Holy Quran Arabic Character Recognition using Neural Networks

Dr. Majdi Salameh,
KSA, Jeddah, university of Jeddah, , computer Science and Engineering collage, CSAI department.
*Email: masalameh@uj.edu.sa

Abstract
Arabic text recognition is one of the most important fields of optical character recognition for Arabic people. Arabic text recognition consider of converting an image of Arabic text into editable format such as doc or pdf. One of its branches is holy Quran text recognition. The key aim of this research is to develop Quran Arabic Recognition System. The study recognises the Arabic text (character) individually . In addition, this study will use Back propagation neural networks for character recognitions. Because of the nature of holy quran the rate of recognition must by extremely high.

Keywords: text recognition, Arabic text recognition, quran text recognition neural networks,

DOI: 10.7176/IKM/13-3-05
Publication date: May 31st 2023

INTRODUCTION:

Arabic text recognition is a stimulating problem and plentiful of its complications come from the characteristics of Arabic text. Some of most important challenges are discussed in the next section.

Fist The connectivity challenge: this is concerned with the way that Arabic characters are connected to each other to form word. **Also The dots challenge** dotting is widely used to distinguish characters sharing similar graphemes. **The multiple grapheme cases challenge**; This is due to the obligatory connectivity in Arabic orthography. The same grapheme representing the same character may have multiple variants according to its relative position within the Arabic word segment, which gives different shape for the character in different position. **The ligatures challenge**; it is concerned with the compound characters. Two or three characters are connected together vertically to implement one block. The characters in that case take different forms, different connection form and connection point. **Diacritic definition**: A diacritical mark is a mark added to a letter to alter a word's pronunciation or to distinguish between similar words. It may be located above, below, or in-between the character. There are some special vowels for the holy Qur'an and many kinds of vowels such as single, double, triple vowels, characters, and words as vowels. **Connecting the vowel;** it is concerned with connecting the vowel with its adjacent character, which is another problem, especially vertical ligature characters another **Broad graphemes set challenge**; which is deals with how the characters are connected together to implement word or part of a word. Some characters came as a last character; some can be connected to a followed character. The characters may take different shapes to different fonts. Multiple grapheme cases as well as the occurrence of ligatures directly lead to broad grapheme sets, which means higher ambiguity and hence more confusion. **Arabic calligraphy challenge;** This is concerned with the variety of forms of the same letter. Also, it may be written in different shapes according to font style and the writer. Old Qur'an text was written in a calligraphic form and many ancient documents too. **Inconsistency in relation of dots and diacritics to body characters**; almost all Arabic typeface fonts have the dots placed arbitrarily. Their relation to the characters and to one another is not standard, especially in hand written and in calligraphy font as to the diacritics.

Holy Qur'an is one of the most suitable texts in Arabic language for recognition and segmentation because it contains many challenges such as dots, vowels, and ligatures; therefore, it considered as good case study for Arabic characters recognition. This research will treat Arabic text recognition as stand-alone case and will solve the problems indicated by using a vowel Arabic text.

OBJECTIVES OF THE RESEARCH

The main aim of the research is to develop technique depending on neural networks that can be used to recognizes Arabic text in holy Quran.

METHODOLOGY

Methodology Stages

This research proposes many stages and techniques to solve previous criticizing points and achieve research objectives as shown in figure 1.

1 Image Pre-processing; The image analysis stage; includes utilities such as rotation, filtering, and cropping.

2 Image analysis; This stage includes reading Qur'an text page image and convert color of the image into gray scale, after that the gray scale presentation of the image will be turned into binary representation which will be used to indicate the page text line areas and space areas.

3 Segmentation stage; In this stage the character line will be dividing into spaces, isolated characters, words and special symbols. After that the words will be divide horizontally using base line technique. Follow that will be dividing compound characters (overlap vertically) using vertical scale. Then dividing the character text area into five major areas which are upper vowel, upper dots, character, lower dots, and lower vowel areas. Finally Connecting vowels with corresponding character using shortest distance and center line technique.

4 Feature Extraction; Feature extraction stage will extracts the strokes of segmented vowels in order to composite them to form one object and will determine characters object as rectangle to use it in recognition stage as training set of neural networks.

5 Recognition; This research recognition stage has three different steps; first step is to recognise character using back propagation neural networks. Second step is to recognize vowel using fuzzy logic technique for pattern recognition.

Characters Recognition

In character recognition stage as shown in figure , all character objects from previous stage are stored as image in array of zeros and ones. These characters will be divided into different categories according to number and position of dots. This array is trained using back propagation neural networks, to get the weight for all characters in different font size and format. Then those weights will be used to recognize new character.

This step consists of neural networks training using back propagation (BPNN), weights storing, character classification, and character recognition.

Character recognition step recognize the characters using neural networks technique. The character is entered from previous stage in specific size as matrix of 0,1. Then it will converted as on dimension array of 0,1. After that, it is trained on a set of standard characters.

BPNN is trained on many types of characters. All weights is stored in database containing all the data about the characters. Next character classification step classify all characters into classes equal to the number of characters in Arabic. After that step, character recognition recognize any new character, using the weights already stored in the database from the training step.

Depending on similar study analysis, and experiments of different design choices, the 2-layer neural network was chosen used in . In order to improve the recognition, first classified the input characters into five sets and for each set designed a separate neural network. The first set contains all the beginning characters, the second set contains all the middle characters, the third set contains all the ending characters, the fourth set contains all the isolated characters, while the fifth set contains all the ligature characters.

The first neural network for beginning characters shown in figure 3, is designed as two layers back propagation network; the first layer consisted of 25 neurons and the second layer consisted of 23 neurons (one neuron for each token). It used the standard back propagation training and testing function. All NNW are designed as two layers.

The second neural network for middle characters as shown in figure below is also designed as a two-layer network; the first layer is composed of 29 neurons and the second layer is composed of 28 neurons. Also, the function is the same.

