I.J. Image, Graphics and Signal Processing, **2013, 9, 21-28**
Published Online July 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2013.09.04

A Hybrid Method for Detection of Edges in Grayscale Images

Jesal Vasavada
Deptt. Of Computer Sc. & Engineering
Faculty of Engineering & Tech., MITS Laxmangarh, (India)
jesal.vasavada@gmail.com

Shamik Tiwari
Deptt. Of Computer Sc. & Engineering
Faculty of Engineering & Tech., MITS Laxmangarh, (India)
shamik_tiwari@rediffmail.com

Abstract — *E*dge detection is the most fundamental but at the same time most important task in image processing and analysis. In the paper a hybrid approach combining Neural Network and Fuzzy logic based edge detection algorithm is proposed to detect edges in grayscale images. To improve the generalization ability, the neural network is trained on fuzzy inputs rather than crisp inputs. The network consists of three layers, one input layer, one hidden layer and one output layer. Fuzzy membership functions are used to convert neurons of input and hidden layer into fuzzy neurons. So the output of first and second layer is the membership value of the corresponding input in the fuzzy set. The proposed technique provides advantage of both neural networks and fuzzy logic and gives satisfactory results for both noisy and noise free images. The method is compared with Roberts, Prewitt, Sobel and Laplacian of Gaussian and other neural network and fuzzy logic based methods and the experimental results reveal that proposed method gives better edge map considering the problem of false edge detection.

Index Terms — Edge Detection, Neural Networks, Fuzzy logic, Backpropagation, hybrid system

I. INTRODUCTION

Edge detection is an important but difficult task in image processing and analysis. It is important because it provides basic structural properties about object like shape, perimeter, area etc. It reduces the less relevant information which reduces the amount of data to be processed and thus saves time and at the same time preserves the most important features. Edge detection is a pre-processing step to extract some low level boundary features of an image, which are then used for higher level processing such as object finding and recognition.

Edge detection is a difficult task as all types of edges, step edge, etc must be identified between the object and the background and the detected edges should not be blurred ones. It becomes more difficult task in case of noisy images. Noisy images and edges both contain high frequency content so detection of edges in noisy images sometimes lead to missing true edges, false edge detection, false edge, localization etc.

The common edge detection methods are constructing the different kinds of differential operator such as Laplacian [1], Roberts [2], Sobel [3], LOG [4], Prewitt etc. These operators have the advantage of high detecting speed. However, they are all sensitive to step change of pixel gray level so that they are sensitive to noises and threshold determining. Therefore, these differential operators need image existing obvious edges between the image and background in order to obtain the ideal edges. Also classical edge detectors don’t work in varying lighting conditions and illuminations.

In recent years, Fuzzy logic [5, 6, 7, 8, 9], Neuron Network [10, 11, 12, 13] and Mathematical Morphology [14] are used for detection of edges. It is advantageous to use ANN than classical edge detectors as its adaptive learning ability helps in detecting edges in images even if they have non-uniform contrast with minor changes in lightening conditions, its generalization ability helps to detect edges in those images also that are not encountered during the training phase, and due to its parallel organization multiple inputs and outputs can be used during training phase. So the use of neural networks rather than classical method gives more successful results. Also the operational load is reduces as it automatically learns from the given set of inputs and desired outputs. On the other hand It is a form of mathematical logic which deals with reasoning that is approximate rather than fixed and exact. It handles the concept of partial truth - truth values between "completely true" and "completely false. Fuzzy logic adds an extra layer of intelligence, provides more practical means of problem solving. By seeing the benefits above benefits of both neural networks and fuzzy logic many researches have been done to combine both to get benefits of both technologies. Different authors have used different ways to combine these two technologies in order to get advantages out of both technologies. Victor Boskovitz and Hugo Guterman in [15] proposed a system that consists of a multilayer perceptron (MLP)-like network that performs image segmentationand edge detection by adaptive thresholding of the input image using labels automatically pre-selected by a fuzzy clustering technique. Fuzzy entropy is used as a measure of the error of the segmentation system as well as a criterion for determining potential edge pixels. Siwei Lu and Ziqing Wang in [16] proposed a fuzzy neural network which comprises of two stages one is adaptive fuzzification and second is detection. It consists of three layers of neurons. The first layer is an input layer which is divided into eight groups corresponding to blocks in the input pattern. The second layer in the network is used to measure the certainty of the classification for each block. The output layer provides the final measurement of classification. The proposed fuzzy neural network is trained by typical patterns to enable it to determine the edge elements with eight orientations. Pixels having high edge membership are traced and assembled into one picture. Dingran Lu et al. [17] artificial neural networks are employed for edge detection based on its adaptive learning and nonlinear mapping properties. Fuzzy sets are introduced during the training phase to improve the generalization ability of neural networks.

The paper is organized as follows. Section 2 gives introduction of neural network and fuzzy logic as well as it is providing brief introduction of integrated system combining the neural network and fuzzy logic. Section 3 explains the proposed method in detail. Results and comparisons are reported in section 4. Finally the section 5 is concluding the paper.

