

Face Recognition using Neuro-Fuzzy Inference System

Shweta Mehta¹, Shailender Gupta², Bharat Bhushan³ and C. K. Nagpal⁴

^{1, 2, 3}YMCA University of Science and Technology

⁴Echleon Institute of Technology

¹shwetamehta16@gmail.com, ²shailender81@gmail.com, ³bhrts@yahoo.com,

⁴nagpalckumar@rediffmail.com

Abstract

Face recognition is the process of identifying one or more people in images or videos. It is an important part of biometric, security & surveillance system, and image indexing systems. Various face recognition techniques have been proposed in literature such as: Eigen-faces, Feature based, Hidden Markov model and Neural network based techniques. The first three techniques mostly include a phase of feature extraction or preprocessing closely related to the type of image to recognize. On the other hand Neural network technique does not need specific data about the type of image, thus can be applied to any type of image and at the same time provides better accuracy. In this paper we made an effort to combine neural network technique with fuzzy logic. Our experimental result shows that combining the two provide better accuracy in comparison to other techniques mentioned above.

Keywords: Face Recognition, Neural Network, Fuzzy Logic

1. Introduction

Face recognition [1-4] is a computer application for automatically identifying or verifying a person from a digital database of images. It is an important branch of pattern recognition [5-8] which has many applications like security & surveillance, biometric attendance, image indexing system etc. There are various approaches for face recognition domain such as:

Eigen Faces [17-21] – This approach is a well studied method of face recognition based on Principal Component Analysis (PCA). Although the approach has now largely been superseded, it is still often used as a benchmark to compare the performance of other algorithms. To have better accuracy in this approach, the number of eigenvectors has to be increased. Though this approach provides good accuracy but has very high computational cost that's why this approach is not used.

Feature-based [22-24] - This approach requires points in the face image with high information content. Interesting feature points in the face image are located by using Gabor filter that helps in uniquely identifying faces.

Hidden Markov model [25-27] - This approach uses a scanning strategy to simulate a human-like saccadic sequence. The scanning system scans the face and stores the information of points in a processor on which certain algorithms are applied for face recognition. This method has a disadvantage that it doesn't take important feature parts of the face into account.

Neural Network [28-30] - This approach uses neural network for training of data set. The training data set consists of input signals assigned with corresponding target (desired output). The neural network is then trained using one of the supervised learning algorithms.

On analyzing the above techniques it was found that the neural network techniques have advantages of high speed, better accuracy and low hardware requirements. Therefore in this paper we firstly checked which neural network technique is the best. In addition to it we combined the technique with fuzzy methodology in order to have higher accuracy values. To implement this we used *nntool* and *fuzzy tool* from the MATLAB-11 toolbox. The task of face recognition is conceded on dataset of face images of K people (Where K was varied from 20 to 80 with step size 20). The network was trained using the database and this technique mentioned helped in deciding whether the test image belongs to the database or not on the basis of comparison with higher accuracy value as depicted from our results.

The rest of the paper has been organized as follows: Section 2 provides the literature of the available face recognition techniques. Section 3 provides the information about the setup parameters, metrics and architecture used. Section 4 provides the simulation and results followed by conclusion and references.

2. Face Recognition Techniques

Before discussing our proposal it is required to know about the face recognition technique that have been already been implemented in past:

2.1. Eigenfaces

Eigenfaces [17-21] is a well known method for face recognition that is based on PCA. PCA is a method of transforming a number of correlated variables into a smaller number of uncorrelated variables. Similar to how Fourier analysis is used to decompose a signal into a set of additive orthogonal sinusoids of varying frequencies, PCA decomposes a signal (or image) into a set of additive orthogonal basis vectors or eigenvectors. The main difference is that, while Fourier analysis uses a fixed set of basic functions, the PCA basis vectors are learnt from the data set via unsupervised training. PCA can be applied to the task of face recognition by converting the pixels of an image into a number of eigenface feature vectors, which can then be compared to measure the similarity of two face images. Observing it's result states that the first eigenvector accounts for 50% of the variance in the data set, while the first 20 eigenvectors together account for just over 85%, and the first 30 eigenvectors for 90%. Increasing the number of eigenvectors generally increases recognition accuracy but also increases computational cost. Although the approach has now largely been outdated, it is still often used as a benchmark to compare the performance of other algorithms against, and serves as a good introduction to approach to face recognition.

2.2. Feature-based

Feature based [22-24] system requires points in the face image with high information content. Typically these are located around the eyes, nose and mouth. User is not interested in the face contour or hair, since most stable and informative features in the human face are located in the center of the face. There are several possible ways to locate points with high information content for this the Gabor filter approach is considered. The Gabor filters are generated from a wavelet expansion of a Gabor kernel, parameterized (determining the wavelength and orientation) by the vector. This filter helps in face recognition.