In the third neural network for ending characters; the first layer is composed of 32 neurons and the second layer is composed of 28 neurons. While the fourth neural network (for isolated characters) first layer is composed of 30 neurons and the second layer is composed of 28 neurons.

The fifth neural network for ligature characters is designed as a two-layer network; and the first layer is composed of 28 neurons, while the second layer is composed of 28 neurons.
The number of neurons in the hidden layer was selected for each neural after several experiments to produce the best recognition rate. Also, the back propagation function was picked because of its output range (0 to 1), which is suitable for classification, and training purposes.

The network is trained to output a 1 in the correct position of the output vector and to fill the rest of the output vector with 0's. After that it converts the output numbers to integers which will represent corresponding number of specific character image. However, input vectors may result in the network not creating perfect 1's and 0's. So, after the network produces its results, results are passed through the competitive transfer function that replaces the largest element of the output vector by 1, and places 0 in place of the other elements.

RESULTS

CHARACTER RECOGNITION

An extensive testing was conducted to test the effectiveness of the presented ACR. First, tested every stage separately, and then performed a complete system testing. The execution time depended on the text length and existing of Qur'an special character and ligatures. After testing, found that the patterns can suffer from a small amount of variation according to the font type and special Qur'an characters.
Total of five back propagation neural networks were used. An input layer of all possible characters for each character class (beginning, ending, middle, isolated, and ligature). Hidden layers were chosen depending on experiments and testing results, and an output layer for each character class set.

The neural network was trained using a set of characters extracted from images containing Arabic text with specific font types. The model was tested with several data sets such as set contained scanned pages composed only of characters found in the training set. Set includes images of pages containing text of characters different than the training set.

Results show a graceful degradation in model accuracy with different character positions (beginning, middle, ending, and isolated). This is due to the changes in the characters and the dots features among the different character types. Another classification for Arabic characters, depending on the number of dots and their locations to the characters, were used to examine results correctness for character recognition neural networks.

Arabic Character recognition results using BPNN

Training set of characters from Othman Qur'an texts were used for each type of Arabic characters network as an input. All training characters and their sets are listed in the following table.
Table 1 shows Character training set for BPNN. Table 2 shows Performance and flow charts of all results from the five BPNN are shown after. The details of the first results for the training of beginning characters listed in table are shown in figure 5.

The neural consist of multilayers; one layer as input, another layer as output layer and two hidden layers. Training takes 18 iterations to achieve the target and it needed 8 minute 6 second to complete. The performance was 2.19 e -05, while gradient was 0.190 and Mu was .0100.

Figure 6 shows the diagrams of training in details including gradient, Mu, and validation check for the entire 18 epochs. Gradient was between 10^0 and 10^5 and stayed at that range until the 16 epoch when it was reduced to 0.018962 which is less than $10^{0.}$ Mu for the first four epochs were between 10^0 and 10^{-5} then it reached 10^0 in epoch five and keep the average of 10^0 for the remaining epochs, while the validation was zero in the entire 18 epochs.

The regression diagram as shown in figure 7 , with R=1 and output equals target plus (.0077) shows that all data fits to target line. The MSE figure 8 shows that the error rate was 10^2 at the beginning and reduced to 10^{-4} at the last epoch.

The second neural results for the middle character training using back propagation neural networks is shown in figure 9.

Middle characters training takes 4 iterations with 4 minute and 48 second as shown in figure 10. Performance for this network was 2.34 e -14, while gradient was 0.000809 and MU was 1.00 e -06 with validation equals zero.

Gradient at the first three epochs was more than 10^5, and then it was reduced at epochs three and four to less than 10^0 as shown in figure 11. The MU started form 10^{-16} and increased to 10^{-4} and stayed the same till the last epoch.

Regression shows that all data fits the target in the target line with R=1 and output= target + 5.7 e- 008. MSE figure 12 shows that at first epoch MSE was 10^5 and reduced to 10^3 at next epoch, then at epoch three it reached 10^{-4}, Finally MSE at epoch four was 10^{-14}.

The third neural networks results for the training of the ending character training using p.b. is shown in figure 13. The network needed 14 iterations for training with 8 minutes and 41 second training time. Performance was 8.77 e-05, gradient was 0.360, and Mu was 0.0100.

Gradient at the first ten epochs was more than 10^5, and then it was reduced at epochs eleven to 10^1 and kept reducing to 0.35955 at epoch fourteen as shown in figure 14. The MU started form 10^{-16} and increased to 10^{-10} at epoch two then it reached 10^0 at the remaining epochs.

Regression in figure 15 shows that all data fits the target at target line with R=1 and output= target + 0.004. MSE in figure 16 shows that at first epoch MSE was 10^2 and reduced to 10^0 at epoch eight, until it reaches 10^{-4} at epoch fourteen.

REFERENCES:

Abroumandnia, S, & Reakhah, V. (2008). Parsian/Arabic handwritten word recognition using M-band packet wavelet transform, image and vision computing, 26(6), 829-842

Ahmed, M.Z., Mohammad, S.Z., & Choong-Yeun, L. (2004). Isolation of dots for Arabic OCR using voronoi diagrams, proceeding of the international conference on electric engineering and informatics, c-01, 199-202.

Al-Sadoun, H., & Amin, A. (1995). A new structural technique for recognizing printed Arabic text. International Journal of Pattern Recognition and Artificial Intelligence, 9(1), 101-125.

Cheung, A., Bennamoun, M., & Bergmann, N.W. (1998). A recognition-based Arabic optical character recognition system. IEEE International Conference on Systems, Man, and Cybernetics. San Diego, CA, USA., 5, 4189-4194.

El-Bialy, A.M., Kandil, A.H., Hashish, M.A., & Yamany, S.M. (2000). Arabic OCR: toward a complete system. Proceedings of SPIE - Document Recognition and Retrieval VII, San Jose, CA. 3967, 42-51.

