Open Access

Multiclass Classification of Unconstrained Handwritten Arabic Words Using Machine Learning Approaches

Jawad H. AlKhateeb*, Jianmin Jiang, Jinchang Ren, Fouad Khelifi and Stan S. Ipson

School of Informatics (EIMC), University of Bradford, BD7 1DP, UK

Abstract: In this paper, we propose and describe efficient multiclass classification and recognition of unconstrained handwritten Arabic words using machine learning approaches which include the K-nearest neighbor (K-NN) clustering, and the neural network (NN). The technical details are presented in terms of three stages, namely preprocessing, feature extraction and classification. Firstly, words are segmented from input scripts and also normalized in size. Secondly, from each of the segmented words various feature extraction methods are introduced. Finally, these features are utilized to train the K-NN and the NN classifiers for classification. In order to validate the proposed techniques, extensive experiments are conducted using the K-NN and the NN. The proposed algorithms are tested on the IFN/ENIT database which contains 32492 Arabic words; the proposed algorithms give good accuracy when compared with other methods.

Keywords: Offline Arabic handwritten recognition, Full word features, Feature extraction, Multi class classification, Machine learning, KNN, NN.

1. INTRODUCTION

There is a rapid development of automated document classification systems which are capable of classifying the text in terms of character level or in terms of word level. In the literature, there are several techniques for classifying the text such as K-NN, NN, Support Vector Machine (SVM), Hidden Markov Models (HMM), Bayesian Classification, decision tree based, and others [1]. Most of the researchers focus their work on Latin and Chinese documents [2]. However, work on Arabic documents is a challenging task due to the complexity of the Arabic Language. In general, it is more difficult to recognize handwritten scripts than printed ones and the former needs to consider the variation of writing, e.g. the stroke length, the stroke regularity and the stroke location [3]. As a result, the performance on handwritten Arabic text classification is still poor and needs to be further improved upon [4].

Handwriting recognition (HWR) is the mechanism of transforming the text to its symbolic representation, which plays essential roles in many useful applications in human computer interaction for different devices such as office automation, cheques verification, mail sorting, and a large variety of banking, business as well as natural human-computer interaction. Handwriting recognition for Latin languages has been conducted and significant achievements have been done. However, there was little work for the Arabic handwriting recognition. This is due to the complexity of the Arabic language and lack of public Arabic handwriting databases [2, 3]. In general, HWR can be categorized into online based and offline based systems [3]. The former relies on dynamic information of pen movement like the strength and sequential order during the writing to assist the recognition. However, such information is missing in the latter cases, which makes offline recognition systems more difficult [1].

Regarding HWR systems for Arabic scripts, they can be categorized into segmentation based and segmentation free ones, depending on whether the scripts need to be segmented into small elements for classification. Segmentation based method is also namely analytical approach, in which words or letters are extracted and used for recognition. On the other hand, segmentation free methods are known as a global approach to take the whole script image as input for the HWR system. Though the global approach makes the recognition process simpler by avoiding the difficulty in character segmentation, it suffers more to huge vocabulary of words than analytical approaches [4].

In this paper, we focus on offline recognition of handwritten Arabic text, using the word level criteria [2]. After word segmentation, we have extracted five groups of features including block-based where the absolute mean value for each block in the word image is computed, global and block-based DCT coefficients, moments and wavelet coefficients. Each group of the features is then respectively inputted to K-NN and NN classifiers to test the effectiveness of classification. To deal with the huge number of classes in classification, a modified solution has been proposed in training NN classifier to achieve convergence. Interesting results are achieved on the IFN/ENIT benchmark database [5].

The rest of this paper is structured as follows. Section 2 describes the characteristics of Arabic language, which is useful in motivating our work in words segmentation etc. Section 3 presents the state of the art in terms of classification of printed and handwritten Arabic scripts. In Section 4, extraction of five groups of features is discussed, and the technical details in training the classifiers are illustrated in Section 5. Experimental results and introduction to the IFN/ENIT database are presented in the Section 5, and finally some concluding remarks are drawn in Section 6.

*Address correspondence to these authors at the School of Informatics (EIMC), University of Bradford, BD7 1DP, UK;
E-mail: J.h.y.alkhateeb@bradford.ac.uk

1876-8253/09 2009 Bentham Open

2. THE ARABIC LANGUAGE

Arabic is widely used and written by more than 250 million people in the world [3]. By nature, Arabic text is cursive, which makes its recognition rate lower than that of printed Latin. Arabic language also uses letters, which is similar to English writing. However, the difference here is that each of the 28 letters within the Arabic alphabet has either two or four shapes depending on its position in the text written, and the whole text is written from right to left in a cursive way. Each letter may have up to four various shapes according to its position in the word within the text, i.e. at the start, in the middle, at the end or alone [1] as shown in Table **1**. For example letter Ayn (ع) has the following shapes: start ع-, middle -ع-, end -ع, and alone ع.

