

Neural Network with Madaline for Machine Printed English Character Recognition

Shaymaa M. Al-mashahadany

College of Computer Sciences and Mathematics
University of Mosul, Mosul, Iraq

Received on: 15 / 10 / 2009

Accepted on: 21 / 2 / 2010

ABSTRACT

The recognition of optical characters is known to be one of the earliest applications of Artificial Neural Networks, which partially emulate human thinking in the domain of artificial intelligence.

In this research the application of neural networks to the problem of identifying English machine printed characters in an automated manner is developed. A preprocessing step is implemented to separate each character from the others. After that a feature extraction process is applied on each character to obtain the minimum nodes by using Mean, Standard Deviation, and Variance. Madaline neural network is trained on a 26 alphabetical English characters with a standard font and size. And tested on these characters to verify each character image belongs to which type of character. This is done by using MATLAB@2008a.

Keywords: Neural Network, Madaline Network, English letters, recognition.

الشبكة العصبية مع Madaline لتميز الأحرف الانكليزية المطبوعة بطريقة آلية

شيماء المشهداني

كلية علوم الحاسوب والرياضيات، جامعة الموصل

تاريخ قبول البحث: 2010/02/21

تاريخ استلام البحث: 2009/10/15

المخلص

تعتبر عملية تمييز الأحرف من أكثر تطبيقات الشبكات العصبية المستخدمة بشكل مبكر والتي تحاكي التفكير البشري في مجال الذكاء الاصطناعي.

تم في هذا البحث تطبيق الشبكات العصبية لغرض تمييز الأحرف الانكليزية المطبوعة بطريقة آلية. وقد تمت عملية معالجة أولية على الصورة المدخلة لفصل الأحرف عن بعضها. تبدأ بعد ذلك عملية استخلاص الخواص من الحرف للحصول على اقل مدخلات للشبكة العصبية من خلال اعتماد المتوسط والتباين والانحراف المعياري. تم تدريب شبكة Madaline على 26 حرف انكليزي بشكل وحجم قياسيين واختبار الشبكة على كافة الأحرف. وقد أعطت الاختبارات التي تم إجراؤها على الشبكة تطابق كامل لكافة صور الأحرف المدخلة. تم انجاز البحث باستخدام برنامج Matlab.

الكلمات المفتاحية: الشبكة العصبية، شبكة Madaline، الاحرف الإنكليزية، تمييز.

1. Introduction

Neural networks have been applied to a wide variety of areas including speech synthesis, character recognition, diagnostic problems, medicine, business and finance, robotic control, signal processing, computer vision and many other problems that fall under the category of pattern recognition^[1]. Neural networks have been shown to be particularly useful in solving problems where traditional artificial intelligence techniques involving symbolic methods have failed or proved inefficient. Such networks have shown promise in problems involving low-level tasks that are computationally intensive, including vision, character recognition, speech recognition, and many other problems that fall under the category of pattern recognition^[2].

One of the most classical applications of the Artificial Neural Network is the Character Recognition System. This system is the base for many different types of

applications in various fields, many of which we use in our daily lives.^[3] Cost effective and less time consuming, businesses, post offices, banks, security systems, and even the field of robotics employ this system as the base of their operations. When processing a check, performing an eye/face scan at the airport entrance, or even teaching a robot to pick up and object, a system of Character Recognition is employed^[3].

Character recognition, both human and machine generated, is a wide and largely studied field in Machine Learning. In fact, nowadays many commercial scanners use Optical Character Recognition (OCR) systems that output a character string having an image of typed text as input^[4].

For some application areas including pattern recognition, neural models show promise in achieving human-like performance over more traditional artificial intelligence techniques^[4].

In this research for machine printed character recognition system for English language applied, with standard font and size based on a Madaline neural network model, is developed and done by using Matlab software.

Some algorithms regarding the English character recognition deal with seven moments which is applied by so many OCR's^[5], also some approaches deal with neural network techniques and are in practical^[6]. Later on the genetic algorithms are introduced to be used in the character detection field^[7].

Besides this introductory introduction, this research is organized as follows: Section 2 describes the principles of character recognition. The concepts of Neural Networks are introduced in Section 3. Section 4 contains a full description of the proposed algorithm. Results and comparisons are given in Section 5. Finally, Section 6 concludes this paper.