II. NEURAL NETWORKS AND FUZZY LOGIC

A. *FeedForward Neural Networks:*

Artificial neural networks are massively parallel adaptive networks of simple nonlinear computing elements called neurons which are intended to abstract and model some of functionality of the human nervous system in an attempt to partially capture some of its computational strengths. So Artificial Neural Network consists of group of artificial neurons which are interconnected and has a natural property for storing experimental knowledge as well as making it available for use in order to process information.

A neural network can be seen as a weighted directed graph in which nodes are represented by artificial neurons and connections between neurons are represented by directed weighted edges. Local groups of neurons can be either connected in feedforward architecture [11], in which a network has no loops or a feedback architecture in which loops occur in the network because of feedback connections. In feedforward neural network, neurons are organized into different layers and they have unidirectional connections between them. Such feedforward networks are static in the sense that the output depends only on the present input. The feedforward neural network (FNN) can be single layer or multiple layer. In single layer feedforward neural network, we have an input layer of source nodes that projects into output layer of neurons. Input layer of source nodes is not counted because no computation is performed here. Multi-layer feed forward neural network has one or more “hidden layers”. The computation nodes of hidden layer are called hidden neurons. The output of 1st layer is input for 2nd layer and output of 2nd layer is input of 3rd layer and so on. The Fig.1 shows the example of 2 layer feedforward neural network.

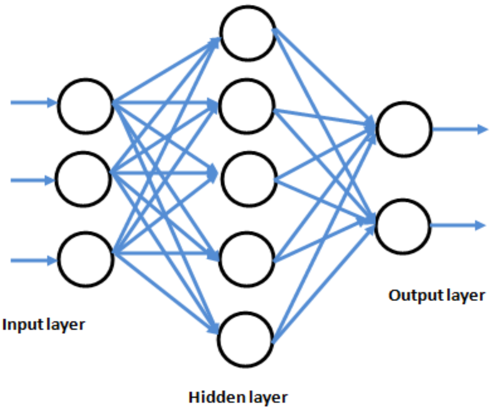


Figure.1: 2 layer feedforward neural network.

B. Fuzzy Logic

A fuzzy logic [18] is introduced by Dr. Lotfi Zadeh of U.C. Berkeley in the 1960's. Truth values in fuzzy logic or membership values (in fuzzy sets) are represented by a value in the range [0.0, 1.0], in which 0.0 represents absolute falseness and 1.0 represents absolute truth. The definition of a fuzzy set then, from Zadeh's paper is “Let **X** be a space of points, with a generic element of **X** denoted by $\boldsymbol{x}$. Thus **X** ={$\boldsymbol{x}$}. A fuzzy set **A** in **X** is characterized by a membership function $\mu_A(x)$ which associates with each point in **X** a real number in the interval **[0,1],** with the values of $\mu_A(x)$ at $\boldsymbol{x}$ representing the "grade of membership of $\boldsymbol{x}$ in **A”**. Thus, if the value of $\mu_A(x)$ is nearer to 1, the higher the grade of membership of x in **A”**. The membership function of a fuzzy set is a generalization of the indicator function in classical sets. In fuzzy logic, membership functions are represented by the degree of truth. Membership functions can take any form, but there are some common examples that appear in real applications. User can choose the membership functions arbitrarily, based on the user’s experience or be designed using machine learning methods (e.g., artificial neural networks, genetic algorithms, etc.) Membership functions can be of different shapes; triangular, trapezoidal, piecewiselinear, Gaussian, bell-shaped, etc. Fig.2 describes the membership of the 'tall' set using Gaussian membership function. From the Fig.2 we can see that classical set theory (Red line) allows the membership of the elements in the set in binary terms, an element either belongs or does not belong to the set while fuzzy set (Gaussian membership) allows its members to have different degree of membership in the interval [0 , 1].

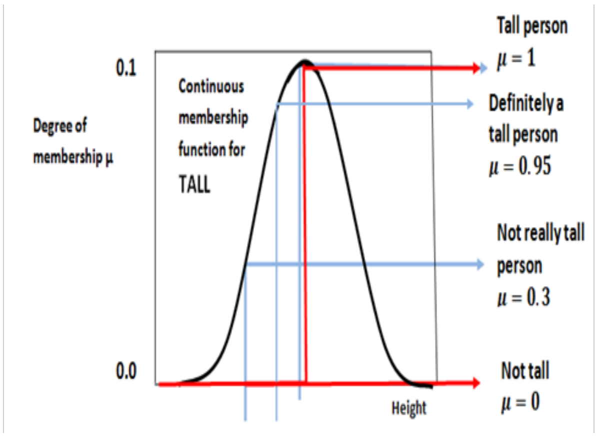


Figure.2: Figure demonstrating need of membership function.

