Asian Journal of Information Technology 5 (9): 924-932, 2006

Recognition of Contour Invariants with Neurofuzzy Classifier

Siti Mariyam Shamsuddin, Azah Kamilah Muda and Tan Shuen Chuan
Faculty of Computer Science and Information System, Universiti Teknologi Malaysia

Abstract: In this study, we explore contour invariants for handwritten digits recognition with neuro-fuzzy classifier. We use fuzzy triangular function in backpropagation network to initialize the weights. The results reveal that fuzzy triangular membership function manages to decrease the network convergence rate with proper parameter setting. In this study, unthinned images are appropriate for training and classification purpose as it preserves the images' significant features. From our experiments, the results show that contour invariants exhibits highest rate of classification compares to geometric and Zernike invariants.

Key words: Contour invariants, geometric invariants, zernike invariants, neurofuzzy

INTRODUCTION

At this information technology age, image processing plays an important role in human daily life. From photocopying materials to printing photographs, all the affairs involve the process of an input image to obtain a favor desired output image. Sometimes, image-processing techniques even cooperate with artificial intelligence in order to achieve certain objectives. As we can see, applications that incorporated neural networks and image processing have been developed[1-3] after taking into consideration of its benefits and conveniences brought to human beings. Its implementation is feasible and necessary as it is robust and powerful[4,5].

However, the most significant enhancement of image processing using computer with the aid of artificial intelligence is shown with the implementation of first application-using neural network in recognition of handwritten digits[1]. This achievement and evolution leads to the mushrooming of substantial image processing applications that constituted of artificial intelligent element such as neural network and fuzzy logic[6-8]. Apparently, the fundamental and most highly demanded applications would be alphanumeric character recognition as it is widely used in miscellaneous applications[5,9].

According to previous studies held by numerous studyers and intellectuals, several neural networks approaches have been used in development of handwritten digits recognition applications[1,7]. Nevertheless, a hybrid method called neuro-fuzzy technique has been proposed though antecedent techniques manage to show satisfactory result of recognition rate[4,8,10].

We explore neuro-fuzzy technique in this study as this hybrid approach shows its potential and competency compare with other methods. However, the use of this study in constructing applications is fewer due to its opacity and newly found. Meanwhile, this technique is suggested as it manages to produce precise result through several characteristic tests[4]. The proposed method is analyzed and tested to validate and verify its accuracy and efficiency. Moreover, this could also identify the strengths and weaknesses of this study. As in feature extraction, invariants invariants which have a broad spectrum of applications in image analysis, such as invariant pattern recognition, object classification and pose estimation are utilized. The strengths and weaknesses of geometric invariants, Zernike invariants and contour sequence invariants for conventional computations are evaluated in this study.

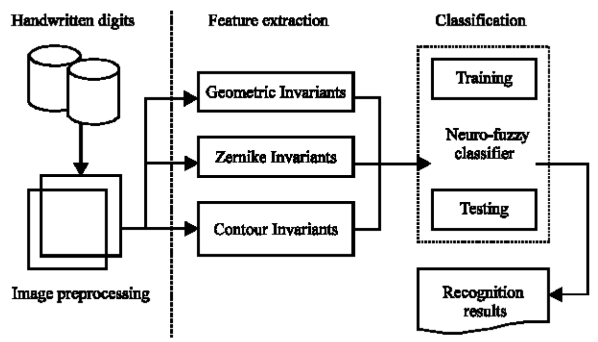

Fig. 1: A framework of neuro fuzzy classifier for handwritten digits recognition

Invariants functions have a broad spectrum of applications in image analysis, such as invariant pattern recognition, object classification and image reconstruction[12]. A set of invariants computed from a digital image, generally represents global characteristics of the image shape and provides a lot of information

Corresponding Author: Siti Mariyam Shamsuddin, Faculty of Computer Science and Information System, Universiti Teknologi Malaysia

about different types of geometrical features of the image. According to Syafirol [11], invariants functions exhibit different results when deal with original image and image that passed through preprocessing stage. Geometric invariants, Zernike invariants and contour sequence invariants react differently when coping with thinned and unthinned image. Thus, this study evaluates and verifies the hypothesis concluded by Syafirol[11]. Fig. 1 illustrates the framework of the neuro-fuzzy recognition system.

