Improved Hierarchical Sparse Method with application to Offline handwritten Arabic character recognition

Ramadhan Abdo Musleh Alsaidi, Hong Li, YantaoWei and Rokan Khaji

Abstract— Offline handwriting Arabic character recognition has received increased attention in recent years. In this paper, we introduce a novel technique to enhance the recognition rate of offline handwritten Arabic characters within the low computation time based on developing Hierarchical Sparse Method offered. The developed principles of this algorithm can be specified as scaling and k-mean techniques. For scaling level, the main advantage is the avoidance of attributes in greater numeric ranges dominating those in smaller numeric ranges, at the same time, the k-mean used to obtain an efficient template selection method included in the derived kernel for developing the object recognition performance. In this paper, the experimental results comprise the developed algorithm matching with other existing techniques.

Index Terms— Offline handwriting Arabic character, scaling, k-mean, Hierarchical Sparse Method,IFN/ENIT database.

---

1 Introduction

THE great rank of Arabic script appearing through several aspects, for example, Arabic characters adopted to use in many languages-like Arabic, Malay, Persian, Urdu, Jawi and Pishtu, (more than a half of a billion people), in addition to all Muslims (almost ¼ of (the) peoples on the earth) can read Arabic this because it is the language of Al-Quran, the holy book of Muslims. Moreover, the study area of recognizing Arabic character is still vacant due to limited studies and therefore, further research is required. On the other hand, it is important to note that spoken Arabic is varies from one country to another, but the Arabic writing has a standardized version for official communication across the Arab world. The standard Arabic writing occasionally called Modern Standard Arabic (MSA). Despite Arabic is one of the most widely used languages in the world, demand for a robust Optical Character Recognition (OCR) for this language could be with great commercial and economic valuables. Script recognition can be either online or offline. If the scripts recognition take place during the user input is refer to online and to offline if the scripts were recognized after the finishing of writing. (It is online if recognizes scripts during the user input. While offline script recognition, by opposition, recognizes the input data after the writing is completed.)

Many works have been proposed for Arabic handwriting recognition. Most of these works are for whole word recognition while, a few for isolated character recognition. Recognition of Arabic characters represents an important goal, not only for the Arabic speaking countries, but also for Urdu speaking Indians, Curds and Persians. In this work, we specifically concerned with Arabic offline handwritten characters recognition. Moreover, to the best of knowledge, in the former years some work has been devoted to improving Arabic offline handwritten characters recognition. Al-Yousefi and Udpa[1] introduced a statistical approach for the recognition of Arabic hand-written isolated characters using moments of the horizontal and vertical projections of the primary character. Abuhabiba. et al.[2] improved an off-line character recognition. An algorithm was developed, which yields skeletons that reflect the structural relationships of the character components. In other study [3], the author's extracted moment based features in order to recognize handwritten Arabic character. They used genetic algorithm for feature selection and use SVM to evaluate the classification error for the chosen feature subset. Khedher and Abandah described the main characteristics of the Arabic writing and provided statistics for PAWs and letter forms [4]. Sari et al. described the general characteristics of Arabic text and used morphological features of the Arabic character such as turning points, holes, ascenders, descanters, and dots for segmentation and recognition [5]. Pechwitz et al. have collected a database of handwritten Arabic names for Tunisian towns and published statistics about the size of this database in words, parts of Arabic words (PAWs), and characters [6]. Menasri et al. identified letter body alphabet for handwritten Arabic character; they classified Arabic character into root shapes and optional tails. Multiple Arabic characters that only differ in the existence and number of dots are mapped to the same root shape. This alphabet also includes common vertical ligatures of joined character [7]. Malas et al. provided statistics about frequencies of Arabic character and letter pairs [8]. Abandah et al. (2009) suggested an approach for text feature extraction to achieve high recognition accuracy of handwritten Arabic character. However, the reported recognition rates for Arabic character need more improvements to be practical [9]. In novel study [10], the authors extract moment based features in order to recognize handwritten Arabic letters. They use genetic algorithm for feature selection and use SVM to evaluate the classification error for the chosen feature subset. Rashad Al-Jawfi[2009] introduced a new network based on LeNet architecture was designed to recognize a set of handwritten Arabic character[11]. Rawan Ismail Zaghloul, introduced a multilevel classifier based on pattern recognition techniques [12]. Finaly in one of the poular studies [13], the authors described hybrid feature extraction for offline handwritten character recognition. The proposed technique was considered as a hybrid of structural, statistical and correlation features.

