

Arabic Character Recognition using ANN Networks and Statistical Analysis

Ahmad M.Sarhan, Omar Helalat
NYIT-New York Institute of Technology

ABSTRACT

In this paper, an Arabic character recognition system based on Artificial Neural Networks (ANN) and statistical analysis of the Arabic characters is presented. In this system, each typed Arabic character is represented by binary values that are used as input to a simple feature extraction system, whose output is fed to an ANN that consists of two layers. Simulation results are provided and show that the proposed system always produces a lower Mean Squared Error (MSE) and higher success rates than the current ANN solutions, especially when the contaminating noise level is low.

1. Introduction

The goal of a character recognition system is to transform a text document typed on paper into a digital format that can be manipulated by word processor software. The system is required to identify a given input data/character form by mapping it to a single character in a given character set. This process can be quite involved since there are several valid forms that a character may take. This is largely due to the many fonts and styles (bold type, italic type, etc....) that can be used. The motivation behind developing character recognition systems is inspired by its wide range of applications including archiving documents, automatic reading of checks, and number plate reading. In this paper, a new scheme for the recognition of typed Arabic characters is developed. The proposed method is compared with the existing current solution to the problem. Simulation results prove that the proposed system always produces a lower MSE than the standard neural-based approach.

1.1 Arabic character recognition

Arabic belongs to the group of Semitic alphabetical scripts in which mainly the consonants are represented in writing, while the markings of vowels (using diacritics) is optional and is rarely used. Arabic is spoken by more than 300 million people and is the official language of many countries.

Tremendous research has been undertaken in the field of recognizing typed and handwritten Latin, Chinese, and Indian characters. Little progress, however, has been made in the recognition of Arabic characters, mainly due to their cursive nature. Unlike most of the other languages, both typed and hand-written Arabic characters are cursive. Furthermore, Arabic characters can take more shapes than Latin characters. Another problem is the unevenness of Arabic fonts; i.e., a certain character in a specific font can be misinterpreted as a different character in another font. In Arabic, some character pairs may be combined together to form another character, that is often referred to as a ligature. The only mandatory ligature is the (Lam Alef). Other ligatures are optional. Ligatures greatly complicate the segmentation task of an Optical Character Recognition (OCR) system.

Other problems facing Arabic recognition systems include:

- Arabic has 28 letters, each of which can be linked in three different ways or separated depending on the case. Therefore, each character can have up to 4 different forms depending on its position.
- Arabic characters have different heights, which hampers noise detection.
- Line mingling (improperly spaced documents).

2. Artificial Neural Networks (ANN)

ANNs were introduced by McCulloch and Pitts in 1943. ANN are trainable algorithms that can "learn" to solve complex problems from training data that consist of a set of pairs of inputs and desired outputs (targets). ANN have been applied successfully in many fields including speech recognition, image processing, and adaptive control. They can be trained to perform a specific task such as prediction, and classification.

ANN consist of interconnected processing elements called *neurons* that work together to produce an output.

2.1 Single Neuron and the Least-Mean-Squares (LMS) algorithm

Fig. 1 shows the structure of a single neuron.

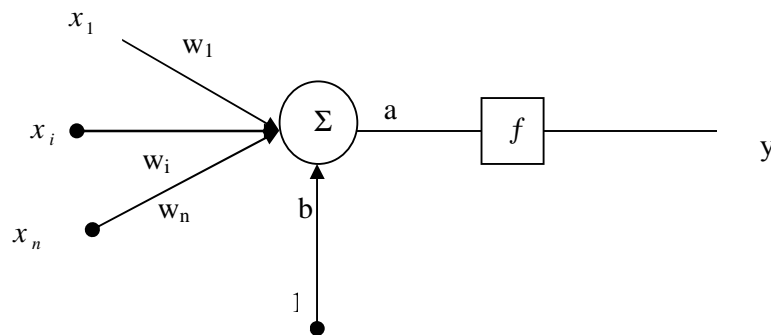


Figure 1: Structure of single neuron

The output a of the neuron is a weighted linear combination of its inputs. The function f in Fig.1 is called the transfer or scaling function. Some commonly used transfer functions are shown in Fig.2.

