Quad-pixel edge detection using neural network

Hamed Mehrara , Mohammad Zahedinejad
Young Researchers club,Science and Research Branch,
Islamic Azad University,Tehran,Iran
h.mehraraa@gmail.com , mohamad.zahedi@gmail.com

Received: 2011/02/10 ;Accepted: 2011/04/15 Pages: 47-54

Abstract

One of the most fundamental features of digital image and the basic steps in image processing, analysis, pattern recognition and computer vision is the edge of an image where the preciseness and reliability of its results will affect directly on the comprehension machine system made objective world. Several edge detectors have been developed in the past decades, although no single edge detectors have been developed satisfactorily enough for all application. In this paper, a new edge detection technique was proposed based on the BP neural network. Here, the edge patterns of a quad-pixel in binary images were classified into 16 possible types of visual patterns. In the following, after training the pre-defined edge patterns, the BP neural network was applied to correspond any type of edges with their related visual patterns. Compared with traditional edge detection techniques, the results demonstrate that the new proposed technique, improved the computations mass and mathematical complexity, turns out better.

Keywords: *Edge detection, Image binarization, Image processing, Neural Networks, Threshold*

1. Introduction

In image processing and computer vision, edge detection is a process which attempts to capture the significant properties of objects in the image. These properties include discontinuities in the photometrical, geometrical and physical characteristics of objects. Such information give rise to variations in the grey level image; the most commonly used variations are discontinuities (step edges), local extrema (lines edges), and 2D features formed where at least two edges meet (junctions) [1]. The purpose of edge detection is to localize these variations and to identify the physical phenomena which produce them.

Up to now, many edge detection techniques, such as first derivative algorithm [2]-[4], second derivative algorithm [5], template matching [6], [7], edge fitting, and statistical approaches [8], have been proposed, and several commercial systems are on the market. As the validity, efficiency and possibility of the completion of subsequent processing stages rely on edge detection, it must be efficient and reliable. To fulfill this requirement, edge detection provides all significant information about the image. For this purpose, image derivatives are computed but image derivatives are sensitive to various sources of noise, i.e., electronic, semantic, and discretization/quantification effects. To regularize the differentiation, the image must be smoothed. However, there are undesirable effects associated with smoothing, i.e., loss of information and displacement of prominent structures in the image plane. Furthermore, the properties of commonly-used differentiation operators are different and therefore they generate different edges.

It is difficult to design a general edge detection algorithm which performs well in many contexts and captures the requirements of subsequent processing stages. Consequently, over the history of digital image processing a variety of edge detectors have been devised which differ in their purpose and their mathematical and algorithmic properties [1]. This research proposed a heuristic approach, which detects edges of an image most efficiently. The key features of our approach which differentiate us from others is the use of image content simulated with Artificial Neural Network (NN) for edge detection of application-specific image. The proposed techniques can be extended for color images as well. This paper described the characteristics of edges, the properties and the methodology of the proposed edge detection approach. From section 2 up to section 5, related work and new edge detection method were introduced, and summarized its results. Finally, this paper analyzed the proposed method advantages and disadvantages.

2. Related Works

There are many methods for edge detection, but most of them can be grouped into three categories, search-based, zero-crossing based and threshold base. The search-based methods detect edges by first computing a measure of edge strength, usually a first-order derivative expression such as the gradient magnitude, and then searching for local directional maxima of the gradient magnitude using a computed estimate of the local orientation of the edge, usually the gradient direction. Edge detection based on second-order difference (zero crossings) was strongly influenced by biological vision. Types of edge detectors based on thresholding can be grouped into two classes: (a) local techniques, which use operators on local image neighborhoods and (b) global techniques, which use global information and filtering methods to extract edge information [9]. Both methods have their advantages and disadvantages on various types of images. Nearly all detectors utilize thresholding of the image for edge detection. Each pixel in the image is compared with this threshold value. If the pixel's intensity is higher than the threshold value, the pixel will set to White in the output image. If it is less than the threshold, it will set to Black. The efficient selection of single threshold value is the most important and the most difficult process in edge detection technique.

