# GENETIC ALGORITHMS IN OPTIMIZATION OF 3-D FACE RECOGNITION SYSTEM USING CYLINDRICAL-HIDDEN LAYER NEURAL NETWORK IN ITS EIGENSPACE DOMAIN

Benyamin Kusumoputro, Martha Yuliana Pangabean and Leila Fatmasari Rachman

Computational Intelligence Research Laboratory, Faculty of Computer Science
University of Indonesia, Depok Campus, INDONESIA
e-mail: kusumo@cs.ui.ac.id

## ABSTRACT

In this paper, a 3-D face recognition system is developed using a cylindrical structure of hidden layer neural network and its optimization through genetic algorithms. The cylindrical structure of hidden layer is constructed by substituting each of neuron in its hidden layer of conventional multilayer perceptron with a circular-structure of neurons. The neural system is then applied to recognize a real 3-D face image from a database that consists of 5 Indonesian persons. The images are taken under four different expressions such as neutral, smile, laugh and free expression. The 2-D images is taken from the human model by gradually changing visual points, which is done by successively varies the camera position from *– 90 to +90* with an interval of *15* degree. The experimental result has shown that the average recognition rate of about *64%* could be achieved when we used the image in its spatial domain and about *84%* when the image data is transformed to its eigen domain. Optimization of the hidden neurons is accomplished using genetic algorithms, which reduced the active neurons up to about *63.7%* while increasing the recognition rate into about *94%* in average.

Keywords: Face recognition system, cylindrical hidden layer neural networks, genetic algorithms, optimization of neural system

## 1. INTRODUCTION

Automated face recognition (AFR) system has attracted much interest in the last few years, motivated by the growth of its applications in many areas. Face identification is very important problem in law enforcement and forensics, authentication of access into building or automatic transaction machine, and searching for faces in video databases, intelligent user interfaces and others. Generally, AFR consists of face detection, which determine the position and size of human face in an image, and a face recognition that compare an input face against models of faces, which are stored in a database of known faces, and face verification for the purpose of authentication. In our study, we assume that the face location in the image is already known. Automatic face recognition system has some drawbacks, however, partly because of different faces may appear very similar which requires high discriminating power of the system. Another problems, for instance, same image of faces may appear differently due to imaging constraints, such as changes of illumination point, facial expressions, the use of glasses and other accessories. When the face undergoes scaling or moderate rotation, a large amount of images should be occluded into the system. It is therefore; in many implementations of face recognition system images are taken with minimal occlusions of facial accessories within a constrained environment with controlled illumination.

In recent years, progress has been made on the problem of face recognition, especially in head-on face images with controlled illumination and scale. Good results has been obtained for 2-D frontal images by the use of template matching of large database[1], combined feature and template matching[2], template matching using Karhunen-Loeve transformation of large set of face images[3][4]. Many researchers are now trying to extent this high recognition capability of the system, to recognize more general view positions of images that cover the entire 3-D viewing sphere. It is argued that 3-D recognition can be accomplished using linear combinations of as few as four or five 2-D

Accepted paper [16 October 2001]. Last Revision [25 November 2001]

viewpoint images[5][6][6], however, good results still could not be expected from the system.

The development of 3-D face recognition system has been currently done, especially due to its importance in multi-media systems; however, it is not yet realized a 3-D face recognition system with high performance of recognition. Many difficulties are occurred, for example, even in a simple 3-D face-image, quite large number of two-dimensional observed images at numerous viewpoints should be calculated. As a consequence, when a 3-D face-image should be memorized for recognition, a large memory size may be required.

Neural network architectural models, recently, have been considered and developed to deal with several important problems in pattern recognition, pattern classification, function approximation and regression. The MLP neural system has shown many marvelous experimental results through its supervised learning paradigm, however, the back propagation neural system has some drawbacks, such as converging to local minima, temporal instability and its low speed of convergence due to full connection of its large size architectural network.

