

# Arabic Hand Written Character Recognition Based on Contour Matching and Neural Network

Marwan Abo Zanona<sup>1</sup>, Anmar Abuhamdah<sup>2</sup> & Bassam Mohammed El-Zaghmouri<sup>1</sup>

<sup>1</sup> Department of Computer Sciences, College of Shari'a and Islamic Studies in Al Ahsaa, Al-Imam Muhammad Ibn Saud Islamic University (IMSIU), Al Ahsaa, Saudi Arabia

<sup>2</sup> Department of Management Information Systems, College of Business Administration, Taibah University, Almadinah almunawarah, Saudi Arabia

<sup>3</sup> Department of Computer Science, College of Computer Science and Information Technology, Jerash University, Jordan

Correspondence: Marwan Abo Zanona, College of Shari'a and Islamic Studies in Al Ahsaa, Al-Imam Muhammad Ibn Saud Islamic University (IMSIU), Al Ahsaa, Saudi Arabia. Tel: 966-531-321-145. E-mail: dr.marwanzanona@yahoo.com

Received: February 3, 2019

Accepted: February 27, 2019

Online Published: April 30, 2019

doi:10.5539/cis.v12n2p126

URL: <https://doi.org/10.5539/cis.v12n2p126>

## Abstract

Complexity of Arabic writing language makes its handwritten recognition very complex in terms of computer algorithms. 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 implements a set of pre-processing functions over a handwritten Arabic characters, with contour analysis, to enter the contour vector to neural network to recognize it. The selection of this set of pre-processing algorithms was completed after hundreds of tests and validation. The feed forward neural network architecture was trained using many patterns regardless of the Arabic font style building a rigid recognition model. Because of the shortcomings in Arabic written databases or datasets, the testing was done by non-standard data set. The presented algorithm structure got recognition ratio about 97%.

**Keywords:** Arabic language, handwritten recognition, contour analysis, recognition, neural networks

## 1. 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, Al-Harigy & Al-Fraidi, 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 & Farah, 2012; Bharath & Madhvanath, 2012).

Arabic language has many features that make the automatic recognition by computer system a hard task to perform in high accuracy (Boukerma & 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.

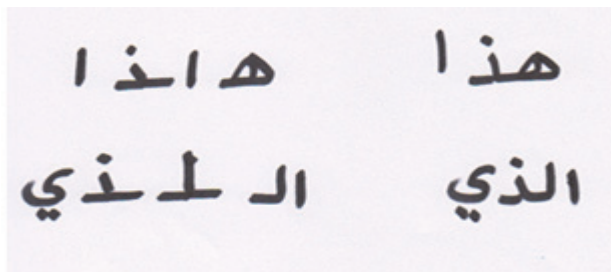


Figure 1. Arabic words with hidden characters. The words as it's normally written (right). The containing characters within those words (left)

More than seven hundred million people are writing in Arabic characters, where different languages are written in Arabic characters including Arabic language itself, Persian, Urdu, and other (Al-nuzaili, Mohamad, Ismail & Khalil, 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 important 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, Aouiti & Belaïd, 2012; Asi, El-Sana & Märgner, 2012).



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

The motivations toward working with Arabic handwritten character recognition to improve the recognition rate in high percentage that can be used commercially with very low falls recognition rate leads to start and present this paper. The contribution of this paper is to achieve a 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 use this approach allows to achieve high recognition ratio due to the ability of the proposed algorithm in treating the general shape of the character. The methodology can handle the variety of the writers' details and hand sketches (Bharath & Madhvanath, 2012).

The proposed algorithm in this paper is based on hundred days of testing, validation and evaluation to specify the best approach that can achieve more than 95% of recognition accuracy with less than 1% of falls acceptance ratio. This algorithm is based on selecting meaningful parameters that contains the Skelton architecture of the character with minimal data size, then training the optimal neural network (feedforward) structure to build a recognition pattern.

Herewith the below sections; section 2 shows some related papers and literature review. Section 3 demonstrates the used dataset. Section 4 describes the working methodology. The results were discussed in section 5 and the discussion for conclusion in section 5.

