Investigating of Preprocessing Techniques and Novel Features in Recognition of Handwritten Arabic Characters

Ahmed T. Sahlol[1,*], Ching Y. Suen[2],
Mohammed R. Elbasyoni[1], and Abdelhay A. Sallam[3]

[1] Computer Teacher preparation Dept., Damietta University, New Damietta, Damietta, Egypt
asahlol@encs.concordia.ca, mmrefaat@hotmail.com
[2] (CENPRMI) Centre for Pattern Recognition and Machine Intelligence, Concordia University, Canada
parmidir@encs.concordia.ca
[3] Department of Electrical Engineering, Port-Said University, Port-Said, Egypt
aasallam@ucalgary.ca

Abstract. There are many difficulties facing a handwritten Arabic recognition system such as unlimited variation in character shapes. This paper describes a new method for handwritten Arabic character recognition. We propose a novel efficient approach for the recognition of off-line Arabic handwritten characters. The approach is based on novel preprocessing operations, structural statistical and topological features from the main body of the character and also from the secondary components. Evaluation of the importance and accuracy of the selected features was made. Our method based on the selected features and the system was built, trained and tested by CENPRMI dataset. We used SVM (RBF) and KNN for classification to find the recognition accuracy. The proposed algorithm obtained promising results in terms of accuracy; with recognition rates of 89.2% for SVM. Compared with other related works and also our recently published work we find that our result is the highest among them.

Keywords: Arabic OCR, Noise removal, Secondaries.

1 Introduction

The Arabic alphabet is used by a wide variety of languages besides Arabic (especially in Africa and Asia) such as Persian, Kurdish, Malay and Urdu. Thus, the ability to automate the interpretation of written Arabic would have widespread benefits. The calligraphic nature of the Arabic script is distinguished from other languages in several ways.

Optical character recognition (OCR) problems can be distinguished into two domains. Off-line recognition; which deals with the image of the character after it inputs to the system for instant scanning. On-line recognition which has different input way,

[*] Corresponding author.

N. El Gayar et al. (Eds.): ANNPR 2014, LNAI 8774, pp. 264–276, 2014.

where the writer writes directly to the system using, for example, light pen as a tool of input. These two domains (offline & online) can be further divided into two areas according to the way that the character itself has been written (by hand or by machine) to handwritten or printed character .In this paper we deal with the Off-line handwritten OCR. Offline recognition of handwritten cursive text is more difficult than online recognition because more information is available in online recognition, like the movement of the pen may be used as a feature of the character, on the contrary the Offline recognition systems must deal with two-dimensional images of the text after it has already been written. Although there are a few commercial Arabic OCR systems for printed text (like Sakhr, IRIS, ABBYY, etc.), there is no commercial product for handwritten Arabic OCR available in the market.

There are many other applications for analysis of human handwriting such as writer identification and verification, form processing, interpreting handwritten postal addresses on envelopes and reading currency amounts on bank checks etc. The main problems encountered when dealing with handwritten Arabic characters are:

– The letters are joined together along a writing line. This big difference between Arabic handwriting and English handwriting, is that the English characters are easier to separate but Arabic are not.
– More than half the Arabic letters are composed of main body and secondary components. The secondary components are letter components that are disconnected from the main body. That secondary component s should be taken into account by any computerized recognition system. Also the type and position of the secondary components are very important features of Arabic letters.
– Each character is drawn in three or four forms when it is written connected to other characters in the word depending on his position of the word. The same letter at the beginning and end of a word can have a completely different appearance.

Various approaches have been proposed to deal with this problem. Many approaches have been adopted in various ways to improve accuracy and efficiency.

In our literature review, we focus on offline Arabic handwritten characters. As for printed Arabic text recognition, some of the recently used techniques can be found in Benjelil et al. [1], Ben Cheikh et al. [2], Kanoun et al. [3], Khan et al. [4], Ben Moussa et al. [5], Prasad et al. [6], Saeeda and Albakoor [7], and Slimane et al. [8].

Also for Recent attempts for online recognition of Arabic characters can be seen in Kherallah et al. [9], [10], Mezghani and Mitiche [11], Saabni and El-Sana [12], and Sternby et al. [13].