Authors have proposed a 3-D face recognition system using neural network with a cylindrical structure of hidden layer, which is constructed by piling many of circular-structure of neurons. This modified neural network is constructed by substituting each neuron in conventional hidden layer of multilayer perceptron with a circular-structure of neurons. This neural system is then called as cylindrical-structure of hidden layer neural network (CHL-NN). The neural system is then applied on a real 3-D face image database that consists of 5 Indonesian persons. The images are taken under four different expressions such as neutral, smile, laugh and free expression. The 2-D image is taken from the 3-D face image by gradually changing visual points, which is done by successively varies the camera position from $-90$ degree to $+90$ degree with an interval of $15$ degree.

This paper is organized as follows: section 2 will describe the architecture of the developed neural system, including with its learning algorithms. In section 3, we will discuss the transformation of the image from spatial domain into its eigen domain, either by Karhunen-Loeve transformation technique or by using Fisherface method. In section 4, we will describe the use of genetic algorithms for optimizing the architectural structure of the neural system, while the experimental results are discussed in section 5. This paper will be closed with a summation in section 6.

## 2. CYLINDRICAL STRUCTURE OF HIDDEN LAYER NEURAL NETWORKS

The cylindrical-hidden layer neural system consists of two-layer network, with the first layer has the same number of neurons as with that of the input image pixels. The hidden layer is called cylindrical-structure of hidden layer that consists of piles of circular-structure of neurons, which is accomplished by substituting each neuron of the conventional hidden layer with a circular-structure of neurons that represent any various visual positions of the input image. The output layer of this MLP is the same with that of the conventional neural network. It consists of neurons that correspond to the number of objects to be recognized. The input image of the system is two-dimensional image taken horizontally at various frontal positions, according to various angles from $0^o$ to $360^o$. In this experiment, we considered only with horizontal visual positions of the 3-D object, for simplicity, omitting the possibilities of the tilting position of the input image.

Number of neurons in the first layer are $1024$ neurons, for $32 \times 32$ pixel of 2-D image at each horizontally visual positions. As the consequence of the hidden layer structure modification, there are two additional factors that should be calculated in order to determine which neuron in the circular-hidden neurons keep the additional information in their weight connections. The first additional factor is calculated for modifying the ordinary MLP equation of the weight connections between input neuron and the neurons in the cylindrical-structure of hidden layer. The second additional factor is calculated for modifying the weight modification between cylindrical-structure of hidden neurons and the output neurons. The two additional factors for the cylindrical-structure of hidden layer with three rings-of-neurons at each of the circular-structure of hidden-neurons can be written as follows:

$$fa_{lh} = \begin{cases} (v(k) \bullet s_{lk})\, r_{lh} & ((v(k) \bullet s_{lk})\, r_{lh} \geq 0) \\ 0 & ((v(k) \bullet s_{lk})\, r_{lh} < 0) \end{cases} \quad (1)$$

$$fb_{lh} = \begin{cases} 1 & ((v(k) \bullet s_{lk})\, r_{lh} \geq 0) \\ 0 & ((v(k) \bullet s_{lk})\, r_{lh} < 0) \end{cases} \quad (2)$$

where $v(k) \bullet s_{lk} = ||v(k)||\ ||s_h||\ \cos\theta$, with $v(k)$ denotes the vector-position of the 2-D image from the object position at the center, with $k$ means the image number. Vector $s_{th}$ denotes the vector-position of the neuron with its index $h$ determined from the center of the circular-structure of the hidden neurons. Index $t$ shows the position of the vertically circular-structure of hidden neurons in the cylindrical-structure of hidden layer. Index $r_{th}$ denotes the distance coefficient between the center of the circular-structure of the hidden layer with the $l$ ring-of-neurons ($l = 1, 2, 3$) at each of the circular-structure of hidden neurons. Index $r_{th}$ could be calculated through:

$$r_{ts} = \frac{1}{\left(\| d(k) \| - \| v_{ts} \| + 1\right)^4} \qquad (3)$$

In our experiments, those parameters are defined as $v(k)=1$, $s_{1h}=1$, $s_{2h}=0.66$ and $s_{3h}=0.33$, for three rings-of-neurons at each of the circular-structure of hidden neurons, respectively.