In addition, Arabic language uses diacritical markings such as fattha, dumma, kasra, hamza(zigzag), shadda, or madda, and the presence or absence of vowel diacritical indicates different meanings [1]. For example مدرسة can be school or teacher and كلية can be college or kidney. Basically, the diacritical markings can be ignored in handwriting, but they are compulsory to differentiate between the possible meanings if the words are isolated. Using dots makes some Arabic letters special as follows [2, 3]:

a) Ten Arabic letters have one dot
(ب،ج،خ،ذ،ز،ض،ظ،غ،ف،ن)

b) Three Arabic letters have two dots (ت،ق،ي)

c) Two Arabic letters have three dots (ث،ش)

d) Several Arabic letters presents loop (ف،ق،م،ـمـ،و،ةص،ض،ط،ظ،ـعـ،ـغـ،)

It is worth noting that removal of any of these dots will lead to a misrepresentation of the character. Therefore, efficient pre-processing techniques have to be used in order to keep these dots and avoid misunderstanding during processing such as removing noise for image enhancement. Generally, the handwritten Arabic text on a page contains several lines, and each line is further divided into words. There are spaces between the lines to separate them; also there are spaces between the words to indicate word boundaries. There are six letters which are not connected from the left resulting in the separation of the word into sub-words. Normally, the space between the sub-words is one third of the space between the words, but it varies in handwritten and leads to difficulties in word segmentation [2].

3. STATE OF THE ART

Abuhaiba *et al.* [6] dealt with several problems in the processing of binary images of handwritten text documents, which mainly targeted for identifying the spurious points from the thinned images. First, straight lines of a textual stroke are extracted through applying the distance transform to the thinned image, which keep the structural information of the original pattern. Second, outlying pixels whose distance exceeds a given threshold are moved. Finally, lines from pages of handwritten text are extracted by finding the shortest spanning tree of a graph formed from the set of main strokes. By arranging extracted strokes in their written order, a list of elements are obtained and used for classification. Their method proved to be powerful and suitable for variable handwriting. This algorithm has been tested to a special database written by two writers. Alma'adeed *et al.* [7] introduced a system using HMM to classify handwritten Arabic words. First, the word images were normalized in terms of stroke width, slope, and the letter height to remove variations in writing. Second, the skeleton of the image was constructed and the features were calculated to represent the useful information of the word image. Thirdly, HMM was used for classification and tested on a special database. The recognition rate achieved was only 45% because some words conflict with each other. Further, the authors also introduced a system to recognize unconstrained handwritten Arabic words using multiple HMMs [8]. To improve the recognition rate, several global features were used to differentiate

Table 1. **Arabic Letter shapes**

Name	Alone (isolated)	Start	Middle	End
Alif	ا	ا	ـا	ـا
Baa	ب	بـ	ـبـ	ـب
Taa	ت	تـ	ـتـ	ـت
Thaa	ث	ثـ	ـثـ	ـث
Jeem	ج	جـ	ـجـ	ـج
Haa	ح	حـ	ـحـ	ـح
Khaa	خ	خـ	ـخـ	ـخ
Dall	د	د	ـد	ـد
Dhaal	ذ	ذ	ـذ	ـذ
Raa	ر	ر	ـر	ـر
Zaay	ز	ز	ـز	ـز
Seen	س	سـ	ـسـ	ـس
Sheen	ش	شـ	ـشـ	ـش
Saad	ص	صـ	ـصـ	ـص
Daad	ض	ضـ	ـضـ	ـض
TTaa	ط	ط	ـط	ـط
Dhaa	ظ	ظ	ـظ	ـظ
Ayn	ع	عـ	ـعـ	ـع
Ghyan	غ	غـ	ـغـ	ـغ
Faa	ف	فـ	ـفـ	ـف
Qaaf	ق	قـ	ـقـ	ـق
Kaaf	ك	كـ	ـكـ	ـك
Laam	ل	لـ	ـلـ	ـل
Meem	م	مـ	ـمـ	ـم
Noon	ن	نـ	ـنـ	ـن
Haa	ه	هـ	ـهـ	ـه
Waw	و	و	ـو	ـو
Yaa	ي	يـ	ـيـ	ـي

the words including the numbers of upper/lower dots and the numbers of segments, ascenders and descenders. This system was tested on a database of 4700 words written by 100 writers [9]. The recognition rate achieved was 60% before post processing. The codebook size was chosen after testing and selected different words for each group. There were eight groups where the first group had 90 words, second had 100 words, the third had 80 words, the fourth had 90 words, and eighth had 120 words. The recognition rate was different for each group. The first group had a 97% recognition rate, while the eighth group had only 60% recognition rate.

