International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622
International Conference On Industrial Automation And Computing (ICIAC- 12-13 April 2014))

RESEARCH ARTICLE OPEN ACCESS

An Edge Detection Method Using Back Propagation Neural Network

Ms. Utkarsha Kale*, Dr. S. M. Deokar**

*(Department of Electronics and Telecommunication, Sinhgad Institute of Technology and Science,Pune, University of Pune, Pune Email: utkarshafk@gmail.com)

** (Head of Department of Electronics and Telecommunication, Sinhgad Institute of Technology and Science,Pune , University of Pune, Pune Email: Smdeokar_sits@sinhgad.edu)

ABSTRACT

Edge detection is an important processing step in Image Processing. Edge detection is the set of mathematical rules which identifies the point in digital image where the brightness of image sharply changes or has discontinuities. The image edges include rich information that is very significant for obtaining the image characteristics by object recognition. So, edge detection is a vital step in image analysis and it is the key of solving many complex problems. In this paper a Feed-forward Neural Network (FNN) based algorithm is proposed to detect edges in gray scale images. The back-propagation learning algorithm is used to minimize the error. Output of Canny Edge detection is used as target. In the end the network is tested for a number of different kinds of grayscale images. The proposed scheme is compared with Prewitt, Sobel and Canny edge detector. This method gives satisfactory results.

Keywords **–** Image Processing, Edge Detection, Neural network, MATLAB, Back-propagation

I. INTRODUCTION

Edges in images are simply the boundary between the object and the background. Edges provide important information of objects such as shape, area, perimeter etc [1]. Edges are often used to distinguish one object from the other and/or separate them from the background. In image processing, edge detection is used to filter out less relevant information and at the same time preserves the basic structural properties of an image. It significantly reduces the amount of data to be processed in the subsequent steps such as feature extraction, image segmentation. Edge detection is the basic or fundamental steps whose results are used in many applications like pattern recognition, image analysis, object recognition etc. The edge detection is used in many applications in image processing. It is currently crucial technique of image processing. Edges can be easily recognized as their detection takes a very short time. Edges are quick changes on the image profile. These quick changes on the image can be detected via traditional difference filters [2]. Also it can be also detected by using canny method [3] or Laplacian of Gaussian (LoG) method [4]. In these classic methods, firstly masks are moved around the image. The pixels which are the dimension of masks are processed. The edge detection is used in many applications in image processing. It is currently crucial technique of image processing. Edges can be easily recognized as their detection takes a very short time. Edges are quick changes on the image profile. These quick changes on the image can be detected via traditional difference filters [2]. Also it can be also detected by using canny method [3] or Laplacian of Gaussian (LoG) method [4]. In these classic methods, firstly masks are moved around the image. The pixels which are the dimension of masks are processed. Then, new pixels values on the new image provide us necessary information about the edge. However, errors can be made due to the noise while mask is moved around the image. These classic methods are sensitive to noise and lead to false edge detections in case of noisy images.

Recently, artificial neural networks (ANN) have been applied to edge detection[4]. An artificial neural network is computational model that is inspired by the structure and /or functional aspects of "biological neurons of brain". So ANN consists of an interconnected group of artificial neurons and has a natural property for storing experimental knowledge and making it available for use in order to process information. Artificial neural network can be used as a very prevalent technology, neural network technology is better than classical edge detectors in various perspectives. First it provides less operation load and has more advantageous for reducing the effect of the noise in images. Second is its adaptive learning ability. If a neural network is trained to detect edges in grayscale images having uniform contrast, it can be easily retrained to deal images having non-uniform contrast with minor changes in lightening conditions as neural network has ability to change its synaptic weights in real time. The third advantage is its generalization ability. Generalization

refers to the neural network's ability to detect edges for those images also that are not encountered during training (learning) phase. Fourth is its non linear mapping ability. An artificial neuron can be linear or non linear. Non–linearity is highly important property, particularly if the underlying physical mechanism responsible for generation of the input image is inherently non-linear. Fifth, its parallel organization permits solutions to problems where multiple constraints must be satisfied simultaneously. Due to this property artificial neural network is more useful, because multiple inputs and multiple outputs can be used during the stage of training. For example in traditional methods one pixel is processed at a time, but in neural networks multiple pixels as inputs can be given. In paper [5], nine pixels of a 3x3 window of an image as inputs are given simultaneously. Also artificial neural networks are fault tolerant in nature as its performance degrades gracefully under adverse conditions [5].

