that the maximum identification performance is 96.49% against 95.00%.

Fig. 11 illustrates the execution time of our system using KPCA technique versus KDA technique used in Khalifa et al. (2015) using the Arabic dataset. From the figure, it can be observed that KPCA technique has improved the execution time of the system compared to KDA. Fig. 12 shows a comparative analysis of the identification performance for Top-1 of the two methods.

Table 5 depicts the system performance in the cases of Top-5 and Top-10 identification rates. In this case, the proposed technique outperforms Khalifa's technique in all cases. The maximum performance of 98.6% is obtained for Top-10 identification compared with 97% for method proposed by Khalifa et al. (2015). Therefore, these results clearly show again that our proposed technique yields

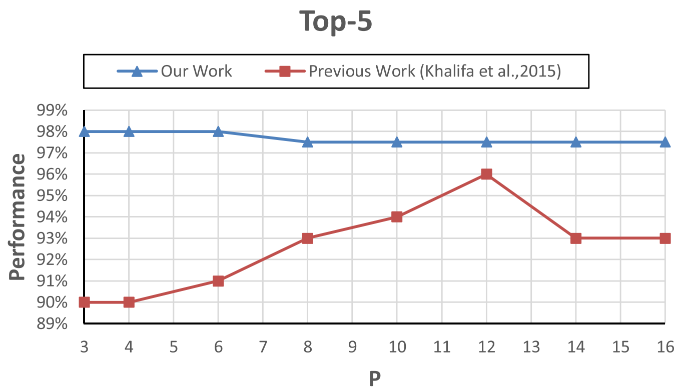

Fig. 13. Comparison of the performance for Top-5 (Arabic Dataset).

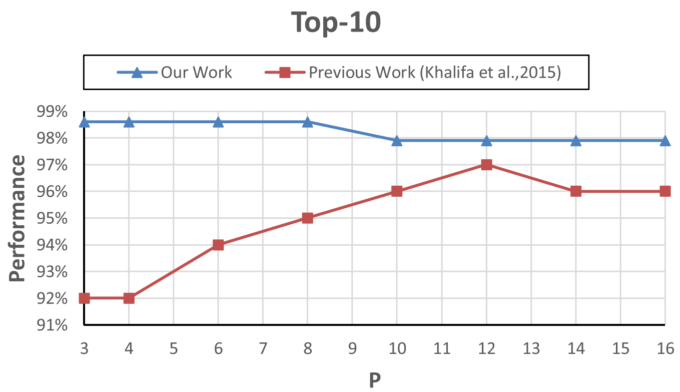

Fig. 14. Comparison of the performance for Top-10 (Arabic dataset).

Table 6
Comparison of the performance of our proposed approach against similar ones.

Approach developed by	Dataset	No. of writers	(Top1) Performance %
(Bulacu et al., 2007)	FIREMAKER	250	83
(Bensefia, Paquet, & Heutte, 2005)	IAM	150	86
(Bulacu et al., 2007)	IAM	650	89
(Siddiqi et al., 2010)	IAM	650	91
(Khalifa et al., 2015)	IAM	650	92
(Khalifa et al., 2015,)	ICFHR2012	206	95
Our approach	**IAM**	**650**	**92**
Our approach	**ICFHR2012**	**206**	**97**

improved recognition performances. The results of this analysis are illustrated in Figs. 13 and 14.

Finally, Table 6 shows a comparison of the identification rates between some of the previous works in the literature. It is to be noted that two issues were encountered to perform this comparison: some of the works have used different datasets, which are not available in some cases. Moreover, in the cases where the IAM dataset was used, the number of writers used in the cited works were different; thus a comparative analysis may not be fair. Therefore, for the sake of a fair comparison, our results have used 650 writers and are compared against other works that have also used the same number of authors. As illustrated in Table 6, the results obtained by our proposed method clearly outperform the other listed works that have been developed their systems based on the same numbers of writers.