C. *Combining Neural Networks and Fuzzy Logic:*

As fuzzy logic deals with the reasoning that is approximate rather than fixed and exact so this advantage/ property of fuzzy logic helps in improving generalization ability of ANN. The simple neurons are converted to fuzzy neurons using membership functions and then the network is trained. When neural network works on crisp values, some of the edge pixels may be ignored but when fuzzy logic is added, partially true values are also considered as they may belong to an edge. The main goal of this approach is to 'fuzzify' neurons of neural networks, using fuzzy logic. In this case, a crisp neuron becomes fuzzy.

The combined system will have the advantages of both neural networks (e.g. learning abilities, optimization abilities and connectionist structures) and fuzzy systems (human like approximate reasoning).

III. THE PROPOSED MODEL

The block diagram of the proposed model is given in Fig.3. Standard deviation and gradient values are used as training patterns as given in paper. Generalization ability of neural networks is improved by feeding them with fuzzy inputs rather than crisp inputs and also by making the hidden neurons as fuzzy neurons. The neurons in the input layer and hidden layer are converted to fuzzy neurons by using Gaussian membership function and the neuron in the output layer produces the crisp output. Traditional Backpropagation algorithm is used but the activation functions are replaced by membership functions. The result of each transfer function is the membership value of the corresponding input from the previous layer.

The proposed algorithm is implemented in MATLAB 7.10.0. The proposed method has seven steps.

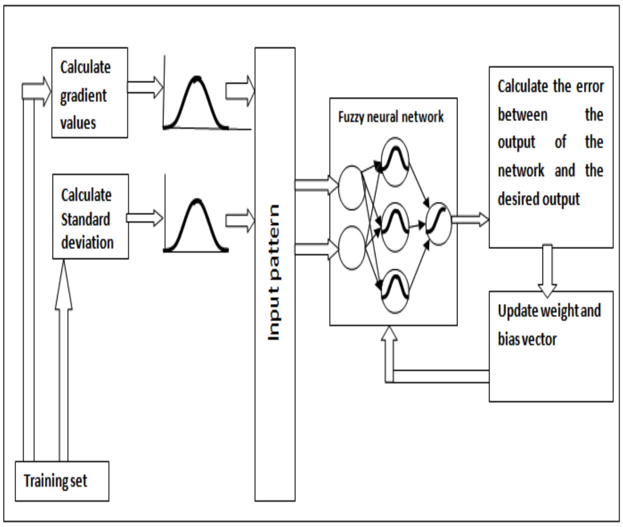

Calculate gradient values
Calculate Standard deviation
Training set
Input pattern
Fuzzy neural network
Calculate the error between the output of the network and the desired output
Update weight and bias vector

Figure.3: Proposed Model.

1) *Calculation of input values*

For the input values gradient and standard deviation of the image to be processed are calculated by dividing the image into 3x3 window as shown in Fig.4 where Z are the intensity values. The gradient values in horizontal direction are calculated using the masks Gx and mask Gy given in Fig.5 for vertical direction. Finally the edge magnitude is calculated using the equation given as:

$$G = (Gx^2 + Gy^2)^{1/2} \qquad (1)$$

$$G = \{[(Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3)]^2 + [(Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7)]^2 \}^{1/2} \qquad (2)$$

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

Figure.4: 3x3 window (neighbourhood).

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
0	0	0
-1	-2	-1

Gy

Figure.5: Sobel convolution masks.

The standard deviation of the pixels of a 3X3 neighbourhood is computed as using the equation as given below in equation 3 where m is the mean value of the pixels in the neighborhood, N is the number of pixels in the neighborhood. Elements of the mask with a non-zero value are considered part of the neighborhood.

$$s = \text{sqrt}\left(\frac{\text{sum}(x-\text{m}).^2}{\text{N}-1}\right) \quad (3)$$

Now both the values are normalized in the range 0 to 1 using the equation 4 Where Z is the input image in matrix form and $n(z)$ the normalized value of the pixel z.

$$n(z) = \frac{z}{\max(Z(:))} \quad (4)$$

2) Training data

The training pattern consist of 120 patterns out of which 25 are non-edge patterns and 96 are edge patterns. The pattern consists of two inputs correspond to gradient value and standard deviation value. These values/inputs can range from 0 to 1 in the interval of 0.1. Before feeding these crisp inputs to the neural networks for training, these crisp values are converted to fuzzy values using the Gaussian membership function (Fig.6) to increase neural networks generalization ability so that more training patterns can be employed by neural network using the equation 5 where $\mu(x)$is..... m is the central value and a standard deviation k > 0. The smaller k is, the narrower the "bell" is. In this research we choose c= 1and σ =0.25.

$$\mu(x) = \exp(-\frac{(x-c)^2}{2\sigma^2}) \quad (5)$$

where $\mu(x)$is..... m is the central value and a standard deviation k > 0. The smaller k is, the narrower the "bell" is. In this research we choose c= 1and σ =0.25.