The *Eq.1* implies two possibilities of conditions. If the angle $\theta$ between vectors $v(k)$ and $s_{kt}$, has a value of $-\pi/2 < \theta < \pi/2$, the additional factor has a positive value, which means that there is an additional information that should be calculated to modify the hidden neuron activation. This additional information will be kept by the neuron and represented in their connection weight after its modification. However, when the angle between those two vectors has a value of $\vartheta \geq \pi/2$ or $\theta \leq -\pi/2$, there is no additional information at the predetermined visual vector-position that should be calculated by the predetermined neuron in the circular-structure of the hidden layer. The last condition means that no connection weight modification is required both in the training process and also in its recognition.

The neuron activation of the neurons in the circular-structure of the hidden layer can be calculated through:

$$Z_{lht}(k) = S\left( fa_{th} \left( \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} u_{ijht}\, x_{ij}(k) + b_{lht} \right) \right) \qquad (4)$$

where $S(x)$ the non-linear binary Sigmoid function, $h = 1,2..,M_1$ and $t = 1,2,\ldots, M_2$, $x_{ij}$ denotes the input signal with $i = 1,2,\ldots, N_1$ and $j = 1,2,\ldots, N_2$ of the $N_1 \times N_2$ input matrix of 2-D image. As in the conventional MLP with back propagation algorithm, the activation of each hidden neuron is fed to the output neuron, and the output activation is then calculated as:

$$O_q(k) = S\left( fb_{lh} \left( \sum_{h=1}^{M_1} \sum_{t=1}^{M_2} \sum_{l=1}^{M_3} w_{qlht}\, z_{lht}(k) + b_q \right) \right) \qquad (5)$$

The connection weights and bias between hidden neuron and its output neuron are then modified through:

$$\Delta w_{qlht}(n+1) = \alpha\, \delta_q\, Z_{lht} . fb_{lh} + \eta .\, \Delta w_{qlht}(n)$$

$$\Delta b_q\,(n+1) = \alpha\, \delta_q + \eta .\, \Delta b_q(n) \qquad (6)$$

where $\delta_q = (t_q - o_q)$ denotes the propagated error-signal of the output neurons. The connection weights and bias between the input and the hidden neurons is then modified through:

$$\Delta u_{ijlht}\,(n+1) = \alpha\, .\delta_{lht} .\, x_{ij} .\, fa_{lh} + \eta .\, \Delta u_{ijlht}(n)$$

$$\Delta b_{lht}\,(n+1) = \alpha ..\delta_{lht} + \eta \Delta b_{lht}(n)$$

$$\delta_{lht} = \sum_q \delta_q \left( \sum_{h=1}^{M_1} \sum_{t=1}^{M_2} \sum_{l=1}^{M_3} W_{qlht}\, fb_{lh}\, Z_{lht}(1 - z_{lht}) \right) \qquad (7)$$

## 3. TRANSFORMATION OF IMAGES INTO ITS EIGEN DOMAIN

Principal component analysis (PCA) is now commonly used as a dimensionality reduction method in computer vision, particularly in face recognition. PCA techniques, which is also called as Karhunen Loeve Transformation (KLT) choose a dimensionality reducing linear projection that maximizes the scatter of all projected samples. Let a set of $N$ sample images of $x_i$, i = 1...N, and assume that each image belongs to one of $c$ classes $\{X_1, X_2, \ldots, X_C\}$. Consider also, there is a linear transformation that mapping the original $n$-dimensional image space into $m$-dimensional feature space, where $m<n$. The new feature vectors $y_i$ are defined by the following linear transformation:

$$y_i = \Phi_i W^T \qquad (8)$$

Total scatter matrix $St$ is then defined as

$$S_T = A^T A \qquad (9)$$

where $A = [\Phi_1; \Phi_2; \ldots; \Phi_N]$, with

$$\Phi_i = x_i - \mu$$

$$\mu = \frac{1}{N} \sum_{i=1}^{N} x_i \qquad (10)$$

K-L transformation of the image vector yields a scatter of the transformed feature vectors

$$WS_TW^T \tag{11}$$

Transformation matrix that is chosen to maximize the total scatter matrix is written as

$$W = \arg\max_W |WS_TW^T| = [w_1; w_2; ...; w_m] \tag{12}$$

where $w_i$, i = 1...m, is the set of eigenvector of $S_T$ that is corresponding to the $m$ largest eigenvalues. These eigenvectors are also called principal components that have the same dimensions with that of the original images and referred as Eigenfaces. The value of $m$ could be derived from the equation

$$m = \min_r \left\{ \frac{\sum_{i=1}^{r} d_i}{\sum_{i=1}^{N} d_i} > \theta \right\} \tag{13}$$

where $\theta$ the threshold value with $0< \theta \leq 1$.

