Combine SVM and KNN classifiers for Handwriting Arabic Word Recognition based on Multifeatures

Assist. Prof. Dr. Alia Karim Abdul Hassan
hassanalia2000@yahoo.com
University of Technology - Computer science
Baghdad, Iraq
Mohammed Alawi
alawialawi3000@gmail.com
University of Technology - Computer science
Baghdad, Iraq

Abstract: *This paper presents a proposed system for recognizing the handwritten Arabic words. The proposed system recognized the Arabic word as one entity without using segmentation stage, which converted the word into parts. A proposed method for feature extraction stage used two groups of feature extraction techniques. First group combines two techniques and the second group used single technique. First group combines gradient (directional) feature method with the Run Length Count method and second group based on Discrete Cosine Translation technique. Classification stage is based on combined SVM with KNN classifiers. A standard data set which is AHDB database is used to simulate the proposed system. The recognition accuracy for the experimental results of the proposed system is 97.11 %.*

Keywords: Handwriting recognition, gradient Feature, SVM, KNN.

1. Introduction

Handwriting recognition (HR) is the process of converting the handwriting text images into a text file that recognizable by the computer and used for many purposes [1]. Many applications for recognition of Arabic handwritten words such as in bank check reading, mail sorting, office automation, and form processing in administration & insurance [2]. Identification of Handwritten word is a difficult process because each writer has a unique style of writing with a different control over writing [3]. There are two approaches of handwriting recognition systems for Arabic scripts, segmented based and segmentation free based. In the first method, divide each word image into characters, then recognizes each character in the word. This is called an analytical -approach. While, the second type treat and recognize the word entity, based on the shape, structure and other features of the word [4]. The proposed system consists of three stages which are: pre-processing, feature-extraction and classification / recognition. The pre-processing stage, which tries to reduce the noise data and keep only the desired information and make the next operation (feature-extraction process) easy to implement. Moreover, the second stage is feature extraction which is the process of extracting the best information from the binary handwriting word image to be used in recognition stage. Support Vector Machine (SVM) and KNN classifiers used in classification and recognition stage. They classified all samples into different classes, then recognized the query word image to desired class.

2. Related Work

The most recent works have been done for the Arabic language. In [5] introduced a handwriting Arabic recognition system using a handwriting database of Algerian city names. The feature extraction which used is namely Freeman code is determined by the contour of the image, Zernike moments, and structural features. In classification phase use multiple classifiers such as K-Means algorithm, Probabilistic Neural Network (PNN), Fuzzy C-Means algorithm (FCM) and K Nearest Neighbor algorithm (KNN). The recognition rate for this system was 80%. In [4] this system the input words are segmented and normalized in size. The extracted features by DCT from each word sample are then utilized to train a neural network for classification. They have been successfully tested their system on IFN/ENIT Database, by using only a small set of 500 word experiments. The Recognition rate was 82.5. In [6] a recognition Arabic words system uses DCT and Histogram of Oriented Gradient HOG techniques as feature-extraction methods and the SVM classifier for classification and recognition data. The proposed system achieved a recognition rate 96.317%. In [7] the seven global features such as the number of loops, the position of ascenders and descanters, the number of segments, lower dots and upper dots are used in the proposed Arabic word handwritten system. The artificial NN classifier which used to classify the Arabic words. Proposed system tested on AHDB database and Recognition results was 63%. In [8] the proposed Arabic handwritten words used global feature (seven features), the number of loops, the position of ascenders and descanters, the number of lower dots, upper dots, and the number of segments for feature extraction. Rule-based classifier is used as a global recognition engine to classified training data into groups, after that for each group, the HMM approach is used for classification. By training and testing the system on the AHDB dataset, the system has obtained a near 60% recognition accuracy.

3. Arabic Handwritten Database (AHDB) database [9]

The AHDB database contains Arabic Words and texts written by a hundred different writers This database provides a training and testing set for Arabic text recognition research. There is some useful preprocessing operation carried out on the AHDB database contains. Words and texts in Arabic languages written by 100 writers and these words used for the quantities and numbers on checks filling. And it has the most famous words in Arabic writing. Figure 1 illustrated word images of the AHDB database.