Khorsheed and Clocksin [10] presented a technique for extracting the structural features from Arabic cursive printed text. Several pre-processing tasks were performed including thinning based on Stentiford's algorithm [11] and skeleton centroid calculation to find a reference point relative to all segment locations. The features included starting and ending points of each segment, and extracted loops (a simple loop, a complex loop, or a double loop). The technique was tested with a small lexicon of 294 words acquired from a different text sources by using the HMM, and a recognition rate of up to 97% was achieved.

A holistic recognition system for recognizing Arabic cursive words was presented in [12], where template matching was employed. First, Fourier coefficients are extracted from a word image after normalization. Then, each word was represented by a template of the average Fourier coefficient values for training. The Euclidean distance is used in matching these word templates and the recognition rate achieved was over 90%. However, this system fails for many fonts. Khorsheed [13] presented another holistic recognition system for recognizing Arabic handwritten words. Preprocessing tasks performed included using the Zhang-Suen thinning algorithm [14] to generate the skeleton graph. Structural features for the handwritten script were extracted after skeletonization by decomposing the word skeleton into a sequence of links with an order similar to the word writing order. Using the line approximation [11], each line was broken into small line segments, which were transferred into a sequence of discrete symbols by using vector quantization (VQ) [15]. With this system, the HMM recognizer was applied with image skeletonization to the recognition of an old Arabic manuscript which can be found in [16]. One HMM was performed from 32 character HMMs, each with no restriction jump margin. The system was tested on 405 character samples of a single font extracted from a single manuscript. The recognition rates achieved were 87% and 72% with and without spell checking respectively.

Khorsheed [17] presented a recognition system based on the HMM to recognize Arabic text. Pre-processing was performed, using a slow median filter, to reduce salt and pepper noise. Statistical features were extracted from text image and fed to the recognizer. The recognizer was built on the HMM toolkit (HTK) [18]. The advantage of this system is the lexicon free approach which offers open vocabulary recognition. The system was able to learn complicated ligatures and overlaps. Different text images with different fonts were tested, and the recognition rate achieved was up to 92.4%. A tri-model implementation showed a better system performance than a mono-model implementation. Paechwitz and Margner [19] used a sliding-window based on the image representation of the word image using pixel values as main features. The sliding window is shifted across the word image from right to left to generate the feature vector. The word image is gray normalized image. The Karhunen Loeve Transformation (KLT) is performed in order to reduce the feature vector dimension.

Similar sliding window is also used in [20], in which the word image is divided into vertical overlapping frames with a constant width and variable heights. The sliding window is shifted from right to left and a feature vector is calculated for each frame. For each frame, 24 features are extracted using foreground pixel densities and concavity features. Also, there are 15 baseline independent features.

ElAbed and Magner [21] used the word skeleton graph where each vertical frame was further split into five zones with equal height. The lengths of all lines in each zone frame were calculated in four directions to form a 20 dimensional feature.

In Benouareth *et al.* [22, 23] statistical and structural features were utilized on the basis of the adopted segmentation in which implicit word segmentation is used to divide images into vertical frames of constant width for feature extraction. Based on maxima and minima analysis of the vertical projection histogram, morphological complexity of the Arabic handwritten characters is further considered.

In this paper, we proposed a system for handwritten Arabic word recognition, in which techniques are discussed in details in terms of preprocessing, feature extraction, and classification. Unlike traditional methods which recognize the whole image or letters, we conduct word-based processing for both effectiveness and flexibility. In addition, our method is validated using the IFN/ENIT database, where invariant moment features of words are inputted to train a neural network for classification.

4. THE PROPOSED METHOD

In our proposed system for multi-class system classification of handwritten Arabic words, three main phases are included, namely preprocessing, feature extraction, and classification. The preprocessing is utilized to remove noise and segment words. Afterwards, quite features are extracted from each word. Then these features are used for classification to identify the words, using machine learning approaches including k-NN and ANN. The proposed scheme is illustrated in Fig. (**1**), with technical details being presented as follows.

