ADAPTIVE NEURAL NETWORK CLASSIFIER FOR EXTRACTED INVARIANTS OF HANDWRITTEN DIGITS

L. H. Keng and S. M. Shamsuddin

Faculty of Computer Science and Information System
Universiti Teknologi Malaysia, 81310 Skudai, Johor

Tel: 07-5532321, Fax: 07-5565044, E-mail: mariyam@fsksm.utm.my
E-mail: kelvinlhk@hotmail.com

ABSTRACT

We propose an adaptive activation function of neural network classifier for isolated handwritten digits that undergo basic transformations. The utilised network is a backpropagation network with sigmoid and arctangent activation functions. The performance of the network with both activation functions is compared. The results show that the network applying an adaptive activation function between layers converged much faster when compared to non-adaptive activation functions with 50% iteration reduction. In this study, we also present experimental results of feature extraction between Zernike and δ-geometric for better feature representations. Results show that Zernike features are better at representing isolated handwritten digits compared to δ-geometric features with accuracy of up to 87%.

Key words: Handwritten Digit, zernike moments, δ-Geometric moments, adaptive activation function

1.0 INTRODUCTION

A character recognition system classifies membership of a digit in the input image. There are several important processing steps between the acquisition of input digit image and the output class membership decision. Generally, it consists of preprocessing, feature extraction, and classification (Figure 1).

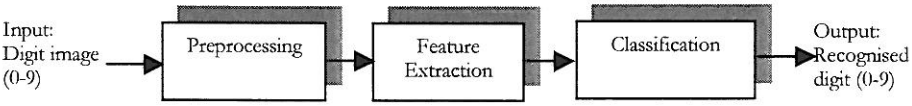

Fig. 1 : Digits Recognition System

The system starts with the scanning of the text image and converting it into an electric signal (Cao, 1995). Generally, there are two kinds of digits inside the text: connected and isolated. This study focuses on the recognition of isolated handwritten digits. The feature extraction stage converts the input image into feature vectors and removes redundancy from the data. Since each individual has his/her own writing style, the input of handwritten digits will vary in style and shape, and is subject to different types of distortion, such as scaling, translation, and rotation. Feature vectors are calculated such that they have small intraclass invariance and larger interclass separation (Altuwijri and Bayoumi, 1994).

Various approaches have been proposed by researchers to try to capture the distinctive features of characters. They generally fall into global analysis and structural analysis (Suen *et al.*, 1992). Global analysis techniques and points distribution features includes template matching, moments, profile projection, border chain-code histograms, mathematical transformations while structural analysis techniques such as loops, end points, junction arcs, concavities and convexities, etc. (Lam and Suen, 1988). Structural and topological features use geometrical and structural properties to represent handwritten digits. They are generally difficult to compute and often sensitive to small distortion. Global analysis and point distribution features are generally less sensitive to distortion and easier to compute (Ye, 2001). Moment functions (global analysis) are studied by researchers around the world because the set of moments computed from a digital image represents global characteristics of the image shape and provides lots of information about the different types of geometrical features of the image (Mukundan and Ramakrishnan, 1998). Various moment functions have been introduced over the years. The most widely used moment functions in research are geometric and Zernike moments. A study by (Khotanzad and

Lu (1990) showed that Zernike features performed better than geometric features with presence of noise. However, when using noiseless images, the performance is almost the same. Zakaria (1987) introduced a faster computation method for geometric moments using δ-method that use only the boundary points of the image region. This study employs δ-method for features extraction and compares the accuracy of classification with Zernike moments and also the geometric moments.

