Combination of Multiple Classifiers for Off-Line Handwritten Arabic Word Recognition

Rachid Zaghdoudi and Hamid Seridi
Laboratory of Science and Information Technologies and Communication, University of 08 may 1945, Algeria

Abstract: *This study investigates the combination of different classifiers to improve Arabic handwritten word recognition. Features based on Discrete Cosine Transform (DCT) and Histogram of Oriented Gradients (HOG) are computed to represent the handwritten words. The dimensionality of the HOG features is reduced by applying Principal Component Analysis (PCA). Each set of features is separately fed to two different classifiers, Support Vector Machine (SVM) and Fuzzy K-Nearest Neighbor (FKNN) giving a total of four independent classifiers. A set of different fusion rules is applied to combine the output of the classifiers. The proposed scheme evaluated on the IFN/ENIT database of Arabic handwritten words reveal that combining the classifiers results in improved recognition rates which, in some cases, outperform the state-of-the-art recognition systems.*

Keywords: *Handwritten Arabic word recognition, Classifier combination, Support vector machine, Fuzzy K-nearest neighbor, Discrete cosine transform, Histogram of oriented gradients.*

Received September 22, 2014; accepted August 31, 2015

1. Introduction

Handwriting recognition is the process of converting images of handwriting into digital text which offers ease of storage as well as edit and search facilities. As a function of the handwriting acquisition process, its recognition is categorized either as offline or online. From the view point of recognition methodology and the way recognition engine perceives words, the recognition systems are classified into two approaches, holistic and analytical. Holistic approaches treat complete words as basic recognition units and are more suitable for problems with a limited vocabulary [18]. Analytical approaches precede by segmentation the word into smaller entities such as characters, graphemes which are then recognized. These approaches are able to handle large vocabularies [2, 4]. Segmentation of words in to meaningful recognition units, however, is a challenging problem in these approaches.

Recognition of handwriting is one of the most researched pattern classification problems [27]. Despite more than twenty years of extensive research, the problem still remains challenging, especially for recognition of handwriting in cursive scripts like Arabic [25] which in fact is the subject of the present study as well. The main challenges in recognition of Arabic handwriting arise from its highly cursive nature, non-uniform inter and intra word distances, overlapping of ligatures and words and a large number of dots and diacritics [12]. Arabic handwriting and word recognition have received significant research interest during the last two decades and a number of recognition systems reporting high recognition rates have been proposed. Despite these developments, the problem still remains open to research due to the aforementioned challenges.

The recent research on Arabic handwriting recognition mainly aims to improve the feature extraction or/and classification techniques to enhance the overall recognition rates. Typical features applied to handwriting recognition include statistical features [16], structural features [17] and global transformations [24]. For classification, state-of-the-art classifiers including Artificial Neural Networks (ANN) [6], Support Vector Machine (SVM) [3, 22], and Hidden Markov Models (HMM) [8, 15] have been extensively applied to Arabic handwriting/word recognition. In some cases, classifiers are combined to improve the overall recognition rates [1, 5, 11, 28]. The Combination of classifiers has received significant research attention in the recent years for improving the recognition rates of different pattern classification problems and same is the case with handwriting recognition in general and Arabic word recognition in particular. Classifiers can be combined under various topologies including a serial, parallel or hybrid combination. In a serial combination of classifiers, the output of one classifier is used as an input to the next classifier in a cascaded form. The main issue with this combination is that errors introduced by a classifier cannot be recovered by the next classifiers. In parallel combination of classifiers, the output of multiple classifiers is combined using a function to make the final decision on the class of the query object. In general, the parallel combination of classifiers is known to achieve better classification results than any of the single classifiers. Hybrid classifiers combine both serial and parallel classifiers, but suffer from the already discussed weaknesses of serial classifiers.

This work explores different combinations of classifiers applied to offline handwritten Arabic word recognition system with the objective of improving the overall recognition rates. Each word is represented by a set of features based on the Discrete Cosine Transform (DCT) and Histogram of Oriented Gradients (HOG). In addition, Principal Component Analysis (PCA) is also applied on the HOG features to reduce the dimensionality of the feature space. Each feature set is fed to two classifiers, SVM and Fuzzy K-Nearest Classifier (FKNN) making a set of four classifiers. Finally, the output of the classifiers is combined using a number of rules to have the final decision. Evaluations carried out on the IFN/ENIT database [29] realized high word recognition rates which are comparable, and in some cases, better than those of the state-of-the art methods on Arabic handwritten word recognition. The main contribution of this study is the investigation of different schemes of classifier combination to improve the Arabic word recognition rate.

