# Edge Detection Using Scaled Conjugate Gradient Algorithm in Back Propagation Neural Network

**Dr. Walaa M. Khalaf**
Engineering College, University of Almustansiriya / Baghdad
Email:walaakhalaf@yahoo.com
**Dr. Mohammed Ali Tawfeeq**
Engineering College, University of Almustansiriya / Baghdad
**Dr. Kadhum Al-Majdi**
Engineering College, University of Almustansiriya / Baghdad

**Received on: 6/3/2013 & Accepted on 5/9/2013**

## ABSTRACT

This paper introduces a proposed method based on a backpropagation artificial neural network using Scaled Conjugate Gradient (SCG) training algorithm so as to gain the edges of any image. A new training image model is suggested to train this artificial neural network, then using this network to find the edges of any image. Computer experiments are carried out for extracting edge information from real images; the results presented are compared with those from classical edge detection methods like Canny. Using this new method does not need to tune any parameter to find the edge of any image, as well as using this method the false edges is reduced.

**Keywords:** Edge Detection Scaled Conjugate Gradient, Neural Network.

# اكتشاف حافات الصور باستخدام خوارزمية متقارنة التدرج المقيسة في شبكات الانتشار العكسية العصبية

## الخلاصة

هذا البحث يقدم طريقة حديثة تعتمد على شبكات الانتشار العكسي العصبية التي تستخدم في تدريبها خوارزمية متقارنة التدرج المقيسة (SCG) للحصول على حافات أي صورة. تم اقتراح نموذج صورة جديد لتدريب الشبكة العصبية الصناعية لتمكين استخدام هذه الشبكة لاحقاً لأيجاد الحافات لأي صورة. التجارب نفذت لفصل معلومات الحافة من صور حقيقية، وهذه النتائج قورنت مع نتائج الطرق التقليدية مثل كاني. ان استخدام هذه الطريقة لا يحتاج لتنغيم اي معاملات لأيجاد حافات أي صورة، فضلاً عن انها تقلل عدد الحافات الكاذبة.

## INTRODUCTION

An edge is the definition of the boundary between an object and background, also refers to the boundary between overlapping objects. This means, if the edges in an image are identified accurately, all the objects can be located and basic properties such as area, perimeter, and shape can be measured [1]. Edge detection is considered as a key issue in image processing, computer vision, and pattern recognition. There are several methods to perform the edges detection process, some of them based on derivatives, optimality criteria, statistical procedures, Genetic algorithms, or by using Support vector machines. As an example of using derivatives is the work of Hardie et. al. [2] where they designed filters have the ability to convert smooth edges to step edges and suppress noise simultaneously. Thus, false alarms due to noise are minimized and edge gradient estimates tend to be large and localized. Kim et al. [3] described an edge magnitude and direction scheme that uses 3×3 ideal binary pixel patterns and a lookup table. Final edges are determined automatically using the non-maximum suppression with edge confidence measure and fuzzy-based edge thresholding, even in a changing environment.

As an example on the method depends on optimality criteria is the work of Canny [4] in 1986 when he saw that the edge detection problem is a signal processing optimization problem, so he developed an objective function to be optimized. The solution to this problem was a rather complex exponential function, but Canny found several ways to approximate and optimize the edge-searching problem. His method consists of the following steps, smoothing the image with a two dimensional Gaussian, taking the gradient of the image, and then applying this decision for each point if it hasn't a maximum gradient it should be suppressed, and the final step is edge thresholding, it makes use of both a high threshold and a low threshold to decide wither the pixel is an edge pixel or not.

Rakesh et al. [5] proposed an edge detector performing thresholding by using statistical principles. Local standardization of thresholds for each individual pixel (local thresholding) depends upon the statistical variability of the gradient vector at that pixel. Such a standardized statistic based on the gradient vector at each pixel is used to determine the eligibility of the pixel to be an edge pixel. However, the values of the input parameters providing the appreciable results in the proposed detector are found to be more stable than other edge detectors and possess statistical interpretation.