FEATURE EXTRACTION

Feature extraction is referred to a process of extracting significant features from a given input. It is a process in which an initial measurement pattern is transformed into a new set of numerical features[14]. These numerical features are obtained through certain computation and are used to classify the image digits. Selection of unique and typical features is a crucial step in the process since the subsequent stages (classification and recognition) only deal with these features. According to Khotanzad[15], good features are those satisfying two requirements:

- Small intraclass invariance-slightly different shapes with similar general characteristics should have numerically close values and,
- Larger interclass separation-features from different classes should be quite different numerically.

Besides, another criterion for choosing features is the computation time of features extraction[16]. A flexible recognition system must be able to identify an object regardless of its orientation, size and location in the field of view. In relation to that, invariants functions that possess rotation, scale and translation invariance properties are ideal as an excellent feature extraction 'tools' as it manages to fulfill these requirements.

Geometric invariants: Geometrical invariants are defined with the basis set $\{x^p, y^q\}$. The (p+q)th order two dimensional geometric invariants are denoted by m_{pq} and can be expressed as

$$\mu_{pq} = \iint_{\delta} x^p y^q f(x,y)dxdy \;;\; p,q = 0,1,2,... \quad (1)$$

where δ is the region of the pixel space in which the image intensity function f(x, y) is defined. By definition, the invariants of order zero (m_{00}) represents the total intensity of image. The first-order functions m_{10}, m_{01} provide the intensity invariants about the y-axis and x-axis of the image, respectively. The intensity centroid (x_0, y_0) is given by

$$x_0 = \left(\frac{m_{10}}{m_{00}}\right); y_0 = \left(\frac{m_{01}}{m_{00}}\right) \quad (2)$$

To make the invariants computation independent of the position of the image reference system, the origin of the reference system is shifted to the intensity centroid of the image. The invariants computed with respect to the intensity centroid are called central invariants and are defined as

$$\mu_{pq} = \iint_{\delta} (x-x_0)^p (y-y_0)^q f(x,y)dxdy \;;\; p,q = 0,1,2,... \quad (3)$$

From the definition of central invariants, we have

$$\mu_{00} = m_{00}; \mu_{10} = \mu_{01} = 0. \quad (4)$$

In special study Hu defines seven values, computed from central invariants through order three, that are invariant to object scale, position and orientation. In terms of the central invariants, the seven invariants invariants are given by

$$\begin{aligned}
&\varphi_1 = \eta_{20} + \eta_{02} \;\; \varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
&\varphi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
&\varphi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
&\varphi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \\
&\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + \\
&\quad = (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \\
&\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
&\varphi_6 = (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\
&+4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})
\end{aligned} \quad (5)$$

and one skew invariants is defined to distinguish mirror images and is given by

$$\begin{aligned}
\varphi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] + \\
&= (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]
\end{aligned}$$

This study only make uses of first four set of geometric invariants invariants.

Contour Sequence Invariants (CSI): Contour sequences obtained from shapes belonging to same class are similar.

Statistical invariants functions derived from the contour sequences can be used to classify the shapes[18,19]. As mentioned above, contour sequence representation is different for each shape and sequence obtained from the same class of shape is similar. The euclidean distance z(i), i = 1, 2, 3, …, N of the vector connecting the centroid and the ordered set of contour pixels forms a single, valued one dimensional functional representation of the contour[20].

Given an N-point contour sequential representation z(i), i = 1, 2, 3, …, N of a binary shape z(x,y), the rth invariants can be estimated as:

$$m_r = \frac{1}{N}\sum_{i=1}^{N}[z(i)]^r \tag{6}$$

and the rth central invariants can be estimated as:

$$M_r = \frac{1}{N}\sum_{i=1}^{N}[z(i) - m_1]^r \tag{7}$$

and the rth normalized central invariants can be defined as:

$$\bar{M} = \frac{M_r}{(M_2)^{\frac{r}{2}}} \tag{8}$$

Invariants of an arbitrarily large order can be derived from the contour sequence and used as features for shape classification. Due to their large dynamic range, higher order invariants are more sensitive to noise and the resulting classifier will be less tolerant to noise. Therefore, a few relatively low order invariants which are stable are selected. For example, a set of good shape features based on the four lower order invariants are given in reference[20]:

Normalized amplitude variation: $F_1 = \frac{(M_2)^{\frac{1}{2}}}{m_1}$,
Coefficient of skewness: $F_2 = \frac{M_3}{(M_2)^{\frac{3}{2}}}$,
Coefficient of kurtosis: $F_3 = \frac{M_4}{(M_2)^2}$,
For the 4th feature: $F_4 = \frac{M_5}{(M_2)^{\frac{5}{2}}}$. (9)