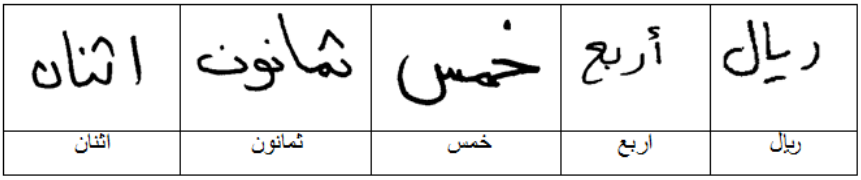


Figure 1: Examples of AHDB databas

4. Basic Concepts and Definitions:

4.1. Clipping method

For clipping the handwritten word used Bounding Box. Bounding box is determined through finding out 4 points for each direction (up, left, down, right). These points are the first black point in each direction, and then determined the boundary box [10]. Algorithm 1 explains the clipping process.

Algorithm 1: clipping step **Input**: Binary image white space around the object **Output**: clipping Binary image
Step1: Read the binary image. **Step2**: P_left = first black pixel founded by scanning the input image from left to right. **Step3**: P_right= first black pixel founded by scanning the input image from right to left. **Step4**: P_up = first black pixel founded by scanning the input image from up to down. **Step5**: P_bottom= first black pixel founded by scanning the input image from down to up. **Step6**: The new width and height of the clipped image: Width = P_right – P_left + 1 Height = P_bottom – P_up + 1 So, the size of the clipped image is (width, height). **Step6**: return clipping image

4.2. Normalization:

The size of handwritten words is variable from person to person and even with the same person at different times therefore Size normalization is required to make all images have the same size. The aim of size normalization to extract the features under the same sizes, and make the recognition process has more accuracy [11].

4.3. Feature extraction

Feature extraction is a process that converted the input data into the set of features. [12] These features have the essential information of the word image which make it different from other words; there are many of Feature extraction methods. That used in differing applications. Some of them may be succeed in one application and Fail in another application. The selected feature extraction method is an important step in achieving a high recognition rate. [11] in next paragraph illustrated three types of feature extraction methods.

4.4. Gradient direction feature

It is used To extract gradient feature, applied Prewitt operator [13] on word image to get two gradient components: strength f(x,y) and direction θ (x,y) for each point (x,y) of the image in horizontal and vertical directions, respectively. Figure 2 shows the Prewitt filter masks [13].

-1	-1	-1
0	0	0
1	1	1

Gx

-1	0	1
-1	0	1
-1	0	1

Gy

Figure 2: Prewitt filter masks for extracting gradient features

Each pixel I(x,y) gradient-strength and direction were calculated using equations 1 and 2 [14]:

$$f(x,y) = \text{squrt}\ ({G_x}^2 + {G_Y}^2) \tag{1}$$

$$\theta(x,y) = \tan^{-1}(G_x/G_y) \tag{2}$$

In equation 2, θ(x,y) gives the direction of a vector (G_x,G_y). θ(x,y) ∊ [-90,90].

4.5. Discrete Cosine Transform (DCT)

DCT is used to convert the image data in the spatial domain into its elementary frequency components in the frequency domain [15]. By DCT can be convert the energy of the image into a few coefficients, these coefficients groups in 2 dimensional array where the coefficients of high value in the upper left corner and coefficients with low values in the bottom right corner [4]. Using DCT coefficients as features which becomes efficient in many recognition problems [16]. When the image p(x,y), the DCT coefficients D(i,j) are computed by equation 3 :

$$D(i,j) = \frac{1}{\sqrt{2N}} \quad C(i)C(j)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} p(x,y)cos\left[\frac{(2x+1)i\pi}{2N}\right] \quad cos\left[\frac{(2y+1)j\pi}{2N}\right] \tag{3}$$

$$C(u) = \begin{Bmatrix} \frac{1}{\sqrt{2}} & if\ u = 0 \\ 1 & if\ u > 0 \end{Bmatrix} \tag{4}$$

Where $p(x,y)$ is the image element, N is the block size of DCT, DCT coefficients is d(i,j).