El-Dabi, S., Ramsis, R., & Kamel, A. (1990). Arabic character recognition system: A statistical approach for recognizing cursive typewritten text. Pattern Recognition, 23(5), 485-495.

Elgammal, A.M., & Ismail, M. (2001). A graph-based segmentation and feature extraction framework for Arabic text recognition. 6th International Conference on Document Analysis and Recognition (ICDAR'01), pp 622-626.

Majdi Salameh (2012).“Isolation Arabic Characters Recognition Using Analytical And Fuzzy Logic For Pattern Recognition Technique” International Arab Conference of e-Technology (IACe-T'2012),Jordan, Zarqa, 2012.

Majdi Salameh,& Suleiman Almasri, 92017)."Enhancing Fuzzy Logic for Pattern Recognition Technique to Recognize Arabic Quran Short Diacritics", International Journal of Electronics Communication and Computer Engineering Volume 8, Issue 1, ISSN (Online): 2249–071X, 2017

Al Jarrah, Al Kiswany, Al Gharaibeh, Fraiwan, & Khasawneh. (2006) . Anew algorithm for Arabic optical character recognition, 1-14.

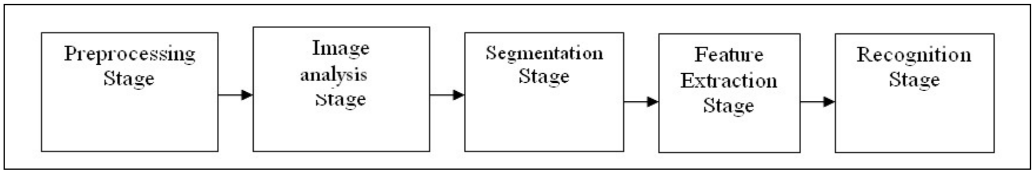


Figure 1: system stages

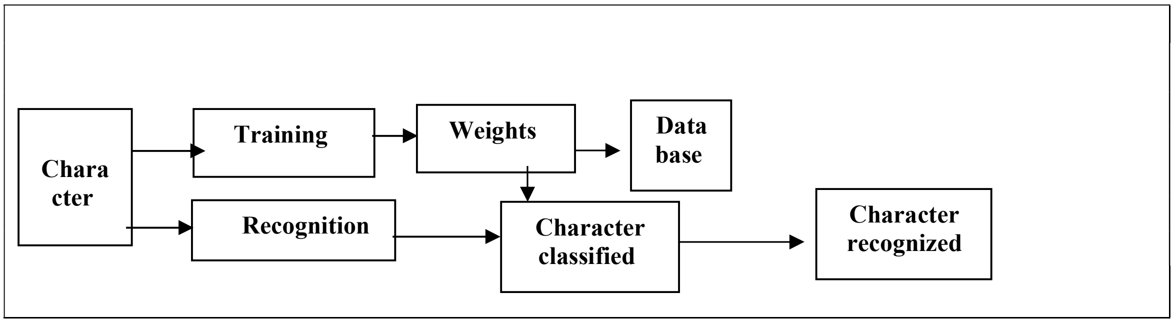


Figure 2: Character recognition architecture using neural network.

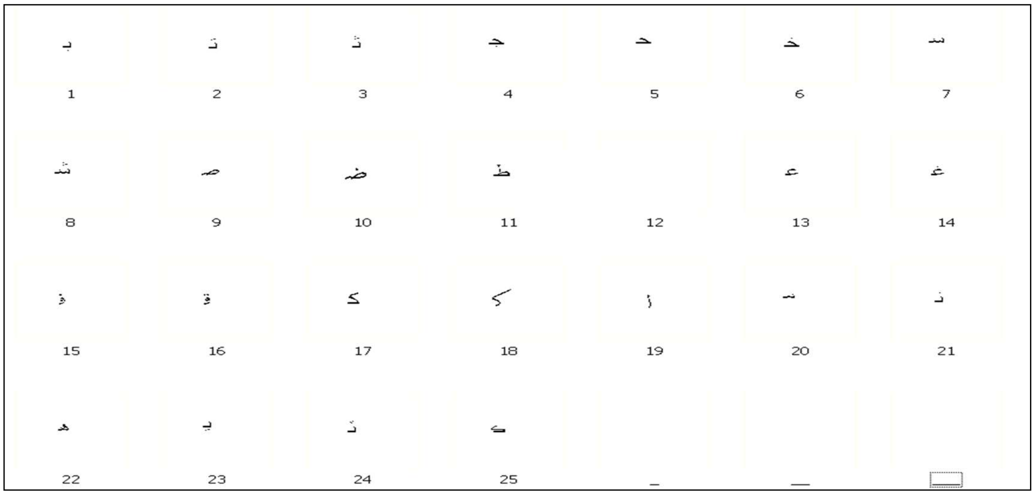


Figure 3: Starting character set.

