

Isolated Arabic Hand Written Letters Recognition Based on Contour Matching and Neural Network

¹Bajeszeyadaljunaecidia, ²Mutasem Shabib Alkhasawneh and ¹Mohammad Ali BaniYounes

¹Computer Science Department, Faculty of Information and Technology,

²Software Engineering Department, Faculty of Information and Technology,
Ajloun National University, P.O. Box 43, Ajloun 26810, Jordan

Article history

Received: 07-08-2018

Revised: 04-09-2018

Accepted: 19-11-2018

Corresponding Author:

Bajes Zeyad Aljunaecidi
Department of Computer
Science, Faculty of Information
and Technology, Ajloun
National University, Jordan
Email: bajesaljunadei@gmail.com

Abstract: Complexity of Arabic writing language makes its handwritten recognition very complex in terms of computer algorithms comparing with other languages such as English and French. The Arabic handwritten recognition has high importance in modern applications. The contour analysis of word image can extract special contour features that discriminate one character from another by the mean of vector features. This paper aims to implements a set of pre-processing functions over a handwritten Arabic character, with contour analysis, to enter the contour vector to neural network to recognize it. For training part, the neural network architecture was trained using many patterns regardless of the Arabic font style building a rigid recognition model. The presented algorithm structure got recognition ratio about 97%.

Keywords: Arabic Language, Handwritten Recognition, Contour Analysis, Recognition, Neural Networks

Introduction

Human activities digitizing demands is accelerating and the computer era moving to internet of things (IoT) increases the demand of automatic user applications. The functionalities are conveying by collaborative technologies inventing and development between computer and human. Such rising scheme increasing the demand and growth of researches that are focusing on handwritten recognition.

Handwritten recognition is taken place and main role in modern computer systems. It becomes very wide in smart mobile phones, archiving systems, scanning systems, etc. Arabic language has complex context style, writing methodology, font and texture. This makes the mission of computer system not easy to solve. In comparison with English language, English has restricted work style and character shape (Parrweej, 2012). Writing English letters in upper and lower case makes different shapes of the same character, but the variety still restricted. Arabic had wide variety of writing styles for the same character, while the connected characters cursive style moves the character between completely different shapes with respect to its location in the word (Abdullah *et al.*, 2012). Also, Arabic words could has hidden characters that are pronounced but not written and the reader can expect such characters by the

texture of the language context. In comparison with Arabic, English is uni-directional writing language where it overall read from left to write, but Arabic is not. Arabic normally read from right to left but the numbers should be written from left to right (Boukerma and Farah, 2012; Bharath and Madhvanath, 2012).

Arabic language has many features make the automatic recognition by computer system to be not an easy task to perform in high accuracy (Boukerma and Farah, 2012). Figure 1 illustrates a sample of hand written Arabic words with hidden characters, while the word can hold more than one of them.

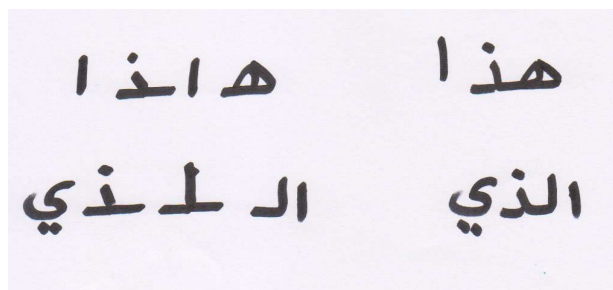


Fig. 1: Arabic words with hidden characters. The words as it are normally written (right). The containing characters within those words (left)

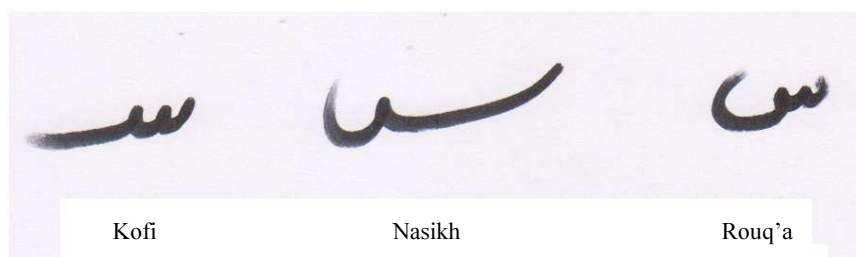


Fig. 2: Sample Arabic character “Seen” with different writing font styles. The character in top and the writing font style under it

More than seven hundred million people are writing in Arabic characters, where different languages are written in Arabic characters in including Arabic language itself, Persian, Urdu and other (Al-nuzaili *et al.*, 2012). The difference between Semitic languages and modern European languages could be described in several points including the complexity of the linguistic structure, hidden implicit characters, concatenation of characters, writing direction, font, writing style and the most issue with respect to computer system is the un-specific writer style. This means, different people can write the same word (and also same character) in different styles where those different styles, without undergoing to the original writing styles. Figure 2 shows one sample character that is written in three different styles of Arabic fonts (this character pronounced as “ss” and named “seen”). The writing style font name is written under each sample (Kacem *et al.*, 2012; Asi *et al.*, 2012).

The motivations toward working with Arabic handwritten character recognition leads to start and present this paper. The contribution of this paper is to achieve high recognition result of single Arabic handwritten character recognition by implementing a contour analysis of the character image and neural network recognition. Applying contour analysis and neural network matching/recognition to a pre-processed characters image as expectation state that using this approach permits achieving high recognition ratio due to the ability of the proposed algorithm at treating the general shape of the character. The methodology can handle the variety of the writers’ details and hand sketches (Bharath and Madhvanath, 2012). Furthermore, Neural network is being inspiring researchers for the last few decades. Beside the various application on the Hand Written Character Recognition, it was applied in Landslide detection (Alkhasawneh *et al.*, 2013; 2014). In addition to that Neural network field is very fast growing area such as (Alkhasawneh and Tay, 2017).