The final step in most character recognition systems is to execute a decision rule. The decision rule can be implemented based on syntactical and statistical techniques. Many researchers have dedicated significant efforts to design classifiers over the past three decades. The classification procedures proposed based on these techniques include nearest neighbor classifiers (Chung *et al.*, 1998), relaxation matching (Lam and Suen, 1988), tree classifiers (Moret, 1982) and model-based classifiers (Mitchel and Gillies, 1989). However, most commonly used classifiers become optimal only when the amount of training data is very large, and feature statistics are perfectly known (Ye, 2001). Developments in the field of neural networks have provided potential alternatives to the traditional techniques of pattern recognition (Khotanzad and Lu, 1990). The neural network approach offers capabilities for solving many difficult problems and has been shown to be very effective in handwritten character classification. Multilayer perceptron (MLP) with standard back-propagation (BP) algorithm is the most commonly used neural network classifier in handwritten digits recognition and has proven to be better than other statistical classifiers such as Bayes, nearest neighbour and minimum-mean-distance (Khotanzad and Lu, 1990; Chung *et al.*, 1998; Agui *et al.*, 1991). However, it suffers from slow convergence during the network training phase, which may be caused by the activation function and initial weights. The activation function used in BP algorithm has significant impact on the convergence of network training. This is because when the output approaches to either of the extreme values of 0 or ±1, the value of the derivative diminishes producing a very small back-propagated error signal resulting in a very small weight change. Thus the output can be maximally wrong without producing any significant weight change. This study applies the adaptive activation function which incorporates the arctangent function between input and hidden layer, and the sigmoid logistic function between hidden and output layer. The comparisons are made between network applying adaptive activation function and sigmoid logistic function in terms of network convergence and accuracy of classification.

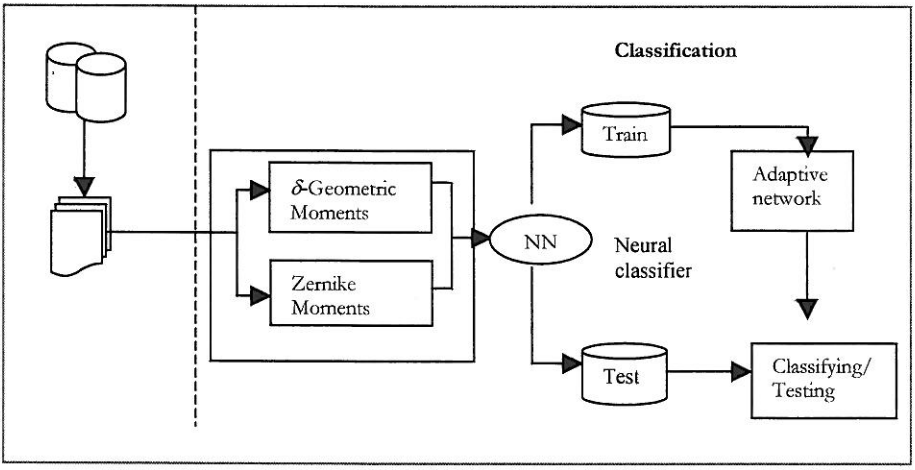

Fig. 2: An Adaptive NN Classifier for Handwritten Digits Recognition

The framework of this study is shown in Figure 2. For each input image, feature extraction is performed with Zernike, δ-Geometric and ordinary geometric moment functions. These features are normalised within the interval of [0, 1] before being fed into the neural classifier for training and classification. Recognition rates of all features as well as the iterations needed for network training are calculated for comparison.

2.0 FEATURE EXTRACTION

Feature extraction is a process by which an initial measurement pattern (binary image in this study) is transformed into a new set of numerical features (Altuwijri and Bayoumi, 1994). These features are used to classify the pattern (digit). Selection of good features is a crucial step in the process since the next stage (classification) sees only these features and acts upon them. Good features are those satisfying two requirements (Khotanzad and Lu, 1990):

i) small intraclass invariance–slightly different shapes with similar general characteristics should have numerically close values and,

ii) larger interclass separation–features from different classes should be quite different numerically.

According to Ye (2001), another criteria for choosing features is the computation time of features extraction. A flexible recognition system must be able to recognise an object regardless of its orientation, size, and location in the field of view.