2.3. Hidden Markov Model

Hidden Markov Models (HMM) [25-27] are a class of statistical models used to characterize the observable properties of a signal. HMM consist of two interrelated processes:

1. An underlying, unobservable Markov chain with a finite number of states governed by a state transition probability matrix and an initial state probability distribution, and
2. A set of observations, defined by the observation density functions associated with each state. A discrete-time Hidden Markov Model can be viewed as a Markov model whose states cannot be explicitly observed. Each state has an associated probability distribution function, modeling the probability of emitting symbols from that state.

The main problem of this HMM-based approaches is that the scanning methodology is fixed (typically a raster scan is employed), and the importance (saliency) of facial parts is ignored.

2.4. Neural Techniques

The objective of the neural technique [28-30] is to train the network to recognize face from picture. To teach the neural network we need training data set. The training data set consists of input signals assigned with corresponding target (desired output). The neural network is then trained using one of the supervised learning algorithms, which uses the data to adjust the network's weights and thresholds so as to minimize the error in its predictions on the training set. If the network is properly trained, it has then learned to model the (unknown) function that relates the input variables to the output variables, and can subsequently be used to make predictions where the output is not known. This technique provides the best result for face recognition. To prove our point we have performed experiment on three neural network techniques which areas below:

Of all the above discussed techniques neural techniques have advantages of high speed, better accuracy and low hardware requirements. To prove this we performed an experiment with database composed of 20 images of resolution 180x200. The results for the same are shown below:

Table 1. Comparison in Terms of Accuracy

Techniques	Eigen faces	Feature-Based	Hidden-Markov	Neural Network
Accuracy(%)	85	75	67.1	83.09

From Table 1 it is observed that Eigenfaces technique have better accuracy with over 30 eigen values. Increased number of eigen values improves the accuracy but also increases the computational cost. Whereas due to high speed, low hardware requirement and low cost neural techniques are more advantageous. Thus it can be concluded that neural network techniques are the best for face recognition. Thus we now checked which neural network technique is the best that can be used in our proposal. For this we took three popular techniques as follows:

2.4.1. Back-Propagation Neural Network (BPNN): In this technique the network [1, 9, 10] has a layered structure. The basic layers are input, hidden and output layer. This network is different from others in the way that the weights are calculated during learning. When the numbers of hidden layers are increased, training becomes more complex. There can be more than one hidden layer in the network, but one layer is sufficient to solve our purpose. The training of BPNN is done in three stages:

- Feed-forward of input
- Calculation of weights and error
- Back-propagation of error

Input layer consists of units which receives external input. There are no connections within a layer. The input is fed to the first layer of hidden units. Hidden unit apply activation function and receives weighted bias, the output of the hidden units is distributed over the next layer of hidden units. This process continues until the last layer of hidden units. The outputs are fed into a layer of output units. Though training of BPNN is very slow, once the network is trained it produce results rapidly.

2.4.2. Recurrent Neural Network: RNN [11-13] is a neural network technique where connections between units form a directed cycle. A simple recurrent network must have an active feedback which has short-term memory. Layer Recurrent network have number of hidden layers. Each hidden layer is updated not only with the external input of the network but also with activation from the previous forward propagation. To form RNN a context layer is added to the structure to retain information. Each time new input pattern is fed to the RNN, the previous contents of the hidden layer are stored in the context layer and provided as feedback to the hidden layer. The feedback also has the option to modify set of weights. RNN is a time based neural network which accepts the input at each time step. Here the activation function used to compute output is purely non-linear. The feedback layer gives a direct impact on the output as the error once introduced in the network affects the final outcome through feedback path.

2.4.3. Cascade Forward Neural Network: CNN [13-15] is also having layered structure in which each hidden layer includes a connection from the input layer and all previous layers. The network grows on demand, new neurons in the hidden layer are added and trained one by one. Training of the network starts with only input and output layers and does not have any hidden layer neurons. During the training process new neurons are added to network and trained. This newly added neuron is called as candidate neuron which has connection to all input and hidden layer neurons. Weight adjustment can be performed only with the candidate neuron while all the other weights in the network are fixed. CNN starts with one input node and then keep on adding new input and hidden neurons. The trained CNN has minimum number of input, hidden neurons and connections.

Table 2. Comparison of Neural Techniques

	BPNN	CNN	RNN
Time Taken (s)	2.56	11.48	3.20
Accuracy (%)	83.09	65.2	62

Now our aim was to test which neural network gives best accuracy. For this we performed an experiment in which we calculated accuracy and time complexity. The results are shown in the Table 2. On the basis of the above overall comparison Table 2, the following points can be inferred as follows:

- The Back-propagation technique is best in terms of accuracy. The accuracy obtained in our experiment of this technique is nearly 83.09%.
- The elapsed time taken by BPNN is also very low (2.56s).Thus using this technique makes pattern recognition faster.

- The order of neural network techniques in terms of accuracy is BPNN, CNN and RNN while in terms of time elapsed is BPNN, RNN and then CNN.

To improve further accuracy we combined neural network techniques with the fuzzy inference system the results are observed to be much better and accurate.