Bhandarkar et al. [6] detect edges by using Genetic algorithms. They formulated the edge detection problem as one of choosing a minimum cost edge configuration. The edge configurations are viewed as two-dimensional *chromosomes* with fitness values inversely proportional to their costs. They tested the genetic algorithm with various combinations of meta-level operators on synthetic and natural images. They showed that their genetic algorithm-based technique is performed very well in terms of robustness to noise, rate of convergence and quality of the final edge image.

Zehng et al. [7] presented an innovative edge detection algorithm, using both the gradients and the zero crossings to locate the edge positions. Based on the least squares support vector machine (LS-SVM) with Gaussian radial basis function kernel, a set of the new gradient operators and the corresponding second derivative operators are obtained. The experimental results indicated that their proposed edge detector is near equal to the Canny in the performance and is fast in the speed.

Edge detection is considered as a fundamental of low-level image processing and good edges are necessary for higher level processing [8]. The quality of edge detection is highly dependent on different conditions, for example, lighting conditions, density of edges in the scene, the presence of objects of similar intensities, and noise. Each of these conditions can be handled by adjusting certain parameters in the edge detector and changing the threshold value for considering this part as an edge or not. Till now, no good method has been determined for automatically setting these values, so they must be manually adjusted by an operator each time the detector is run with a different set of data [9].

In this paper a proposed method is presented to detect the edges without using or calculating the derivative approximation or mathematical method. The main idea in this work is to apply a supervised learning method depends on training a classifier by training a neural network to detect the presence of edge into an image. The proposed method consists of two phases, first one is the training phase which is performs by using a small own-created image with a pre defined edge locations to train the neural network how to take the decision on the second phase which is called testing phase when the resulted classifier from training phase must be applied on the image to find its edges. The resulted edges obtained from this new method are similar to those from standard methods but by using a different point of view. This work is implemented by using MATLAB R2010b 32-bit (win32).

The rest of the paper is organized as follows; section 2 introduces some details about the proposed neural network which is used for edge detection. The classification model for edge detection is described in section 3 while some results on a set of images usually adopted as benchmark tests are put in section 4. Section 5 presents the conclusion of this work.

## NEURAL NETWORK SYSTEM

Neural networks have been used in various fields, including classification, clustering, pattern recognition, identification, speech, vision, control systems. In the field of classification, it is found that a two layer feedforwad networks can solve many practical classification problems.

Backpropagation is the generalization of the delta rule to multiple-layer networks and nonlinear differentiable transfer functions. Input and target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors in an appropriate way as defined by the user. Standard backpropagation is a gradient descent method, in which the network weights are moved along the negative of the gradient of the performance function to minimize the total squared error of the output computed by the net. The term backpropagation refers to the manner in which the gradient is computed for nonlinear multilayer networks.

Training a network by backpropagation involves three stages: the feed-forward of the input training pattern, the backpropagation of the associated error, and the adjustment of the weights. A flow chart illustrates the training process is shown in Figure (1).

In this work, a feed forward backpropagation neural network is used to classify any input image pixels into two categories, edge or no edge. The architecture of the net is consisting of input layer with four nodes, one hidden layer, and a single unit output layer with full interconnection links between the consecutive layers.

Several experiments were done to determine the optimal net architecture that can fit the data of the proposed model. It is found that the best net response is by choosing 44 neurons in the hidden layer. Several learning algorithms were tested also to get fast convergence with best performance, such as: the adaptive learning rate with momentum algorithm (GDX), Fletcher-Reeves update algorithm (CGF), and the scaled conjugate gradient algorithm (SCG). Although the last two algorithms are both conjugate gradient, but the CGF algorithm required a line search at each iteration, which is computationally expensive because it requires that the network response to all training inputs be computed several times for each search. The SCG algorithm was designed to avoid the time-consuming line search [10].