The main steps involved in the proposed system are summarized in Figure 1 while each of these steps is discussed in detail in the following sections.

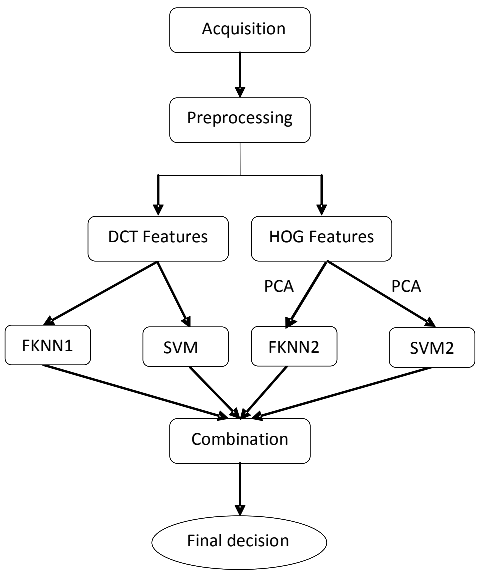

Figure 1. Overview of the proposed system.

2. Preprocessing

Preprocessing is carried out on the word image to remove noise introduced during the acquisition step, eliminate writer-dependent variations and irregularities and represent the image in a form that is appropriate for extraction of features. Typical preprocessing tasks for handwriting recognition and similar problems include binarization, slope and slant correction [7], normalization and Skeletonization [34]. Since the images in the IFN/ENIT database are already binarized, the preprocessing in our case comprises slope and slant correction to eliminate the writing style dependent variations, Skeletonization to reduce the writing instrument dependency of the computed features and the word images are normalized experimentally to a predefined size of 100*400 pixels. The implementation details of the preprocessing steps can be found in [19]. Figure 2 illustrates an original and the respective preprocessed image. Once the word image is preprocessed, we proceed to the next step of feature extraction as discussed in the following section.

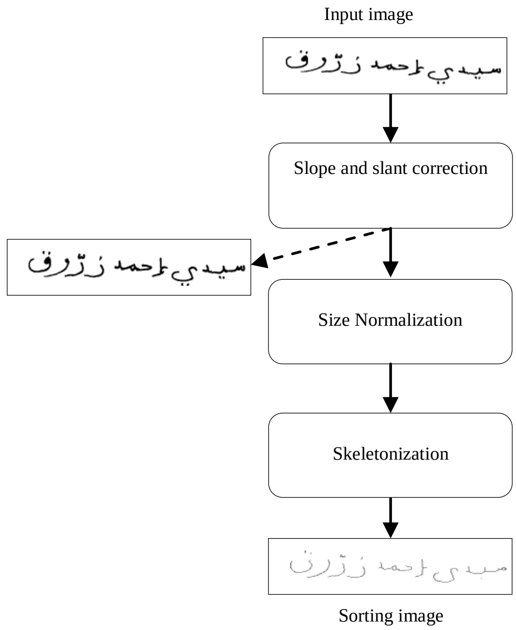

Figure 2. Preprocessing of a word image.

3. Feature Extraction

Feature extraction is the most important component of any pattern recognition system which aims to find an appropriate representation of the pattern under study. For our problem of Arabic word recognition, we have chosen to represent the skeletonized word images by statistical features for which rich classifiers are known to exist. For each word image, we extract a set of features based on the Discrete Cosine Transform (DCT) and the Histogram of Gradients (HoG). Each of these features is discussed in the following sections.

3.1. Discrete Cosine Transform

The Two-Dimensional Discrete Cosine Transform (2D-DCT) converts a signal into elementary components where each component has its own amplitude and frequency [30]. Applying 2D-DCT to a rectangular matrix representing a normalized word image result in a matrix (of DCT coefficients) of the same size as the word image with most of the useful information concentrated in a few coefficients (upper left part of the output matrix). These DCT coefficients are computed as presented in Equation 1.

$$F(u,v)=C(u)C(v)[\sum_{x=0}^{(M-1)}\sum_{y=0}^{(N-1)} f(x,y)cos\frac{(2x+1)u\pi}{2M}cos\frac{(2y+1)v\pi}{2N}] \quad (1)$$

