An Arabic Script Recognition System

Yasser M. Alginahi[1,2*], Mohammed Mudassar[3], Muhammad Nomani Kabir[4]
[1]IT Research Center for the Holy Quran and its Sciences (NOOR)
[2]Dept. of Computer Science, Deanship of Academic Services
Taibah University, P.O. Box 344, Postal code: 41411, Madinah, Saudi Arabia
[email]: yginahi@taibahu.edu.sa
[3]Dept. of Computer Science, College of Computer Science and Engineering
Taibah University, P.O. Box 344, Postal code: 41411, Madinah, Saudi Arabia
[email]: mpinjar@taibahu.edu.sa
[4]Dept. of Computer Systems, University of Pahang, 26300 Gambang, Kuantan, Pahang Darul Makmur, Malaysia
[email]: nomanikabir@ump.edu.my
*Corresponding Author: Yasser M. Alginahi

Received December 31, 2014; revised June 4, 2015; accepted June 29, 2015; published September 30, 2015

Abstract

A system for the recognition of machine printed Arabic script is proposed. The Arabic script is shared by three languages i.e., Arabic, Urdu and Farsi. The three languages have a descent amount of vocabulary in common, thus compounding the problems for identification. Therefore, in an ideal scenario not only the script has to be differentiated from other scripts but also the language of the script has to be recognized. The recognition process involves the segregation of Arabic scripted documents from Latin, Han and other scripted documents using horizontal and vertical projection profiles, and the identification of the language. Identification mainly involves extracting connected components, which are subjected to Principle Component Analysis (PCA) transformation for extracting uncorrelated features. Later the traditional K-Nearest Neighbours (KNN) algorithm is used for recognition. Experiments were carried out by varying the number of principal components and connected components to be extracted per document to find a combination of both that would give the optimal accuracy. An accuracy of 100% is achieved for connected components >=18 and Principal components equals to 15. This proposed system would play a vital role in automatic archiving of multilingual documents and the selection of the appropriate Arabic script in multi lingual Optical Character Recognition (OCR) systems.

Keywords: Arabic, Script Recognition, KNN, PCA

http://dx.doi.org/10.3837/tiis.2015.09.023 **ISSN : 1976-7277**

1. Introduction

The research on script identification started in the early 1990s; however, the work on the identification of languages using cursive scripts, such as Arabic, started towards the end of the 20th century. The research work on script identification emanates from document image analysis of multi script document images to identify the different languages script which may exist in such documents. Identifying the script of the language makes it easy to deal with character recognition if the script language is known prior to passing the document into any character recognition system. There is very little research reported that considers the identification/recognition of the Arabic script from multilingual documents.

The Arabic alphabet is the Arabic script as it is codified for writing the Arabic language. "The Arabic script is a writing system used for writing several languages of Asia and Africa, such as Arabic, Persian, and Urdu. After the Latin script, it is the second-most widely used writing system in the world [1]". Some of the main languages which are currently written with the Arabic alphabet are: Persian, Urdu, Pashtun, Kurdish, Sindhi, Uyghur, Kashmiri etc. Arabic language script is written from right to left, in a cursive style, and includes 28 letters [1 – 2]. The Urdu language contains 38 alphabet derived from the Persian (Persian) and the Arabic alphabet, where the majority of the letters are borrowed from Arabic while only four alphabets are primarily borrowed from Persian. Very seldom, letters may be borrowed from Sanskrit. Even though the letters actually are inherited from these languages, the names used for these letters differ in some cases [3]. Persian is written from right to left. Its alphabet consists of 32 letters which are used in constructing and writing words following certain rules which are used to write the majority of the Persian words. Though there are a few exceptions [4]. There are many common words in Arabic, Persian and Urdu. There are some letters which are unique to Arabic and others unique to Persian and/or Urdu ,in general if a person speaks Arabic he/she can read the Persian and Urdu script with some difficulties, however, without understanding the meaning and vice versa. Therefore, due to the high similarities and complexity of the Arabic script the problem of identifying the languages written in Arabic script is not that easy to solve.

The objective of this work is to develop an Arabic script recognition system to identify documents in the following languages: Arabic, Persian and Urdu from a set of multi-lingual documents. This research will first identify/recognize documents of Arabic script. The aim is to segregate documents with Arabic scripts from documents with other scripts then identify the specific Arabic script language of each document. A Script recognition system goes through many steps these are: document image digitization, pre-processing, feature extraction and script classification. Following the process of separating Arabic script documents from other scripts, the documents containing Arabic scripts go through the proposed Arabic script recognition system whose output decides the specific language available in the document.

The current work proposes an Arabic script recognition system, the first of its kind for Arabic script, which has two sub-systems, one for identifying the Arabic script and the second for identifying the languages in the script. The main challenge in realizing the proposed system was the identification of Arabic scripted languages (Arabic, Farsi, and Urdu). This challenge was addressed very well at document level by adapting PCA-KNN model that relies on the extraction of 8-connected components from the documents. A through experimentation was done to understand the important parameters affecting the model recognition accuracy. It was found that the number of 8-connected components is a very critical factor in determining the accuracy of identification, since considerable number of 8-connected components had to be extracted for achieving good accuracy. This gives a direction for a more challenging future research to identify the Arabic scripted languages at line/word level where very few 8-connected components would be available. Further, a different strategy for performing a multi class classification of Arabic scripted language is proposed than the conventional method which suggests the use of One-Vs-One (OVO) or One-Vs-All (OVA) classifiers. The proposed strategy recommends a two-stage process. The first stage would identify the Arabic scripted languages using a PCA-KNN model trained on all the three languages of the Arabic script. The optional second stage would be employed in case of ties where the identification would be done by an ensemble of OVO classifiers. This strategy has the benefit of leveraging the capabilities of the ensemble of OVO classifiers without increasing the computational burden of identification, since the ensemble is used only for resolving the ties.

This paper is organized as follows: section 2 provides the literature survey, section 3 explains the methodology of the proposed script recognition system, section 4 presents the experimental results and observations, and finally section 5 concludes this paper.

2. Literature Survey

This literature survey section includes the methods developed for script recognition systems with concentration on research methods, which include Arabic scripts. After studying the techniques found in literature two survey papers were found providing excellent guidance in this area [5-6]. In the last two decades, the research developed on Arabic scripts is in the tens compared to other scripts such as Latin or Indian.

The work of Spitz [7], described a page-level method for discriminating Han based (Chinese, Japanese, Korean) and Latin based (includes both European and non-European) scripts using the spatial relationships of features related to character structures. In his paper, Spitz, stated that the "Development of techniques to recognize highly connected languages has not been initiated. Handling of Arabic, for example, depends not only on handling connectedness, but also on independence from the effects of the horizontal elasticity, called Khasheda, found in Arabic print [7]." Therefore, it was not until after 1997 when the work on highly connected languages started.

