ATLANTIS PRESS

Advances in Engineering Research, volume 114

International Conference on Advances in Materials, Machinery, Electrical Engineering (AMMEE 2017)

Research on Image Edge Detection Algorithm Based on Eigenvector and Improved BP Neural Network

Mingyang Yu [a] and Xiaoyu Huang [b]

School of Information Science and Engineering, Central South University, Changsha 410075, China.

amingyang@csu.edu.cn, b1741110589@qq.com

Keywords: Edge detection, eigenvector, BP neural network, improved algorithm, filter out noise.

Abstract. The purpose of edge detection is to distinguish different regions of the image, usually the edge information is determined by the gray-scale change between regions. In the image with noise, the traditional edge detection method is easy to determine the noise as the edge point. While adding filtering measures can reduce noise, it also reduces edge information. In this paper, an edge detection method based on eigenvector and improved BP neural network is proposed. The eigenvector of the pixel is composed of differential and median value of the grayscale. Select the sample image to extract the eigenvector of each pixel and enter the BP network for training. Through the error between output value and the tutor signal to adjust network parameters. The improved BP network with learning rate self-regulation and momentum factor can improve the network performance [1]. After the training is completed, use the *Cameraman* image for edge detection and compared with the traditional detection methods. Experiments show that the method can remove the noise while preserving the edge.

1. Introduction

Image edge detection is the most basic part of image analysis, has a very important significance in the digital image processing. Many scholars at home and abroad have done a lot of research on this, and proposed a classic first-order edge detection operator based on gray gradient, such as Roberts, Sobel, Prewitt, etc; and second-order edge detection operator based on linear filtering, such as LOG, Canny and so on [2]. These classic operators still have a wide range of application value today, but their limitations are gradually reflected, such as the classic first-order operator only has a better detection effect on simple structure images, but unsatisfactory on the detection of complex images, some weak edges are often being missed. Although the second-order detection operator can detect the details of the image, but will be mistakenly to detect some non-edge points, it's relatively weak for the high-frequency signal noise. For this reason, the image edge detection have given rise to a prominent contradictions that algorithm noise immunity and detection accuracy is difficult to guarantee.

In recent years, with the rapid development of computer science, many advanced intelligent algorithms have been proposed and used in various fields of research, and made remarkable contributions to the results. The application of these algorithms to image edge detection has become an inevitable trend, and some scholars have begun this research and achieved some results, but there is still a broad space for development in this area.

This article is adhering to the idea of such a link between the past, proposed an edge detection algorithm based on the image eigenvector and the improved BP neural network. The eigenvector includes eight differential components and one median component, which can preserve important details while filtering out noise. Because the edge detection of the image is to distinguish the edge points and non-edge points in the pixel points, which belongs to a clustering problem. Therefore, it is reasonable to introduce the BP neural network with excellent screening clustering effect.

2. Algorithm Introduction

The algorithm is divided into two parts, the first is the eigenvector extraction based on grayscale image, the second is improved BP Neural Network. Using the eigenvector of an image as an input sample of the improved BP network, the network is trained to adjust the weight and threshold of each node. Image Edge Detection with trained BP network is the core idea of the algorithm.

2.1 Extract the Tigenvector of the Pixel.

Although it's easy to use the image grayscale for edge extraction, but a lot of important details are usually being missed, and there is no distinction between noise points and edge points. The differential components extract the gray level of a pixel and the grayscale of the surrounding directions as a measure of the gray distribution of the pixel. If the gray value of the pixel A(x,y) in the image is f(x,y), the differential component in each direction is defined as:

$$\begin{cases} V_1 = |f(x+1,y) - f(x,y)| & V_2 = |f(x-1,y) - f(x,y)| \\ V_3 = |f(x,y+1) - f(x,y)| & V_4 = |f(x,y-1) - f(x,y)| \\ V_5 = |f(x+1,y+1) - f(x,y)| & V_6 = |f(x-1,y+1) - f(x,y)| \\ V_7 = |f(x+1,y-1) - f(x,y)| & V_8 = |f(x-1,y-1) - f(x,y)| \end{cases} \tag{1}$$

