A New and Effective Approach for Fingerprint Recognition by using Feed Forward Back Propagation Neural Network

Jyoti Rajharia
School of Engineering & Technology,
Jaipur National University,
Jaipur, Rajasthan, India

P.C. Gupta, PhD.
Department of Computer Science & Informatics,
University of Kota,
Kota, Rajasthan, India

ABSTRACT
Fingerprint recognition is one of the oldest form of biometric identification. It has been used for over a century because of their uniqueness and consistency over time. In this paper a new approach has been used in which feed forward back propagation neural network is implemented through matlab. The result obtained shows that the proposed approach somewhat improves the performance compared to the previous approaches.

Keywords
Pattern recognition, biometric identification, feed forward back propagation, neural network, matlab.

1. INTRODUCTION
Recognition of person by means of biometric characteristics is an emerging phenomenon in modern society. Fingerprint recognition is one of the oldest form of biometric identification. It has been used for over a century because of their uniqueness and consistency over time. Nowadays it is one of the most important and popular biometric technology mainly because of the inherent ease in acquisition, distinctiveness, persistence and high matching accuracy rate [1]. It is one of the most successful method used for person identification, which takes an advantage of the fact that, fingerprint of every individual is considered to be unique. No two people have the same set of fingerprints. Even identical twins do not have identical fingerprints. Finger ridge patterns do not change through out the life of an individual. This property makes fingerprint an excellent biometric identifier. It is also used as forensic evidence. It has received more and more attention during the last period due to the need for society in a wide range of applications. Among the biometric features, the fingerprint is considered one of the most practical ones. Fingerprint recognition requires a minimal effort from the user and provides relatively good performance. Fingerprint recognition refers to the automated method of verifying a match between two human fingerprints. Fingerprints are one of many forms of biometrics used to identify individuals and verify their identity. For fingerprint identification different works have been carried out. These are described as follows:

Mohamed et al., [2] presented fingerprint classification system using Fuzzy Neural Network. The fingerprint features such as singular points, positions and direction of core and delta obtained from a binarised fingerprint image. The method is producing good classification result.

Ching-Tang Hsieh and Chia-Shing – Hu [3] has developed anoid method for Fingerprint recognition. Ridge bifurcations are used as minutiae and ridge bifurcation algorithm with excluding the noise–like points are proposed. Experimental result shows the humanoid fingerprint recognition is robust, reliable and rapid.

Haiping Lu et al., [4] proposed an effective and efficient algorithm for minutiae extraction to improve the overall performance of an automatic fingerprint identification system because it is very important to preserve true minutiae while removing spurious minutiae in post-processing. The proposed novel fingerprint image post-processing algorithm makes an effort to reliably differentiate spurious minutiae from true ones by making use of ridge number information, referring to original gray-level image, designing and arranging various processing techniques properly, and also selecting various processing parameters carefully. The proposed post processing algorithm is effective and efficient.

Prabhakar S, Jain. A.K. et al., [5] has developed filter-based representation technique for fingerprint identification. The technique exploits both local and global characteristics in a fingerprint to make identification. Each fingerprint image is filtered in a number of directions and a 640-dimensinal feature vector is extracted in the central region of the fingerprint. The feature vector is compact and requires only 640 bytes. The matching stage computes the Euclidian distance between the template finger code and the input finger code. The method gives good matching with high accuracy.

Ballan M [6] introduced Directional Fingerprint Processing using fingerprint smoothing, classification and identification based on the singular points (delta and core points) obtained from the directional histograms of a fingerprint. Fingerprints are classified into two main categories that are called Lasso and Wirbel. The process includes directional image formation, directional image block representation, singular point detection and decision. This method gives matching decision vectors with minimum errors, and also this method is very simple and fast.

G. Sambasiva Rao et al., [7] proposed fingerprint identification technique using a gray level watershed method to find out the ridges present on a fingerprint image by directly scanned fingerprints or inked impression.

M. R. Girgisa et al., [8] proposed a method to describe a fingerprint matching based on lines extraction and graph matching principles by adopting a hybrid scheme which consists of a genetic algorithm phase and a local search phase. Experimental result of this method demonstrates the robustness of algorithm.

Duoqian Maio et al., [9] used principal graph algorithm by kegl to obtain principal curves for auto fingerprint identification system. From principal curves, minutiae extraction algorithm is used to extract the minutiae of the fingerprint. Results shows that curves obtained from graph algorithm are smoother than the thinning algorithm.

Alessandra Lumini and Loris Nanni [10] developed a method for minutiae based fingerprint and its approach to the problem as two - class pattern recognition. The obtained feature vector by minutiae matching is classified into genuine or imposter by Support Vector Machine resulting remarkable performance improvement.