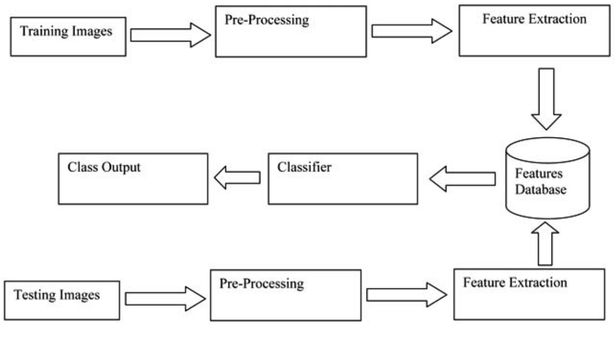

Fig. (1). The proposed scheme for word recognition.

4.1. Preprocessing

Pre-processing usually includes many relevant techniques like thresholding, skew/slant correction, noise removal, thinning, baseline estimation and segmentation of words. The main goal for preprocessing is to enhance the inputted signal and to represent it in a way which can be measured consistently for robust recognition. Since in our test sets of IFN/ENIT database all words have been cropped out during the development stage [5], the only thing we need to do as preprocessing is baseline detection and normalization. Regarding technical details on word segmentation as well as base line detection, please refer to our previous work in [24].

As for normalization, it is essential to remove the variation in the handwritten images for consistent analysis and robust recognition. Among many algorithms proposed for this purpose, the skeletonization technique is the most popular one used, and in our system the normalization algorithm in [21] is employed.

4.2. Feature Extraction

The main goal for feature extraction is to remove the redundancy from the data and gain more effective representation of the word image by a set of numerical characteristics. These features are then mapped into a classifier in order to classify input words. In this paper,

Five various methods are used to extract the word image features. The feature vector of the word image is computed. The database consists of 32492 Arabic words representing 937 classes. So, the dataset has been preprocessed so that the dataset is represented by 937 folders. Each folder was named with the city/village postcode. All the images have been read and saved in the proper folder based on their post code. This is has been done for sets a, b, c, and d as training set and set e as testing set. All the images on the training/testing set were read and their features were extracted and saved.

4.2.1. Overlapped Blocks (OB)

Knowing that the edges in the word image represent all the information in the image, a horizontal and vertical high pass filters are applied to the normalized word image. This is to extract the edges from the word image. Given the input image *I*, the horizontal high pass filter using prewitt operator G_h is applied to obtain I_h

$$I_h = G_h(I) \tag{1}$$

Likewise, the vertical high-pass filtered image I_v can be obtained with the vertical Prewitt operator G_v as follows

$$I_v = G_v(I) \tag{2}$$

Eventually, the filtered image *I'* is computed as

$$I'(i,j) = \max(abs(I_h(i,j), abs(I_v(i,j)) \tag{3}$$

According to one sample images in binary format in Fig. (**2a**), its normalized image and the filtered image result in grey are shown in Fig. (**2b**) and Fig. (**2c**) respectively.

The filtered word image is divided into overlapping blocks where the size of each block is 12 pixels and the length of the overlapping is 2 pixels which have been chosen empirically. The feature vector of the word image is computed by constituting the absolute mean value for each block in the filtered word image.

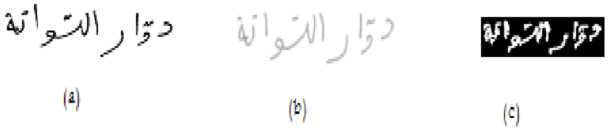

Fig. (2). One sample images of a word (**a**), normalized image (**b**), filtered image (**c**).

4.2.2. The DCT Features (DCT)

As it is well acknowledged that DCT coefficients can help to reduce redundancy and focus the energy of the image in a very limited frequency range [25], the DCT features of DCT coefficients are used in our system and extracted *via* two dimensional DCT. Given an image $f(i,j)$, its 2D DCT transform is defined as follows:

$$f(u,v) = \alpha(u)\alpha(v)\sum_{i=0}^{I-1}\sum_{j=0}^{J-1} f(i,j)\cos\left[\tfrac{(2i+1)}{2I}\right]\cos\left[\tfrac{(2j+1)}{2J}\right] \tag{4}$$