Median filtering is a nonlinear local spatial filtering technique, commonly used in pre-processing of image edge detection, it has a good noise filtering effect and completely retention of image information, so it is highly respected in the field of edge detection [3]. In this paper, the median component is extracted from the median filter of the idea, the extraction method is as follows:

1) Take the neighborhood of a pixel as the median component extract window;

2) Calculate the median gray

$$f_m(x,y) = \underset{3\times 3}{\text{Med}}\{f(x,y)\} \tag{2}$$

The definition of the Med function is as follows:

For ordered arrays

$$\mathrm{X} = [x_1, x_2, ...x_{n-1}, x_n];\ (x_1 \le x_2 \le ... \le x_n) \tag{3}$$

$$\text{Med}(\mathrm{X}) = \begin{cases} x_{\frac{n+1}{2}} & n \text{ is odd} \\ \frac{1}{2}\left[x_{\frac{n}{2}} + x_{\frac{n}{2}+1} \right] & n \text{ is even} \end{cases} \tag{4}$$

3) Median component

$$V_9 = |f_m(x,y) - f(x,y)| \tag{5}$$

The extracted differential components are taken as the eigenvector of the pixel point together with the median component, then

$$V = [V_1, V_2, ...V_8, V_9] \tag{6}$$

2.2 BP Neural Network and Its Improved Algorithm.

2.2.1 Introduction to BP Neural Network.

BP neural network is the multi-layer feedforward neural network, which is the most widely used in the neural network model. It's suitable for some nonlinear problems which cannot be accurately modeled [4]. And it is very effective for many screening clustering problems just like screening edge points from an image pixels.

As the three-tier structure of the BP network can solve most of the problems, so the general BP network are used three-tier structure model, including the input layer, the middle hidden layer and the output layer. Each layer contains a certain number of neuronal nodes, the weight and threshold of each node can be adjusted according to the feedback error [5]. Through the training of samples, optimize the weight and threshold of each node, so that the BP network output tends to the ideal state.

2.2.2 Improved Algorithm of BP Neural Network.

In this paper, an improved BP network combining adaptive learning rate and momentum factor is proposed to solve the problems of slow convergence and local minimum, the description is as follows:

1) Adaptive learning rate.

The high learning rate of the BP network will cause unstable, and the convergence rate is too slow when the value is small. So the value needs to be adjusted according to the actual error, the formula is as follows:

$$\mu_{(k)} = \begin{cases} 1.05\mu_{(k-1)} & e_{(k)} < e_{(k-1)} \\ 0.75\mu_{(k-1)} & e_{(k)} > 1.05e_{(k-1)} \\ \mu_{(k-1)} & else \end{cases} \tag{7}$$

Where $\mu_{(k)}$ isthe learning rate for the *k*th iteration, $e_{(k)}$ is the value of the *k*th iteration output error.

2) Momentum factor.

Introduce the momentum factor, when the amount of correction of the iteration is the same as the previous time, it indicates that the system is in steady state and the correction amount can be increased. When the correction amount is different from the previous time, it means that there is oscillation and we should reduce the amount of correction. The formula is:

$$\Delta\omega_{(k)} = \alpha\Delta\omega_{(k-1)} + (1-\alpha)\mu_{(k-1)}\frac{\partial e_{(k-1)}}{\partial\omega_{(k-1)}} \tag{8}$$

$$\omega_{(k)} = \omega_{(k-1)} + \Delta\omega_{(k)} \tag{9}$$

Where ω is the weight, μ is learning rate, e is output error.

2.3 Image Edge Detection Method Based on Eigenvector and Improved BP Neural Network.

The schematic diagram of this method is shown in Fig 1 and the flow chart is shown in Fig 2.