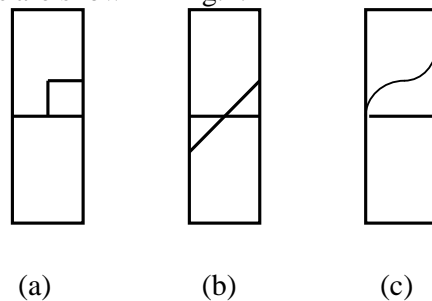


Figure 2: Transfer functions: (a) hard line (b) pure line, and (c) logsig

Using training data (input—target pairs), the weights of the neuron can be iteratively adjusted to give local or global optima. Optimum weights in the sense of least square errors were derived by Widrow and Hoff [9] and the algorithm was called the *LMS* algorithm and is commonly known as the *Widrow-Hoff rule* and has become a widely accepted algorithm. In the LMS algorithm, the network weights are moved along the negative of the gradient of the performance function. Specifically, after each iteration or epoch (new set of input—target pairs) the weights are adjusted according to the following rule

$$\mathbf{W} \leftarrow \mathbf{W} + \mu e \mathbf{X}, \quad (1.1)$$

where μ is the learning /adaptation speed, and the input vector $\mathbf{X} \in \mathbf{R}^n$ is given by

$$\mathbf{X} = [x_1, x_2, x_3, x_4, x_5, \dots, x_n]^T, \quad (1.2)$$

and $\mathbf{W} \in \mathbf{R}^n$ is the vector of weights and is given by

$$\mathbf{W} = [w_1, w_2, w_3, w_4, \dots, w_n]^T. \quad (1.3)$$

The output a of the neuron is defined by the following expression

$$a = w_1 x_1 + w_2 x_2 + w_3 x_3 + \dots + w_n x_n. \quad (1.4)$$

The neuron's scaled output y is given by $y = f(a)$, where f is the transfer function (Fig. 2). The error e is the difference between the neuron's output and the desired output. The initial values of the weights can be set explicitly if a priori information is available. Alternatively and in most practical cases, the weights are initially set to zeros or some random values.

2.2 Multilayer ANN

In general, a multilayer ANN has the architecture depicted in Fig. 3, which shows a two-layer network

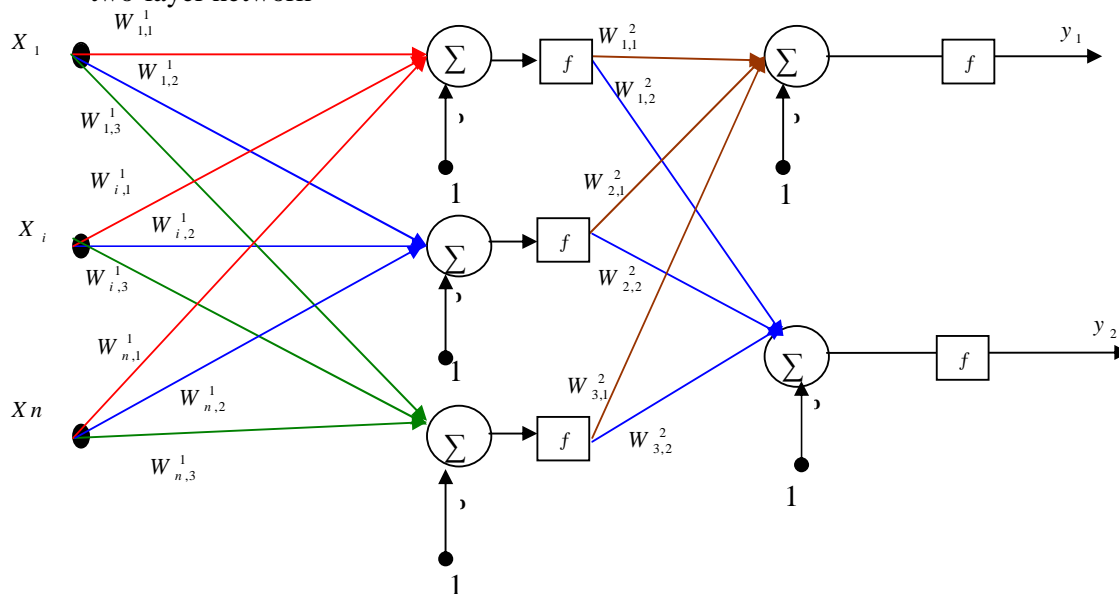


Figure 3: 2-layer ANN

ANN is designed so that the application of a set of inputs produces the desired set of outputs. One of the main characteristics of an ANN is its set of weights. The weights for the first (input) layer and the second (hidden) layer of the ANN in Fig. 3 are shown in Fig. 4.

