ISSN 2047-3338

Face Recognition Using Neural Network

Deepshikha Arora[1] and Darshna Kundu[2]
[1,2]MRIU, Faridabad, India

***Abstract*— In this article, we are going to do the face recognition, using the neural network concept. In this whole process of face recognition, canny edge detection operator as well as improved canny edge detection has been used to detect a wide range of edges in image. Gaussian filter is also used to smooth the image in order to remove the noise. At last neural network toolbox software uses the network object to store all the information that defines a neural network.**

***Index Terms*— Neural Network, Improved Canny Edge Detection Technique and Gaussian Filters**

I. INTRODUCTION

IN recent years many sensing devices, computational powers and intelligent papers have been developed in the field of image processing. In last 20 years, machine recognition of faces is becoming a growing interest. Face recognition consist of neural network design process. In which data is being collected. Then creation and configuration of network is done. After network configuration, the adjustable network parameters (called weights and biases) need to be tuned, so that the network performance is optimized. This tuning process is referred to as training the network. The validation of network is done. In all these steps, canny edge detector has its major role. It uses a multi stage algorithm to detect range of edges in images and Gaussian filter is used to filter out the noise.

A) Neural Network

Generally systems of interconnected neurons which can compute values from inputs, and are capable of machine learning as well as pattern recognition because of their adaptive nature are known as neural network. Like other machine learning methods, neural networks have been used to solve a wide variety of tasks that are hard to solve using ordinary rule-based programming, including computer vision and speech recognition as well as face recognition.

B) Canny Edge Detection

To detect a wide range of edges in images, the canny edge detector is being used. It is an edge detection operator that uses a multi-stage algorithm. It was developed by John F. Canny in 1986. Canny also produced a computational theory of edge detection explaining why the technique works.

Edge detection, especially step edge detection has been widely applied in various different computer vision systems, which is an important technique to extract useful structural information from different vision objects and dramatically reduce the amount of data to be processed. Canny has found that, the requirements for the application of edge detection on diverse vision systems are relatively the same. Thus, a development of an edge detection solution to address these requirements can be implemented in a wide range of situations.

C) Improved Canny Edge Detection

While traditional canny edge detection provides relatively simple but precise methodology for edge detection problem, with the more demanding requirements on the accuracy and robustness on the detection, the traditional algorithm can no longer handle the challenging edge detection task.

II. METHODOLOGY

The work flow for the neural network design process has seven primary steps:

1) Collect data
2) Create the network
3) Configure the network
4) Initialize the weights and biases
5) Train the network
6) Validate the network
7) Use the network

This topic discusses the basic ideas behind steps 2, 3, 5, and 7. The details of these steps come in later topics, as do discussions of steps 4 and 6, since the fine points are specific to the type of network that you are using. (Data collection in step 1 generally occurs outside the framework of Neural Network Toolbox™ software, but it is discussed in Multilayer Neural Networks and Back propagation Training.).

The Neural Network Toolbox software uses the network object to store all of the information that defines a neural network. This topic describes the basic components of a neural network and shows how they are created and stored in the network object.

After a neural network has been created, it needs to be configured and then trained. Configuration involves arranging the network so that it is compatible with the problem you want to solve, as defined by sample data. After the network has been configured, the adjustable network parameters (called weights and biases) need to be tuned, so that the network performance is optimized. This tuning process is referred to as training the network. Configuration and training require that the network be provided with example data. This topic shows how to format the data for presentation to the network. It also explains network configuration and the two forms of network training: incremental training and batch training.