2. Character Recognition Principles

The field of character recognition is one of the major fields of the pattern recognition area which has been the subject of much research in the past three decades^[8]. A Character Recognition or Optical Character Recognition (OCR), is the process of converting scanned images of machine printed or handwritten text (numerals, letters, and symbols), into a computer format text^[9].

There are two types of character recognition systems: on-line and off-line systems. Each system has its own algorithms and methods. The main difference between them is that in an on-line system the recognition is performed in the time of writing while the off-line recognition is performed after the writing is completed^[10]. In this research, an off-line recognition system is used.

In an on-line recognition system, which is also referred to as real time or dynamic recognition, the machine recognizes the symbols as they are drawn. Within such system, the direction of drawing is important, and there is no need for skeletonization or contour extraction. This property makes the recognition stage of the on-line systems easier than it is in the off-line systems. The digitizing tablet is used as a writing surface in most on-line systems^{[11][12]}.

In off-line recognition systems, input text is read and digitized by an optical scanner. Each character is then located and segmented. The resulting array is fed into a preprocessor for smoothing, elimination of noise, size normalization and other operations, to facilitate the extraction of features in the subsequent stage^{[11][12]}.

3. The Concepts of Neural Networks

Neural networks are composed of simple elements operating in parallel. These elements are inspired by biological nervous systems. As in nature, the connections between elements largely determine the network function. You can train a neural network to perform a particular function by adjusting the values of the connections (weights) between elements. [13]

Typically, neural networks are adjusted, or trained, so that a particular input leads to a specific target output. The next figure illustrates such a situation. There, the network is adjusted, based on a comparison of the output and the target, until the network output matches the target.

Typically, many such input/target pairs are needed to train a network. Neural networks have been trained to perform complex functions in various fields, including pattern recognition, identification, classification, speech, vision, and control systems. Neural networks can also be trained to solve problems that are difficult for conventional computers or human being.

4. The Proposed Algorithm

Figure (1) shows an overview of the proposed character recognition system steps applied in this research. The following subsections describe these steps.

