12

Sorting VLSI heuristics serach security

Impact factor = 0.781

IN	3470	US	685
CN	435	PK	246
IR	238	GB	223
ID	155	MY	142
AU	88	EG	88
BR	80	PH	75
DE	71	CA	71
SA	68	NG	62
TW	60	DZ	57
TN	56	FR	52
Newest:		MV You:	NG

Supercounters.com

HOME ABOUT LOGIN REGISTER SEARCH CURRENT ARCHIVES EDITORIAL TEAM

Home > About the Journal > **Editorial Team**

Editorial Team

Editors

Chief Editor, Council for Innovative Research, United States
Satpreet Singh, MCA CEO, ARDASS Corporation, USA, United States
Oye Nathaniel David, Department of Information Systems Faculty of Computer Science and Information Systems Universiti Teknologi Malaysia 81310 Skudai, Johor, Malaysia, Malaysia, Malaysia
Editor in Chief
Rajan Manro, Assistant Professor and Coordinator, Desh Bhagat Instiitute of Mangement and Computer Science, Mandi Gobindgarh, India
Dr. M.K. Sharma, Associate Professor & Head MCA Program Department of Computer Science Amrapali Institute -Haldwani (Uttarakhand), India
Dr. Shalini Vermani, Associate Professor in Apeejay School of Management, Dwarka, New Delhi, India, India
Samriti sharma, Assistant Professor SSIET, Patti, Tarn Taran
Dr. Rajinder Virk, Guru Nanak Dev University Amritsar, India
Dr. Dolia Prashant Mahendrabhai, Assistant Professor, Department of Computer Science, Maharaja Krishnakumarsinhji Bhavnagar University, Bhavnagar, India
Dr. G. Arutchelvan, Director,Adhiparasakthi College of Science, Kalavai

Section Editor

Chief Editor, Council for Innovative Research, United States

ISSN: 2278-5183
www.ijcdsonline.com

International Journal of Computers and Distributed Systems
Vol. No.3, Issue II, Jun-July 2013

BACK-PROPAGATION ARTIFICIAL NEURAL NETWORK TECHNIQUES FOR OPTICAL CHARACTER RECOGNITION – A SURVEY

[1]Dike U. Ike, [2]Adoghe U. Anthony
[1]Department of Electrical and Information Engineering, Covenant University, Ota.
[2] Department of Electrical and Information Engineering, Covenant University, Ota.

ABSTRACT
Character recognition, a digitization concept, is an important research area in the field of image processing, and pattern recognition. Optical character recognition is a method of digitizing printed texts so that they can be searched electronically, stored compactly and used in machine processes such as text-to-speech, and machine translation. This paper describes the techniques for converting a type or handwritten document into machine readable form using Back-Propagation Artificial Neural Networks.

KEYWORDS - digitization, image processing, optical character recognition, back-propagation artificial neural network.

1.0 INTRODUCTION

Optical Character Recognition (OCR) is a system that provides a complete alphanumeric recognition of printed or handwritten characters at electronic speed by simply scanning the document [1]. Type or handwritten documents are scanned by the use of a scanner and fed to the OCR device which recognizes the characters in the document and converts them into machine recognizable ASCII data. There are two major methods used for implementing OCR, they include: Matrix or Template Matching and Feature Extraction. Matrix Matching compares the output of the OCR scanner with a library of templates or character matrices. When an image matches with one of the given matrices of dots with a specified level of similarity, the computer tags the image as the corresponding ASCII character. Feature Extraction technique is without strict matching to specified templates. The computer searches for general features such as closed shapes, open areas, diagonal lines, intersecting lines, and so on. This method is more flexible than Matrix Matching.
This paper shows how the use of Back-Propagation Artificial Neural Network simplifies the development of an Optical Character Recognition Application while achieving maximum quality of recognition.

2.0 OPTICAL CHARACTER RECOGNITION

Optical Character Recognition (OCR) is the process of translating images of handwritten or printed text into a format that is easily understood by computers for the purpose of editing, indexing and formatting. Much of the world's information is in hard copy documents, OCR systems liberate this information by converting the text on paper into electric form.
The most vital process in the implementation of OCR with Artificial Neural Network is the selection of input to the Feed Forward Backward Propagation Artificial Neural Network.