Benouareth et al. [14] described an offline Arabic handwritten word recognition system based on segmentation-free approach and hidden Markov models.

Abandah et al. [15] extracted 96 features from the letter's secondary components, main body, skeleton, and boundary. These features are evaluated and best subsets of varying sizes are selected using five feature selection techniques. The evolutionary algorithm has the highest time complexity but it selects feature subsets that give the highest recognition accuracies.

Abdelazeem et al [16] used vertical and horizontal projections which gave more valuable information to capture the distribution of ink along one of the two dimensions in the character. Another kind of useful feature is topological features.

Aburas [17] presented new construction of OCR system for handwriting Arabic characters using the technique similar to that is used in wavelet compression.

The proposed algorithm obtained promising results in terms of accuracy (reaches 97.9% for some letters at average 80%) as well as in terms of time consuming.

Bluche and Ney [18] and [19].made a combination of a convolutional neural network with a HMM gave better results compared with recurrent neural networks, instead of using only HMM in [20].

Prum et al [21] introduced a novel discriminative method that relies, in contrast, on explicit grapheme segmentation and SVM-based character recognition. In addition to single character recognition with rejection, bi-characters are recognized in order to refine the recognition hypotheses. In particular, bi-character recognition is able to cope with the problem of shared character parts. Whole word recognition is achieved with an efficient dynamic programming method similar to the Viterbi algorithm.

Chowdhury et al [22] formulated a distance function based on Levenshtein metric to compute the similarity between an unknown character sample and each training sample. He studied also the effect of pruning the training sample set based on the above distance between individual training samples of the same character class. The proposed approach has been simulated on different publicly available sample databases of online handwritten characters. The recognition accuracies are acceptable.

Chherawala et al [23] built a recognition model is based on the long short-term memory (LSTM) and connectionist temporal classification (CTC) neural networks. This model has been shown to outperform the well-known HMM model for various handwriting tasks, In its multidimensional form, called MDLSTM, this network is able to automatically learn features from the input image. The IFN/ENIT database has been used as benchmark for Arabic word recognition, where the results are promising. A more recent survey on Arabic handwritten text recognition can be found in was presented in [24].

The goal of this work is to develop a reliable offline OCR system for handwritten Arabic characters. In order to overcome the writing variations described before:

First we make different kind of noise removal then we used different kind of features (Whole body features, Main body features and Secondary component features):

Support vector machine and K-nearest neighbor are then used to classify the characters based on the features that were extracted from the input character. Figure 1 summarizes the methodology adopted in this paper.

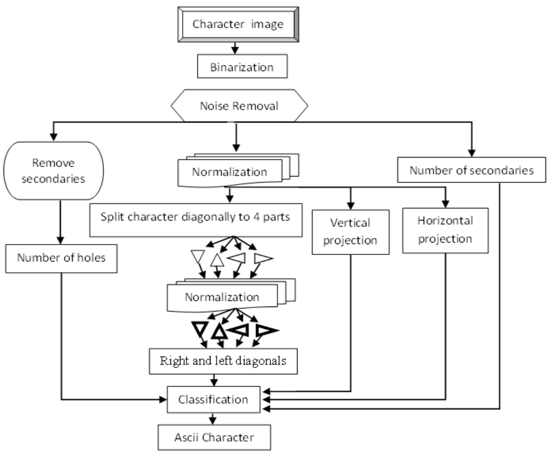


Fig. 1. The proposed method

2 Materials and Methods

2.1 Binarization

We use Otsu's method [25] to convert the grey character image to a binary image which is a normalized intensity value that lies in the range [0, 1]. We don't use this default value of binarization (0.5) because by experiment we find out that useful information have been lost from the character, so we compute the level of intensity for each character and then replace all pixels in the input image with luminance greater than level with the value 1 (white) and replaces all other pixels with the value 0 (black).

2.2 Noise Removal

Although that noise removal techniques have the effect of slightly distorting the actual image, but often this is a small price to pay for the removal of distracting noise and also we were so circumspect when choosing suitable techniques and their parameters.