F_1 is a dispersion statistic that characterizes the variability of a density function and also can be regarded as measure of the amplitude variations in a contours sequence. F_1 is non-negative with a value equal to zero only if the contour is a prefect circle. is the transformation invariant statistic. F_2 and F_3 are shape measures which relate to the degree of symmetry and peakedness of the density function, respectively. Skew is positive when the majority of contour sequence samples value and kurtosis is positive if most of the samples are concentrated around the mean. The four components F_1, F_2, F_3 and F_4 will be input to the neuro-fuzzy classifier as feature vectors and only the contour pixels are used.

Zernike invariants: Zernike Invariants were first introduced by Teague based on orthogonal functions called Zernike polynomials. Zernike invariants have proved to be superior in terms of their feature representation capability and low noise sensitivity though are computationally very complex compared to geometric and Legendre invariants. The integral on the right-hand side of zernike polynomial can easily be expressed as a series of geometric invariants functions and this is given as:

$$\begin{aligned} Z_{20} &= \left(\frac{6}{\pi}\right)(m_{20} + m_{02}) - \left(\frac{3}{\pi}\right)m_{00} Z_{22} = \left(\frac{3}{\pi}\right) \\ &(m_{20} - m_{02} - 2im_{11}) Z_{31} = \left(\frac{12}{\pi}\right)(m_{30} + m_{12}) - \\ &\left(\frac{12}{\pi}\right)i(m_{03} + m_{21}) - \left(\frac{8}{\pi}\right)(m_{10} - im_{01}) \\ Z_{33} &= \left(\frac{4}{\pi}\right)(m_{30} - 3m_{12}) + \left(\frac{4}{\pi}\right)i(m_{03} - 3m_{21}) \end{aligned} \tag{10}$$

The above Eq. are valid only when the geometric invariants are evaluated for the normalized pixel coordinates transformed to the range (-1,1).

NEUROFUZZY CLASSIFICATION

The extracted features from invariants functions are used in training and classification using neuro-fuzzy classifier. Results of classification using these features are compared in terms of accuracy with and without applying thinning operation. Feed forward neural network with back propagation learning algorithm and sigmoid activation function are utilized in classification and recognition stage. The network weights are initialized using triangular membership functions. The steps of neuro-fuzzy classification are explained as below:

- Feed the neuro-fuzzy system with n input patterns. In our case, four geometric invariants feature, four Zernike invariants feature and four contour sequence invariants feature are utilized, respectively.
- Setup the neural network model: one input layer with n neurons, M hidden layer with N neurons and one output layer with P neuron. n is the number of input features used. In our case, we set the M to 2 and an

appropriate N is determined using try and error approach. P is set to 10 as our output is a combination of 10 features. Figure 2 illustrates our back-propagation neural network with one input layer, one hidden layer and one output layer, respectively.

where

$a_1 \dots a_n$ neuron at input layer,
$h_1 \dots h_l$ neuron at hidden layer,
$y_1 \dots y_k$ neuron at output layer,
$w_{11}^1 \dots w_{1n}^1$ weights between input and hidden layer,
$w_{11}^2 \dots w_{kl}^2$ weights between hidden and output layer,
$e_1 \dots e_k$ error for the output neuron.

- Set the learning rate η and invariant sum rate, α.
- Initialize the connection weights and node threshold (bias,) of hidden and output layer to small random values, range between (-0.5, 0.5) using triangular fuzzy membership function.

a) Generate a random values, x.
b) Pass x into the triangular membership function.

$$\mu(x) = \begin{cases} 0, & \text{if } x \le a - \frac{b}{2} \\ 1 - \frac{2|x_i - a|}{b}, & \text{if } a - \frac{b}{2} \le x < a + \frac{b}{2} \\ 0, & \text{if } x \ge a + \frac{b}{2} \end{cases}$$

c) Assign the generated value to initial network weights for each node.