In [8], Pal et al. developed a script line identification system in printed multi-script documents which can identify 12 Indian scripts including Latin and Urdu scripts. The method's classification is based on headlines, horizontal projection profile, water

reservoir-based features, left and right profiles and features based on jump discontinuity features. This method provided a 97.52% average recognition rate.

Kanoun et al. in [9] developed different methods for the differentiation of Latin and Arabic scripts in the scope of text block, text line and connected component levels. The text block differentiation method for Arabic and Latin text is based on morphological analysis and the text line and connected components methods are based on geometrical analysis. The outcomes are quite promising without providing any quantitative results.

In [10] Hochberg et al. developed a method for script identification in printed documents and tested their method on 13 scripts including Latin and Arabic scripts and obtained an average accuracy of 96%. The data used in this work consisted of 268 typeset document images for the 13 scripts tested. In their work, during the training process textual symbols were obtained from documents of known scripts, then they were normalized to a size of 30 x 30 pixels following this, clustering was used to generate template symbols for the script class. The same was repeated for all scripts to create the template databases. In the classification stage, textual symbols extracted from the input document images are compared to the templates stored in the database using the Hamming distance then best average score is chosen as the script of the document.

One of the earliest methods on script identification is the work developed by Wood et al. in [11], in this work, the authors argued that the projection profiles of document images are sufficient to characterize different scripts: Latin, Arabic and Han. However, no results were presented to support their argument.

Namboodiri and Jain [12], proposed an online text-line handwritten script recognition for multi-script languages containing six different languages: Cyrillic, Hebrew, Roman, Arabic, Devnagari and Han scripts. Eleven features were used to achieve an average accuracy rate of 95.5% for complete text lines. The authors employed six different classifiers, as well as, a combination from these classifiers: KNN, Bayes quadratic classifier with mixture of Gaussian densities, decision tree-based classifier and support vector machine classifier.

Busch et al. in [13] tested 7 different kinds of features on multi-script documents written in 8 different scripts these are: Latin, Han, Japanese, Greek, Cyrillic, Hebrew, Devengari, and Persian. The features used were applied separately into the classification stage producing separate recognition rates for each feature used. The features used are: GLCM features, Gabor energy, wavelet energy, wavelet log mean deviation, wavelet log co-occurrence, wavelet log co-occurrence signatures, and wavelet scale co-occurrence signatures. The classification was carried in using a Gaussian Mixture Model (GMM) and the best results were obtained from using the wavelet log co-occurrence features at 99% followed by the wavelet co-occurrence features at 98%; however, the worse results were obtained using the GLCM features.

In [14], Hochberg et al. developed a method of handwritten script identification for distinguishing Arabic, Chinese, Cyrillic, Devnagari, Japanese and Latin. In this method, several features were measured order to obtain the mean skew and standard deviation. Finally, Fisher linear discriminants were used in the classification stage. This method provides an 88% accuracy rate in distinguishing the tested scripts.

In [15] Peake and Tan applied gray-level co-occurrence matrices (GLCM) and multi-channel Gabor filter (16-channel filter with four frequencies at four orientations) to printed multi-script documents written in Korean, Latin, Chinese, Greek, Russian, Persian and Malayalam after applying several pre-processing techniques to provide a document with uniform character spacing. The recognition step was performed using KNN classifier. GCLM produced 77.14% accuracy while the Gabor filter approach produced accuracy rate of 95.71% applied on the whole document.

The most recent work found in literature in this area is the work of Benjelil et al., [16], it proposes a language and script identification system at word level for printed and handwritten Arabic and Latin scripts based on steerable pyramid transform. The features extracted from the pyramid sub bands are used to classify the scripts on only one script among the scripts to identify. This system provided an identification rate of 97.5%.

The work of Lu and Tan identifies the script and languages from degraded and distorted document images [17]. The work of Tan, [18], applied the rotation invariant texture features on six languages (Chinese, English, Greek, Russian, Persian, and Malayalam). The results show that the average classification accuracy of combined features is 96.7%. Therefore, this work demonstrated the potential of such a texture-based approach in script identification. The work in [19] proposed a technique based on stroke density and distribution that uses KNN to detect the script and orientation of document images of different scripts (Arabic, Chinese, Roman, and Hebrew) and provided a recognition rate of 95.63%. The technique is based on the observation that documents of the same script at the same orientation have similar stroke density and distribution.

Finally, the work in the area of Arabic language and script identification has not been given enough attention by researchers in this area. In addition, some recent works have addressed the issue of handwritten script and language identification even though the problem of printed scripts has not been solved yet. In this work, we propose an Arabic scripted language identification system based on PCA to identify the following three languages: Arabic, Persian and Urdu from digital Arabic printed text documents.

3. The Proposed script recognition system

Script recognition is an important area of research that has not been extensively investigated especially in documents containing Arabic script. Script identification/recognition is very essential in countries with multiple ethnic backgrounds, countries having more than one official language, as well as, for security matters. To the best knowledge of the author and from the literature survey conducted there is no work that discriminates between different Arabic scripted languages; thus, the proposed system recommends the usage of monochromatic bitmap images of printed documents with minimal noise containing different scripts such as Latin, Han and Arabic. In this work, documents contain only one language, i.e., the recognition is done at page-level.

The Arabic scripted languages under investigation include: Arabic, Persian and Urdu, **Table 1** shows examples of some text from these languages.

Table 1. Examples of languages using Arabic script

Text	Language
خدا رحم کرنے والا کے نام پر	Urdu
به نامخداوندمهربان	Persian
بِسْمِ ٱللَّهِ ٱلرَّحْمَٰنِ ٱلرَّحِيمِ	Arabic

The main objectives of the proposed Arabic script recognition system are to identify Arabic scripted documents from documents with other scripts and to classify each Arabic scripted document into its corresponding language. **Fig. 1**, shows the two main subsystems of the proposed Arabic scripts recognition system, thus, the purpose of the first subsystem is to identify Arabic scripts from other scripts. This sub-system goes through many steps, as shown in **Fig. 2**, these are: document image digitization, pre-processing, and script classification using horizontal and vertical projection profiles. Digitized image documents go through the pre-processing stage to provide a document image that is ready for further processing. The pre-processing stage includes binarization of gray-level images and noise removal. The horizontal and vertical projection profiles are the main features that help in discriminating Arabic scripts from other scripts. Once the Arabic Scripted Document is identified by the first subsystem the document is fed to the second subsystem which identifies the language of the Arabic Scripted document.