Jain, Prabhakar and Hong's [11] proposed a Multi-channel Classification approach. This method can be found to be more accurate while classifying the fingerprint images as compared to its previous counterparts. The fingerprint images are classified into five categories: whorl, right loop, left loop, arch, and tented arch. The algorithm uses a novel representation (Finger Code) and is based on a two stage classifier to make a classification. The two-stage classifier uses a k-nearest neighbor classifier in its first stage and a set of neural network classifiers in its second stage to classify a feature vector into one of the five fingerprint classes. This algorithm suffers from the requirement that the region of interest be correctly located, requiring the accurate detection of center point in the fingerprint image. Otherwise, the algorithm can be found to be very effective.

Masayoshi Kamijo's [12] proposed Artificial Neural Network based approach, where a neural network for the classification of fingerprint images is constructed, that classifies the complicated fingerprint images. It uses a two-step learning method to train the four layered neural network which has one sub-network for each category. It carries out the principal component analysis (PCA) with respect to the unit values of the second hidden layer and also studies the fingerprint classification state represented by the internal state of the network. Consequently, the method confirms that the fingerprint patterns are roughly classified into each category in the second hidden layer and the effectiveness of the two-step learning process. However, in case of larger data sets this method can be found to give limited results.

Various methods have been used to identify a person through the fingerprint image. In this work to increase the accuracy of the result a new approach have been implemented in which each sample is divided into four parts and then feed forward back propagation neural network have been applied to identify a person. Section II describes about the pattern matching and its methods, in section III detailed methodology is shown, section IV contains data preparation while experiment, result and discussions are shown in the section V. Section VI contains conclusion and references are shown in the last section.

2. PATTERN MATCHING AND RECOGNITION

In computer science, Pattern matching is used to compare two patterns in order to determine whether they match or not (i.e., are they same or they differ). Pattern matching is the act of checking some sequence of tokens for the presence of the constituents of some pattern. In contrast to pattern recognition, the match usually has to be exact. The patterns generally have the form of either sequences or tree structures. Uses of pattern matching include outputting the locations of a pattern within a token sequence, to output some component of the matched pattern, and to substitute the matching pattern with some other token sequence. Pattern recognition is the assignment of a label to a given input value. An example of pattern recognition is classification, which attempts to assign each input value to one of a given set of classes [13]. In other words Pattern recognition is the research area that studies the operation and design of systems that recognize patterns in data.

There are various pattern matching methods. These are described as follows:

2.1 Statistical
2.2 Neural network

2.1 Statistical approach of Pattern Matching

Bayes theorem is an approach of statistical pattern matching. The goal of this approach is to classify the character in such a way that it minimizes the probability of misclassification.

2.2 Neural Network

It is a collection of processing units and adaptive connections that are designed to perform a specific processing function. They rely on a computing model that is loosely based on how the brain computes. The main features include a collection of many simple processors connected by adaptive connections. These simple processors sum up their inputs and calculate an output value, or activation. This output is then sent to other processing units in the neural network. The connections are called adaptive because they are adjusted during the training of the neural network. This training process usually consists of presenting examples of input/output relationships to the network. The connection weights are adjusted to minimize the difference between the actual network output value and the desired output value. Neural networks are very good at pattern-recognition and pattern-matching tasks [14]. The most popular neural network model, back propagation, is a feed-forward model using the generalized delta rule to adjust the weights. It is a pattern-matching network that can be used as a classifier, a time-series forecaster, or as a predictive model much like statistical regression models. It builds an internal model of the relationships between the input and output training patterns. When it is subsequently given an input vector, it tries to reproduce the associated output vector. If the input is one it has never seen before, it produces an output similar to the one associated with the closest matching training input pattern. This generalization capability of producing a reasonable output, even for an input pattern it has never seen before, is one of the advantages of neural networks.

In this work we are using feed forward neural network pattern matching method to recognize each person through its fingerprint image.