In this paper feed-forward neural network based method has been proposed to detect edges in grayscale images using Canny Edge detector output as target for training.

II. LITERATURE SURVEY

The edges provide important visual information since they correspond to major physical, photometrical or geometrical variations in scene object. Edge detection is a terminology in image processing that refers to algorithms which aim at identifying edges in an image. Many works have been done to detect the edges using neural networks. Li and Wang in [6] proposed a parallel model of Back-Propagation (BP) neural network to detect tile defect. Firstly, the BP neural network for edge detection of binary image was designed. Secondly, the gray image was divided into 8 binary images according to the bit plane. Thirdly, a parallel model of BP neural network, which was composed by 8 sub BP neural network for edge detection of binary image, was constructed. Fourthly, the sub BP neural network was used for edge detection of every binary image. Finally, the output of every sub BP neural network was adjusted according the weight of every bit plane, and the accurate edge of gray image will be obtained. Terry and Vu in [6] investigated the application of multi-layer feed-forward neural networks for the edge detection of the LADAR (laser radar) image of a bridge. Multiple neural networks are trained by synthetic edge patterns; each one of them can detect a specific edge pattern (e.g., horizontal, vertical, diagonal, etc.). If desired, one can also combine the outputs from the "group" of neural networks to detect multiple types of edges in images[4]. Lu et al. in [5] proposed the ANN model which has one input layer, one output layer, and one hidden layer. There are 9 neurons in the input layer; in other words, the input of this network is a 9×1 vector which is converted from a 3×3 mask. There are 10 hidden neurons in the hidden layer; and one neuron in the output layer which indicates where an edge is detected. The neural network is fully trained by the 17 binary patterns.

III. NEURAL NETWORKS

3.1) Feed-forward Neural Networks-

feed-forward neural network [4] is a biologically inspired classification algorithm. It consists of a (possibly large) number of simple neuron-like processing units, organized in layers. Every unit in a layer is connected with all the units in the previous layer. These connections are not all equal, each connection may have a different strength or weight[4]. The weights on these connections encode the knowledge of a network. Data enters at the inputs and passes through the network, layer by layer, until it arrives at the outputs.

It is classified into single layer feed-forward or multilayer feed-forward neural network. In single layer feed-forward neural network, we have an input layer of source nodes that projects into output layer of neurons. Multi-layer feed forward neural network distinguishes itself by the presence of one or more "hidden layers" whose computation nodes are called hidden neurons. Hidden layer has ability to extract higher order statistics particularly when size of input layer is large. The source node in the input layer constitute the input signals applied to neurons in the second layer(first hidden layer).The output signal of second layer are used as inputs to 3rd layer and so on for the rest of the network. Fig.1 shows an architecture of 2 layer feed-forward neural network.

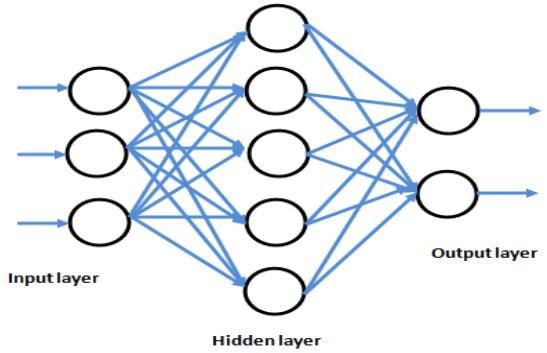


Fig 1: 2 layer Feed-forward neural network

3.2) Back propagation Learning-

During learning, the error function decreases and the weights are updated. The decrease may be accomplished with different optimization techniques such as the Delta rule, Boltzmann's algorithm and the back-propagation learning.