$$\begin{array}{ccc}
 \left(\begin{array}{ccc}
 W_{1,1}^1 & W_{1,2}^1 & W_{1,3}^1 \\
 \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot \\
 W_{i,1}^1 & W_{i,2}^1 & W_{i,3}^1 \\
 \cdot & \cdot & \cdot \\
 \cdot & \cdot & \cdot \\
 W_{n,1}^1 & W_{n,2}^1 & W_{n,3}^1
 \end{array} \right) & & \left(\begin{array}{cc}
 W_{1,1}^2 & W_{1,2}^2 \\
 W_{2,1}^2 & W_{2,2}^2 \\
 W_{3,1}^2 & W_{3,2}^2
 \end{array} \right) \\
 \text{(a)} & & \text{(b)}
 \end{array}$$

Figure 4: Weight matrices of the ANN in Fig. 3: (a) Layer 1, and (b) Layer 2

Note that $w_{i,j}^k$ denotes the weight for the i^{th} input in the j^{th} neuron of the k^{th} layer.

1.3 Backpropagation Algorithm

When a multilayer ANN uses the Widrow-Hoff learning rule and nonlinear differentiable transfer functions. The resulting ANN, known as Backpropagation, can approximate almost any function with a finite number of discontinuities. Properly trained backpropagation networks have proven to give reasonable answers when presented with inputs that they have never seen. In many cases, it is possible to train a network on a representative set of input—target pairs and get good results without training the network on all possible input—target pairs.

3. Methods

As shown in Fig. 5, the proposed system, referred to henceforth as *netStd*, consists of two stages, a feature extraction stage, whose output is fed to an ANN which constitutes the second stage. Fig. 6 shows the current approach to the Arabic character recognition problem, called here *netOld*. As seen from Fig. 6, *netOld* depends entirely on the ANN to achieve the recognition task and does not include any statistical analysis. In the simulation section, a performance comparison between the two systems is presented.

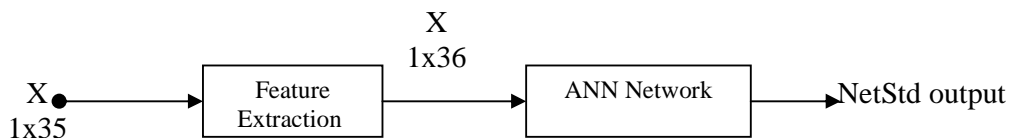


Figure 5: Flow diagram of the proposed network, NetStd

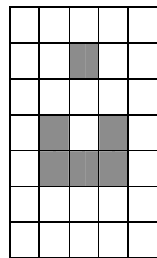


Figure 6: Flow diagram of the standard network, NetOld

3.1 Character modeling

In the proposed system, each character is represented by a matrix of 7×5 binary pixels, producing a 35-element input vector. As an example, Fig. 7 shows the binary image of the Arabic letter "Noon" and its corresponding matrix representation. The number of Arabic characters is 28, producing 28 different classes or system outputs. A careful examination of the matrix representations of the Arabic characters shows that the pixel values are highly uncorrelated. A graph showing the standard deviation of the matrices of the Arabic characters is shown in Fig.8, which reveals that there is a considerable variance in the standard deviations of the Arabic characters. This fact instigated the use of the standard deviation as a valid feature that can significantly help the ANN in its classification job.

Each Arabic character will, therefore, be represented by a vector of 36 elements, with the additional element being the standard deviation of the original 35 elements.



(a)

$$\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

(b)

Figure 7: The Arabic letter "Noon" (a) Bit map image, and (b) matrix representation