2.1 STAGES IN CHARACTER RECOGNITION

The stages that are common to many character recognition systems are as follows:
I. **Optical Scanning** - A digital image of the original document is gotten through scanning. Optical scanners have a sensing device that converts light intensity into gray levels. Thus the multilevel image is converted into a black and white bi-level image
II. **Segmentation** - this is performed in order to determine the constituents of an image, so that the parts of the document with printed data can be located and differentiated from figures or, and graphics.
III. **Feature Extraction** – this is performed in order to capture the vital characteristics of the symbol. Here, certain features that characterize the symbols, while excluding unnecessary features are extracted.
IV. **Classification** – this is performed using the feature vectors computed in the feature extraction. In this paper, the feature vector is fed into an artificial neural network for character recognition.
The stages are summarized in Figure 1 below.

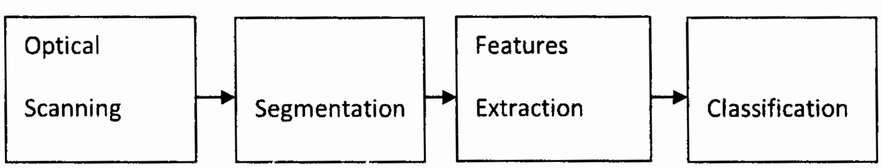

Figure 1: Stages in Optical Character Recognition

2.2 PRINCIPLES OF ARTIFICIAL NEURAL NETWORK

Artificial Neural Networks are electronic models based on the neural structure of the brain [1]. Artificial neural networks are parallel computing systems consisting of a large number of processors interconnected. A neural network is a set of parallel elements called neurons, which work together to achieve a specific goal which is set by the designer. Artificial neural networks are different from a programmed algorithm, because they must be trained in order to function. This algorithms are divided into supervised training and unsupervised training. In supervised training, the network learns by example whereas in unsupervised learning no example is given. Unsupervised learning is very difficult and complex to implement [2]. Although computing these days is truly advanced, there are certain tasks that a program made for a common microprocessor is unable to perform. Similar to the brain, artificial neural networks transforms inputs into outputs to the best of its ability, because it is composed of many neuron like elements that co-operate to perform the desired function. Figure 1 below gives a simplified similarity between artificial neural networks and the brain.

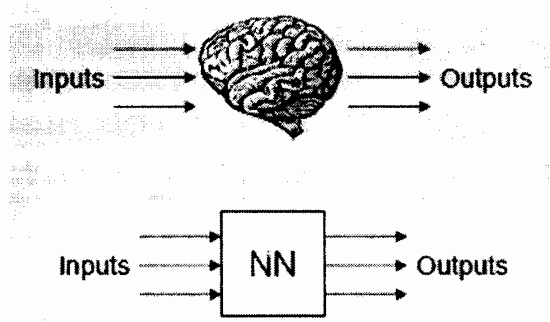

Figure 1: Artificial Neural Network and Brain Analogy

The output of a neuron is a function of the weighted sum of the inputs plus a bias. Figure 2 shows diagrammatic view of the output of a neuron.

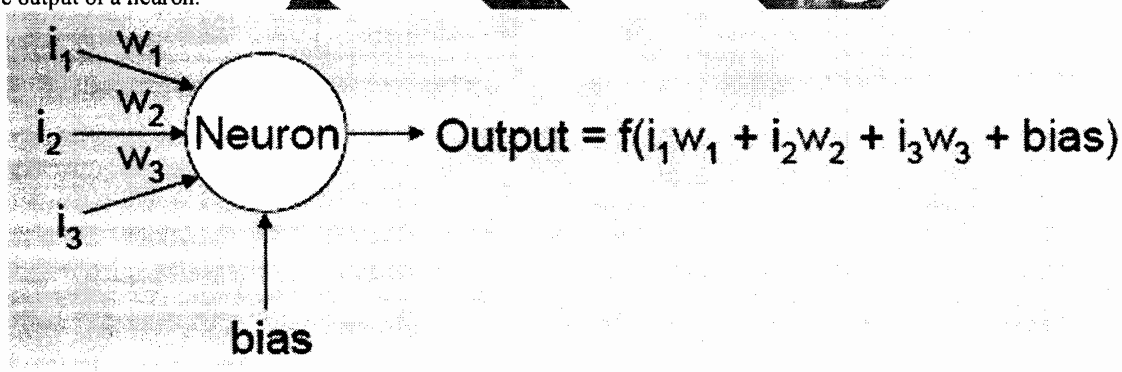

Figure 2: The Output of a Neuron

The function of the entire neural network is simply a deterministic computation of the inputs of all the neurons.