Where $f(x, y)$ is the input image, M denoted the number of lines, N: number of columns, and:

$$C(u)=\frac{1}{\sqrt{M}}\ if\ u=0\ or\ \sqrt{\frac{2}{M}}\ if\ 1\leq u\leq M-1 \quad (2)$$

$$C(v)=\frac{1}{\sqrt{N}}\ if\ v=0\ or\ \sqrt{\frac{2}{N}}\ if\ 1\leq v\leq N-1 \quad (3)$$

The matrix of DCT coefficients is mapped to a feature vector by scanning the matrix elements in a zig-zag fashion. This arranges the coefficients in a one dimensional vector in such a way that the low-frequency coefficients containing relevant information are at the beginning of the vector which are useful in recovering the original image.

3.2. Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) is a gradient based feature descriptor that was originally proposed Dalal and Triggs for detection of humans [14]. Since then, it has been widely employed to a number of computer vision problems with varying degrees of success. Recently, HOG based features have been applied for recognition of handwritten words and characters and have shown promising recognition rates [20, 32]. The basic idea behind HOG is that the shape of objects in an image can be characterized by the distribution of local intensity gradients representing the dominant edge directions. To compute the distribution of local gradients, the image is divided into cells and the gradient of each pixel in a cell is calculated to form a histogram of gradients for that cell. The following steps outline the computational details of the HOG descriptor [10].

- *Step* 1. The horizontal and vertical gradient of the image is computed by convolving the image with the respective gradient masks ([−1, 0, 1] and $[-1, 0, 1]^T$).
- *Step* 2. *The* strength and orientation of the gradient are computed using Equations 4 and 5.

$$SG=\sqrt{G_h(x,y)^2+G_v(x,y)^2} \quad (4)$$

$$OG=arctan\frac{G_h(x,y)}{G_v(x,y)} \quad (5)$$

Where: G_h and G_v denote the horizontal and vertical gradient; SG and OG represent the strength and orientation respectively at point (x, y) in the image.
Step3. The image is divided into N*N cells and a histogram of orientations is computed for each cell. If the histogram is divided into k bins based on the orientation, the value of the i^{th} bin V_i for cell C is computed as follows.

$$V_i=\sum_{(x,y)\in C} SG(x,y)/OG(x,y)\in bin_i \quad (6)$$

- *Step* 4. The histogram of each cell is normalized by L2 Norm.
- *Step* 5. The histograms of all cells are concatenated to form the descriptor.

The dimensionality of the descriptor is a function of the cell size N and the number of bins k in the histogram [26]. In our implementation, we divide each word image into cells that vary from 3*3 to 12*12 while the histogram ([0, π]) is evenly partitioned into k bins. The final descriptor has a size of N*N*K which in our case varies from 81 to 1296 for different evaluation scenarios. For comparison purposes, we also apply the Principal Component Analysis (PCA) on the HOG descriptor to reduce the dimensionality of the feature vector.

4. Classification

Classification is carried out to determine the class of the query word image and hence recognize it. Features extracted from the training examples (words in our case) are used to train the classifier to learn to discriminate between different classes. Features of the query sample are then fed to the trained classifier to find the output class label. In our study, we aim to investigate the different combinations of classifiers and analyze their respective performance on word recognition. We have chosen a SVM and F-KNN as classifiers which are combined using different combination heuristics. The DCT and HOG based features computed from the training word images are fed (separately) to each of the classifiers making a total of four classifiers. In the following sections we briefly discuss each of the classifiers (SVM and F-KNN) and later present the combination rules.

4.1. Multi-Class SVM Classifier

SVM is a learning algorithm that has been successfully applied to a wide variety of regression and classification problems. The robustness, accuracy and the greater generalization ability of SVM make them an attractive choice in many machine learning tasks. Support vector machine is based on the structural risk minimization principle of the statistical learning theory [13]. The aim of SVM is to map the input data to a higher dimensional space by using kernel functions and find the optimal hyper-plane in the space that maximizes the margin between classes. A special attribute of SVM is its ability to simultaneously minimize the empirical classification error and maximize the geometric margin [31]. The original SVM was designed to be a binary classifier as presented in Equation 7 and was later extended to solve multi-class problems as well. A multi-class classifier is produced by combining several binary classifiers generally following one-against-one or one-against-all strategies.