3. METHODOLOGY

The complete work has been shown in the following diagram:

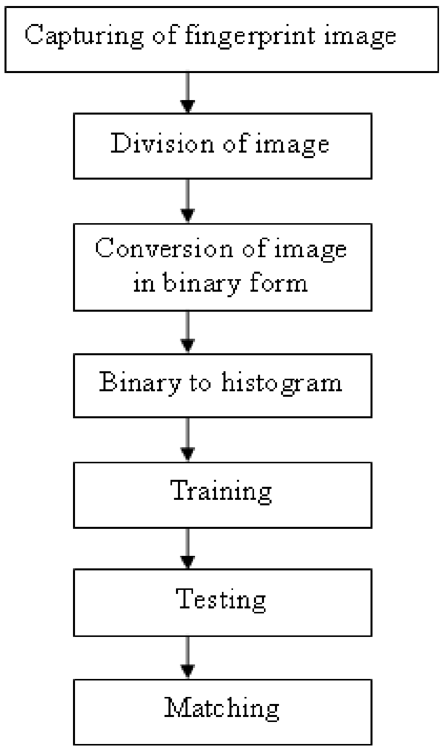

Fig 1: Fingerprint Recognition process

The various stages in a fingerprint recognition system are shown in figure 1. The first stage is the capturing of the image in which a fingerprint image is obtained from an individual. The next stage is division of the image in which every image is divided into 4 equal parts. The divided image is then converted into the binary form. The binary image is then converted into the histogram equivalent. A network is designed which contains neurons in the input layer in the hidden layer as well as in the output layer. The network is trained by applying different number of samples. Once the training phase is completed different samples are applied to the network to check its performance. Finally matching is used to check whether the network is able to recognize any existing image.

The goal in this classification problem is to develop an algorithm which will assign any image, represented by a vector x, to one of the class denoted by C_k, where, K= 1, 2,……..5, so that class C_1 corresponds to the fingerprint image of one person, C_2 corresponds to the fingerprint image of second person and similarly C_5 corresponds to the fingerprint image of fifth person. We have collected five different fingerprint images of each person. Such a collection will be referred to as a dataset. An image is represented by an array of pixels, each of which carries an associated value which we shall denote by xi (where the index i label the individual pixels). The value of x_i might, for instance, range from 0 for a completely black pixel. It is often convenient to gather the x_i variables together and denote them by a single vector $x = (x_1, \ldots\ldots x_d)^T$ where d is the total number of such variables and the superscript T denotes the transpose.

Our problem which we face stems from the high dimensionality of the data which we are collecting. For a typical image size of 256*256 pixels, each image can be represented as a point in a d-dimensional space, where d= 65536. The axes of this space represent the grey-level values of the corresponding pixels, which in our work represented by 8-bit number. In principle we might think of storing every possible image together with its corresponding class label. This is completely impractical due to the very large number of possible images: for a 256*256 image with 8-bit pixel values there would be $2^{8*256*256*256} \approx 10^{158000}$ different images. We might have a few thousand examples in our training set. It is clear then that the classifier system must be designed so as to be able to classify correctly a previously unseen image vector. The presence of a large number of input variables can present some severe problems for pattern recognition systems. One technique to help alleviate such problems is to combine input variables together to make a smaller number of new variables called features [15].

The above discussion can be processed in the following ways:

1. Scanned fingerprint images (Fig 2)
2. Divide the images into four equal parts (Fig3- Fig6)
3. Binary conversion of images
4. Finding out the histogram equivalent of given binary numbers
5. Training of each part individually using neural network tool through Matlab
6. Collect output of all the parts and apply it as input

7. All the above steps are used without dividing the image into different parts also

8. Results obtained from steps 6 and 7 are compared to check the accuracy of this approach.

Fig 2: fingerprint image

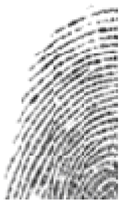

Fig 3

Fig 4

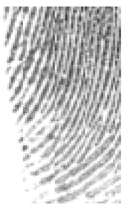

Fig 5

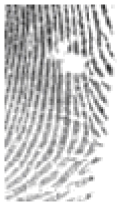

Fig 6

Following matlab functions have been used to carry out different task.

1. imread ( )
2. imshow ( )
3. Logical
4. im2bw
5. hist

In our work we have used imread ( ) to bring the scanned image into the matlab environment. For example, the statement

f=imread (‘sample.jpg’)

reads the image from the JPEG file sample into image array f. After reading the image, it is displayed on the matlab desktop using the function imshow ( ), by using syntax imshow (f). Where, f is an image array. Logical function converts numeric values to logical. Logical (f) converts the elements of the array f into logical, thus returning an array that can be used for logical indexing or logical tests. Logical can have the values 0 and 1 corresponding to false and true, respectively. Any nonzero real element of input array f is converted to a logical 1 while zeros in f become logical 0. Im2bw convert image to binary image. It produces binary images from indexed, intensity or RGB images. To do this it converts the input image to grayscale format and then converts this image to binary. The output binary image BW has values of 1 (white) for all pixels in the input image with luminance greater than level and 0 (black) for all other pixels. hist function finds the histogram values of the image. For example:

Xh= hist(x (:), n) ;

computes n-bin histogram. Where bin is the number of bins used to generate the histogram.