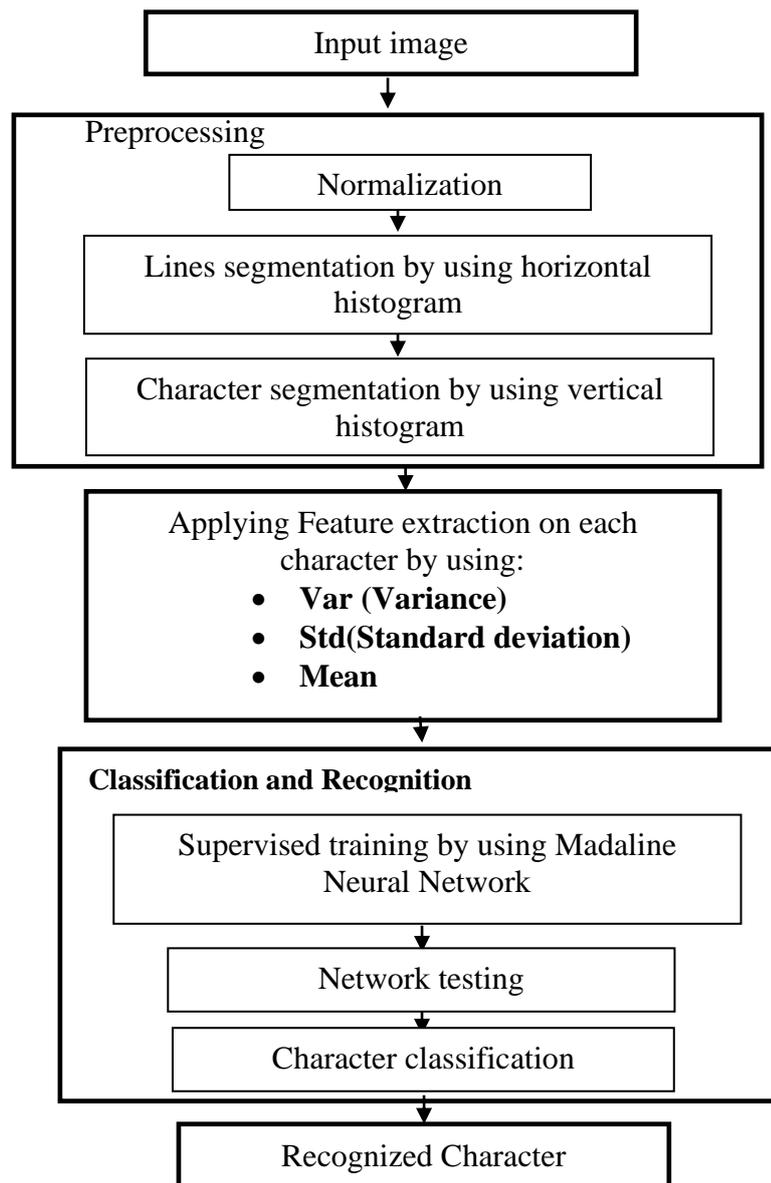


Figure (1): The structure of the character recognition system

4.1 Preprocessing

Preprocessing of the image is done to prepare it for other stage. It increases the accuracy of the recognizing algorithm by applying some of some techniques ^[14].which increase the performance of algorithm to extract the features ^[15].

After importing an image to the system, the preprocessing will begin by separating each line in single slip and the each slip [line] to be truncated to its characters so each character will be processed as single sub-image. The following steps will show the details for that:-

Line Segmentation

Between each two lines there is a free space and it is used to segment lines. Horizontal histogram of image is used to separate lines as shown in Figure (2).

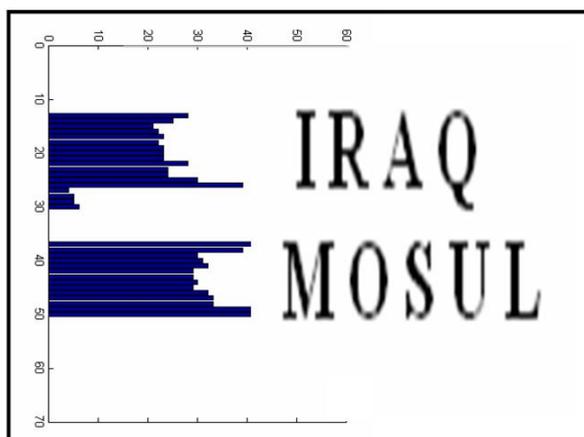


Figure (2): Using horizontal histogram to separate lines

Character segmentation

The printed English characters are separable, so, there is a space between each two characters in the word. In this paper vertical histogram is used to separate the characters as depicted in Figure (3).

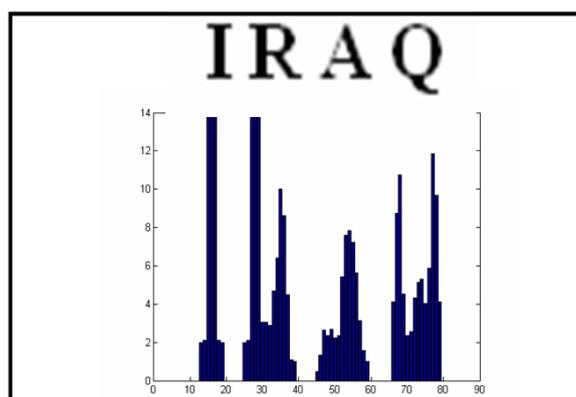


Figure (3): Using vertical histogram to truncate characters

4.2 Feature Extraction

Feature extraction is the process to represent the image by a suitable set of features ^[16]. When the input data to an algorithm is too large to be processed and it is suspected to be notoriously redundant (much data, but not much information) then the

input data will be transformed into a reduced representation set of features (also named features vector). Transforming the input data into the set of features is called *features extraction*. If the features extracted are carefully chosen it is expected that the features set will extract the relevant information from the input data in order to perform the desired task using this reduced representation instead of the full size input ^[17].

The main purpose of the feature extraction process is to reduce the number of inputs into the classifier while maintains the important features or properties of the image ^[18]. The more meaningful of the features that are extracted, the better this feature extractor is. Additionally, the features must satisfy other desirable requirements such as fast processing speed, low computational cost and low complexity of the feature extraction techniques. Thus, simpler and more powerful features cannot be easily found ^[19].