Motivation

Many deficiencies accompanied previous research, such as data is few and it does not cover all the character cases. In contrast, this work uses data set of offline handwritten Arabic character images (OHACI) which carefully selected from IFN/ENIT-database with some modification that may be used in training and testing for Arabic (OCR). It covers the various forms of Arabic alphabet in most positions (i.e.alone, beginning, medial, and end). The database includes 209992 samples of Arabic letters written by 411 different writers. Moreover, this work deals with the problem of overlap between the characters when writing and making it more difficult to distinguish characters. In the other hand, the advantage of developed methodology with respect to Hierarchical sparse methods can be summarized as; firstly, the hierarchical algorithms use template set as a dictionary in the sparse coding operation to develop a dictionary. In opposition, the developed method proposed an efficient template selection process involved in the derived kernel for improving the object recognition performance based on k-mean technique. The major advantage behind this approach is to consider not only the label information of the training images, but also the elimination of redundant information. For scaling process, the main benefit is to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. In summary, this develop aim is to achieve high accuracy with low computational complexity for object recognition.

This paper is organized in 6 sections. In section 2, the basic concept of Hierarchical sparse coding, scaling and k-mean techniques is discussed. The essential background for Arabic letters is described in Section 3. The process of finding the Offline Handwritten Arabic Character images is discussed in Section 4. Section 5 describes the feature extraction techniques used in this research. It also includes the implementation of these techniques and experimental result report. Finally, the concluding remarks are given in Section 6.

2 Some Characteristics of Arabic Character:

This section provides a comprehensive list of characteristics of the Arabic characters with figures to illustrate the concepts. We aim to introduce a source for researchers to start with. These characteristics will be presented as argued in [13, 14].

TABLE 1
SAMPLES SHOWING VARIATIONS IN HANDWRITTEN LETTERS.

NO	Secondary Type	Example
1	No Secondary	ا ى ح در س ص ط ع م ل ه و ء
2	One Dot	ج ب خ ز ذ ظ ض غ ف ن
3	Two Dots	ة ت ي ق
4	Three Dots	ش ث
5	Zigzag	ؤ ئ أ ك
6	Vertical Bar	ط
7	Vertical Bar and a dot	ظ
8	Long Stroke	ک

As we mention above Arabic alphabet is the script used for writing several languages of Asia and Africa. The calligraphic style of the Arabic cursive is distinguished from other languages in many ways. Arabic text is written from right to left in both handwritten and printed forms, with the alphabet having 28 basic characters. Furthermore, sixteen Arabic letters have between one to three dots. Position and number of these dots differentiate between the otherwise similar characters. Moreover, some letters (like ؤ ئ أ ك) can have a zigzag-like stroke (Hamza ء). These Hamza and dots are sometimes called secondaries and they are located above the letter primary part as in ALEF(أ), or in the middle like JEEM (ج), or below like BAA(ب). Written Arabic text is cursive in both handwritten and printed-machine text (The details of the letter secondaries are illustrated in Table 1).

Within a word, some letters connect to the preceding and/or following characters, and some do not connect as show in table. Clearly, the above six characters (و, ز, ر, ذ, د, ا), if appeared in a word, will cause the word to be divided into blocks of connected components called subwords. Thus a word can have one or more subwords. Subwords are also separated by spaces, but usually shorter than the one between words. So, this issue needs to be considered to avoid segmenting a word into two parts.