Edge detectors based on local techniques, use local feature for selecting threshold value. Similarly, edge detectors based on global techniques, use global feature for selecting threshold value. An image contains variations at different levels. Using single global threshold over the whole image gives poor results. But local threshold method also detects false edges due to noise. The global threshold value depends on the presence of noise in the image.

More recently there have been several papers published on the use of neural networks for edge detection [10-14]. Paik, Brailean, and Katsaggelos considered multi-state ADALINES for edge detection. Li & Wang [15] proposed algorithm used bit plane slicing to binarize gray level image , an optimized neural network using 3x3 sliding window is trained to extract edge of any bit plane and finally are weighted to detect edges accurately by the parallel model. Terry & He [16,17] proposed algorithm used NN by introducing primitive and constrained pattern of each edge map to train the network ,also Basturk [18] exert cellular NN for this purpose.

3. The Proposed Method

Neural networks can be a useful tool for edge detection, since a neural network edge detector is a nonlinear filter. An edge-detection 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. A simple method is recommended for the edge detection training problem. In the subsections the fundamental and implementation of this new method by NN was described.

A. Identify the Neural Network Algorithm

The suggested algorithm was shown in Figure 1 to detect edges in a grey level image, Firstly, the image was binarized by modified Otsu's method threshold value. Researchers have proposed a number of techniques to improve selecting thresholds or to provide some criteria for optimal decisions for threshold selection. No local threshold will solve this problem [19-22].Otsu [23] suggested minimized group variances for the probable distribution of gray value as optimal criteria. Different from Otsu's suggestion, Kittler and Illingworth [23] used a mixture of two Gaussian distributions, by adjusting the proportions of two distribution, to approximate histograms. As shown in Figure 8, some thresholding methods (e.g. mean value, median value…) contain so much detail that computer vision recognizer doesn't require them and sometimes make identifying error.

Best threshold value which conclude better output image is Otsu's method empirically. Base on histogram, Otsu selects single gray level (index) for clipping intensity but this method loses some detail in up and down of this level. In modified Otsu , gray level under the index stretched by scaled gamma function above it and above the index stretched down Figure 2. This will reduce the contrast in an image, but shrieked histogram is capable to reconstruct eliminated pixels under and above index. By applying slope of the line used to modify the gray levels within the specified range, image evolved. The slope may be positive, negative, or zero which negative and zero slope will cause all values in the given range mapped to the same gray level value. Finally binary image will be produced.

Binary image disintegrates in 2x2 windows and generates a set of image pattern and then edge patterns were classified in binary images into 16 categories, as shown in Figure 3a and the neural network on these patterns was trained. In Figure 3, the blank elements in each 2x2 window indicate white (pixels va1ue :1s) in binary images, whereas the dark elements indicate black (pixels value: 0s). To extract edges from a binary image, the four pixel (quad-pixel) window was used in output pattern that keep all pattern except black-black-black-black (binary code: 0000) and white-white-white-white (1111). These patterns do not include intensity variation and aren't edge points and neural network returning white-white-white-white (Whitewash) for both. The noisy patterns can also be reduced by Whitewash. Noisy patterns contain one black pixel, replaced by Whitewash (see Table. 1). Thinning procedure also was done by whitewashing one black pixel of (1000), (0100), (0010), (0001) that makes these patterns diagonal. In Figure 3b output patterns represented. After the network is trained, it can recognize the input pattern as the most similar pattern in the edge pattern bank.