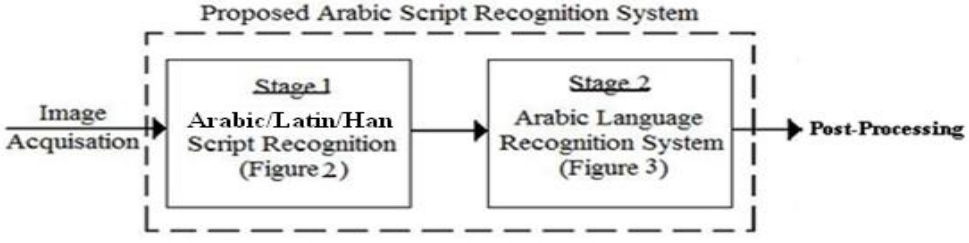

Fig. 1. Block diagram of the proposed Arabic script recognition system

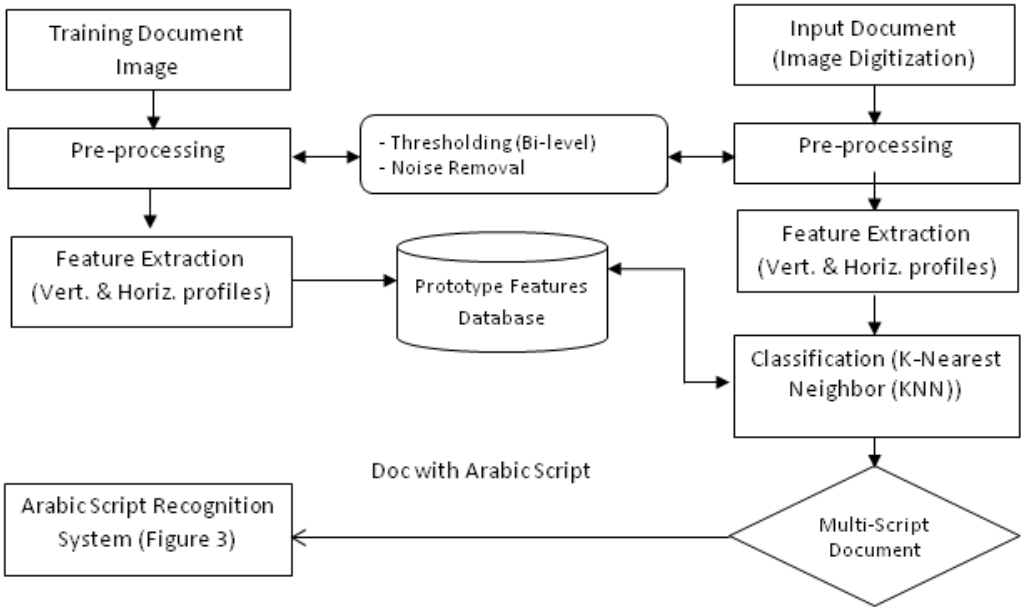

Fig. 2. Multi-Script Recognition System

The second subsystem which is the Arabic script recognition system is shown in **Fig. 3**.

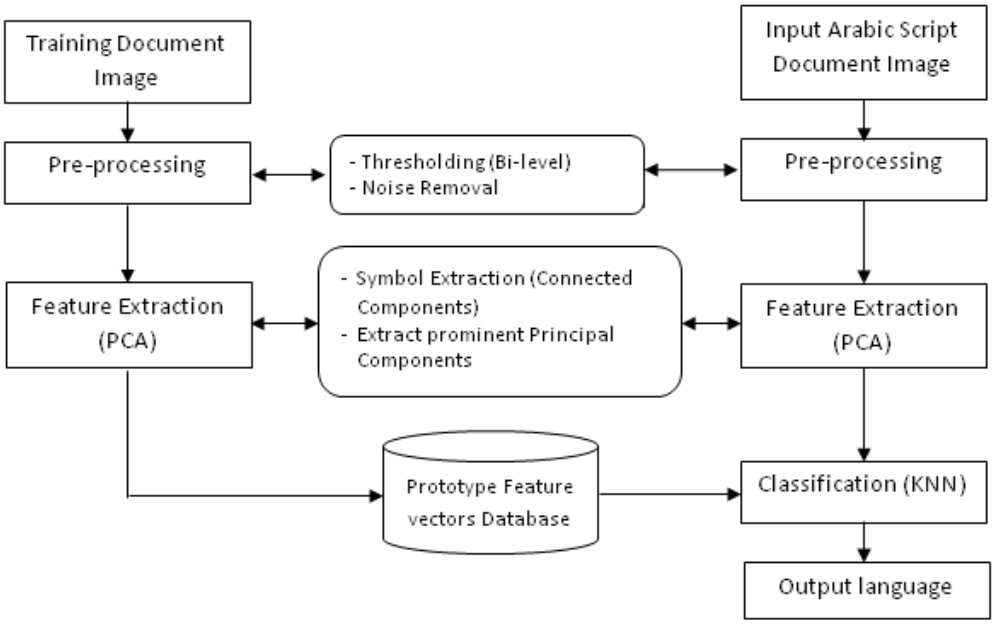

Fig. 3. Arabic Script Recognition System

The subsystem in **Fig. 3** is a PCA based subsystem. PCA is a well-known statistical method, [20], that analyzes the covariance structure of multivariate statistical observation for determining the features that explain as much of the total variation in the data as possible with as few features as possible. The curse of dimensionality pertinent to many learning algorithms leads to both the drastic increase of computational complexity and classification error in high dimensions. A template based strategy for identifying languages within Arabic script is adopted, where each template or textual symbol is an image of 30 *30 extracted from the document. This gives us 900 features which are quite high and therefore, PCA is applied on the templates before applying a learning algorithm.

The Arabic script languages Arabic, Urdu and Persian have a lot of words in common, thus, making the data more complex and nonlinear. KNN is a traditional simplest learning algorithm which is good in handling non-linear and multi modal data [21]. Further it is a non-parametric algorithm and one need not deal with the unknown densities. In their work, Novakovic and Rankov, [22], concluded that the results of classification performance for KNN, also called IB1 algorithm, are better compared to other classifiers. One of the major draw backs of this algorithm is that it cannot scale to higher dimensionality of the data. Real-time recognition on low-cost computers demands the efficiency of feature extraction and the simplicity of classification algorithms. In this work, PCA followed by KNN are adopted for identifying Arabic scripted languages in the PCA based subsystem. Classification problems involving multiple classes like the proposed system can be addressed in two different ways: by using a classifier that can deal directly with them, or alternatively, dividing the original data set into two-class subsets, learning a different binary model for each new subset. The latter technique is known as binarization. The major drawback with binarization is the extra computational burden it comes with. In the proposed work we are using a non-parametric classifier KNN to identify 3 different languages of Arabic script. In, [23], it has been shown that using binarization improves classification accuracy of KNN; however, the improvement is not that significant. Thus, to leverage the possible benefits of binarization and at the same time not hitting much on computational complexity, KNN classifier does the initial identification of 3 Arabic scripted languages and later binarization is used to resolve ties. The ties referred to here are not the ties that incur due to K nearest neighbors in KNN algorithm, but the ties that arise due to more than one class label having equal counts of the different 8- connected components extracted from the document under test. Whenever the KNN classifier is dealing directly with 3 classes and has more than one language attached to equal number of extracted components, the extracted components are input to the KNN ensemble of binary classifiers.