## 2. 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.

El-Sana and Biadisy (El-sana & Biadisy, 2013) were presented a method for online character recognition of Arabic text, where the text was represented in strokes form. The structure of the geometrical data then represented as a model of the strokes. The features vectors then constructed from the delay and orientation of the strokes. The constructed vector represents the character or word pattern. The observation sequence used to predict the word depending on higher probability word.

The authors was developed this algorithm (El-sana & Biadisy, 2013) from their previous work in (El-sana & Biadisy, 2010). The work in (El-sana & Biadisy, 2010) was concerned on online detection and tracing of the strokes in order to sample the two dimensional strokes data. The delayed strokes was used as a key feature to predict the behaviour and thus, the geometry of the word. The extracting of strokes and generation of observation sequence was introduced in that the first patent. The algorithm of two dimensional vector representation was modified and improved in the second patent. A simple probability detection was introduced in the first patent and the modification, improvement, and validation was presented in the second patent.

Krayem et al in (Krayem, Sherkat, Evett & Osman, 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, in (Gargouri, Kanoun & Ogier, 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.

TanvirParvez et al in their work (Parvez & 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, in (Hamdani, Doetsch & Ney, 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 in (El Qacimy, Hammouch & Kerroum, 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.

Musleh D., Halawani K., and Mohammed S., was presented in the patent referenced in (Musleh, Halawani & Mohammed, 2016) Arabic handwritten numbers and characters recognition algorithm. This algorithm was based on fuzzy logic system design and modelling. This algorithm pre-processes a template of Arabic characters and numbers, extracts the features, and stores the features. Then, during inline work, the extracted features will be compared with the stored features using fuzzy logic model. This methodology can recognize single characters.

Jayeck et al in (Jayeck, Mahjoub, & Essoukri Ben Amara, 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.

### 3. 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.

Why standard dataset is not used? While the standard dataset provides homogenous environment for testing of different approaches and helps to make the attained results to be directly compared with compared results with other scientific researches. The answer is that, the lack of researches in Arabic language and thus, Arabic handwritten causes a shortage of resources, there are some datasets available online, but we can consider them as reputable data sets. Most of them has many errors, mistakes, heterogeneity, and cannot enable us to attain the benefits of using standard datasets.

Arabic characters are dot-based. This means, some characters could be written in similar way but one or more dots should be used to distinguish between them. Figure. 3 shows 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 figure.



Figure 3. Sample Arabic handwritten characters and it’s pronunciation in English. Arabic character (right), pronunciation in English characters (left)

Table 1. Arabic characters list

أ	ز	ق
ب	س	ك
ت	ش	ل
ث	ص	م
ج	ض	ن
ح	ط	ه

خ	ظ	و
د	ع	ي
ذ	غ	
ر	ف	

Arabic has 28 characters, and they are all listed in table 1. All those 28 characters 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.

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 samples are needed to build up a precise neural network. Also, the character could handle few changes in some details if it was written 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. So, 10 persons helped in creating the dataset, where each person wrote the same character 10 times. Thus, the total dataset becomes 100 samples of each character, each 10 of them has been written by the same person. The total dataset is 28 characters times 10 persons times 10 samples of each person, then it is 2800 samples, similar to those shown in Figure 3.

#### 4. Method

The presented algorithm runs two parallel paths and two sequential phases. The parallel paths are the training and running paths. The three phases are running sequentially where the output of the previous phase is the input of the next phase. Those phases are:

1. Preprocessing
2. Contour analysis
3. Neural network training or simulation.

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 go into successive trials 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.

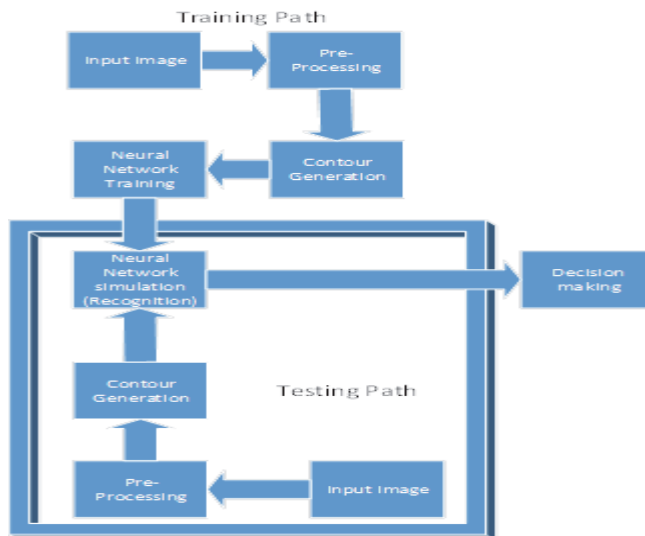


Figure 4. Presented algorithms block diagram, showing both; training path and testing path