2.3 COMPONENTS OF AN ARTIFICIAL NEURAL NETWORK

I. **Weighting Factors** – each input to an Artificial Neural Network (ANN), has its own weight which gives the input the strength it needs to impact the processing element summation function. Some inputs are given more weight than others so that they have more effect on the processing element. Weights can be adjusted in response to various learning algorithms.

II. **Summation Function** – the weighted sum of all inputs to an ANN is done using the summation function. The inputs and weight coefficients can be added in many ways before passing it to the transfer function. Apart from the product summing of the input and weighting coefficient, the summation function can select the maximum, minimum or several other normalizing algorithms depending on the network architecture. In order to allow the summation output to vary with time, an additional activation function is applied.

III. **Transfer Function** – The summation function result, usually the weighted sum, is transformed to a standard output through an algorithm function known as transfer function. In the transfer function the weighted sum can be compared with some threshold to give the neural output. If the sum is more than the threshold value, the processing element generates a signal, if the weighted sum of the input is less than the threshold, the processing element generates no signal. The transfer function is usually non – linear, its output of is proportional to its input, making it of little computational use. Figure below shows some sample transfer functions.

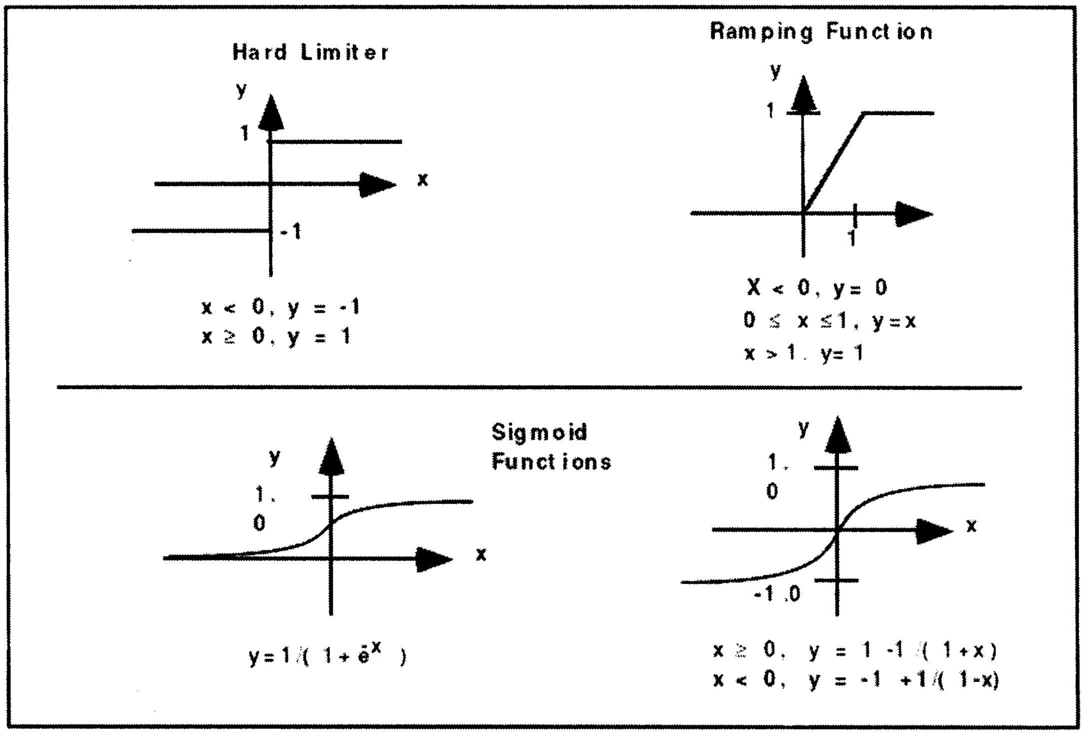

Figure 4: Sample Transfer Functions [1]

The transfer function could be a step function (hard limiter) which output one and minus one, zero and one, or other numeric combination, depending on whether the result of the summation function is positive or negative. The transfer function could be a ramping function which mirrors the input within a given range. It is a linear function that has been clipped to minimum and maximum values, thus making it non-linear . The transfer function of choice is the sigmoid function, the curve of the function approaches a minimum and maximum value at the asymptotes. The curve is called a sigmoid when it ranges between zero and one, it is called a hyperbolic tangent if it ranges between minus one and one. The sigmoid function and its derivatives are continuous. Other transfer functions exists which are dedicated to specific network topologies.