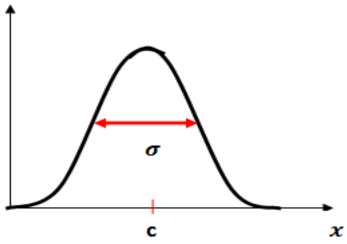


Figure.6: Gaussian membership function

When both the fuzzy inputs are low, the desired output is low else the desired output will be 1. We have considered all the values below 0.5 as low.

3) Testing data

The fuzzy values of crisp standard deviation and gradient values calculated using step 1 of the images to be tested are given as inputs to the network.

4) Network architecture

Network has one input Layer, one hidden layer and one output layer as shown in Fig.7. So it is a 2 layer feedforward network. Here 2 neurons at input layer corresponding to fuzzy values of standard deviation and gradient, 3 neurons at the hidden layer and 1 neuron at output layer. Tan-sigmoid transfer function shown in Fig.8 is used at output layer. Hyperbolic tangent transfer function in the terms of neural networks, is related to a bipolar sigmoid which has an output in the range of -1 to +1.

Gaussian Transfer function is used at hidden layers convert crisp hidden neurons into fuzzy neurons. The Gaussian membership function is shown in Fig.6.

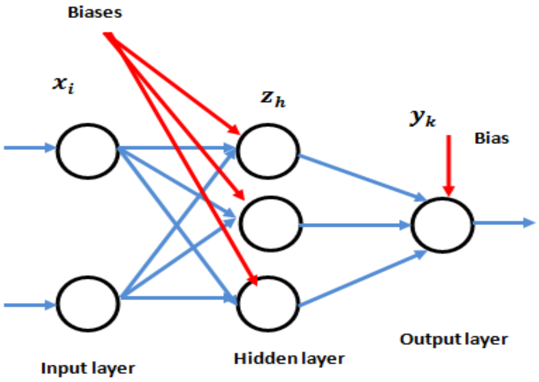


Figure.7: Error propagation through hidden layer.

5) Parameter Adjustment and Weight Initialization

The weights between the input and the hidden layer and between hidden and the output layer are initialized randomly. Learning rate is 0.5 initially. The network is trained for 1000 epochs.

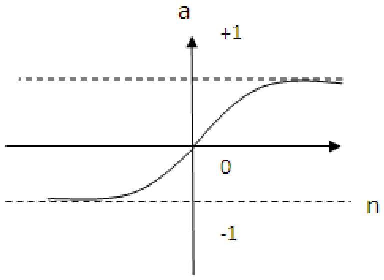


Figure.8: Hyperbolic tangent transfer function

$$\text{a} = \text{Tansig}(\text{n}) = \frac{2}{1+\text{e}^{-2\text{n}}} - 1 \quad (6)$$

6) Training

The network training is done using backpropagation learning algorithm [11] which minimizes the error and update the weights during learning until the calculated outputs are within the margin of the known outputs.

The steps in Backpropagation algorithm are as follows:

6.1) Given:

A training set T comprising vectors $X_k \in R^n$ and desired output vectors $D_k \in R^P$ and an *n-p-q* architecture neural network *N*.

6.2) Initialize:

Randomize weights w_{ih}^1 to small values , set $\Delta w_{ih}^0 = 0$, *i= 0,…..,n: h=1…..q*. Randomize weights w_{hj}^1 to small values , set $\Delta w_{hj}^0 = 0$, *h= 0,…..,q: j=1…..p*. Set k=1 and ϵ, α, and the error tolerance τ to desired values

6.3) Iterate:

Repeat

{

Select a training pair (X_k, D_k) ∈ *T*

Compute signals on forward pass in the following sequence

$$\delta(x_i^k) = x_i^k \qquad i=……..n \qquad (7)$$

$$\delta(x_0^k) = 1$$

$$Z_h^k = \sum_{i=0}^{n} w_{ih}^k \ \delta(x_i^k) \qquad h=1……q \qquad (8)$$

Applying activation function:

$$\delta(Z_h^k) = f(Z_h^k) \qquad (9)$$

Activation function applied here is Gaussian membership function which is shown in Fig.6:

$$\delta(Z_h^k) = \exp(-\frac{(z_h^k - c)^2}{2\sigma^2}) \qquad h=1……q \qquad (10)$$

$$\delta(Z_0^k) = 1 \qquad (11)$$

$$y_j^k = \sum_{h=0}^{q} w_{hj}^k \ \delta(z_{hi}^k) \qquad j=1………p \qquad (12)$$

Applying tan-sigmoid activation function:

$$\delta(y_j^k) = \frac{1}{1+\exp(-y_j^k)} \qquad j=1………p \qquad (13)$$

Compute deltas and errors at output neurons

$$\delta_j^k = \left(d_j^k - \delta(y_j^k)\right) \delta'(Z_h^k) \quad j=1……..p \qquad (14)$$

$$\Delta w_{ih}^k = \varepsilon \delta_j^k \delta(Z_h^k) \qquad h=0…..q, j=1……p \qquad (15)$$