2.1 Zernike Moments

Teague (1980) introduced Zernike moments based on the orthogonal functions called Zernike polynomials defined over the polar coordinates inside a unit circle. The Zernike moment of order p is defined as,

$$Z_{pq} = \frac{(p+1)}{\pi} \int_0^{2\pi} \int_0^1 V_{pq} * (r,\theta)\, f(r,\theta) r \; dr\, d\theta \;,\;\; r \leq 1, \tag{1}$$

where

p is a non-negative integer and q is an integer such that p-$|q|$ is even and $|q| \leq p$.The function $V_{pq}(r,\theta)$ denotes Zernike polynomials of order p with repetition q, and * denotes the complex conjugate. If N is the number of pixels along each axis of the image, then the above equation can be written in the discrete form as,

$$Z_{pq} = \frac{(p+1)}{\pi(N-1)^2} \sum_{x=1}^{N} \sum_{y=1}^{N} V_{pq} * (r,\theta)\, f(x,y), \tag{2}$$

where

$$r = \frac{(x^2 + y^2)^{1/2}}{N} \text{ and } \quad \theta = \tan^{-1}(\frac{y}{x}).$$

Zernike Polynomials

Zernike polynomials are an orthogonal set of complex-valued polynomials (Mukundan and Ramakrishnan, 1998; Dehghan and Faez, 1997).

$$V_{nm}(x, y) = \mathrm{R}_{nm}(x, y) e^{(im \tan^{-1}(\frac{y}{x}))}, \tag{3}$$

where R_{nm} is a real-valued radial polynomial defined as,

$$R_{nm}(x,y)=\sum_{s=0}^{\frac{n-|m|}{2}}(-1)^s\frac{(n-s)!}{s!(\frac{n-2s+|m|}{2})!(\frac{n-2s-|m|}{2})!}(x^2+y^2)^{n-2s},\ n=0,1,2,...,\infty;$$

$0\leq|m|\leq n$ and $n-|m|$ is even.

To achieve scale and translation invariants for Zernike moments, the image needs to be normalised and Zernike features are then extracted from the normalised image (Chung *et al.*, 1998). Scale and translation normalisation is carried out using regular geometric moments. Zernike moment invariants and their corresponding expressions in geometric moments used in this study are given below:

$$\begin{aligned}
&Z_{20}=(3/\pi)\left[2(m_{20}+m_{02})-m_{00}\right]\\
&|Z_{22}|^2=(3/\pi)^2\left[(m_{20}-m_{02})^2+4m_{11}{}^2\right]\\
&|Z_{31}|^2=(12/\pi)^2\left[(m_{30}+m_{12})^2+(m_{03}+m_{21})^2\right]\\
&|Z_{33}|^2=(4/\pi)^2\left[(m_{30}-3m_{12})^2+(m_{03}-3m_{21})^2\right].
\end{aligned}\tag{4}$$

2.2 δ-Geometric Moments

Geometric moments (also known as *Cartesian* or *regular moments*) are the simplest among moment functions with the kernel function defined as a product of the pixel coordinates (Mukundan and Ramakrishnan, 1998). The 2D geometric moments of order $(p+q)$ of a density distribution function $\rho(x,y)$ are defined as (Hu, 1962),

$$m_{pq}=\int\int x^p y^q \rho(x,y)\ dx\,dy,\qquad p,q=0,1,2,\ldots\tag{5}$$

For digital image, the integral are replaced by summations,

$$m_{pq}=\sum_x\sum_y(x)^p(y)^q\ \rho(x,y).\qquad p,q=0,1,2,\ldots\tag{6}$$

Zakaria (1987) introduced the δ-method for computing the geometric moments of a binary image using only the boundary points of the image region. A binary image defined over a closed convex region ζ can be represented by the set, $(< x_{1k}, x_{2k} >, y_k)$, $k = 0,1,2,...n$ where x_{1k}, x_{2k} denote the x-coordinates of the boundary of the k^{th} image row having an ordinate value y_k. Among all the image rows, y_0 and y_n are the minimum and maximum values of the y-coordinates respectively, and n denotes the number of image rows in the region (Figure 3).