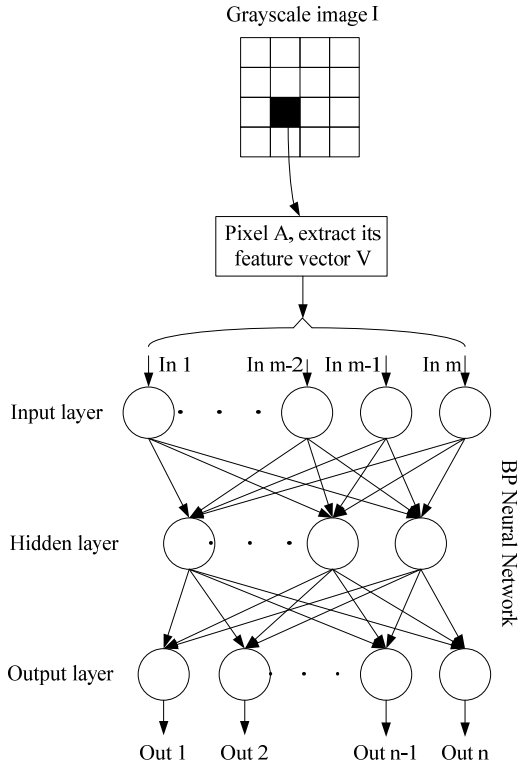


Fig. 1 Schematic

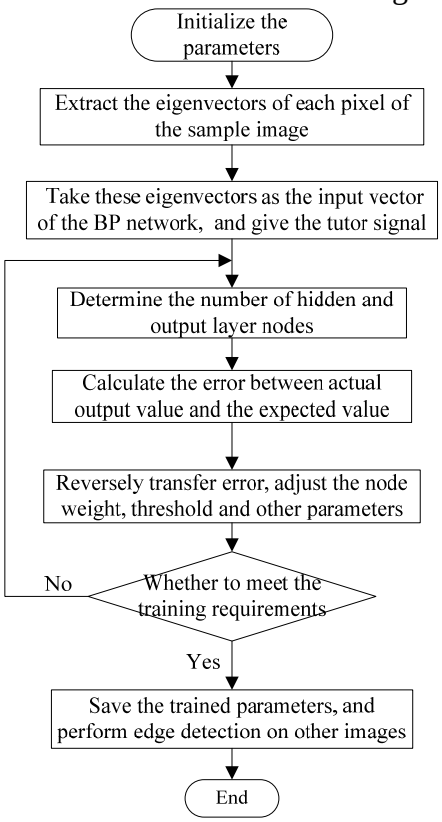


Fig. 2 Flow chart

3. Experimental Steps and Analysis of Results

3.1 Training Neural Network.

1) Determine the number of nodes in each layer of BP neural network.

The number of input nodes of the neural network is determined by the dimension of the input vector [6]. The image eigenvector extracted in this paper is 9-dimension, so it can be determined that the number of input nodes of the network is 9. The output layer requires 1 node, when the detection point is edge the output is 1, otherwise the output is 0.

The hidden layer is very important for the convergence speed and learning effect of BP network. For the determination of the number, the current academic circles have no clear way, generally using the following empirical formula:

$$n_1 = \sqrt{m+n} + \gamma \tag{10}$$

Where m and n are the number of input and output nodes, γ is an integer within 1-10.

2) Select training sample and tutor signal.

The training sample set requires a rich image texture and simple structure, so BP network can get more comprehensive training and converge as soon as possible.

This article has been screened and decided to use the picture shown in Fig 3(a), its structure is simple and texture is rich, more in line with the requirements.

In order to target the training BP network on the noise signal processing effect, add salt and pepper noise with a density of 0.05 to Fig 3(a), as shown in Fig 3(b). It is used as a training sample set.

After selecting the training sample, we also need to clear the expected output (tutor signal) of the BP network, according to the error between tutor signal and actual output to reverse the adjustment parameters [7]. Using the Canny operator to extract the edge of the Fig 3(a), and then appropriate to correct it as a tutor signal, as shown in Fig 3(c).

(a) (b) (c)

Fig. 3 Training sample and tutor signal

3) Normalized input vector.

BP neural network generally uses S-type function as the activation function [8], the range is $(0,1)$, and when the input value is too large, the weight of the adjustment will be very small, the network clustering function is not obvious. The input feature vector of this paper is based on image grayscale extraction, the range is $[0,255]$, so it needs to be normalized, methods as below:

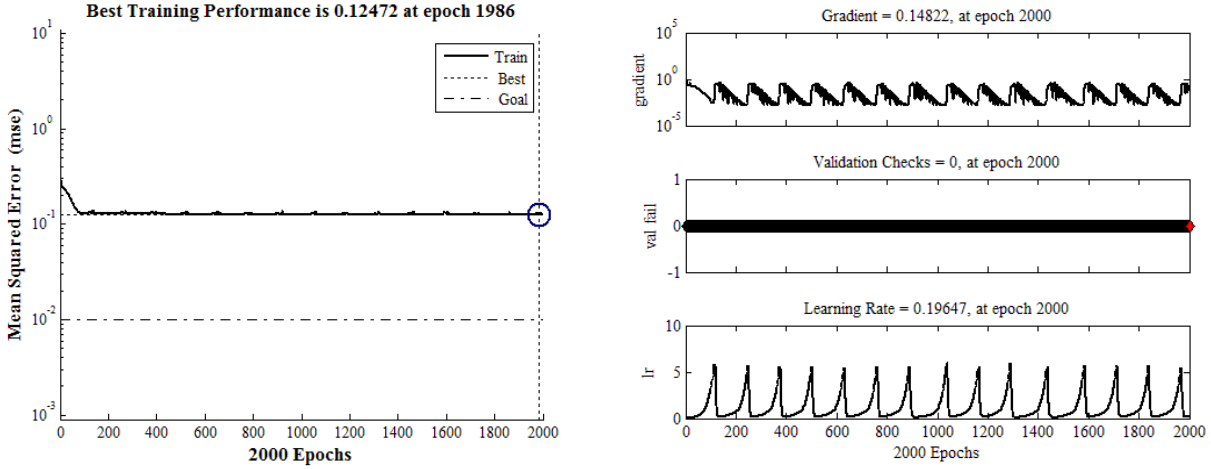
Find maximum value $V_{\max}$ and minimum value $V_{\min}$ in the eigenvector;

$$V_i^{'} = \frac{V_i - V_{\min}}{V_{\max} - V_{\min}} \qquad i = (1,2,...,9) \tag{11}$$

4) Training BP neural network.

Modeling and Simulation with Matlab Neural Network Toolbox, use the newff() function to create a BP network, set the hidden layer and output layer activation functions as 'logsig' and 'purelin', BTF as 'traingdx', learning rate growth ratio $lr\_inc = 1.05$, learning rate drop ratio $lr\_dec = 0.75$, momentum factor $mc = 0.9$, maximum number of iterations times $epochs = 2000$, accuracy $goal = 0.01$.

Select N samples of a region from the sample set, using the gray scale matrix to calculate the $9 \times N$ feature matrix consists of eigenvectors. It is normalized as the input vector of the network. The tutor signal of the same area is expanded into a $1 \times N$ matrix as the desired output. The number of hidden layer nodes is set to 5~10 one by one simulation, when the number of nodes is 7 reached the best results, as shown in Fig 4.



(a) (b)

Fig. 4 Simulation results (Take N = 1600)

Fig 4(a) is the error convergence curve of the network, it can be seen from the figure that the error begins to stabilize after 100 iterations and stable at 0.125 when 2000 iterations are completed. Fig 4(b) is the training curve of the network, including the gradient, validation checks and learning rate parameters, it can be seen from the figure that the gradient and the learning rate are cyclically changed, the range of the gradient is $1e^{-3} \sim 1$, the range of learning rates varies 1~7. The validation checks value of 0 indicates that the network is converging.

Some of the parameters obtained after training are as follows:

Hidden layer to output layer weights:

$w = [0.3551 \ -0.1428 \ 0.2213 \ -0.0355 \ 0.2767 \ -0.2113 \ -0.2008]$

Neural thresholds of Hidden layer:

$theta1 = [3.7424 \ 8.4460 \ -2.1058 \ -4.3589 \ 6.9690 \ -1.1679 \ 5.4535]^T$

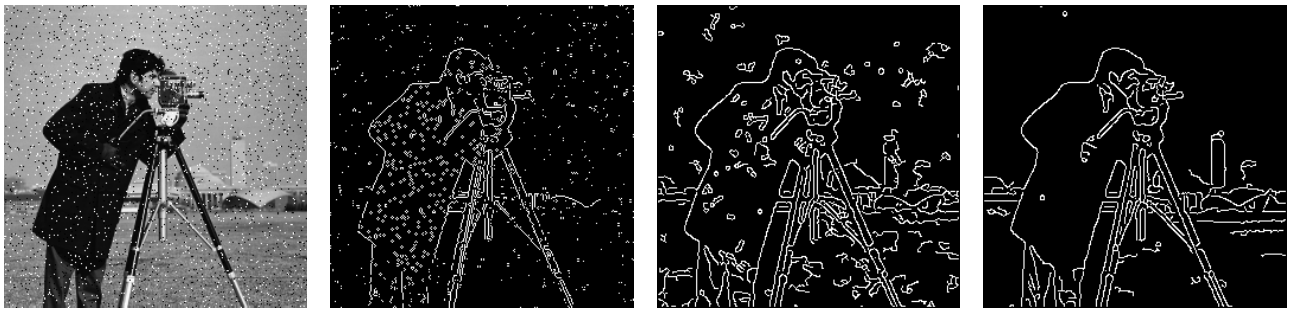
Neural threshold of output layer:

$theta2 = -0.1204$

3.2 Image Edge Detection Using Neural Network.

The trained BP neural network is used to detect the edge of the *Cameraman* and its noise map, and the results are compared with those of the sobel and canny operators. The results are shown in Fig 5.

(a) Cameraman (b) Sobel (c) Canny (d) The algorithm



(e) Noise map (f) Sobel (noise) (g) Canny (noise) (h) the algorithm (noise)

Fig. 5 Edge detection results

Fig (b) (c) (d) is the result of the detection of *Cameraman* original picture. Fig (f) (g) (h) is the detection result of the *Cameraman* plus noise map. In the detection of the original map, we can see that Sobel operator omits many weak edges but canny operator effect is much better, the edge lines are basically detected out. The detection effect of this algorithm is equivalent to the canny operator, but it is not as good as Canny in the edge of the smooth, this is because the tutor signal is based on canny detection and the detection effect is limited.

In the detection of the Noise map, Sobel operator has the worst detection effect, after setting the threshold of 0.2, we still only see the faint edge curve, and the other area is basically the noise floor. The Canny operator is subjected to a double threshold [0.1, 0.3] and smoothing filtering, the detection results have been significantly improved, but there are many false edges in the figure, making the lines very rough. He algorithm has the best detection effect, only a little subtle noise exists, the detection of the edge lines are very clear.

4. Summary

In this paper, an edge detection method based on image feature vector and neural network is proposed. The eigenvector of pixel extraction is based on image grayscale, using the improved BP neural network combined with adaptive learning rate and momentum factor and select the sample image to train the network [9]. Through the simulation experiment, optimize the network structure and the node weight, threshold, then save these parameters. Using the trained network to detect the edge of the *Cameraman* original image and its noise map, compared with the classical Sobel and Canny operator test results, we can conclude that this algorithm has good edge detection performance and good anti-noise performance.

References

[1]. J. Canny. A computational approach to edge detection. IEEE Trans. Vol. 8 (1986) No. 6, p. 679-698.
[2]. A. Bovik, T. Huang, D. JR. Non-parametric tests for edge detection in noise. Pattern Recognition. Vol. 19 (1986) No. 3, p. 209-219.
[3]. Z. He, M. Siyal. Edge detection with BP neural networks. The Int'l Conf. Signal Processing, Beijing, 1998, p. 608-113.
[4]. Meng, Zhang-rong. The requirement analysis on selections among various color models. Journal of Image and Graphics. Vol. 1 (1996) No. 3, p. 238-241.
[5]. Li Weiqing, Wang Chengbiao, Wang Qun, et al. An edge detection method based on optimized BP neural network. International Symposium on Information Science and Engineering. Shanghai, 2008, p. 40-44.
[6]. Guiffaut C, Mahdjoubi K. A parallel FDTD algorithm using the MPI library. IEEE Antennas and Propagation Magazine. Vol. 43 (2001) No. 2, p. 94-103.
[7]. M. Fesharaki, G. Hellestrand. A new edge detection algorithm based on a statistical approach. The Int'l Conf. Speech of Image Processing and Neural Networks. Hong Kong, 1994, p. 864-870.

[8]. Dang Xiangying: Research and Application of High Precision Fast Image Interpolation Algorithm Based on Edge Direction (Master's degree, Jiangnan University, China, 2008). P. 24.
[9]. Wang Jiawen, Cao Yu. MATLAB 6.5 Graphic Image Processing. National Defense Industry Press, 2004, p. 145-147.