- Set the maximum allowed network error, E_{max}.
- Activate the back-propagation neural network by applying inputs $x_1(p)$, $x_2(p), \dots, x_n(p)$ and desired outputs $y_{d,1}(p)$, $y_{d,2}(p), \dots, y_{d,n}(p)$.

a) Calculate the actual outputs of the neurons in the hidden layer:

$$y_j(p) = \text{sigmoid}\left[\sum_{i=1}^{n} x_i(p) \times w_{ij}(p) - \theta_j\right],$$

where n is the number of inputs of neuron j in the hidden layer and sigmoid is activation function (f_s) is used as and expressed as:

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

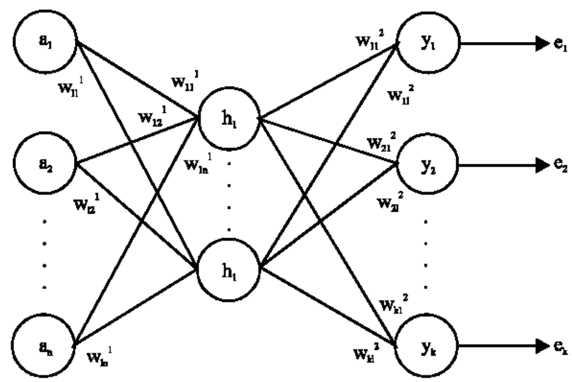

Fig. 2: Back-propagation architecture

Their first derivatives are calculated as:

$$f'_s(\text{net}) = o(1 - o)$$

b) Calculate the actual outputs of the neurons in the output layer:

$$y_k(p) = \text{sigmoid}\left[\sum_{j=1}^{m} x_{jk}(p) \times w_{jk}(p) - \theta_k\right],$$

where m is the number of inputs of neuron k in the output layer.

- Update the weights in the back-propagation network propagating backward the errors associated with output neurons.

a) Calculate the error gradient for the neurons in the output layer:

$$\delta_k(p) = y_k(p) \times [1 - y_k(p)] \times e_k(p)$$

where

$$e_k(p) = y_{d,k}(p) - y_k(p)$$

b) Calculate the weight corrections:

$$\Delta w_{jk}(p) = \alpha \times y_j(p) \times \delta_k(p)$$

Update the weights at the output neurons:

$$w_{jk}(p+1) = w_{jk}(p) + \Delta w_{jk}(p)$$

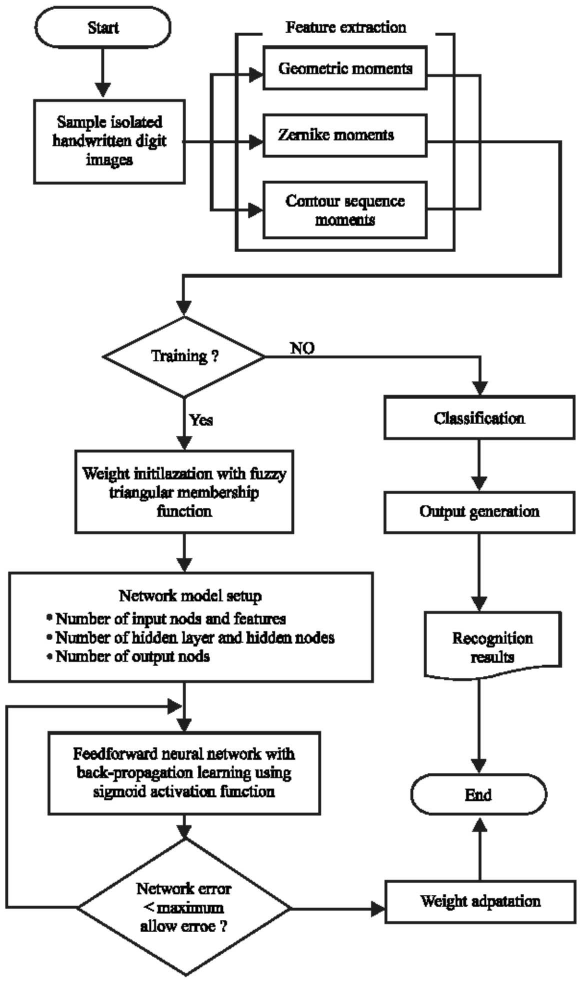

Fig. 3: A framework of neuro-fuzzy training and classification system

- Update the weights in the back-propagation network propagating backward the errors associated with hidden neurons.

a) Calculate the error gradient for the neurons in the hidden layer:

$$\delta_j(p) = y_j(p) \times [1 - y_j(p)] \times \sum_{k=1}^{l} \delta_k(p) \times w_{jk}(p)$$

b) Calculate the weight corrections:

$$\Delta w_{ij}(p) = \alpha \times x_i(p) \times \delta_j(p)$$

Update the weights at the hidden neurons:

$$w_{ij}(p+1) = w_{ij}(p) + \Delta w_{ij}(p)$$

- Compute the network squared error, Error at the output layer:

$$E = \frac{1}{2} \sum_p \sum_k \left(t_{kp} - o_{kp} \right)^2$$

where t_{kp} and o_{kp} are the target and actual outputs of neuron 'k' for pattern 'p'.