For a given input *x*, the decision function of an SVM binary classifier is given by the following equation.

$$f(x) = sign(\sum_{i=1}^{n} y_i \alpha_i k(x, x_i) + b) \quad (7)$$

Where x_i represent the input training feature vector and $y_i \in [-1,1]$ the respective output label; b represents the bias, α_i is the Lagrange multiplier while $k(x,x_i)$ corresponds to the kernel function.

In our study, we employ the Radial Basis Function (RBF) kernel with one-against-one strategy to implement a multi-class SVM. This requires training N*(N-1)/2 binary SVMs each trained on features from training examples of two classes. A query pattern is classified according the "Max Wins" voting strategy. For implementation, we have employed the LIBSVM 3.17 [9] while the values of the kernel parameter (γ) and the soft margin (C) are determined empirically.

4.2. Fuzzy K-Nearest Neighbor Classifier

Keller proposed a fuzzy version [21] of the classical k-nearest neighbor algorithm by introducing the concept of fuzzy set theory in the original KNN algorithm. The fuzzy k-nearest neighbor classification or F-KNN assigns, to each pattern, a degree of membership as a function of its distance from its k-nearest neighbors. For a given sample *x*, its degree of membership to each class is given by the following equation.

$$u_i(x) = \frac{\sum_{j=1}^{k} u_{ij} \left[1/ \|x - x_j\| \right]^{\frac{2}{m-1}}}{\sum_{j=1}^{k} \left[1/ \|x - x_j\| \right]^{\frac{2}{m-1}}} \quad (8)$$

$u_i(x)$ Gives the degree of membership of pattern (word) *x* to class *i* where *i* varies from 1 to *c* (the number of classes) and *k* represents the number of neighbors considered by the algorithm. The term $\|x - x_j\|$ represents the Euclidean distance between x and the neighbor x_j. The term $u_i(x)$ is 1 if $x \in C_i$ and is 0 otherwise. The parameter m is the fuzzifier that determines how the membership values vary with respect to the distance from the (nearest) neighbors and is commonly fixed to 2 [21]. The class label which the highest membership value is assigned to the query sample x.

4.3. Combination of Classifiers

The objective of classifier combination is to combine the output of multiple classifiers (four in our case) to realize improved classification rates as compared to any of the individual classifiers. The way different classifiers are combined is a function of the type of classifiers outputs [33]. This combination can be carried out at three levels, the abstract level, rank level and measurement level. In the abstract level fusion, each classifier produces a unique class label for each input pattern and the final decision is based on predefined heuristics. For rank level combination, each classifier produces a ranked list of class labels for an input sample while for measurement level fusion; each classifier generates a score for a query sample. In our study, depending upon the combination scheme, we have employed abstract level and measurement level fusions. The different classifier combination schemes investigated in our study include majority vote, minimum rule, maximum rule, sum rule, average rule, product rule, decision template, Bayesian method and Dempster-Shafer rule [23]. Each of these combination rules is briefly discussed in the following:

- Majority voting: Assigns the class label on which the majority of the classifiers agrees.
- Minimum rule: Selects the classifier with minimum objection and assigns the respective class label to the query pattern.
- Maximum rule: Selects the classifier with a maximum score (confidence) and assigns the respective class label to the input pattern.
- Sum rule: Sums up the scores of individual classifiers for each class and assigns the label with the highest aggregate score.
- Average rule: Takes the average of the scores of each class and picks the class label with the highest average.
- Product rule: Multiplies the scores of each class and selects the class with the highest product.
- Decision Template: A decision template is created for each class and the query pattern is classified by comparing its decision profile with the decision templates of each class using a similarity measure.
- Dempster-Shafer rule: Inspired by the Dempster-Shafer (DS) theory of evidence, this rule employs decision templates with a degree of belief rather than a similarity measure.
- Bayesian: Assumes that the classifiers are mutually independent and combines the outputs at the abstract level by using the confusion matrices of the member classifiers.

5. Experimental Results

The performance of the proposed recognition system is evaluated by conducting a series of experiments on the well-known IFN/ENIT database [19] comprising 26,459 handwritten words of 946 Tunisian town/village names. The writing samples are contributed by 411 different writers and the database is divided into four subsets 'a', 'b', 'c' and 'd'. Three subsets 'a', 'b' and 'c' are used for training and the subset 'd' is used for testing. For all experiments, each word image is normalized experimentally to a predefined size of 100*400 pixel. Sample words from the database are illustrated in Figure 3.