Given the DCT result of one image $f(u,v)$, its energy in frequency domain can be measured:

$$E = \sum_{u=0}^{U-1}\sum_{v=0}^{V-1} f^2(u,v) \tag{5}$$

After applying DCT to the full normalized word image, the features are extracted in a vector sequence by using the DCT coefficient set in the zigzag order. There are two ways to convert 2D DCT coefficients into a 1D vector, namely zigzag order, as illustrated in Fig. (**3**), where Fig. (**3a**) is used in our method. The matrices of the features are normalized into the range of [-1, 1]. It can be easily found that most of the DCT coefficients appear as zero or black namely energy reservation property of DCT. The number of DCT coefficients has been chosen to be 100 which is the empirically tested number for recovering the word image without perceivable degradation. The vector size of the original image is 12105. By choosing a small number of coefficients, we can find that most of the energy is concentrated as shown in Table **2**. This reflects the fact that DCT helps to reduce redundancy as quite limited coefficients are sufficient to recover the original image. As a result, the number of DCT coefficients used train the classifier is dramatically decreased.

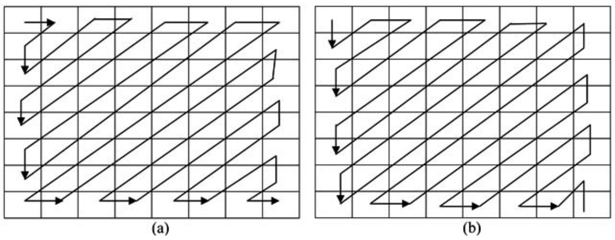

Fig. (3). Two zigzag orders in sorting 2D DCT coefficients into one vector.

Table 2. The Energy Reservation with Different Number of DCT Coefficients

Number of DCT coefficients used	Image1 Energy %	Image2 Energy %
All	100	100
1	99.16	98.91
5	99.18	98.96
10	99.20	98.96
20	99.24	98.98
100	99.78	99.17

4.2.3. The Blocked DCT Features (b-DCT)

To extract blocked DCT features, firstly each word image is divided into non overlapped vertical blocks/frames of fixed width. Then, the DCT is applied to each block. Finally, from each block the first five DCT coefficients in the zigzag order are selected to form a feature vector for the whole image. The features extracted under various block width have different performance in word classification, and the best width is suggested as 20 pixels in our experiments.

4.2.4. The Moment Invariant Features (MOM)

The Hu moment invariants are computed from the word image as described in [26]. Applying the thinning algorithm to the normalized word image, the skeleton of the word image is achieved. The invariant moments are extracted from the skeleton image. Given a digital image $f(i,j)$, its 2D moments defined as follows:

$$\eta_{pq} = \sum_i \sum_j x^p y^q f(i,j) \tag{6}$$

Knowing that parameters p and q refer to the order of the moments, the set of the invariant moments are then defined by $\phi_k \mid k \in [1,7]$ as follows:

$$\phi_1 = \eta_{20} + \eta_{02} \tag{7}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \tag{8}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \tag{9}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \tag{10}$$

$$\begin{aligned} \phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 \\ & - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\ & [3(\eta_{30} + \eta_{12}) - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{11}$$

$$\begin{aligned} \phi_6 = {} & (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + \\ & 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \end{aligned} \tag{12}$$

$$\begin{aligned} \phi_7 = {} & (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\ & + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12}) - (\eta_{21} + \eta_{03})^2] \end{aligned} \tag{13}$$

Instead of computing the moment invariant for each word image, the absolute value of the logarithm of each moment invariant is computed. The logarithm function is used to reduce the dynamic range while the absolute values are used to avoid dealing with complex numbers.

4.2.5. The Wavelet Features (DWT)

Two dimensional discrete wavelet transform (DWT) is used to extract the features, as it is well acknowledged that DWT coefficients can provide a powerful insight into an image's frequency and spatial characteristics. Given an image $f(i,j)$, its 2D DWT transform is defined as follows:

$$f(u,v,..) = \sum_{i,j} f(i,j) g_{u,v,...}(i,j) \tag{14}$$

Where i and j are the spatial variables and u, v,.. are transform domain variables.

The wavelet features are coefficients of 2-D discrete wavelet transform, as it is well acknowledged that they can provide a powerful insight into both frequency and spatial characteristics of images. In our implementation, each word image is decomposed by 4 levels of wavelet transform. At the *i*th level, each image can be decomposed into four subbands namely LL_i, LH_i, HL_i and HH_i. The LL_i sub band represents the low frequency components, and HH_i represents the high frequency components. The LH_i sub band represents the horizontal low and the vertical high frequency components, whilst HL_i represents the horizontal high and the vertical low frequency components. In order to obtain the next level in the decomposition, the LL_i sub band can be further decomposed to reach LL_4, and wavelet features used in our system are extracted from the LL_4 sub band.