Two ways are used to convert the DCT coefficients into a one dimensional vector namely zigzag order. Figure 2 illustrated these types, where Figure. 3(a) is used in our method [17].

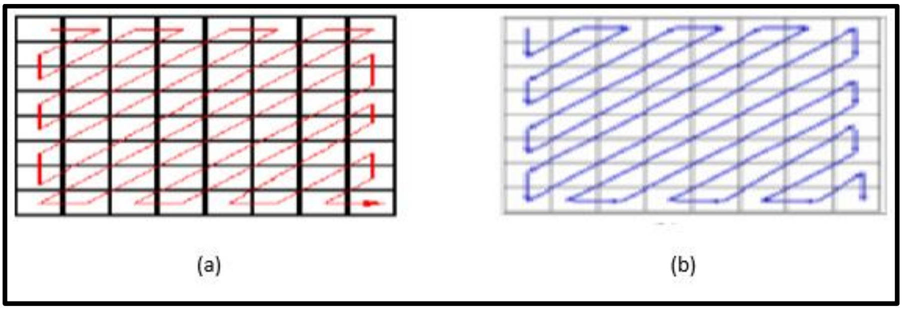


Figure 3: Two types of zig zag orders

4.6. Run Length Count Features

The transitions from one to zero and from zero to one represented important information for edge identification of binary image. Therefore they contribute to the object recognition process. RLC can address the variations in writing style. The horizontal RLC is obtained by scanning the image/block from left to right. And counting the number of continuous 1's in all the rows. The vertical RLC is obtained by scanning the image/block from top to bottom and counting the number of continuous 1's in all the columns. The vertical and horizontal RLCs of the blocks in an image can be represented as a feature vector. Figure 4 illustration of horizontal run length count method. The red, and blue arrows illustrated the transition from 1 to 0 and 0 to 1 respectively[18].

1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	0	1	1	1
1	1	1	1	1	0	1	0	1	1
1	1	1	1	0	1	1	0	1	1
1	1	1	0	1	1	1	0	1	1
0	0	0	0	0	0	0	0	1	1
1	1	1	1	1	1	1	1	1	1
1	1	1	1	1	0	0	1	1	1

Figure 4: Illustration of horizontal run length count

4.7. Classification and Recognition

The next stage after finishing from the feature extraction process is classified all training images into different classes, then recognize the unknown image to one of them. Several types of classifiers can be used for Off-line Handwriting word recognition problems.

4.8. SVMs Classifiers

The Support vector machines (SVM) are a type of supervised machine learning models which can be used for classifications or regression problems [19]. SVMs were designed for binary classification, In order to use SVM for multiclass classification problems there are two types of approaches such as one-against-one and one-against-all approaches [20]. In higher dimensional feature spaces SVM use kernels to separate between classes. SVMs can find the optimal separating hyper plane by different kernel functions which can transform a non-linear separable problem into a linear separable one and projecting data into the feature space [21].

4.9. K-Nearest Neighbor

K-Nearest Neighbor (KNN is a technique used to classify objects based on closest training samples in the feature space. KNN is a simple algorithm of all machine learning algorithms. The training samples are vectors with a class label for each vector. The training phase consists only of storing the feature vectors with its labels of the training samples. In the classification phase to find out the test sample to which class it belongs. First step computes its distance to every training sample. Then, keep the k (k is a positive integer) closest training samples, second by a majority vote of its neighbors find the best class for test samples. There are several distance functions used In KNN algorithm the best methods are Euclidean distance (equation 5) [22].

$$\mathrm{d}(\mathrm{x},\mathrm{y}) = \left[\sum_{i=0}^{m}(x_i - y_i)^2\right]^{1/2} \qquad \textbf{(5)}$$

5. The Proposed Word Recognition System

The proposed system consists of three stages: pre-processing, feature-extraction and classification/recognition stages.