There are two well-known strategies for binarization, one-vs-one (OVO) and one-vs-all (OVA). OVA have received less attention in literature in comparison to OVO. Further OVA strategy, is usually affected by imbalanced data, which is a really hard problem in Machine Learning [24]. Rifkin and Aldebaro claimed that OVA scheme is as accurate as any other approach, but when the base classifiers are well-tuned. In our work, we employed OVO binarization strategy with the simplest voting scheme .The voting scheme in OVO combines the outputs of different binary classifiers [25].

Algorithm 1: *Training Algorithm*

Input: Set of Arabic D_A, Persian D_P and Urdu D_U documents;
Output: Training matrices T, T_{AP}, T_{AU}, T_{PU}; Mean vectors $\bar{S}, \bar{S}_{AP}\bar{S}_{AU}\bar{S}_{PU}$; Eigen matrices E, E_{AP}, E_{AU}, E_{PU}; Variance vectors V, V_{AP}, V_{AU}, V_{PU};

1. Extract p 8-connected components (symbol) from each document in D_A. Consider the symbol for training only if it has an Aspect Ratio of 1.5. Scale each symbol to a standard matrix with size of $n \times n$ where $n = 30$. Convert the matrix to a row vector of size $m = n \times n$. Formulate the symbol matrix S_A by placing all the row vectors of p symbols. Use the same procedure for documents in D_P and D_U to obtain the symbol matrices S_P and S_U.
2. The different text symbol matrices are computed as follows:
   $S \coloneqq S_A \cup S_P \cup S_U$
   $S_{AP} \coloneqq S_A \cup S_P$
   $S_{AU} \coloneqq S_A \cup S_U$
   $S_{PU} \coloneqq S_P \cup S_U$
3. Compute the Mean Vector $\bar{S}$ by obtaining the mean of every column of matrix S using the formula (2.1). Use (2.2) to compute the difference matrix ΔS by centering the matrix S by subtracting the mean vector $\overline{S}$ from S. Similarly compute $\bar{S}_{AP}\bar{S}_{AU}\bar{S}_{PU}$ and center the matrices S_{AP}, S_{AU},S_{PU}
4. Perform principle component analysis (PCA) on the matrix ΔS to obtain the following items.
   V- Variance vector where each element provides the cumulative variance defined in (2.3)
   E- A matrix of Eigen vectors arranged in decreasing order of their Eigen values.
   T – The training matrix computed using (2.4)
   Follow the same procedure with ΔS_{AP}, ΔS_{AU}, ΔS_{PU} to obtain the corresponding ($T_{AP,} E_{AP,} V_{AP}$), ($T_{AU,} E_{AU,} V_{AU}$) and ($T_{PU,} E_{PU,} V_{PU}$).
5. Add a column to the training matrix T such that each row of this column represents the language of the corresponding 8-Connected component in that row. Do the same for T_{AP},$T_{AU,}$ $T_{PU.}$

Algorithm 2: *Identification of the Language*

Input: Arabic script document D under scrutiny, Cumulative percentage variance v of the principal components to be considered. Training matrices T, T_{AP}, T_{AU}, T_{PU}; Mean vectors $\bar{S}, \bar{S}_{AP}\bar{S}_{AU}\bar{S}_{PU}$; Eigen matrices E, E_{AP}, E_{AU}, E_{PU}; Variance vectors V, V_{AP}, V_{AU}, V_{PU};
Output: Language Identified I

1. Extract n_c 8- connected components (symbols) from the document D. Consider the symbol for testing only if it has an Aspect Ratio of 1.5. Scale each symbol to a standard matrix with size of $n \times n$ where $n = 30$. Convert the matrix to a row vector of size $m = n \times n$. Formulate the test symbol matrix U by placing all the row vectors of q symbols.
2. Center the data with respect to $\overline{S}$ to obtain matrix ΔU by subtracting it with the mean vector $\overline{S}$.
3. Transform the matrix ΔU by multiplying with the required features (Eigen Vectors whose cumulative percentage variance is >= desired cumulative percentage variance v)to obtain the transformed matrix F which is computed as in (2.6)
4. Initialize the counts: $I_{A,} = 0$, $I_P = 0$, $I_U = 0$.

5. For i = 1 to q
Use KNN algorithm with inputs T *and* F_i to find the language G.
If (G='Arabic') then
I_A++;
Else if(G='Persian') then
I_P++; Else I_U++;
End (if)
End (For)
6. Find the maximum count of I_A, I_P, I_U. Language is selected as the one with the maximum count.
7. If top 2 counts are equal, then tie resolution technique is used as follows.
8. Repeat step 2& 3 to obtain F_{AF} using ($\bar{S}_{AP}$, V_{AP}, E_{AP}). Similarly obtain F_{AU} and F_{PU} using ($\bar{S}_{AU}$, V_{AU}, E_{AU}), ($\bar{S}_{PU}$, V_{PU}, E_{PU}), respectively.
9. Initialize the counts : I_A = 0, I_P = 0, I_U = 0.
For i = 1 to q
Use KNN algorithm with (T_{AP}, F_{AP}) as inputs to obtain the language G_{AP}. Similarly use KNN
with (T_{AU}, F_{AU}) and (T_{PU}, F_{PU}) to obtain the languages G_{AU} and G_{PU} respectively. The language G is the one with maximum occurrence among (G_{AP}, G_{AU} and G_{PU}).
If (G='Arabic') then
I_A++;
Else if(G='Persian') then
I_P++; Else
I_U++;
End (if)
10. End (For)
11. Find the maximum count of I_A, I_P, I_U. Language is selected as one with the maximum count.
12. End(if)

4. Experimental Results and Observations

The impact of the proposed research is on automatic archiving of multicultural documents, automatic selection of script specific OCR in a multilingual environment, automatic extraction of Arabic text from multi-script documents, training officers in security agencies or academic/research institutions in using such a system and providing a new research dimension by developing modified/new methods for the identification/recognition of different Arabic scripts. The experimental results from the implementation of the proposed system in **Fig. 3** with different fractions of variances from the PCA features are provided in this section.