IV. **Output Function** – each processing node is allowed one output signal which it may output to other neurons. Most times the output is equivalent to the result of transfer function. Some network architectures modify the result to incorporate competition among neighbouring nodes.

V. **Error Function** – in majority of learning networks, the difference between the current output and the desired output is an error. This error is transformed by the error function to match a given network topology. Some topologies use this error directly, some square the error and retain its sign, others modify the error to fit their own use. The error is then propagated into the learning function of another node, this error term is usually called the current error. The current error is usually propagated backwards to a previous layer. This back propagated value can either be the current error, the current error scaled in some way (usually by the derivative of the transfer function) or some other output depending on the network topology. Usually this back propagated value, after scaling by the learning function is multiplied by each of the incoming connection weight so as to modify them prior to the next learning cycle.

VI. **Learning Function** – it modifies the variable connection weights on the inputs of each node according to some algorithm. This process of changing the weights of the input connections to achieve some desired result can also be called the adaptation function, as well as the learning mode.

2.4 BACK- PROPAGATION ARTIFICIAL NEURAL NETWORK

The back-propagation algorithm is applied in feed-forward artificial neural networks (ANNs). The nodes are organized in layers, and send their signals forward with errors propagated backwards. The network receives inputs by neurons in the input layer, and the output of the network is obtained by the neurons on an output layer. There may be several hidden layers. Each stage is connected to the next layer as shown in figure below. The back-propagation algorithm uses supervised learning, which means that we provide the algorithm with examples of the input and outputs we want the network to compute, and then the error is computed. The aim of the back-propagation algorithm is to reduce the error, until the ANN learns the training data. The training starts with random weights, and aim is to adjust them to arrive at minimal error. The number of layers and the number of artificial neurons per layer are important decisions to make when applying this architecture. The complexity between the input data and desired output determines the number of nodes in the hidden layer. Also, the amount of training data sets set an upper bound for the number of nodes in the hidden layer. This upper bound is calculated by dividing the number of input-output pairs examples in the training set by the total number of input and output nodes in the network. Then divide the result by scaling factor between five and ten.

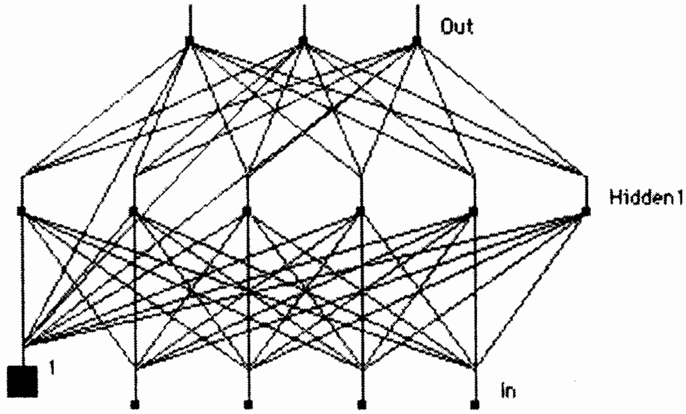

Figure 5: A Feed Forward Back-Propagation Network [1]

The activation function A_j is a weighted sum of the inputs x_i multiplied by their respective weights w_{ji}:

$$A_j(\bar{x}, \bar{w}) = \sum_{i=0}^{n} x_i w_{ji} \qquad (1)$$

The activation depends only on the inputs and the weights and the output function O_j is the sigmoid function:

$$O_j(\bar{x}, \bar{w}) = \frac{1}{1 + e^{A(\bar{x}, \bar{w})}} \qquad (2)$$

The output depends on the activation function, which depends on the weighted value of the inputs. The aim of the training process is to obtain a desired output when certain inputs are given. Since the error is the difference between the actual and the desired output the error depends on the weights, and we need to adjust the weights in order to minimize the error [4]. The error function of the output of each artificial neuron is given as:

$$E_j(\bar{x}, \bar{w}, d) = \left(O_j(\bar{x}, \bar{w}) - d_j\right)^2 \qquad (3)$$

