Edge Detection Based on Standard Deviation Value and Back Propagation Algorithm of Artificial Neural Network

Ali Hussien Miry*, Mohammed Hussien Miry*
& Ammar Sabr Majed*

Received on: 25/2/2010
Accepted on: 3/2/2011

Abstract

This paper presents a proposed neural network based edge detection algorithm. we have used artificial neural network system to decide about whether each pixel is edge or not. First standard deviation values are computed for mask (3*3), Then after training a neural network system to recognize structural patterns (these pattern represents edges), it decides on each pixel if its edge or not. Finally we have test the proposed method on different images. Experimental results show the ability and high performance of proposed algorithm.

Keywords: Edge Detection; Neural Network, and Standard Deviation.

اكتشاف الحافات اعتماد على مقياس الانحراف و خوارزمية الانتشار الخلفي لشبكات العصبية الذكية

الخلاصة

هذا البحث يقدم طريقة مقترحة لاكتشاف الحواف بالاعتماد على الشبكة العصبية. حيث نعتمد على ما تقرره الشبكة العصبية الذكية في اعتبار النقطة حافة أو لا.في البداية نأخذ قناع (3نقاط *3 نقاط) ونحسب الفرق بين قيمة كل نقطة في القناع و معدل النقاط وذلك لغرض لمعرفة ما إذا كانت النقطة الموجودة في مركز القناع مرشحة إن تكون حافة أو لا .بعد ذلك تدخل الفرو قات المحسوبة في الخطوة السابقة إلى الشبكة العصبية المدربة على أشكال خاصة تمثل حافات في الصورة. والقرار النهائي يكون إلى الشبكة العصبية لتقرر فيما إذا كانت النقطة حافة أو لا. أثبتت الطريقة المقترحة قدرتها على اكتشاف الحافات بصورة جيدة وقدرتها أيضا على التمييز بين الضوضاء والحافة وهذا ما يعطيها فائدة كبيرة

1. Introduction

Edge detection is one of the most important algorithms in image processing. It plays a fundamental role in higher level processing [1, 2]. .Edge detection is an intermediate step in image pattern recognition in digital images and has the opposite effect of noise elimination; it consists on emphasizing pixels with gray tones That are different than their neighbors [3]. In a grey level image, an edge may be defined as a sharp change in intensity and Edges in images are areas with strong intensity contrasts, a jump in intensity from one pixel to the next [2]. Edge detection is the process of detecting the presence and location of these intensity transitions. The edge representation of an image drastically reduces the amount of data to be processed, yet it retains important [4]. Edge detection is extensively used in image processing applications to separate the object from the background in images. Using edges, significant amount of data can be reduced by filtering out the useless information while preserving the

*** Al-Khwarizmy College of Engineering, University of Baghdad/Baghdad**
*** Electrical & Electronic Engineering Department, University of Technology /Baghdad**
*** Education Center, University of Baghdad/Baghdad**

structural properties of an image. A lot of research has been done in the field of image segmentation using edge detection. Some of the earliest operators to detect edges in an image were proposed by Sobel, Prewitt, Roberts, etc [5].Edge is defined as object border, and extracted by features such as gray, color or texture discontinuities [5]. Luminance and geometrical features, lightening condition and noise volume has a great impact on shaping the edge. Edge contains important information of image and provides object's location.

2. Artificial Neural Network Model

An artificial neural network is an information-processing system that has certain performance characteristics in common with biological neural network]. It can also be defined as parallel-distributed information processing structure that has an interconnection of neurons [6,7], such that neuron outputs are connected through weights to all other neurons including themselves. Thus, a neural network consists of several interconnected processing elements or neurons. Each processing element has its own interconnection and local memory to the value of some previous computations, [6]. Each neuron model consists of a processing element with synaptic connections and single output.The usual neural networks, such as feed forward [3] neural networks, are employed to recognize structural patterns.

2.1 Back Propagation Training Algorithm (BP)

The type of neural network used in this approach is the multilayer neural network (MLNN) with the sigmoid function as activation function. This type of network consists of an input layer, an output layer, and one or more hidden layers (Figure 2). The choice of MLNN is based on the fact that this type of network is a supervised neural network that learns through a process called back propagation, which is a form of gradient descent, which is suitable for our problem. Here the network is supplied with series of input and corresponding correct (desired) output. The network then tries to set its own parameter until it can approximate an unknown function that can associate input data with corresponding desired output [8,9].

2.1.1 BP Program-Training Process:

Step 1: Design the structure of neural network and input parameters of the network.

Step 2: Get initial weights W and initial θ (threshold values) from randomizing.

Step 3: Input training data matrix X and output matrix T.

Step 4: Compute the output vector of each neural units.