The Fisherface method is derived based on Fisher's Linear Discriminant (FLD)[8], which tries to find the transformation matrix W (see *Eq. 12*) in such a way that the ratio between-class scatter and the within-class scatter is maximized. This expression could be written as

$$W_{opt} = \arg\max_W \frac{|WS_BW^T|}{|WS_WW^T|} \tag{14}$$

where $w_i$, $i = 1...m$, the set of eigenvectors of $S_B$ (between-class scatter matrix) and $S_W$ (within-class scatter matrix) that correspond to the $m$ largest eigenvalues λ through:

$$S_Bw_i^T = d_iS_Ww_i^T \tag{15}$$

where $i = 1...m$ and $d_1>d_2> ... >d_m$. The $S_B$ and $S_W$ are defined as

$$S_B = \sum_{i=1}^{C} N_i(\mu_i - \mu)^T(\mu_i - \mu) \tag{16}$$

$$S_W = \sum_{i=1}^{C} \sum_{j=1, x_j \in X_i}^{N_i} (x_j - \mu_i)^T (x_j - \mu_i) \tag{17}$$

Since there is a possibility that $S_W$ is singular, PCA is then used firsly, to reduce the dimension of the feature space to $N$-$c$, and then applying the standard FLD in *Eq.14* to reduce the dimension to $c$-$1$.

## 4. GA IN OPTIMIZATION OF CYLINDRICAL STRUCTURE OF HIDDEN LAYER

The genetic algorithm (GA) is a stochastic searching algorithms that inspired by the proses of natural evolution. GAs, originally developed by Holland[8], and its mathematical framework is developed and presented in Holland's pioneering book[9], which is intensively observed and implemented by Goldberg[10]. The basic element processed by a GA is the string formed by concenating substrings, each of which is a binary coding of a parameter of the search space. Thus, each spring represents a point in the search space and hence a possible solution to the problem. Each string is decoded by an evaluator to obtain its objective function value. This value, which should be minimized by the GA is converted to a fitness value which determines the probability of the individual undergoing genetic operators. The population then evolves from generation to generation through the application of the genetic operators. The total number of strings included in a population is kept unchenged trough generations. A simple genetic algorithm that yields good results in many practical problems is composed of these operators: reproduction, crossover and mutation.

The optimization procedure of the cylindrical-hidden layer NN through GA is initially started by making a network with a rather big and complex structure. In their optimization process, GA will search the most optimal subset of the initial basic structure. Each subset structure will become an individual in the population to be processed, which is represented by an individual string. As the knowledge representation formed in CHL-NN is kept in its neuron activation, the GA optimization is directed in its number of neurons. Preliminary experimental result also showed that optimization of the networks hidden neurons works more efficient and accurate compare with that of networks connection weights. The optimization procedure of GA is implemented by initially encoding the problem and defining the objective function.

The process of the problem encoding can be summarized as follows. As the problem parameter that should be optimized in the network structure lies in its number of neurons, all these connections are encoded to a chromosome chain. Each chromosome forms a binary string *<100....1001>* that represents the network structure, and the chromosome length equivalents to the number of neurons in the network.

The objective function of the system to be optimized is done through its fitness value. To calculate the fitness value, each individual chromosome is decoded back to a CHL-NN structure and be trained using backpropagation learning algorithm. Weight initialization of each structure is performed using Nguyen-Widrow method, and the network is trained until small error tolerance is accomplished or maximum epochs is reached. After the learning process is completed, the fitness value is calculated by:

*number_of_non_activated_connections / (error_rate * number_of_epochs)*[11].