We remove from character all connected components (objects) that have fewer than 5 pixels. By experience we find that less than 5 connected pixels can be determined as noise and this operation has no bad effect on character main shape or any secondary components.

Median Filtering

Median filtering [26] is an image processing filter used to reduce the effects of random noise. We adopt a 3 × 3 median filter was because it gave us the best result.

Dilation [27]

It is an operation that grows or thickens objects in a binary image the specific manner and extent of this thickening is controlled by a shape referred to as a structuring element. In this paper we use a square of 2x2 of ones as a structuring element as it gives us the best dilation job.

Morphological Noise Removal

- Filling: fill isolated interior pixels, for each pixel p if the number of non-zero neighbors are 7 pixels.
- Cleaning: Remove isolated pixels. For each pixel p if all neighbors are zeros.
- Adjacent neighbors and Diagonal neighbors: For each pixel we check each pixel diagonal ({(x−1,y−1 (x−1,y+1), (x+1,y+1), (x+1,y−1)}) and adjacent ({(x−1,y), (x,y+1), (x+1,y), (x,y−1)}) neighbors.

If three from its four neighbors are zeroes, so it become zero.

2.3 Normalization

Size normalization is an important pre-processing technique in character recognition because the character image is mapped onto a predefined size so as to give a representation of fixed dimensionality for classification.

We use the Linear Backward mapping method [28].

2.4 Feature Extraction

We divide our features into 3 groups in terms of the kind of information we want to extract:

Features from Whole Character (main body and secondaries)

Vertical and Horizontal Projections.

Vertical profile is the sum of white pixels perpendicular to the y axis. Similarly, the horizontal projection profile is sum of black pixels but it is perpendicular to the x axis.

Right and Left Diagonal of Each Part of the Four Triangular Character Parts.

We divide each character into four triangular and crop each part by determining the boundaries for the last non-zero pixel as shown in Figure 1.

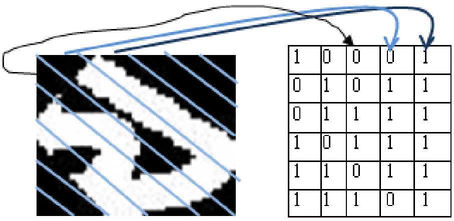
Fig. 2. a. Upper triangle b. Right triangle c. Left triangle d. Lower triangle

Then we get the right and the left diagonal for each triangle of the character by:

The columns of the first output matrix contain the nonzero diagonals of the character. See Figure2 "the dark blue arrow".

The longest nonzero diagonal in the character is determined.

For the nonzero diagonals below the main diagonal of the character, extra zeros are added at the tops of columns.



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

Fig. 3. a. Right diagonal of "Haa" b. Matrix for non-zero diagonals

Number of Secondaries.
This feature recognizes the connected components and number of them like Hamza and dots.

We use the connected component labeling techniques [29].

We identify the main body easily as it is usually the largest component so any other connected components are considered as secondaries.

Features from Only Main Body of the Character
Which represent only the character body without any secondaries.

Number of Holes.
We see that this feature can give accurate results if we eliminate all secondaries with the character correctly. So we keep only the main body for this feature and remove any secondaries connected with the character:

We use again the algorithm [29] but in another way:

1. After getting labels of all connected components.
2. Sort the secodaries by size so that the largest is the first.
3. Keep only the largest connected-components; at this point we eliminate all secondaries.
4. Trace the boundaries of holes inside the character by using Moore-Neighbor tracing algorithm modified by Jacob's stopping criteria [30].

By experiments we find some defects resulted from quick hand written as shown in the figure below

Fig. 4. Number of holes for “Haa” a. Num =1 b. Num =2 c. Num = 3 d. Num =4

Although the above figures represent the same character (ـه) after preprocessing yet we unfortunately extract different number of holes from each of them.

Feature from the biggest secondary component of the character (dots and hamzas).

Position of Secondaries.
As we said before that secondary position is the only way to distinguish between a character and another. Those groups of characters (ج , خ ,نـ , بـ , يـ, تـ) can be distinguished by machine or by human eye only by the position of the secondary component

We utilize again the connected component labeling techniques [29].