Some researchers apply resizing for each separated character after truncation applies resizing to fit into 25 x 25 image size. The new resized image is then converted to vector of data with 625 dimensions, this vector is sent to a module which extracts features of the data that minimize the number of input data of the neural network to 3 nodes by using variance (Var), standard deviation (Std) and Mean. The three inputs are calculated as follows:

A. Variance (Var)

The variance (σ^2), is defined as the sum of the squared distances of each term in the distribution from the mean (μ), divided by the number of terms in the distribution (N) as given in equation (1).

$$\sigma^2 = \frac{\sum (X - \mu)^2}{N} \quad (1)$$

$V = \text{Var}(X)$ returns the variance of X for vectors. For matrices, $\text{Var}(X)$ is a row vector containing the variance of each column of X. For N-dimensional arrays, Var operates along the first non-singleton dimension of X. The result V is an unbiased estimator of the variance of the population from which X is drawn, as long as X consists of independent, identically distributed samples ^[20].

Var normalizes V by N-1 if N>1, where N is the sample size. This is an unbiased estimator of the variance of the population from which X is drawn, as long as X consists of independent, identically distributed samples. For N=1, V is normalized by N.

B. Standard deviation (Std)

A standard deviation is a measure of dispersion around a central value. To compute the standard deviation, the sum of the squared differences between each individual data point and the average of all the data points is taken and then divided by the number of data points included (or, in the case of sample data, the number of data points included minus one). The square root of this value is then taken to obtain the standard deviation ^[20].

The definition for the standard deviation S of a data vector X given in equation (2) .

$$S = \left(\frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})^2 \right)^{\frac{1}{2}} \quad (2)$$

where

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

and n is the number of elements in the sample.

$S = \text{Std}(X)$, where X is a vector, returns the standard deviation. The result S is the square root of an unbiased estimator of the variance of the population from which X is drawn, as long as X consists of independent, identically distributed samples.

If X is a matrix, $\text{Std}(X)$ returns a row vector containing the standard deviation of the elements of each column of X. If X is a multidimensional array, $\text{Std}(X)$ is the standard deviation of the elements along the first non-singleton dimension of X [21].

C. Mean (Average or mean value of array)

The average is defined as the sum of all data points divided by the number of data points included as shown in equation (3).

$$M = \frac{\sum_{i=1}^n x_i}{n} \tag{3}$$

It is a measure of central tendency and is the most commonly used $M = \text{mean}(X)$, returns the mean values of the elements along different dimensions of an array [21].

If X is a vector, $\text{mean}(X)$ returns the mean value of X. If X is a matrix, $\text{mean}(X)$ treats the columns of X as vectors, returning a row vector of mean values. If X is a multidimensional array, $\text{mean}(X)$ treats the values along the first non-singleton dimension as vectors, returning an array of mean values [21].

4.3 The Madaline Neural Network

The Madaline (Many Adaline) is a multilayer extension of the single-neuron bipolar Adaline to a network. It is also due to B. Widrow (1988), its basic structure is given in Figure (4) which is in terms of two layers of Adalines, plus an input layer which merely serves as a network's input distributor [22].