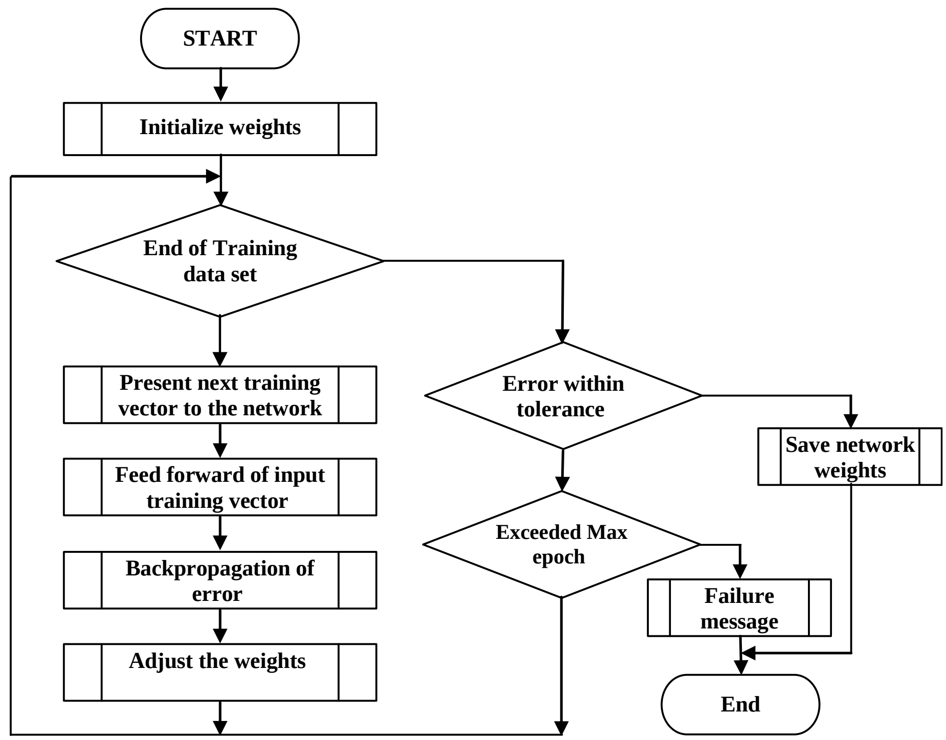


**Figure (1) back propagation training algorithm.**

On the other hand, the GDX and SCG algorithms are compared by learning the net on each algorithm separately and check the response of the net using the proposed model's data as input to the net. The network architecture is shown in Figure (2).

The responses of the trained net to the same data are shown in Figure (3). In this figure, it is clear that the output of the GDX net does not meet the target values in different points. In addition, it remarked that the performance of the SCG net is much better. For these reasons the SCG algorithm is chosen to learn the net.

The activation functions are used in this net are; the hyperbolic tangent (***tanh***) for the first layer, and the symmetric saturating linear function for the output layer. The reason behind using the saturated linear function is to accommodate the chosen target value (-1 for edge pixels and 1 for No edge pixels).