3. Proposal

At present many methods for image recognition are available but most of them include feature extraction or processing related to type of image. The method proposed in this paper can be applied to any type of images. The proposal is divided into two phases: the training phase and the recognition phase.

3.1. Training of the Neural Network:

The training of neural network consists of following steps as shown in Figure 1.

- The first step for training is to provide the network with data set. For this purpose identical row from the image matrix is considered as input for designing the structure

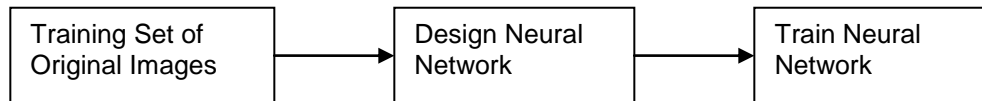


Figure 1. Training of Neural Network

The dataset obtained from step 1 is now used to design the neural network architecture as shown in Figure 2. The network designed has number of input equals to number of columns in the dataset matrix. Here, BPNN is working in feed forward mode. This network [1, 7, 8] has a layered structure. The basic layers are input, hidden and output layer. This is different from others based on the way the weights are calculated during learning. When the numbers of hidden layers are increased, training becomes more complex. There can be more than one hidden layer in the network, but one layer is sufficient to solve our purpose. The training of BPNN is done in three stages:

- Feed-forward of input
- Calculation of weights and error
- Back-propagation of error

Input layer consists of units which receives external input. There are no connections within a layer. This input is fed to the first layer of hidden units. Hidden unit apply activation function and receives weighted bias, the output of the hidden units is distributed over the next layer of hidden units. This process continues until the last layer of hidden units. The outputs are fed into a layer of output units. Though training of BPNN is very slow, once the network is trained it produce results rapidly.

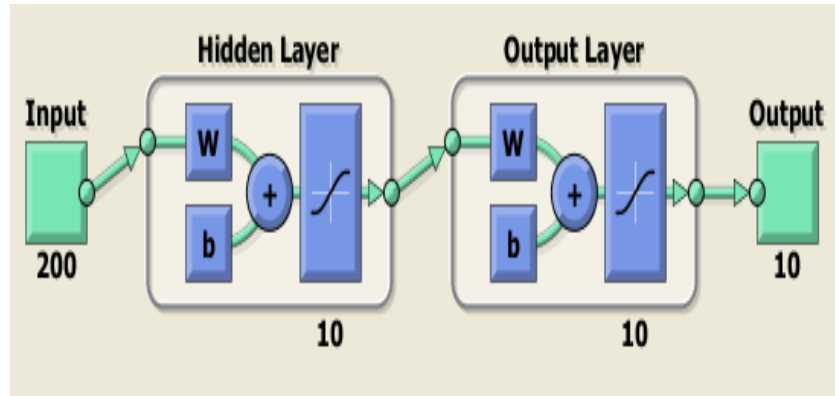


Figure 2. Designed Neural Network

In the final step training of neural network is performed, after the database is created target is set corresponding to the images we want to store in our database. The above designed network is then trained such that output of the network well matches with the target values. Much closer the trained value, accuracy comes out to be more. Thus, for this purpose network can be trained up to 4 to 5 times. This step completes the first phase.

3.2. Recognition of the data using Fuzzy Logic:

Accuracy of the neural network is calculated on the two parameters namely

- **Epochs-** Fewer epochs mean network learns in small repetitions. Less time means network achieved goal easily and shortly. Lower value of epochs is associated with higher network accuracy.
- **Gradient-** Low value of gradient plot indicates that the network is learning up to a large extent which means finer adjustments in the weights and bias. This in turn makes network more accurate and reliable, avoiding chances of false predictions.

Figure 3 depicts two output parameters of neural network on basis of which accuracy is calculated is given to Fuzzy Inference System as inputs. The two input blocks in the figure shows the same.

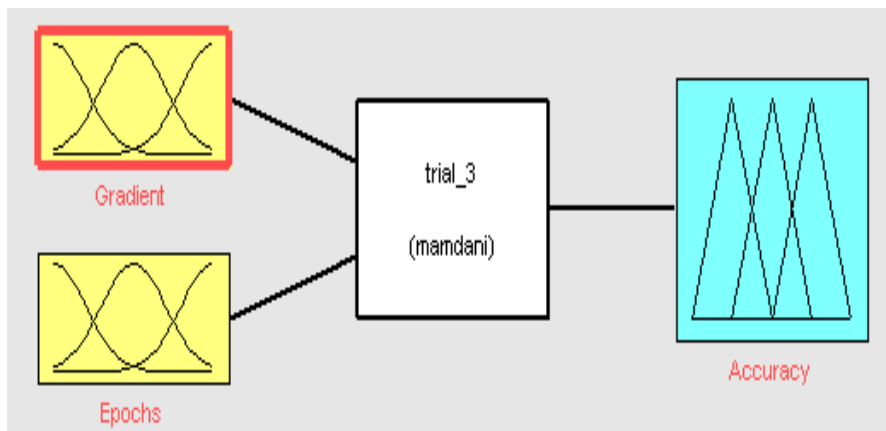


Figure 3. Designed FIS

The input to the FIS is defined by the membership function which is defined between the ranges in which the two input values lies. Here range is selected as 0 to 50. We can select from any membership function which are already defined or customize our own membership.