The square of the difference between the output and the desired output is used in order to have an always positive result and higher precision. The error E of the network is the sum of the errors of all the artificial neurons in the output layer:

$$E(\bar{x}, \bar{w}, \bar{d}) = \sum_j \left(O_j(\bar{x}, \bar{w}) - d_j\right)^2 \qquad (4)$$

The weights are adjusted depending on their impact on the error E using the method of gradient descendent:

$$\Delta w_{ji} = -\eta \frac{\partial E}{\partial w_{ji}} \qquad (5)$$

Equation (5) is applied until we find appropriate weights resulting to minimum error.

The goal of the back-propagation algorithm is to find the derivative of E with respect to w_{ji} and since we need to achieve this backwards. First, we need to calculate how much the error depends on the output O_j, which is the derivative of E with respect to O_j from equation (3):

$$\frac{\partial E}{\partial O_j} = 2(O_j - d_j) \tag{6}$$

We then calculate how much the output depends on the activation, which also depends on the weights from equation (1) and equation (2):

$$\frac{\partial O_j}{\partial w_{ji}} = \frac{\partial O_j}{\partial A_j}\frac{\partial A_j}{\partial w_{ji}} = O_j(1-O_j)x_i \tag{7}$$

From equations (6) and (7):

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial O_j}\frac{\partial O_j}{\partial w_{ji}} = 2(O_j - d_j)O_j(1-O_j)x_i \tag{8}$$

The necessary adjustment to each weight is gotten from equations (5) and (8):

$$\Delta w_{ji} = -2\eta(O_j - d_j)O_j(1-O_j)x_i \tag{9}$$

Equation (9) is applied in training an ANN with two layers. Some additions will be made in order to train the network with one extra layer. If we want to adjust the weight of a previous layer, we first calculate how the error depends on the input from the previous layer. Here we use v_{ik} as the weight, and then we simply replace x_i with w_{ji} in equations (7), (8), and (9). We then determine how the error of the network depends on the adjustment of v_{ik}. Thus we have:

$$\Delta v_{ik} = -\eta\frac{\partial E}{\partial v_{ik}} = -\eta\frac{\partial E}{\partial x_i}\frac{\partial x_i}{\partial v_{ik}} \tag{10}$$

Where:

$$\frac{\partial E}{\partial Xi} = 2(O_j - d_j)\, O_j\, (1-O_j)\, w_{ji} \tag{11}$$

Assuming that there are set of inputs x_i into the neuron with v_{ik}, and then we have from eqution (7):

$$\frac{\partial x_i}{\partial v_{ik}} = x_i(1-x_i)v_{ik}$$

If we want to add yet another layer, we can do the same, calculating how the error depends on the inputs and weights of the first layer [3].

3.0 ARTIFICIAL NEURAL NETWORK TECHNIQUES

Many techniques have been used to design an OCR using ANN back-propagation algorithm. Some of the major techniques include: vertical and horizontal projection, character matrix feature vector, fundamental feature method, and structure adapting network.

3.1 VERTICAL AND HORIZONTAL PROJECTION FEATURE VECTOR [4]

This technique uses a feed-forward artificial neural network using back propagation algorithm. The inputs are first normalized to scale to 32by32 pixels. The feature vector used comprises of the concatenation of 32 elements from the vertical feature vector and 32 elements from the horizontal feature vector [5]. The artificial neural network consists of 64 input neurons that are connected to one hidden layer, and one output layer. The output layer has a neuron for each output class. Each character to be recognized has a unique output class. It takes 1255 iterations to drop the error to a value below 0.0001 using a learning rate of 0.90. The system gives a character recognition accuracy of 90% for noiseless input, and 70% for noisy input.

3.2 CHARACTER MATRIX FEATURE VECTOR [6]

This method utilises a feed-forward artificial neural network using back-propagation algorithm. The system can accurately recognize 12 popular font faces with sizes ranging from 8 to 32 point. The image is first pre-processed by thresholding the image to reduce noise, centering the character on the characters centroid and normalization through scaling [5]. The input layer contains 2500 neurons. The input image consists of a 50by50 pixel matrix, the feature vector is a 2500 element vector, which is fed to the neural network. The input layer connects to a hidden layer having 100 neurons, the hidden layer connects to an output layer having 94 neurons, each neuron corresponds to a given character class. The network is trained using the centroid dithered images to 8,650,000 iterations to reach a mean squared error of 0.000002.