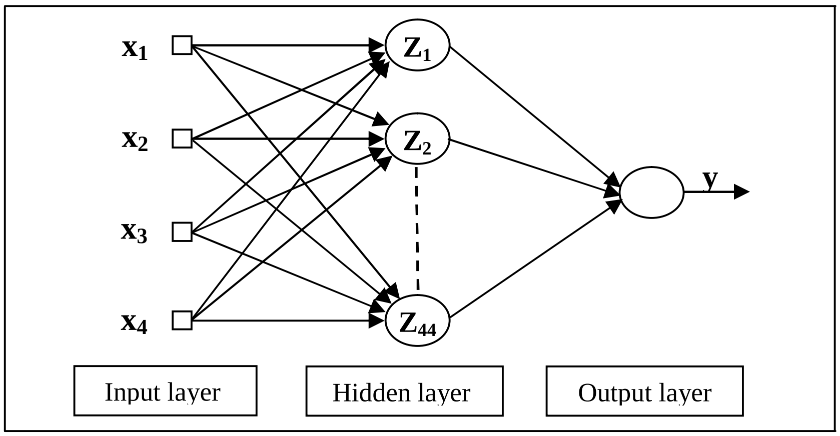


**Figure (2) Neural network Architecture.**

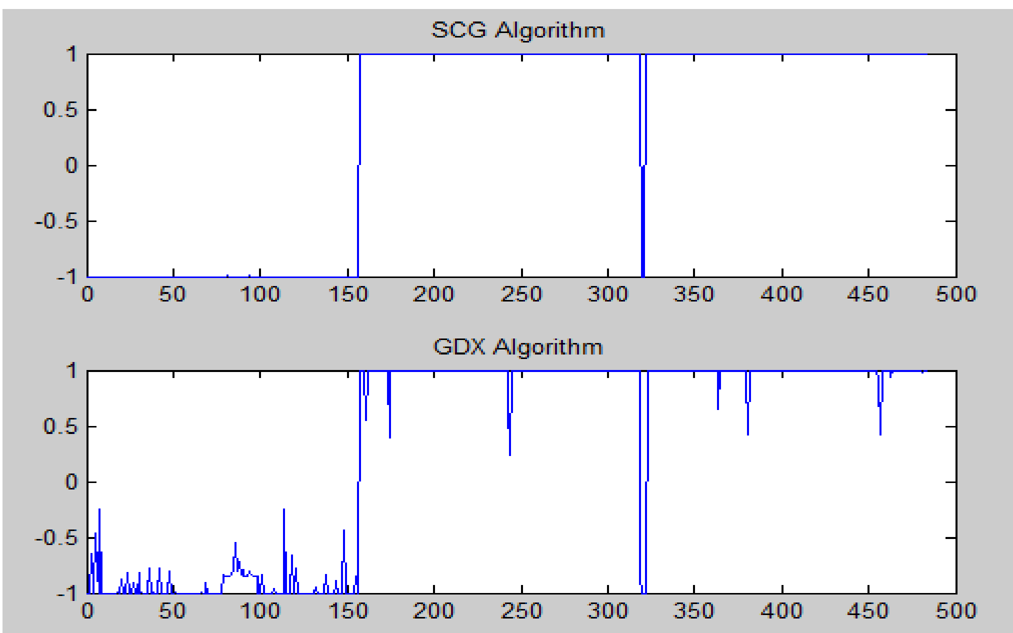


**Figure (3) Response of the neural network using SCG algorithm and GDX algorithm.**

## EDGE DETECTION METHOD

A novel way for detecting edges using the classification system mentioned above will be explained in this section. In this new suggested method the decision needed is between "the pixel belongs to an edge" or "the pixel does not belong to an edge"[11].

| | | |
|---|---|---|
| | x-1,y | |
| x,y-1 | x,y | x,y+1 |
| | x+1,y | |

**Figure (4) Image Window.**

The extracted information from the image needed to find its edge must be put in a form that the neural network classification model can deals with it. The suggested solution is to form a vector with the pixels in a window around each one into the image. The window size is of a 3×3, the four components vector is calculated at each pixel as show in Figure (4). This vector is formed for each pixel given the absolute difference between this one and the pixels in a 3×3 neighborhood around it except the corners of the window. This four components vector is calculated at each pixel in the image except for the border of the image in which the differences cannot be calculated. In the remaining of this paper we shall refer to this vector as the Diff-vector. The formed Diff-vectors are used as inputs to the neural network in the training phase and then to the tested image in the test phase.

### Training Phase

The image model used to train the neural network classification system is shown in Figure (5). This image is designed and created by the authors to obtain a good model for the detection. The pixels considered as edges are those in the border between bright and dark zones, i.e. the points in the bright zone near the dark one and vice versa, this means the edges position is already defined. The bright and dark regions are not homogenous and the intensity at each pixel is a random value (Gaussian values).

In this model the size and the deviation between dark and bright regions must be controlled. The image in Fig. 4 has 24×24 pixels size. This image will be used in the training process; the total number of processed vectors (Diff-vectors) will be 484. The number of vectors defined as Edge vectors is 156, while the number of vectors defined as No Edge vectors is 328 in the training process. The values at each region are designed so that it never reaches those of the other region. In this model the range of pixel values in bright regions is among 150 to 255, while in the dark region is among 0 to 100. By using these random values the detection of false edge will be avoided. The used data in the training process is the absolute difference between dark and bright zones. This data controls the sensibility of the edge detector. A little difference makes the detector more sensible and a greater one reduces this sensibility.

In this phase, as each input is applied to the network, the network output is compared to the target. The error is calculated as the difference between the target output and the network output.