Figure 4 shows the membership function selected for inputs Gradient & Epochs.

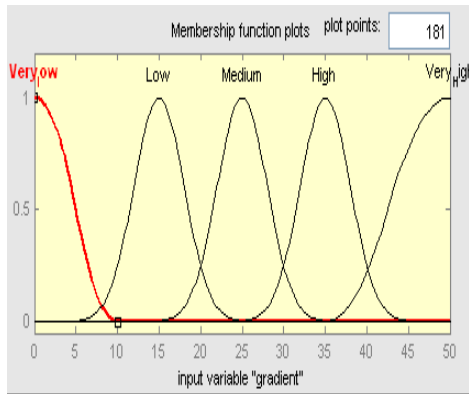


Figure 4. (a) Membership Function for Input Gradient

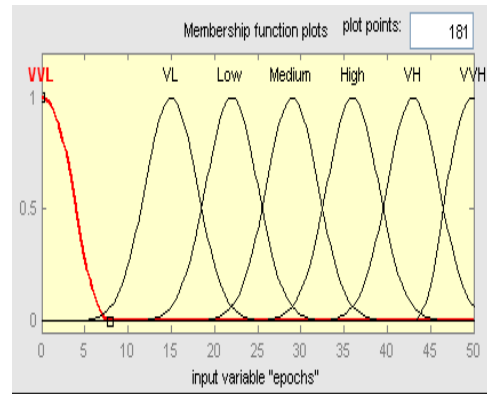


Figure 4. (b) Membership Function for Input Epochs

Figure 4. Membership Function

The central block consists of the rules for the network which are based on the combination of two input parameters to provide the desired output. The fuzzy rules are redundant for each input, we must repeat the set of rules for input parameter. The following rules were used in our experiment.

1. If (gradient is Very_low) and (epochs is VVL) then (Accuracy is known) (1)
2. If (gradient is Very_low) and (epochs is VL) then (Accuracy is known) (1)
3. If (gradient is Very_low) and (epochs is Low) then (Accuracy is known) (1)
4. If (gradient is Very_low) and (epochs is Medium) then (Accuracy is known) (1)
5. If (gradient is Very_low) and (epochs is High) then (Accuracy is known) (1)
6. If (gradient is Very_low) and (epochs is VH) then (Accuracy is >Identical) (1)
7. If (gradient is Very_low) and (epochs is VVH) then (Accuracy is >Identical) (1)
8. If (gradient is Low) and (epochs is VVL) then (Accuracy is known) (1)
9. If (gradient is Low) and (epochs is VL) then (Accuracy is known) (1)
10. If (gradient is Low) and (epochs is Low) then (Accuracy is known) (1)
11. If (gradient is Low) and (epochs is Medium) then (Accuracy is >Identical) (1)
12. If (gradient is Low) and (epochs is High) then (Accuracy is >Identical) (1)
13. If (gradient is Low) and (epochs is VH) then (Accuracy is Identical) (1)
14. If (gradient is Low) and (epochs is VVH) then (Accuracy is <Identical) (1)
15. If (gradient is Medium) and (epochs is VVL) then (Accuracy is known) (1)
16. If (gradient is Medium) and (epochs is VL) then (Accuracy is known) (1)

17. If (gradient is Medium) and (epochs is Low) then (Accuracy is known) (1)
18. If (gradient is Medium) and (epochs is Medium) then (Accuracy is Identical) (1)
19. If (gradient is Medium) and (epochs is High) then (Accuracy is <Identical) (1)
20. If (gradient is Medium) and (epochs is VH) then (Accuracy is <Identical) (1)
21. If (gradient is Medium) and (epochs is VVH) then (Accuracy is <Identical) (1)
22. If (gradient is High) and (epochs is VVL) then (Accuracy is known) (1)
23. If (gradient is High) and (epochs is VL) then (Accuracy is >Identical) (1)
24. If (gradient is High) and (epochs is Low) then (Accuracy is >Identical) (1)
25. If (gradient is High) and (epochs is Medium) then (Accuracy is Identical) (1)
26. If (gradient is High) and (epochs is High) then (Accuracy is <Identical) (1)
27. If (gradient is High) and (epochs is VH) then (Accuracy is <Identical) (1)
28. If (gradient is High) and (epochs is VVH) then (Accuracy is unknown) (1)
29. If (gradient is Very_High) and (epochs is VVL) then (Accuracy is >Identical) (1)
30. If (gradient is Very_High) and (epochs is VL) then (Accuracy is >Identical) (1)
31. If (gradient is Very_High) and (epochs is Low) then (Accuracy is >Identical) (1)
32. If (gradient is Very_High) and (epochs is Medium) then (Accuracy is Identical) (1)
33. If (gradient is Very_High) and (epochs is High) then (Accuracy is <Identical) (1)
34. If (gradient is Very_High) and (epochs is VH) then (Accuracy is unknown) (1)
35. If (gradient is Very_High) and (epochs is VVH) then (Accuracy is unknown) (1)

The third and final block in recognition phase is the output membership function which is defined in the range of recognition rate from [0 100]. When the two inputs are low in range accuracy is calculated to be more as defined by the rules. Figure 5 shows the output membership.