A. Preprocessing stage

The important step in any handwritten word recognition system is pre-processing stage because it has effectiveness role on the recognition accuracy. Preprocessing stage need only clipping and normalization steps due to more steps are used when contracted this database. Algorithm 2 described preprocessing algorithm that used in the proposed system .Step1 removing the white space around the word object by using clipping algorithm that illustrated in algorithm1. In step 2 normalize the image into three types of size normalization such as 200*100 ,120*120 ,128*128 sizes because each feature extraction method needs a special size to complete its steps.

Algorithm 2: Proposed Preprocessing Stage **Input**: Word image **Output**: Preprocessed word image
Step1: Remove the white space by using clipping algorithm. **Step2**: Normalization, each image was normalized to three different sizes, 256*256, 200*100 and 120*120 pixels. **Step3**: Return preprocessed word images (Image1, Image2 and image3).

B. Feature Extraction stage

A proposed feature extraction method extracted the important information from word image by using two groups of feature extraction techniques. The first group is combined gradient direction and RLC methods. The second group used DCT technique. Algorithm-3 illustrated all steps of the proposed feature extraction method. By step1,2 and 3 read the capping images with 200*100 ,120*120 and 128*128 sizes to contracted

Image1,Image2,Image3, respectively. Step4 apply gradient direction algorithm (Agorathim4) on image1 to contracted the first feature vector FV1 of Group1, step5 apply RLC algorithm (Agorathim5) on image2 to contracted the second feature vector FV2 of Group1, then step6 combination the FV1 with FV2 to contracted feature vectors of Group1 which called FV3. Step7 apply DCT algorithm (agorthim6) on image3 to contract the first feature vector FV4 which denoted Group 2. Step8 return the feature vectors FV3 and FV4 which are used for training and testing data.

Gradient directional method is selected the clipping image with 200*100 normalization, then applied Prewitt filter masks on this image. These masks are shown in Figure 2. After applying the Prewitt filter masks gets two components which are vertical and horizontal gradient components which denoted (G_y) and (G_x) that calculated by equations 6 and 7.

$$G_x(i,j) = f(i-1,j-1)-f(i-1,j)-f(i,j+1)+f(i+1,j+1)+f(i+1,j)+f(i+1,j+1) \quad (6)$$

$$G_y(i,j) = f(i-1,j-1)-f(i,j-1)-f(i+1,j-1)+f(i1,j+1)+f(i,j+1)+f(i+1,j+1) \quad (7)$$

Each point (x,y) in the gradient image has direction component θ(x,y) which calculated by applying equation2. The value of θ(x,y) in the range [-90,90]. This method reducing the range values into 5 values only: [-90,-45, 0, 45, 90]. If the value of θ(x,y) between two values, which can take the nearest value to it. The gradient image was divided into 4*4 blocks, as shown in Figure 6 b. For each block, find the histogram for 5 directions. The size of the feature vector from this method will be (4×4×5) = 80 features. Algorithm 3 illustrated all steps of this method.

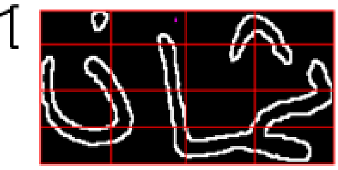
Algorithm 3: Proposed Feature Extraction Stage **Input**: clipping binary image **Output**: Features vector5
Step1: Read the clipping image with 200*100 sizes And called it Image1. **Step2**: Read the clipping image with 120*120 sizes And called it Image2. **Step3**: Read the clipping image with 128*128 sizes And called it Image3. **Step4**: Apply gradient direction Algorithm on Image1 To contracted feature vector FV1. **Step5**: Apply RCL Algorithm on Image2 to constructed Feature vector FV2. **Step6**: combination gradient direction feature vector FV1 and RCL feature vector FV2 to constructed group One of feature vectors which called FV3. **Step7**: Apply DCT Algorithm on Image3 to constructed feature vector which denoted the group two of Feature vectors which called FV4. **Step8**: Return the feature vector FV3 and FV4.