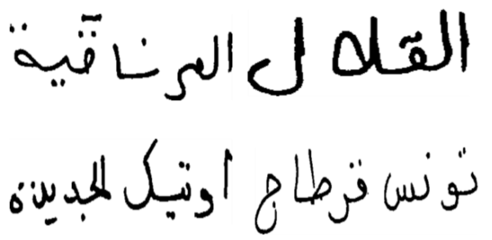

Figure 3. Sample words from the IFN/ENIT database.

We first present the results of the experiments on individual classifiers followed by the results on different combinations of these classifiers. Finally, we present a comparison of the results achieved with the proposed combination schemes with state-of-the-art methods on the subject.

5.1. Performance of Individual Classifiers

As discussed earlier, the DCT and HOG based features are separately fed to two different classifiers, SVM and FKNN. The results are reported with varying the number of DCT coefficients in case of DCT based features and varying the number of cells (N) in case of HOG based features. The tuning parameters for both classifiers were determined experimentally. The SVM classifier has two parameters the penalty (C) and the RBF kernel parameter (γ). The values for these parameters were found through a grid search using 5-fold cross-validation. Different values of (C, γ) were tried and the one with the best 5-fold cross-validation accuracy is selected. FKNN classifier has a single parameter the number of nearest neighbors (k) which is also found experimentally.

Table 1 summarizes the recognition rates realized by the two classifiers for different number of DCT coefficients. The highest recognition rates achieved stand at 90.43% and 83.81% for SVM and FKNN respectively. It is interesting to note that both the classifiers achieve best performance at nearly the same number of DCT coefficients (61 and 60 respectively for SVM and FKNN). In all cases, SVM reports better recognition rates than FKNN.

Table 1. Recognition rates as a function of the number of DCT coefficients.

DCT Coefficients	F-KNN (k=5)	SVM (c=100, γ=0.05)
400	66.37	85.21
100	80.88	89.95
90	82.41	90.15
80	82.53	90.10
70	82.67	90.10
65	83.07	90.36
61	83.48	90.43
60	83.81	90.24
55	83.30	90.13
50	52.35	89.36
40	81.32	88.19

The HOG based features are computed by adopting the implementation presented in [26]. The computation of these features requires setting of two important parameters, the number of cells per bounding box (N) and the number of orientation bins (K) producing a descriptor of dimension N*N*K For experiments, we consider a number of configurations for the parameters (N, K). In addition, Principal Component Analysis (PCA) is also applied to the HOG features to reduce the dimensionality of the descriptor. The number of principal components is determined empirically after many tests. Table 2 summarizes the recognition rates of the two classifiers by computing the HOG features for different values of N, fixing the value of K to 9.

Table 2. Recognition rates as a function of number of cells (N).

Hog features (N*N*K)	HOG-PCA	F-KKN (k=7)	SVM (C=10, γ=0.02)
3*3*9	81	46.27	69.00
4*4*9	100	60.58	78.84
5*5*9	150	70.78	84.93
6*6*9	240	81.83	90.72
7*7*9	270	85.95	92.80
8*8*9	290	86.27	93.19
9*9*9	320	87.94	93.65
10*10*9	380	89.90	94.24
11*11*9	540	90.18	93.93
12*12*9	580	88.93	92.60

It can be observed from Table 2 that in general, the classification accuracies increase with the increase in the number of cells. The highest recognition rates realized are 94.24% for SVM and 90.18% for FKNN at cell sizes of 10*10 and 11*11 respectively.

The impact of the number of orientation bins in the HOG features on the overall recognition rates is studied by fixing the cell size N to 10 and varying the number of bins K. The results of these evaluations are presented in Table 3.

Table 3. Recognition rates as a function of number of orientation bins (K).

Hog features (N*N*K)	HOG-PCA	F-KKN (k=7)	SVM (C=100, γ=0.02)
10*10*4	240	77.03	82.23
10*10*5	270	87.48	92.68
10*10*6	320	88.24	93.23
10*10*7	340	90.10	94.24
10*10*8	370	89.60	94.33
10*10*9	380	89.90	94.24
10*10*10	490	90.13	94.26
10*10*12	570	89.67	94.21
10*10*16	630	89.69	94.21
10*10*18	720	89.67	93.23

It can be seen that increasing the number of bins (K) results in improving the recognition rates. However, beyond 8-10 bins, the recognition rates stabilize and do not show significant variations. Recognition rates of as high as 94.33% and 90.13% are achieved with SVM and FKNN respectively.