The introduction for this work was introduced in this section, section 2 provided a work related to this paper. The data set details was provided in section 3. The methodology of this work provided in section 4, while the results and discussion provided in section 5. Section 6 and 7 have the conclusion and references in continuously.

Literature Review

Arabic handwriting recognition is a significant difficulty, which requires lots of researches and development considerations in order to construct theoretical and realistic concepts and outcomes for the language writing structure with respect to digital signal processing and computer system. The following is discussion of some related up to date researches related to handwritten recognition illustrating main points of study field.

Jayech *et al.* (2016), presented a new graphical model based on probability I order to model and recognize the Arabic handwritten words that represents Tunisian cities names. The tools used in this research are all based on the Bayesian networks hierarchy. The main goal of this study is to conduct the optimal model for Arabic handwritten recognition whose recognition process complexity is reduced. The results then tested using IFN/ENIT database and it was very promising. This study is useful for the research to increase the aware of much more literatures that investigates the Arabic handwritten words recognition regardless to the proposing approaches and techniques used.

Parvez and Mahmoud (2013), presented the results of their research for the off-line Arabic handwriting recognition by structural techniques. Statistical technique is well known to report the research in Arabic handwritten recognition. However, the Structural technique stayed uninhabitable widely. Both of structural and statistical techniques could be efficiently incorporated in multi-classifier systems that implemented by a number of procedures and techniques. An Arabic text line was divided to number of words/sub-words and multiple dots were extracted. An adaptive slant correction method has been shown to correct various slant angles of the various partitions of a text line. This study proposed new techniques to recognize the Arabic handwritten words and sub words, which make it valuable to document it in the research related works regardless of all its differences with the research methods and techniques.

Hamdani *et al.* (2014), introduced the CART trees to use it in the state tying. Taken in concentration a set of questions, those are very important in the CART tree that

creation by a “loss mapping” compartmentalization for the Arabic shapes. The system that used for validation process was hybrid that because they combined the Recurrent Neural Networks and Hidden Markov Models in one technique. However, the researchers noticed a clear differentiation between them that the Hidden Markov depended on CART tree labels, but the other one used the baseline label. They relied on Open Hart database to test their data set, thus they attained 2.9% ratio in the Word Error Rate term. This study used HMM which can be related to research used methods but with several differences in both the other techniques and methods used and the sample they used to apply their methods over.

El Qacimy *et al.* (2015), provided a brief review for the feature extraction methods to the offline handwritten Arabic text recognition in addition to presenting both description for the used database and the recognition rate in the proposed approaches in this study. However, it will be such a quite profit to the research discussion and methodology because it includes a fully discussion for the methods of feature extraction that will help in the contour extraction section in our research.

Krayem *et al.* (2013), presented a system of a word recognition which neither based on any segmentation processes nor the detection of baseline descended and ascended requirements. The idea of their system was to build a novel holistic recognizer of Arabic printed words using the discrete hidden Markov classifier and the block-based discrete cosine transform. The built database was consisting of balanced word-image to get even samples of words distribution. The sample characteristics were Arabic handwritten words within five different fonts with a fourteen points of size in a plain style. This study considered critical to this research subject because its system also was trained and tested on an actual scanned images of handwritten words and used the hidden Markov model to build their recognizer but with one difference that they used also the block based discrete cosine transform. They concluded that the system they demonstrated is multiple recognition and the outcomes were promising because it achieved a high accurate values of 97.65 percent which is compared to other studies are higher and better. These results can be useful to compare our research results with.

Gargouri *et al.* (2013), used a novel framework that solving the problems of Arabic handwritten recognition process based on word as primeval, many statistic and dynamic characteristics set used in various stages of the word. Such as the stroke, the space between strokes and

the whole word were extracted from the point, to estimate the framework they used the method of Support Vector Machine and Dynamic Time Warping (DTW) and they choose set one from the ADAB database (Arabic Database). This study overview different methods for recognition to also different sample rather than the research sample and methods of recognition but still related to our research subject.

Data Set

The input handwritten characters are considered to be images that need to be processed and manipulated before applying the main processing phases (i.e., contour analysis and neural network recognition). So, the pre-processing phase will be done over images.

Arabic characters are dot based characters. This means, some characters could be written in similar way but one or more dots should be used to recognize between them. Table 1 show a sample of five characters that has the same skeleton but it varies with number and location of dot(s), the pronunciation in English is illustrated on the left of each character where the character itself is located on the right of the table.

Arabic characters are 28 characters those are all listed in Table 2. All those 28 character are involved in the presented algorithm, so, the training was done over different samples of each character shown in Table 1 as will be illustrated in section 5. Indeed, this paper focuses only on the isolated letters.

Table 1: Sample Arabic handwritten characters and it’s pronunciation in English. Arabic character (right), pronunciation in English characters (left)

charcter	Isolated	Initial	Midle	End	Final
Alif	ألف	ا	ا	ا	ا
Ba’	باء	ب	ب	ب	ب
Ta’	تاء	ت	ت	ت	ت
Tha’	ثاء	ث	ث	ث	ث
Jeem	جيم	ج	ج	ج	ج
Ha’	حاء	ح	ح	ح	ح

Table 2: Arabic characters list

أ	ز	ق
ب	س	ك
ت	ش	ل
ث	ص	م
ج	ض	ن
ح	ط	و
خ	ظ	هـ
د	ع	ي
ذ	غ	
ر	ف	

The neural network training requires historical data that matches specific input to a specific output target (it will be discussed in section 5) and it needs more than one sample to enable network training. In fact, a lot of sample is needed to build a precise neural network. Also, the character could handle few changes in some details if it was two times by the same person, but it will handle large changes and more details if it was written two times by two different persons (different hands). This makes the dataset required for training and testing larger. Ten Arabic native speakers helped in creating the dataset, each person wrote the same character 10 times. Thus, the total dataset becomes 100 sample of each character, each 10 of them has been written by the same person. The total dataset is 28 character times 10 persons time 10 samples of each person, the total is 2800 samples, similar to those are shown in Table 2.