By using fitness value evaluated by the objective function, GA searches an individual best network structure with large number of non-activated neurons, small error rate, and small number of epochs, by conducting evolution process through the operation of reproduction, crossover, and mutation. The whole process could be written as:

Step 0. Problem parameter-encoding. The parameter of the problem is the number of neuron in the hidden layer of the network. Defining objective function for optimization procedure (fitness function), that is *number_of_non_activated_connections/(error_rate *number_of_epochs).*

Step 1. Generate initial population randomly.

Step 2. Calculate fitness value of each individual string by decoding each string to the CHL-NN and run the objective function that will train it.

Step 3. Reproduction, crossover and mutation

Step 4. New generation formed.

Step 5. Back to step 2 until number of maximum generation is reached.

Step 6. At the last generation, there will be an individual string with the highest fitness value as the solution, and the best string that will form an optimal winner network.

## 5. EXPERIMENTAL PROCEDURE AND ITS RESULT

The experimental procedure is conducted using a 3-D face database that consists of 5 Indonesian persons. The images are taken under four different expressions. such as, neutral, smile, laugh and free expressions. The 2-D image is given from 3-D face image by gradually changing visual points, which is successively varied from $-90^0$ to $+90^0$ with an interval of $15^0$. Figure 1 shows some example of images that used in this recognition system, including with different type of expressions.

Experiments are designed, firstly, to know the ability of the increasing the percentage of training/testing data to the recognition rate; and second, to know the ability of the system to recognize faces with an unknown expression, which is not included in the training stage. For each of experiments, the recognition rate of the system is calculated based on the averaging of 10 times recognition of each face. The data set of the *Experiments#1* is depicted in Table 1, while for *Experiments#2* is depicted in Table 2, respectively. For all of the data used in the experiments, the face-image for testing is always in the different viewpoint compare with that of its training stage. Increasing the number of training viewpoints should also follow by increasing the number of hidden neuron in each circular hidden neuron respectively.

### 5.1 Experimental Result without GA Optimization Procedure

In these experiments, we firstly converted the gray-level value of the images on its spatial domain into values on its eigen domain. The Karhunen-Loeve transformation technique as written in Section 3 is utilized for transforming the data set in the spatial domain into its eigen domain. This transformation is then followed by Fisherfaces technique, and the result is directly inputted to the 3-D recognition system. Using the data set as in the ***Experiments#1*** and ***Experiments#2***, respectively, results of these experiments are depicted in Table 3 and Table 4. It can be seen that increasing the percentage of training face-image in different viewpoints will increase the recognition rate of the system. By using training/testing percentage of *60/40*, the recognition rate of the system is about *84%* at most. Results of the ***Experiments#2*** are presented in Table 4, that showing the same tendencies of higher recognition rate when using higher percentage of training/testing paradigm. The highest recognition rate could be achieved when using training/testing percentage of *58%*, with its recognition rate of *87%* at most. Those results show that the recognition rates of the system to the *Experiments#1* and *Experiments#2* could be considered as high enough. As for comparison, by using 3-D face images on its spatial domain, which its value is directly fetched to the neural system, the recognition rate could only reach of about *64%* and *74%* at most, for *Experiment#1* and *Experiment#2*, respectively[11]. However, to increase further the recognition rate of the system, we then propose the use of genetic algorithm to optimize the architectural structure of the circular-hidden layer.

