Neural Edge Enhancer for Supervised Edge Enhancement from Noisy Images

Kenji Suzuki, *Member*, *IEEE*, Isao Horiba, and Noboru Sugie, *Member*, *IEEE*

Abstract—We propose a new edge enhancer based on a modified multilayer neural network, which is called a neural edge enhancer (NEE), for enhancing the desired edges clearly from noisy images. The NEE is a supervised edge enhancer: Through training with a set of input noisy images and teaching edges, the NEE acquires the function of a desired edge enhancer. The input images are synthesized from noiseless images by addition of noise. The teaching edges are made from the noiseless images by performing the desired edge enhancer. To investigate the performance, we carried out experiments to enhance edges from noisy artificial and natural images. By comparison with conventional edge enhancers, the following was demonstrated: The NEE was robust against noise, was able to enhance continuous edges from noisy images, and was superior to the conventional edge enhancers in similarity to the desired edges. To gain insight into the nonlinear kernel of the NEE, we performed analyses on the trained NEE. The results suggested that the trained NEE acquired directional gradient operators with smoothing. Furthermore, we propose a method for edge localization for the NEE. We compared the NEE, together with the proposed edge localization method, with a leading edge detector. The NEE was proven to be useful for enhancing edges from noisy images.

Index Terms—Supervised edge enhancer, noisy image, robustness, neural network, edge detection, contour extraction.

✦

1 INTRODUCTION

EDGE enhancement is one of the most fundamental operations in image analysis, and there are probably more algorithms for enhancing and detecting edges in the literature than for any other single subject [1], [2], [3], [4], [5]. Edges form the outline of an object, and an edge is the boundary between an object and the background. If the edges in an image can be identified accurately, all of the objects can be located, and basic properties such as area, perimeter, and shape can be measured. Edge detection, in general, consists of two parts: edge enhancement, which is a process for calculating the edge magnitude at each pixel; and edge localization, which is a process for determining the exact edge location. Once an edge is enhanced properly, the location of the edge can be identified accurately. Thus, the performance of edge detection depends on that of edge enhancement.

Because the objects in an image obtained from real-world scenes are generally buried in noise, 1) robust enhancement against noise is required for edge enhancement algorithms. In addition, there exist various scales, orientations, and magnitudes of edges in an image. The desired edges for a certain application may differ from those for another one. If we employ an active contour model such as a SNAKES [6]

to extract contours, it functions relatively successfully by use of the edges enhanced as thick rather than thin curves. Therefore, 2) enhancing the desired edges appropriately for each application is required for edge-enhancement algorithms.

Many studies on requirement 1) have been carried out and, thus, many edge enhancers/detectors have been proposed [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]. However, in an attempt to enhance edges from an image with a large amount of noise, inadequate results may be obtained: Not only edges are enhanced, but also noise; edges are enhanced discontinuously. Furthermore, most studies have focused attention on enhancing/detecting edges from noisy images, and less attention has been given to requirement 2). Therefore, how to enhance the desired edges clearly from noisy images has remained a serious issue.

Recently, in the field of signal processing, nonlinear filters based on a multilayer neural network (NN), called neural filters (NFs), have been studied [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35]. In the NF, a multilayer NN is employed as a convolution kernel. The NFs can acquire the function of various linear and nonlinear filters through training. Suzuki et al. developed NFs for reduction of additive Gaussian/quantum noise in natural/medical images and reported that the performance of the NFs was superior to that of leading nonlinear filters [28], [29], [30], [31], [32], [33], [34], [35].

We have now extended the NFs to accommodate an edge enhancement task, and we developed a new edge enhancer based on a modified multilayer NN, which is called a neural edge enhancer (NEE), to enhance the desired edges clearly from noisy images. In order to handle continuous values such as the edge magnitude, we modified the structure of a multilayer NN, and we developed a training

- K. Suzuki is with the Kurt Rossmann Laboratories for Radiologic Image Research, the Department of Radiology, The University of Chicago, 5841 South Maryland Avenue, Chicago, IL 60637.
  E-mail: suzuki@uchicago.edu.
- I. Horiba is with the Faculty of Information Science and Technology, Aichi Prefectural University, 1522-3 Ibaragabasama, Kumabari, Nagakute, Aichi, 480-1198 Japan. E-mail: horiba@ist.aichi-pu.ac.jp.
- N. Sugie is with the Faculty of Science and Technology, Meijo University, 1-501 Shiogamaguchi, Tempaku, Nagoya, 468-8502 Japan.
  E-mail: sugie@ccmfs.meijo-u.ac.jp.

method for the modified structure. In the NEE, a modified multilayer NN is employed as a nonlinear convolution kernel for edge enhancement. Through training with a set of input noisy images and teaching edges, the NEE acquires the function of a desired edge enhancer. Thus, the NEE is a supervised edge enhancer.