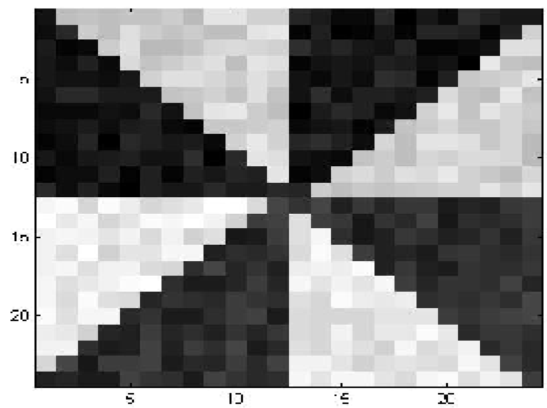

**Figure (5) Training Image.**

The goal is to minimize the average of the sum of these errors. The neural network training performance (the mean square error (MSE)) is shown in Figure (6), in which, the network reaches its best performance (MSE = 9.85× 10-6) at epoch 383.

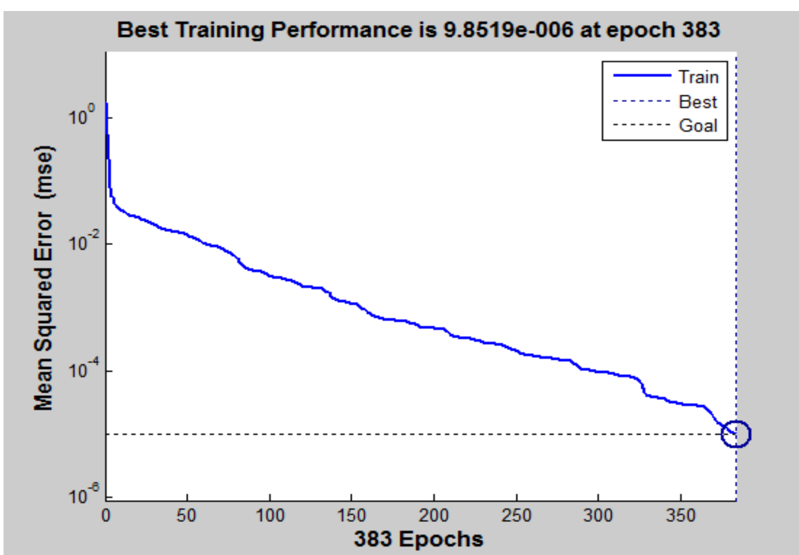


**Figure (6) Neural network training performance.**

To evaluate the classification system, the training image model is used as a test image input to the system. The resulted image is shown in Figure (7).

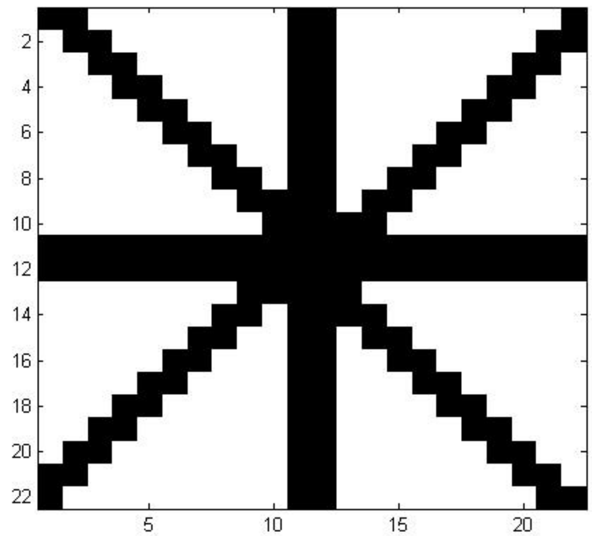


**Figure (7) Training image resulted from the testing phase.**

**Testing Phase**

In this phase, before entering the data of any tested image it is required to rescale its Diff-vectors to be proportional to the values of the trained model vectors. This is done by multiplying the tested image vectors by a scale factor that represents the ratio of the standard deviation of the trained model's Diff-vectors to the standard deviation of the tested image's Diff-vectors.