Compute deltas and errors at output neurons

$$\delta_h^k = (\sum_{j=1}^{p} \delta_j^k \ w_{hj}^k) \ \delta'(Z_h^k) \quad h=1…….q \qquad (16)$$

$$\Delta w_{ih}^k = \varepsilon \delta_h^k \delta(x_i^k) \qquad i=0…….n, h=1…….q \qquad (17)$$

Update weights

$$w_{hj}^{K+1} = w_{hj}^K + \Delta w_{hj}^K + \alpha \Delta w_{hj}^{K-1} \qquad h=0....q, j=1.....p \qquad (18)$$

$$w_{ih}^{K+1} = w_{ih}^K + \Delta w_{ih}^K + \alpha \Delta w_{ih}^{K-1} \qquad i=0....n, h=1.....q \qquad (19)$$

Collect pattern error E_k

$$\} \text{ until } (E_{av} = \frac{1}{Q}\sum_{k=1}^{Q} E_k < \tau \ ) \qquad (20)$$

7) Testing

Now after successful training of network, it is tested for a number of different kinds of images and then desired output can be obtained for any kind of grayscale. To obtain the best and accurate result threshold of 0.5 is applied during testing.

IV. RESULTS AND COMPARISONS

The proposed algorithm can detect all possible kinds of edges in grayscale images and to prove this, the algorithm is tested over a number of different grayscale images and compared with traditional operators. In the paper we have shown the results of flower image, beans image, rainbow image, house image and a simple text image. Also we have detected the edge maps of images using Neural Network based method and using fuzzy logic based method and compared the results with the edge map obtained by our algorithm. We can clearly see from the Table I that when we combine both techniques, it gives better results than any other method. From the results of bean image, it is clearly seen that Roberts, Prewitt and Sobel fail to detect edges and Neural Network detects some false edges. The results of Fuzzy and LoG are better but are not able to detect dark spots on beans while proposed method detects true edges as well as dark spots. The result of flower image shows that only our method provides the best edge map. The third image is a simple text “JESAL”, the results of traditional method shows that edges of the letters are not complete and the edge map produced by the Neural Network gives rough edges while our method detects the edges af all methods and also provides smooth edges. The rainbow image has six boundaries but Roberts and Prewitt detects only four in which second and third are not clear. Sobel detects four clear boundaries but two are missing. LoG, Fuzzy and Neural Network produced incomplete six lines while our method is able to detect all six smooth edges. In the last house image also, our method provides best edge map. Fig.9 clearly shows that the number of edge pixels detected by proposed method is highest.

The proposed method also provides satisfactory results in case of noisy images and to prove that, three types of noise ‘salt and pepper’, ‘speckle’ and Gaussian noise at noise level of 60 db are added to the images. To compute PSNR (peak signal to noise ratio) value, first MSE (mean square error) is computed.

$$\text{MSE} = \frac{\sum_{\text{M,N}}[\text{I}_1(\text{m,n}) - \text{I}_2(\text{m,n})]^2}{\text{M}*\text{N}} \qquad (21)$$

In the above equation, *M* and *N* are the number of rows and columns in the input images, respectively. I_1 is the original image and I_2 is the noisy image.

Noise level (PSNR) is measured using the equations as given below:

$$\mathrm{PSNR} = 10\log_{10}\left[\frac{R^2}{\mathrm{MSE}}\right] \tag{22}$$

In the above equation, *R* is the maximum fluctuation in the input image data type. R is 255 for an 8-bit unsigned integer data type.

From Fig.10 to Fig.12, it can be clearly seen that even after introducing three kinds of noise, our algorithm is able to detect highest edge pixels.

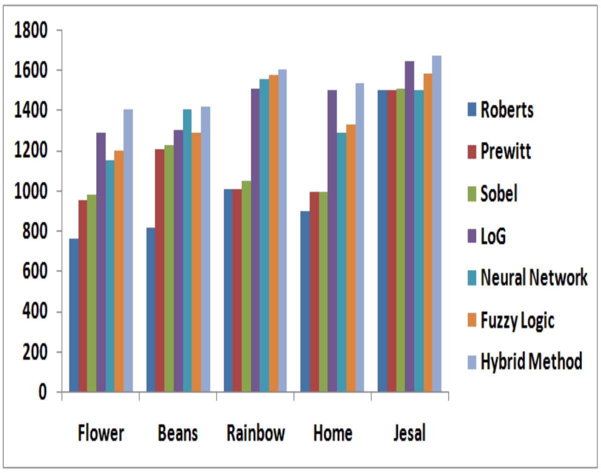


Figure.9: Number of edge pixels detected by different operators for different images.

