Journal of Engineering and Development, Vol. 14, No. 4, December (2010) ISSN 1813-7822

Eye-Identification System Based on Back-Propagation NN Recognizer

Dr. Mohammed A. Abdala
Asst. Prof.
mohammedalmushdany@yahoo.com

Mrs. Ban M. Khammas
Asst. Lect.
ban.moh79@yahoo.com

Mrs. Hamsa A. Abdullah
Asst. Lect.
hamssa_kareem@yahoo.com

Information Engineering College/ Nahrain University / Baghdad/Iraq

Abstract

This research paper deals with the implementation of an eye recognition system using neural network (recognition classifier). The proposed system contains two phases, preprocessing and recognition. The resized image is providing faster processing for training and testing, the size of image is 120×90 pixels. The preprocessed phase extract features from eye image and use it as input to neural network which uses back-propagation algorithm to recognize the eyes. The proposed algorithm use single neural network as classifier, which consists of three layers with tangent sigmoid, and linear transfer function respectively. The system trained 60 eye samples. After testing the system with (100) eye samples a recognition rate of 100% is obtained. This recognition rate value is perfectly suitable for eye recognition systems.

Keywords— eye recognition, neural network, back-propagation, pre-processing.

الخلاصة

لقد تم في هذا البحث تنفيذ لنظام تمييز العين باستخدام الشبكة العصبية. ان النظام المقترح يتضمن جزئين : معالجة اولية للمعلومات وتمييز المعلومات. لقد تم توحيد حجم صورة العين كي تؤدي الى زيادة سرعة تدريب الشبكة واختبارها حيث ان حجم الصورة المستخدمة في هذا البحث هو (120×90)وحدة.

يقوم الجزء الاول من النظام (المعالجة الاولية) باستخراج العناصر ألاساسية لصورة العين وادخالها للشبكة العصبية التي تستخدم خوارزمية (back-propagation) لتمييز العين. وان الجزء الثاني من النظام المقترح (تمييز المعلومات) يستخدم شبكة عصبية أحادية تحتوي على ثلاث طبقات (طبقة الادخال, طبقة الاخفاء, طبقة الاخراج) تعتمد دالة (tangent sigmoid) ودالة التحويل الخطية (linear transfer function) بالتعاقب. لقد تم تدريب الشبكة العصبية للنظام باستخدام (60) نموذج مختلف للعيون.

بعدها تم اختبار النظام بأستخدام (100) نموذج وكانت نسبة التمييز 100%. ان هذه النسبة العالية للتمييز تؤهل النظام لاستخدامه في تمييز العيون.

1. Introduction

With the advent of electronic medium, especially computer, society is increasingly dependent on computer for processing, storage and transmission of information**.** Computer plays an important role in every parts of today life and society in modern civilization. With increasing technology, man becomes involved with computer as the leader of this technological age and the technological revolution has taken place all over the world based on it. This enable humankind to enter into a new world, commonly known as the technological world. Computer vision is a part of every day life. One of the most important goals of computer vision is to achieve visual recognition ability comparable to that of human [1],[2],[3].

In this paper we propose a computational model of eye recognition, which is fast, reasonably simple, and accurate in constrained environments such as an office or a household. The proposed approach has advantages over the other eye recognition schemes in its speed, simplicity, and learning capacity. The proposed feature extraction approach is relatively insensitive to small or gradual changes in the eye image.

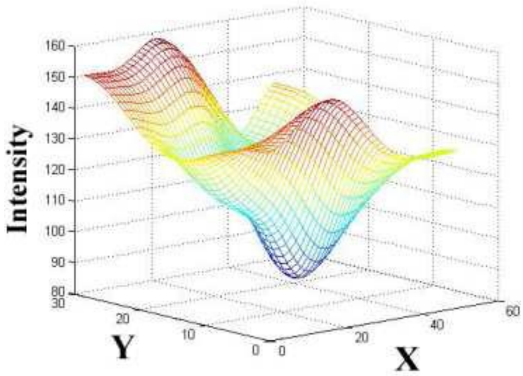
There are no requirements for expensive or specialized equipment; a system may be built using a simple video camera and a personal computer. The system is passive. There is no need to touch something by fingers or palm, no need to say any word or lean eye to a detector. Any person just may walk or stay before the camera, and the system performs recognition. It is especially useful in everyday usage. The design of a neural network-based eye recognition system is presented using both image processing approach and Neural Networks (adapting the Matlab utilities). A series of experiments with back propagation neural networks convert synthetic video images into eye coordinates by an enhanced feed-forward neural network with multiple winning hidden layer nodes.