We get easily the largest component so any other connected components are considered as secondaries then eliminate the largest component which is considered the main body of the character then eliminate all the other smallest components after sorting them by size so we start by the smallest, except for the last one B (which is considered the big connected component after the main body) then determine the row and the column for this component, then divide height of B by width of B to get the height/width feature, then count the total number of white and black pixels of B to get Density feature.

Normalization of Feature Data: The attribute data which might have different ranges (min to max) is scaled to fit in a specific range 0, 1. We use Min–max method [31] for normalization.

Handwritten Arabic Characters Dataset: Our database of handwritten Arabic samples is CENPRMI dataset [32]. It includes Arabic off-line isolated handwritten characters. The database contains 11620 characters. These characters were written according to 12 different templates by 13 writers, with each template adopted by 5–8 writers.

2.5 Classification

Support Vector Machine

The Support Vector Machine (SVM) was proposed by Vapnik in [33]. SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other classes. The best hyperplane for an SVM means the one with the largest margin among the classes. The RBF kernel is a measure of similarity between two examples (training and testing data).We use SVM package called LIBSVM [34].

The SVM uses RBF kernel parameters C and γ where C (cost) is a regularization parameter which controls the penalty for imperfect fit to training labels, and gamma (γ) controls the shape of the separating hyperplane. Increasing gamma usually increases number of the support vectors. Using grid search several experiments were carried out. After several trials of tuning parameters we find that c= 12 and gamma parameter γ= 0.04 give the best results; Accuracy = 89.2 %. Figure 4 below shows the relation between gamma value and recognition rate.

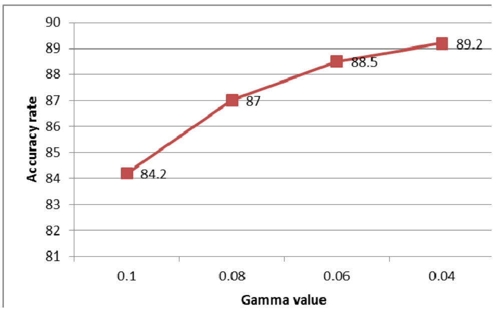


Fig. 5. The relationship between gamma γ and the classification rate

It is obvious from the previous figure that Gamma γ affects significantly in the classification rate.

K-Nearest Neighbor (k-NN) [35]
K-NN Calculates distances of all training vectors and picks k closest vectors Calculate average/majority. Classification using an instance-based classifier can be a simple matter of locating the nearest neighbor in instance space and labeling the unknown instance with the same class label as that of the located (known) neighbor.

The k-nearest-neighbor classifier is commonly based on the Euclidean distance between a test sample and the specified training samples. More robust models can be achieved by locating k, where k > 1, neighbors and letting the majority vote decide the outcome of class labeling. A higher value of k results in a smoother, less locally sensitive, function. We tried to tune neighbor parameter on our Arabic handwritten database and from the figure below we can see the effect of number of neighbors on accuracy.

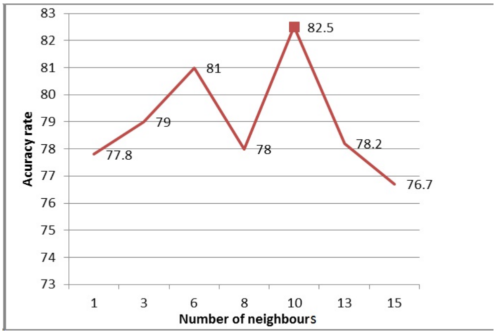


Fig. 6. The relationship between number of neighbors and recognition rate

As shown above. K=10 gives us the best results, but more than 10 neighbors the results get worst.

3 Results and Discussions

The recognition results of KNN were compared to those of the SVM classifier. Table I shows the recognition rate of KNN and SVM.

Table 1. Comparison between the Used Classifiers

Classifier	KNN	SVM (RBF)
Accuracy Rate (%)	82.5	89.2

Table II shows summarized the most recent work in the isolated handwritten Arabic characters. As we can see they are sorted according to the published date. The data used, the feature set and the improvement of recognition rate can also be seen.