The training data consists of 189 documents (63 for each language Arabic Urdu & Persian). The documents were picked randomly from the books available on line. Each document was converted to a binary image with suitable trimming at a resolution of 300. Training involved extracting 8-connected components from each document. Since alphabets in Arabic scripted languages are wider in comparison to their height only those

extracted components that had an aspect ratio of 1.5 were considered. About 25 components were extracted per document

The test dataset consists of 186 documents containing three different Arabic scripted languages (i.e., Arabic, Persian, Urdu) again picked randomly from a separate set of online books. Similar to training, every document was converted to a binary image with suitable trimming at a resolution of 300. The details of the dataset ratios are explained in **Table 2** below.

Table 2. Experimental Setup

Language of	Size of	% of Training	Size of Testing	% of Testing
Arabic	63	50.0%	63	50.0%
Farsi	63	50.8%	61	49.2%
Urdu	63	50.4%	62	49.6%
Total	189	50.4%	186	49.6%

The documents were extracted from various online books containing different Arabic script languages. The set of books for the training and testing were different. As seen from **Table 2** the overall train to test ratio is 50.4:49.6. The result achieved is 100 %, but at the cost of many components. Thus, more components have to be extracted to identify the language of the document. This is similar to ensembles, where each component in the ensemble is a connected component.

The PCA-KNN Model adapted in the current work for identifying Arabic Scripted languages at document level relies on extracting 8-conncected components from a document. This model was mainly tested with respect to two important model parameters the number of principal components and the number of 8- connected components extracted. The parameter k of the KNN algorithm was set to 10. This is a sensible setting as used in [19][26].Further it was seen that with the proposed strategy 100 % recognition accuracy was observed with principal components >= 15 and number of 8-connected components >=18 on a 51 %Training: 49 %Test dataset where the document for testing were extracted from the books not present in the training dataset. Thus, it was seen the accuracy was mainly impacted by the parameters Number of connected components extracted from the test document and the number of principal components. Though changing K might affect the accuracy for the situations with less principal components and less connected components, a significant improvement in accuracy is not guaranteed. For the identification at a document level as in the current work extraction of 18 connected components might not be a challenge, but for identification at line level /word levels these many numbers of connected components might not be available. Therefore, the number of 8-connected components is a very important factor for the accuracy of identifying Arabic scripted languages and obtaining high-level accuracy with least number of 8-connected components could be an interesting direction for further research.

Figs. 4 – 7 show the initial results of testing the proposed system on 186 documents containing three different Arabic scripted languages (i.e., Arabic, Persian, Urdu) at

variances: 30, 40, 50, 60, 70, 80, 90, and 100 each was tested to recognize the language of the document using different numbers of components ranging from 1 to 25. **Figs. 4 - 6** show the results for each language and **Fig. 7** show the average results for the whole recognition system. The purpose of experimenting with different variance and number of components is to find a combination of both that would give the optimal accuracy with less computational burden. It is evident that having more features (high variance) and more components would achieve higher accuracy, but with extra computational burden. It is desirable to obtain good accuracy at lower variance and small number of connected components. Lower variance indicates less number of features and this along with less number of components would reduce the computation time for actual recognition. **Table 3** shows the percentage variance and corresponding number of features of the training dataset.

Table 3. Percentage Variance & corresponding number of features of the training dataset.

Required Percentage Variance	30	40	50	60	70	80	90	100
Cumulative Percentage Variance	35	42	50	60.2	70	80	90	100
Number of principal components/Features	3	5	8	15	27	53	126	900
% of Dimensionality Reduction	99.7	99	99.1	98.3	97	94	86	0

From the experimental results, it is noticed that the system was unable to recognize some documents since there was a tie in recognizing some documents; for example, in some cases the system gets a 50-50 tie between two languages and is unable to decide. The reduction in accuracy is due to ties and it is more prominent with even number of components, **Figs. 4 – 6**. Therefore, a further step was incorporated into the system in order to resolve such ties. As explained above the language for the document is based on majority voting scheme where the language for the document is the language of the majority of the 8-connected components. In case of ties the ensemble of 3 KNN binary classifiers is used to resolve ties. The average percentage recognition accuracy rate after passing the unclassified documents through the ensemble is shown in **Fig. 8**.

It is noticed that it may be difficult to get enough components from some documents due to the Aspect Ratio filter. The number of documents tested for some PCA dimensionalities maybe less than the number of documents tested for other PCA dimensionalities, and this is mainly noticed when the number of components needed is above 16. However, the lowest total number of documents used in testing the three languages for a specific dimension was 141 from 186 totally used in this work.