Algorithm 4 : Gradient feature method **Input**: Clipped Binary image(image1) **Output**: Gradient Features (FV1)
Step1: Input the clipped binary image (image1) **Step2**: Apply the Prewitt edge detection operator on the Clipping binary image. **Step3**: Compute vertical and horizontal gradient components(G_x,G_y) by equations 6 and 7, then calculate gradient direction of each pixel in image1 by equations 2.

Step4: Divide the gradient image into of 4*4 blocks
Step5: for each block find the histogram gradient
Directions, for 5 angles which are [-90,-45, 0, 45, 90].
Step6: Put the histogram values into 1D vector called it
FV1.
Step7: Return FV1 (80 values)

Theta	-90	-45	0	45	90
Count theta	20	10	23	21	5

(a)



(b)

Figure 5: Gradient direction feature a) divided the Gradien Image into 4*4 Blocks b) The histogram gradient direction for first block on image word “ثمان"

RLC method is used the clipping image with 120*120 size. Then divided the image into 3*3 blocks. For each block, find the run Length count in vertical and horizontal direction. These values denoted the number of translation from 0 to 1 and from 1 to 0. The RLC feature is 9*2=18 features. Algorithm 5 illustrated all steps of this method.

Algorithm 5: RCL method

Input: clipped Binary image(image2)

Output: Feature (FV2)

Step1: Input the clipped binary image (image2)

Step2: Divided image2 into 3*3 blocks

Step3: For each block find the vertical and horizontal run Length. **Step4**: build a feature vector of length 18 (FV2) **Step5**: Return FV2 (18 values)

The DCT technique is applied on the clipping image with 128*128 size. The DCT coefficients of high values are clustered in upper left corner. In order to get the feature vector from this method, taking the first 50 values of DCT coefficients in zigzag order. Figure 6 shows the DCT image on clipping binary image. All steps of this method illustrated in algorithm 6.

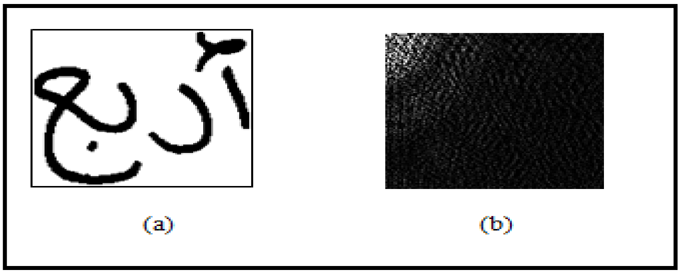


Figure 6: a) Original image b) DCT image

Algorithm6: DCT method **Input**: Clopping Binary image (Image3) **Output**: feature vector FV4
Step1: Read the clipping binary image (Image3) **Step2**: Calculate DCT for Image3 by using Equation (3) to constructed new image DCT_image . **Step3**: Convert the DCT_image with 2D array to 1D array by using zigzag fusion. **Step4**: Choose the first 50 attriputes of the 1D array saved in new vector called FV4. **Step5**: return the FV4 (50 values)

C. Classification / Recognition stage

This stage is divided into two phases: the first phase is called training phase which classified all training data into their classes .The second phase is called testing phase which recognized the testing image by finding the closed vectors of training data. Training phase is used SVM and KNN classifiers, testing phase recognized the unknown word image by using a new method which combine SVM with KNN classifiers. The proposed recognition method gave best and efficient recognition accuracy when using a single classifier. The proposed system used (Accord.NET) library Ver 3.02 which supports multi-class problem for SVM and the KNN classifiers. This system was programed by VB.NET programming. Algorithm 7 illustrated all the steps of the proposed classification and recognition stage. Step1 read the training data which consist of two set (train data and validation data), then read the test image. Steps 2,3 and 4 recall SVM classifier function from Accord.net library which used to train the train data and KNN classifier is used to train validation_data.Steps 5,6,7 find the class label of the test image by using SVM classifier through calculate FV3 for it,if did not recognize the class label ,then using KNN classifier through calculate FV4 for it.