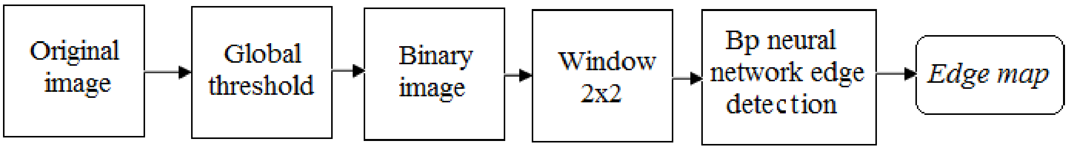

Figure 1.Proposed algorithm

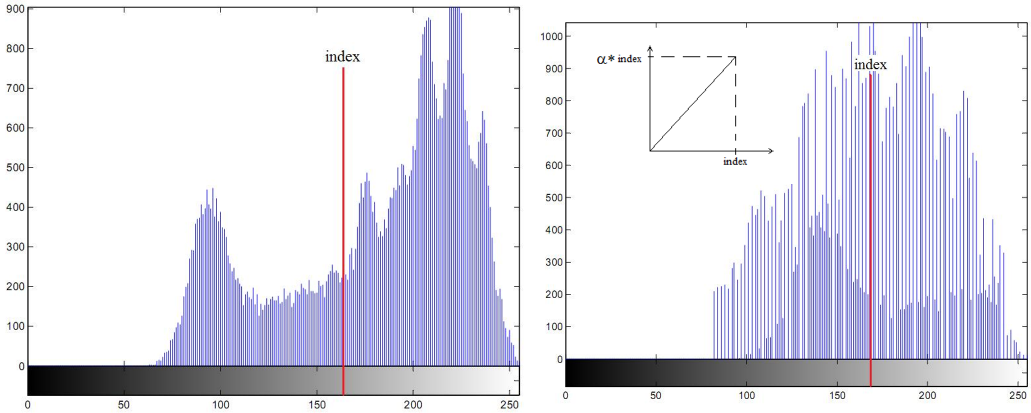

Figure 2. Histogram shrinks by α constant

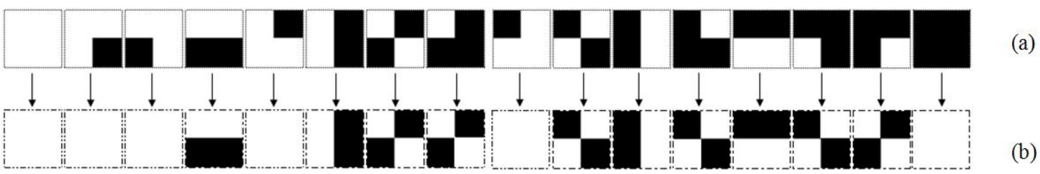

Figure 3.All possible type of input patterns and output of neural network

B. Designed Neural Network

The selection of images for the algorithm has some special characteristics because binary images are the input to our algorithm. Log-sigmoid function was used so that output extreme from a network node are 0/1. Training can be accomplished by preparing a dataset in the following manner: Take an image object to be learned and slide it from point to point across all locations of a window which will be the input window to the pattern detection network. 2x2 windows were used because all other windows reduce detail and include more training set (2 Window Size) but this is efficiently simple and accurate. In the system pixel values range from 0 to 1, usually 0 represents black on the display, at every pixel location. If the detection window is 2x2 pixels to be learned with overlapping, then there will be 2 (2x2) patterns to train on.

As neural networks go this is a small training sample, so training is easy. The network structure for this example could be 4xHx4 (4 inputs, H hidden and 4 outputs). In experiment H can be equal to 12. Training will converge in less than 6 epoch using Levenberg-Marquardt back propagation (more memory efficient) or in 1.1537 second than a minute by using conjugate gradient with momentum and adaptive learning rate function minimization. The network was trained on 16 pre-defined edge patterns as in Figure 4, input layer represented sliding window elements and those which capitalize, demonstrated output layer. All training were done by use of error back propagation learning rule with a learning rate η and a momentum μ (in the experiment η =0.01 and μ =0.9). It was evidently observed that the application of momentum can effectively prevent the training progress from local minima, although the selection of the momentum value was a trial and error procedure. When the whole binary image was scanned by the four window pixels, the edges pattern would be obtained. Note that the image was processed four pixels by four pixels, and the windows were not overlapped i.e. for covering all pixels this process was used 4 times; initially the start position was (x_1y_1, x_2y_2, x_1y_2, x_2y_2) that all obtained images finally were multiplied pixel-by-pixel. This reduced the computation time dynamically. It was also noticed that the two margin rows and two margin columns around the image cannot be processed and a kind of padding required. But, in practice, the effect of margin pixels on the whole image can be ignored.