Figure 4 summarizes the highest recognition rates achieved with the DCT and HOG features using SVM and F-KNN. Among the two feature types, HOG based features outperform the DCT features for each of the classifiers. Comparing the two classifiers, SVM achieve better recognition rates on both the features. Overall, the highest recognition rate of 94.33% is achieved using HOG features and SVM as a classifier.

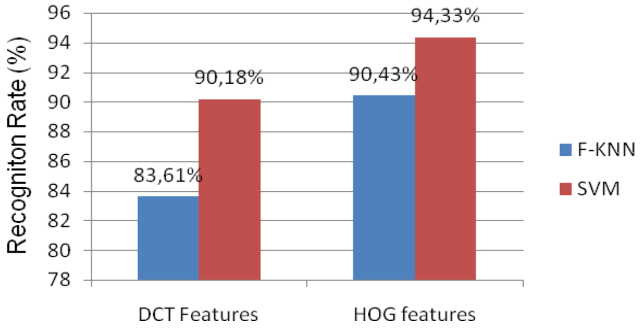

Figure 4. Highest recognition rates of the two classifiers.

5.2. Performance of the Ensemble Classifiers

After having discussed the performance of individual classifiers, we now present the results of combining the outputs of the four classifiers. The classifiers are combined using the majority vote, minimum rule, maximum rule, sum rule, average rule, product rule, decision template, Bayesian method and Dempster-Shafer rule as presented in Section 4.3. The recognition rates of these combinations are summarized in Figure 5.

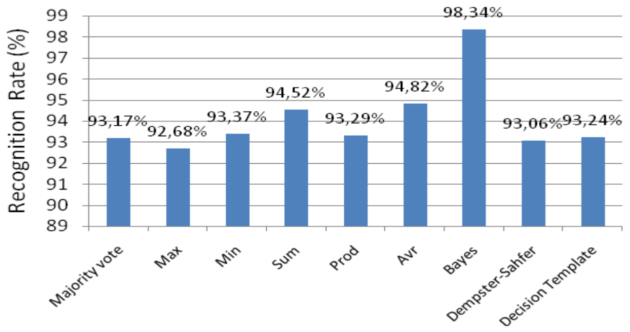

Figure 5. Recognition rates on classifier combinations.

It can be observed from Figure 5 that the classifier combination using sum, average and Baye's method achieve better recognition rates than the highest recognition rate among the individual classifiers.

The most significant improvement can be seen in case of Baye's method which realizes a recognition rate of 98.34% as opposed to 94.33% achieved with HOG features and SVM classifier. It should be noted that a recognition rate of 97.65 % is achieved over the distorted handwritten words. Which shows the great capacity of the propped system, to recognize the handwritten words, even when they are badly written.

5.3. Performance Comparison

We also present a comparative analysis of the proposed recognition system with the state-of-the-art Arabic word recognition systems presented in the literature.

Table 4 presents a comparative overview of different systems evaluated on the same (IFN/ENIT) database using the same evaluation protocol as the one used in our study. It can be seen that the proposed combination of classifiers achieves a maximum recognition rate of 98.34%, better than any of the existing systems validating the ideas put forward in this study.

Table 4. Comparison of recognition rates.

System	Features	Classifier	Accuracy (%)
Abdel Azeem [1]	Gradient and concavity features	Fusion of 3HMMs	97.70
Alalshekmubarak [3]	Zone based features	SVM	92.34
Al-Hajj [5]	Distribution and concavity features	Combination of HMMs	90.96
Alkhateeb [6]	DCT	Neural Network	80.75
Benouareth [8]	Structural and statistical features	HMM	83.79
L. Chergui [11]	Hu, Zernike and Tchebichef moments	Combination of Neural Networks	90.10
El Abed [15]	Skeleton directions and local features	HMM	89.10
Khalifa [22]	DCT	SVM	91.70
Proposed Method	DCT+HOG	Combination of SVM and F-KNN	98.34

6. Conclusions

This study presented an Arabic handwritten word recognition system based on fusion of classifiers. Features based on histogram of gradients and discrete cosine transform extracted from the handwritten words were fed to two different classifiers, SVM and F-KNN giving a total of four different classifiers. The proposed recognition system was evaluated on the widely used IFN/ENIT database of Arabic handwritten words. Among the individual classifiers, SVM achieved the highest recognition rate of 94.33% using HOG features. The four classifiers were then combined using different combination schemes, including majority vote, minimum rule, maximum rule, sum rule, average rule, product rule, decision template, Baye's method and Dempster-Shafer rule.