4.3. Multi-class Classification

Multi-class pattern recognition has a vast range of applications, object recognition, speech recognition, text recognition, and protein classification. The application of multi-class classification in this paper focuses on unconstrained Arabic handwritten word recognition. In multi-class classification, the input features were mapped from the feature space into the output space. Classification methods vary and depend on the nature and the type of the extracted features, and the k-NN and NN classifiers are employed in our system for training and testing purposes.

4.3.1. T k-NN Classification

The k-NN is a fast supervised machine learning algorithm which is used to classify the unlabeled testing set with a labeled training set. In order to classify a word image, the features for the word image is compared to the training features based on their similarity. Then, prediction class of the testing image is found based on the minimum difference, measured by the Euclidean distance, between the testing word image and the training samples. For example, given a query instance for a word image, the K nearest instances to this query word image is the most common class.

4.3.2. The NN Classification

NN have been successfully applied to printed Arabic recognition systems [27]. The NN is a non linear system which is characterized by a network topology, which is decided by the characteristic of the neurons and the learning methodology. In our system, the multi-layer perception (MLP) back propagation (BP) neural network is used for training and classification of Arabic word images. The NN consists of three main layers: the input layer, the hidden layer, and the output layer, and its architecture is shown in Fig. (**4**). The NN is designed and trained to classify the 937 words of the Tunisian cities and villages' lexicon. The input feature vectors are defines as a matrix of input vectors. Because of the memory capacity, the target vectors cannot be defined in a 937 element vectors for training and testing. To overcome this problem, we have proposed an effective solution to represent each target vector using binary equivalent. As a result, only a 10-element vector is needed since 2^{10}=1024>937. Accordingly, we have 10 neurons in the output layer to classify and identify the words. In addition, the number of neurons in the hidden layer has been empirically chosen as 15 as it helps to generate better performance. It is worth noting that the solution above helps the NN to converge in comparison with the case without such binary representation of output classes.

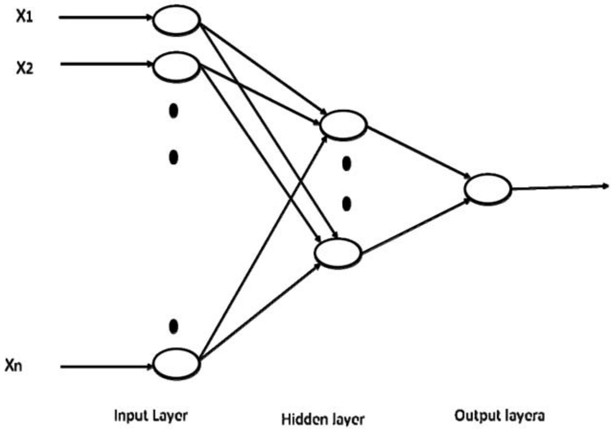

Fig. (4). The architecture of the used neural network.

5. EXPERIMENTAL RESULTS

5.1. The Test Set of IFN/ENIT

Any recognition system needs a large database to train and test the system. Real data from banks or the post code are confidential and inaccessible for non commercial research. Although some work was conducted in Arabic handwritten words, but generally they had small databases of their own or the presented results on databases which were unavailable to the public. Consequently, there was no benchmark to compare the results obtained by researches. The work for Arabic script recognition has started more than three decades ago. There was no standard database till 2002 when the IFN/ENIT database (www.ifnenit.com) becomes available free for non commercial research. This database is very important in this context as it has been used as a standard test database in such a context [5]. In total more than 1000 different people were selected as writers to put their names. Also each writer was asked to fill one or more than one form with handwritten pre-selected names of Tunisian town/villages with the corresponding postcode. All the forms were scanned with 300dpi and converted to binary images. The images are divided into five sets so that researches can use some of them for training or testing, respectively

5.2. Evaluation

In order to evaluate the performance of our recognition system, several experiments are conducted on the IFN/ENIT database which contains 32492 Arabic words handwritten by more than 1000 different writers and divided into five distinct sets a, b, c, d, and e. In our experiments, we use cross validation to verify the performance of our neural classifier. Each time 80% of the samples in the database (sets a-d) are used for training and the remaining 20% (set e) for testing.