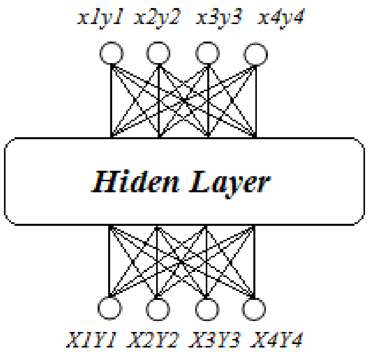

Figure 4. Input pattern and proposed neural network

Table 1. Proposed neural network and input pattern

Decimal code	Input pattern	Detected as	Output pattern
1	0 0 0 0	None edge	1 1 1 1
2	0 0 0 1	Corner edge	1 0 0 1
3	0 0 1 0	Corner edge	0 1 1 0
4	0 0 1 1	Horizontal edge	0 0 1 1
5	0 1 0 0	Corner edge	0 1 1 0
6	0 1 0 1	Parallel edge	0 1 0 1
7	0 1 1 0	Diagonal edge	0 1 1 0
8	0 1 1 1	Pseudo noise	1 1 1 1
9	1 0 0 0	Corner edge	1 0 0 1
10	1 0 0 1	Diagonal edge	1 0 0 1
11	1 0 1 0	Parallel edge	1 0 1 0
12	1 0 1 1	Pseudo noise	1 1 1 1
13	1 1 0 0	Horizontal edge	1 1 0 0
14	1 1 0 1	Pseudo noise	1 1 1 1
15	1 1 1 0	Pseudo noise	1 1 1 1
16	1 1 1 1	None edge	1 1 1 1

4. Results and Discussions

In this section, the proposed method is applied to some representative images. The performance of proposed method is compared with Canny, Roberts, Prewitt and Sobel method. The following detections executed on Matlab and Figure 5, 6 and7 showed that proposed method had good output performance. Roberts's operator has bad continuity on the contour of the images. Canny operator has distortion on the contour of the image. Sobel operator has better performance, but the some image part is incomplete. The experiment has been executed on an Intel core2 Duo 2.5GHz processor and 3GB RAM computers. The detection results showed that all the well-known Canny, Roberts, Prewitt and Sobel edges have somewhat distortion on the binary image; while the new method had better visual performance such as the detail of the characters. The use of the NN algorithm was demonstrated by showing the effects of the parameters. By increasing α, details of hidden objects extend and by decreasing α to zero and under zero, basic Otsu thresholding returned. This also demonstrates the importance of selecting α correctly. If α is set too high, then the important image edges are lost.

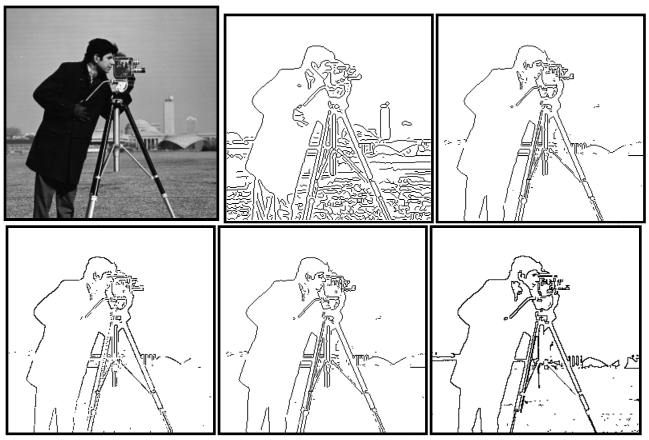

Figure 5. Edge map: Cameraman image, canny edge, Sobel edge, Roberts's edge, Perwit edge, new detector