After finding the histogram equivalent of each part of the fingerprint image of the given binary numbers, these histogram values are taken as input and some random values are assigned as target values to train, test and validate each part individually using feed forward back propagation neural network tool. After training of each part we have taken all the four trained output as a single input and trained it in a unique value. This whole process described above is also implemented with the full fingerprint image of a person without any partition.

4. DATA PREPARATION

In our work we have taken 5 samples of each person. So, we have collected 25 samples and every sample is divided into 4 parts, each part is trained individually in matlab’s neural network tool. For simplicity the complete process of one person is shown in this paper. Each fingerprint image is divided into four equal parts and each part is converted into the appropriate form by using different functions described in the section II. To minimize the size of input data binary values are converted into the 4-bin histogram values. After finding the histogram equivalent of the value, training of each part is done in matlab’s neural network tool. Table1 shows that the histogram values which we have find is taken as input data and some random values using 4 binary digits are taken as target values to train each part of every sample individually.

Table 1: Sample data of one person

S.NO.	PERSON	SAMPLE	PART	INPUT DATA				TARGET DATA
1	1	1	1	0	0	480	0	1 1 0 1
2	1	1	2	18	0	0	590	1 0 1 0
3	1	1	3	13	0	0	467	1 1 0 0
4	1	1	4	25	0	0	515	0 0 0 1
5	1	2	1	85	0	0	4051	1 1 0 0
6	1	2	2	195	0	0	4789	0 0 0 1
7	1	2	3	46	0	0	3772	1 1 1 0
8	1	2	4	43	0	0	4429	0 1 1 1
9	1	3	1	139	0	0	8246	0 0 0 1
10	1	3	2	260	0	0	9265	0 1 1 0
11	1	3	3	302	0	0	8538	1 0 1 1
12	1	3	4	354	0	0	9396	0 1 0 0
13	1	4	1	439	0	0	5933	1 0 0 0
14	1	4	2	738	0	0	6338	1 0 0 1
15	1	4	3	293	0	0	4191	0 0 1 0
16	1	4	4	896	0	0	7034	1 1 0 1
17	1	5	1	248	0	0	6772	0 0 1 1
18	1	5	2	447	0	0	6807	1 0 0 0
19	1	5	3	402	0	0	9023	0 1 0 1
20	1	5	4	510	0	0	8274	1 1 0 1

In table 2 target data of each part of fingerprint is collected and is taken as a new input data for each sample of the same person. A new target data is set to map all the samples of the same person.

Table 2: Different sample Inputs and concerning Output for one person

S.NO.	PERSON	SAMPLE	INPUT DATA	TARGET DATA
1	1	1	1101101011000001	
2	1	2	1100000111100111	
3	1	3	0001011010110100	0101010101010101
4	1	4	1000100100101101	
5	1	5	0011100001011101	

Table3 shows 5 fingerprint samples of the same person without dividing them into different parts.

Table 3: Different fingerprint samples of the same person without composition

S.NO.	PERSON	SAMPLE	INPUT DATA				TARGET DATA
1	1	1	0	0	2829	0	
2	1	2	183	0	0	20793	
3	1	3	382	0	0	36512	0101010101010101
4	1	4	2307	0	0	27453	
5	1	5	1432	0	0	31142	

5. EXPERIMENTS AND RESULTS

All the experiments and their results and discussion are shown in this section. A two layer feed-forward back propagation neural network has been used for complete experiment. Following diagram of the network is used for the complete experiment.

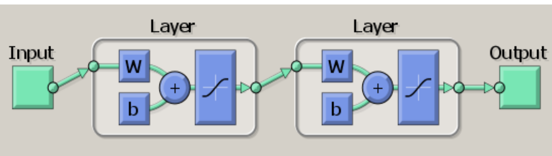

Fig 7: Neural Network used for experiments

Neural Network tool of Matlab 10 have been used for the complete work. Three different kinds of samples are applied on the network to perform different activities:

Training: These are presented to the network during training and the network is adjusted according to its error.

Validation: It is used to measure network generalization, and to halt training when generalization stops improving.

Testing: It is used only for testing the final solution in order to confirm the actual predictive power of the network. In our work we have conducted several training sessions. The training measures the performance on the basis of Mean Squared Error. It is the average squared difference between output and target. Lower values of mean square errors are considered as better one while zero denotes no error.

EXPERIMENT 1: Part 1 of First Sample of Person One

In this experiment we have taken 1000 epochs and the training was completed in 12 iterations. The network is trained through Feed-forward back propagation algorithm. Figure displays the training progress of the network.