An edge-detection using neural network can be trained with back-propagation using relatively few training patterns. The most difficult part of any

neural network training problem is defining the proper training set. Back-propagation [4] is the most commonly used method for training the artificial neural network to minimize the gradient as shown in Fig.2. The steps in Back-propagation algorithm are as follows:

3.2.1 Randomly initialize the weight.

3.2.2 Feed the training sample.

3.2.3 Propagate the inputs forward

This step computes the net input and output of each unit in the hidden and output layers as follows-

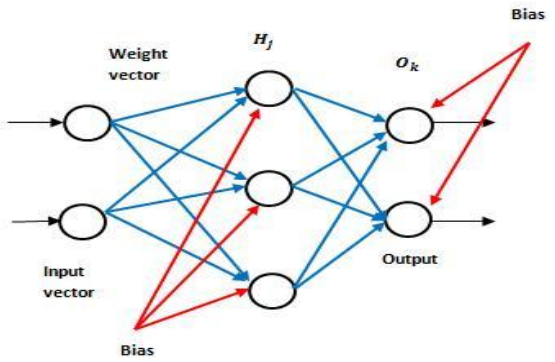


Figure 2: Error propagation through hidden layer [3]

Each hidden unit (H_j) sums its weighted input signals as,

$$H_{inj} = \sum w_{ij} x_i + w_{oj} \quad (1)$$

Where, w_{ij} is the weight between hidden and input layer; x_i is the input training vector; and w_{oj} is the bias of the unit. Applying Activation function,

$$H_j = f(H_{in}) \quad (2)$$

And sigmoid function is calculated as,

$$f(Hin) = \frac{2}{1+e^{-2Hinj}} - 1 \quad (3)$$

And this is provided as input for the output layer. Each output unit (Ok, k=1….m) sums its weighted input signals as,

$$O_{ink} = \sum w_{jk} H_j + w_{ok} \quad (4)$$

And output signal after applying activation function,

$$O_k = f(O_{ink})$$

3.2.4 Back propagate the error to the hidden layer

When reaching the Output layer, the error is computed and propagated backwards to hidden. For a unit k in the output layer the error is computed by a formula,

$$\partial_k = (D_k - O_k) f(O_{ink}) \quad (5)$$

∂_k is error at output unit O_k .Each hidden unit H_j sums its delta input from above layer inputs as,

$$\partial_{inj} = \sum_k \partial_k w_{ik}$$

The error is calculated as,

$$\partial_j = (\partial_{inj}) f(H_{inj}) \quad (6)$$

Where, ∂_j is Error at hidden unit H_j [3]

1.2.5 Update weights

Training and learning functions are mathematical procedures used to automatically adjust the network's weights and biases.

Weights are updated by the following equations[3]:

$$\Delta w_{ij} = \alpha \partial_j x_i$$

$$\Delta w_{ik} = \alpha \partial_j H_j$$

Where α is a learning rate.

Biases are updated by the following equations:

$$\Delta w_{oj} = \alpha \partial_j$$

$$\Delta w_{ok} = \alpha \partial_k$$

So new weights are:

$$w_{ij(new)} = w_{ij(old)} + \Delta w_{ij}$$

$$w_{jk(new)} = w_{jk(old)} + \Delta w_{jk}$$

And new biases are:

$$w_{oj(new)} = w_{oj(old)} + \Delta w_{oj}$$

$$w_{oj(new)} = w_{ok(old)} + \Delta w_{ok}$$

IV. PROPOSED METHOD

The proposed method has seven steps.

4.1 Calculation of input values

To calculate input values of image, divide the grayscale image into 3x3 windows as shown in Fig.3 where Z is the intensity value[5].