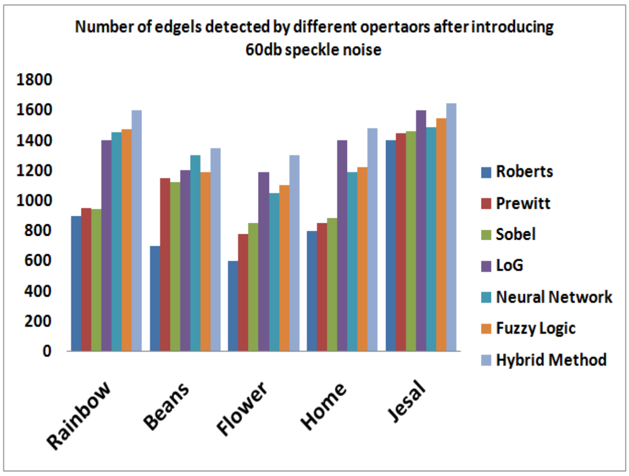


Figure.10: Number of edge pixels detected by different operators for different images.

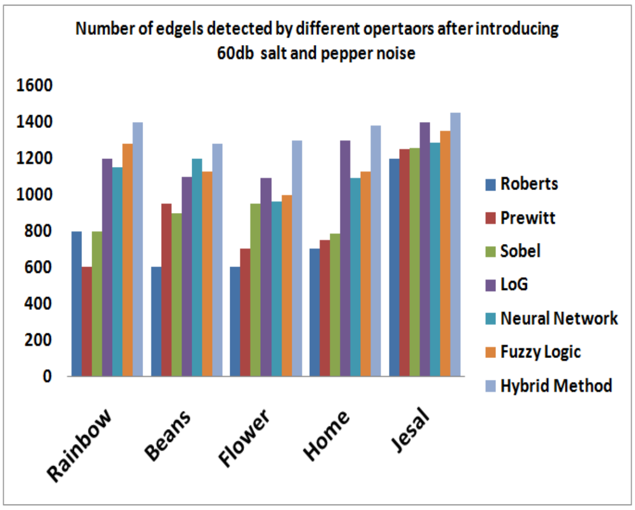


Figure.11: Number of edge pixels detected by different operators for different images