(a) Compute the output vector H of the hidden layer

$$net_K = \sum w_{ik} x_i - \theta_k \qquad (1)$$

$$H_k = f(net_k) \qquad (2)$$

(b) Compute the output vector Y of the output layer

$$net_j = \sum w_{kj} H_i - \theta_j \qquad (3)$$

$$Y_j = f(net_j) \qquad (4)$$

(c) Compute the root of mean square

$$RMS = \sqrt{\frac{\sum (y_i - T_j)^2}{n}} \qquad (5)$$

Step 5: Compute the distance δ

(a) Compute the distance δ of the output layer

$$\delta_j = (T_j - Y_j) f^{'}(net_j) \qquad (6)$$

(b) Compute the distance δ of the hidden layer

$$\delta_k = (\sum_j \delta_j w_{kj}) f^{'}(net_j) \quad (7)$$

Step 6: Compute the modification of W and θ (η is the learning rate, α is the momentum coefficient)

(a) Compute the modification of W and θ of the output layer

$$\Delta w_{kj}(n) = \eta \delta_j H_k + \alpha \Delta w_{kj}(n-1) \quad (8)$$

$$\Delta \theta_j(n) = -\eta \delta_j + \alpha \Delta \theta_j(n-1) \quad (9)$$

(b) Compute the modification of W and θ of the hidden layer

$$\Delta w_{ik}(n) = \eta \delta_k X_i + \alpha \Delta w_{ik}(n-1) \quad (10)$$

$$\Delta \theta_k(n) = -\eta \delta_k + \alpha \Delta \theta_k(n-1) \quad (11)$$

Step 7: Renew W and θ

(a) Renew W and θ of the output layer

$$w_{kj}(p) = w_{kj}(p-1) + \Delta w_{kj} \quad (12)$$

$$\theta_j(p) = \theta_j(p-1) + \Delta \theta_j \quad (13)$$

(b) Renew W and θ of the hidden layer

$$w_{ik}(p) = w_{ik}(p-1) + \Delta w_{ik} \quad (14)$$

$$\theta_k(p) = \theta_k(p-1) + \Delta \theta_k \quad (15)$$

Step 8: Repeat step 3 to step 7 until converge.

BP Program-Testing Process:

Step 1: Input the parameters of the network

Step 2: Input the W and θ

Step 3: Input the unknown data matrix X

Step 4: Compute the output vector

(a) Compute the output vector H of the hidden layer (according to equation (1) and equation (2))

$$net_K = \sum w_{ik} x_i - \theta_k$$

$$H_k = f(net_k)$$

(b) Compute the output vector Y of the output layer(according to equation (3) and equation (4))

$$net_j = \sum w_{kj} H_i - \theta_j$$

$$Y_j = f(net_j)$$

3. The Proposed Method

The proposed method consists of two steps: the first step performs edge extraction while the second step performs edge detection. The powerful of this method is assaying all the pixels of the processed image by studying the situation of each neighbor of each pixel. The condition of each pixel is decided by using a floating 3x3 mask which can be scanning the all grays. Therefore theses two steps are applied for each pixel in the image.

Step-1-Primary Edge Extraction

In this paper, first based on **Standard Deviation** (SD) of pixels value, edges are separately extracted then based on neural network, final decision about whether each pixel is edge or not is made. In this method, for each pixel SD is computed by 3×3 mask, over adjacent neighborhood pixels. Similarly pixels with DS grater than a threshold value are edge candidate.The final decision about edge candidate pixels is based on the threshold value. If the difference between SD and pixel value is greater than a threshold value, then, the corresponding pixel is classified as White; otherwise, the pixel is classified as Black. The pixels that classified to white are candidate to become edge while the other pixels candidate to become background. For example, we apply this step to the (3x3) sub image in Fig. 2-a. Figure 3 shows sub images contain edges and intensity of the edge and the pixels neighboring of this edge. The pixels values equal one in the masks shown in Figure 3 refers to white while value of pixels equal zero refer to black. The center of each mask lies between white region and black region. Therefore the pixel in an image represent edge if its center of one of following mask (after applying this step on the neighboring of this pixel) and theses mask are used in neural network to train the network to detect the edge.