Multi-layer Neural Network

In the wide field of prediction and classification problems MLP is the first researches choice. The popularity of MLP gained from its small structure size, computational simplicity, stability and its finite parameterization (Alkhasawneh *et al.*, 2013). Most of the time, three layers are conducted of MLP neural network. They are input layer, that usually has input as much as data features, secondly the hidden layer that set between the input and output lay, number of hidden layer is varied based on the how complicated is the problem (Alkhasawneh and Tay, 2017). Third layer is the output layer as shown in Figure 3. Neurons are the independent units consist every layer in the MLP. Neurons are

responsible of processing and information inside the MLP. The neurons in the different layers in the MLP are linked to each other through the weight. The relationship between the output and the inputs values determinations by alternating the weight values. Therefore, Adjusting the weights between the layers and the neurons is a difficult task, to come over this issue all known neural network used a learning algorithm. In addition to that learning algorithm is used to reduce the errors between the input and output (Alkhasawneh *et al.*, 2013).

Input layer has neurons as much as experiments data required. Every input can goes to one neuron in the input layer. Each input is multiplied by random weight called initial weight. Then the sum of the product processed by activation function, later one the result sends to the next layer. For one hidden layer of the MLP network as shown Fig. 3, the input and output of the j th neuron in the hidden layer is given by Equation 1 and 2 respectively:

$$net_j = \sum_{i=1}^I w_{ij} \cdot o_i \tag{1}$$

Hence, net_j denotes the input to the hidden layer, j and i are denotes the different neurons in the networks hidden and input layers respectively. I represent the size of the input vector, w_{ij} represents the weight passing between the input layer and the hidden layer and the inputs to the first layer denots by o_i . The inputs to the hidden layer neurons is net_j and uses it as the argument for a function and produces an output o_j given by:

$$o = f(net_j) \tag{2}$$

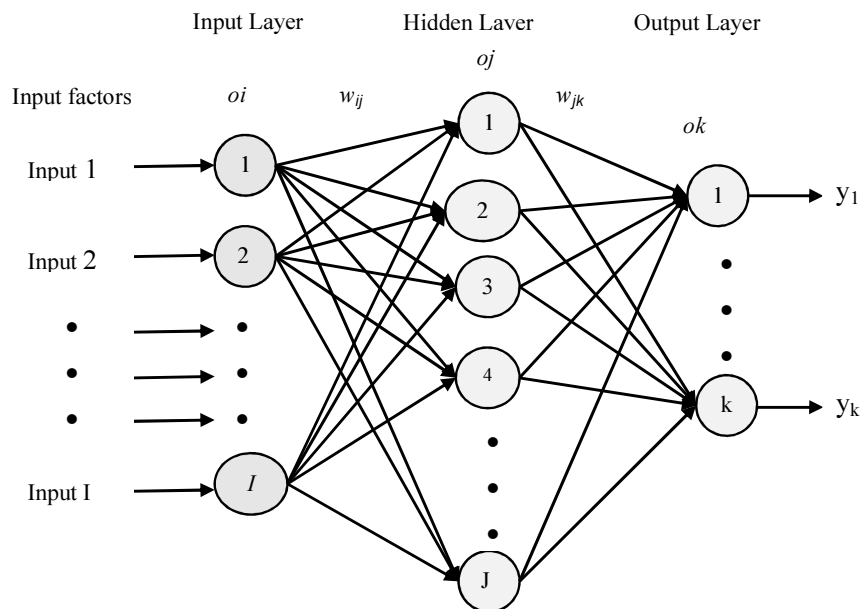


Fig. 3: Architecture of MLP neural network

The function used to sum the weight is non-linear sigmoid $f(*)$. Equation 3 can express the sigmoid function:

$$\tilde{f}(net_j) = f(net)(1 - f(net_j)) \quad (3)$$

MLP with one hidden layer the output is in Equation 4:

$$y_k = \sum_{j=1}^J w_{jk}(\tilde{f}(net_j)) \text{ for } 1 \leq j \leq J \quad (4)$$

The difference between the network output, o_k and the target output value, d_k is in Equation 5:

$$e_k = (d_k - o_k) \quad (5)$$

The sum of squared error it can be used to indicate the performance and can be calculated as in Equation 6:

$$E = \frac{1}{2} \sum_{k=1}^K e_k^2 \quad (6)$$

The number of neurons in the output layer represents by K . Adjusting the weights between layers can minimized the errors that propagated back through the MLP as expressed as in Equation 7:

$$\Delta w_{ij}(n+1) = \eta(\delta_j \cdot o_i) + \alpha \Delta w_{ij}(n) \quad (7)$$

where, α is the momentum coefficient, $\Delta w_{ij}(n+1)$ and $\Delta w_{ij}(n)$ are weight changes in epochs $(n+1)$ and (n) , η is the learning rate parameter, δ is an index of the rate of change in the error. This process of feeding forward and returning signals is repeated iteratively until the error of the network is as minimized as possible.

Experimental Setup

This section, introduces the experimental process which include collecting the data set, which includes three major steps viz., Preprocessing, Contour analysis and MLP Classifications.