Several applications of NNs to edge detection have been studied so far. They can be classified into four broad categories:

1. edge detectors based on cellular NNs [36], [37], [38], [39], [40], [41], [42], [43];
2. edge detectors based on self-organizing maps [44], [45], [46];
3. edge detectors based on Hopfield networks [47], [48], [49], [50];
4. edge detectors based on multilayer NNs [51], [52], [53], [54].

The above edge detectors, except class 4, are unsupervised ones and, thus, they do not have the function of enhancing the desired edges. Therefore, they do not necessarily satisfy requirement 2). Furthermore, because they were not intended to detect edges robustly against noise, they may not satisfy requirement 1). As for class 4, the NN is used as a classifier in these edge detectors. They cannot handle continuous values such as edge magnitude, i.e., these NNs directly classify whether a certain pixel belongs to the class, an edge, or to the class, a background. In many applications, a map of edge magnitudes is more useful than only a map of edge locations. Therefore, an edge enhancer that outputs the edge magnitude would be more useful for many applications than an edge detector that identifies edge locations.

2 NEURAL EDGE ENHANCER

2.1 Edge-Enhancement Problem

In general, an image obtained from real-world scenes is corrupted by noise. Noise in an image can be classified into two major types. One is signal-dependent noise, such as quantum noise. The other is signal-independent noise such as additive Gaussian noise. Quantum noise is the dominant form in an image obtained in darkness, an image obtained from an infrared camera, an X-ray image, a radiograph, etc. Quantum noise originates from a signal-dependent, Poisson-distributed noise source. The variance of the Poisson noise increases linearly with signal amplitude. The Poisson-distributed noise can be approximated by a Gaussian when the number of quanta is relatively large. Therefore, we can use signal-dependent Gaussian noise as a model of quantum noise. Assuming that noise is quantum noise, a noisy image can be represented by

$$g(x, y) = f(x, y) + N(\sigma), \tag{1}$$

where $f(x, y)$ is a noiseless image, $N(\sigma)$ is white Gaussian noise when its standard deviation σ is $K_N \sqrt{f(x, y)}$, and K_N is a parameter determining the amount of noise. The desired edge magnitude can be calculated from the noiseless image as follows:

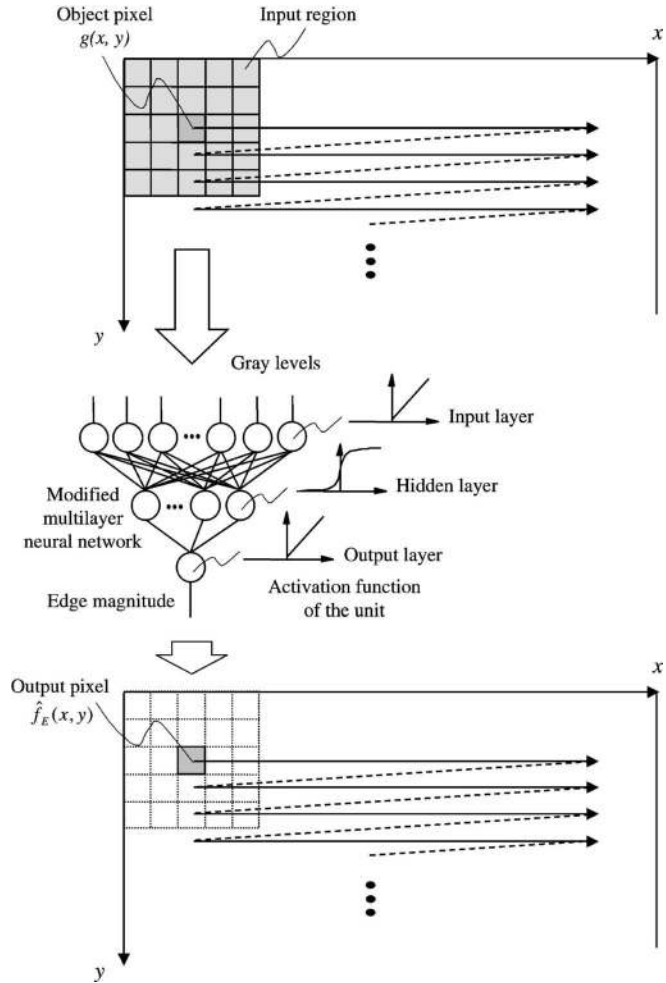$$f_E(x, y) = \varphi\{f(x, y)\}, \tag{2}$$