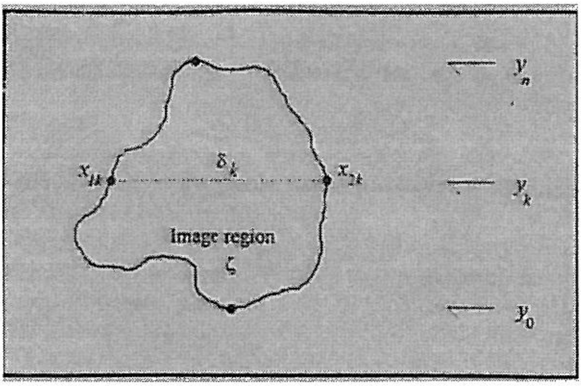

Fig. 3 : Image Region in Terms of Boundary Points

δ_k is the distance between the boundary pixels on the k^{th} row,

$$| x_{2k} - x_{1k} | = \delta_k \qquad ; \qquad | y_{k+1} - y_k | = 1.$$

Equation (6) can be rewritten as,

$$m_{pq} = \sum_{k=1}^{n} (y_k)^q \sum_{i=1}^{\delta_k} (x_{1k} + i)^p . \tag{7}$$

The above expression can be evaluated for different values of p and q. For example,

$$m_{0_q} = \sum_{k=1}^{n} \delta_k (y^k)^q, \qquad q = 0,1,2,3,....$$

$$m_{1_q} = \sum_{k=1}^{n} \left[\delta_k x_{1k} + (\delta_k^{\;2} - \delta_k)/2\right](y_k)^q, \qquad q = 0,1,2,3,...$$

$$m_{2_q} = \sum_{k=1}^{n} \left[\delta_k x_{1k}^{\;2} + (\delta_k^{\;2} - \delta_k) + (2\delta_k^{\;3} - 3\delta_k^{\;2} + \delta_k)/6\right](y_k)^q, q = 0,1,2,3,... \tag{8}$$

From the above equations, the double summations over each pixel of the image in equation (6) can be reduced to single summations over the boundary points using the factor δ_k. In addition, to obtain translation invariance, central moments are used:

$$\mu_{pq} = \sum_x \sum_y (x - x_0)^p (y - y_0)^q \, \rho(x, y). \tag{9}$$

where the intensity centroid (x_0, y_0) is given by,

$$x_0 = \frac{m_{10}}{m_{00}} \text{ and } y_0 = \frac{m_{01}}{m_{00}}.$$

Central moments is normalised to become invariant to scale change by defining

$$\eta_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{(p+q+2)/2}} \tag{10}$$

Thus a set of geometric moment invariants to translation, scale, and rotation is derived (Hu, 1962). A set of second and third order geometric moment invariants is given.

$$\begin{aligned}
\varphi_1 &= \eta_{20} + \eta_{02} \\
\varphi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}{}^2 \\
\varphi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\varphi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\varphi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\
&\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
\varphi_6 &= (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})
\end{aligned} \tag{11}$$

The numerical values of φ_1 to φ_6 are very small (Khotanzad and Lu, 1990; Chung *et al.*, 1998), the logarithms of the absolute values of the functions i.e. $\log_{10} \mid \varphi_i \mid$ i = 1, 2, …, 6 are used as features representing the image.

3.0 MULTILAYER PERCEPTRON (MLP) CLASSIFIER

In this study, the multilayer perceptron network (MLP) (Figure 4) with back-propagation (BP) learning is used. Adaptive activation functions are experimented with the arctangent function between input and hidden layer, and the sigmoid logistic function between hidden and output layer. Three layers of a fully connected network have been used with only one hidden layer of 9 hidden nodes. All output nodes are set to 0 except for the node that is marked 1 to correspond to the class the input is from. The inputs either the δ-geometric moment features or the Zernike moment features extracted from the image.