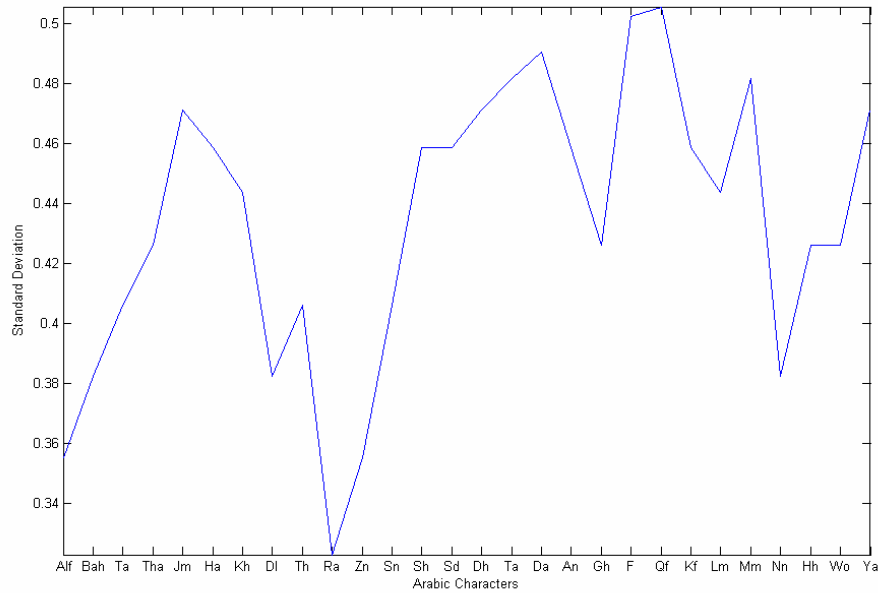


Figure 8: *Standard deviation of the Arabic characters*

3.2 ANN Design

It was shown in [17] and [1] that the best ANN for the problem at hand is a two-layer network (1 hidden and 1 output) with 10 neurons in the hidden layer. The same network is also used here for both netOld and netStd systems. Furthermore, log-sigmoid functions, Eq. (3.1), were used as the transfer functions of the output layer since they can approximate binary values.

$$\log \text{sig}(n) = \frac{1}{1 + e^{-n}} \quad (3.1)$$

The network receives the 36 Boolean values as a 36-element input vector. It has 28 outputs, where each output corresponds to an Arabic character.

4. Simulations:

netOld and netStd systems were trained with sets of noisy and clean inputs. The contaminating noise was the Gaussian noise which has a probability density function (pdf) given by

$$f(x) = \frac{e^{-\frac{(x-\mu)^2}{2\sigma^2}}}{\sigma\sqrt{2\pi}} \quad (3.2)$$

where μ and σ are the mean and standard deviation of the noise, respectively. Both networks were then presented with new sets of noisy inputs, where the noise was additive Gaussian noise, Eq. (3.2). Fig.7 shows the performance of both networks when presented with Gaussian noise of zero mean and varying standard deviation (noise level). As can be seen from Fig. 7, netStd always outperforms netOld and produces a lower MSE. This is especially the case when the contaminating noise level is low.