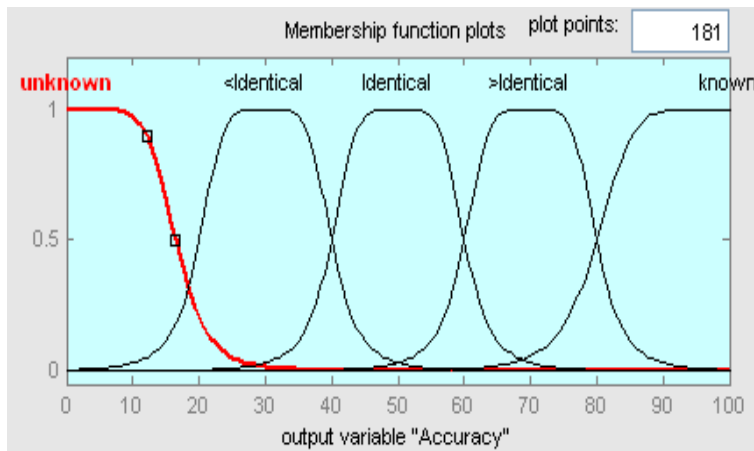


Figure 5. Membership Function for Output Variable “Accuracy”

The combined algorithm of both the phases is shown in Figure 6:

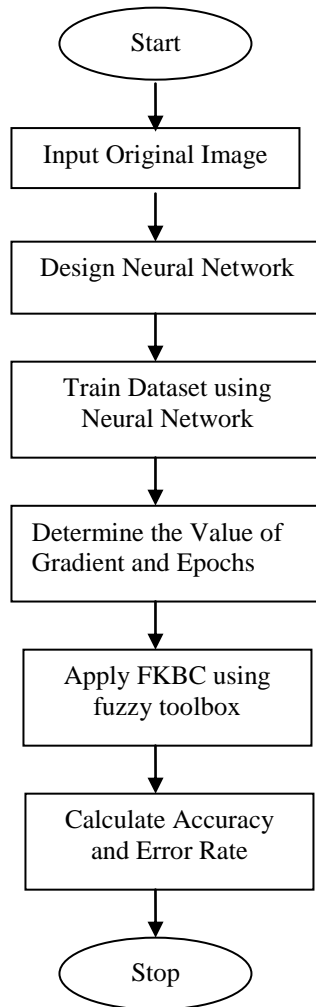


Figure 6. Algorithm for the Proposed Method

4. Simulation Set up Parameters

4.1. Designed Architecture

After the compilation of the two phases designed network is shown in Figure 7. The figure shows the combination of neural network and the fuzzy system. The first two blocks in the figure shows the input to the network. Membership function is applied to the inputs. Rules are implemented on the basis of defined input and output membership function. The final output block gives the accuracy of the complete network and is the final output of the combined networks.