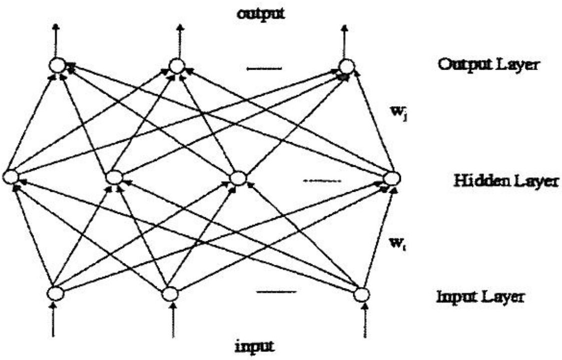

Fig. 4 : Architecture of MLP with One Hidden Layer

3.1 Activation Function

An activation function is a nonlinear function that, when applied to the net input of a neuron, determines the output of that neuron (Masters, 1993). The commonly used activation function in back-propagation learning is either sigmoid logistic or hyperbolic tangent. The sigmoid (f_s) and hyperbolic tangent (f_{th}) are expressed as,

$$o = f_s(net) = \frac{1}{1+e^{-net}}$$

and

$$o = f_{th}(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}}. \qquad (12)$$

respectively.

Their first derivatives are calculated as

$$f_s'(net) = o(1-o)$$

and

$$f_{th}'(net) = (1-o^2) \tag{13}$$

respectively.

In this study, the adaptive activation function is used to study the effectiveness of adaptive learning to network convergence and classification. An arctangent function is applied to the input-to-hidden layer while a sigmoid logistic function is used for the hidden-to-output layer. The arctangent function (*f_{it}*) and its first derivative is expressed as,

$$o = f_{it}(net) = \arctan(net)$$

and

$$f_{it}'(net) = \cos^2 O \tag{14}$$

respectively.

The error signal of the arctangent function to be used in the neural network from hidden-to-input layer is shown in the following section.

3.2 Change of Weights between Hidden and Input Layers

The change of weights between hidden (j) and input (i) layers is given as,

$$\Delta W_{ji} \propto -\frac{\partial E}{\partial W_{ji}} = -\eta \frac{\partial E}{\partial W_{ji}}.$$

By chain rule,

$$\frac{\partial E}{\partial W_{ji}} = \frac{\partial E}{\partial net_j} \bullet \frac{\partial net_j}{\partial W_{ji}} \tag{15}$$

Know that

$$net_j = \sum_j W_{ji} O_i + \theta_j$$

$$\therefore \frac{\partial net_j}{\partial W_{ji}} = O_i \ . \qquad (16)$$

Let $\delta_j = -\frac{\partial E}{\partial net_j}$ where $\delta_j =$ [error signal] from hidden layer to the input layer. Substituting

$\delta_j = -\frac{\partial E}{\partial net_j}$ and $\frac{\partial net_j}{\partial W_{ji}} = O_i$ into (15) gives

$$\frac{\partial E}{\partial W_{ji}} = -\delta_j O_i \ . \qquad (17)$$

Therefore, the weight between input layer and hidden layer can be adapted according to

$$\Delta W_{ji} = \eta \delta_j O_i \ . \qquad (18)$$

By chain rule,

$$\delta_j = \frac{\partial E}{\partial net_j} = \frac{\partial E}{\partial O_j} \bullet \frac{\partial O_j}{\partial net_j} . \qquad (19)$$

It is known that $O_j = f\left(net_j\right) = \tan^{-1}\left(net_j\right)$

so, $\frac{\partial O_j}{\partial net_j} = \frac{1}{1 + net_j^{\ 2}} = \cos^2 O_j$.

Thus, $\frac{\partial E}{\partial net_j} = \frac{\partial E}{\partial O_j} \bullet \cos^2 O_j$ (20)

and using the chain rule,

$$\frac{\partial E}{\partial O_j} = \frac{\partial E}{\partial net_k} \bullet \frac{\partial net_k}{\partial O_j} \quad . \tag{21}$$

From $net_k = \sum_k W_{kj} O_j + \theta$

we get $\frac{\partial net_k}{\partial O_j} = W_{kj}$ (22)

thus,

$$\frac{\partial E}{\partial O_j} = \delta_k W_{kj} .$$

Now, $\frac{\partial E}{\partial net_j} = \delta_k W_{kj} \bullet \cos^2 O_j = \delta_j$ (23)

therefore,

$$\Delta W_{ji} = \eta \delta_j O_i$$

$$= \eta \delta_k W_{kj} \cos^2 O_j \bullet O_i . \tag{24}$$

With a momentum term, α

$$\Delta W_{ji}(n) = \eta \delta_k W_{kj} \cos^2 O_j \bullet O_i + \alpha \Delta W_{ji}(n-1), \tag{25}$$

and the adaptation of the weight follows $W_{ji} = W_{ji}(n-1) + \Delta W_{ji}(n)$