Preprocessing

Proposing phase designed in module separately based on morphological operations whenever the module runs, it will bring the input from its past state (pure input image data), process them and produce the output prepared for the next module. The output is a binary image that is ready for contour analysis phase. Such module has the ability to solve any problems that may face on module without affecting the other next modules components. Separating phases in modular design simplifies the specifications and modification processes. As it is significant to design such an easily modifiable

research system that permits concepts changing, components of the research system modifying and alternate algorithms more than one time till the targeted objectives accomplished.

Contour analysis phase takes the input image from the previous pre-processing phase, generate contour vector data and prepare it for the next phase related to artificial neural network. The first two phases (i.e., pre-processing phase and contour analysis phase) are common for both paths; training and testing. The last phase represents the difference between the two paths. Where in training path, the neural network is being build and in the testing path, the neural network is being used to make a decision. The artificial neural network phase takes the input vector from the contour analysis phase and input it to the designed network either for training or for testing. In training stage, the input vector as well as the output target represent the input data to the network and it will goes into successive trails to set the entire values of neurons weights and biases. In comparison, in testing stage, the built network in the training stage is loaded and the input vector from contour analysis gives to it and let the network simulate it and output the recognition result and then it will be decided either which character is recognized, or nothing is recognized during the process.

It generally illustrates the main components those are described previously. The result of training path is the completed design artificial neural network that is loaded by the second path and used in recognition and decision making.

Figure 5 shows an Arabic character as an example of which could be processed by the techniques presented in this study. However, observing Fig. 4 which shows a scanned handwritten Arabic character, conducts several deficiencies such as; uneven ink amount all over the line which are completely non continuous width of the line itself and although this word was written with blue ink. User can write it with any other color including red, black and whichever potential ink color the proposed algorithm in this research is entirely based on contour analysis; therefore several pre-processing is a must till extracting the word contour.

Figure 6 demonstrates the preprocessing phase where the input raw character image may be gotten as expected in different ink colors, thickness and continuity. The first stage is to work with binary image, removing many of color effects (actually not all effects, because the intensity and color will affects the banalization process). The binarization in this study is being done in two steps, the first is to convert image into gray and the second step is to convert the gray image into black and white based on the average intensity of the image. Where the average intensity of the gray image is calculated and every pixel is examined, all pixels intensities lower than the average will be converted to black and otherwise will be converted to white pixels.

Contour Analysis

In comparison with handwritten in European languages, Arabic documents are more sensitive towards two noise types; speckle noise and salt and pepper noise, due to the mostly of Arabic characters whose having the feature of sharing the body of its words whose common to others as well as the one to three dots presence or absence distinguishing. Therefore, the dots in the handwritten documents in Arabic language potentially results in failures in terms of noise which in turn affects the quality of recognition

for this research work. Consequently, applying a proficient filtering technique is a must.

The handwritten Arabic documents quality may suffer consequently from noise caused throughout scanning processes or handwritten documents whose quality is bad to be distinguished correctly or successfully. Such noises can impact the other processes whose following this process and based to the quality of its resulted image. Hence, the noise removal process is critical algorithm to be performed for eliminating the noise of the scanned image based on the noise type of noise. Any noise type can be handled within the use of certain algorithms.

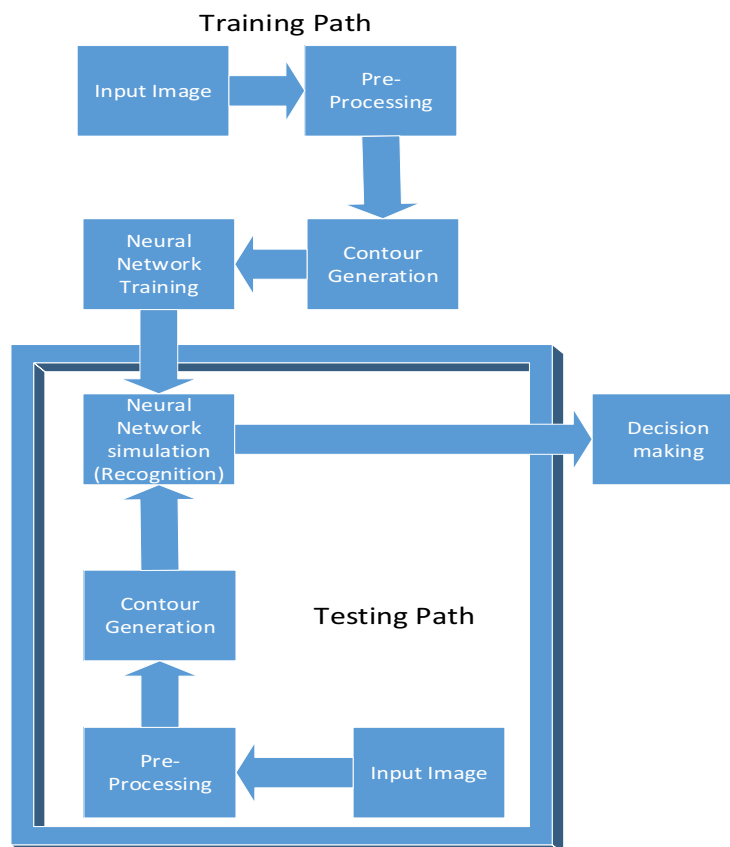


Fig. 4: Presented algorithms block diagram, showing both; training path and testing path