Table 2. Comparison between Previous Results and Ours

Previous study	Approach	Results
A.A. Aburas et al [36]	Haar Wavelet transform	70%
M. Z. Khedher, et al [37]	Not mentioned	73.4%
G. Abandah et al [15]	Combination of multi-objective genetic algorithm and SVM	not mentioned exactly
A. T. Al-Taani et al [38]	Decision tree	75.3%
G. A. Abandah et al [39]	Linear Discriminant Analysis	87%
Our previous work [40]	Feed forward neural network.	88%
Proposed method	SVM	**89.2%**

From the previous table, it is obvious that our system does the best when compared with other systems in terms of recognition rate, although other systems make great contributions especially in terms of accuracy and using of modern classification techniques.

The main contribution of this research includes building of a new offline Arabic handwritten character recognition system which is developed based on the novel extracted feature after some new techniques of preprocessing operations. The evaluation of our system is done by applying those features on SVM as well as KNN. The proposed method obtained competitive accuracy rates at 89.2%.

The results illustrate that higher recognition accuracies are achieved using the proposed feature extraction technique. The proposed method (by SVM) gives a recognition rate of about 100 % for (هـ , ح, ج, خ, ل, م, ن, ت , ي, أ).

The worst recognized characters was (ف) by SVM and also by KNN. It was misclassified as (ن and ق) this is because the similarities between those two characters in some writing styles and also they all have upper secondaries and holes. The second misclassified character is (ع) was misclassified as (ح) by SVM as well as by KNN. We think that this is because the similarities in their shapes especially at the lower part. The third misclassified character is (س) was misclassified as (ص) by SVM as well as by KNN. We think that this is because the similarities in their left part.

We think that preprocessing operations as well as selecting most proper feature can minimize classification error. For example we use different kinds of noise removal (statistical and morphological) for erasing useless parts of the character which can occur during hand writing process, ink stain or even by digitizing the image. We make also dilation for fixing damaged pixels of the character occurred as a result of preprocessing operations (binarization- noise removing) or during the digitizing process. Any of those preprocessing operations could have bad effects on character shape if they don't used properly and this can reflect on the quality of the extracted features for example if we overuse of noise removal techniques we can easily remove a dot if it was written slightly and consequently we lose very important information of this character dots.

We extract features from the whole character, as well as, its main body and secondary components themselves which provide more valuable features that exploit the recognition potential of the secondary components of handwritten Arabic letters. These results also confirm the importance of the secondary components of the handwritten Arabic characters. For example if we make a comparison between س and ش, ص and ض we will find no differences between each pair of them except for the secondary component.

We use not only different kinds of features (structural features, statistical features, topological features) which represent different aspects of the character's characteristic, but also (after many trials) we choose the most significant features for distinguishing between characters. After careful examination of the samples that were incorrectly recognized, we concluded that most of these samples are hard to recognize by native or even by a human expert reader. However, we think that the door is open to search for extracting new features that capture subtle differences in loop shapes and secondary types.

4 Conclusion

This paper presents a novel approach for extracting features to achieve high recognition accuracy of handwritten Arabic characters. We tune the used parameters during the preprocessing phase including binarization, normalization and some noise removal methods accurately to preserve all useful information that can be extracted from the character.

Selecting proper features for recognizing handwritten Arabic characters can give better recognition accuracies, therefore we included statistical, morphological and topological features. Also we pay more attention to the secondaries like secondary position, ratios and density because we think that may overcome some of handwritten characters variations. Although, there are some challenges with some characters, the overall recognition rate is encouraging especially when compared to other handwritten Arabic character systems.

After examining the recognition accuracy of each character using SVM and KNN we found that the best accuracy is given by SVM which is 89.2%. The other misrecognized characters such as (ـف, ع) we think that this is because those characters similarities between those characters and others in some writing styles and also they have secondaries and holes. Our future work includes increasing the efficiency of the proposed approach especially for the characters that were not recognized well by finding out more powerful features, also including variations in writing the main body of the character and also the secondaries.