5. Conclusion

This paper has presented a novel writer identification approach using the concept of Oriented Basic Image feature extraction and its combination with the graphemes codebook method. The proposed algorithm has resulted in an improved identification performance when compared against similar techniques. In addition, the use of KPCA, which is a nonlinear dimensionality reduction technique, has resulted in a reduction of the computational complexity. Further improvement of the identification performance can be achieved by using deep learning (convolutional neural networks) concept. However, the technique is computationally intensive so an implementation using Graphics Processing Unit (GPU) is currently being investigated.

Acknowledgments

This work is supported by the Qatar National Research Fund through National Priority Research Program(NPRP) No 7-442-1-082. The contents of this publication are solely the responsibility of the authors and do not necessarily represent the official views of the Qatar National Research Fund or Qatar University.

References

Abdi, M., & Khemakhem, M. (2015). A model-based approach to offline text-independent Arabic writer identification and verification. *Pattern Recognition, 48*(5), 1890–1903.

Abdi, M., & Khemakhem, M. (2012). Arabic writer identification and verification using template matching analysis of texture. *Proceedings of the IEEE 12th international conference on computer and information technology*.

Al-Maadeed, S. (2012). Text-dependent writer identification for arabic handwriting. *Hindawi Publishing Corporation - Journal of Electrical and Computer Engineering, 2012* Article ID 794106.

Arunasakthi, K., & Priya, L. (2014). A Review on linear and nonlinear dimensionality reduction techniques seven techniques for dimensionality reduction. *An international journal of machine learning and applications (MLAIJ). 1*.

Awaida, S., & Mahmoud, S. (2012). State of the art in off-line writer identification of handwritten text and survey of writer identification of Arabic text. *Educational Research and Reviews, 7*(20), 445–463.

Awaida, S., & Mahmoud, S. (2013). Writer identification of arabic text using statistical and structural features. *Journal Cybernetics and Systems, 44*(1), 57–76.

Bensefia, A., Paquet, T., & Heutte, L. (2005). A writer identification and verification system. *Pattern Recognition Letters, 26*(13), 2080–2092.

Bulacu, M., Schomaker, L., & Vuurpijl, L. (2003). Writer identification using edge-based directional features. *Proceedings of the seventh international conference on document analysis and recognition (ICDAR). 2. Proceedings of the seventh international conference on document analysis and recognition (ICDAR)* (pp. 937–941).

Bulacu, M., & Schomaker, L. (2007). Text-independent writer identification and verification using textural and allographic features. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 29*, 701–717.

Bulacu, M., & Schomaker, L. (2006). Combining multiple features for text-independent writer identification and verification. *Proceedings of the 10th international workshop on frontiers in handwriting recognition* (pp. 281–286). .

Bulacu, M., Schomaker, L., & Brink, A. (2007). Text-independent writer identification and verification on offline Arabic handwriting. *Proceedings of the ninth international conference on document analysis and recognition* (pp. 769–773). .

Crosier, M., & Griffin, L. (2008). Texture classification with a dictionary of basic image features. *Computer vision and pattern recognition. CVPR 2008. IEEE Conference on*.

Cunningham, J., & Ghahramani, Z. (2015). Linear dimensionality reduction: Survey, insights, and generalizations. *Journal of Machine Learning Research,* (16), 2859–2900.

Daniels, Z. A., & Bairs, H. S. (2013). Discriminating features for writer identification. *Proceedings of 12th international conference on document analysis and recognition* (pp. 1385–1389). .

Djeddi, C., Meslati, L., Siddiqi, I., Ennaji, A., El Abed, H., & Gattal, A. (2014). Evaluation of texture features for offline arabic writer identification. *Proceeding of the 11thIAPR international workshop on document analysis systems* (pp. 106–110). .

Fiel, S., & Sablatnig, R. (2012). Writer retrieval and writer identification using local features. *Proceedings of 10th IAPR international workshop on document analysis systems DAS* (pp. 145–149). .

Ghiasi, G., & Safabakhsh, R. (2013). Offline text-independent writer identification using codebook and efficient code extraction methods. *Proceedings of image and vision computing. 31. Proceedings of image and vision computing* (pp. 379–391).