Fig. 5: Sample Arabic handwritten character to be processed

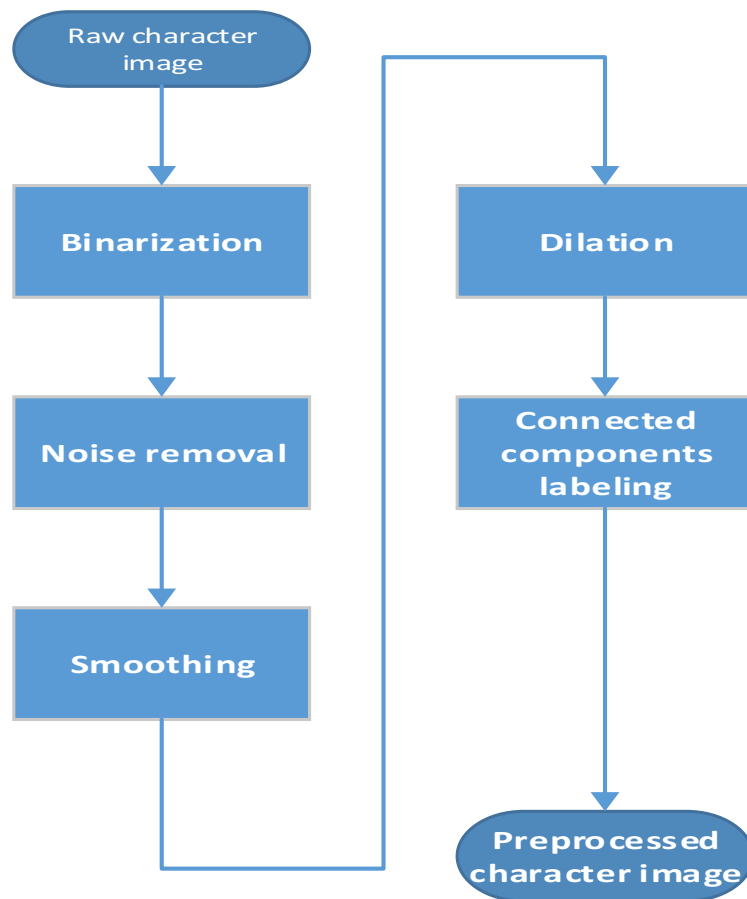


Fig. 6: Flow chart of preprocessing phase

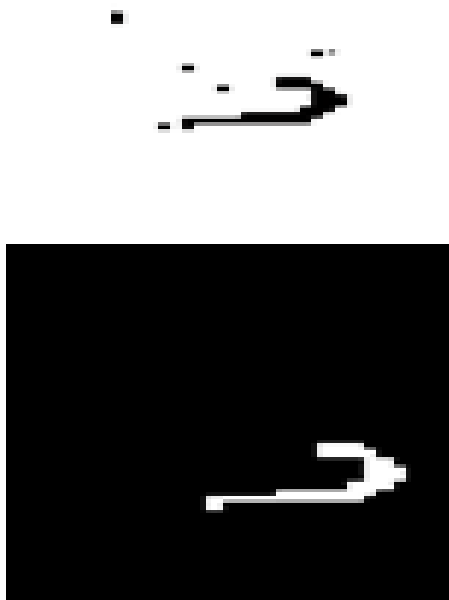


Fig. 7: A sample handwritten character with noise (above) and after applying averaging and median filters (bellow)

Both filters has been used to remove the noise in the targeted character images; averaging filter and median value filter. The noise illustrated in Fig. 6 above. The averaging value filter make a smoothness on the image and removes the unwanted sharpness in addition to very small details that are unwanted in this processing case. While the median filter is better in salt and paper noise removing. The median filter selects the median value of neighborhood pixels within selected size that is 5×5 pixels matrix. Figure 7 illustrates the character image after noise removal in the described technique.

The disconnected contours may be revealed as a result for applying the previous processes and as a result of the characteristics of the handwritten Arabic character image. Similarly, the disconnected contours are capable of affecting the successive processes which in turn clarify the significance of removing the entire existed disconnected contours. This process however called smoothing.

The gotten image so far does not generate soft character sketch. Image dilation technique applied to produce the input image. This morphological technique uses a structural element in order to expand the

contained shapes in the input image. This process assists in concluding the fitting and missing shapes in the image based on the philosophy of getting more information from an image depending on the manner of probing it. The characteristics those are related to this technique are the shape and size. Figure 8 shows the character image obtained in the last stage and the effect of dilation on it.

Labelling algorithm for the connected component operates by scanning the segments of binary image and groups "ink" pixels into components relying on 4-neighborhood connectivity pixels. Once all groups are formed, each connected pixels are labelled according to the component it is assigned to. Only the components that containing a sufficient number of pixels within the area of the main body word are retained for further processing. As a result, small "ink" regions will be omitted. This process connects the character segments together if those are not connected in the last dilation process. Figure 9 and 10 illustrates the connecting neighborhood directions, while Fig. 9 illustrates the connecting of related components that connect bad in writing parts of the same character.

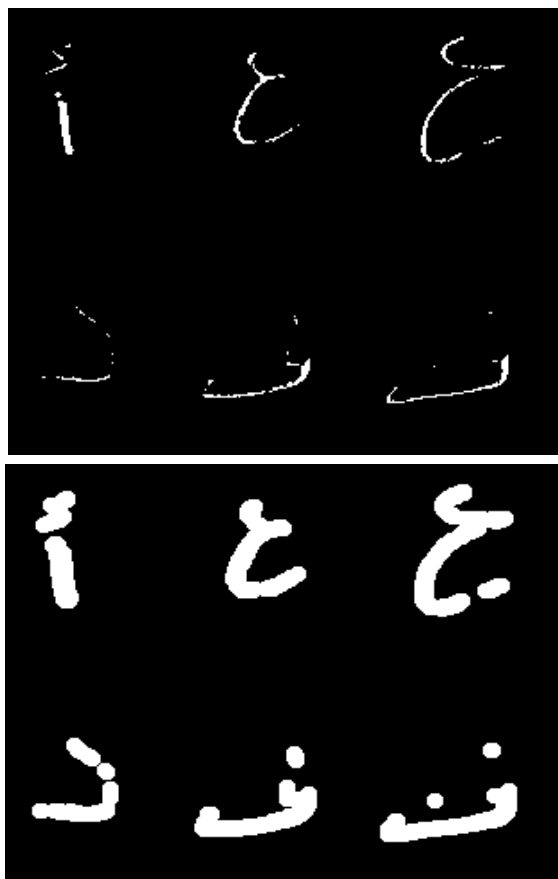


Fig. 8: Dilation process effect on the character image; characters before dilation (above), characters after dilation (bellow)

Once the segmented parts of the image have been connected, the pre-processing phase is finished and the image is ready for the next stage; contour analysis. The term contour extraction can be defined as the extraction of a single closed contour for each character shape. First of all, canny edge detection technique was applied on the image resulting the image shown in Fig. 11. The contour itself is a group of vectors those are describing the Skelton of the edge or outline shape of the character inside the image.