By applying the classifier that has been gotten from the training phase to any image, a value for each pixel of that image is obtained. This value must be either -1 or +1. The sign of these values indicates whether a pixel is an edge or not.

**RESULTS**

In this work several standard images have been tested by using the proposed system, as example "House" and "Cameraman" images. These images are 256×256 eight bits gray-scaled. Figure (8 and 9) show the original and the output of the system for each of the above mentioned images, in which the output form the neural network classifier is converted to binary images, 1 represents no edge (White) and 0 represents an edge (Black).

**Figure (8) Original image of House and its edge detection.**

**Figure (9) Original Cameraman image and its edge detection.**

Figure (10) shows a comparison between the Canny edge detector and the proposed detector. The image used as example presents a complex texture in the walls due to bricks. Thus, the parameters of the Canny detector have been set trying to reduce the false edge detection of this bricks. We can see how the performance of the method proposed is similar to the Canny edge detector (considered as a standard method) although the proposed method is not using any blurring filter and threshold value like in Canny edge detector.

The training image model can be applied to any image just scaling the input image data by calculating the standard deviation of the Diff-vectors for both of the test image and the training image model.

**Figure (10) Results of Canny Method**

## CONCLUSIONS

In this paper a proposed training image model is created to cover the most types of image edges, this model is put into an artificial neural network system using backpropagation Scaled Conjugate Gradient training algorithm for training this system to take a decision later for classifying any pixel as an edge or no edge pixel. In the proposed method, the standard deviation of the training model must be calculated to be used later in scaling the data set of any tested images.

The proposed edge detection system consists of two phases, phase one is called training phase, in this phase the artificial neural network system is trained through putting the proposed training image model to find the best classifier with minimum number of misclassified pixels. This classifier will be used in phase two which is called testing phase, in this phase the neural network system will be used to classify the scaled pixels of any image as an edge or no edge pixel.

The results of the proposed system are compared with a standard edge detection image method like Canny. The comparison showed that the new method found all the edges, and reduced the number of false edges, as well as it is faster and cheaper than canny method.

## REFERENCES

[1]. Nadernejad, E. "Edge Detection Techniques: Evaluations and Comparisons", Applied Mathematical Sciences, Vol. 2, 2008, no. 31, 1507-1520.

[2]. Hardie, R.C., Boncelet, C.G., "Gradient-based edge detection using nonlinear edge enhancing prefilters", IEEE Trans. Imaging Process, 1995, pp. 1572–1577.

[3]. Kim, D., Lee, W., Kweon, I., "Automatic edge detection using 3x3 ideal binary pixel patterns and fuzzy-based edge thresholding", Pattern Recognition Lett. 25 (1), 2004, pp. 101–106.

[4]. Canny, J.,"A computational approach to edge detection", IEEE Trans. Pattern Anal. Machine Intell. 8, 1986, pp. 679–697.

[5]. Rakesh, R.R., Chaudhuri, P., Murthy, C.A., "Thresholding in edge detection: A statistical approach. IEEE Trans. Image Process. 13 (7), 2004, pp. 927–936.

[6]. Bhandarkar, S.M., Zhang, Y., Potter, W.D., "An edge detection technique using genetic algorithm based optimization. Pattern Recognition 27 (9), 1994, pp. 1159–1180.

[7]. Zheng, S., Liu, J., Tian, J.W., "A new efficient SVM-based edge detection method", Pattern Recognition Lett. 25 (10), 2004, pp. 1143–1154.

[8]. Gonzalez R. C. and Woods R. E., "Digital Image Processing", 2nd Edition, Prentice Hall, 2002.

[9]. Nadernejad, E. "Edge Detection Techniques: Evaluations and Comparisons", Applied Mathematical Sciences, Vol. 2, 2008, no. 31, pp. 1507-1520.

[10]. Demuth, H. M. Beale, and M. Hagan "Neural Network Toolbox™ User's Guide", The MathWorks, Inc., 2009.

[11]. Gómez-Moreno, H., Maldonado-Bascón, S., López-Ferreras, F., Edge detection in noisy images by using the support vector machines. IWANN, *Lecture Notes on Computer Science*, Vol. 2084. Springer-Verlag, Heidelberg, 2001, pp. 685-692.