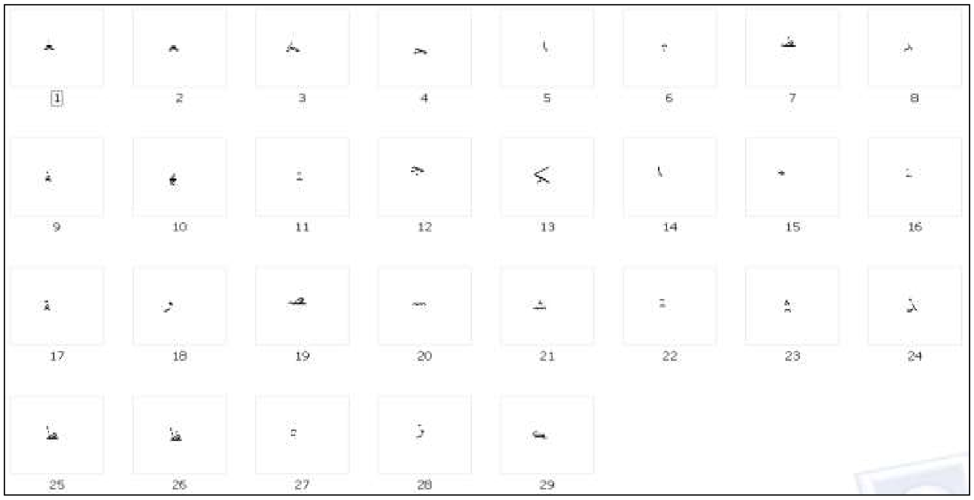


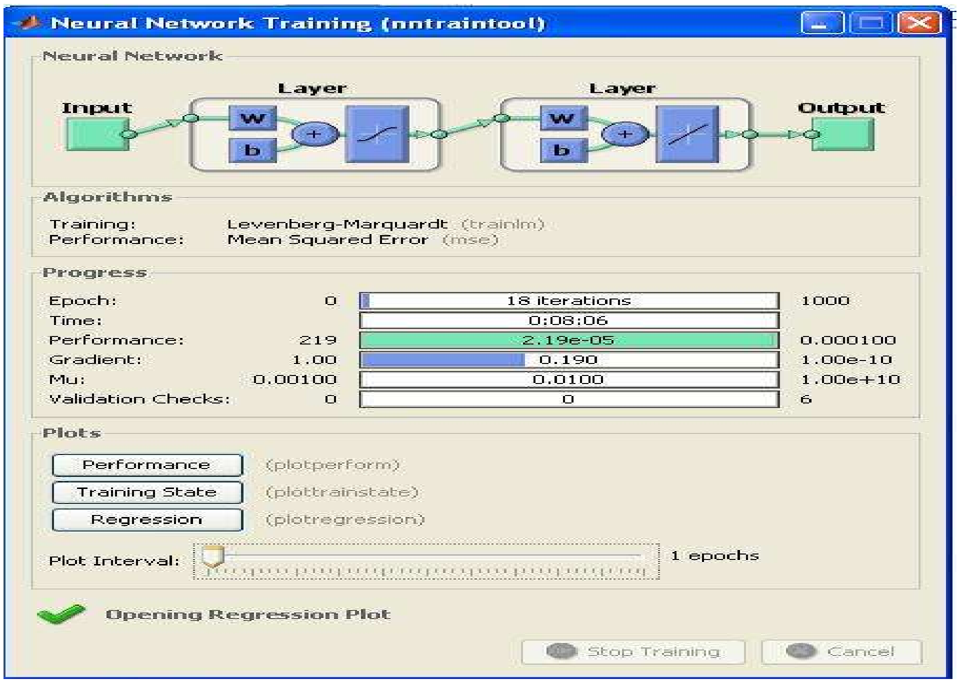
Figure 4: Middle character set

Table 1: Performance and flow charts of all results from the five BPNN are shown after.

Character set	Character type	Input characters	Output characters
Set 1	Beginning	25	23
Set 2	Middle	29	28
Set 3	Ending	32	28
Set 4	Isolated	30	28
Set 5	Ligature	28	28

Table 2: Performance of all results from the five BPNN tested.

Character set	Performance	Regression	M.S.E	Time
Beginning	2.19 e -05	0.189	0.0100	8 min 6 sec
Middle	2.34 e -14	0.000809	1.00 e -06	4 min 48 sec
Ending	8.77 e -05	0.360	0.0100	8 min 41 sec
Ligature	1.3 e-5	0.216	0.0100	12 min 10 sec
Isolated	3.34 e-10	0.0903	1.00 e -04	7 min 22 sec



Neural Network Training (nntraintool)

Neural Network

Algorithms

Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)

Progress

Epoch:	0	18 iterations	1000
Time:		0:08:06	
Performance:	219	2.19e-05	0.000100
Gradient:	1.00	0.190	1.00e-10
Mu:	0.00100	0.0100	1.00e+10
Validation Checks:	0	0	6

Plots

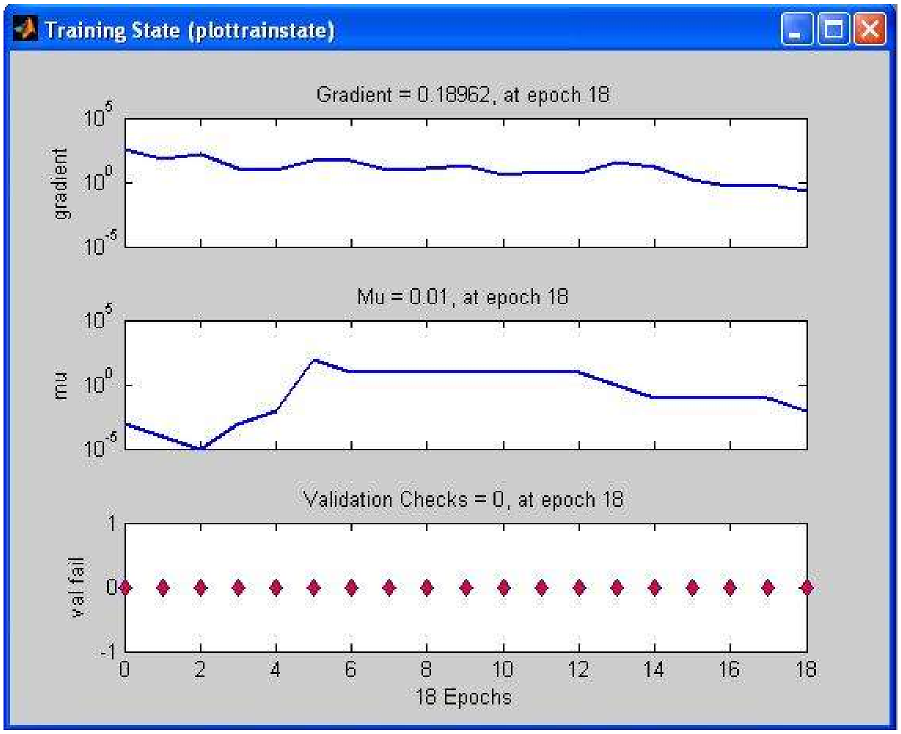
Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

✓ Opening Regression Plot

Stop Training
Cancel

Figure 5: The neural networks' tool used for training the beginning characters.