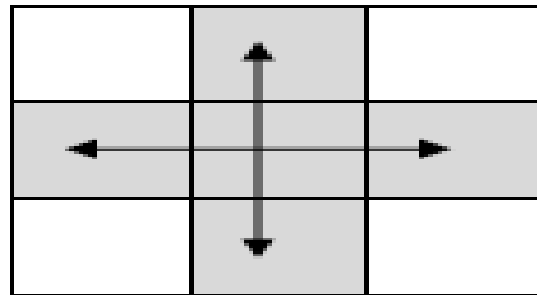


Fig. 9: Connected neighborhood directions

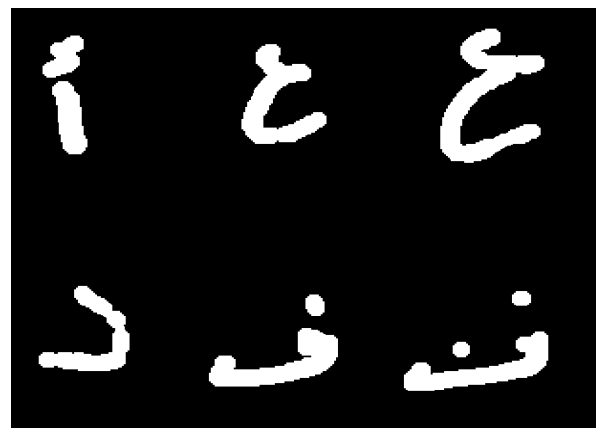


Fig. 10: Connecting segmented parts of the same character

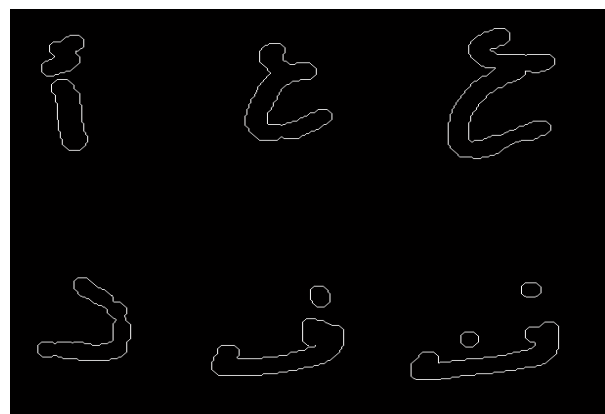


Fig. 11: Edge map of the character image

It is important in this stage to un-miss any edge occurred in the image, moreover, localize the edge point and minimize the distances between the detected edge pixels and the actual edge. Finally, only one response should be found for each edge. Considering the mentioned conditions, canny edge initially smooth images in order to eliminate any noise. Then, highlight regions with high spatial derivatives by finding the image gradient. These regions will be tracked then by the algorithm, then any fund pixel are not at the maximum it will be automatically suppressed. Now, hysteresis reduce ay gradient array, which is used to track the remaining not suppressed pixels.

The critical dilemma in contour matching is the choice of features used to measure variations between the contour segments. For that end, shape analyses techniques whose representations are based on multi-scale were become the most powerful adoption for recognition. Although multi-scale representations are quite valuable in analyzing contours, they are extremely outmoded as well; this can be defined as the high cost, which rewarded due to the explicit making of significant information that ordered in structural manner. Consequently, it is easy to conclude that as much as the technique used for matching and quantification of similarity between shapes is simpler and more full-bodied it could be optimal for obtaining those results. This is possible by the use of a solid shape representation extraction where definite properties of the shape are computed and recorded throughout generating contour.

This adoption for such technique is proposed in this research by initiating the optimization of such solid shape description method termed Multi-scale Convexity Concavity-Discrete Cosine Transform MCC-DCT approach; this approach is responsible for representing each contour point, information about the amount of convexity and concavity at different scale levels within the use of two-dimensional matrix. The contour similarities are the criteria of matching in this study. The MCC can be defined as a description method of the character shape which used to extracts some information about each contour convexity and contour concavity shape at various different scales. In other words, it's mathematical contour extraction methodology. Equation 8 illustrates the how to calculate the contour vector between two points:

$$d(u_A, u_B) = \sum_{i=0}^{I-1} P_i \cdot \frac{|f_i^{uA} - f_i^{uB}|}{x_i^A + x_i^B} \quad (8)$$

where, i is the index of contour point, d is the vector between two contour points u_A and u_B P_i is a weight value of DCT distance value, x_i is the contour's coefficient dynamic range. f_i is the DCT coefficients. The output of MCC analysis of character image is a vector containing the contour point. Each element at this vector covers convexity and concavity information at both definite contour points.