Figure 4, shows the basic block diagram of the presented algorithm. 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. Sample Arabic handwritten character to be processed

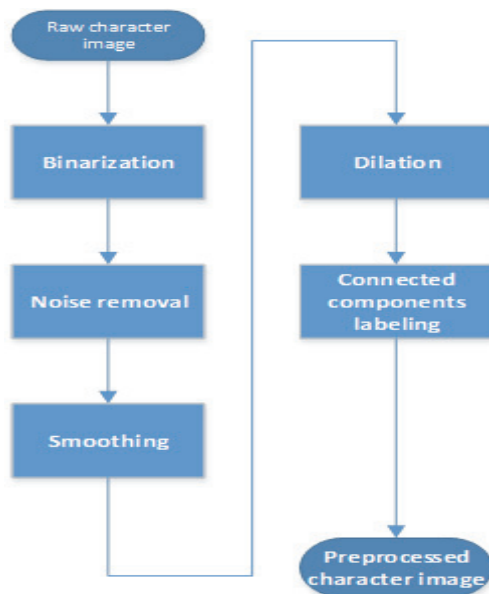


Figure 6. Flow chart of preprocessing phase

Figure 5, 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 affect the banalization process). The

binarization in this paper 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.

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 result 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 type of noise. Any noise type can be handled within the use of certain algorithms.



Figure 7. A sample handwritten character with noise (above) and after applying averaging and median filters (below)

Both filters has been used to remove the noise in the targeted character images; averaging filter and median value filter. The noise illustrated in Figure 7 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 5x5 pixels matrix. Figure 8 (below) illustrates the character image after noise removal in the described technique.



Figure 8. Dilation process effect on the character image; characters before dilation (above), characters after dilation (below)

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 is called smoothing.

The obtained image so far does not generate soft character sketch. Image dilation technique is 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 that are related to this technique are the shape and size. Figure 7 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, illustrates the connecting neighborhood directions, while Figure 10, and illustrates the connecting of related components that connect bad in writing parts of the same character.

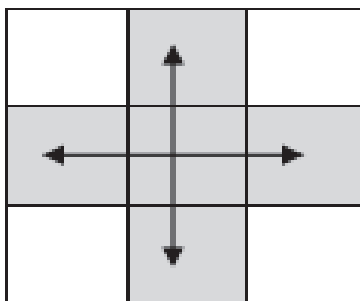


Figure 9. Connected neighborhood directions



Figure 10. Connecting segmented parts of the same character

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 Figure 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.

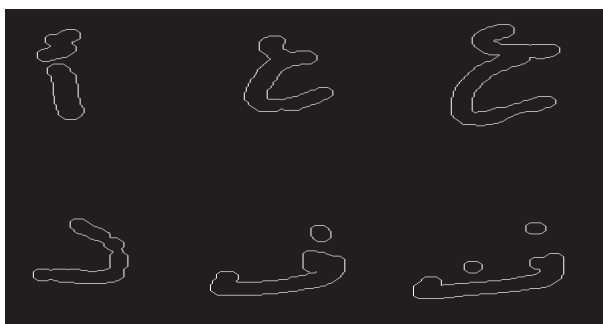


Figure 11. Edge map of the character image

It is important in this stage to include any edge occurred in the image, 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 by the algorithm, then any fund pixel are not at the maximum will be automatically suppressed. Now, hysteresis reduce ay gradient array, which is used to track the remaining un 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 paper. 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 1 illustrates the how to calculate the contour vector between two points.

$$d(u_A, u_B) = \sum_{i=0}^{l-1} p_i \cdot \frac{|f_i^{u_A} - f_i^{u_B}|}{x_i^A + x_i^B} \tag{1}$$

Where,

i is the index of contour point.

d is the vector between two contour points uA, and uB

Pi is a weight value of DCT distance value

xi is the contour's coefficient dynamic range.

fi 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 point.