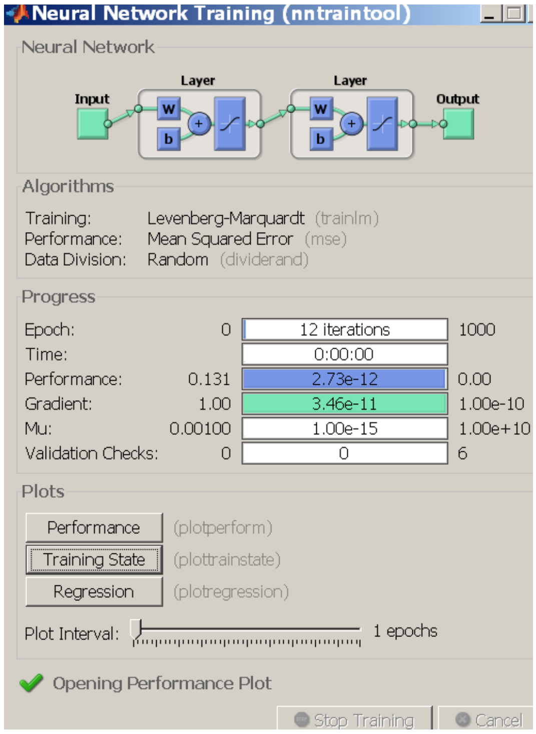

Fig 8: Network training result

From the above figure it seems clear that the total number of iterations to train the network is 12. The performance in terms of mean squared errors and the value of gradient and validate checks is also shown in the above figure.

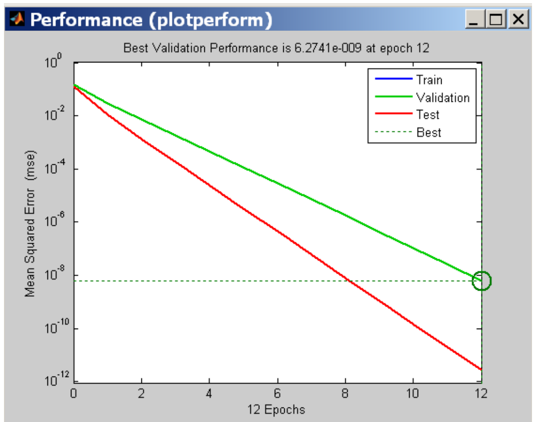

Fig 9: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 6.2741e-009 at 12 epochs.

EXPERIMENT 2: Part 2 of First Sample of Person One

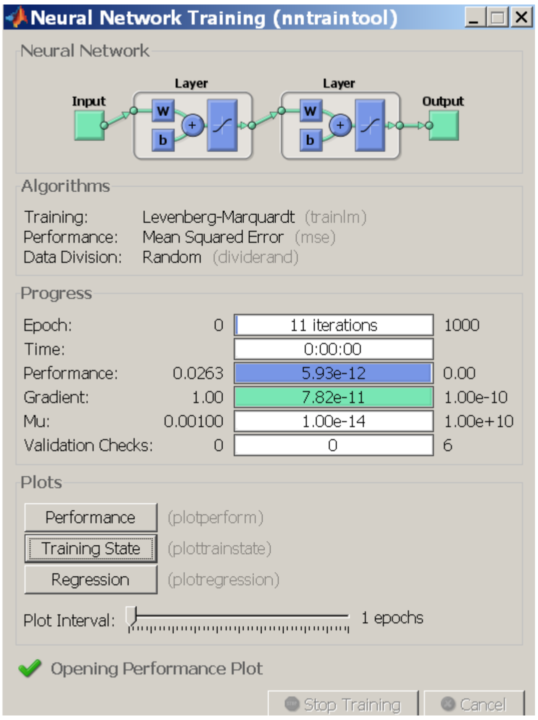

Fig 10: Network training result

From the above figure the total number of iterations to train the network is 11. The performance in terms of mean squared errors, the value of gradient and validate checks is also shown in the figure.