MLP Performance Measurements

The resulted contour vector will be the output of this stage and that is the input to the neural network. The high efficiency neural network produces a high classification results. For that, the structure of the MLP is very important and varied to come over this matter, the MLP with three layers viz., input, hidden and output are chosen as mentioned earlier neural network with three layers are enough to solve most of the classifications problems. The number of the neurons in the hidden layer was increased by one neuron in every training. In main while, in every increasing neuron number the network trained ten times. The number of epoch start from 1 to 10,000 to find the best generalizations. The process repeated until the 100 neurons. Finding the classification accuracy of MLP is a crucial factors, there for the Average Mean Square error method was used in this study to do that As in Equation 9 and the for the 10 times training average accuracy is calculated for every single letter. Based on the Equation 10:

$$MSE = \frac{1}{N} \sum_{i=1}^N (Actual\ output(i) - predicted(i))^2 \quad (9)$$

Hence, N denotes number of samples in each data set:

$$Average\ MSE = \frac{1}{10} \sum_{i=1}^{10} MSE_i \quad (10)$$

Results and Discussion

As discussed before, the sample acquired in this research are 2800 sample, for 10 persons and 10 trials for each character. The training was built based on 70% of the samples. This means 7 trails of each person for each character. And the rest 30% samples was left for testing. The training was built carefully and the final training result is shown in Fig. 12. The performance function was selected to be some square error "SSE".

Once the algorithm has been implemented as illustrated in the past section, the testing and evaluation of results should take its place. The testing was done over the research dataset that is divided into two groups; training group that contains 70% of the dataset and testing group that represents the remaining 30% of the dataset. It's expected to get different recognition results or ratio between the testing and training dataset. In neural network that is a rule of thumb.

The accuracy shows how the design and processing is correct in terms of recognition output. The algorithm was developed using MATLAB. And it's well known that, MATLAB is not reasonable in time estimation and consumption. So that, the results will not show the running time for each process.

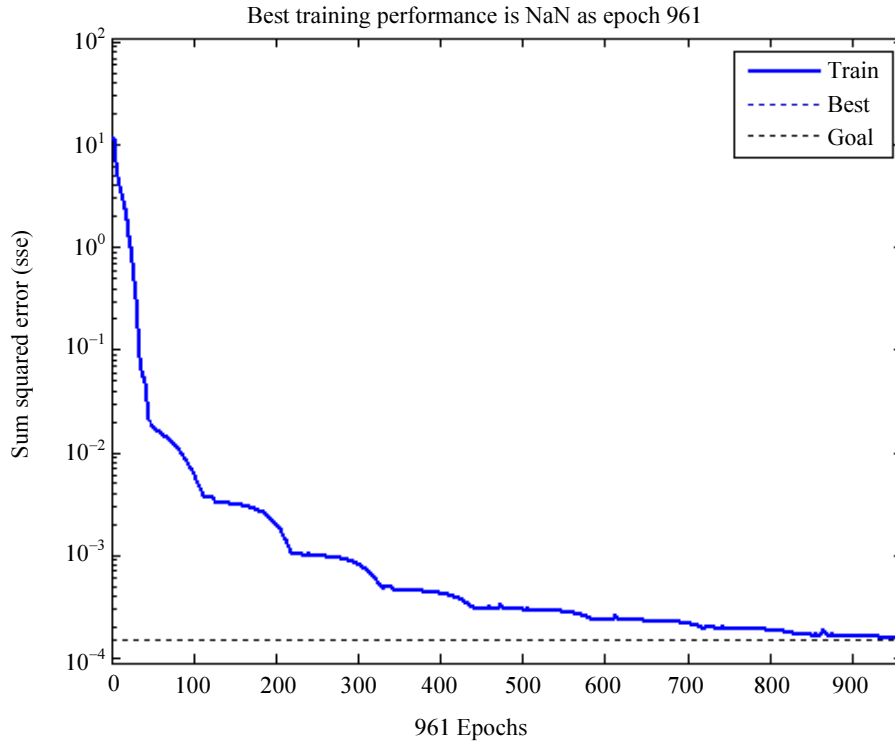


Fig. 12: Neural network training performance evaluation

Table 3: Recognition accuracy for all characters individually

Character	Recognition accuracy training set (%)	Recognition accuracy over testing set (%)
ا	100	100
ب	99	97
ت	97	93
ث	99	93
ج	100	100
ح	100	100
خ	99	97
د	100	100
ذ	99	97
ر	100	97
ز	97	97
س	100	100
ش	100	100
ص	100	97
ض	100	97
ط	100	100
ظ	100	100
ع	100	100
غ	97	97
ف	97	93
ق	99	97
ك	100	100
ل	100	100
م	100	100
ن	97	97
هـ	100	100
و	99	97
ي	100	100

Table 4: Average recognition results for all characters

	Training set	Testing set	Complete set
Average accuracy	99.25%	98%	98.8%

Table 3 shows the recorded results related to each character individually, separating the accuracy of training set and testing set. As a fact neural network achieved higher results in the training section than that in testing section. The highest classification accuracy achieved by MLP 100.0% while 97.0% achieved for the rest of the letters.

It can be seen that letters like (ع , ط, ظ, ش, س, ح, ج, ا) has chive 100% classification accuracy while the lowest classification accuracy was 93.0% achieved by (ق, ت, ث), the MLP achieved 97.0% classification accuracy for the rest of the letters.

Table 4 shows, the final performance results including accuracy for testing and training data set. The complete dataset accuracy is 98.8%. These results are consider sufficient result in comparison with the complexity of Arabic handwritten characters. The designed algorithm ensured that its power and rigid design handling such type of data patterns.

Conclusion

Automatic handwritten recognition has been moved many steps in the last few years and goes from researches into daily applications. But most concentration in handwritten is related English language.

English and similar European languages are simple in its writing structure in comparison with Arabic language. Arabic writing language has many difficulties and challenges due to its complex structure with respect to automatic handwritten recognition.