2. Eye detection

Eye detection in 2D images is a crucial step in many machine vision applications such as face detection, face recognition, expression analysis. A proper solution for the addressed problem plays an important role in developing systems for new applications such as measuring awareness of car drivers, human computer interaction (HCI), video conferencing and disabled people aiding system. The success of these applications directly depends on accuracy and robustness of eye detection. Significant variation of eye appearance in image which is result of eye size, position of head, eye closing, lighting condition and occlusion by hair and frame of glasses, makes eye detection a challenge. The methods for eye detection are classified in three categories: template based, feature based and appearance based methods. In the first group of methods a template based on sketch of eye model is constructed. A given face image is matched against the template to find the eyes location [4]. The success of this approach totally depends on consistency of the model and eyes from lighting condition, rotation and scaling points of view.

Regardless of this problem, computational complexity of this approach is high. The attempts in feature based methods are to search for discriminative eye features in the image such as eye corners, intensity of iris or color distribution of iris. Finally an appearance based eye detection method aims to learn eye image using raw images. The learnt system is then ready to search for presence [5].

The representation of an eye intensity image can be considered as 3D function as shown in Fig.1. As seen from this figure eye surface has an obvious local minimum (pit) at centre surrounded by hillside features. This gives us a hint that eyes can be detected by exploring their terrain features.



(a) *(b)*

Figure 1. (a) Intensity image of eye (b) eye representation as a surface in 3D space

The second group of method is started with Wang et al [6] proposed a method for eye detection based on terrain feature matching. In this method, first a face image is represented using topographic features from which a terrain map is generated. The terrain map is composed of topographic labels. Second, similarity between the terrain map of eye model and that of the test image is measured. The authors simply use the statistical distribution of terrain features for comparing two terrain maps. There are some problems with the way of comparing the topographic features in Wang et al method. First, the distributions of terrain features do not characterize shape and geometrical distribution of a topographic label. So many regions in image may have similar statistical distribution for terrain features. Second, the statistical distribution of terrain features is not invariant to geometrical transformations. So simply scale change between eye model and face image degrades the eye detection performance [7].

3. Background Theory

Feature extraction is the process of interacting with images and performs extraction of meaningful information of images with descriptions based on properties that are inherent in the images themselves. Color information is the most intensively used feature for image retrieval because of its strong correlation with the underlying image objects.

This research used two methods to extract features from color eyes images (color moments and geometric moment). Then these features will be used as input for NN. The proposed system is shown in fig.2.

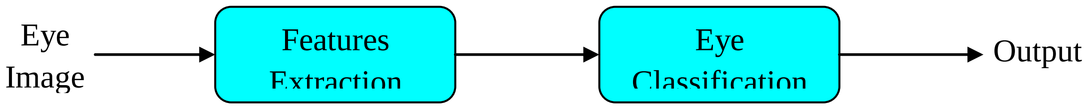


Figure 2. The Proposed System

4. Eye Feature Extraction with Moments

Moments of images provide efficient local descriptors and have been used extensively in image analysis applications. Their main advantage is their ability to provide invariant measures of shape. Moment based feature descriptors have evolved into a powerful tool for image analysis applications. Geometric moments present a low computational cost. Furthermore reconstruction is extremely difficult. Although not invariant under rotation, Hu's invariants that are derived from geometric moments present invariance under linear transformations. Complex moments provide with additional invariant descriptors. Computational complexity, however, becomes a major issue, and real-time implementation in software has not been reported. Moments of discrete orthogonal basis have been proposed recently [8]. They are fast to implement, present adequate noise tolerance and very accurate image reconstruction. Their major drawback is the lack of invariance under transformation. Image normalization should be used prior to moment extraction for applications requiring invariance [9].

4.1 Geometric Moments

Geometric moments have proven to be a very efficient tool for image analysis. Examples of the use of moments for aircraft identification, scene matching, shape analysis, image normalization, character recognition, accurate position detection, color texture recognition, image retrieval and various other image processing tasks. For a two-dimensional density function p(x,y) the (p+q)th order geometrical moments m_{pq} are defined by [10]:

$$m_{pq} = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} x^p\, y^q\, p(x,y)dxdy \qquad (1)$$

If p(x,y) is a piece-wise continuous function and has non-zero values only in the finite part of the x-y plane, then moments of all orders exist for p(x,y), and the moments sequence m_{pq} is uniquely determined by p(x,y) and vice-versa. Although originally described in continuous form, discrete formulae are commonly in use for practical reasons [9]. Two-dimensional moments of a digitally sampled $M \times N$ image that has gray function f(x, y), $(x = 0, \ldots M)$ and $(y=0, \ldots N)$ is given as:

$$m_{pq} = \sum^{M} \sum^{N} x^p \, y^q \, f(x,y) \qquad \textbf{(2)}$$

It should be noted that eq. (1) can assume very large values, especially for high order moments (large p, q). This often leads to numerical instabilities as well as high sensitivity to noise. Furthermore, image reconstruction is not straightforward, due to the fact that geometric moments are not of orthogonal basis (Cartesian monomial function).