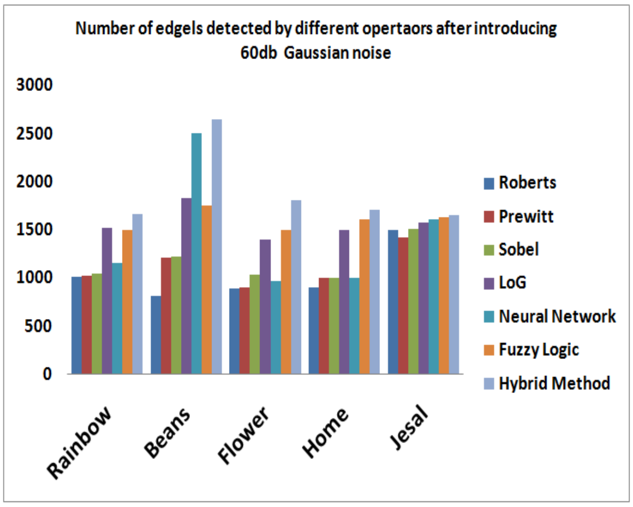
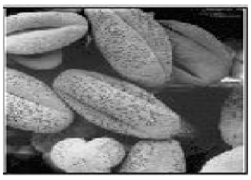
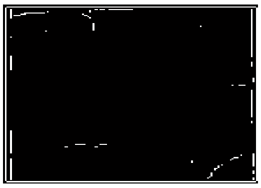
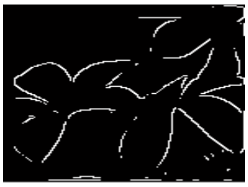
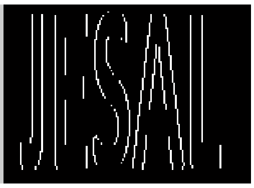
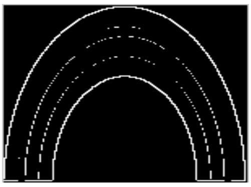
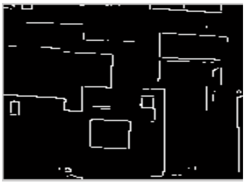
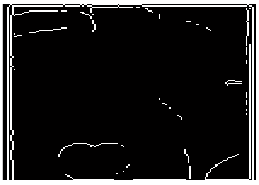
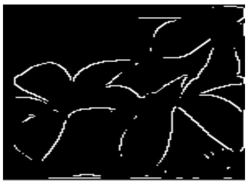
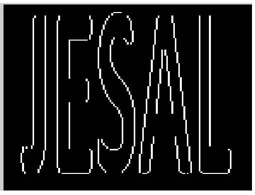
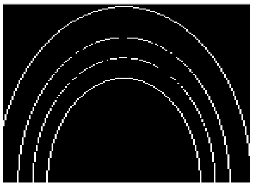
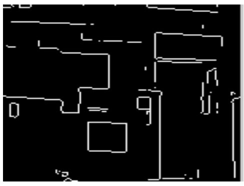
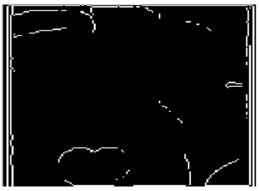
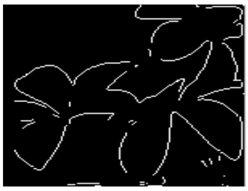
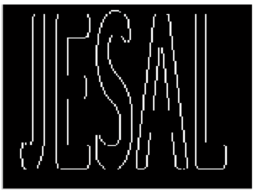
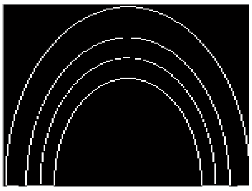
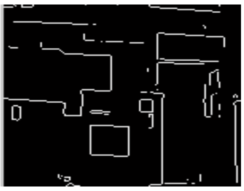
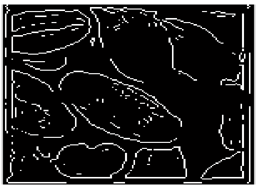
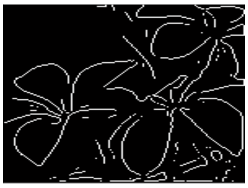
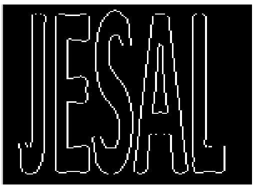
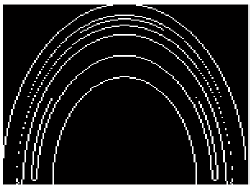
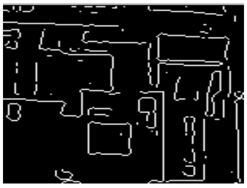
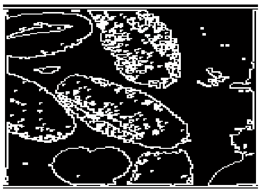
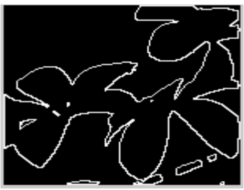
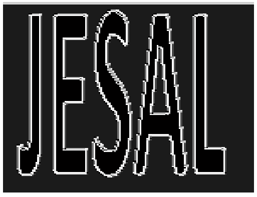
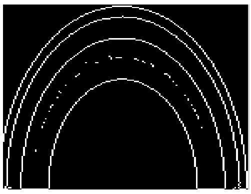
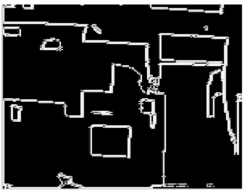
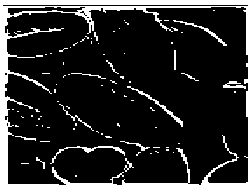
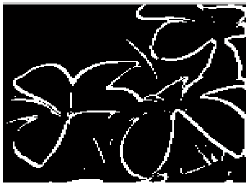
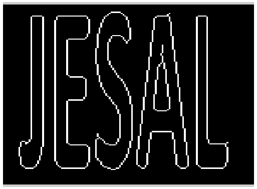
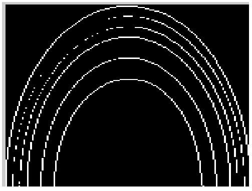
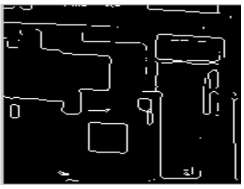
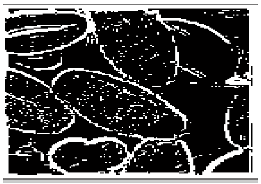
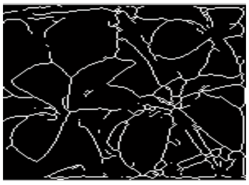
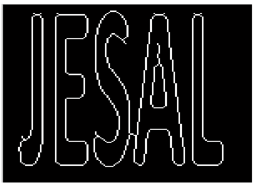
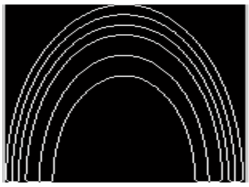
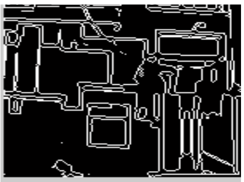


Figure.12: Number of edge pixels detected by different operators for different images.

V. CONCLUSION

A hybrid approach combining Neural Network and fuzzy logic for detection of edges in grayscale images is presented in this paper. Supervised learning method is used. The proposed method is compared with traditional edge detectors as well as with the other neural network and fuzzy logic based method. On the basis of visual perception and edgel counts of edge maps of various grayscale images it is proved that our algorithm is able to detect highest edge pixels in noise free images as well as in case of noisy images. Also it gives smooth and thin edges without distorting the shape of images.

TABLE I. EDGE MAPS OBTAINED BY VARIOUS METHODS.