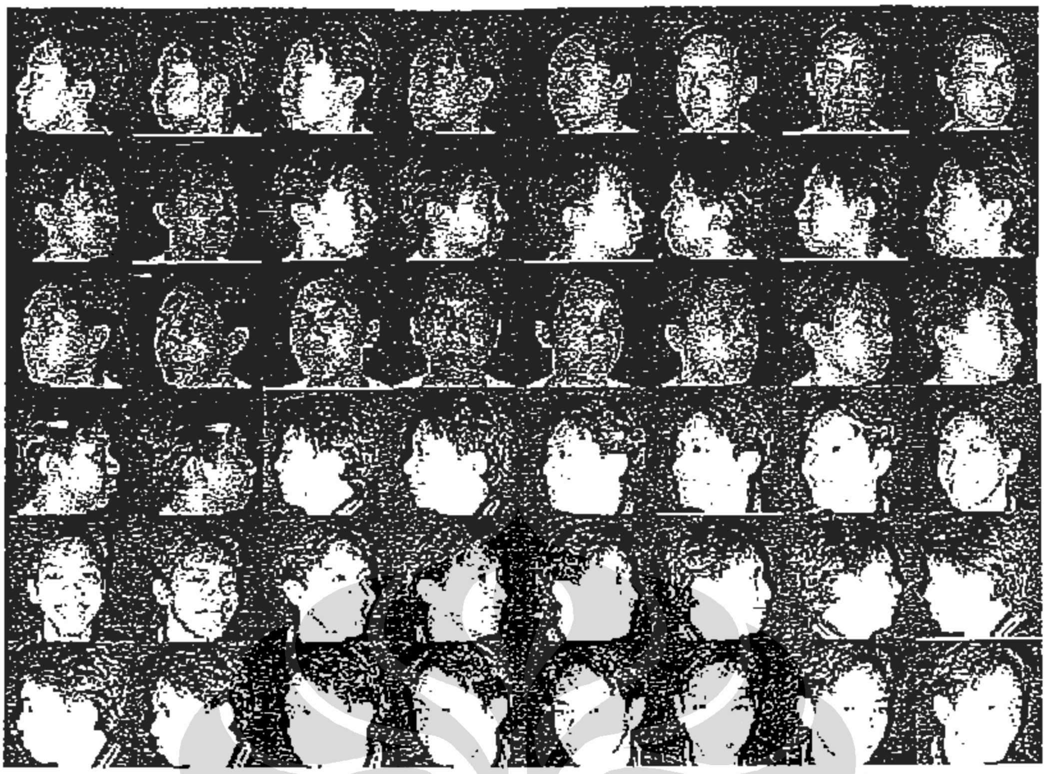

Figure 1. Example of images with different viewpoints and expressions.

Experiments are designed, firstly, to know the ability of the increasing the percentage of training/testing data to the recognition rate; and second, to know the ability of the system to recognize faces with an unknown expression, which is not included in the training stage. For each of experiments, the recognition rate of the system is calculated based on the averaging of 10 times recognition of each face. The data set of the *Experiments#1* is depicted in Table 1, while for *Experiments#2* is depicted in Table 2, respectively. For all of the data used in the experiments, the face-image for testing is always in the different viewpoint compare with that of its training stage. Increasing the number of training viewpoints should also follow by increasing the number of hidden neuron in each circular hidden neuron respectively.

## 5.1 Experimental Result without GA Optimization Procedure

In these experiments, we firstly converted the gray-level value of the images on its spatial domain into values on its eigen domain. The Karhunen-Loeve transformation technique as written in Section 3 is utilized for transforming the data set in the spatial domain into its eigen domain. This transformation is then followed by Fisherfaces technique, and the result is directly inputted to the 3-D recognition system. Using the data set as in the *Experiments#1* and *Experiments#2*, respectively, results of these experiments are depicted in Table 3 and Table 4. It can be seen that increasing the percentage of training face-image in different viewpoints will increase the recognition rate of the system. By using training/testing percentage of *60/40*, the recognition rate of the system is about *84%* at most. Results of the *Experiments#2* are presented in Table 4, that showing the same tendencies of higher recognition rate when using higher percentage of training/testing paradigm. The highest recognition rate could be achieved when using training/testing percentage of *58%* with its recognition rate of *87%* at most. Those results show that the recognition rates of the system to the *Experiments#1* and *Experiments#2* could be considered as high enough. As for comparison, by using 3-D face images on its spatial domain, which its value is directly fetched to the neural system, the recognition rate could only reach of about *64%* and

*74%* at most, for *Experiment#1* and *Experiment#2*, respectively[11]. However, to increase further the recognition rate of the system, we then propose the use of genetic algorithm to optimize the architectural structure of the circular-hidden layer.