If Error > E_{max}, then repeat step 6-9. Else, terminate the network training. In our case, we set the number of maximum epoch for network training. Therefore, our training will terminate when the Error < E_{max} or the network fails to converge within a specific epoch size. A framework of the neuro-fuzzy training and classification system is illustrated in Fig. 3.

EXPERIMENTAL RESULTS

Data collection: The isolated handwritten images utilized in this study are taken from Technion-Israel Institute of Technology[26] at http://www.ee. technionac. il/courses/046195. This dataset is chosen because of it diversity, noise free and already in isolated form. 50 image files from each digit class are utilized in training stage while 200 image files are used for testing and classification purpose. The separated handwritten image digits of 0 to 9 are in grayscale format and in uniform size of 28x28 pixels. These images are converted and saved into binary raw format using image thresholding technique. Figure 4 illustrate sample of isolated handwritten digits used in network training.

The input data for the model are numerical values extracted from isolated digit images using geometric invariants, Zernike invariants and contour sequence invariants. Five hundreds samples of isolated handwritten digit images are used in network training. Two samples digit with each sample contains 50 set digits image are applied in testing and classification phases. Four set of invariants values computed from Geometric invariants, Zernike invariants and Contour Sequence invariants are used, respectively as the input to the neuro-fuzzy networks constructed. The number of output nodes is depending to the number of patterns to be classified. Thus, 10 nodes are used where all nodes are set to 0 except for the node that is marked with 1 which correspond to the class of the input. Number 0 for example, the output node is 1 for the first node and 0 for the remaining nodes.

Intraclass invarianceness: According to[24], good features are those features with small intraclass invariance and

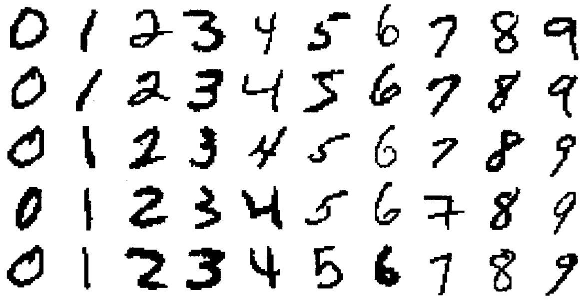

Fig. 4: Sample of isolated handwritten digits for network training

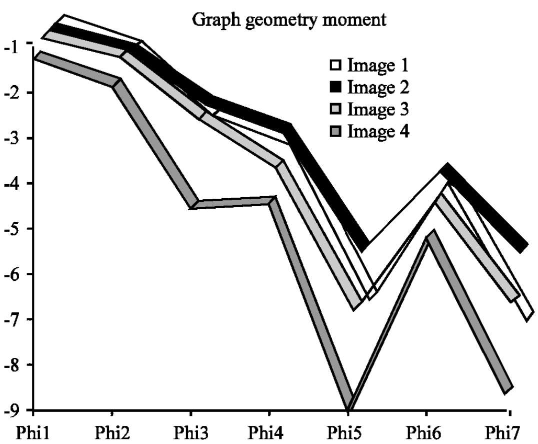

Fig. 5: Features of digit 0 extracted using geometric invariants

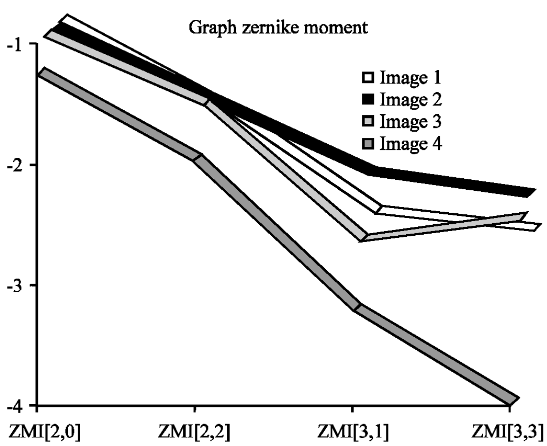

Fig. 6: Features of digit 0 extracted using Zernike invariants

larger interclass separation where features from difference classes should exhibit dissimilarities numerically. From Fig. 5-7, it can be seen that features extracted from contour sequence invariants show better intraclass representation with close similarity compared with