First of all, the k-NN classifier has been used where each word image in the database was normalized to 45*269 since it is the largest size in the database and it has been divided into overlapping blocks. The block size is fixed to 12, and the overlapped block size was variable. Several experiments have been carried out by varying the size of the overlapping blocks and the results are summarized in Table **3**. As can be seen, increasing size of overlapped contents slowly degrades the recognition rate, this is due to the variation of writing and the spacing between subwords.

Table 3. Recognition Results with Size Variation of the Overlapped Block

Overlapped Block Size	Recognition Rate %
2	82.0421
3	75.9600
4	72.1415
5	71.0650
6	66.4388
7	65.1703
8	65.9236

To test the effectiveness of different features, the five groups of features are inputted separately into our k-NN classifier and NN classifier. The results from the two classifiers are reported in Table **4** and Table **5**, respectively. In both Table **4** and Table **5**, it is found that DCT features yield the best results, followed by OB and MOM features. Surprisingly, b-DCT and DWT generates very poor results which are about 15-25% worse than that of DCT. This on one hand

Table 4. k-NN Recognition Results Using Different Features

Feature Method	Recognition Rate %
OB	76.04
DCT	78.67
b-DCT	61.45
MOM	72.14
DWT	50.83

is due to the high inconsistency of handwritten images, which leads to poor frequency correspondence between features derived using spatial division. On the other hand, it suggests that the word image needs to be treated as whole when extracting frequency features.

Table 5. NN Recognition Results

Feature Method	Recognition Rate %
OB	77.75
DCT	80.75
MOM	75.75

To further compare our system with others, Table **6** depicts the recognition results of the ICDAR 2005 competition [28] where the same test set are adopted. The results of us listed in Table 6 from k-NN and NN classifiers are derived using DCT features. As can be seen, our results are the best in terms of the recognition rate, and once again it has fully validated the effectiveness of our system in multi-class classification of handwritten Arabic words.

Table 6. Performance of ICDAR 2005 Systems *vs.* our Results

System ID	Recognition Rate (%)
1	65.74
2	35.70
3	29.62
4	75.93
5	15.36
6	74.69
Proposed k-NN	78.67
Proposed NN	80.75

6. CONCLUSIONS

We have proposed a system to use k-NN and neural network for the classification of handwritten words. The performance of both k-NN and NN classifiers on Arabic handwriting word recognition are reported. Actually, both classifiers are excellent in classifying Arabic handwritten words. The system has been applied to the well-known IFN/ENIT database containing handwriting words written by different writers. We have found that DCT features are effective in our classifiers, and good results of recognition rate have been achieved. In addition, this system can be applied to other patterns for recognition with slightly adaptation. This paper, however, demonstrates that MLP networks, and *k* nearest-neighbor (k-NN) classifiers, provide good accuracy on a large handwritten word database.

Nevertheless, the simplicity of the algorithm and fast training characteristics makes the k-NN classifier an attractive candidate in handwritten word classification tasks. Further investigation is to apply other classifiers such as the radial basis function (RBF) networks, the hidden Markov models (HMM), and the Dynamic Bayesian Networks (DBN).

REFERENCES

[1] A. Amin, "Off-line Arabic character recognition: the state of the art," *Pattern Recognition*, vol. 31, pp. 517-530, 1998.

[2] L. M. Lorigo and V. Govindaraju, "Offline Arabic handwriting recognition: a survey," *IEEE Transactions on pattern Analysis and Machine Intelligence*,, vol. 28, pp. 712-724, 2006.

[3] A. Amin, "Off line Arabic character recognition: a survey," presented at 4th *International Conference on Document Analysis and Recognition*, 1997.

[4] M. S. Khorsheed, "Off-Line Arabic character recognition – A review " *Pattern Analysis & Applications*, vol. 5, pp. 31-45, 2002.

[5] M. Pechwitz, S. S. Maddouri, V. M¨argner, N. Ellouze, and H. Amiri, "IFN/ENIT - Database of Arabic Handwritten words," Presented at *Colloque International Franco-phone sur l'Ecrit et le Document (CIFED),* 2002.

[6] I. S. I. Abuhaiba, M. J. J. Holt, and S. Datta, "Processing of binary images of handwritten text documents," *Pattern Recognition*, vol. 29, pp. 1161-1177, 1996.

[7] S. Alma'adeed, C. Higgens, and D. Elliman, "Recognition of Off-line handwritten arabic words using hidden markov model approach, " *16th International Conference on Pattern Recognition (ICPR'02)* vol. 3, 2002, pp. 481-484.