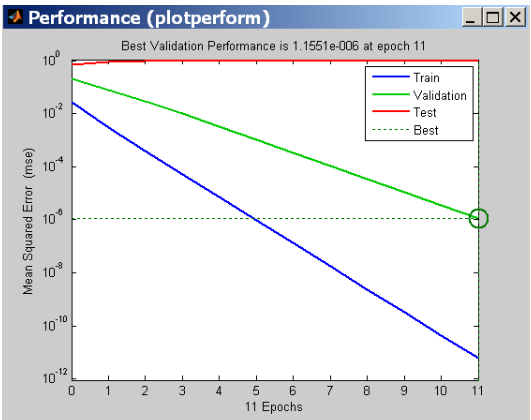

Fig 11: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 1.1551e-006 at 12 epochs

EXPERIMENT 3: Part 3 of First Sample of Person One

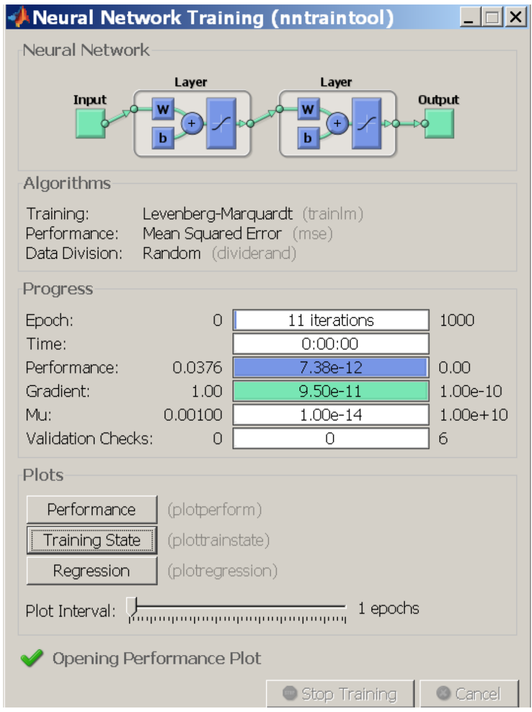

Fig 12: Network training result

Above figure shows the total number of iterations to train the network is 11.

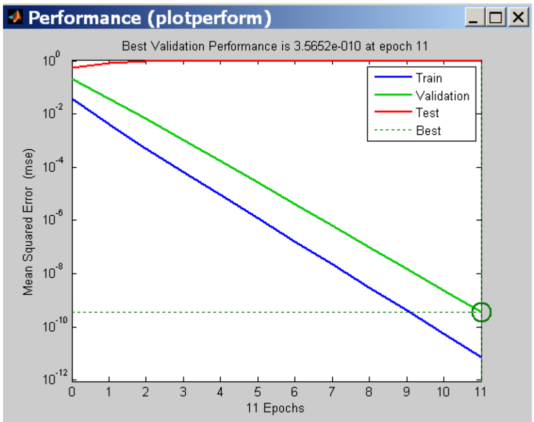

Fig 13: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 3.5652e-010 at 11 epochs

EXPERIMENT 4: Part 4 of First Sample of Person One

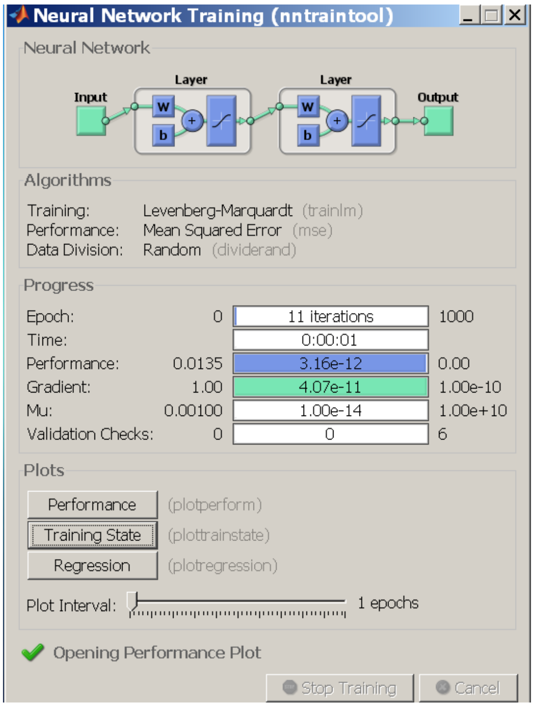

Fig 14: Network training result

In the above figure we have taken 1000 epochs and number of iterations to train the network is 11. The performance in terms of mean squared errors and the value of gradient and validate checks is also shown in the above figure.

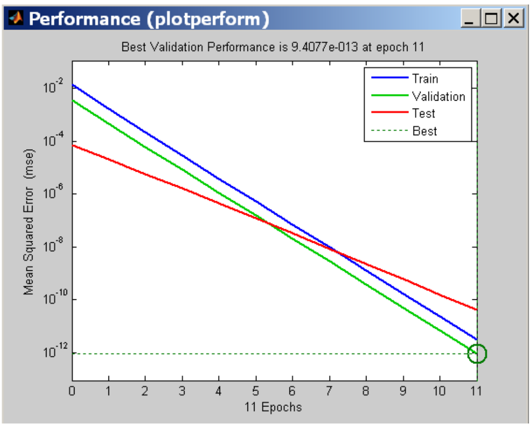

Fig 15: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 9.4077e-013 at 11 epochs.