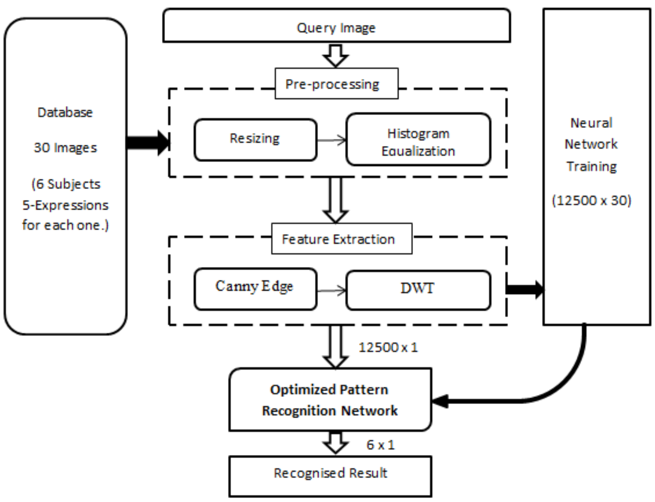

Fig. 1: Block diagram of face recognition process

III. RESULTS

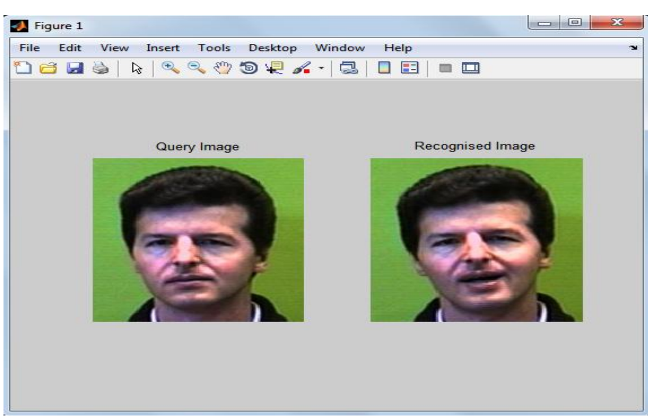

Fig. 2: Recognized Image

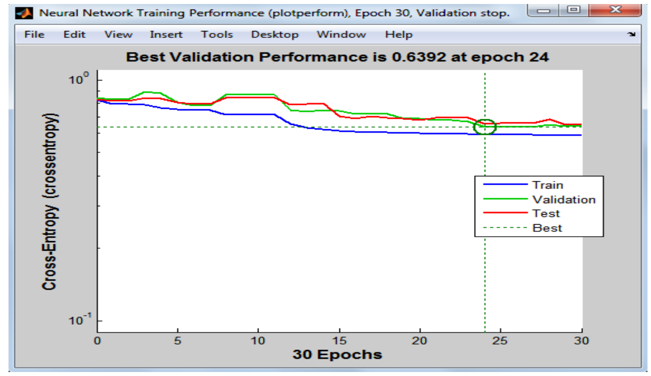

Fig. 3: Training performance of neural network

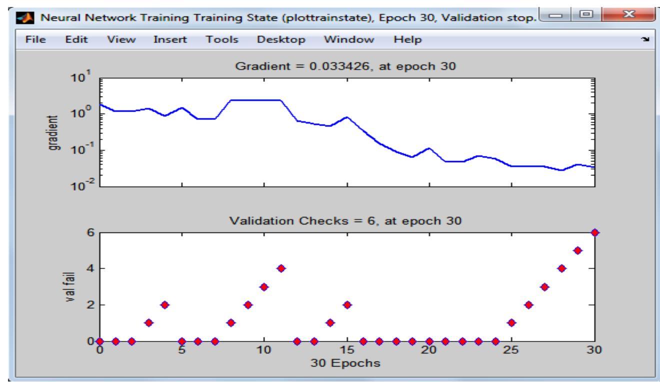

Fig. 4: Training State of Neural network

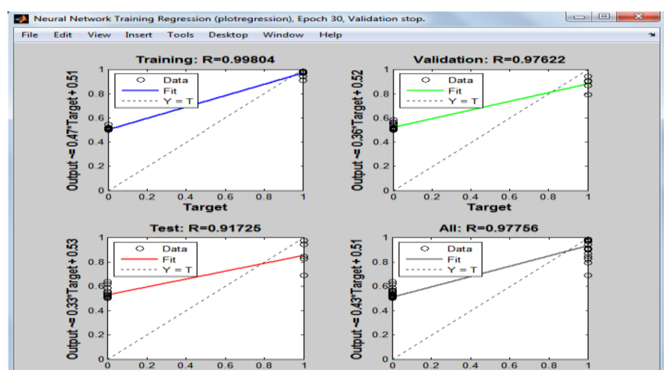

Fig. 5: Regression plot

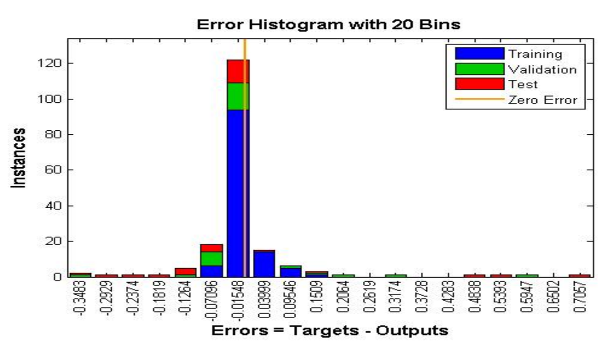

Fig. 6: Error histogram

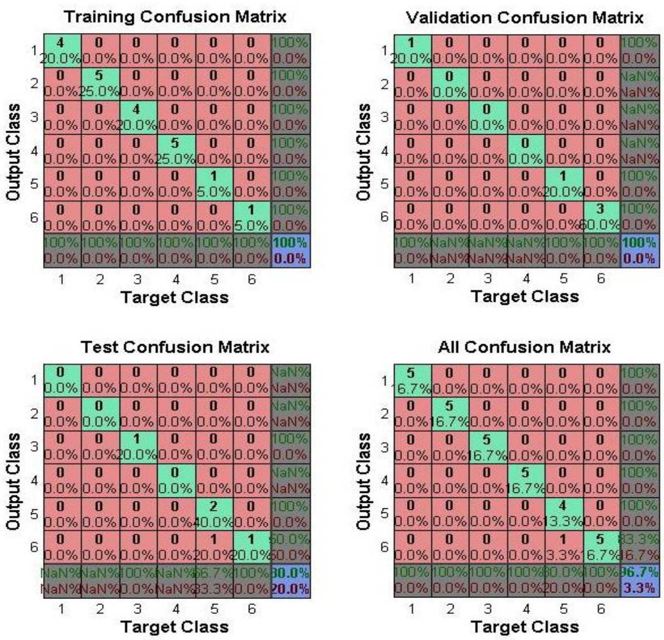

Fig. 7: Confusion matrix

12 Subjects 6 Groups

Sr. No.	Query	Results G1	G2	G3	G4	G5	G6	Decision Group	Classification
1	Group-1 Subject-1	0.742151	0.001358	0.180043	0.014674	0.058102	0.003672	G1	TRUE
2	Group-1 Subject-2	0.89662	0.000277	0.042924	0.018578	0.039872	0.001729	G1	TRUE
4	Group-2 Subject-2	0.000487	0.978807	0.003887	0.010513	0.002091	0.004215	G2	TRUE
5	Group-3 Subject-1	0.020054	0.000823	0.961741	0.001419	0.007133	0.008829	G3	TRUE
6	Group-3 Subject-2	0.019806	0.000823	0.962236	0.0014	0.007064	0.00867	G3	TRUE
7	Group-4 Subject-1	0.010171	0.004627	0.004892	0.972271	0.00499	0.003049	G4	TRUE
8	Group-4 Subject-2	0.007744	0.008536	0.004418	0.970618	0.004806	0.003878	G4	TRUE
9	Group-5 Subject-1	0.045099	0.008921	0.003749	0.039886	0.896414	0.005931	G5	TRUE
10	Group-5 Subject-2	0.088554	0.003342	0.010749	0.03293	0.861875	0.002549	G5	TRUE
11	Group-6 Subject-1	0.051171	0.032627	0.087496	0.017345	0.041169	0.770192	G6	TRUE
12	Group-6 Subject-2	0.107553	0.005585	0.310888	0.096955	0.061914	0.417105	G6	TRUE

Fig. 8: Results in tabular form

IV. CONCLUSION

The Canny algorithm is adaptable to various environments. Its parameters allow it to be tailored to recognition of edges of differing characteristics depending on the particular requirements of a given implementation.