Training State (plottrainstate)

Figure 6: the training results of beginning characters.

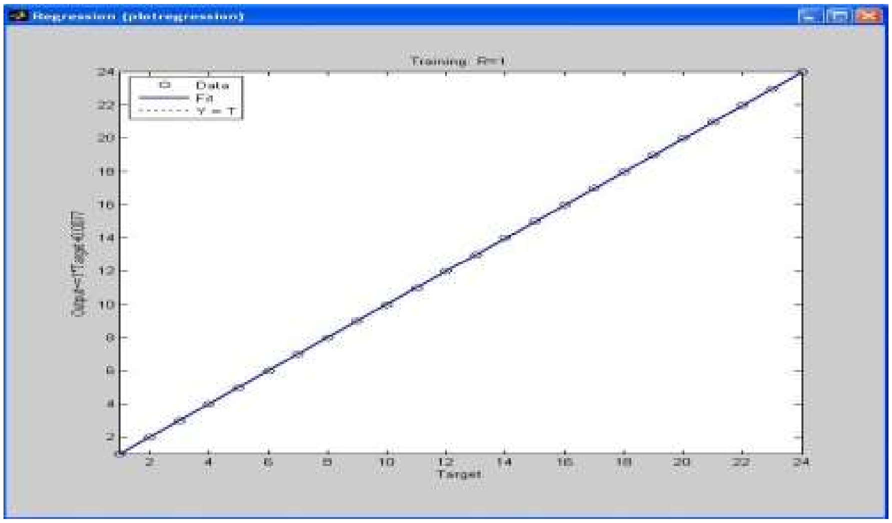


Figure 7: The regression results of training for the beginning characters.

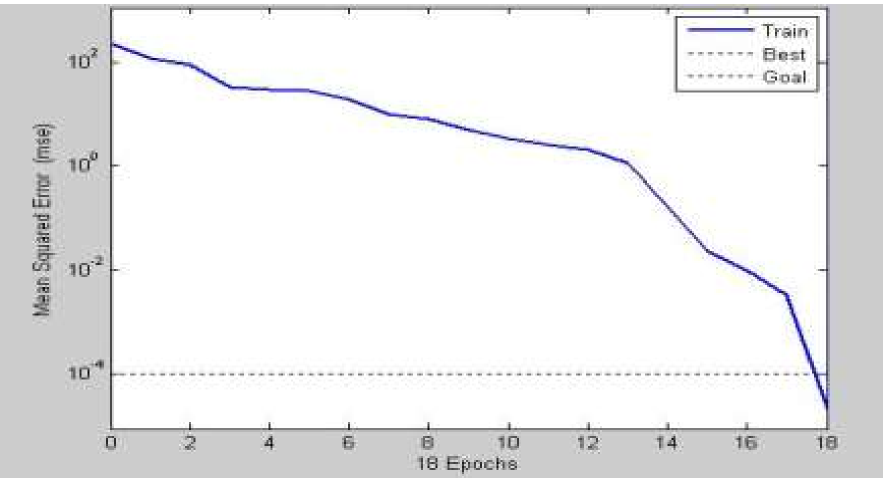
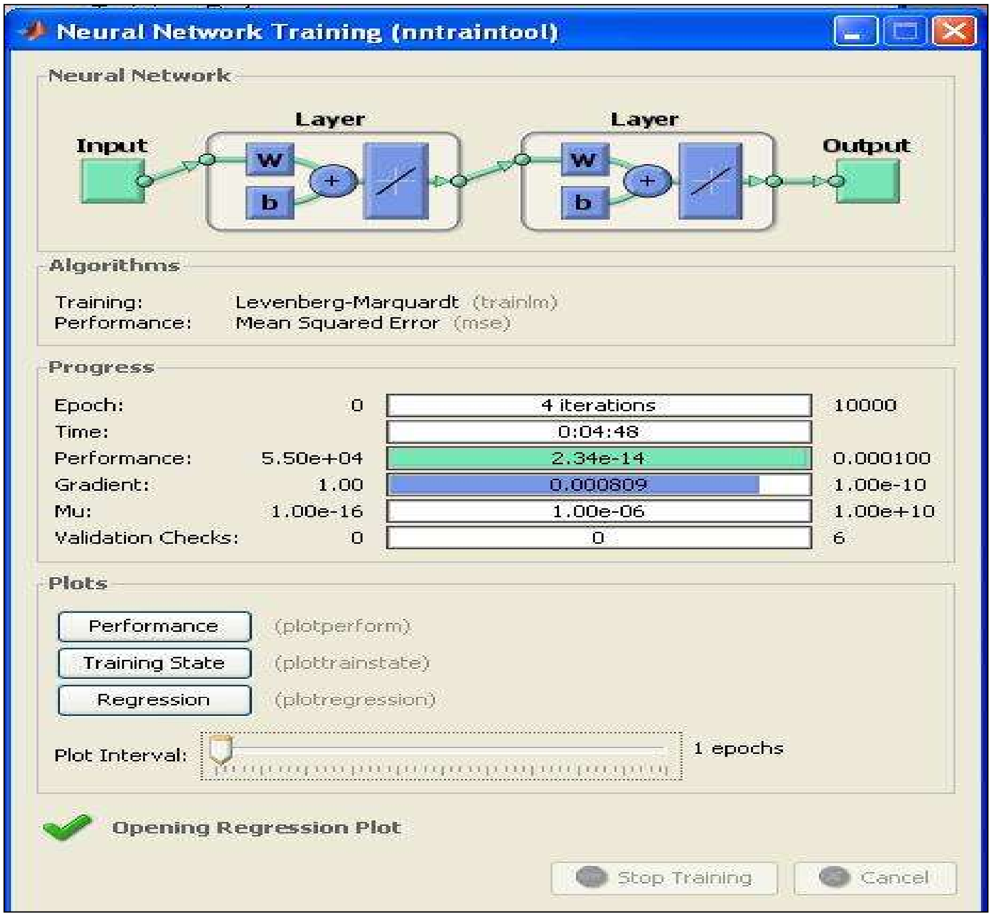


Figure 8: The mean squared error for training the beginning characters.