6. Experimental Results and Discussion

All stages of the proposed system are programmed by visual basic.net language except SVM classifier which uses the Arecord.net library Ver. 3.2. AHDB database is used to evaluate the proposed recognition system. From this database 3150 word images for 30 classes are selected. These data are divided into two sets. First set has 2250 images, then divided into two subset, one is called training data which has 1470 image and other is validation set which has 780 images. The second set is testset which has 900 images. (70% of data for training and 30% for testing). In the preprocessing stage need only clipping and normalization steps because the other steps are used in the constructed this database. The outputs from preprocessing stage three word images with different sizes which are 200*100,120*120 and 128*128, each one is used for a special feature extraction method. In feature extraction stage finding out two groups of feature extraction methods. first group is combination two methods which are gradient feature applied on clipping binary image of 200*100 size then divided the image in to 4*4 blocks to find the feature vector FV1 and RCL feature this method applied on clipping binary image with size 120*120 and divided into 3*3 blocks in order to find its feature vector FV2. By combining these feature vectors to construct a single vector FV3 which used for training and testing the train/test data. The second group has one method which is DCT technique that used images of 128*128 size and its vector called FV4, then applied to train validation data. Then all vectors of training images and validation images are fed to train SVM, KNN classifiers respectively. In order to recognize unknown image (test image) find out group1 feature (FV3), then used SVM to predication its class label. If the class label, is rejected then find out group2 (FV4) feature for test image, then use the KNN classifier to recognize its class label. Several experiments were performed with single feature, combination features, Different image sizes and different filters for gradient feature extraction.

Table1: Recognition rate with different gradient filters by using SVM

Gradient Feature types for Fv4	Feature length	Recognition Accuracy
Sobel Filter	80	91.89%
Prewitt Filter	80	92.33%

Table 1 illustrated the recognition accuracy when extracting the gradient features by Prewitt filter and Sobel filter. The recognition accuracies for these filters are: 91.89%, 92.33%

respectively by using SVM classifiers. Therefore, choose Prewitt filter for the proposed system.

Table 2 illustrated the recognition accuracy for 3 experiments by using three types of classifiers which are: SVM classifier with polynomial kernel, KNN classifiers and combination SVM-KNN classifiers. When using gradient feature FV1 its result is 92.33, 85.11 and 96.0 respectively. And the results of RCL method are 82.22, 72.56 and 93.22 but when combine FV1 with FV2 the result are: 94.11%, 88.11% and 97.11. The best recognition accuracy for the proposed system when combine feature extraction and combination SVM with KNN classifiers and their result is 97.11.

Table 2: Recognition accuracy of different features extraction methods by SVM, KNN and combination SVM with KNN classifiers.

Feature Vector	RCL Method	Gradient Feature (Prewitt mask)	Feature vector length	Recognition Accuracy% SVM	Recognition Accuracy % KNN	Recognition Accuracy % SVM-KNN
Fv1		Y	80	92.33	85.11	96.0
Fv2	Y		18	82.22	72.56	93.56
Fv3	Y	Y	98	94.11	88.67	97.11

Table 3 illustrated the recognition accuracy for different types of SVM kernels which combination with KNN classifier. The Polynomial kernel has a higher recognition rate than other types which is 97.11%.

Table 3: Recognition accuracy in combination SVM by different kernels with KNN classifiers.

SVM type	Recognition rate
Linear	96.67
Gaussian	95.67
Polynomial (2)	97.11

7. Conclusion

The proposed system recognizes the handwritten Arabic words as one entity without using segmentation stage to readable format. The proposed hybrid classifier used increases the performance of the recognition system more than individual classifiers by making specific features for each classifier, e.g. using SVM classifier followed by KNN classifier. The accuracy of the proposed system is 97.11% when tested on AHDB database. The proposed system has the capability to process off-line handwriting/ printed Arabic numbers or letters images (for both English and Arabic characters).