The resulted contour vector will be the output of this stage and that is the input to the neural network. A feed forward neural network has been designed as illustrated in Figure 12. The network architecture is a three neuron layers; one input, one output, and one hidden layer. Number of neurons in the input layer is 20, with tangent sigmoid activation function, number of neurons in the hidden layer 50 with logarithmic sigmoid activation function, and the number of neurons in the output layer is 28 with linear activation function. The output layer contains one neuron for each character. Where the output will be a vector of 28 element that indicates the probability for each character to be recognized. This means that, if the output “x” is near to 1, so, the input is expected to be character “x”, otherwise it's not “x”, and so for all 28 output neurons (character indices) of the output layer.

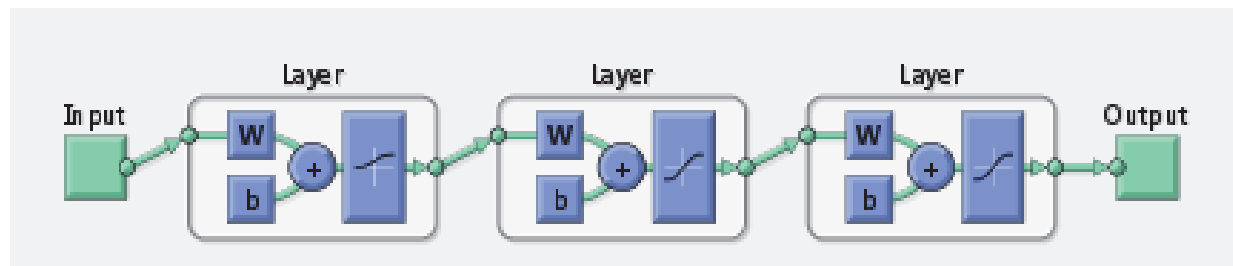


Figure 12. Designed neural network architecture

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 Figure 13. The performance function was selected to be some square error “SSE”.

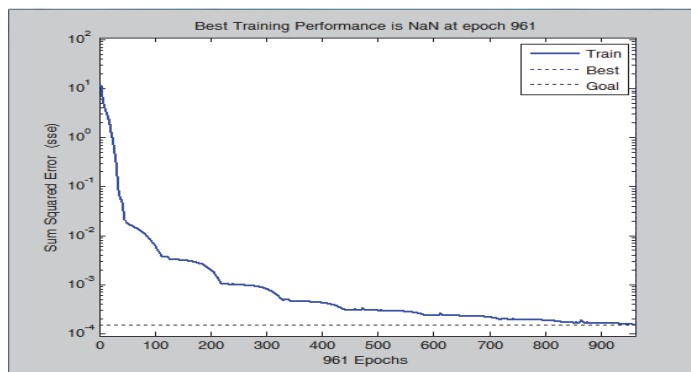


Figure 13. Neural network training performance evaluation

**5. Results**

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 performance evaluation was done in different criteria; recognition accuracy, and precision. The precision shows how the design is stable and can generates reasonable outputs. 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.

Table 2 show the recorded results related to each character individually, separating the accuracy of training set and testing set. The good designed algorithm that makes the gap between the training set and training set as small as possible, and this is what achieved in this paper as illustrated in table 2.

Table 2. Recognition accuracy for all characters individually

<i>Character</i>	<i>Recognition accuracy over training set (%)</i>	<i>Recognition accuracy over testing set (%)</i>
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 3 show the final performance results including accuracy and precision for training set, testing set, and all dataset together. The complete dataset accuracy is 98.8% and complete dataset precision is 99.4%. This results are excellent 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.

Table 3. Average recognition results for all characters

	<i>Training set</i>	<i>Testing set</i>	<i>Complete set</i>
<b><i>Average accuracy</i></b>	99.25%	98%	98.8%
<b><i>Average precision</i></b>	99.5%	99%	99.4%

## 6. Discussion

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 was 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 paper takes all the 28 Arabic character in individual case and achieved average recognition accuracy near to 99%, and this value is very excellent in contrast with the complexity of Arabic writing.

The result was gotten in this paper opens the path to apply the described algorithm over more concatenation of Arabic handwritten. Yes, this paper was limited to individual characters, so, the future work will be on complete holistic words and should be extended to handle full documents. The handling of handwritten documents has high importance in Arabic language. The future improvement of the presented algorithm should include those documents.