References

1. Benjelil, M., Kanoun, S., Mullot, R., Alimi, A.M.: Arabic and latin script identification in printed and handwritten types based on steerable pyramid features. In: Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 591–595 (2009)

2. Cheikh, I.B., Bela, A., Kacem, A.: A novel approach for the recognition of a wide Arabic handwritten word lexicon. In: Proceedings of the 19th International Conference on Pattern Recognition (ICPR), pp. 1–4. IEEE (2008)
3. Kanoun, S., Slimane, F., Guesmi, H., Ingold, R., Almi, A.M., Hennebert, J.: Affixal approach versus analytical approach for off–line arabic decomposable vocabulary recognition. In: Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 661–665 (2009)
4. Khan, T.K., Azam, S.M., Mohsin, S.: An improvement over template matching using k-means algorithm for printed cursive script recognition. In: Proceedings of the 4th IASTED International Conference on Signal Processing, Pattern Recognition, and Applications, pp. 209–214 (2007)
5. Benmoussa, S., Frissard, Q., Zahour, A., Benabdelhafid, A., Alimi, A.M.: New features using fractal multi-dimensions for generalized Arabic font recognition. Pattern Recogn. Lett. 31(5), 361–371 (2010)
6. Prasad, R., Saleem, S., Kamali, M., Meermeier, R., Natarajan, P.: Improvements in hidden markov model based Arabic ocr. In: Proceedings of the 19th International Conference on Pattern Recognition, ICPR (2008)
7. Saeeda, K., Albakoor, M.: Region growing based segmentation algorithm for typewritten and handwritten text recognition. Appl. Soft Comput. 9(2), 608–617 (2009)
8. Slimane, F., Ingold, R., Kanoun, S., Alimi, A.M., Hennebert, J.: A new Arabic printed text image database and evaluation protocols. In: Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 946–950 (2009)
9. Kherallah, M., Haddad, L., Alimi, A.M., Mitiche, A.: On-Line handwritten digit recognition based on trajectory and velocity modeling. Pattern Recogn. Lett. 29(5), 580–594 (2008)
10. Kherallah, M., Bouri, F., Alimi, A.M.: On-line Arabic handwriting recognition system based on visual encoding and genetic algorithm. Engin. Appl. Artif. Intell. 22(1), 153–170 (2009)
11. Mezghani, N., Mitiche, A.: A gibbsian kohonen, “Network for online arabic character recognition”. In: Bebis, G., et al. (eds.) ISVC 2008, Part II. LNCS, vol. 5359, pp. 493–500. Springer, Heidelberg (2008)
12. Saabni, R., El-sana, J.: Hierarchical on-line Arabic handwriting recognition. In: Proceedings of the 10th International Conference on Document Analysis and Recognition (ICDAR), pp. 867–871 (2009)
13. Sternby, J., Morwing, J., Andersson, J., FRriberg, C.: On-Line Arabic handwriting recognition with templates. Pattern Recogn., New Frontiers Handwrit. Recogn. 42(12), 3278–3286 (2009)
14. Benouareth, A., Ennaji, A., Sellami, M.: Arabic Handwritten Word Recognition Using HMMs with Explicit State Duration. Journal on Advances in Signal Processing 2008, 1–13 (2008)
15. Abandah, G.A., Anssari, N.: Novel moment features extraction for recognizing handwritten Arabic letters. Journal of Computer Science 5(3), 226–232 (2009)
16. Abdelazeem, S., EL-Sherif, E.: Arabic handwritten digit recognition. International Journal on Document Analysis and Recognition (IJDAR) 11(3), 127–141 (2008)
17. Aburas, A.A., Rehiel, S.M.: Off-line Omni-style handwriting Arabic character recognition system based on wavelet compression. In: Arab Research Institute in Sciences & Engineering ARISER, pp. 123–135 (2007)