Algorithm7: Classification and Recognition stage **Input**: Training vectors called **train_data** by group1 feature vector, validation vectors by group2 called **validation_data**, testing image vector1 by group1 called **test_image1**, testing image vector2 by group2 called **test_image2**, no of classes called **no_classes**, **N1** number of attributes for vector1, **N2** number attributes for vector2. **Output**: **class_test_label** for test image.
Step1: Read **train_data**, **test_image**, **no_classses**. **Step2**: Recalls Arecord.net functions for SVM. **Step3**: Train the SVM with polynomial kernel type for **train_data** to classify the training images. **Step4**: Train the KNN by **validation _data** to classify the Validation images. **Step5**: Find out the class of **test_image** by using SVM Which called **class_test_label**. **Step6**: If correct recognition goto step8 else **Step7**: Using KNN classifier to recognize the **test image**

Called **class_test_label**.
Step8: return **class_test_label**

References

[1] Abdul Hassan A. K., Kadhm M.S., "Handwriting Word Recognition Based on SVM Classifier", (IJACSA) International Journal of Advanced Computer Science and Applications. vol. 6, no. 11, pp.64-68, 2015.

[2] Al Khateeb J. H., Ren J., Jiang J. and Al - Muhtaseb H., "Offline handwritten Arabic cursive text recognition using Hidden Markov Models and re-ranking", Pattern Recognition Letters., vol. 32, pp.1081– 1088, 2011.

[3] Shruthi A., Patel M. S., "Offline Handwritten Word Recognition using Multiple Features with SVM Classifier for Holistic Approach", International Journal of Innovative Research in Computer and Communication Engineering., vol. 3, Issue 6, 2015.

[4] AI Khateeb J. H, Ren J., Jiang J. and Ipson S. S., "Word-based Handwritten Arabic Scripts Recognition using DCT Features and Neural Network Classifier", 5th International Multi-Conference on Systems, Signals and Devices, IEEE Publisher, Amman, Jordan, pp.1 – 5, 2008.

[5] Nemouchi S., Meslati L. S. and Farah N., "Classifiers Combination for Arabic Words Recognition Application to Handwritten Algerian City Names," ICISP, vol. 7340, pp. 562-570, Agadir, Morocco, 2012.

[6] Abdul Hassan A. K., and Kadhm M. S., "Arabic Handwriting Text Recognition Based on Efficient Segmentation, DCT and HOG Features", International Journal of Multimedia and Ubiquitous Engineering, vol.11, No.10, pp.83-92, 2016.

[7] Al Ma'adeed. S., "Recognition of Off-Line Handwritten Arabic Words Using Neural Network", Geometric

Modeling and Imaging –New Trends, Page(s): 141 – 144 Publishers IEEE, Pages: 141 - 144, 2006.

[8] Al-Ma'adeed S.., Higgins C., Elli man D., "Off-line recognition of handwritten Arabic words using multiple hidden Markov models", Research and Development in Intelligent Systems, Published in Springer London, pp. 33-40,2004.

[9] Al-Ma'adeed S., Elliman D., and Higgins C., "A Data Base for Arabic Handwritten Text Recognition Research", The International Arab Journal of Information Technology, Vol. 1, No. 1, January 2004.

[10] Abdul Hassan A. K., Kadhm M. S., "An Efficient Preprocessing Framework for Arabic Handwriting Recognition System", Diyala Journal for pure sciences vol. 12, No:3 , 2016.

[11] Lawgali A., "A Survey on Arabic Character Recognition", International Journal of Signal Processing, Image Processing and Pattern Recognition, Vol. 8, No. 2, pp. 401-426, 2015.

[12] Kadhim M. K., "Offline Hand Written Letter Recognition Using Neural Networks Based on Multiple Feature Extraction Algorithms", M.Sc. Thesis, University of Technology, Computer Engineering Department, 2014.