4.2 Complex Moments

Complex moments are defined similarly to geometric moments. If an image is considered as a discrete function f(x,y) with x=0,1,…,M and y=0,1,…,N, then[10]:

$$m_{pq} = \sum^{M} \sum^{N} (x + iy)^p (x - iy)^{q} f(x,y) \qquad (3)$$

They present rotation invariance. Hu’s invariants have been proven to be a special case of complex moments. Again, these moments are not of orthogonal basis, and present the same drawbacks with geometric ones.

5. Color Feature Extraction

Color moments are measures that can use differentiate images based on their features of color. The basis of color moments lays in the assumption that the distribution of color in an image can be interpreted as a probability distribution. Probability distributions are characterized by a number of unique moments (e.g. Normal distributions are differentiated by their mean and variance). It therefore follows that if the color in an image follows a certain probability distribution, the moments of that distribution can then be used as features to identify that image based on color. Stricker and Orengo use three central moments of an image's color distribution in which p^k_{xy} is the value of the k-th color component of the xy-image pixel, M is the height of the image, and N is the width of the image. They are Mean, Standard deviation and Skewness [11] [12]:

5.1 Moment1 – Mean:

$$E_k = \frac{1}{MN} \sum_{x=1}^{M} \sum_{y=1}^{N} f^{k}(x,y) \qquad (4)$$

Mean can be understood as the average color value in the image.

5.2 Moment 2 - Standard Deviation:

The standard deviation is the square root of the variance of the distribution.

$$SD_k = SQRT\left(\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(f^k(x,y) - E_k)^2\right) \qquad (5)$$

5.3 Moment 3 – Skewness:

Skewness can be understood as a measure of the degree of asymmetry in the distribution.

$$S_k = \left(\frac{1}{MN}\sum_{x=1}^{M}\sum_{y=1}^{N}(f^k(x,y) - E_k)^3\right)^{1/3} \qquad (6)$$

6. Eye Classifier

Multilayer Perceptron (MLP) Neural Network is a good tool for classification purposes [13]. It can approximate almost any regularity between its input and output. The NN weights are adjusted by supervised training procedure called back-propagation. Back-propagation is a kind of the gradient descent method, which search an acceptable local minimum in the NN weight space in order to achieve minimal error. Error is defined as a root mean square of differences between real and desired outputs of NN. During the training procedure MLP builds separation hypersurfaces in the input space. After training MLP can successfully apply acquired skills to the previously unseen samples. It has good extrapolative and intrerpolative abilities. Typical architecture has a number of layers following one by one [13]. MLP with one layer can build linear hypersurfaces, MLP with two layers can build convex hypersurfaces, and MLP with three layers can build hypersurfaces of any shape.

We have used images scaled down by factor of 1000 in order to speed up learning. Previous experiments [14] have showed that such scaling did not change recognition rate [15]. The back-propagation algorithm is a supervised learning technique based on error correction and works by passing through the network in two ways. ***In Forward Pass,*** the input signal propagates through the layers of the network producing the output signals.

During this process the synaptic weights are fixed. However, in ***Backward Pass*** the error signal propagates backwards in the neural net, updating the synaptic weights in order to make the outcome closer to the desired. Back-propagation algorithm is the most popular approach to implement learning in neural networks (also called generalized gradient descent and generalized delta rule) [16][17].

This algorithm is a multi-layer network using a weight adjustment based on the sigmoid function. The sigmoidal function is very popular for neural networks, because it performs very similar to a step function, but it is everywhere differentiable.

Keep in mind that the goal of the ANN is to learn a proper, fixed set of weights that will reduce total error and thus give good predictions of insurance risk based on color (and other input variables). Whatever these weights may be, they represent the total effect of color upon all downstream nodes.

The activation level will further modulate that effect, but changes to that activation are severely constrained as to possible changes in effect. In essence, they are constrained to linear behavior. If the activation level goes up, then the absolute value of the effect will also increase. This method of training is an example of a supervised learning, where the target of the function is known.