Evaluations on these combinations revealed that classifier combination using Bayesian rule achieves a word recognition rate of 98.34%, about 4% better than the best recognition rate of the individual classifiers. A comparison of the proposed scheme with state-of-the-art Arabic word recognition systems revealed that the suggested combination outperforms existing methods in terms of recognition rates.

In our further study on the subject, we intend to employ an enhanced feature set and an increase number of classifiers. We also plan to implement a feature/classifier selection mechanism to come up with the optimal set of features and classifiers for this problem.

Acknowledgements

The authors would like to thank Dr Djedi Chawki for his useful comments that helped them to improve the content and the presentation of the paper.

References

[1] Abdel-Azeem S. and Ahmed H., "Effective Technique for the Recognition of Offline Arabic Handwritten Words using Hidden Markov Models," *International Journal on Document Analysis and Recognition*, vol. 16, no. 4, pp. 399-412, 2013.

[2] Abdulla S., Al-Nassiri A., and Abdul-Salam R., "Off-Line Arabic Handwritten Word Segmentation Using Rotational Invariant Segments Features," *The international Arab Journal of Information Technology*, vol. 5, no. 2, pp. 200-208, 2008.

[3] Alalshekmubarak A., Hussain A., and Wan Q., *Neural Information Processing*, Springer Link, 2012.

[4] Alginahi Y., "A Survey on Arabic Character Segmentation," *International Journal on Document Analysis and Recognition*, vol. 16, no. 2, pp. 105-126, 2013.

[5] Al-Hajj R., Likforman-Sulem L., and Mokbel C., "Combining Slanted-Frame Classifiers for Improved HMM-based Arabic Handwriting Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 7, pp. 1165-1177, 2009.

[6] AlKhateeb J., Jiang J., Ren J., Khelifi F., and Ipson S., "Multiclass. Classification of Unconstrained Handwritten Arabic Words Using Machine Learning Approaches," *The Open Signal Processing Journal*, vol. 2, pp. 21-28, 2009.

[7] Al-Rashaideh H., "Preprocessing phase for Arabic Word Handwritten Recognition," *Information Transmissions in Computer Networks*, vol. 6, no. 1, pp. 11-19, 2006.

[8] Benouareth A., Ennaji A., and Sellami M., "Semi-continuous HMMs with Explicit State Duration for Unconstrained Arabic Word Modeling and Recognition," *Pattern Recognition Letters*, vol. 29, no. 12, pp. 1742-1752, 2008.

[9] Chang C. and Lin C., "LIBSVM: A Library for Support Vector Machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1-39, 2011.

[10] Chen S. and Liu C., *Computer Analysis of Images and Patterns*, Springer Link, 2011.

[11] Chergui L., kef M., and Chikhi S., "Combining Neural Networks for Arabic Handwritten Recognition," *The international Arab Journal of Information Technology*, vol. 9, no. 6, pp. 588-595, 2012.

[12] Cheriet M., *Lecture Notes in Computer Science*, Springer Link, 2008.

[13] Cortes C. and Vapnik V., *Support-vector network Machine Learning*, Springer Link, 1995.

[14] Dalal N. and Triggs B., "Histograms of Oriented Gradients for Human Detection," *in Proceeding of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, California, 2005.

[15] El-Abed H. and Märgner V., "Comparison of Different Preprocessing and Feature Extraction Methods for Offline Recognition of Hand-Written Arabic Words," *in Proceeding of 9th International Conference on Document*, Parana, 2007.

[16] El-Feghi L., Elmahjoub F., Alswady B., and Baiou A., "Offline Handwritten Arabic Words Recognition using Zernike Moments and Hidden Markov Models," *in Proceeding of International Conference on Computer Applications and Industrial Electronics*, Kuala Lumpur, pp. 165 - 168, 2010.

[17] El-Melegy M. and Abdelbaset A., "Global features for offline recognition of handwritten Arabic literal amounts," *in proceeding of 5th International Conference on Information and Communications Technology*, Cairo, pp. 125-129 2007.

[18] Farah L., Farah N., Bedda M., "Arabic word Recognition by fuzzy classifier," *Journal of applied sciences*, vol. 6, no. 3, pp. 647-650, 2006.