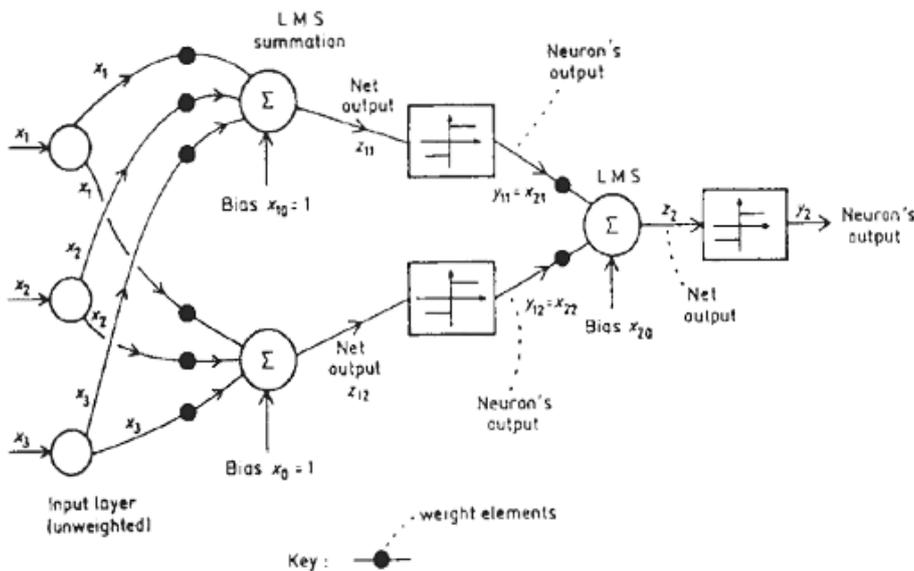


Figure (4): The Madaline Network of 2 layers

Madaline Training

Madaline training differs from Adaline training in that no partial desired outputs of the inside layers are or can be available. The inside layers are thus termed hidden layers. Just as in the human Central Nervous System (CNS), we may receive learning information in terms of desired and undesired outcome, though the human is not conscious of outcomes of individual neurons inside the CNS that participate in that learning, so in ANN no information of inside layers of neurons is available [22].

The Madaline employs a training procedure known as Madaline Rule II, which is based on a Minimum Disturbance Principle, [23] as follows :

1. All weights are initialized at low random values. Subsequently, a training set of L input vectors x_i ($i = 1, 2, \dots, L$) is applied one vector at a time to the input.
2. The number of incorrect bipolar values at the output layer is counted and this number is denoted as the error e per a given input vector.
3. For all neurons at the output layer:
 - a. Denoting $[th]$ as the threshold of the activation function (preferably 0), check: $[z-th]$ for every input vector of the given training set of vectors for the particular layer that is considered at this step. Select the first unset neuron from the above but which corresponds to the lowest $abs [z-th]$ occurring over that set of input vectors. Hence, for a case of L input vectors in an input set and for a layer of n neurons, selection is from $n \times L$ values of z .
This is the node that can reverse its polarity by the smallest change in its weights, thus being denoted as the minimum-disturbance neuron, from which the procedures name is derived. A previously unset neuron is a neuron whose weights have not been set yet.
 - b. Subsequently, one should change the weights of the latter neuron such that the bipolar output y of that unit changes randomly.
 - c. The input set of vectors is propagated to the output once again.
 - d. If the change in weight reduced the performance cost "e" of Step 2, then this change is accepted. Else, the original (earlier) weights are restored to that neuron.
4. Repeat Step 3 for all layers except for the input layer.
5. For all neurons of the output layer: Apply Steps 3, 4 for a pair of neurons whose analog node-outputs z are closest to zero, etc.
6. For all neurons of the output layer: Apply Steps 3, 4 for a triplet of neurons whose analog node-outputs are closest to zero, etc.
7. Go to next vector up to the L 'th vector.
8. Repeat for further combinations of L vectors till training is satisfactory.

The same can be repeated for quadruples of neurons, etc. However, this setting then becomes very lengthy and may therefore be unjustified. All weights are initially set to (different) low random values. The values of the weights can be positive or negative within some fixed range, say, between -1 and 1. The initial learning rate μ should be between 1 and 20. For adequate convergence, the number of hidden layer neurons should be at least 3, preferably higher. It is preferable to use a bipolar rather than a binary configuration for the activation function [22][23].

Design of network

The applied network is shown in Figure (5). It is implemented with 3 layers, input (3 neurons), hidden (3 neurons), and output (26 neurons) that represent number of

alphabetical English character from A to Z. The weights of the network are initially set randomly in the range $\{-1, 1\}$. The network is trained to output 1 in the correct position of the output vector and to fill the rest of the output vector with 0's. The element set to 1 indicates the classification of that input pattern.