Neural Network Training (nntraintool)

Neural Network

Algorithms

Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)

Progress

Epoch:	0	4 iterations	10000
Time:		0:04:48	
Performance:	5.50e+04	2.34e-14	0.000100
Gradient:	1.00	0.000809	1.00e-10
Mu:	1.00e-16	1.00e-06	1.00e+10
Validation Checks:	0	0	6

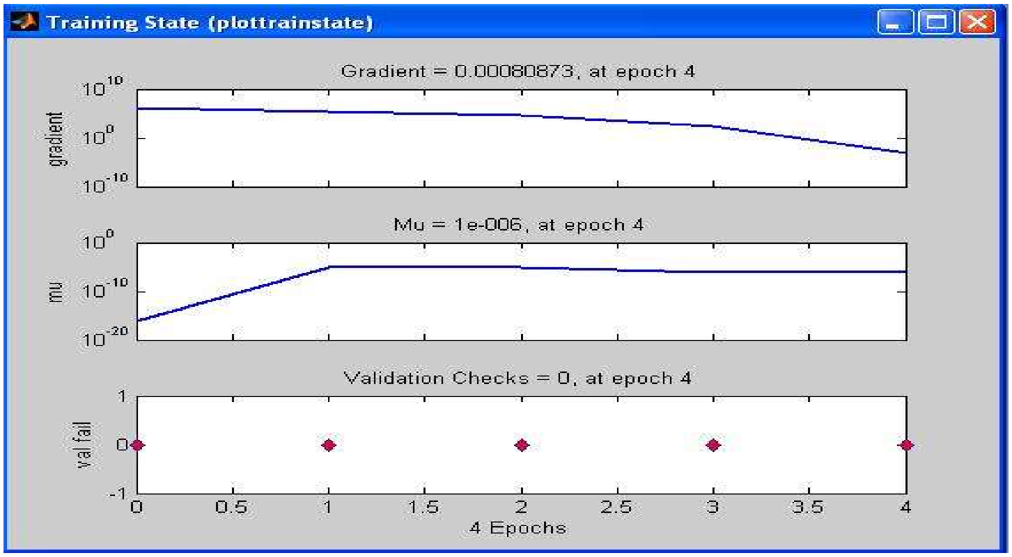
Plots

Performance (plotperform)
Training State (plottrainstate)
Regression (plotregression)

Plot Interval: 1 epochs

Opening Regression Plot

Figure 9: The neural networks tool used for training middle characters.



Training State (plottrainstate)