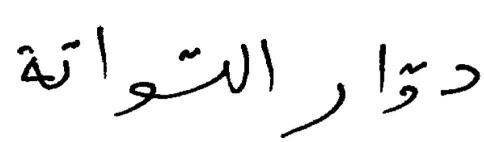

Fig. 1 Example of characters causing word divided.

ISSN 2229-5518

TABLE 2
DIFFERENT FORMS OF CHARACTERS DEPENDING ON ITS POSITION IN THE WORD.

Character Name	Isolated	Initial	Middle	Final	Forms
Alif ألف	ا	-	-	ـا	2
Ba' باء	ب	بـ	ـبـ	ـب	4
Ta' تاء	ت	تـ	ـتـ	ـت	4
Tha' ثاء	ث	ثـ	ـثـ	ـث	4
Jeem جيم	ج	جـ	ـجـ	ـج	4
H'a' حاء	ح	حـ	ـحـ	ـح	4
Kha' خاء	خ	خـ	ـخـ	ـخ	4
Dal دال	د	-	-	ـد	2
Thal ذال	ذ	-	-	ـذ	2
Rai راي	ر	-	-	ـر	2
Zai زاي	ز	-	-	ـز	2
Seen سين	س	سـ	ـسـ	ـس	4
Sheen شين	ش	شـ	ـشـ	ـش	4
Sad صاد	ص	صـ	ـصـ	ـص	4
Dhad ضاد	ض	ضـ	ـضـ	ـض	4
Tta' طاء	ط	طـ	ـطـ	ـط	4
Dha' ظاء	ظ	ظـ	ـظـ	ـظ	2
A'in عين	ع	عـ	ـعـ	ـع	4
Ghain غين	غ	غـ	ـغـ	-	2
Fa' فاء	ف	فـ	ـفـ	ـغ	4
Qaf قاف	ق	قـ	ـقـ	ـق	4
Kaf كاف	ك	كـ	ـكـ	ـك	4
Lam لام	ل	لـ	ـلـ	ـل	4
Meem ميم	م	مـ	ـمـ	ـم	4
Noon نون	ن	نـ	ـنـ	ـن	4
Ha' هاء	ه	هـ	ـهـ	ـه	4
Waw واو	و	-	-	ـو	2
Ya' ياء	ي	يـ	ـيـ	ـي	4
Hamza همزة	ء	-	ـئـ	-	2
Ya'Lainh ياء لينة	ى	-	-	ـى	2
Ta'Mrbotah تاءمربوطة	ة	-	-	ـة	2
LamAlif لام ألف | لا | لآ | لأ | لإ | 4 | إ | إ | آ | - | 3	لا	-	-	ـلا	2

The shape of an Arabic alphabetic depends on its position in the word; every alphabetic up to four different shapes depending on the position of the letter in the word: whether it is isolated letter, connected only from right (initial form), connected only from left (ending form), or connected from both sides (medial form) (The details of the character shapes are shown in Table 2.). Some letters, in a word may overlap vertically wit especially in Arabic handwritten, may overlap with their neighboring characters forming what is called "ligature". Figure2 illustrates some ligature examples. These include: presence of baseline (i.e. Arabic letters are 'normally' connected on an imaginary line called baseline).

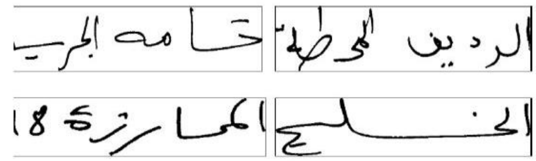

Fig. 2 some words include ligature.