18. Bluche, T., Ney, H., Kermorvant, C.: Feature extraction with convolutional neural networks for handwritten word recognition. In: 12th International Conference on Document Analysis and Recognition, pp. 285–289 (2013)
19. Kozielski, M., Doetsch, P., Ney, H.: Improvements in RWTH's system for off-line handwriting recognition. In: 12th International Conference on Document Analysis and Recognition, pp. 935–939 (2013)
20. Rothacker, L., Vajda, S., Fink, G.A.: Bag-of-features representations for offline handwriting recognition applied to Arabic script. In: Proceedings of the 3rd International Conference on Frontiers in Handwriting Recognition (ICFHR), Bari, Italy, pp. 149–154 (2012)
21. Prum, S., Visani, M., Fischer, A., Ogier, J.M.: A Discriminative Approach to On-Line Handwriting Recognition Using Bi-Character models. In: 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 364–368 (2013)
22. Chowdhury, S.D., Bhattacharya, U., Parui, S.K.: Online Handwriting Recognition Using Levenshtein Distance Metric. In: 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 79–83 (2013)
23. Chherawala, Y., Roy, P.P., Cheriet, M.: Feature design for offline Arabic handwriting recognition: handcrafted vs automated?. In: 12th International Conference on Document Analysis and Recognition (ICDAR), pp. 290–294 (2013)
24. Parvez, M.T., Mahmoud, S.A.: Offline Arabic handwritten text recognition: A Survey. Journal of the Association for Computing Machinery 45(2), 23:1–23:35 (2013)
25. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. IEEE Transactions on Systems, Man, and Cybernetics 9(1), 62–66 (1979)
26. Jae, S.L.: Two dimensional signal and image processing. PRENTICE HALL PTR, Upper Saddle River, New Jersey, 07458 (1990)
27. Rosenfeld, A., Kak, A.: Digital Picture Processing, 1st edn. Academic Press, New York (1976) ISBN: 10: 0125973608
28. Cheriet, M., Kharma, N., Liu, C.L., Suen, C.Y.: Character Recognition Systems, pp. 36–38. John Wiley (2007)
29. Rosenfeld, A., Kak, A.: Digital Picture Processing, 1st edn. Academic Press, New York (1976) ISBN: 10: 0125973608
30. Gonzalez, R.C., Woods, R.E., Eddins, S.L.: Digital Image Processing Using MATLAB. Gatesmark Publishing (2009)
31. Hann, J., Kamber, M.: Data Mining: concepts and techniques, 3rd edn. Morgan Kaufman (2011)
32. Alamri, H., Sadri, J., Suen, C.Y., Nobile, N.: A novel comprehensive database for Arabic off line handwriting recognition. In: The 11th International Conference on Frontiers in Handwriting Recognition (ICFHR), pp. 664–669 (2008)
33. Vapnik, V.: Statistical Learning Theory. Wiley, Chichester (1998)
34. Hsu, C.W., Lin, C.J.: A comparison of methods for multi-class support vector machines. IEEE Trans. Neural Network 13, 415–425 (2002)
35. Mitchell, T.: Machine Learning. McGraw-Hill, New York (1997)
36. Aburas, A.A., Rehiel, S.M.: Off-line Omni-style handwriting Arabic character recognition system based on wavelet compression. In: Arab Research Institute in Sciences & Engineering ARISER, pp. 123–135 (2007)
37. Khedher, M.Z., Abandah, G.A., Al-Khawaldeh, A.M.: Optimizing feature selection for recognizing handwritten Arabic characters. World Academy of Science, Engineering and Technology 4, 81–84 (2005)

38. Al-Taani, A.T., Al-Haj, S.: Recognition of on-line Arabic Handwritten Characters Using Structural Features. Journal of Pattern Recognition Research, 23–37 (2010)
39. Abandah, G.A., Younis, K.S., Khedher, M.Z.: Handwritten Arabic character recognition using multiple classifiers Based on letter form. In: Conf. on Signal Processing, Pattern Recognition& Applications, Austria, pp. 128–133 (2008)
40. Sahlol, A.T., Suen, C.Y., Sallam, A.A., Elbasyoni, M.R.: A proposed OCR Algorithm for cursive Handwritten Arabic Character Recognition. Journal of Pattern Recognition and Intelligent Systems (PRIS), 90–104 (2014)