3.3 FUNDAMENTAL FEATURE VECTOR [7]

This method utilizes 17 character features as inputs to a neural network, and it can detect isolated capital letters from 16 different fonts. The character recognition is performed by a back propagation neural network. The 17 features contain 14 general features and 3 special features used to differentiate certain characters. The features include observations based on line slope, curvature, space interconnection, line interconnection, and other geometrical and topological features. The 17 features serve as input to a 17 nodes input layer, each node corresponds to a feature in the feature vector. The hidden layer comprises of 12 nodes connected to a single output node. The network is trained using the Levenberg-Marquardt algorithm to 100 iterations to reach a mean squared error of 0.082.

3.4 STRUCTURE ADAPTING NETWORK [8]

This method utilizes a structure adapting approach to an artificial neural network. The method recognizes single fonts of multiple sizes. The features used for this network and run length coding of the characters. Run length encoding (RLE) is a loss-less method of feature extraction where a sequence of consecutive values of data are stored as a single value and count. RLE works best in binary images of characters. Consider the following example where the input length is 30 and the RLE is only 3 (value, count) pairs.

S = 0000011111000

RLE = {(0,5), (1,5), (0,3)}

3.5 COMPARISON OF THE VARIOUS FEATURE EXTRACTION TECHNIQUES

Table 1 below gives a brief summary of the differences between the various techniques surveyed in this paper.

Table 1: Comparison Of The Various Feature Extraction ANN Techniques

Feature Extraction	Lossy/Loss-less	Font Size	Number of Fonts	Accuracy
Character Matrix	Lossless	Any Size	12	100%
Vertical and Horizontal Projection	Lossy	Any Size	1	99%
Fundamental Features	Lossy	Any Size	16	99.3%
Run Length Encoding	Lossless	Any Size	1	99%

4.0 CONCLUSION

A number of techniques that are used for Optical Character Recognition which uses Back-Propagation Artificial Neural Network have been discussed. Artificial Neural Networks are widely used to perform Optical Character Recognition due to their very high noise tolerance, thus the system yields excellent result. The most important step in Optical Character Recognition is feature extraction. An incorrectly chosen feature set will lead to poor classification rates by any Artificial Neural Network. The main research is currently going on in extending Optical Character Recognition to all popular native languages in Nigeria like Hausa, Yoruba, Igbo, and so on. Regardless of the complexity involved, Artificial Neural Networks offer significant benefits in Back-Propagation Network and classification in the sense of emulating to a small extent adaptive human intelligence.

REFERENCES

[1] Dave Anderson, George Mc Neil. Artificial Neural Networks Technology. Kaman Sciences Corporation, New York, 1992.

[2] Ahmad M. Sarhan, and Omar I. Al Helalat, "Arabic Character Recognition using Artificial Neural Networks and Statistical Analysis".

[3] Carlos Gershenson. Artificial Neural Networks for Beginners.

[4] N. Mani and B. Srinivasan. Application of Artificial Neural Network Model for Optical Character Recognition Systems, Man, and Cybernetics, 1997. 'Computational Cybernetics and Simulation', 1997 IEEE International Conference on, Vol. 3, pp.2517-2520, October 1997.

[5] Introduction to Artificial Intelligence OCR using Artificial Neural Networks by Kurt Alfred Kluever (kurt@kloover.com) Department of Computer Science Golisano College of Computing and Information Sciences Rochester Institute of Technology February 18, 2008.

[6] H.I. Avi-Itzhak, T.A. Diep, and H. Garland. High accuracy optical character recognition using neural networks with centroid dithering. Transactions on Pattern Analysis and Machine Intelligence, 17(2):218{224, Feb 1995.

[7] E.M. de A. Neves, A. Gonzaga, and A.F.F. Slaets. A Multi-Font Character Recognition Based On Its Fundamental Features By Artificial Neural Networks. Proceedings., Second Workshop On Cybernetic Vision, pp. 196-201, December 1996.

[8] J.M. Ramirez, P. Gomez-Gil, and D. Baez-Lopez. On Structural Adaptability of Neural Networks In Character Recognition. 3rd International Conference on Signal Processing, Vol. 2, pp.1481-1483, October 1996.