4.0 EXPERIMENTAL RESULTS

The collection of isolated handwritten digits was taken from Technion–Israel Institute of Technology at http://www.ee.technionac.il/courses/046195. This dataset contains isolated handwritten digits in grayscale format with uniform size of 28 x 28 pixels. These samples are then converted into a binary raw file to be used in feature extraction. A training set of 300 samples was randomly chosen and 100 were used as a test set. Figure 5 is a sample of the handwritten digits in the dataset.

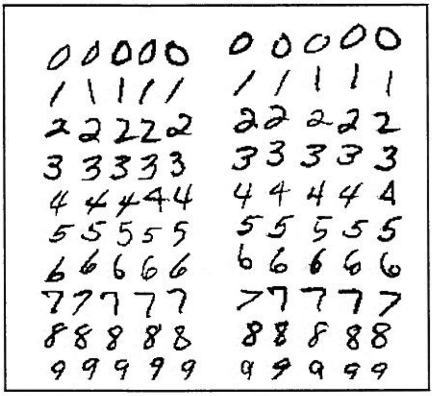

Fig. 5 : Sample of Isolated Handwritten Digits

Six features of δ-geometric and ordinary geometric moments, and four features of Zernike moment invariants were used as the input of the neural network. This means that for testing δ-geometric or ordinary geometric moments, the input node for the neural net is 6 whereas 4 input nodes were used for Zernike moments. The number of output nodes is dependent on the number of patterns to be classified. In this study, there are 10 digits to be classified. Therefore, 10 nodes are used where all nodes are set to 0 except for the node that is marked 1 to correspond to the class of input. For example, for digit 0, the output node is 1 for the first node and 0 for the remaining nodes.

Table 1 shows the convergence of network training applying only the sigmoid logistic activation function and using adaptive activation functions. The application of the adaptive activation function showed significant reduction in

Table 1 : Convergence Rates of Sigmoid and Adaptive Activation Function (# of Iterations)

Group	Sigmoid Function			Adaptive Function		
	Zernike	δ-Geometric	Geometric	Zernike	δ-Geometric	Geometric
Sample 1	11367	11348	8759	4317	5282	3379
Sample 2	11508	54131	8216	5191	18614	5548
Sample 3	12153	24044	10678	5329	8699	10175
Sample 4	7217	19983	5375	3521	7384	1719

the number of iteration needed to train the network. For instance, the numberof iteration needed to train the Zernike features in sample 1 is reduced from 11367 iterations (using the sigmoid logistic function) to only 4317 iterations using the adaptive function, a reduction of 62% while for δ-Geometric in sample 3, the reduction is 64%.

Table 2: Recognition Rates of Zernike, δ-Geometric and Geometric Invariants Using Sigmoid and Adaptive Activation Function

Moment Invariants	Sigmoid Function	Adaptive Function
	Test Sample	Test Sample
Zernike		
Sample 1	86.53	83.92
Sample 2	90.19	87.51
Sample 3	83.85	81.86
Sample 4	86.02	84.02
δ-geometric		
Sample 1	81.70	80.83
Sample 2	79.16	77.00
Sample 3	78.08	75.71
Sample 4	74.96	74.69
Geometric		
Sample 1	83.32	84.23
Sample 2	90.44	85.38
Sample 3	83.17	78.25
Sample 4	79.76	82.86

Table 2 shows the recognition rates of Zernike, δ-geometric and ordinary geometric features. Zernike features showed higher classification accuracies of around 87%, while δ-geometric and ordinary geometric features had approximate accuracies of 79% and 84% respectively, when applying sigmoid logistic activation function. Although adaptive activation function sped up network training significantly, the recognition rates are lower, approximately 84%, 77%, and 83% for Zernike, δ-geometric and ordinary geometric features respectively. In, both cases, the performance of Zernike features detection is the highest, followed by ordinary geometric moments and finally δ-geometric moments.