3 A SUMMARY OF HIERARCHICAL SPARSE CODING METHOD (HSCM)

In this part, we will give a brief introduction to the Hierarchical sparse coding Method. Who cares is encouraged to consult for a more detailed treatment. The classic HSCM feature detector, as described in [15], is essentially combining the hierarchical architectures with the sparse coding technique. As far as proposed layered model, at each layer of hierarchy, it concerned two components that were used are sparse coding and pooling operation. While the sparse coding was used to solve increasingly complex sparse feature representations, the pooling operation by comparing sparse outputs was used to measure the match between a stored prototype and the input sub-image. It is recommended that value of the best matching should be kept and discarding the others. In simple words, this approach defines a nonlinear mapping from the lower layer to the upper layer. The nonlinear mapping approach has been defined as a nonlinear mapping function f as:

$$N^{v} = f(x^{v}, T_{u}).$$

Here, the fundamental nonlinear mapping function f consists of a chain of processes i.e. $f = p \circ g$, where p denotes pooling operation (Pooling operation can be taken as average, max, and energy, however, in this particular study, we are only concerned about max pooling) and g denotes sparse coding function which uses to solve optimization problem

$$\min \|s\|_0 \ s.t. \ x = Ds, \qquad (1)$$

as: $s = g(x, D)$,

Where x is a column signal, D is the dictionary, $\|\cdot\|_0$ is the l^0 pseudo-norm which counts the non-zero entries and s is the sparse representation of x. The significance of this relation is based on the fact that function f is able to map a set of feature vectors from the lower layer to a single feature vector at the upper layer.

To avoid attributes in greater numeric ranges dominating those in smaller numeric ranges we do scaling.

Scaling:

Scaling is a technique used for two goals. First, to avoid numerical difficulties during the calculation. Second, to avoid attributes in greater numeric ranges dominating those in smaller numeric ranges. This process is done through using this formula

$$\hat{V} = \frac{V - min_a}{max_a - min_a}$$

Where V is original value, min_a is low bound of the template value, max_a is upper bound of template value, and $\hat{V}$ is scaled value. Note that, this work concerning linearly scaling each attribute to the range [-1; +1] or [0; 1].

Template selection

In HSCM, the templates are randomly sampled. Certainly, these templates often not fully reflect the structure characteristics of the object to be recognized. This developed considers the hierarchical with three-layer architecture. Hence, two template sets need to be constructed.

(i) Selecting the first layer templates

- An initial template set, Pu, is created, where Pu is a large pool of image patches of size u randomly obtained from cropped images that hold the instance of the object to be recognized.
- We adopted the K-means algorithm to cluster all the image patches into |T u| clusters. Note that the image patches in Pu are previously transformed into vectors.
- Taking the clustering centers as the first layer templates.

  In this manner, a small number of templates can be constructed. In this phase, size of the template is determined experimentally. In general, the first patch templates should be large enough to cover basic elements of the objects, e.g., corners of the eyes.

(ii) Selecting the second layer templates

(1) Sample a large number of candidate templates, denoted by $p_i \in p_v (i = 1, 2, ..., M)$. They are obtained randomly from the cropped image set C in which the objects fill most of the images.

(2) Compute the score of each candidate template in Pv by

$$S(p_i) \triangleq \frac{S_w(p_i) + S_b(p_i)}{S_w^2(p_i)}, \text{ where}$$

$S_w(p_i)$ is within-class scatter, $S_b(p_i)$ is between-class scatter and $S(p_i)$ is a criterion for template selection.

In this case, the label information of the training images is uesed.

(3) Rank the candidate templates in descending order according to their scores. Consequently, we rewrite Pv as

$$P_v \triangleq P_v^r = \{p_1^r, p_2^r, ..., p_M^r\},$$

Where $p_i^r (i = 1, 2, ..., M)$ is the ith image patch in P_v^r

(4) The ith image patch is selected as a template, if the correlation coefficients between the ith image patch and the selected templates are all smaller than a given threshold T. The correlation coefficient of two image patches $p_i^r, p_j^r \in P_v^r$ is defined by

$$\mathbb{C}(p_i^r, p_j^r) = \frac{\mathbb{C}^{\sim}(p_i^r, p_j^r)}{\sqrt{\mathbb{C}^{\sim}(p_i^r, p_i^r)\mathbb{C}^{\sim}(p_j^r, p_j^r)}},$$