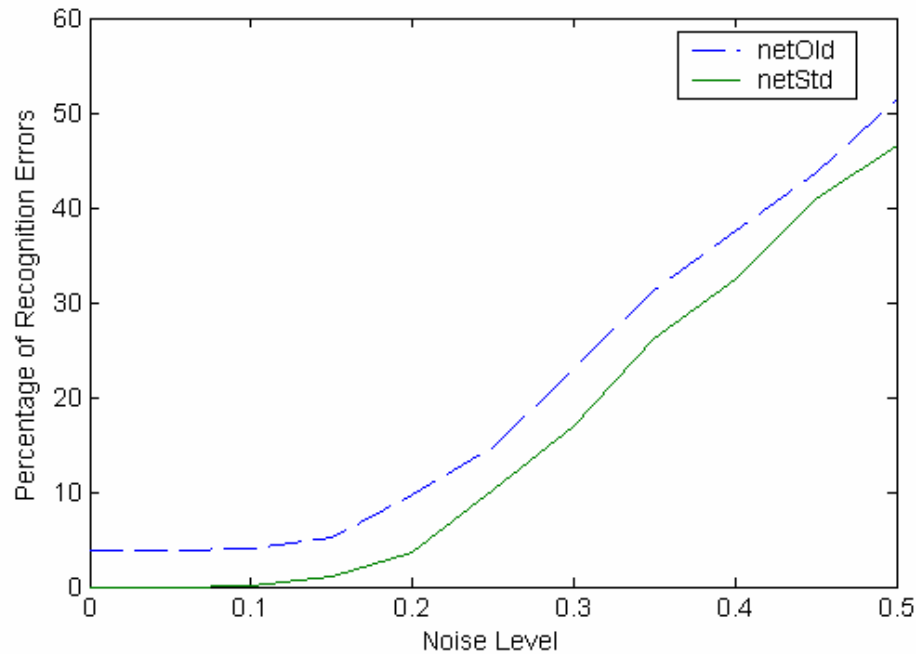


Figure 9: MSE for NetOld and netStd

5. Conclusion

In this paper, a new system for the recognition of typed Arabic characters is presented. The system is composed of two stages, a feature extraction stage and an ANN stage to perform pattern-classification/identification. The system inputs are the Arabic characters, where each character is represented by a matrix of 7 x 5 binary pixels, producing a 35-element input vector, which is presented to the feature extraction stage. It was shown that the standard deviation of the pixel values provides valuable information about the characters. Subsequently, the feature extraction stage obtains the standard deviation of the input vector and produces a 36-element vector with the additional element being the standard deviation of the original 35 values. This output vector is then fed to the ANN. The ANN is composed of two layers. The first layer consists of 10 neurons and the second layer consists of 28 neurons, the number of Arabic characters. Simulation results are provided and show that the proposed system always produces a lower MSE and higher success rates than the current ANN solutions, especially when the contaminating noise level is low.

5. References

- Amin and G. Masini, "Machine recognition of multifold printed Arabic texts", in Proc.8th Int. Conf. Patt. Recogn. (Paris, France),pp. 392-395, 1986
- [1] A.M. Sarhan and R. C. Hardie, "Partition-based filters", In *Proceedings of the 1995 IEEE National Aerospace and Electronic Conference (NAECON)*, volume 2, Dayton, Ohio, May 1995.
- [2] A.M. Sarhan, R. C. Hardie, and K. E. Barner, "Partition-based adaptive estimation of single-response evoked potentials", In *Proceedings of the 1995 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 1995.
- [3] A.M. Sarhan. *nonlinear partition-based filters for signal restoration*. Ph.D. Thesis University of Dayton, July 1996.
- [4] D. J. Hand. *Discrimination and Classification*. New York: Wiley, 1981.
- [5] E. W. Brown, "Character Recognition by Feature Point Extraction", Northeastern University internal paper, 1992
- [6] F. Hussain and J. Cowell, "Character Recognition of Arabic and Latin Scripts", *Proceedings, IEEE International Conference on Information Visualization* ,pp. 51–56, 2000
- [7] H. Al-Yousefi and S. S. Udpa, "Recognition of handwritten Arabic characters," in Proc. SPIE 32nd Ann. Int. Tech. Symp. Opt. Optoelectric Applied Sci. Eng. (San Diego, CA), Aug. 1988.
- [8] Haykn S.,*Adaptive Filter Theory*,Englewood Cli s.N.J: Prentice Hall(3ed) ,1996
- [9] J.F Canny. "A Computational Approach to Edge Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, pp. 679-698, 1986.
- [10] K. E. Barner, A. M. Sarhan, and R. C. Hardie. "Partition-based weighted sum filters for image restoration". *IEEE Transactions on Image Processing*, Vol. 8, No. 5, May 1999.

- [11] K. Khatatneh , "Probabilistic Artificial Neural Network For Recognizing the Arabic. Hand Written Characters", *Journal of Computer Science* 3 (12), 881-886, 2006
- [12] K. Badi and M. Shimura, "Machine recognition of Arabic cursive script" *Trans. Inst. Electron. Commun. Eng.*, Vol. E65, no. 2, pp. 107-114, Feb. 1982.
- [13] K. Badi and M. Shimura, "Machine recognition of Arabic cursive scripts" in *Pattern Recognition in Practice*. Amsterdam: North Holland, 1980.
- [14] L.Hammami and D.Berkani , "Recognition system for printed multi-font and multi-size Arabic characters", *The Arabian Journal for Science and Engineering*, Volume 27, Number 1B, pp:57-72, April, 2002
- [15] M Altuwajri , M.A Bayoumi , "Arabic Text Recognition Using Neural Network" *ISCAS 94. IEEE International Symposium on Circuits and systems*, Volume 6, 30 May-2 June 1994.
- [16] N. Ben Amor, N. Essoukri Ben Amara : "A hybrid approach for Multifont Arabic Characters Recognition", 5th *WSEAS Int.Conf. On Artificial Intelligence, Knowledge Engineering and Data Bases (AIKED'06)* Madrid, Spain, February 15-17, 2006.
- [17] R. A. Dosari, R. C. Hardie, and A. M. Sarhan. "Multi-channel nonlinear filters for signal restoration". In *Proceedings of the 1997 IEEE National Aerospace and Electronic Conference (NAECON)*, Dayton, Ohio, May 1997.