Step-2- Edge Detection Based on Neural Network

The output of BP Neural network explains to how extent a pixel could be edge**.** After learning a neural network, the output of this neural network is classified to one of two classes. The first class, correspond to pixels with low probability value (low probability means the output of the ANN is less than the threshold) to belong to edge pixels set, while the second class correspond to high probability to belong to edge pixels set.A pixel with probability grater than a threshold is classified as edge; otherwise it is classified as background. A typical appropriate threshold value is 1/2. ANN will gives an output "1" if detects an edge otherwise it gives "0". Therefore the ANN will trained to give "1" if the one of the masks shown in Figure. is detected during scanning the image . These masks corresponding to an edge in the image. Here, each neuron input is a SD value at an image pixel. For the output of a neuron at the centre of a 3×3 cluster of image pixels, 9 neurons are selected as inputs. Each input neuron is a surrounding pixel of the central pixel. The nine neurons are the inputs of the first input layer.The detection of edge depends on the SD of the eight neighbors gray level pixels, if the neighbors SD are blacks or of whites. In this section, some of the desired masks are explained. Fig. 3 and Fig. 4 show some cases for vertical, horizontal, diagonal, and corner edges in some images and the value of the neighbors of the edge pixel. if the grays(SD) represented in one line are black and the remains grays are white then the checked pixel is edge (Fig.4-a).Fig. 5 shows the Flowchart of Proposed Method.

4. Experimental results

The proposed algorithm is simulated in MATLAB and applied on several test images. We used a neural network with three layers, with 9 neurons in the input layer, 9 neurons in the hidden layer and 1 neuron in the output layer, for classification of the input data. A simple back propagation algorithm is used to update weights according to desired values. The main images and the resultant images are show in figure 6. The same experiment run over other test images are shown in this Figure. Similarly, extracted edges by proposed methods clearly show the superiority of the proposed algorithm.

5. Conclusions

This paper described a new method for edge detection based on neural network. A new edge detection scheme based on multiple masks been presented. This technique uses the edge strength information derived using different masks instead of using a single mask as used in most of the edge detection techniques. The advantage of using multiple masks is that it avoids detection of edges corresponding only to spurious noise, which is often the case with single-mask edge detection. Also, another advantage of the proposed method, it can recognize more kinds of edges than with other methods by learning the neural network on all edge-orientation because a Neural networks have an excellent capability to recognize specific patterns.

References

[1] V. Mittal and M. S. Batra, "Edge Detection Technique By Using Nero Fuzzy System," Proceedings of 2nd National Conference on Challenges & Opportunities in Information Technology (COIT-2008), RIMT-IET, Mandi Gobindgarh. March 29, 2008.

[2] R. Gonzalez and R. Wood, Digital Image Processing, Addison – Wesley, 1992.

[3] Olivia Mendoza1 , Patricia Melin, Juan R. Castro1 "The Use of Interval Type-2 Fuzzy Logic as a General Method for Edge Detection "IFSA-EUSFLAT2009

[4] Lesa. M. Kennedy Mitra Basu. "Image Enhancement Using Human Visual System Model ,"Pattern Recognition ,Vol ,30. NO ,12. pp 2001.-2014, 1997

[5] M. Hanmandlu, Rohan Raj Kalra, Vamsi Krishna Madasu "Area Based Novel Approach For Fuzzy Edge Detection" 2006 IEEE.

[6] R. Wang, L. Gao, S. Yang, and Y. Liu, "An Edge detection method by combining fuzzy logic and neural networks," Machine Learning and Cybernetics, 7(2005) 4539-4543, 18- 21, Aug 2005.

[7] W. Yu and A. Ferreyra,"System Identification with State-Space Recurrent Fuzzy Neural Networks " pp ,5111 -5106.FrC023,43.rd IEEE Conference on Decision and Control December. 2004 ,17-14

[8] D. C. Reddy and K. Ghosh, "Identification and Interpretation of Manufacturing Process Patterns Through Neural Networks", Mathl.Comput. Modeling, Vol. 27, No. 5, pp. 15-30, 1998.

[9] B.S. Murdianto1, R. Sutattyo, A. Haris," Predicting Lateral Sonic Log Using Artificial Neural Network and Multiattribute Transform,"Seminar Nasional MIPA, 2005 FMIPA– Universitas Indonesia Depok J1E-04, 24-26 November 2005.

[10] M. Chang and C. To ," Comparison of Support Vector Machine and Back Propagation Neural Network in Evaluating the Enterprise Financial Distress" International Journal of Artificial Intelligence & Applications (IJAIA), Vol.1, No.3, July 2010