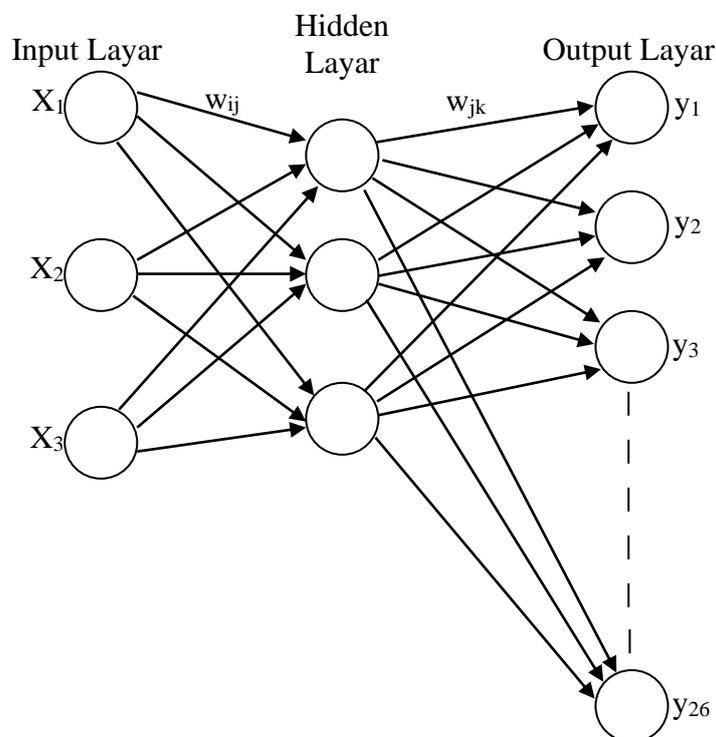


Figure (5): The Network Architecture

Network training algorithm

The following are the basic steps for Training algorithm: ^[23]

1. Generate a training data set with 3 sets.
2. Feed this training set to the network.
3. Set weights of the network randomly in the range $\{-1, 1\}$.
4. Use hard-limit transfer function for each neuron.

$$Y(n) = \begin{cases} +1 & x \geq 0 \\ -1 & x < 0 \end{cases}$$

5. Each output is passed as input to the successive layer.
6. The final output is compared with the desired output and cumulative error for the 3 inputs is calculated.
7. If the input data not equal the desired output, weight is changed using:
8. $\text{weight}_{\text{new}} = \text{weight}_{\text{old}} + 2 * \text{constant} * \text{output}(\text{previous layer}) * \text{error}$
9. Weight(s) are updated and the new error is determined.
10. Weights are updated for various neurons until there is no error or the error is below a desired threshold.
11. Test data set is fed to the network with updated weights and the output (error) is obtained thereby determining the efficiency of the network.

5. Brief comparison with previous work

Most of the OCR deal with different factors such as seven moments which was used with so many scanners and their result was good enough to be obtained by commercial companies. But with the algorithm given in this paper the approach insert the neural network to decrease the recognition time in addition to the percentage of accuracy goes high.

Also the algorithm adopted some statistical factor for feature extraction of the printed character, Table (1) give brief comparison with some methods.

Table (1): Comparison with some methods

Method	Character used	Main features	Percentage
A Stroke-Order Free Chinese Handwriting Input System Based on Relative Stroke Positions and Back-propagation Networks ^[24]	Chinese symbols	Back-Propagation Neural Network with 3 layers	90%
Cursive handwritten word recognition using multiple segmentation determined by contour analysis ^[25]	English characters	Projection profile	70%
On Optical Character Recognition Of Arabic Text ^[26]	Arabic Characters	Segmentation and matching of a candidate character shape to the pre-built prototypes	97%
An Efficient Fuzzy Method for Handwritten Character Recognition ^[27]	English characters	Fuzzy logic	95%
A simple and efficient optical character recognition system for basic symbols in printed Kannada text ^[28]	Kannada characters	Seven Moments + RBF Neural Networks	82%
Offline Handwriting Recognition using Genetic Algorithm ^[29]	English characters	Neural Networks + Genetic algorithms	71%.
Self Evolving Character Recognition using Genetic Operators ^[30]	English characters	Genetic Operators	79.23%
Our approach	English characters	Madaline Neural Network	100%

6. Results and Discussion

The results are obtained by applying the algorithm on very closed characters. They show that a reasonable difference between the parameter adopted [Mean, Srd, and Var] can be seen clearly as in Table (2).