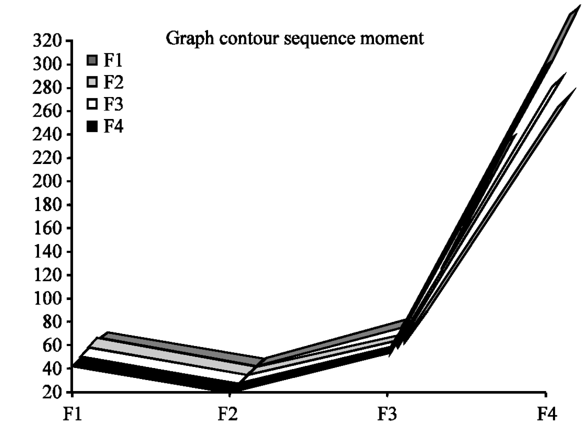

Fig. 7: Features of digit 0 extracted using contour sequence invariants

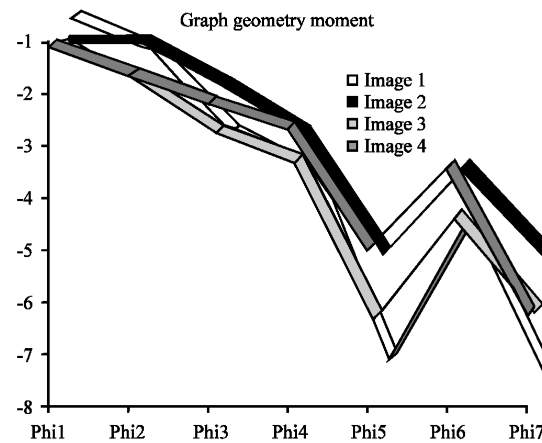

Fig. 8: Interclass invariance for features of digit 0 to 4 extracted using geometric invariants

geometric invariants and Zernike invariants. As illustrated in Fig. 5-7, both geometric invariants and Zernike invariants have possess dissimilarity features values. The purpose of classifications is to differentiate between classes; therefore, contour sequence invariants in this case better in representing isolated handwritten digits from same class.

Interclass invarianceness: Features extracted from different classes supposed to show variation in order to be used in classification and recognition purpose. The features used should be typical and unique in symbolizing a particular class of digits. In relation to that, a small deviation between features should be considered for a better differentiation rate. Fig. 8-10 illustrated the interclass invariance for features of digit 0 to 9 using geometric invariants, Zernike invariants and contour sequence invariants. From these Fig, it can be

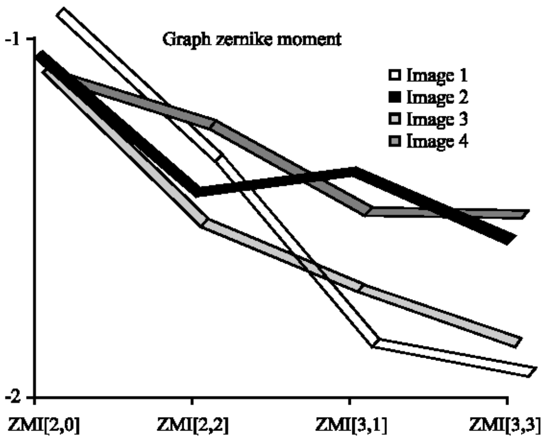

Fig. 9: Interclass invariance for features of digit 0 to 4 extracted using Zernike invariants

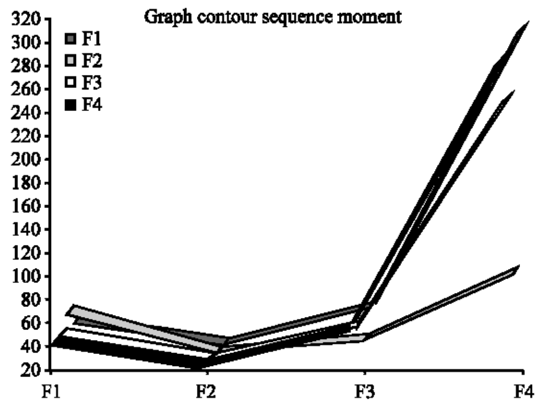

Fig. 10: Interclass invariance for features of digit 0 to 4 extracted using contour sequence invariants