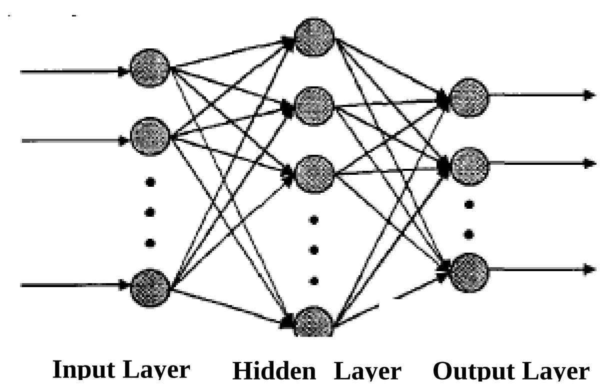

Figure 1: The BP Neural Network

200	50	210
56	60	56
250	190	45

1	0	1
0	0	0
1	1	0

Figure 2: a)3x3 Mask b)Mask after Determine SD and Thresholding

1	0	0
0	0	0
1	1	0

1	0	1
0	0	0
1	1	0

1	0	1
0	0	0
1	1	0

1	0	1
0	0	0
1	1	0

1	0	1
0	0	0
1	1	0

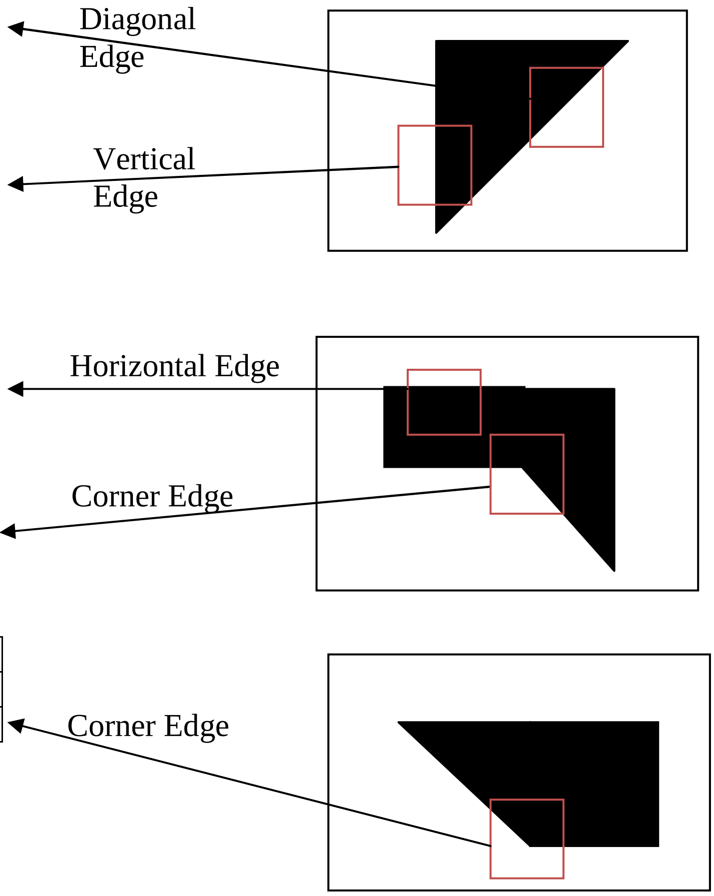

Figure 3: Sub Images Contain Edge and Corresponding Mask for Edge Region

1	0	0
0	1	1
0	0	0

0	0	1
0	1	1
0	0	1

1	0	1
0	1	0
0	0	0

0	0	0
1	1	0
0	1	0

0	1	0
1	1	0
0	0	0

0	0	0
0	1	0
1	0	1

0	0	0
1	1	0
0	0	1

1	0	0
0	1	1
0	0	0

0	0	0
0	1	1
1	0	0

Figure. 4: Masks Used in Training Neural Network

[i-1, j-1]	[i-1, j]	[i-1, j+1]
[i, j-1]	**[i,j]Checked Pixel**	[i, j-1]
[i+1, j-1]	[i+1, j-1]	[i+1, j-1]

Floating Mask 3*3

I[i-1, j-1]	I [i-1, j]	I [i-1, j+1]
I [i, j-1]	**Checked Pixel**	I [i, j-1]
I [i+1, j-1]	I [i+1, j-1]	I [i+1, j-1]

Original Image

Determine SD and Thresholding →

p[i-1, j-1]	p [i-1, j]	p [i-1, j+1]
p [i, j-1]	**Checked Pixel**	p [i, j-1]
p [i+1, j-1]	p [i+1, j-1]	p [i+1, j-1]

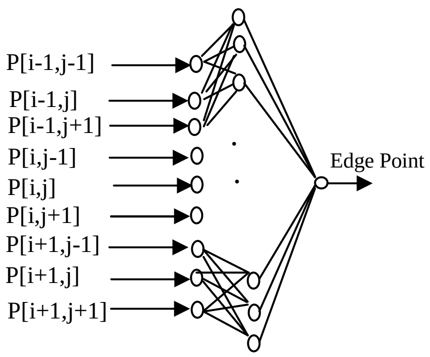

Figure 5: Proposed Method

(a) (b) (c)

Figure. 6.: a) Main image, b) Sobel results, c) Proposed Neural algorithm results