5.0 CONCLUSION

The experiments have shown that by using a dataset from Technion–Israel Institute of Technology, and applying adaptive activation functions between network layers, the convergence rates are much faster compared to applying the sigmoid logistic function. Encouraging results were obtained in which the iterations needed to converge to solution during network training is reduced by at least 50%. However, the significant reduction in network training does not show the same magnitude in recognition rate. In fact, there is a slight reduction in recognition rate. Depending on the type of applications, this tradeoff might be reasonable. More studies and experiments need to be conducted with other datasets to study the effectiveness of adaptive activation functions between layers.

Zernike moment invariants are superior to δ-geometric and ordinary geometric moment invariants in representing features of the isolated handwritten digits. The recognition rates for Zernike moment invariants are higher with accuracy of around 84% while for δ-geometric and ordinary geometric are 77% and 83% respectively, confirming the results obtained by Khotanzad and Lu (1990).

6.0 REFERENCES

Agui, T., Takahashi, H., & Nagahashi, H. (1991). Recognition of handwritten Katakana in a frame using moment invariants based on neural network. *IEEE International Joint Conference on Neural Networks*, *1*, 18-21 Nov (pp. 659 – 664). Singapore: IEEE Publications.

Altuwijri, M. M., & Bayoumi, M. A. (1994). Arabic text recognition using neural networks. *Proc. of IEEE International Symposium on Circuits and Systems. Systems*, *6*, 30 May-2 June (pp. 415-418). London UK: IEEE Publications.

Cao, J. (1995). *Handwritten numeral recognition with neural networks and fuzzy fusion*: Doctoral Dissertation, University of Windsor.

Chung, Y. Y., Wong, M. T., & Bennamoun, M. (1998). Handwritten character recognition by contour sequence moments and neural network. *Proc. of IEEE International Conference on System, Man and Cybernatics,* 5, 11-14 Oct. 1998 (pp. 4184 – 4188). San Diego, CA USA: IEEE Publications.

Dehghan M., & Faez, K.(1997). Farsi handwritten character recognition with moment invariants. *Proceeding of 13TH IEEE International Conference on Digital Signal Processing DSP 97, 2,* 2-4 July 1997 (pp. 507-510). Santorini Greece: IEEE Publications.

Hu, M. K. (1962). Visual pattern recognition by moment invariants. *IRE Transaction on Information Theory, 8(2),* 179 –187.

Khotanzad, A., & Lu, J. H. (1990). Classification of invariant image representations using a neural network. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 38*(6), 1028-1038.

Lam, L., & Suen, C. (1988). Structural classification and relaxation matching of totally unconstrained handwritten zip-code numbers. *Pattern Recognition, 21*(1), 9-31.

Master. T. (1993). *Practical neural networks recipes in C++*. San Diego CA: Academic Press, Inc.

Mitchel, B. T., & Gillies, A. M. (1989). A model-based computer vision system for recognizing handwritten ZIP codes. *Machine Vision and Applications, 2,* 231-243.

Moret, B. M. E. (1982). Decision trees and diagrams. *ACM Computer Surveys, 14(4)*, 593-623.

Mukundan, R., & Ramakrishnan, K. R. (1998). *Moment functions in image analysis: Theory and applications*. Singapore: World Scientific Publishing Co.Pte.Ltd.

Suen, C. Y., Nadal, C., Legault, R., Tuan, A. M., & Lam, L. (1992). Computer recognition of unconstrainted handwritten numerals. *Proceeding of the IEEE, 80*(7), 1162-1180.

Teague, M. R. (1980). Image analysis via the general theory of moments. *Journal of the Optical Society of America, 70*(8), 920-930.

Technion–Israel Institute of Technology. (03 January 2003). Machine Learning-EE046195. Retrieved April 03, 03, from http://www.ee.technion.ac.il/courses/046195.

Ye, X.Y. (2001). *A high accuracy offline handwritten numeral recognition system.* Master Thesis, University of Windsor.

Zakaria, M. F. (1987). Fast algorithm for the computation of moment invariants. *Pattern Recognition, 20*(6), 639-643.