Table (2): Results of applying feature extraction

character	Mean	Std	Var
a	0.9277	0.1418	0.0023
b	0.8829	0.1818	0.0059
c	0.9332	0.1460	0.0030
d	0.8943	0.1716	0.0058
e	0.8924	0.1828	0.0062

f	0.9113	0.1766	0.0057
g	0.9088	0.1584	0.0047
h	0.8644	0.1775	0.0075
i	0.9403	0.1595	0.0050
j	0.9421	0.1482	0.0041
k	0.8886	0.1697	0.0052
l	0.9246	0.1568	0.0047
m	0.8538	0.1635	0.0062
n	0.9017	0.1579	0.0042
o	0.9227	0.1508	0.0035
p	0.9108	0.1716	0.0054
q	0.9080	0.1611	0.0038
r	0.8856	0.1750	0.0057
s	0.9210	0.1667	0.0044
t	0.9174	0.1623	0.0048
u	0.9132	0.1503	0.0050
v	0.9377	0.1242	0.0016
w	0.8875	0.1322	0.0025
x	0.9130	0.1533	0.0028
y	0.9234	0.1445	0.0033
z	0.9230	0.1621	0.0033

The results also show that the time needed to train such network on the parameters of Table (2) is very short (few micro second).

7. Conclusions

In this research the image has been painted using paint program. This image represents multiple numbers of text lines. Each line contains a number of characters with single font and size.

The character image after preprocessing is resized to be of dimensions 25 X 25. This image is converted into a vector and finding the Mean, STD, Var as feature extraction. Using this data as input nodes to the network, the Madaline neural network has been used to recognize the English character patterns by using Matlab@2008a. The Madaline Neural Network is able to train the character data to verify that each character image belongs to which character. The system is able to recognize the samples of characters in training with 26 character images of 100% training recognition rate and testing with 26 images of 100% testing recognition rate which can be seen clearly in Table (2).

REFERENCES

- [1] Ingrid R., Zdravko M., Todd N., "Character Recognition and Learning with Neural Networks", UMAP Journal Vol. 14, No.1, 2005.
- [2] Fakhraddin M. , Jamal F. "Character Recognition using Neural Networks", Near east university, North Cyprus, Turkey via Mersin-10, KKTC,2006. <http://ww1.ucmss.com/books/LFS/CSREA2006/ICA5025.pdf>
- [3] Alexander J., "Using Neural Networks to Create an Adaptive Character Recognition System", Cornell university, Ithaca, New Orleans: ACM Press, 2002. http://alumni.media.mit.edu/~faaborg/research/cornell/hci_neuralnetwork_finalPaper.pdf
- [4] Farah, R.A., M.T. Khadir and M. Sellaim, "Artificial Neural Network Fusion: Application to Arabic words Recognition", ESANN 2005 Proc., Bruges (Belgium), pp: 27-29, 2005.
- [5] Flusser J., "On the independence of rotation moment invariants", Pattern recognition, Vol. 33, PP. 1405-1410, 2000.
- [6] K. khatatneh, I. M. El Emary and B. Al- Rifai, "Probabilistic Artificial Neural Network For Recognizing the Arabic Hand Written Characters", Journal of Computer Science Vol. 2, No. 12, pp.879-884, 2006.
- [7] Gyeonghwan K., Sekwang K., "Feature selection using genetic algorithms for handwritten character recognition", In L.R.B. Schomaker and L.G. Vuurpijl (Eds.), Proceedings of the 17th International Workshop on Frontiers in Handwriting Recognition, Amsterdam, pp 103-112, 2000.
- [8] A. Amin, "Recognition of Hand-Printed Characters Based on Structural Description and Inductive Logic Programming", Pattern Recognition Letters, Vol. 24, pp. 3187- 3196, 2003.
- [9] Deepayan S., "Optical Character Recognition using Neural Networks", university of Wisconsin-Madison, ECE 539 project report, 2003. <http://homepages.cae.wisc.edu/~ece539/project/f03/sarkar-ppt.pdf>
- [10] A. Cheung, M. Bennamoun, N. W. Bergmann, "An Arabic Optical Character Recognition System using Recognition- Based Segmentation", Pattern Recognition, Vol. 34, pp.215-233, 2001.
- [11] R. Plamondon and S. Srihari, "On-line and off-line handwritten recognition: a comprehensive survey", IEEE Transactions on Pattern Analysis and Machine Intelligence 22, pp. 62–84, 2000.
- [12] Ahmed M. Zeki and Mohamad S. Zakaria, "Challenges in Recognition Arabic Characters", the 17th computer national conference (informatics serving Alrahmans Guests), king Abdulaziz university, pp. 445-453, 2004.
- [13] Ben K., Van S. , "An introduction to Neural Networks" , 8th edition, The University of Amsterdam, 1996. <http://www.scribd.com/doc/3998428/An-Introduction-to-Neural-etworksSecond-edition>
- [14] Frederix G., and Pauwels E. "A statistically principled approach to histogram segmentation", Tech. rep., CWI, 2004.