Gilliam, T. (2011). *Writer identification in medieval and modern handwriting, chapter-6* p. 133.

Hannad, Y., Siddiqib, I., & El-Kettan, M. (2016). Writer identification using texture descriptors of handwritten fragments. *Expert Systems with Applications, 47*, 14–22.

Hassane, A., & Al-Madeed, S. (2012). ICFHR 2012 competition on writer identification challenge 2: Arabic scripts. *Proceeding of the international conference on frontiers in handwriting recognition (ICFHR), Bari-Italy. 835–. Proceeding of the international conference on frontiers in handwriting recognition (ICFHR), Bari-Italy* (pp. 840–). .

Kam, M., Fielding, G., & Conn, R. (1997). Writer identification by professional document examiners. *Journal of Forensic Sciences, 1997*(42), 778–785.

Khalifa, E., Al-maadeed, S., Tahir, M., & Bouridane, A. (2015). Off-line writer identification using an ensemble of grapheme codebook features. *Pattern Recognition Letters, 59*, 18–25.

Marti, U., & Bunke, H. (2002). The IAM- database: An english sentence database for off line hand writing recognition. *International Journal on Documents Analysis Recognition, 5*, 39–46.

Moubtahij, H., Halli, A., & Satori, K. (2014). Review of feature extraction techniques for offline handwriting Arabic text recognition. *International Journal of Advances in Engineering & Technology, 7*(1), 50–58.

Newell, A., & Griffin, L. (2014). Writer identification using oriented basic image features and the delta encoding. *Pattern Recognition, 47*(6), 2255–2265.

Newell, A., & Griffin, L. (2011). Natural image character recognition using oriented basic image features. *Digital image computing techniques and applications (DICTA), international conference on*.

Paraskevas, D., Stefanos, G., & Ergina, K. (2014). Writer identification using a statistical and model based approach. *Proceeding of the 14th international conference on frontiers in handwriting recognition (ICFHR)*.

Prufungsarbeit, S. (2011). *GPU assisted implementation of kernel-based template attacks*MSC thesis submitted to. Germany: The University of Bochum.

Raj, A., & Chaudhary, A. (2016). A review of personal identification using handwriting. *Proceedings of international journal of engineering science and computing. 6. Proceedings of international journal of engineering science and computing* (pp. 3454–3457).

Ross, I. (2008). *Nonlinear dimensionality reduction methods in climate data analysis*PhD thesis submitted to. The University of Bristol.

Schlapbach, A. (2007). A writer identification and verification system using HMM based recognizers. *Proceedings of journal of pattern analysis & applications archive. 10. Proceedings of journal of pattern analysis & applications archive* (pp. 33–43).

Siddiqi, I., & Vincent, N. (2010). Text independent writer recognition using redundant writing patterns with contour-based orientation and curvature features. *Pattern Recognition, 43*(11), 3853–3865.

Sreeraj, M., & Idicula, S. (2011). A survey on writer identification schemes. *International Journal of Computer Applications, 26*(2), 23–33.

Tan, G., Viard-Gaudin, C., & Kot, A. (2009). Automatic writer identification framework for online handwritten documents using character prototypes. *Pattern Recognition Journal, 42*, 3313–3323.

Tang, Y., Wu, X., & Bu, W. (2013). Offline text-independent writer identification using stroke fragment and contour based features. *Proceedings of 2013 international conference on biometrics* (pp. 1–6). .

Vasquez, J. L., Travieso, C. M., & Alonso, J. B. (2013). Off line writer identification based on graphometric parameters. *Proceedings of the ninth international conference on document analysis and recognition* (pp. 219–223). .

Wu, X., Tang, Y., & Bu, W. (2014). Offline text-independent writer identification based on scale invariant feature transform. *Proceeding of IEEE Trans. on Information Forensics and Security. 9*.

Yang, W., Jin, L., & Liu, M. (2016). Deep writer ID: An end-to-end online text-independent writer identification system. *IEEE Intelligent Systems, 31*(2).