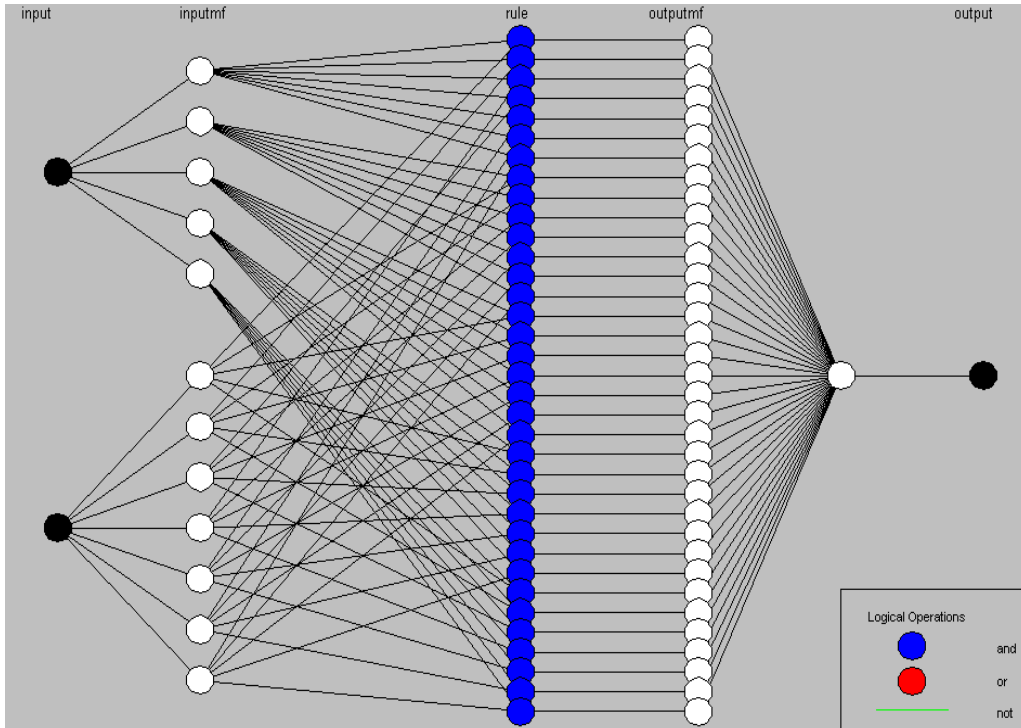


Figure 7. Designed Architecture of Neuro-Fuzzy Network

Output of the above network is studied with the help of Figure 8. The figure represents the rule base and the corresponding inputs and output. The use of fuzzy inference makes the system more flexible in a way that it accepts all the values between specified range.

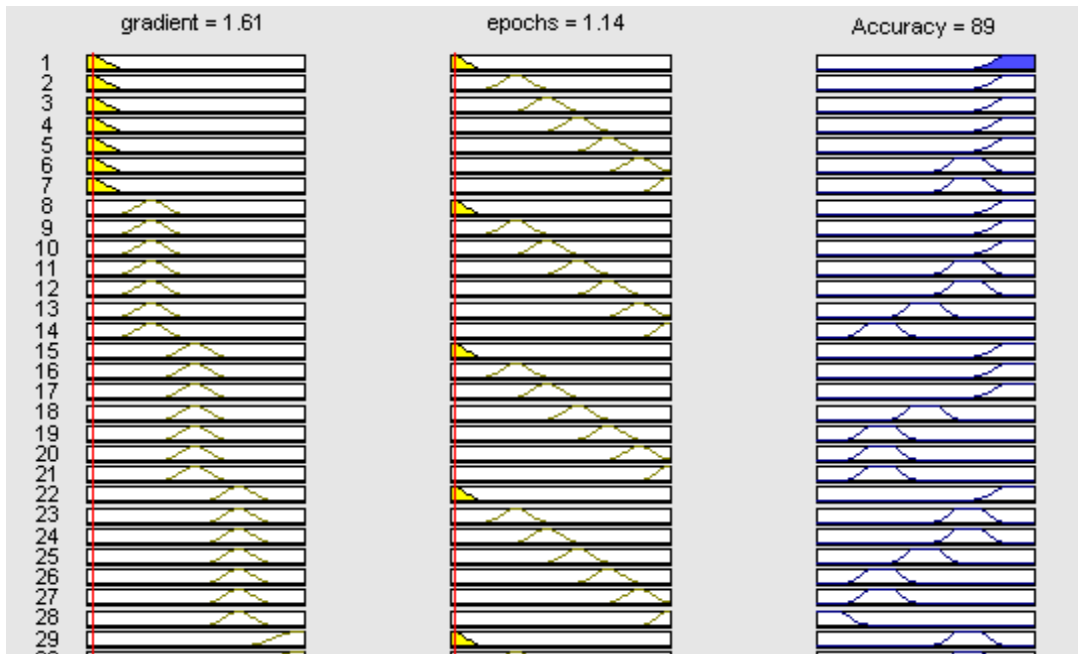


Figure 8. Rules for the Fuzzy System

4.2. Performance Metrics

The performance of the system is analyzed on the basis of two metrics:

Accuracy- The ability to match the measured value of quantity to actual value is termed as accuracy. Accuracy here is measured by Epochs and Gradient value, fewer epochs mean network learns in small repetitions and smaller gradient means the fine adjustment of weights and bias which avoid false prediction.

Error Rate- Error rate is a term that describes the degree of errors encountered during data transmission or network connection. The higher the Error Rate the less reliable the system will be.

4.3. Setup Parameters

For the conclusion we have performed the tests with database composed of 20, 40, 60 and 80 images of resolution 180x200. These are images of 10 persons captured at different time with varying expressions and lightening variations.

Table 3 shows the various simulation set up parameters used in our experiment

Table 3. Setup Parameters

Total number of images	20, 40, 60, 80
Image of each person	2, 4, 6, 8
Image resolution	180x200
Gender	Male(5), Female(5)
Background	Plain Green
Lighting variation	Yes
Expression variation	Yes

5. Simulation Results

5.1. Impact on Accuracy

Figure 9(a) and Figure 9(b) show the accuracy of Neural Network techniques and the proposed technique. The following inference can be drawn:

- BPNN technique is the best in terms of accuracy followed by CNN and RNN.
- It can be easily seen that when neural network techniques are combined with Fuzzy technique, the result of accuracy improves.
- As the number of images is increased, the accuracy of Neural Network techniques improves but it is nearly same when NN technique is combined with Fuzzy.