In Canny's original paper, the derivation of the optimal filter led to a Finite Impulse Response filter, which can be slow to compute in the spatial domain if the amount of smoothing required is important (the filter will have a large spatial support in that case). For this reason, it is often suggested to use Rachid Deriche's infinite impulse response form of Canny's filter (the Canny–Deriche detector), which is recursive, and which can be computed in a short, fixed amount of time for any desired amount of smoothing. The time taken to train the network is 0.02 sec.

REFERENCES

[1]. Mahendra Pratap Panigrahy and Neeraj Kumar: Face Recognition using Genetic Algorithm and Neural Networks, International Journal of Computer Applications (0975 – 8887) Volume 55, No. 4, October 2012.

[2]. Pankaj Bhandari, Pankaj K Gupta, Karthik U.S Goutham Reddy, Jeeva.B: Analysis of Face Recognition Based On Edge Detection Algorithm with Hardware Interfacing, Vol. 3, Special Issue 3, April 2014.

[3]. Prof K.H. Wanjale, Saurabh Padmane, Prashant Sethi, Pratik Shah, Sagar Thakur: Face Recognition in Video-A Survey, International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS).

[4]. Henry A. Rowley, Shumeet Baluja, Takeo Kanade: Neural Network Based Face Detection, 1996.

[5]. S.Venkatesan and M.Karnan: Advanced Classification using Genetic Algorithm and Image Segmentation for Improved Face Detection., computer research and Development 2010 second International Conference (ICCRD), 7-10 May 2010, pp. 364-368.

[6]. Gur, E., Zalevsky, Z., 2007, Single-Image Digital Super-Resolution A Revised Gerchberg Papoulis Algorithm, IAENG international Journal of Computer Science, 34:2, IJCS_34_2_14.

[7]. Y. Suzuki, H. Saito, D. Ozawa, Extraction of the human face from natural background using GAs, Proceedings of the IEEE TENCON, Digital Signal Processing Applications, Vol. 1, 1996, pp. 221-226.

[8]. A.M. Mohamed, A. Elgammal, Face detection in complex environments from color images, Proceedings of International Conference on Image Processing 3 (1999), pp. 622-626.

[9]. Brunelli, R. and Poggio, T., "Face recognition: features versus templates," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 10, pp. 1042 -1052, 1993.

[10]. Chellappa, R., Wilson, C.L. and Sirohey, S., "Human and machine recognition of faces: a survey," Proceedings of the IEEE, Vol. 83, No. 5, pp. 705 -741, 1995.

[11]. A.Samal and P.A.Iyengar (1992): -Automatic recognition and analysis of human faces and facial expressions: A survey. Pattern Recognition.

[12]. K. Okamoto, S. Ozawa, and S. Abe. A Fast Incremental Learning Algorithm of RBF Networks with Long-Term Memory. Proc. Int. Joint Conf. on Neural Networks, 102-107, 2003.

[13]. M.A.Turk and A.P.Petland, (1991) "Eigenfaces for Recognition," Journal of Cognitive Neuroscience. vol. 3, pp.71-86.

Deepshikha Arora is a M.Tech student of MRIU, Faridabad. She is perusing masters in communication engineering. She has done her B.Tech in field of electronics & communication engineering from HCTM, Haryana. She has done her research work in field of neural networks. She has selected her topic for research is face recognition.