Where

$$\mathbb{C}^{\sim}(p_i^r, p_j^r) = \begin{pmatrix} N_{Sq}(I_1)(p_i^r) \\ .... \\ N_{Sq}(I_N)(p_i^r) \end{pmatrix}^T \begin{pmatrix} N_{Sq}(I_1)(p_j^r) \\ .... \\ N_{Sq}(I_N)(p_j^r) \end{pmatrix}$$

It is easy to find that the correlation coefficient is related to candidate templates and training images. In addition, T can be determined experimentally. You can find detail Information in [16].

The proposed developed in this paper is decomposed into the several following steps:

First step: deals with the defining of template sets t_i of size u by randomly extracting n sub-image patches of size u from each gray-scale input images.

Second step: deals with the concatenating of sub-images into a single matrix T_u, as given:

$$T_u = [t_1, t_2, ..., t_M] = \begin{pmatrix} t_{11} & \cdots & t_{M1} \\ \vdots & \ddots & \vdots \\ t_{1N} & \cdots & t_{MN} \end{pmatrix} \in \mathbb{R}^{N \times M}$$

Where $t_i \quad (i = 1, 2, ..., M)$ is the ith candidate and $T_u \subset \text{Im}(u)$. Here N is the dimension of each feature vector t_i after transformed into vectors and M is the total number template from all class.

In the same way we can construct the second layer template set T_v but with deferent size v as:

$$T_v=[t_1,t_2,...,t_L]=\begin{pmatrix} t_{11} & \cdots & t_{L1} \\ \vdots & \ddots & \vdots \\ t_{1K} & \cdots & t_{LK} \end{pmatrix} \in \mathbb{R}^{K\times L}$$

i.e. L signals of dimension K.

Third step: we applied scaling and k-mean as

$$T_u \xrightarrow{Scaling \wedge k-mean} New\ \tilde{T}_u \ \&\ T_v \xrightarrow{Scaling \wedge k-mean} New\ \tilde{T}_v$$

It is noteworthy to mention that the output templates $(\tilde{T}_u\ ,\tilde{T}_v)$ from this step is characterized by small dimensions and smaller numeric ranges. While the best size of the template is determined experimentally, the training set is denoted by $x=[x_1,x_2,...,x_q]$, where x_j $(j=1,2,...,q)$ is the jth training image and $x\subset \mathrm{Im}(Sq)$.

Fourth step: Initial sparse coding this step include three process

1. Taking a training image $x^v \in \mathrm{Im}(v)$.
2. Generation image pieces of size u using following relation:

$$x_j^v=\{x^v\circ h_j,\ j=1,2,...,|H_u|\}$$

Computation related to the sparse representation of each piece x_j^v. In othe rwords, we solve Eq. (1) as

$$s_j=g(x_j^v,\tilde{T}_u)=\begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{\tilde{M}} \end{pmatrix}.$$

Note that function g is an implementation of the Batch Orthogonal Matching Pursuit (BOMP)[17] and s_j is a vector contains many zeroes elements.

We can then write feature vector x_j^v as a linear combination of the entries in $\tilde{T}_u$ as: $x_j^v=\tilde{T}_u s_j=\sum_{i=1}^{\tilde{M}} t_i\alpha_i$.

Finally we get: $S^v=[s_1,s_2,...,s_{|H_u|}]$.

Fivth Step: Localized pooling step:

$$N^v(x^v)(t)=\max\{S^v : x^v\in \mathrm{Im}(v), t\in \tilde{T}_u\}.$$

Thus, the first layer neural response of training image x_j at a given candidate template $t_i\in \tilde{T}_u$ is denoted by $N^v(x_j^v)(t_i)$.