Table 1: Weight initialization with triangular membership function

Invariants functions	Triangular membership (iteration)	Random number (iteration)
Geometric	14569	14636
Zernike	11129	8985
Contour sequence	4666	6514

Table 2: Recognition rates of unthinned isolated digits (0-9) using geometric invariants, Zernike invariants and contour sequence invariants

Invariants functions	Recognition accuracy (%)		
	Group 1	Group 2	Average
Geometric	36	40	38
Zernike	38	43	40.5
Contour sequence	51	60	55.5

observed that Zernike invariants performed better in representing the digit images as it exhibited greater divergences compared with geometric invariants and contour sequence invariants.

Network weights initialization: Network weights are initialized using Triangular Membership function to verify its feasibility in enhancing the network convergence rate. Table 1 shows that network weights can be initialized with appropriate parameters using Triangular Membership Function and in this case decrease the iteration number required to converge to the solution. The classification and recognition results of unthinned isolated handwritten digits are summarized in Table 2.

Unthinned digit images are used in training stages while thinned and unthinned digit images are tested in classification to validate its effectiveness. The classification and recognition results of thinned and unthinned digit images (50 images for each group) are summarized in tables below.

DISCUSSION OF RESULTS

Experimentation results shown that Zernike invariants invariants are superior to geometric invariants and contour sequence invariants in representing features of isolated handwritten digits with obvious interclass invariants. In terms of intraclass invariants, contour sequence invariants exhibited better result with small deviation. Our network suffers from low recognition rates may due to the network is trained inappropriately resulting with high network error. However, from Table 3-5, it can be observe that contour sequence invariants possess higher recognition rates even though Zernike invariants shown higher interclass invariants. From these 3 Tables, we also can conclude that thinning operation should be excluded as it brings down the recognition rate of isolated handwritten digits.

Triangular membership function is applied to generate initial weights for the network. In this study, network training convergence rate is improved compared to random method. But in most of the case, the parameter used should be appropriately set. The recognition accuracy should be higher and the network convergence will rise in theory but in our case, it deteriorates the recognition rate with introduction of ambiguity. Thus, in our case normalization operation can be excluded.

Our neuro-fuzzy network suffered from low recognition rate may be due to the following reasons:

- The network architecture (2 hidden layers, with 150 neurons in the first layer and 75 neurons in the second layer) may be improperly set up.
- The maximum allow network error is high, around 0.04.
- 500 set training data is inadequate for network training.

Table 3: Recognition rates of thinned and unthinned digit 0 to 9 using geometric invariants

Geometric invariants	Recognition accuracy		
	Group 1	Group 2	Average (%)
Thinned image	32	36	34
Unthinned image	36	40	38

Table 4: Recognition rates of thinned and unthinned digit 0 to 9 using Zernike invariants

Zernike invariants	Recognition accuracy		
	Group 1	Group 2	Average(%)
Thinned image	21	20	20.5
Unthinned image	38	43	40.5

Table 5: Recognition rates of thinned and unthinned digit 0 to 9 using CSI

Contour sequence invariants	Recognition accuracy		
	Group 1	Group 2	Average (%)
Thinned image	40	49	44.5
Unthinned image	51	60	55.5

- Standard back-propagation instinctly suffered from slow convergence, thus the number of epoch used (20000) should be increased.

CONCLUSION

This study presents a comparison of invarianceness effectiveness among contour sequence invariants, geometric invariants and Zernike invariants in representing the description of an image. Network training and testing (classification) is conducted using standard back-propagation with network weights are initialized using triangular membership function. Based on the experimentations performed, it can be concluded that:

- Operation thinning conducted to a digit image decreases the classification and recognition accuracy rate and thus can be neglected.
- Fuzzy Triangular membership function reduces neural network training duration with proper parameters setting.
- Contour sequence invariants are better in representing an image description compared to geometric invariants and Zernike invariants.