The basic idea behind back-propagation learning is to gradually adjust the weights of an artificial neural network (ANN) so as to reduce the error between the actual and desired outputs on a series of training cases. Each case is typically a pair, (di; ri), indicating an element of the mapping between a domain and a range for some (yet unknown) function. By training the ANN to reduce this error, one effectively discovers the function. Figure 3 summarizes the basic process, wherein training cases are presented to the ANN, one at a time. The domain value, di, of each case is encoded into activation values for the neurons of the input layer.

These values are then propagated through the network to the output layer, whose values are then decoded into a value of the range, r*, which is compared to the desired range value, ri. The difference between these two values constitutes the error term.

Figure 3 the essence of the back-propagation algorithm: During training, cases are encoded and sent through the network. The resulting outputs are compared to desired outputs to compute an error term, E, which is then used to compute changes to the weights of the network so as to reduce E. Error back-propagation learning is the chosen method to update the free parameters of the neural network. It is based on an error-correction technique.

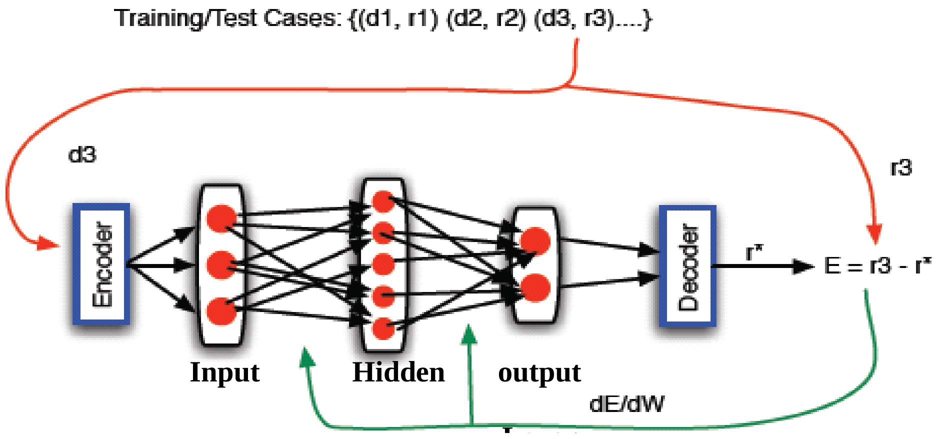


***Figure 3**. Back-propagation System*

7. Proposed Eye Identification System

7.1 Proposed Technique:

We describe here the foundation of this work and the development of the neural network algorithms to correctly detect the eye. Since the Back Propagation Network (BPN) allows each hidden layer node to contribute to the resulting output of the network, it should do better. Unfortunately, the BPN performance is susceptible to the training set and to the network configuration. Choosing the right network configuration and training the network with the right data and for the right number of epochs can improve the results dramatically. However, trial training sessions are time consuming and mostly empirical (trial and error) [18].

This work deals with eye recognition using low resolution images through Neural Networks. In the proposed technique (see Fig. 4), image resizing is applied to a unified image size for all eye samples. In order to accomplish eye recognition task, a single neural network is incorporated. Implementation is divided into two phases. The pre-processing phase and the neural processing phase. In Pre-processing phase time effective preprocessing is performed in order to make image data best fit for neural network input. Image size (120×90) pixel is reduced in order to make it light but efficient and best fit for neural processing phase. Before handling images to neural processing, all the images data undergoes process of vectorization discussed in upcoming section.

This reduction is done to reduce redundant information and to make neural network input data low dimensional, because of the fact that input layer of neural network require the same number of neurons as number of pixels in an input image.

In this way, less number of input neurons makes it less complex and easy for neural network training and testing phases. This phase output images of low resolution, which have the information required for recognition. Neural network further got two steps training and testing, in training neural network learns through examples, training set of images is presented as input, each layer output is calculated through transfer functions acting as summation junction, and it uses back propagation algorithm to accomplish recognition training part to achieve desired results.