[8] S. Alma'adeed, C. Higgins, and D. Elliman, "Off-line recognition of handwritten Arabic words using multiple hidden Markov models," *Knowledge-Based Systems*, vol. 17, pp. 75-79, 2004.

[9] S. Al-Ma'adeed, D. Elliman, and C. A. Higgins, "A data base for Arabic handwritten text recognition research," Presented at *8th International Workshop on Frontiers in Handwriting Recognition* 2002.

[10] M. S. Khorsheed and W. F. Clocksin, "Structural Features Of Cursive Arabic Script," Presented at the 10th *British Machine Vision Conference,* The university of Nottingham, UK, 1999.

[11] J. R. Parker, *Algorithms For Image Processing and Computer Vision* John Wiley and Sons, Inc 1997.

[12] M. S. Khorsheed and W. F. Clocksin, "Multi-font Arabic word recognition using spectral features," Presented at *Proceedings of 15th International Conference on Pattern Recognition,* 2000.

[13] M. S. Khorsheed, "Recognising handwritten Arabic manuscripts using a single hidden Markov model," *Pattern Recognition Letters*, vol. 24, pp. 2235-2242, 2003.

[14] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Communications of the ACM*, vol. 27, pp. 236 - 239 1984.

[15] R. M. Gray, "vector quantization," *IEEE Trans. ASSP*, pp. 4-29, 1989.

[16] M. S. Khorsheed, "Automatic recognition of words in arabic manuscripts " in *Computer Laboratory*, vol. P.h.D: University of Cambridge, 2000, pp. 220.

[17] M. S. Khorsheed, "Offline recognition of omnifont Arabic text using the HMM ToolKit (HTK)," *Pattern Recognition Letters*, vol. 28, pp. 1563-1571, 2007.

[18] S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK Book*: Cambridge University Engineering Department., 2001.

[19] M. Pechwitz and V. Maergner, "HMM based approach for handwritten arabic word recognition using the IFN/ENIT - database," presented at *Proceedings of 7th International Conference on Document Analysis and Recognition*, 2003.

[20] R. El-Hajj, L. Likforman-Sulem, and C. Mokbel, "Arabic handwriting recognition using baseline dependant features and hidden markov modeling," presented at *8th International Conference on Document Analysis and Recognition (ICDAR'05)* 2005.

[21] H. El Abed and V. Margner, "Comparison of different preprocessing and feature extraction methods for offline recognition of handwritten arabicwords," presented at *Document Analysis and Recognition, ICDAR Ninth International Conference on,* 2007.

[22] A. Benouareth, A. Ennaji, and M. Sellami, "HMMs with explicit state duration applied to handwritten arabic word recognition," presented at *18th International Conference on Pattern Recognition, 2006. ICPR* 2006.

[23] A. Benouareth, A. Ennaji, and M. Sellami, "Semi-continuous HMMs with explicit state duration for unconstrained Arabic word modeling and recognition," *Pattern Recognition Letters*, vol. 29, pp. 1742-1752, 2008.

[24] J. H. AlKhateeb, J. Ren, S. S. Ipson, and J. A. J. J. Jiang, "Knowledge-based baseline detection and optimal thresholding for words segmentation in efficient pre-processing of handwritten arabic text," presented at *5th International Conference onInformation Technology: New Generations,* ITNG 2008.

[25] J. Jiang, Y. Weng, and P. Li, "Dominant colour extraction in DCT domain," *Image and Vision Computing*, vol. 24, pp. 1269-1277, 2006.

[26] H. Ming-Kuei, "Visual pattern recognition by moment invariants," *Information Theory, IRE Transactions on*, vol. 8, pp. 179-187, 1962.

[27] A. Amin, "Recognition of printed arabic text based on global features and decision tree learning techniques," *Pattern Recognition*, vol. 33, pp. 1309-1323, 2000.

[28] V. Margner, M. Pechwitz, and H. E. Abed, "ICDAR 2005 Arabic handwriting recognition competition," presented at *8th International Conference on Document Analysis and Recognition,* 2005.

Received: June 02, 2009 Revised: August 10, 2009 Accepted: August 11, 2009

© AlKhateeb *et al.*; Licensee *Bentham Open*.

This is an open access article licensed under the terms of the Creative Commons Attribution Non-Commercial License (http://creativecommons.org/licenses/by-nc/3.0/) which permits unrestricted, non-commercial use, distribution and reproduction in any medium, provided the work is properly cited.