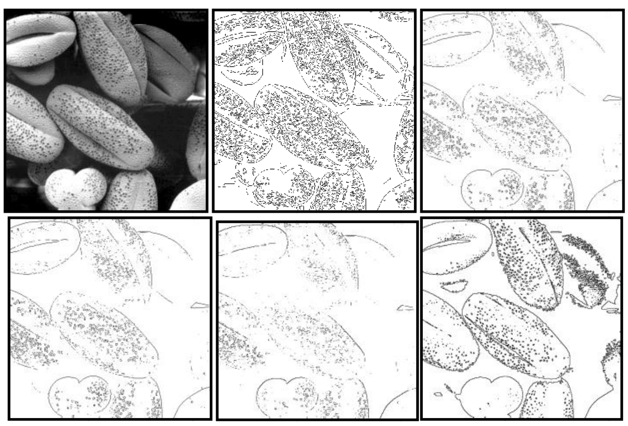

Figure 6. Edge map: Original image, canny edge, Sobel edge, Roberts's edge, Perwit edge, new detector

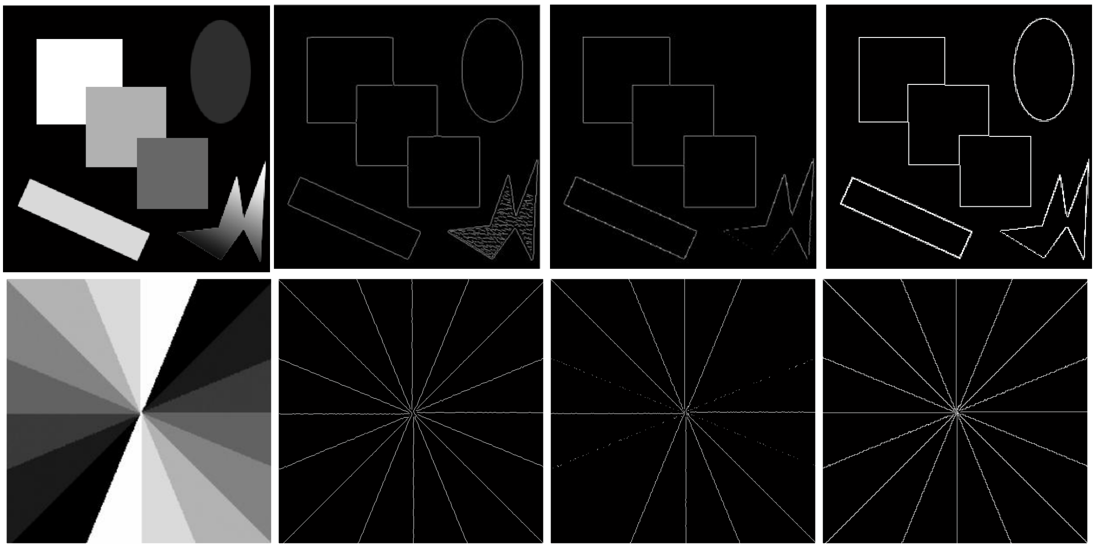

Figure 7. Original image, canny edge, Sobel edge, new detector

Figure 8. Binary image: Global threshold (128), Median 3x3 thresholds, Mean 3x3 thresholds, Otsu's method

5. Conclusion

In this paper, a novel edge detection approach was described. A number of experiments were conducted and the results showed that the designed neural network simply converging because of small training sets. The proposed approach was superior to traditional edge detection operators as it solved the problem of difficult convergence if the BP neural network was used directly for edge detection of gray image because a huge training sample set was needed. To obtain better edge detection, cautiously selecting of threshold value for binarizing the grey level was recommend. The future work concentrates on local thresholding to make important details obvious .