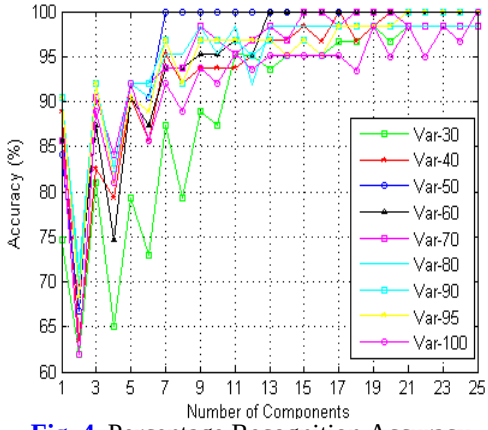

Fig. 4. Percentage Recognition Accuracy Results for Arabic Language Documents

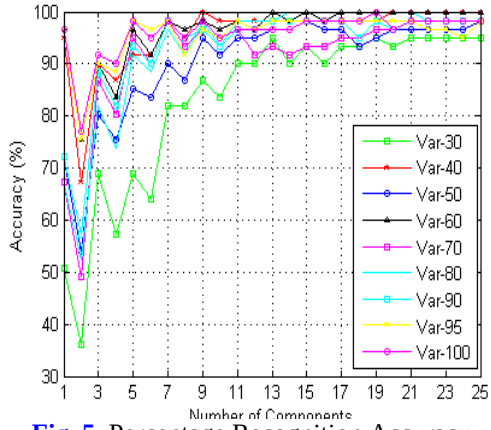

Fig. 5. Percentage Recognition Accuracy Results for Persian Language Documents

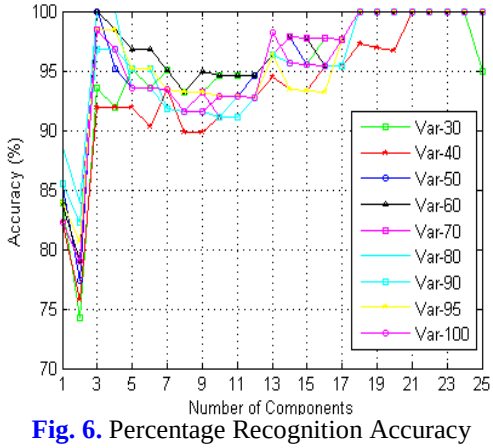

Fig. 6. Percentage Recognition Accuracy Results for Urdu Language Documents

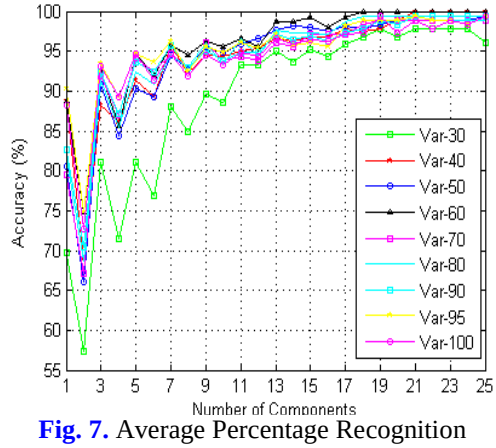

Fig. 7. Average Percentage Recognition Accuracy Rate for the proposed System

The graphs in **Fig. 8** show the average recognition rate for all languages exceed 95% for components above 12 and since some documents may not provide more than 21 acceptable components to be used in the recognition then the interested region of interest is decided to be between 13 and 21 components and a variance value between 50 and 70.

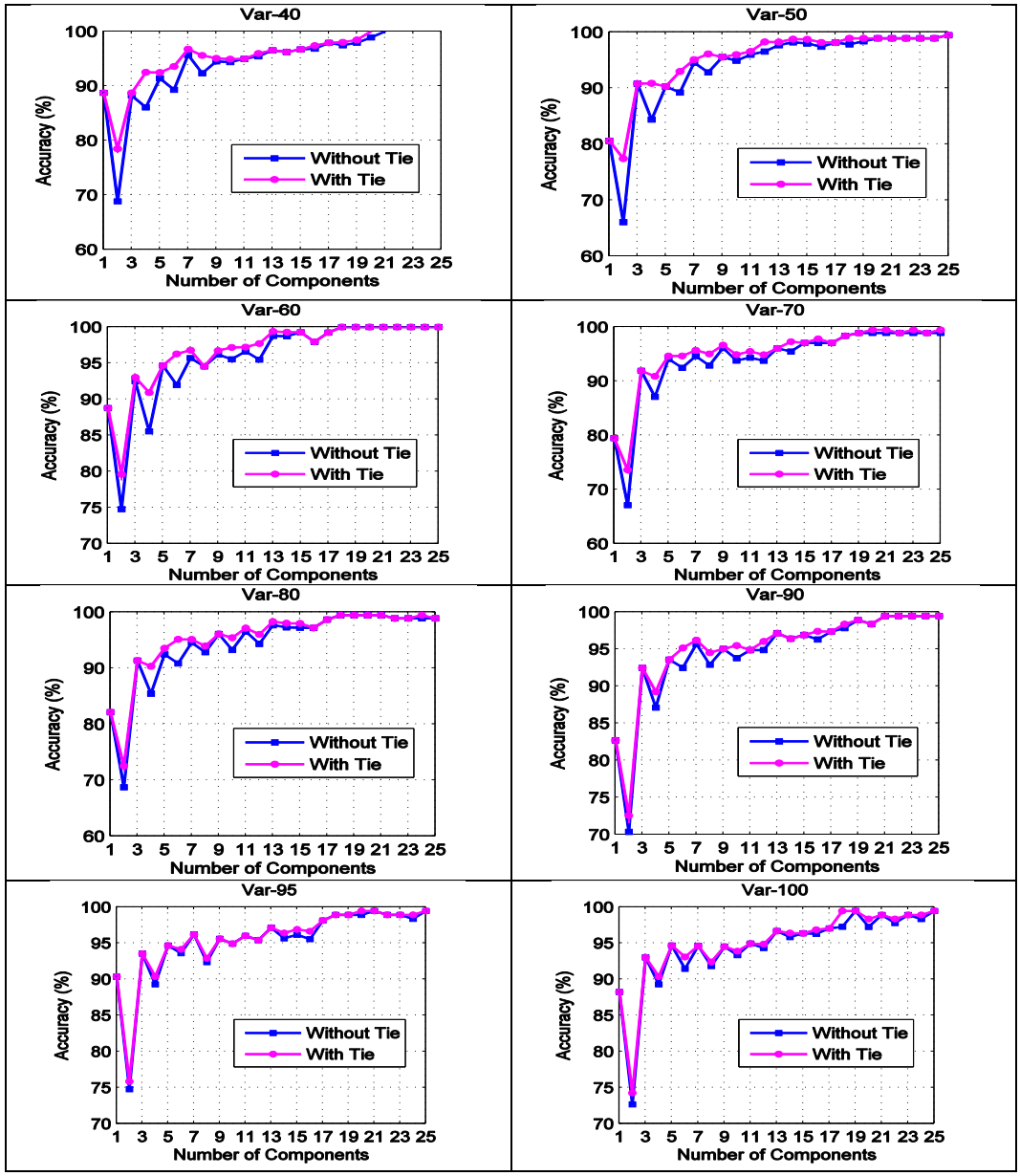

Fig. 8. Average Percentage Recognition Accuracy Rate after Tie Resolution for the proposed Arabic Scripted Language System

Table 4 and **Table 5** show the percentage average misclassified and unclassified documents respectively for the given region of interest. Finally, the average recognition rate (% Accuracy rate) is provided in **Table 6**. From this table, the optimal values for the number of components is provided to be any component greater or equal to 18 and a

variance value of 60% in order to get a perfect, 100% accurate recognition rate from the proposed Arabic Script Recognition System.

Table 4. The Number of Components Vs. the Average Percentage of Misclassified Documents for Variances 50, 60 and 70.