## 5.2. Experimental Result with GA Optimization Procedure

The GA used the incremental (or steady state) replacement strategy where generation creates one new string, which is inserted in place of one of the worst *40%* of the current population. Simple one-point crossover is used and after a preliminary series of experiments is under taken; a set of suitable GA parameters could be determined. In these experiments, all of the image data is firstly transformed into its eigen domain. Result of experiments of using GA optimization procedure on the CHL-neural networks is depicted in Table 5 and Table 6 for *Experiment#1* and *Experiment#2*, respectively.

It can be seen that the optimization procedure of the neuron number in its cylindrical structure of hidden layer could increase the recognition system for all faces image. For the ***Experiment#1***, the average recognition rate (Table 5) is increased significantly up to *94%* for all of the data set, even the number of active neurons in the hidden layer decreased significantly to just about *63.7%* of the original number of neurons. Results for the ***Experiment#2***, as depicted in Table 6, shows the same behavior as for the *Experiment#1*, with the highest recognition rate is about *94%*. These results shows high recognition capability of the system to recognize unlearn faces with unknown position and various expressions. The comparis n of the recognition rate between all of the methodologies used in this system is concluded in Table 7. It shows clearly that the increasing of the recognition rate could be achieved up to *94 %* using different methods.

Table 1. Data set for the ***Experiments#1***.

| Data-Set | Train Images | Training position | Test Images | Testing position | Hiden Node |
|---|---|---|---|---|---|
| 1 | 100 | 0,45,90,270, 315 | 160 | 30,75,285, 330 | 8 |
| 2 | 140 | 0,30,60,90, 270,300,330 | 120 | 15,45,75, 285,315,345 | 12 |
| 3 | 180 | 10,30,50,70, 90,270,290, 310,330,350 | 80 | 0,20,40,60, 80,280,300, 320 | 16 |

Table 2. Data set for the ***Experiments#2***.

| Data Set | Expression | | Number of Images | | Neuron Position | |
|---|---|---|---|---|---|---|
| | Training | Testing | Training | Testing | Training | Testing |
| 1 | Normal & Smile | Free | 100 | 130 | 0, 45, 90 270, 315 | 0, 15, 30, 45, 60, 75 90, 270 |
| 2 | Normal & Smile | Free | 140 | 130 | 0, 30, 60, 90 270, 300, 330 | 0, 15, 30, 45, 60, 75 90, 270 |
| 3 | Normal & Smile | Free | 180 | 130 | 0,30,45,60 90,270,300, | 0, 15, 30, 45, 60, 75 90, 270 |

Table 3. Experimental results of the data set in *Experiments#1* in its eigen domain.

| Data | Procentage of | | Recognition Rate (%) | | | | | |
|---|---|---|---|---|---|---|---|---|
| Set | Training | Testing | Face#1 | Face#2 | Face#3 | Face#4 | Face#5 | Ave |
| 1 | 40 | 60 | 94 | 81 | 94 | 75 | 63 | 81 |
| 2 | 50 | 50 | 100 | 81 | 94 | 75 | 63 | 83 |
| 3 | 60 | 40 | 100 | 88 | 96 | 75 | 71 | 84 |

Table 4. Experimental results of the data set in *Experiments#2* in its eigen domain.

| Data | Percentage of | | Recognition Rate (%) | | | | | |
|---|---|---|---|---|---|---|---|---|
| Set | Training | Testing | Face#1 | Face#2 | Face#3 | Face#4 | Face#5 | Ave |
| 1 | 43 | 57 | 92 | 92 | 92 | 69 | 62 | 81 |
| 2 | 52 | 48 | 100 | 100 | 92 | 77 | 63 | 86 |
| 3 | 58 | 42 | 100 | 100 | 92 | 76 | 69 | 87 |

Table 5. Recognition rate of the CHL-neural networks system with GA optimization procedure for the *Experiment#1.*

| Data | Procentage of | | Recognition Rate (%) | | | | | |
|---|---|---|---|---|---|---|---|---|
| Set | Training | Testing | Face#1 | Face#2 | Face#3 | Face#4 | Face#5 | Ave |
| 1 | 40 | 60 | 100 | 81 | 97 | 100 | 84 | 92 |
| 2 | 50 | 50 | 100 | 94 | 94 | 100 | 81 | 94 |
| 3 | 60 | 40 | 100 | 88 | 96 | 100 | 88 | 94 |