REFERENCES

1. Burel, G., I. Pottier and J.Y. Catros, 1992. Recognition of Handwritten Digits by Image processing and neural network. J. Pattern Recognition, pp: III666-III671.
2. Knerr, S., L. Personnaz and G. Dreyfus, 1992. Senior Member, IEEE. Handwritten digit recognition by neural networks with single-layer training. IEEE Transactions on Neural Network, 3,pp: 962-968.
3. Starzyk, J.A. and Nasser Ansari, 1992. Feedforward neural network for handwritten character recognition. J. Pattern Recognition, pp: 2884-2887.
4. Fauzi Bouslama, 1996. Neuro-fuzzy Techniques in the recognition of handwritten Arabic characters. IEEE Transactions on Pattern Recognition, pp: 142-146.
5. Negnevitsky, M., 2002. Artificial Intelligent Systems. 1st Edn. Edinburgh Gate, England: Pearson Education Limited.
6. Lu, W., B.X. Shi and Z.J. Lee, 2001. Handwritten digits recognition with Neural Networks and Fuzzy Logic. Institute of Microelectronics, Tsinghua University, Beijing, P.R.China.
7. Lu, W., B.X. Shi and Z.J. Lee, 1996. A hybrid handwritten digits recognition system based on neural networks and fuzzy logic. Institute of Microelectronics, Tsinghua University, Beijing, P.R.China, pp: 424-427.
8. Manish Sarkar, B. Yegnanarayana and Deepak Khemani, 1998. Backpropagation learning algorithms for classification with fuzzy mean square error. Pattern Recognition Letters, pp: 43-51.
9. Zhao, B., H. Su and S.W. Xia, 1997. A new method for segmenting unconstrained handwritten numeral string. J. Image Processing, pp: 524-527.
10. Arafeh, L.M., 1996. Neuro Fuzzy System. IEEE Transactions on machine vision, pp: 430-435.
11. Syafirol Hafiz Arshad, 2003. Kajian perbandingan teknik pengekstrakan fitur berasaskan momen. Master Thesis. University Technology Malaysia.
12. Mukundan, R. and K.R. Ramakrishnan, 1998. Invariants functions in image analysis – Theory and applications. Singapore: World Scientific Publishing Co.Pte.Ltd.
13. Lau, Hui Keng, 2003. Neuro-fuzzy for handwritten digits recognition. Master thesis. University Technology Malaysia.
14. Pejnoviæ, P., L.J. Buturoviæ and Z. Stojiljkoviæ, 1992. A note on pattern recognition and object classification. IEEE International Conference on Pattern Recognition Methodology and System, 2,pp: 434-437.
15. Khotanzad, A. and J.H. Lu, 1990. Classification of invariant image representations using a Neural Network. IEEE transactions on acoustics, Speech and signal processing, 38,pp: 1028-1038.
16. Ye, X.Y., 2001. A high accuracy offline handwritten numeral recognition system. Master thesis. University of Windsor.
17. McCulloch, W.S. and W. Pitts, 1943. A logical calculus of the ideas immanent in nervous activity. Bulletin of Mathematical Biophysics, 5: 115-117.

18. Chung, Y.Y. and M.T. Wong, 1998. High accuracy handwritten character recognition system using contour sequence invariants. Proceedings of ICSP, pp: 1249-1252.
19. Chung, Y.Y., M.T. Wong and Mohammed Bennamoun, 1998. Handwritten character recognition by contour sequence invariants and neural network. Proceedings of ICSP, pp: 4184-4188.
20. Gupta, L. and M.D. Srinath, 1987. Contour sequence invariants for the classification of close planar shapes. Pattern Recognition, 20: 267-272.
21. Joarder Kamruzzaman and S.M. Aziz, 2002. A note on activation function in multilayer feedforward learning. IEEE Transaction on machine vision.
22. Ajith Abraham, 2002. Beyond Integrated Neuro-fuzzy Systems : Reviews, prospects, perspectives and directions. School of computing and Information Technology, Monash University, Victoria, Australia.
23. Rokiah Ahmad, Maslina Darus, Siti Mariyam Shamsuddin and Azuraliza Abu Bakar, 2004. Pendiskretan set kasar menggunakan ta'akulan boolean terhadap pencaman simbol matematik. J. Inform. Tech. Multimedia, UKM, 1: 15-26.
24. Khotanzad, A. and J.H. Lu, 1990. Classification of inivariant image representation usign a neural network. IEEE Transanctions on Acoustics, Speech and Signal Processing, 8: 9-31.
25. Mukundan, R. and K.R. Ramakrishnan, 1998. Invariants Functions in Iamge Analysis: Therory and Applications. Singapore: World scientific publishing Co. Pte. Ltd.
26. Techion-Israel Institute of Technology, 2003. Machine Learning-EE046195. Retrieved 04 03,03 from http://www.ee.technion.ac.il/courses/046195.