Sixth step: Sparse coding:

This step includes four processes are given as follow:

1. Replacement of the dictionary $\tilde{T}_u$ by $N^v(\tilde{T}_v)$ (i.e. since we are dealing with a hierarchy we used the output of the upper layer).

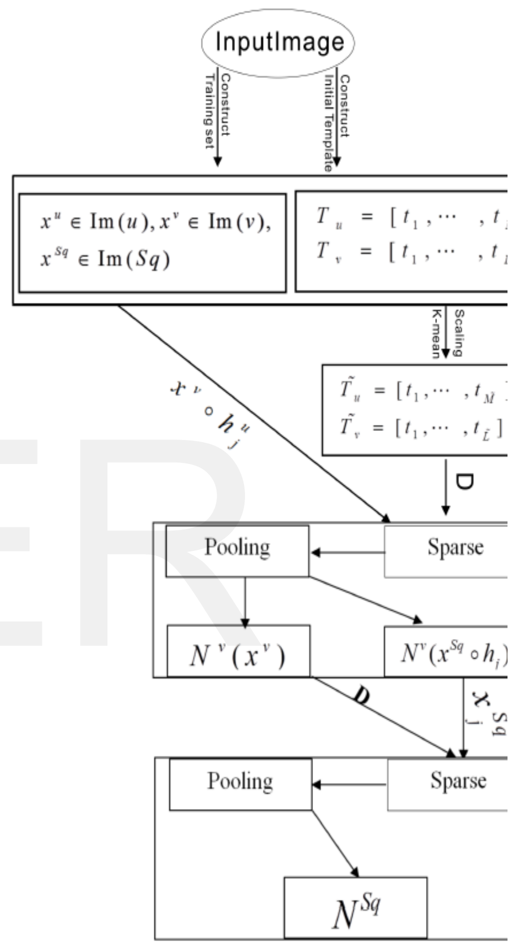

Fig. 3. Architecture of the developed model

2. Taking a gray-value input image $x^{Sq}\in \mathrm{Im}(Sq)$.
3. Decomposition of image x^{Sq} into pieces of image patch of size v by applying the following relation:

$$x_j^{Sq}=N^v(x^{Sq}\circ h_j),\quad j=1,2,...,|H_u|$$

4. Calculation related to the sparse representation for every pieces x_j^{Sq} by applying:

$$s_j = g(x_j^{Sq}, N^v(\tilde{T}_v)) = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{\tilde{L}} \end{pmatrix}.$$

Again at the end of sparse coding we get:

$$S^{Sq} = [s_1, s_2, ..., s_{|H_v|}].$$

The **final step** is related to the computation of a global maximum over all scales of S^{Sq} as:

$$N^{Sq}(x^{Sq})(t) = \max\{S^{Sq} : x^{Sq} \in \mathrm{Im}(Sq), t \in \tilde{T}_v\}.$$

Here, N^{Sq} represented a new feature. Applied these procedures to both training and testing, we get new set of features and recognition is performed in this new feature space. The introduced method is summarized in Fig. 3.

4 Experiments and Results

4.1 The database:

In this work, we use IFN/ENIT database [5, 17]. IFN/ENIT is the first database of Arabic handwritten script accessible to the scientific community. The total number of binary images of handwritten text is 26,459 with its ground truth information. Those images were labeled according to 946 name classes, and they were written by 411 writers. The database is arranged into four sets, a, b, c, and d, in order to conduct fourfold cross validation experiments. To make IFN/ENIT database suitable to our task each word image need to be segmented into its original characters. Therefore, we manually segmented all words images. The collection for beginning, middle, end and isolated letter forms as follows:

– Read the character labels from the ground truth file associated with the image.
– Define the boundaries of each character by hand.
– Assign the label for each character segment.
– Save each character as an independent image with its label.
– Group characters which have the same label in one folder.