Table 6. Recognition rate of the CHL-neural networks system with GA optimization procedure for the *Experiment#2.*

| Data | Percentage of | | Recognition Rate (%) | | | | | |
|---|---|---|---|---|---|---|---|---|
| Set | Training | Testing | Face#1 | Face#2 | Face#3 | Face#4 | Face#5 | Ave |
| 1 | 43 | 57 | 92 | 85 | 77 | 100 | 77 | 86 |
| 2 | 52 | 48 | 100 | 85 | 92 | 100 | 85 | 92 |
| 3 | 58 | 42 | 100 | 92 | 92 | 100 | 85 | 94 |

Table 7. Comparison of the recognition rate using spatial domain, eigen domain and its optimization structure through genetic algorithms for data set in *Experiments#1* and *Experiments#2*, respectively.

| Data | Recognition Rate | | | | | |
|---|---|---|---|---|---|---|
| | Experiment #1 | | | Experiment#2 | | |
| Set | Spatial | Eigen | GenAlg | Spatial | Eigen | GenAlg |
| 1 | 58 | 81 | 92 | 65 | 81 | 86 |
| 2 | 59 | 83 | 94 | 68 | 86 | 92 |
| 3 | 64 | 84 | 94 | 74 | 87 | 94 |

## 5. CONCLUSIONS

This paper presented 3-D face recognition system using modified multi-layer perceptron architectural network with back-propagation learning rule. It is shown that the use of image data in its eigen domain could increase the recognition rate of the system. Furthermore, optimization of number of neurons in its cylindrical hidden layer is performed by using genetic algorithms technique. Results show that even though the number of active neurons decreased to only about *63.7%*, the recognition rate of the system could increase to about *94%*. This result shows that the developed system could be used to recognize 3-D face images properly. The over all improvement in the classification accuracy that could be achieved by the GA based algorithm, is evidence that the lack of globally optimal strategy in determining the relation between number of neuron and the networks error calculation has a negative impact on their performance.

## REFERENCES


[1]. R. Brunelli and T. Poggio, "Face recognition through geometrical features", *Proceedings of ECCV 92, Santa Margherita Ligure*, pp. 792-800, 1992.

[2]. Craw, D. Tock and A. Bennet, "Finding face features", *Proceedings of ECCV 92, Santa Margherita Ligure*, pp. 93-96, 1992.

[3]. M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human face", *IEEE Trans PAMI*, vol 12, no.1, pp. 103-108, 1990.

[4]. M. Turk and A. Pertland, "Face recognition using Eigenfaces", *Proc. IEEE CCVP'91*, pp. 586-591, 1991.

[5]. S. Ullmann and R. Basri, "Recognition of linear combination of models," *IEEE Trans. PAMI*, vol13, no.10, pp. 992-1007, 1991.

[6]. T. Poggio and S. Edelman, " A network that learns to recognize three dimensional objects," *Nature*, vol. 343, no. 6255, pp. 263-266, 1990.

[7]. Belhumeur, P.N., Hespanha, J. P., Kriegman, D. J., *Eigenfaces vs. Fisherfaces: Recognition Using Class Spesific Linear Projection*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, No.7, Juli 1997.

[8]. J. H. Holland, "Outline for a logical theory of adaptive systems", *J. Assoc. Comput. Machinery*, 3, 297-314, 1962.

[9]. J. H. Holland, " *Adaptation in natural and artificial systems"*, Ann Arbor, University of Michigan Press, 1975.

[10].D. E. Goldberg, *Genetic Algorithm in Search, Optimization, and Machine Learning*, Addison – Wesley, 1989.

[11].B. Kusumoputro and A. Sulita, "Genetic algorithms in optimization of cylindrical-hidden neural network for 3-D object recognition system, *Journal of Neural Network World*, vol. 4, 631-639, 2000