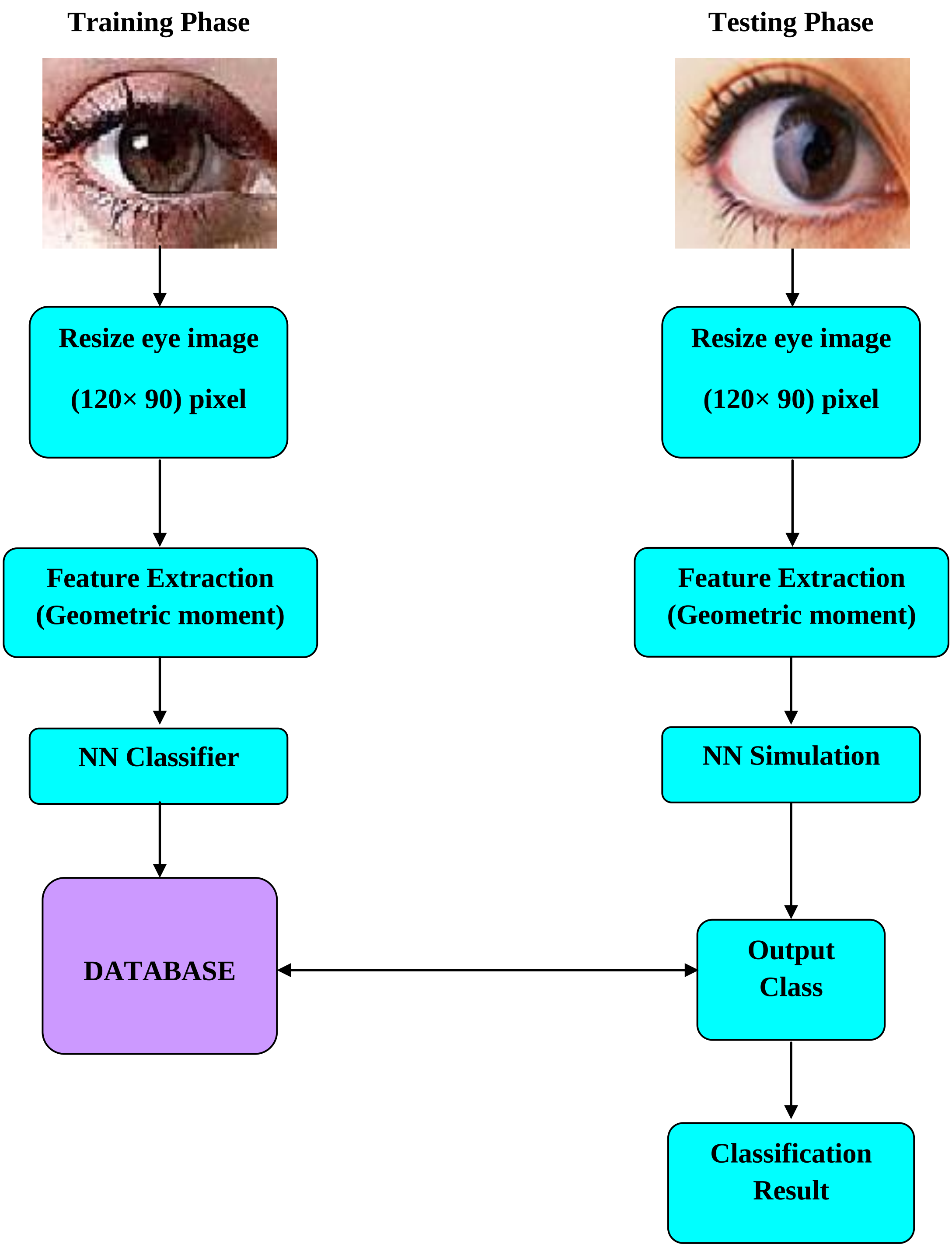


Figure 4. Proposed Eye-Classification System

7.2 Applied Matlab Tools Proposed

In this study, we use Matlab as a programming language for both image processing and neural network phases since Matlab is widely used in all areas of applied mathematics in education and research. Matlab stands for matrix Laboratory and the software is built up around vectors and matrices [19]. This makes the software particularly useful and great tool for solving algebraic and differential equations and for numerical integration. Matlab also has some toolboxes useful for signal processing, image processing, neural network, Database, Wavelet… etc, which have ready built-in functions. The image is stored as matrix using standard Matlab matrix conventions. There are five basic types of image supported by Matlab: Indexed images, Intensity images, Binary images, RGB images and 8-bit images [19-20]. The images that used in this paper are RGB images.

Artificial Neural Networks (ANNs) as a tool in Matlab are networks of interconnected simple units that are based on a greatly simplified model of the brain. ANNs are useful learning tools by being able to compute results quickly interpolating data well. ANNs are composed of units (also called nodes) that are modeled after neurons, with weighted links interconnecting the units together (see Fig. 5). The main difference between ANNs and other learning mechanisms is that it is composed of these simple units and they work together in a highly parallel manner. The Neural Networks have many algorithms that specify how this network should work [17].