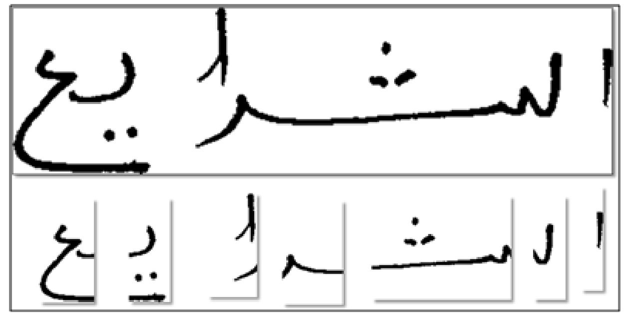

Fig. 4 Example of segmentation.

An efficient segmentation stage is required in order to divide a cursive word or subword into its constituting characters, but it is difficult -at the time being-. Therefore in this work we used previous approach for segmentation to avoid errors that may appear from an automatic character segmentation process.

4.2 Results of Experiments

In this research, proposed method on Offline handwritten Arabic character recognition was adopted. The developed recognition algorithm has been tested using training data and testing data from two distinct vocabularies. There are 209201 samples in total. We randomly chose 50% of the samples for each class from the reference data as training samples, and the remaining 50% as testing samples. Two tests were conducted using the same data.

The first, letters were distributed on the basis of their place in the word; total categories in this test became 113 classes. The second, every character was compiled regardless of its place in the word in one category, the total categories in this test was 38. For every test, we used the developed method as a feature extraction step for a classification algorithm; we adopted Support Vector Machines (SVM). The performance of the proposed method is significantly affected by the following parameters: patch size (u,v), number of elements in each linear combination (L), size of the available training set (Tr), selections of template size (Te) and number of clustering for each layer (Cl1, Cl2). After carefully scrutinizing the above experiments, we find the optimal parameters values used to achieve this accuracy are: L = 3; u = 14; v = 25; Te = 5; Cl1 = 500 and Cl2 = 700. Table 3 shows the main information of the previously methods, their results on one hand and those obtained using the developed method on the other and. Despite each method has different data but aiming at the same target.

ISSN 2229-5518

It is right that the accuracy found in this work is modest, but when we compared it to the challenges faced by this experiment, the result becomes acceptable. To the best of our knowledge, we can highlight the most important of these challenges as follows:

1. Through this work we dealt with a lot of data, so the extent of errors increased.
2. Some of those who participated in the writing of these letters did not abide by the rules of the Arabic writing of the letters, which caused more errors.
3. An essential difference between Latin scripts and Arabic is the fact that many letters only differ by dots, but the primary stroke is exactly the same. This makes a task to differentiate between letters very hard.
4. Moreover, the study deals with the problem of overlapping between the characters that makes it difficult to distinguish characters.

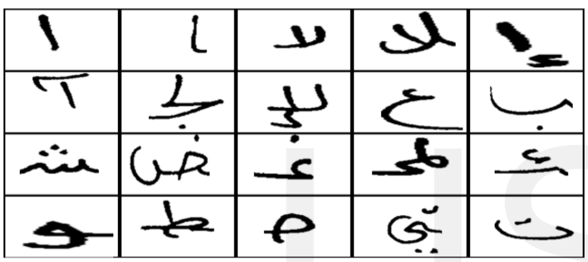

Fig. 5 Twenty instances from the set of training examples for Arabic character.

5 Conclusion

In this research, we present a novel technique based on developed hierarchical sparse coding to effectively construct new feature for characters. Through experiment, the method has been proved to be accurate at 63:75%. After the above experiments, we concluded that the recognition of Arabic handwritten character is hard not only because of the handwriting ambiguity but also because of the similarity between characters according to their position in a word. Consequently, to design a handwriting Arabic character recognition system with an accuracy rate of 100% is hard –at this time- because even human beings are not capable to recognize every handwritten script without any a doubt. Finally, we hoped that the developed system will be a step towards a hierarchical approach to robustly solve the aforementioned problem. Now, it remains for further research to build on this foundation and work towards automatic segmentation and recognition of offline Arabic words.