References

1. D. Ziou and S. Tabbone, "*Edge Detection Techniques –An Overview*" TR-195, Département de Math et Informatique,Université de Sherbrooke, Québec, Canada,1997, pp. 1-41.
2. Fang J and Huang T S, "*Some experiments on estimating the3-D motion parameters of a rigid body from two consecutiveimage frames*", IEEE Trans. Pattern Analysis Machine Intell.,PAMI-ti-5 ,pp545-554,1984
3. Roberts,L.G.,"*Machine perception of three-dimensional sol-ids*", Optical and Electro-Optical Information Processing, J. T.Tippett et. al(eds.),MIT Press:Cambridge,ppl59-197, 1965
4. Sobel, I.,"*Camera models and machine perception*", StanfordAI Lab,AIM-2l,ppl21, 1970
5. Marr, D. and Hildreth, H.,"*Theory of edge detection*", Proc of the Royal Society of London, B-207,pp187-217, 1980
6. Kirsch, R., "*Computer determination of the constituent struc-ture of biological image*", Computer and Biomedical Reserch,4-3,pp315-328, 1971
7. Prewitt, J.M.S.,"*Object enhancement and extraction*", Pic-ture Processing and Psychopictoris B.Lipkin and A. Rosen-feld(eds.),Academic Press:Cambridge,pp75-149, 1970
8. Yakimovsky, Y., "*Boundary and object detection in real world images*", Journal of ACM, 234,pp599-618, 1976
9. M.Bilal Ahmad and T.Choi,. "*LOCAL THRESHOLD AND BOOLEAN FUNCTION BASED EDGE DETECTION*", IEEE Transactions on Consumer Electronics, Vol. 45, No. 1, 1999
10. L. Moura and F. Martin, "*Edge detection through cooperation and competition*," Proc. IJCNN 1991. Singapore, Vol. 3, pp.2588-2593.
11. Lepage and D. Poussart, "*Multi-resolution edge detection*," Proc. IJCNN 1992, Baltimore, Vol. 4. pp. 438-443.
12. P. Bhatia, V. Srinivasan and S.H. Ong, "*Single layer edge detector with competitive unsupervised learning*," Proc. UCNN 1991. Singapore, Vol. 1, pp. 634-639.
13. J. K. Paik, J. C. Brailean and A.K. Katsaggelos, "*An edge detection algorithm using multi-state adalines* ", Pattern Recognition, Vol25-12, pp. 1495-1504.1992.
14. K. Etemad and R. Chellapa, "*A neural network based edge detector*," Proc. 1993 IEEE Conference on Neural Networks, Vol 1, pp. 132-137.
15. W. Wang and Qun Wang and G.Chen, "*An Edge Detection Method Based on Optimized BP Neural Network* " 2008 International Symposium on Information Science and Engineering, 12-15 July 2008,DOI 10.1109/ISISE.2008.310, 978-0-7695-3494-7/08 / IEEE 2008
16. P. John Terry and Duc Vu, "*Edge Detection Using Neural Networks*" US. Government Work Not Protected by U.S. Copyright
17. Zhengquan He and M Y. Siyal, "*Edge Detection With BP Neural Networks*" 0-7803-4325-519861,proceeding of ICSP ` 98
18. A.Basturk and E.Gunay, "*Efficient edge detection in digital images using a cellular neural network optimized by differential evolution algorithm*" Elsevier ,Expert Systems with Applications 36 2645–2650 (2009)
19. A.Bilal Ahmad and T.Choi, "*LOCAL THRESHOLD AND BOOLEAN FUNCTION BASED EDGE DETECTION*" IEEE Trans. on Consumer Electronics, Vol. 45, No. 1, AUGUST 1999
20. M. Vemis , "*The use of Boolean functions and local operations for edge detection in images*" ELSEVIER, Signal Processing 45 (1995) 161-172 K.Ohtani and M.Baba, "*A Fast Edge Location Measurement with Subpixel Accuracy Using a CCD Image*" February 10, 2007 , Vol. 5, No. 2 ,CHINESE OPTICS LETTERS, 20071
21. L.Jiang, "*Image edge detection based on adaptive weighted morphology*" IEEE Instrumentation and Measurement Technology Conference Budapest, Hungary, May 21-23, 2001
22. N. Otsu, "*A threshold selection method form gray- level histogram*", IEEE Trans. System, Man, and Cybernetics, Vol. SMC-9, 1979, pp.62-66.
23. J. Kittler and J. Illingworth, "*On threshold selection using clustering criteria*", IEEE Trans. System, Man, and Cybernetics, Vol. SMC-15, 1985, pp652-655.