## References

- Abdullah, M. A., Al-Harigy, L. M., & Al-Fraidi, H. H. (2012). Off-Line Arabic Handwriting Character Recognition Using Word Segmentation. *arXiv preprint arXiv:1206.1518*. 2012.
- Al-nuzaili, Q., Mohamad, D., Ismail, N. A., & Khalil, M. S. (2012). Feature extraction in holistic approach for Arabic handwriting recognition system: A preliminary study. In *Signal Processing and its Applications (CSPA), 2012 IEEE 8th International Colloquium on* (pp. 335-340). IEEE. March, 2012. <https://doi.org/10.1109/CSPA.2012.6194745>
- Asi, A., El-Sana, J., & Märgner, V. (2012). Hierarchical scheme for Arabic text recognition. In *Information Science*,

- Signal Processing and their Applications (ISSPA)*, 2012 11th International Conference on (pp. 1266-1271). IEEE. July, 2012.
- Bharath, A., & Madhvanath, S. (2012). HMM-based lexicon-driven and lexicon-free word recognition for online handwritten Indic scripts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4), 670-682. 2012. <https://doi.org/10.1109/TPAMI.2011.234>
- Boukerma, H., & Farah, N. (2012). Preprocessing algorithms for Arabic handwriting recognition systems. In *Advanced Computer Science Applications and Technologies (ACSAT)*, 2012 International Conference on (pp. 318-323). IEEE. November, 2012. <https://doi.org/10.1109/ACSAT.2012.59>
- El Qacimy, B., Hammouch, A., & Kerroum, M. A. (2015). A review of feature extraction techniques for handwritten Arabic text recognition. In *Electrical and Information Technologies (ICEIT)*, 2015 International Conference on (pp. 241-245). IEEE. March, 2015. <https://doi.org/10.1109/EITech.2015.7162979>
- El-sana J., & Biadisy F. (2010). Online Arabic handwriting recognition, *USPTO 12/374773*, May 2010
- El-sana J., & Biadisy F. (2013). Online Arabic handwriting recognition, *USPTO 12/374773*, Dec 2013
- Gargouri, M., Kanoun, S., & Ogier, J. M. (2013). Text-Independent Writer Identification on Online Arabic Handwriting. In *Document Analysis and Recognition (ICDAR)*, 2013 12th International Conference on (pp. 428-432). IEEE. August, 2013. <https://doi.org/10.1109/ICDAR.2013.93>
- Hamdani, M., Doetsch, P., & Ney, H. (2014). Improvement of Context Dependent Modeling for Arabic Handwriting Recognition. In *Frontiers in Handwriting Recognition (ICFHR)*, 2014 14th International Conference on (pp. 494-499). IEEE. September, 2014.
- Jayech, K., Mahjoub, M. A., & Essoukri Ben Amara, N. (2016). Arabic handwritten word recognition based on dynamic Bayesian network. *Int. Arab J. Inform. Technol.(IAJIT)*, 13(3).2016.
- Kacem, A., Aouiti, N., & Belaïd, A. (2012). Structural features extraction for handwritten Arabic personal names recognition. In *Frontiers in Handwriting Recognition (ICFHR)*, 2012 International Conference on (pp. 268-273). IEEE. September, 2012.
- Krayem, A., Sherkat, N., Evett, L., & Osman, T. (2013). Holistic Arabic whole word recognition using HMM and block-based DCT. In *Document Analysis and Recognition (ICDAR)*, 2013 12th International Conference on (pp. 1120-1124). IEEE. August, 2013. <https://doi.org/10.1109/ICDAR.2013.227>
- Musleh D., Halawani K., & Mohammed S. (2016). Arabic like online alphanumeric character recognition system and method using automatic fuzzy modeling, *USPTO 14/808586*, Sep 2016.
- Parrweej, F. (2012). An empirical evaluation of off-line Arabic handwriting and printed characters recognition system. *Int J Comput Sci*, (9), 2012.
- Parvez, M. T., & Mahmoud, S. A. (2013) Arabic handwriting recognition using structural and syntactic pattern attributes. *Pattern Recognition*, 46(1), 141-154.

## Copyrights

Copyright for this article is retained by the author(s), with first publication rights granted to the journal.

This is an open-access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).