[19] Farou B., Hallaci S., and Seridi H., "Système Neuro-Markovien Pour la Reconnaissance de L'écriture Manuscrite arabe à Vocabulaire Limité," *in Proceeding of 2nd Conférence Internationale sur l'informatique et ses Applications*, Saida, pp. 1-11, 2009.

[20] Iamsa S. and Horata P., "Handwritten Character Recognition Using Histograms of Oriented Gradient Features in Deep Learning of Artificial Neural Network," *in Proceeding of International Conference on IT Convergence and Security*, Macao, pp. 1-5, 2013.

[21] Keller J., Gray M., and Givens J., "A Fuzzy K Nearest Neighbor Algorithm," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 15, no. 4, pp. 580-585, 1985.

[22] Khalifa M. and BingRu Y., *Advanced Research on Electronic Commerce Web Application, and Communication, Communications in Computer and Information Science* , Springer Link, 2011.

[23] Kuncheva L., *Combining Pattern Classifiers: Methods and Algorithms*, Wiley-Interscience, 2004.

[24] Lawgali A., Bouridane A., Angelova M., and Ghassemlooy Z., "Handwritten Arabic Character Recognition: Which Feature Extraction Method?," *International Journal of Advanced Science and Technology*, vol. 34, pp. 1-8, 2011.

[25] Lorigo L. and Govindaraju V., "offline Arabic Handwriting Recognition: a Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 5, pp.712-724, 2006.

[26] Ludwig O., Delgado D., Goncalves V., Nunes U., "Trainable Classifier-Fusion *Schemes: An Application to Pedestrian Detection," Proc. 12th International IEEE Conference on Intelligent Transportation Systems*, St. Louis, vol. 1, pp. 432-437, 2009.

[27] Mirvaziri H., Javidi M., and Mansouri N., *Visual Informatics: Bridging Research and Practice*, Springer Link, 2009.

[28] Nemouchi S., Meslati L., and Farah N., *Image and Signal Processing, Lecture Notes in Computer Science*, Springer Link, 2012.

[29] Pechwitz M., Maddouri S., Märgner V., Ellouze N., and Amiri H., "IFN/ENI-Database of Arabic Handwritten Words," *in Proceeding of 7th Colloque International Francophone sur l'Ecrit et le Document*, Hammamet, pp. 129-136, 2002.

[30] Rao K. and Yip P., *Cosine Transform: Algorithms, Advantages, Applications*, Academic Press, 1990.

[31] Srivastava D. and Bhambhu L., "Data Classification Using Support Vector Machine," *Journal of Theoretical and Applied Information Technology*, vol. 12, no. 1, pp. 1-7, 2010.

[32] Williams K., Suleman H., and Jorgina K., "A Comparison of Machine Learning Techniques for Handwritten |Xam Word Recognition," *in Proceeding of the South African Institute for Computer Scientists and Information Technologists Conference*, East London, pp. 37-46, 2013.

[33] Xu L., Krzyzak A., and Y-Suen C., "Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418-435, 1992.

[34] Zhang T. and Suen C., "A Fast Parallel Algorithm for Thinning Digital Patterns," *Communication of the ACM*, vol. 23, No. 3, pp. 236-239, 1984.

Rachid Zaghdoudi received his MSc degree in Electronics from University of 08Mai 1945, Guelma, Algeria, in 2008. Currently, he is a researcher in the laboratory of Science and Information Technologies and Communication "LabSTIC", University of 8 Mai 1945 of Guelma and member in GADM team at LabSTIC laboratory, Guelma, Algeria. His current research interests include Handwriting recognition, Classification, artificial Intelligence, pattern recognition, and document processing.

Hamid Seridi received his Bachelor's degree with honours in 1981, from the University of Annaba, Algeria, and the Master's degree from the Polytechnic Institute of New-York, USA in 1984, both in Electrical Engineering. He received his PhD in Computer Science with distinction in 2001 from the University of Reims, France. He was Vice Dean of the Post-Graduation, Scientific Research and External Relations in the University of Guelma. Currently he is Professor and Director of Laboratory of Science and Information Technologies and Communication "LabSTIC". He is also Chairman of the Scientific Council of the Faculty of Mathematics and Computing and Material Sciences. He is an expert member at the national committee for evaluation and accreditation national projects research. His research interests include approximate knowledge management, pattern recognition and artificial intelligence, data mining, video mining, machine learning and cryptography.