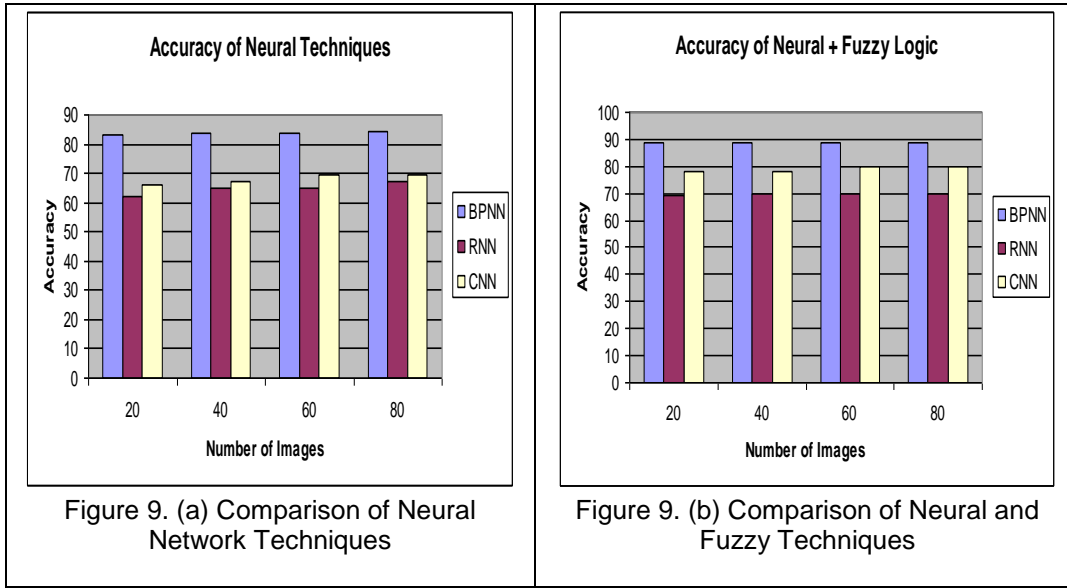


Figure 9. Accuracy Plots for Neural Techniques and Neuro-Fuzzy Techniques

5.2. Impact on Error Rate

Error rate for neural techniques and Neural combined with Fuzzy Logic is shown in Table 4: The table shows the hybrid technique *i.e.*, combination of fuzzy with neural network techniques has comparatively lesser error then neural network technique alone.

Table 4. Error Rate

Neural Technique	16	33	31
Neural + Fuzzy Logic	11	30	20

Table 5 shows the overall comparison of the proposed techniques with the techniques available in literature. The following inferences can be drawn:

- The neural network when combined with Fuzzy has higher accuracy than any other technique given in the literature.
- The Eigenface technique provides good accuracy with an additional burden of computational cost.

Table 5. Comparison with other Techniques in Literature

Techniques	Eigen faces	Feature-Based	Hidden-Markov	Neural Network	Neuro-Fuzzy
Accuracy(%)	85	75	67.1	83.09	89

6. Conclusion

In this paper an effort has been made to improve the existing face recognition techniques on the basis of recognition rate by implementing fuzzy logic with the best performing technique *i.e.*, neural network technique. We can conclude that if the best performing technique is employed with fuzzy logic can give far better results. In this paper we consider

two parts of the network one is neural network for training, second is fuzzy inference system which helps us improve the performance result in face recognition. Fuzzy logic has proved to be a tool that can improve the performance of the existing system.