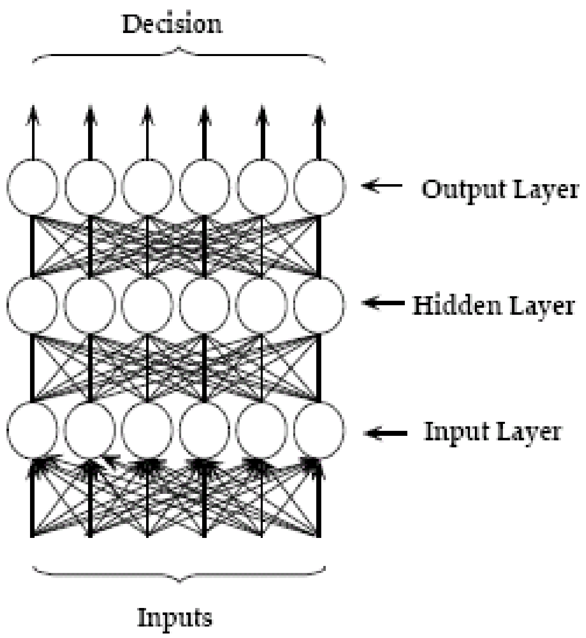


Figure 5. Neural Network Architecture

8. System Results

8.1 Vectorization and Targets Creation

Vectorization is the process of transforming image data in the form of vectors. The 2D image is vectorized to form 1-dimensional vector as neural network requires 1D vector for processing. Hence all the images are contained in a matrix, in this way a matrix column represents an image's whole data. The research work is based upon *supervised learning* rule in which targets are provided to the neural network in order to make suitable adjustments of weights to achieve the desired output while training. The outputs neurons are told in advance what there response should have to be, whereas the method used for error convergence is Least Mean Square (LMS). Target vector matrix is constructed before handing over the images as input to the Neural Network.

8.2 Training and Transfer Functions

The architecture of the proposed single neural network is based upon multi layers, it uses three layers, input, hidden and output layer. Pre-processed images in the vector format are presented as input to the neural network which contains 13 neurons in the input layer, and 27 neurons in the hidden or processing layer of this network. Finally, the output layer contains neurons equal to the number of subjects under consideration. The algorithm make efficient use of single neural network and minimizes the gradient of the error through adjusting weights and biases continuously with momentum. Momentum acts like a low pass filter and ignores small features in error surface so that network do not stuck into a shallow local minimum. Neural Network is trained upon some set of images, and tested upon unseen images. In proposed technique neural network uses back propagation algorithm for error computation and new weights calculation for each neuron link.

The network undergoes process of training, continuously in an iterative manner it calculates output from each layer, extracting the mean square error and propagating it backwards if it is not approaching targets. Due to this backward error propagation, error-signal for each neuron is calculated. Which in fact is used for neuron weight updating. If its approaching targets then training is considered done. It has been observed that as the number of subjects increases, the training time also increases (as the complexity of the input increases with the increase in the number of eye images (subject classes)). In this way network approaches the set known correct outputs (targets) in order to be trained. A number of eye images (subject classes) are used to subjects train the network. Appendix (A) shows the (60) eye images used in this work. The process of training these images is shown in Fig. 6, in which training curve is approaching its goal through readjustment of weights and biases.