Z1	Z2	Z3
Z4	Z5	Z6
Z7	Z8	Z9

Figure 3: 3x3 window (neighborhood)

Normalize input values in the range 0 to 1 using the equation,

$$n(x) = \frac{x}{\max(X(:))}$$

Where n(x) is the normalized value of pixel x and X is the image in matrix form.

4.2 Training data

70x70 image is given to the canny edge detector and output of this is considered as a target for training. These values/inputs can range from 0 to 1 in the interval of 0.1. The input values and the desired output are as shown in Figure.

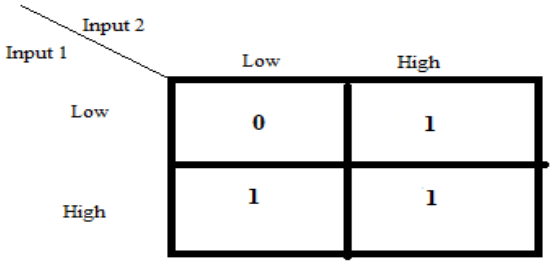


Fig 4: Input values and training patterns

Thus when both the 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.

4.3 Testing data

Normalized values of input image are taken and given to the network.

4.4 Network architecture

Network has one input Layer, one hidden layer and one output layer. So it's a 2 layer Feed-Forward network. Here 9 inputs, 6 neurons at the hidden layer and 1 neuron at output layer. Tan-sigmoid transfer function is used. Hyperbolic tangent transfer function (TANSIG) in the term of neural networks, is related to a bipolar sigmoid which has an output in the range of -1 to +1.

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

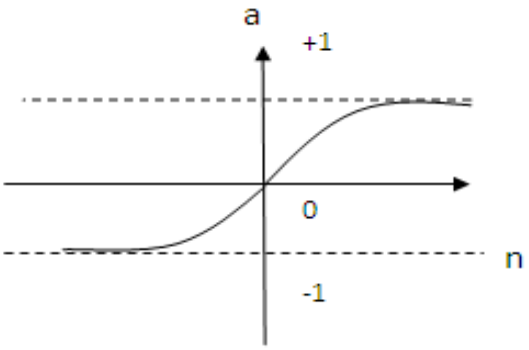


Fig 5: Hyperbolic tangent function

4.5 Parameter Adjustment and Weight Initialization

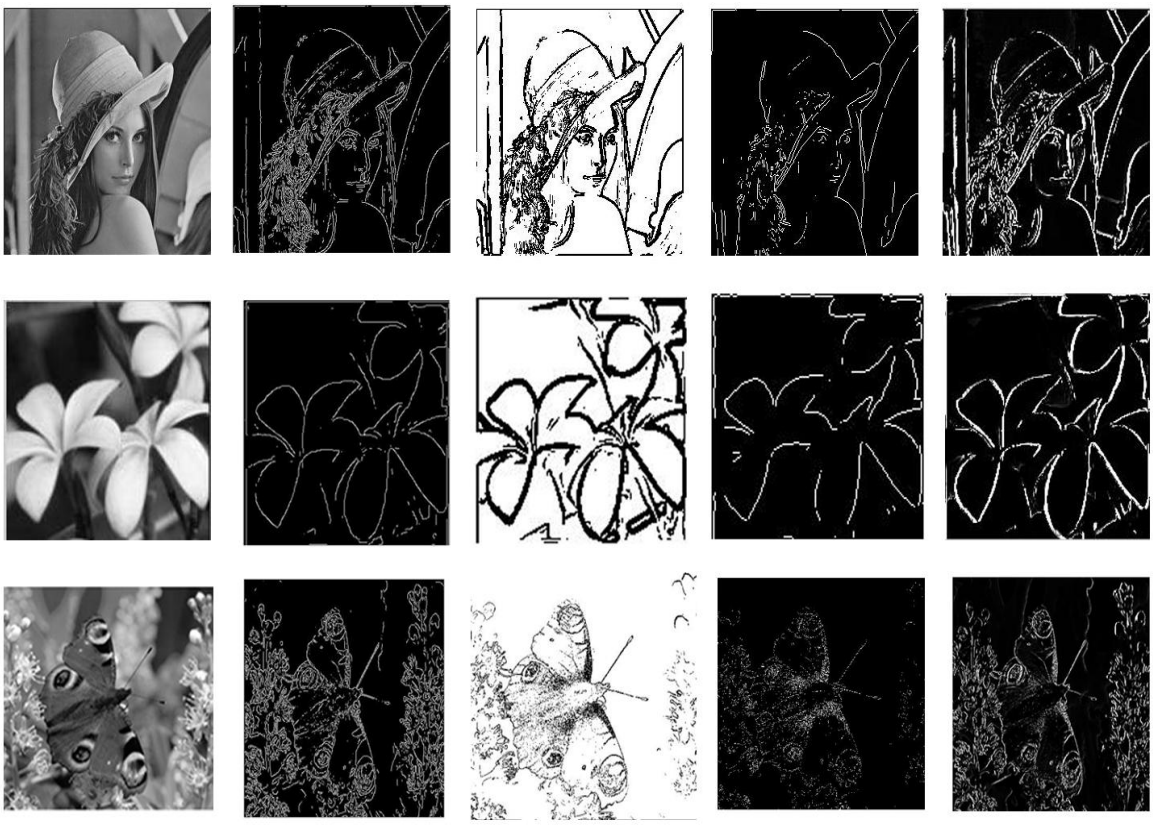
The weights between the input and the hidden layer and between hidden and the output layer are initialized randomly. Learning rate is 0.025 initially. The network is trained for 1000 epochs.