References

- [1] T. Ahmad, A. Jameel and Dr. Balal Ahmad, "Pattern Recognition Using Statistical and Neural Techniques", published in International Conference organized by IEEE, (2011), pp. 87-91.
- [2] X. Guo, X. Liang and X. Li, "A Stock Pattern Recognition Algorithm Based on Neural Networks", published in Third International Conference on Natural Computation, organized by IEEE, vol. 2, (2007), pp. 518-522.
- [3] S. Azuan Nazeer, N. Omar, K. Faisal Jumari and M. Khalid, "Face detecting using Artificial Neural Networks Approach", published in First Asia International Conference on Modelling & Simulation, IEEE, (2007).
- [4] Y.-M. Lui, D. Bolme, J. Phillips, R. Beveridge and B. Draper, "Preliminary Studies on The Good, The Bad and The Ugly Face Recognition Challenge Problem", Computer Vision and Pattern Recognition Workshops, IEEE, (2012).
- [5] S. Ghorpade, J. Ghorpade and S. Mantri, "Pattern Recognition using Neural Networks", published in International Journal of computer science and Information Technology IJCSIT organized by IEEE, vol. 2, (2010), pp. 92-98.
- [6] Y. Perwej and A. Chaturvedi, "Machine Recognition of Hand Written Characters using Neural Networks", Published in International Journal of Computer Applications, IJCA, vol. 14, no. 2, (2011), pp. 6-9.
- [7] J. Urias, P. Melin and O. Castillo, "A Method for Response Integration in Modular Neural Networks using Interval Type-2 Fuzzy Logic", published in Fuzzy Systems Conference organized by IEEE, IEEE, (2007), pp. 1-6.
- [8] S. O'Hara and B. Draper, "Scalable Action Recognition with a Subspace Forest", published in Conference on Computer Vision and Pattern Recognition, IEEE, (2012).
- [9] R. Polikar, L. Udpa, S. S. Udpa and V. Honavar, "An incremental learning algorithm for supervised neural networks", published in IEEE International Conference organized by IEEE, vol. 6, (2000), pp. 3414-3417.
- [10] Y. Shi, G.-S. Gai, X.-T. Zhao, J.-J. Zhu and P. Zhang, "Back Propagation Neural Network Simulation Model", published in International Conference organized by IEEE, (2010), pp. 1-4.
- [11] F. A. Gers and J. Schmidhuber, "LSTM Recurrent Networks Learn Simple Context Free and Context Sensitive Languages", published in International Conference organized by IEEE, (2000), pp. 1-14.
- [12] A. Graves, S. Fernandez and M. Liwicki, "Advances in Neural Information Processing Systems", published in Conference organized by MIT, MIT, (2009), pp. 545-552.
- [13] L. S. Liu and X. F. Peng, "Diagonal Recurrent Neural Network with output feedback and its application", published in 6th International Conference ICCSE organized by IEEE, (2011), pp. 286-288.
- [14] A. Graves and J. Schmidhuber, "Frame-wise Phoneme Classification with Bidirectional LSTM and other Neural Network Architectures", published in International Symposium on Neural Networks organized by IEEE, (2005).
- [15] J. M. Zurada and J. Books, "Introduction to Artificial Neural Systems".
- [16] URL: <http://cswww.essex.ac.uk/mv/allfaceS/faces94.html>.
- [17] M. Turk and A. Pentland, "Eigenfaces for Face Recognition", published in Journal of Cognitive Neuroscience, vol. 3, no. 1, (1991), pp. 71-86.
- [18] J. Ruiz-del-Solar and J. Quinteros. "Illumination compensation and normalization in eigenspace-based face recognition: A comparative study of different pre-processing approaches", published in Elsevier Science Inc., vol. 29, no. 14, (2008), pp. 1966-1979.
- [19] J. Ruiz-del-Solar and P. Navarrete, "Eigenspace-based face recognition: a comparative study of different approaches", Published in Man and Cybernetics, Part C, IEEE, vol. 35, no. 3, (2005), pp. 315-325.
- [20] P. N. Belhumeur, J. P. Hespanha and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: Recognition using Class Specific Linear Projection", published in Proc. of the 4th European Conference on Computer Vision, ECCV, (1996), pp. 45-58.
- [21] K. Delac, M. Grgic and S. Grgic, "Statistics in Face Recognition: Analyzing Probability Distributions of PCA, ICA and LDA Performance Results", published in Proceedings of the 4th International Symposium on Image and Signal Processing and Analysis, ISPA, (2005), pp. 289-294.
- [22] B. Draper, K. Baek, M. S. Bartlett and J. R. Beveridge, "Recognizing Faces with PCA and ICA, Computer Vision and Image Understanding", published, IEEE, vol. 91, no. 1-2, (2003), pp. 115-137.
- [23] H. Moon and P. J. Phillips, "Computational and Performance aspects of PCA-based Face Recognition Algorithms", published, IEEE, vol. 30, (2011), pp. 303-321.

- [24] J. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "Face Recognition Using LDA-Based Algorithms", published in IEEE Trans. on Neural Networks, IEEE, vol. 14, no. 1, **(2003)**, pp. 195-200.
- [25] A. V. Nefian and M. H. Hayes III, "Hidden Markov Models for Face Recognition", published in Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP, vol. 5, **(1998)**, pp. 2721-2724.
- [26] A. V. Nefian and M. H. Hayes, "Maximum likelihood training of the embedded HMM for face detection and recognition", published in Proc. of the IEEE International Conference on Image Processing, ICIP, vol. 1, **(2000)**, pp. 33-36.
- [27] K. Delac, M. Grgic and S. Grgic, "Generalization Abilities of Appearance-Based Subspace Face Recognition Algorithms", published in Proceedings of the 12th International Workshop on Systems, Signals and Image Processing, IWSSIP, **(2005)**, pp. 273-276.
- [28] A. K. Jain, R. P. W. Duin and J. Mao, "Statistical Pattern Recognition", published in IEEE Transactions on Pattern Analysis and Machine Intelligence, IEEE, vol. 22, no. 1, **(2000)**, pp. 4-37.
- [29] L. Sirovich and M. Meytlis, "Symmetry, Probability, and Recognition in Face Space", published in Proceedings of the National Academy of Sciences, PNAS, vol. 106, no. 17, **(2009)**, pp. 6895-6899.
- [30] P. Sinha, B. Balas, Y. Ostrovsky and R. Russell, "Face Recognition by Humans", published in Proceedings of the IEEE, vol. 94, no. 11, **(2006)**, pp. 1948-1962.