EXPERIMENT 5: Collecting Output of Each Part and Setting it as Input to Recognize Each Person

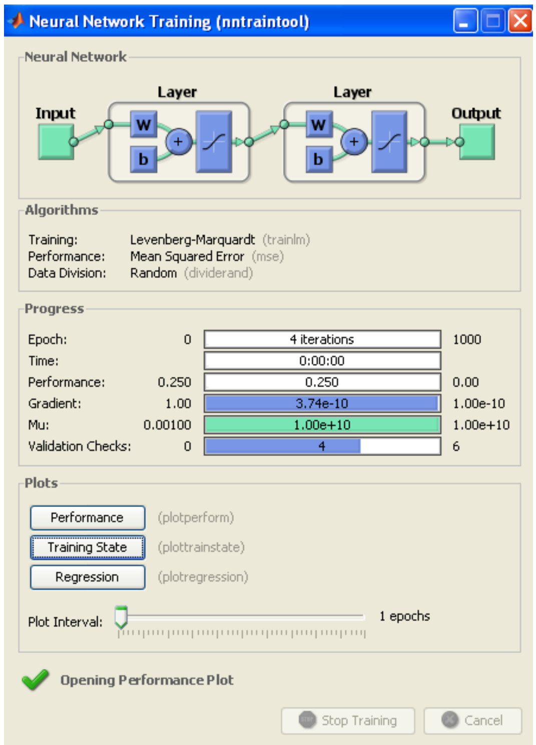

Fig 16: Network training result

From the above figure it seems clear that the total number of iterations to train the network is 4. The performance in terms of mean squared errors and the value of gradient and validate checks is also shown in the above figure.

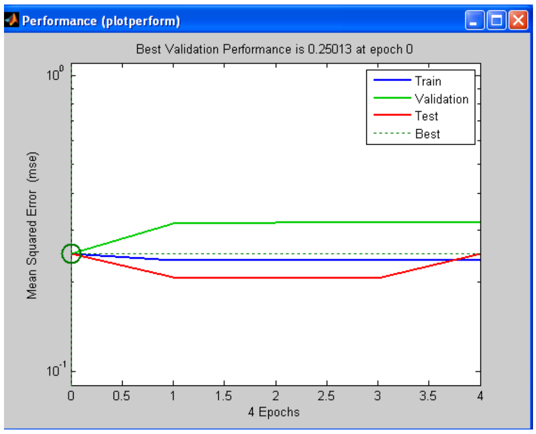

Fig 17: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 0.25013 at 0 epochs

EXPERIMENT 6: First Person Fingerprint Sample without Dividing Into Parts

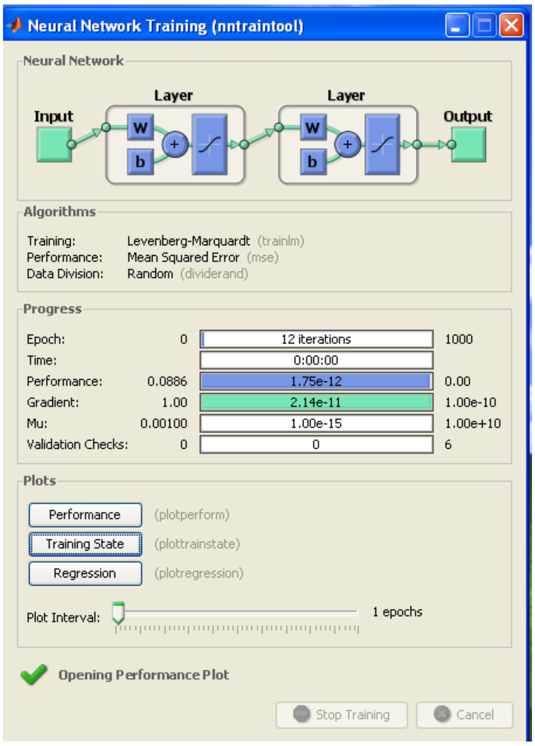

Fig 18: Network training result

From the above figure it seems clear that the total number of iterations to train the network is 12. The performance in terms of mean squared errors and the value of gradient and validate checks is also shown in the above figure.