[13] Kaur K., Malhotra S., "A Survey on Edge Detection Using Different Techniques", International Journal of Application or Innovation in Engineering & Management (IJAIEM) Vol. 2, Issue 4, No.77, PP.496-500, 2013.

[14] Yadav N., Ashraf M., Yadav M., " Improvement of Edge Detection Technique in Images Using Soft Computing", International Journal of Innovations & Advancement in Computer Science (IJIACS), Vol. 4, Issue 6 , P. 74-77, 2015.

[15] Al-Haj A., "Combined DWT-DCT Digital Image Watermarking", Journal of Computer Science, Vol. 3, Issue 9, pp. 740-746, 2007.

[16] McLaren M., Schaeffer N., Ferrer L., and Lei Y., "Effective Use Of DCTs For Contextualizing Features For Speaker Recognition", Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE International Conference in Italy, 2014.

[17] Al Khateeb J. H., Al seid M., "DBN – Based learning for Arabic Handwritten Digit Recognition Using DCT Features", 6th International Conference on Computer Science and Information Technology (CSIT), 2014, IEEE Conference Publications, 2014.

[18] Moni B. S., Raju G. "Modified Quadratic Classifier for Handwritten Malayalam Character Recognition using Run length Count", International Conference on Emerging Trends in Electrical Computer Technology, Nagercoil, India, IEEE Conference Publications, pp. 600 – 604, May 2011.

[19] Vapnik V., The Nature of Statistical Learning Theory, New York, USA, 2000.

[20] Milgram J., Cheriet M., Sabourin R., "One against One "or" One against All: Which One is better for Handwriting Recognition with SVMs?" Guy Lorette, Tenth International Workshop on Frontiers in Handwriting Recognition, LA Baule (France), 2006.

[21] Sagheer M.W., He C. L., Nobile N., "Holistic Urdu Handwritten Word Recognition Using Support Vector Machine", International Conference on Pattern Recognition. pp. 1900 - 1903, IEEE Conference Publications, 2010.

[22] LiLi L., Xia Z. Y. and Heng Z. Y., "K-Nearest Neighbors for automated classification of celestial objects", Science in China Series G. PhysMech Aston, Vol. 51, No.7, pp. 916-922. 2008.

دمج مصنفات الجار الاقرب والة دعم المتجهات لتمييز الكلمات العربية المكتوبة بخط اليد بالاعتماد على الصفات المتعددة

أ.م.د علياء كريم عبد الحسن
hassanalia2000@yahoo.com
الجامعة التكنولوجية - قسم علوم الحاسوب - بغداد - العراق
محمد علاوي عباس
alawialawi3000@gmail.com
الجامعة التكنولوجية - قسم علوم الحاسوب - بغداد - العراق

المستخلص

نقدم في هذا البحث نظاما مقترحا لتمييز الكلمات العربية المكتوبة بخط اليد. النظام المقترح يمييز الكلمة العربية ككيان واحد دون استخدام مرحلة التقسيم والتي تحول الكلمة إلى أجزاء. في مرحلة استخراج الصفات تم دمج تقنيات استخلاص الصفات الى مجموعتين. المجموعة الأولى تتالف من تقنيتين. تجمع بين طريقة اتجاه الانحدار gradient (directional) feature مع طريقة حساب طول المسار Run Length Count بينما المجموعة الثانية تستخدم تقنية واحدة وهي طريقة تحويل الجيب تمام المنقطع Discrete Cosine Transform. مرحلة التصنيف تعتمد على دمج نوعين من المصنفات وهما SVM و KNN آلة دعم المتجهات و الجار الاقرب. تم اختبار النظام المقترح باستخدام قاعدة بيانات قياسية التي هي AHDB. ان دقة التميز النتائج التجريبية للنظام المقترح هي 97.11٪.

الكمات الرئسية: تمييز الخط اليدوي، صفات التدرج، آلة دعم المتجهات، الجار الاقرب.