- [15] M. Salmani Jelodar, M.J. Fadaeieslam, N. Mozayani, M. Fazeli, "A Persian OCR System using Morphological Operators", world Academy of science, Engineering and technology, pp.241-244, 2005.
- [16] Duda R., Hart P., and Strok, D." Pattern Classification", John Wiley & Sons, Inc., 2000.
- [17] Isabelle G., André E., "An Introduction to variable and Feature selection", JMLR 3, PP. 1157-1182, 2003.
- [18] Lee S.W., "Multilayer cluster Neural Network for totally unconstrained handwritten numeral recognition", Neural Networks, Vol. 8, No. 5, pp.783-792, 2002.
- [19] D. Dhanya and A. G. Ramakrishnan,"Optimal feature extraction for bilingual OCR", In DAS'02: Proceedings of the 5th International Workshop on Document Analysis Systems-V, pages 25–36, London, UK, 2002.
- [20] Tihomir A., Bengt M., "Robust Chi Square Difference Testing with Mean and Variance Adjusted Test Statistics", Mplus technical appendices No. 10, Los Angeles, CA:Muthen & Muthen, 2006.
- [21] H. Joarder, "Introducing Mean and Variance without Integration by Parts", King Fahd University of Petroleum and Minerals, the Arabian Journal for science and engineering, Vol. 31, No.1B, pp91-102, 2000.
- [22] Daniel G., "Principles of Artificial Neural Networks", 2nd Ed., World Scientific Publishing Co. Pte. Ltd. , University of Illinois, Chicago, USA, 2007.
- [23] Howard D., Mark B., "Neural Network Toolbox User's Guide", The MathWork, Inc, 2008.
- [24] L. Wing-nin, Ch. Kam-Shun, "A Stroke-Order Free Chinese Handwriting Input System Based on Relative Stroke Positions and Back-propagation Networks", Proceedings of the 1996 ACM Symposium on Applied Computing, Philadelphia, 1996.
- [25] H. Yamada and Y. Nakano, "Cursive handwritten word recognition using multiple segmentation determined by contour analysis", IEICE Transactions on Information Systems, 79(5), pp. 464-470,1996.
- [26] A. Zidouri , M. Sarfraz, " On Optical Character Recognition Of Arabic Text", The 6th Saudi Engineering Conference, KFUPM, Dhahran, Vol. 4., pp.109-118, 2002.
- [27] R. Ranawana , V. Palade , and G.E.M.D.C. , "An Efficient Fuzzy Method for Handwritten Character Recognition", LNAI 3214, pp. 698–707, 2004.
- [28] R. Sanjeev Kunte and R. D. Sudhaker Samuel, "A Simple and Efficient Optical Character Recognition System for Basic Symbols in Printed Kannada Text", Sadhana Vol.32, Part5, pp.521–533, 2007.
- [29] Shashank Mathur, Vaibhav Aggarwal, Himanshu Joshi, Anil Ahlawat, "Offline Handwriting Recognition using Genetic Algorithm", Sixth International Conference on Information Research and Applications –i.Tech 2008, Varna, Bulgaria, pp. 21-27, 2008
- [30] Shashank Mathur, "Self Evolving Character Recognition using Genetic Operators", Seventh International Conference on Information Research and Applications –i.Tech 2009, Varna, Bulgaria, pp.132-138, 2009.