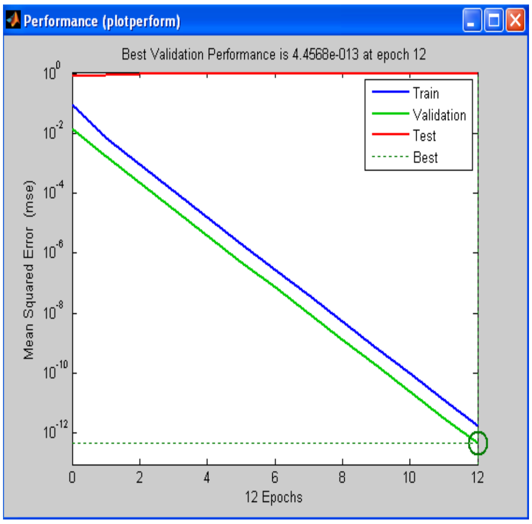

Fig 19: Performance curve produced by the network for the given sample

The above figure shows the performance curve produced while training, testing and validation of the network. We get the Best Validation Performance 4.4568e-013 at 12 epochs

6. CONCLUSIONS

In this work it has been shown that if a fingerprint image of a person is given then the network can recognize the image. The whole work is completed through the following steps:

1. 25 fingerprint images have been used for the experiment. Fingerprint images of five persons have been collected by taking five different samples of each person.
2. Each image is divided into four equal parts and their histogram values are obtained.
3. Feed Forward Back Propagation neural network have been used to train, test and validate the network for each part of the image.
4. Output of all four are collected and set as the new input and a new target is set to train, test and validate the entire parts of each sample.
5. Fingerprint image without dividing into parts have also been applied in the network.

When the fingerprint image was divided into four parts to identify a person then it took 4-epoch while when the image was not decomposed then the average number of iterations were 12. Image division in this work has been done by using a matlab tool. Future work could be to develop a tool to automate some of the processes which have been done manually here.

7. REFERENCES

[1] http://en.Wikipedia.org/wiki/Fingerprint_recognitin.

[2]. Mohamed. S. M and Nyongesa.H, "Automatic Fingerprint Classification System using Fuzzy Neural techniques", IEEE International Conference on Artificial Neural Networks, vol. 1, pp. 358-362, (2002).

[3] Ching-Tang Hsieh and Chia-Shing –u, "*Humanoid Fingerprint Recognition Based on Fuzzy Neural Network*", International Conference on Circuit, Systems, Signal and Telecommunications, pp. 85-90, (2007).

[4]. Haiping Lu, Xudong Jiang and Wei-Yun Yau, "Effective and Efficient Fingerprint Image Post processing", International Conference on Control, Automation, Robotics and Vision, vol. 2, pp. 985- 989, (2002)

[5]. Prabhakar, S, Jain, A.K, Jianguo Wang, Pankanti S, Bolle, "Minutia Verification and Classification for Fingerprint Matching", International Conference on Pattern Recognition vol. 1, pp. 25-29, (2002).

[6] Ballan.M, "*Directional Fingerprint Processing*", International Conference on Signal Processing, vol.2, pp. 1064-1067, (1998).

[7]. G.Sambasiva Rao, C. NagaRaju, L. S. S. Reddy and E. V. Prasad, "A Novel Fingerprints Identification System Based on the Edge Detection", International Journal of Computer Science and Network Security, vol

[8]. M. R. Girgisa, A. A. Sewisyb and R. F. Mansourc, "*Employing Generic Algorithms for Precise Fingerprint Matching Based on Line Extraction",* Graphics, Vision and Image Procession Journal, vol. 7, pp. 51-59, (2007).

[9]. Duresuoquian Miao, Qingshi Tang, and Wenjie Fu, "*Fingerprint Minutiae Extraction Based on Principal*

Cures", the Journal of the Pattern Recognition Letters, vol. 28, pp. 2184 2189, (2007).

[10]. Alessandra Lumini, and Loris Nann, "*Advanced Methods for Two-Class Pattern Recognition Problem Formulation for Minutiae Based Fingerprint Verification",* the Journal of the Pattern Recognition Letters, vol. 29, pp. 142-148, (2008).

[11] Jain Anil, Prabhakar Salil, and Hong Lin, "A Multichannel Approach to Fingerprint Classification,' IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, pp. 348- 359, 1999

[12] Kamijo Masayoshi, "Classifying Fingerprint Images using Neural Network: Deriving the Classification State," IEEE International Conference on Neural network, vol.3, pp. 1932-1937, 1993.

[13] http://en.wikipedia.org/Wiki/Pattern_matching

[14]homepage.inf.ed.ac.uk/rbf/IAPR/researchers/PPRPAGES/pprtut.htm

[15] Bhishop M. Christopher, (2010) "Neural Networks for Pattern Recognition", pp. 1-10