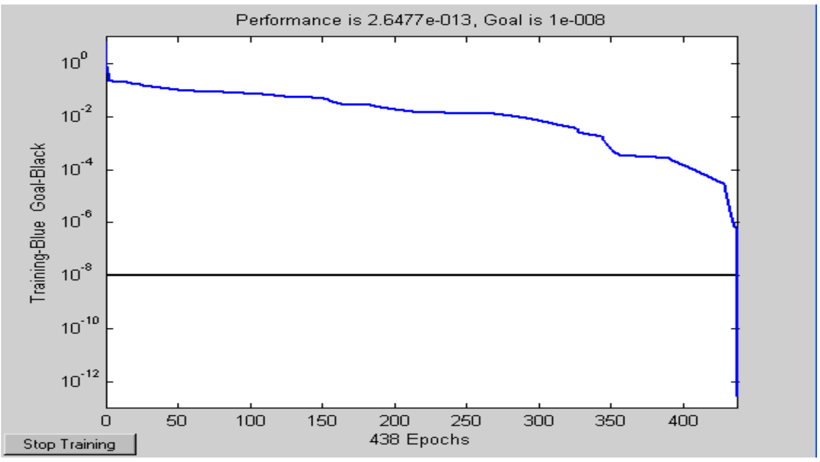


Figure 6. Process of training

The response of the Neural Network is dependant upon weights, biases and transfer functions. The transfer functions used in the feed-forward back propagation neural network are purelin, tansig. These functions acts as summation junction and calculates the output from the inputs presented. After the training phase is completed, the identification process must be implemented in order to evaluate the proposed system. The evaluation process is accomplished by testing the system with known and newly eye images. A new images for testing are applied to the trained neural network along with already trained images for calculating the percentage of accuracy and error. A set of (100) eye samples are used to test the proposed system (as shown in appendix B).

8.3 Results and Discussion

The algorithm is implemented using *2.4* GHz Pentium 4 machine with Windows XP and MATLAB 8.0 as the development tool. Two set of images are required, one for the training of the neural network and another set of images upon which testing is done. Each image is of size 120×90 pixels. In this research, the testing subjects reach to 100 images. These 100 images are divided into 60 known images (previously trained one) and 40 images (newly untrained one). The first part evaluation computes moments. Two set of moments are evaluated, the first one for each of the three color components. Each color component yield a feature vector of three elements mean, standard deviation, and skewness. These nine feature vectors are calculated for each eye image. The second one evaluate four geometric moments for the gray eye images. These four feature vectors are calculated for each gray eye image. Thus, a total of thirteen feature vectors are calculated for each eye image.

The second part enters these features to the NN in order to be trained by them. Table (1) shows the recognition phase of the 100 testing eye samples. These results are analyzed to calculate the recognition rate of the system as shown in table (2).

A recognition rate of 100% is obtained for this system. This recognition rate value is perfectly suitable for eye recognition systems.

Table (1) Recognition Phase

Eye Samples	Actual Class	Calculated Class	Eye Samples	Actual Class	Calculated Class
1	1	1	51	51	51
2	2	2	52	52	52
3	3	3	53	53	53
4	4	4	54	54	54
5	5	5	55	55	55
6	6	6	56	56	56
7	7	7	57	57	57
8	8	8	58	58	58
9	9	9	59	59	59
10	10	10	60	60	60
11	11	11	61	New	None
12	12	12	62	New	None
13	13	13	63	New	None
14	14	14	64	New	None
15	15	15	65	New	None
16	16	16	66	New	None
17	17	17	67	New	None
18	18	18	68	New	None
19	19	19	69	New	None
20	20	20	70	New	None
21	21	21	71	New	None
22	22	22	72	New	None
23	23	23	73	New	None
24	24	24	74	New	None
25	25	25	75	New	None
26	26	26	76	New	None

27	27	27	77	New	None
28	28	28	78	New	None
29	29	29	79	New	None
30	30	30	80	New	None
31	31	31	81	New	None
32	32	32	82	New	None
33	33	33	83	New	None
34	34	34	84	New	None
35	35	35	85	New	None
36	36	36	86	New	None
37	37	37	87	New	None
38	38	38	88	New	None
39	39	39	89	New	None
40	40	40	90	New	None
41	41	41	91	New	None
42	42	42	92	New	None
43	43	43	93	New	None
44	44	44	94	New	None
45	45	45	95	New	None
46	46	46	96	New	None
47	47	47	97	New	None
48	48	48	98	New	None
49	49	49	99	New	None
50	50	50	100	New	None

Table (2) Recognition Rate of Test Samples

Type	No. of Samples	Recognition Rate %
Previously Trained Samples	60	100
New Samples	40	100
Overall Recognition Rate %		100

9. Conclusions

Eye Recognition is basically a classification problem. In this research work neural network using back-propagation has been trained as an eye classifier to recognize eye with time effective pre-processing, which greatly increases the performance of the network. By lowering the resolution and using single neural network for whole recognition task, computational complexity has been reduced many times. A recognition rate of 100% is obtained for proposed eye identification system.

References

[1] A. Jain, **"Fundamentals of Digital Image Processing"**, Prentice-Hall, 1989.

[2] E. Trucco, and A. Verri, **"Introductory Techniques for 3-D Computer Vision"**, Prentice-Hall Inc., 1998.

[3] L. Shapiro, and G. Stockman, **"Computer Vision"**, prentice-Hall Inc., 2001.

[4] H. Tan, Y. Zang, and R. Li, **"Robust Eye Extraction Using Deformable Template and Feature Tracking Ability",** Proc. ICICS-PCM, Vol. 3, pp: 1747 – 1751, 2003.