Figure 10: the training results of middle characters

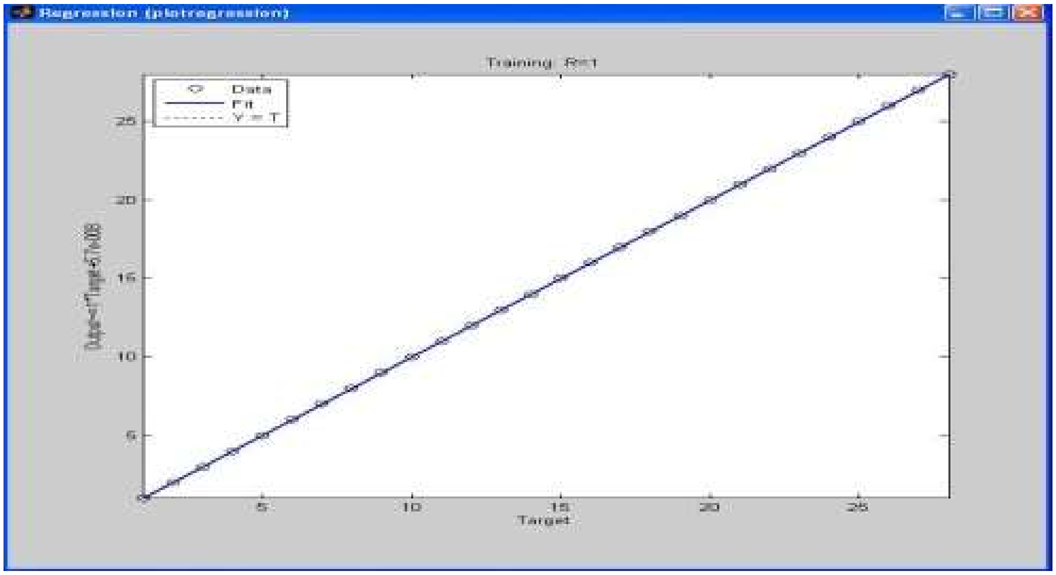


Figure 11: The regression results of training for the middle characters

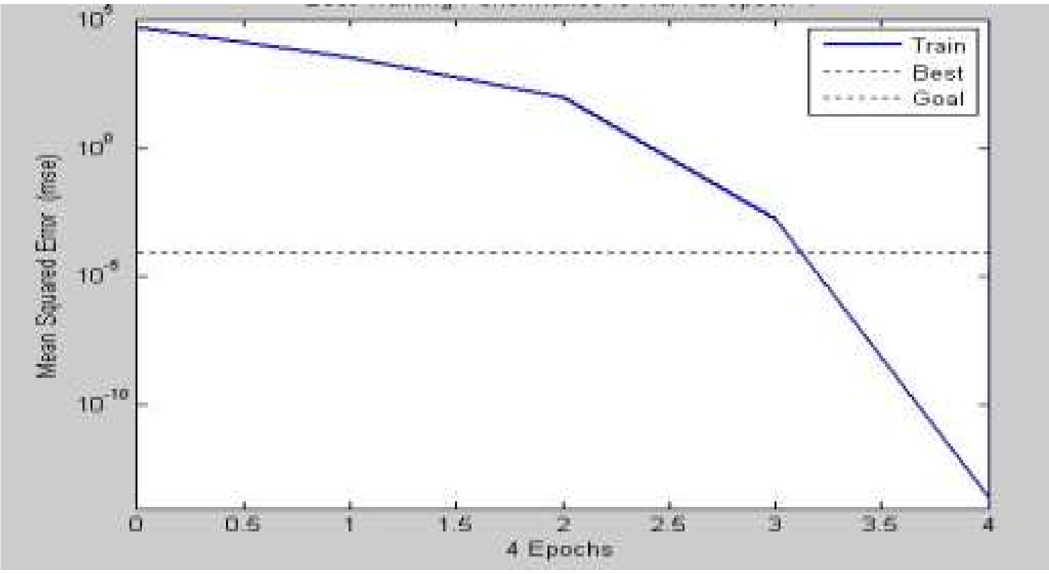
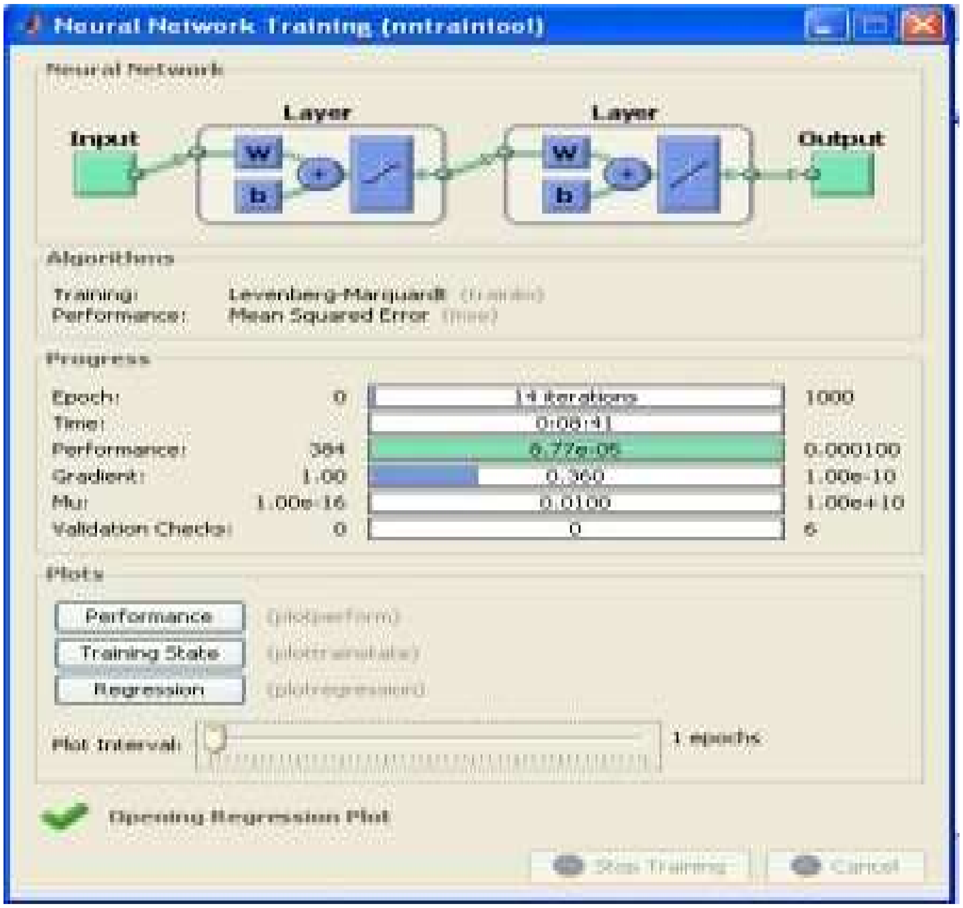


Figure 12: The mean squared error for training the middle characters.



Algorithms

Training: Levenberg-Marquardt (trainlm)
Performance: Mean Squared Error (mse)

Progress

Epoch:	0	14 iterations	1000
Time:		0:08:41	
Performance:	384	6.77e-05	0.000100
Gradient:	1.00	0.360	1.00e-10
Mu:	1.00e-16	0.0100	1.00e+10
Validation Checks:	0	0	6

Plot Interval: 1 epochs

Opening Regression Plot

Figure 13 The neural networks tool used for training end characters.