Acknowledgment

The authors wish to thank the anonymous reviewers for their insightful comments and suggestions.

TABLE 3

COMPARISON BETWEEN THE PROPOSED METHOD AND SOME PREVIOUS WORKS.

Method Name	Accu. %	Class
Neural Network (*10*)	58	
Discrete Wavelet transform (1) (*19*)	52.14	Isolated
Discrete Wavelet transform (2) (*19*)	26.36	All Forms
Artificial Neural Network (*20*)	52.1	Isolated
Probabilistic Model (*21*)	59.2	All Forms
Developed Method(1)	63.75	
Developed Method(2)	73.75	

References

[1] H. Al-Yousefi, S. Udpa, Pattern Analysis and Machine Intelligence, IEEE Transactions on 14, 853 (1992).

[2] I. Abuhaiba, S. A. Mahmoud, R. J. Green, Pattern Analysis and Machine Intelligence, IEEE Transactions on 16, 664(1994).

[3] G. Abandah, N. Anssari, Journal of Computer Science 5, 226 (2009).

[4] M. Khedher, G. Abandah, Third Int'l Conf. on Language Resources and Evaluation (LREC 2002), Arabic Language Resources and Evaluation–status and prospects workshop (2002).

[5] T. Sari, L. Souici, M. Sellami, Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on (IEEE, 2002), pp. 452–457.

[6] M. Pechwitz, et al., Proc. of CIFED (Citeseer, 2002), vol. 2, pp. 127–136.

[7] F. Menasri, N. Vincent, E. Augustin, M. Cheriet, Document Analysis and Recognition, 2007. ICDAR 2007. Ninth International Conference on (IEEE, 2007), vol. 2, pp. 969–973.

[8] T. M. Malas, S. S. Taifour, G. A. Abandah, Proc. 9th Int'l Middle Eastern Multiconf. on Simulation and Modeling (MESM 2008) pp. 50–54 (2008).

[9] G. Abandah, F. Khundakjie, Dirasat Engineering Sciences Journal 31, 165 (2004).

[10] R. Al-Jawfi, Int. Arab J. Inf. Technol. 6, 304 (2009).

[11] R. I. Zaghloul, E. F. AlRawashdeh, D. M. Bader, Journal of Computer Science 7, 512 (2011).

[12] M. N. Ayyaz, I. Javed, W. Mahmood, Pak. J. Engg. & Appl. Sci. Vol 10, 57 (2012).

[13] A. M. Zeki, M. S. Zakaria, International Islamic University Malaysia (IIUM), Kuala Lumpur, Malaysia, National University of Malaysia (UKM), Bangi, Selangor, Malaysia (2008).

[14] A. Amin, Handbook of character recognition and document image analysis pp. 397–420 (1997).

[15] R. A. M. Alsaidi, H. Li, Y. Wei, R. Khaji, Y. Y. Tang, International Journal of Wavelets, Multiresolution and Information Processing 11 (2013).

[16] H. Li, Y. Wei, L. Li, Y. Yuan, Neurocomputing 83, 110 (2012).

[17] R. Rubinstein, M. Zibulevsky, M. Elad, CS Technion (2008).

[18] V. M¨argner, H. El Abed, M. Pechwitz, et al., Actes du 9`eme Colloque International Francophone sur l'Ecrit et le Document (2006), pp. 259–264.

[19] A. Lawgali, A. Bouridane, M. Angelova, Z. Ghassemlooy, International Journal of Advanced Science and Technology 34, 1 (2011).

[20] A. A. Aburas, S. A. Rehiel, Advances in Robotics, Automation and Control ISBN pp. 78–953.

[21] P. Xiu, L. Peng, X. Ding, H. Wang, Document Analysis Systems VII (Springer, 2006), pp. 402–412.