	BEANS IMAGE	FLOWER IMAGE	TEXT IMAGE	RAINBOW IMAGE	HOME IMAGE
ORIGINAL GRAY SCALE IMAGES					
ROBERTS RESULTS					
PREWITT RESULTS					
SOBEL RESULTS					
LOG RESULTS					
NEURAL NETWORK RESULTS					
FUZZY RESULTS					
PROPOSED METHOD RESULTS					

REFERENCES

[1] Rafael C. Gonzalez, Richard E. woods, Steven L. Eddins, "Digital *Image Processing using Matlab*", Second Edition, Chapter 1.

[2] Muthukrishnan. R and M. Radha, "*Edge detection techniques for image segmentatio*n", International Journal of Computer Science & Information Technology (IJCSIT) Vol-3, No-6, pp.250-254, 2011.

[3] Wenshuo Gao, Lei Yang, Xiaoguang Zhang, Huizhong Liu, "*An Improved Sobel Edge Detection*", 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), Vol-5, pp.67 – 71, 2010.

[4] D. Marr and E. Hildreth, "*Theory of edge detection"*, Pross. Royal Soc., London, Vol-207, pp.187-217, 1980.

[5] George J. Klir, Bo Yuan, "*Fuzzy Sets and Fuzzy Logic*", First Edition, Chapter 13.

[6] Aborisade, D.O, "*Novel Fuzzy logic Based Edge Detection Technique*", International Journal of Advanced Science and Technology, Vol-29, pp.75-82, 2011.

[7] Wafa barkhoda, Fardin Akhlaqian Tab, Om-Kolsoom Shahryari, "*Fuzzy Edge Detection Based on Pixel's Gradient and Standard Deviation Values*", Proceedings of the International Multiconference on Computer Science and Information Technology, pp.7 – 10, 2009.

[8] Ching-Yu Tyan and Paul P Wang, "*Image Processing - Enhancement, Filtering and Edge Detection Using the Fuzzy Logic Approach*", IEEE International Conference on image processing, Vol-1, pp.600 – 605, 1994.

[9] Todd Law, Hidenori Itoh, Hirohisa Seki, "*Image Filtering, Edge Detection, and Edge Tracing Using Fuzzy Reasoning*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol-18, No-5, pp.481 – 491, 1996.

[10] Symon Haykin, "*Neural Networks and Learning Machines*", Third Edition.

[11] Satish kumar, "*Neural Networks -A Classroo Approach"*, Second Edition.

[12] Terry, P., Vu, D., *Edge Detection Using Neural Networks*, Conference on Signals, Systems and Computers, Vol. 1, pp. 391-395, 1993.

[13] Li, W., Wang, C., Wang, Q., Chen, G., "*An Edge Detection Method Based on Optimized BP Neural Network"*, Proceedings of the International Symposium on Information Science and Engineering, pp. 40-44, 2008.

[14] Zhao Yu-qian, Gui Wei-hua, Chen Zhen-cheng , Tang Jing-tian , Li Ling-yun., "*Medical Images Edge Detection Based on Mathematical Morphology*", Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference ,Shanghai, China, pp. 6492 - 6495 , 2005.

[15] Victor Boskovitz and Hugo Guterman, "*An Adaptive Neuro-Fuzzy System for Automatic Image Segmentation and Edge Detection*", IEEE Transactions on Fuzzy Systems, Vol-10, No-2, pp.247 – 262, 2002.

[16] Siwei Lu , Ziqing Wang, "*Fuzzy Neural Networks for Edge Detection*", IEEE Canadian Conference on Electrical and Computer Engineering Vol. 2, pp. 446 – 449,1917.

[17] Dingran Lu, Xiao-Hua Yu, Xiaomin Jin, Bin Li, Quan Chen, Jianhua Zhu, "*Neural Network Based Edge Detection for Automated Medical Diagnosis"*, Proceeding of the IEEE International Conference on Information and Automation, pp.343-348, 2011.

[18] L. A. Zadeh, "*Fuzzy sets*," Information and Control, Vol-8, pp.338-353, 1965.

Jesal Vasavada was born in India, in 1989. She has completed her B.Tech. (Computer Sc. & Engineering) in 2011 and M.Tech. (Computer Sc. & Engg.) in 2013 from Rajasthan Technical University and Mody Institute of Technology & Science, Deemed University, Laxmangarh respectively. Her research interest lies in Digital Image Processing.

Shamik Tiwari was born in India, in 1980. He has completed his B.Tech. (Computer Sc. & Engineering) in 2003 from RGPV University Bhopal and M.Tech. (Computer Sc. & Engg.) in 2007 from Dr. B. R. Ambedkar University Agra. He is working as an Assistant Professor in Mody Institute of Technology & Science, Deemed University Laxmangarh. Presently, he is pursuing Phd. in Computer Sc. & Engg. from the MITS Lakshmangarh. He has published more than 10 papers in refereed journals and conference proceedings. He is an author of the book ―Digital Image Processing from Dhanpat Rai Publishing (India). His research interest lies in Digital Image Processing, Blur Detection, and their applications in computer vision.