		% Misclassified Arabic Documents			% Misclassified Persian Documents			% Misclassified Urdu Documents			Ave. % Misclassified Documents		
		50	60	70	50	60	70	50	60	70	50	60	70
		Variance											
No. of Components	13	0.00	0.00	1.61	1.64	0.00	6.56	1.85	0.00	1.85	1.13	0.00	3.39
	14	0.00	0.00	0.00	1.64	0.00	1.64	2.17	0.00	0.00	1.18	0.00	0.59
	15	0.00	0.00	0.00	1.64	0.00	6.56	2.22	2.22	0.00	1.19	0.60	2.38
	16	0.00	0.00	0.00	1.64	0.00	4.92	2.27	0.00	0.00	1.20	0.00	1.81
	17	0.00	0.00	0.00	3.28	0.00	4.92	2.33	0.00	2.33	1.82	0.00	2.42
	18	0.00	**0.00**	0.00	3.28	**0.00**	3.28	0.00	**0.00**	0.00	1.26	**0.00**	1.26
	19	0.00	**0.00**	0.00	3.28	**0.00**	3.28	0.00	**0.00**	0.00	1.29	**0.00**	1.29
	20	0.00	**0.00**	0.00	3.28	**0.00**	1.64	0.00	**0.00**	0.00	1.32	**0.00**	0.66
	21	0.00	**0.00**	0.00	3.28	**0.00**	1.64	0.00	**0.00**	0.00	1.34	**0.00**	0.67

Table 5. The Number of Components Vs. the Average Percentage of Unclassified Documents for Variances 50, 60 and 70.

		% Unclassified Arabic Documents			% Unclassified Persian Documents			% Unclassified Urdu Documents			Ave. % Unclassified Documents		
		50	60	70	50	60	70	50	60	70	50	60	70
		Variance											
No. of Components	13	0.00	0.00	0.00	0.00	0.00	0.00	1.85	1.85	1.85	0.56	0.56	0.56
	14	0.00	0.00	1.61	0.00	0.00	3.28	0.00	2.17	0.00	0.00	0.59	1.78
	15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	16	0.00	0.00	0.00	1.64	0.00	1.64	0.00	4.55	0.00	0.60	1.20	0.60
	17	0.00	0.00	1.64	0.00	0.00	0.00	0.00	2.33	0.00	0.00	0.61	0.61
	18	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00
	19	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00
	20	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00
	21	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00	0.00	**0.00**	0.00

From the comparison in **Table 7**, it is noticed that none of the work on page level script recognition addressed the recognition/identification of more than one language continuing Arabic script. This is due to the complexity of the script and the similarity of the different languages within the Arabic script.

Table 6. Total Average Recognition Rate for the proposed Arabic Scripted Languages Identification System

No. of Components	Average Misclassification	Average Unclassified	Total Average Recognition Rate
13	0.00	0.56	99.44
14	0.00	0.59	99.41
15	0.60	0.00	99.40
16	0.00	1.20	98.80
17	0.00	0.61	99.39
18	0.00	0.00	100.00
19	0.00	0.00	100.00
20	0.00	0.00	100.00
21	0.00	0.00	100.00

Table 7. Comparison of Page-level Script Recognition Methods

References	Method		Scripts identified	Accuracy
	Features	Classifier		
Proposed Arabic Script Recognition System	Horizontal/Vertical projection profiles	Euclidean	Arabic, Latin and Han Scripts	100%
	PCA	KNN	Arabic, Persian and Urdu	100%
Hochberg et al. [10]	Textual symbols	Hamming Distance Classifier	Arabic, Armenian, Devnagari, Chinese, Cyrillic, Burmese, Ethiopic, Japanese, Hebrew, Greek, Korean, Latin, Thai	96%
Wood et al. [11]	Horizontal/Vertical projection profiles	-	Arabic, Cyrillic, Korean, Latin	*NA
Hochberg et al. [14]	horizontal and vertical centroids, white holes, aspect ratio and sphericity	Fisher linear discriminants	Arabic, Chinese, Cyrillic, Devnagari, Japanese, Latin	88%
Peake & Tan [15]	GLCM	KNN	Korean, Latin, Chinese, Greek, Russian, Persian and Malayalam	77.14%
	Multi-Channel Gabor filter			95.71%
Tan [18]	Rotation Invariant texture features	Weighted Euclidean distance	Chinese, Greek, Malayalam, English, Russian, Persian	96.70%
Lu and Tan [19]	Stroke density and distribution	KNN	Arabic, Chinese, Hebrew and Roman	95.63%

Table 7 shows the accuracy results for the different methods found in literature and the closest work to our proposed approach is the work of Hochberg [10] where he used a similar approach by first filtering the Han scripts from Latin scripts, then using LDA and Euclidean distance he implemented a system to distinguish between the three different Han scripts used in his study. However, our approach first used vertical and horizontal projection profiles to separate the Arabic documents from other scripts. This provided a100% accuracy due to the cursive nature of the Arabic printed script compared to Latin printed scripts. Following this, the Arabic documents were passed through the proposed

Arabic Scripted Language Identification System in order to identify the language of the Arabic script by using PCA and KNN, here, the number of PCA features were optimized providing a 100% accuracy with lower dimensionality level, hence only 15 features from the 900 features available for each component were used with a minimum of 18 connected components needed from each document. The comparison above may not be a fair comparison due the fact that the database of documents images used for the different approaches shown in **Table 7** are not the same. Each approach used its own database and contains different languages/scripts, therefore, to conclude in order to provide a fair comparison with other methods available in literature the same database and scripts studied must be used to be able to draw realistic and fair comparisons. Therefore, to the best knowledge of the authors from surveying the literature this is the only work available that addresses multi-languages containing Arabic Script. Finally, the proposed approach with the accuracy obtained proves to be suitable for this application or similar research work.

5. Conclusion

The diversity of population demography urges the need for script recognition systems that not only distinguishes between scripts, but also identifies the language of the script. This is necessary to convert document images into text and translate the text into the official language of the country. This work recommends that more work is needed in this area to attract the attention of researchers to this sensitive area of research. Therefore, the authors conclude that more work is still needed in this area in order to provide satisfactory results; in addition, it is recommended that work on creating a common dataset through a specific agency which carry research on multi-script documents is very crucial to speed up the research in this area.

The proposed PCA based script recognition system is divided into two subsystems. The first subsystem distinguishes between Latin, Han and Arabic scripts by extracting horizontal and vertical profiles of the document. This technique has a 100% accuracy rate in identifying Arabic Script from other scripts. The second subsystem applies PCA on the textual symbols of the Arabic scripted document to extract features. The results show that the average recognition rate for the 186 documents tested in this work provide acceptable average minimum recognition rate of 95% when using a minimum of 12 components for any variance value above 30. The results of using PCA for extracting features and dimensionality reduction concludes that the optimal number of components for the system is 18 at a variance of 60% (i.e. only 15 features are used instead of 900) provide a 100% recognition rate in identifying the Arabic scripted languages investigated in this work. In this work, a total of 186 documents were tested divided evenly between the three language scripts. The KNN is proved to provide excellent results when used with PCA for this application. The results convince us that the proposed PCA based script recognition system will have a profound impact on automatic archiving of multilingual documents and the selection of script specific OCR in multi lingual environments. Finally, it may play a crucial role in the area of national security where the

confiscation of documents in different languages from suspected individuals could be identified using such a system.