4.6 Training and testing

The network training is done using back-propagation learning algorithm which minimizes the error and update the weights during learning until the calculated outputs are within the margin of the known outputs. After successful training of network, the network is tested for a number of different kind of images. To obtain the best and accurate result threshold of 0.5 is applied during testing.

V. RESULTS AND COMPARISIONS

Finally this algorithm can detect all edges of any kind of grayscale images and prove that this image is tested on all possible kind of grayscale images. In this section comparison is done between the proposed method and with conventional methods like Roberts, Sobel and Prewitt edge detectors. In figure 6 comparison of all edge detectors is done on visual perception.



(a) (b) (c) (d) (e)

Figure 6: a) Main image, b) Roberts result c) Sobel Result d) Prewitt result e) Result of developed method

VI. CONCLUSION

A feed-forward neural network based system for detection of edges in grayscale images using neural network is presented in this paper. Back-propagation learning method is used. The proposed method is compared with traditional edge detectors. On the basis of visual perception, the results show that this algorithm is able to detect highest edges and perform robustly for all kind of images encountered in real world applications as it does not distort the shape and also retains the important features. We used MATLAB 2010b neural network tool box for implementation of network and for back-propagation algorithm.

REFERENCES

Journal Papers:

[1] Ramani Maini and Dr. Himanshu Aggarwal, Study and Comparison of Various Image Edge Detection Techniques, *International Journal of Image Processing (IJIP), Vol. 3, Issue 1, 2011.*

[2] S.Lakshmi and Dr. V. Sankaranarayan, Study of Edge Detection Techniques Segmentation Computing Approaches, *IJCA Special Issue on Computer Aided Soft Computing Techniques for Imaging and Biomedical Applications, CASCT, 2010.*

[3] Jesal Vasavada, Shamik Tiwari, An Edge detection method for grayscale images based on BP feedforward Neural network, *International Journal of Computer applications(0975-8887), Vol 67-No.2*

Books:

[4] Simon Haykin, *Neural Networks and Learning Machines*, 3rd edition.

[5] Gonzalez, R., Woods, R., *Digital Image Processing*, 3rd Edition. Prentice Hall, 2008

International Conference:

[6] N. Senthilkumaran and R. Rajesh, Edge Detection Techniques for Image Segmentation- A Survey, *International Conference on Managing Next Generation Software Applications, pp.749-760, 2008.*