[5] H. Gu, G. Su, and C. Du, **"Feature Points Extraction from Faces**", Proc Image and Vision Computing. New Zealand, pp: 154-158, 2003.

[6] J. Wang and L.Yin. **"Detecting and Tracking Eyes Through Dynamic Terrain Feature Matching"**, IEEE CVPR05 Workshop on Vision for Human Computer Interaction (V4HCI) in conjunction with IEEE CVPR, San Diego, CA, 2005.

[7] S. Ferdowsi and A. Ahmadyfard **, "Using Statistical Moments as Invariants for Eye Detection",** 16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, Department of Electrical Eng. Shahrood University of Technology Shahrood, Iran, saideh_ferdosi@yahoo.com, ahmadyfard@shahroodut.ac.ir, 2008.

[8] R. Mukundan, **"Image analysis by Tchebichef moments"**, IEEE Trans. on Image Proc., Vol. 10, No. 9, pp: 1357-1364, 2001.

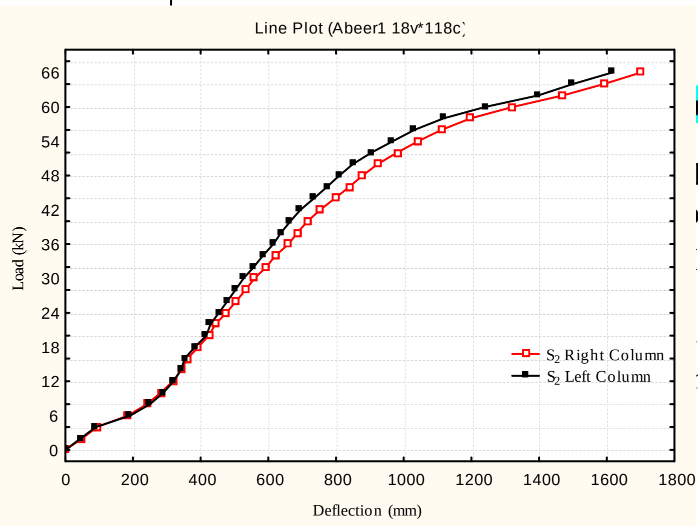
[9] L. Kotoulas and I. Andreadis **, " Image Analysis Using Moments"**, 5th Int. Conf. on Technology and Automation, Thessaloniki, Greece, pp: 360-364, 2005.

[10] M. Mercimek, K. Gulez and T. Mumcu**, "Real object recognition using moment invariants",** Yildiz Technical University, Electrical-Electronics Faculty, Electrical Engineering Department, Besiktas-Istanbul, Turkey, e-mail: mercimek@yildiz.edu.tr, gulez@yildiz.edu.tr, tmumcu@yildiz.edu.tr, 2005.

[11] L. Kotoulas and I. Andreadis, **"Image moments based on Hahn polynomials"**, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 29, pp: 2057-2062, 2007.

[12] S. Silakari, M. Motwani, and M. Maheshwari**, "Color Image Clustering using Block Truncation Algorithm"**, IJCSI International Journal of Computer Science Issues, Vol. 4, No. 2, pp:31-35, 2009

[13] G. Vladimir and G. Vitaly, **"Recirculation Neural Network Training for Image Processing"**, International conference on advanced computer systems Advanced Computer Systems, pp: 73-78, 1999.



plication of Recirculation Neural Network and **r Face Recognition"**, 2nd International Conference ntelligence. - Minsk: BSUIR, pp: 136-142, 2001.

ccess Control by Face Recognition Using Neural ng Cybernetics, Laboratory of Image Processing and 220012 Minsk, BELARUS, bdv78@mail.ru,

[16] Sandberg, Kristian, **"Introduction to Image Processing in Matlab"**, http://amath. colorado.edu/scico/tutorials/matlab/Image Process. Html

[17] I. El Emary, **"On the Application of Artificial Neural Networks in Analyzing and Classifying the Human Chromosomes",** Journal of Computer Science, Vol. 2, No.1, pp: 72-75, 2006.

[18] B. Wolfe and D. Eichmann, **”A Neural Network Approach to Tracking Eye Position**”, International Journal of Human-Computer Interaction, Vol. 9, pp: 59 – 79, 1997.

[19] **The Image Processing Toolbox (Version 3) of MatLab 6.5.**

[20] R. Gonzalez, **"Digital Image Processing",** 2nd Ed., Prentice-Hall, Inc, 2002.