Acknowledgement

The authors would like to thank the deanship of scientific research at Taibah University for their support.

References

[1] Arabic Alphabet. Encyclopaedia Britannica online.http://www.britannica.com/eb/article-9008156/Arabic-alphabet. Retrieved 23-11-2013.

[2] Y. M. Alginahi, "A Survey on Arabic Character Segmentation," *International Journal on Document Analysis and Recognition (IJDAR)*, Vol.16, pp. 105 – 126, No.2 2013. Article (CrossRef Link).

[3] Wikipedia.org, http://en.wikipedia.org/wiki/Urdu. Retrieved 17-12-2013.

[4] Studypersion.com, http://www.studypersian.com/starter/alefba.htm, viewed15-12-2013.

[5] S. Abirami & D. Manjula, "A Survey of Script Identification Techniques for Multi-Script Document Images," *International Journal of Recent Trends in Engineering*," Vol. 1, No. 2, May 2000.

[6] D. Ghosh, T. Dube, and A. P. Shivaprasad, "Script Recognition – A Review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 32, pp. 2142 –2161, No. 12, 2010. Article (CrossRef Link).

[7] A. L. Spitz, "Determination Of The Script And Language Content Of Document Images," *IEEE Tran. On Pattern Analysis and Machine Intelligence*, Vol. 19, pp.234-245, No. 3 1997. Article (CrossRef Link).

[8] U. Pal, S. Sinha and B. Chaudhuri, "Multi-Script Line Identification from Indian Documents," in *Proc. of International Conference on Document analysis and Recognition*, Edinburgh, pp. 880-884, Aug. 2003. Article (CrossRef Link).

[9] S. Kanoun, A. Ennaji, Y. Lecourtier, and A.M. Alimi, "Script and Nature Differentiation for Arabic and Latin Text Images," in *Proc. of the International Workshop Frontiers in Handwriting Recognition*, Niagra, pp. 309-313, Aug. 2002. Article (CrossRef Link).

[10] J. Hochberg, P. Kelly, T. Thomas, and L. Kerns, "Automatic Script Identification from Document Images Using Cluster-based Templates," *IEEE Trans. Pattern Analysis & Machine Intelligence*, Vol. 19, pp. 176-181, No. 2, Feb. 1997. Article (CrossRef Link).

[11] S.L. Wood, X. Yao, K. Krishnamurthi, and L. Dang, "Language Identification for Printed Text Independent of Segmentation," in *Proc. of Int'l Conf. Image Processing*, Washington D.C., Vol. 3, pp. 428431, 1995. Article (CrossRef Link).

[12] A. M. Namboodiri and A. K. Jain, "Online Handwritten Script Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 26, pp. 124 -130, No. 1, January 2004. Article (CrossRef Link).

[13] A. Busch, W. W. Boles, and S. Sridharan, "Texture for Script Identification," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 27, pp. 1720 – 1732, No. 11, Nov. 2005. Article (CrossRef Link).

[14] J. Hochberg, K. Bowers, M. Cannon, and P. Kelly, "Script and Language Identification for Handwritten Document Images," *Int'l J. Document Analysis & Recognition*, Vol. 2, pp. 45-52, No. 2/3, Dec. 1999. Article (CrossRef Link).

[15] G. S. Peake and T. N, Tan, "Script and Language Identification from Document Images," in *Proc. of Lecture Notes in Computer Science*, *Asian Conference on Computer Vision*, Hong Kong, LNCS-1352, pp. 97 – 104, Jan 1998.

[16] M. Benjelil, R. Mullot and A. M. Alimi, "Language and script identification based on Steerable Pyramid Features," in *Proc. of Frontiers in Handwriting Recognition*, pp. 716 – 721, 2012. Article (CrossRef Link).

[17] S. Lu, and C. L. Tan. "Script and language identification in noisy and degraded document images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30.1, pp. 14-24, 2008. Article (CrossRef Link).

[18] T. N. Tan, "Rotation invariant texture features and their use in automatic script identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20.7, 751-756, 1998. Article (CrossRef Link).

[19] Lu, Shijian, and Chew Lim Tan. "Automatic detection of document script and orientation," in *Proc. of Ninth International Conference on Document Analysis and Recognition*, ICDAR 2007. Vol. 1. IEEE, 2007. Article (CrossRef Link).

[20] I. T. Jolliffe, Principal Component Analysis, Springer, New York, NY, 1986. Article (CrossRef Link).

[21] Q. P. He and J. Wang. "Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes," *IEEE Transactions on Semiconductor Manufacturing*, Vol.20, No. 4, pp. 345-354, 2007. Article (CrossRef Link).

[23] M. Galar, et al. "An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes," *Pattern Recognition*, Vol. 44, No. 8, pp. 1761-1776, 2011. Article (CrossRef Link).

[24] N. V. Chawla, N. Japkowicz, and A. Kotcz. "Editorial: special issue on learning from imbalanced data sets," ACM Sigkdd Explorations Newsletter, Vol. 6, No. 1, pp. 1-6, 2004. Article (CrossRef Link).

[25] R. Rifkin, A. Klautau, "In defense of one-vs-all classification," *Journal of Machine Learning Research*, Vol. 5, 101–141, 2004.

[26] Chiang, T. H., Lo, H. Y., & Lin, S. D. A Ranking-based KNN Approach for Multi-Label Classification. In ACML. 2012, pp. 81-96.

Yasser M.Alginahi earned a Ph.D., in Electrical Engineering from the University of Windsor, Canada. He is a licensed Professional Engineer and a senior member of IEEE and IACSIT. He is an Associate Prof. at the Dept. of Computer Science, Deanship of Academic Services, Taibah University. His current research interests include Information Security, Image Processing, Document Image Analysis, Pattern Recognition, and Modeling and Simulation.

Mohammd Mudassar Obtained M.Tech in computer Cognition Technology from University of Mysore in December 2004. Have an overall of 10 years of experience In Software Development, Applied R &D and Teaching. Worked in different capacities as Senior Software Engineer, Scientist, Project Manager & Tech lead. Currently working as a lecturer in Taibah University Al Madinah Al Munawarrah.

Muhammad Nomani Kabir obtained his MSc in Computational Sciences in Engineering and PhD in Computer Science from the University of Braunschweig, Germany. He is currently a Senior Lecturer at the Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, Malaysia. His research interests include issues related to modeling and simulation, computational methods, image processing and computer networks.