This research adopted different principles together in order to achieve high recognition accuracy result. Different pre-processing techniques were adopted then contour extraction was applied to extract contour vector of the Arabic handwritten character. Then, the resulted vector entered to well designed and evaluated neural network to recognize the character pattern. The work in this study takes all the 28 Arabic character in individual case and achieved average recognition accuracy near to 99%. The future work can focus on connected Arabic letters, Arabic word and many issued face the recognition of the Arabic letters.

Acknowledgment

The authors would like to thank Ajloun National University for the facilities used in this work.

Author's Contributions

Dr. Bajeszeyadaljunaecidia: Lead research project, coordinate developer, doing experiments, adapt analysis and writing the manuscript.

Dr. Mutasem Shabib Alkhasawneh: Advise research project, adapting analysis for the neural network part and writing manuscript and proof reading

Dr. Mohammad Ali BaniYounes: English proof reading and adapting analysis and result verifications

Ethics

This article is original and contains unpublished material. The corresponding author confirms that all of the other authors have read and approved the manuscript and there are no ethical issues involved

References

- Abdullah, M.A., L.M. Al-Harigy and H.H. Al-Fraidi, 2012. Off-line Arabic handwriting character recognition using word segmentation.
- Alkhasawneh, M.S. and L.T. Tay, 2017. A hybrid intelligent system integrating the cascade forward neural network with Elman neural network. *Arabian J. Sci. Eng.* DOI: 10.1007/s13369-017-2833-3
- Alkhasawneh, M.S., U.K. Ngah, L.T. Tay and N.A.M. Isa, 2014. Determination of importance for comprehensive topographic factors on landslide hazard mapping using artificial neural network. *Environ. Earth Sci.*, 72: 787-799. DOI: 10.1007/s12665-013-3003-x

- Alkhasawneh, M.S., U.K. Ngah, L.T. Tay, M. Isa and N. Ashidi *et al.*, 2013. Determination of important topographic factors for landslide mapping analysis using MLP network. *Scientific World J.*, 2013: 415023-415023. DOI: 10.1155/2013/415023
- Al-nuzaili, Q., D. Mohamad, N.A. Ismail and M.S. Khalil, 2012. Feature extraction in holistic approach for Arabic handwriting recognition system: A preliminary study. *Proceedings of the IEEE 8th International Colloquium on Signal Processing and its Applications*, Mar. 23-25, IEEE Xplore Press, Melaka, Malaysia, pp: 335-340. DOI: 10.1109/CSPA.2012.6194745
- Asi, A., J. El-Sana and V. Märgner, 2012. Hierarchical scheme for Arabic text recognition. *Proceedings of the 11th International Conference on Information Science, Signal Processing and their Applications*, Jul. 2-5, IEEE Xplore Press, Montreal, QC, Canada, pp. 1266-1271. DOI: 10.1109/ISSPA.2012.6310486
- Bharath, A. and S. Madhvanath, 2012. HMM-based lexicon-driven and lexicon-free word recognition for online handwritten Indic scripts. *IEEE Trans. Patt. Anal. Machine Intell.*, 34: 670-682. DOI: 10.1109/TPAMI.2011.234
- Boukerma, H. and N. Farah, 2012. Preprocessing algorithms for Arabic handwriting recognition systems. *Proceedings of the International Conference on Advanced Computer Science Applications and Technologies*, Nov. 26-28, IEEE Xplore Press, Kuala Lumpur, Malaysia, pp: 318-323. DOI: 10.1109/ACSAT.2012.59
- El Qacimy, B., A. Hammouch and M.A. Kerroum, 2015. A review of feature extraction techniques for handwritten Arabic text recognition. *Proceedings of the International Conference on Electrical and Information Technologies*, Mar. 25-27, IEEE Xplore Press, Marrakech, Morocco, pp: 241-245. DOI: 10.1109/EITech.2015.7162979
- Gargouri, M., S. Kanoun and J.M. Ogier, 2013. Text-independent writer identification on online Arabic handwriting. *Proceedings of the 12th International Conference on Document Analysis and Recognition*, Aug. 25-28, IEEE Xplore Press, Washington, DC, USA, pp: 428-432. DOI: 10.1109/ICDAR.2013.93
- Hamdani, M., P. Doetsch and H. Ney, 2014. Improvement of context dependent modeling for Arabic handwriting recognition. *Proceedings of the 14th International Conference on Frontiers in Handwriting Recognition*, Sept. 1-4, IEEE Xplore Press, Heraklion, Greece, pp: 494-499. DOI: 10.1109/ICFHR.2014.89
- Jayeck, K., M.A. Mahjoub and N. Essoukri Ben Amara, 2016. Arabic handwritten word recognition based on dynamic Bayesian network. *Int. Arab J. Inform. Technol.*, 13: 276-283.

- Kacem, A., N. Aouiti and A. Belaïd, 2012. Structural features extraction for handwritten Arabic personal names recognition. Proceedings of the International Conference on Frontiers in Handwriting Recognition, Sept. 18-20, IEEE Xplore Press, Bari, Italy, pp: 268-273. DOI: 10.1109/ICFHR.2012.276
- Krayem, A., N. Sherkat, L. Evett and T. Osman, 2013. Holistic Arabic whole word recognition using HMM and block-based DCT. Proceedings of the 12th International Conference on Document Analysis and Recognition, Aug. 25-28, IEEE Xplore Press, Washington, DC, USA, pp: 1120-1124. DOI: 10.1109/ICDAR.2013.227
- Parrweej, F., 2012. An empirical evaluation of off-line Arabic handwriting and printed characters recognition system. *Int. J. Comput. Sci.*, 9: 29-35.
- Parvez, M.T. and S.A. Mahmoud, 2013. Arabic handwriting recognition using structural and syntactic pattern attributes. *Patt. Recog.*, 46: 141-154. DOI: 10.1016/j.patcog.2012.07.012