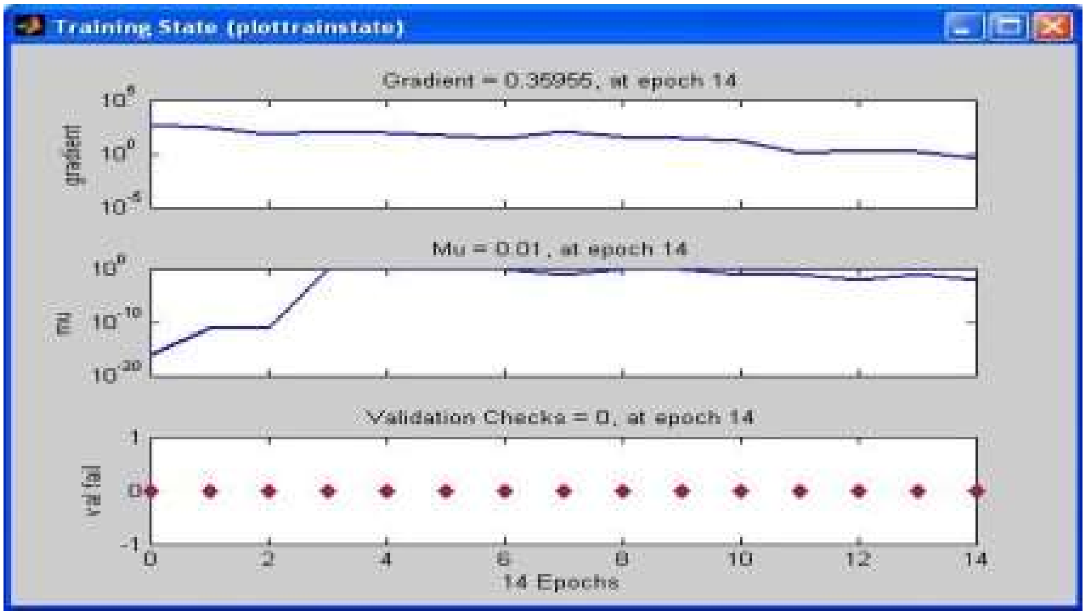


Figure14: The training results of end characters.

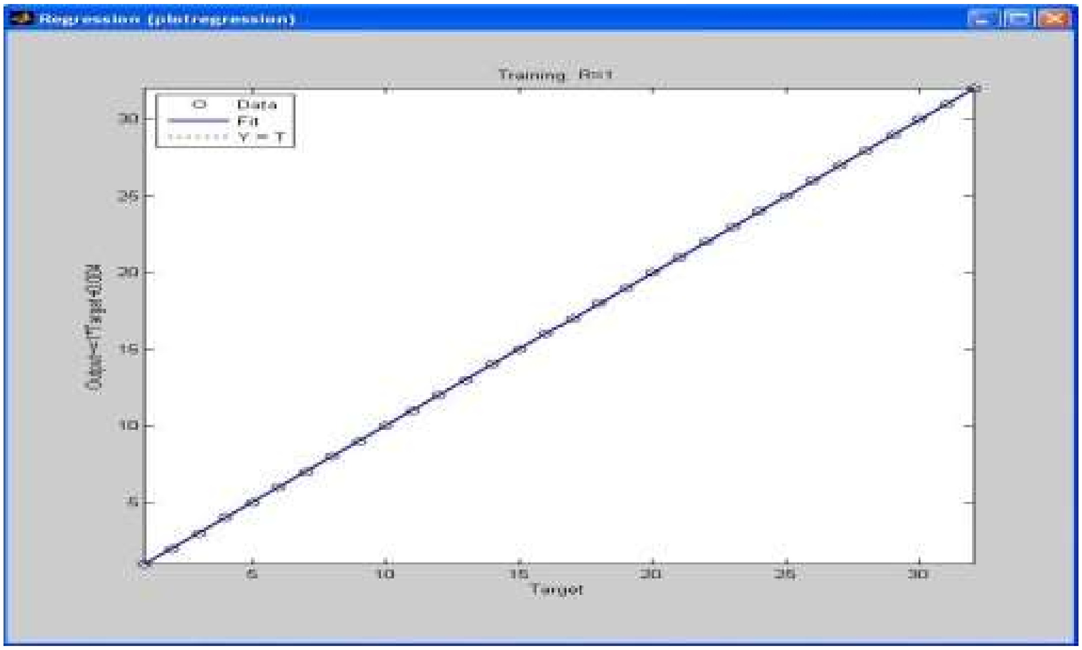


Figure15: The training regression results for the ending characters.

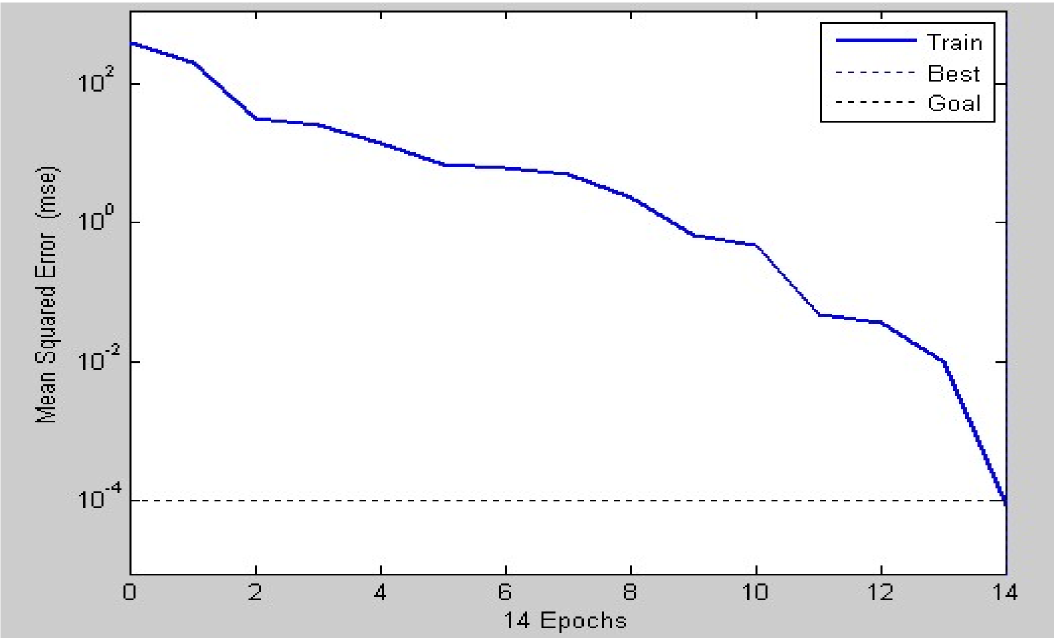


Figure 16: The performance by the mean