Fig. 1. Architecture of the neural edge enhancer (NEE). In the NEE, the modified multilayer NN is employed as a nonlinear convolution kernel for edge enhancement. The entire image is obtained by scanning with the modified multilayer NN.

where φ is an operator calculating the desired edge magnitude. Because edge enhancement is represented as a technique for finding an operation that transforms the noisy image into a map of the desired edge magnitudes, edge enhancement can be formulated as

$$\hat{f}_E(x, y) = \vartheta\{g(x, y)\}, \tag{3}$$

where $\hat{f}_E(x, y)$ is an estimate for the desired edge magnitude and $\vartheta(\cdot)$ is an operator realizing edge enhancement.

2.2 Architecture

The architecture of the NEE is shown in Fig. 1. The NEE consists of a modified multilayer NN, which can directly handle input gray levels and output edge magnitudes. In the NEE, the modified multilayer NN is employed as a nonlinear convolution kernel for edge enhancement. In the modified multilayer NN, the activation functions of the units in the input, hidden, and output layers are an identity, a sigmoid, and an identity function, respectively. We employ an identity function instead of the ordinarily used sigmoid one as the activation function of the unit in the output layer because the characteristics of an NN were improved significantly with an identity function when

applied to continuous mapping issues such as image processing [55], [56]. It should be noted that a conventional NN which uses a sigmoid function as the activation function of the unit in the output layer is useful for applications where the output is in the form of binary values such as a classification task.

The pixel values in an input region R_S are normalized and then input to the NEE. The inputs to the NEE are a normalized object pixel value and spatially adjacent normalized pixel values. Although the most common use of an NN is as a classifier that determines whether a certain pixel belongs to the class, such as an edge or a background, the output of the NEE is not a class, but the estimate for the edge magnitude, represented by

$$\hat{f}_E(x,y) = G_M \cdot NN[\{g(x-i, y-j)/G_M | i,j \in R_S\}], \quad (4)$$

where G_M is a normalization factor and $NN(\cdot)$ is the output of the modified multilayer NN. The entire image is obtained by scanning with the modified multilayer NN. The modified multilayer NN, therefore, functions like a convolution kernel. The multilayer NN as a nonlinear kernel is designed by training such that the input images are converted to a map of the desired edge magnitudes. The universal approximation property of a multilayer NN [57], [58] guarantees the capability of the NEE, i.e., the multilayer NN can realize any continuous mapping approximately. This means that the use of the multilayer NN as a convolution kernel can realize, through the convolution operation, various image-processing techniques, including high-pass, low-pass, and band-pass filtering, noise reduction, and edge enhancement. For example, the modified multilayer NN-based kernel can act as an averaging operation, gradient operation, Laplacian operation, etc. Thus, the NEE has a high potential for solving some of existing problems in edge enhancement/detection as well as image processing.

2.3 Training

The NEE is trained with a set of noisy input images and the teaching images, including the desired edge magnitudes, by adjusting the weights between layers. The error to be minimized by training is defined by

$$E = \frac{1}{2P} \sum_p (f_E^p/G_M - \hat{f}_E^p/G_M)^2, \quad (5)$$

where p is a training pixel number, f_E^p is the pth training pixel in the teaching images, $\hat{f}_E^p$ is the pth training pixel in the output images, and P is the number of training pixels. The NEE is trained by the modified back-propagation algorithm in [56], [55], which was derived for the above structure in the same way as in the derivation of the back-propagation algorithm [59], [60]. The correction of a weight between the mth unit in the hidden layer and the unit in the output layer is represented by

$$\Delta W_m^O = -\eta \cdot \delta \cdot O_m^H = -\eta(f_E/G_M - \hat{f}_E/G_M)O_m^H, \quad (6)$$

where η is a leaning rate, O_m^H is the output of the mth unit in the hidden layer, and δ is the delta of the delta rule [59], [60]. By use of delta, the corrections of any weights can be derived in the same way as in the derivation of the

back-propagation algorithm. The NEE is trained by the above modified back-propagation algorithm. After training, the NEE will output the desired edge magnitudes. An edge localization method, such as the one that we propose in this paper (described later) or an existing one, can be performed accordingly on the map of the edge magnitudes obtained by the trained NEE.

2.4 Comparison of the Proposed NN with a Conventional One

2.4.1 Property

In order to clarify the basic property of the proposed NN, we consider the relationship between the proposed NN and a conventional one theoretically. As for the structure, we can understand easily that it is hard for the conventional NN to output values near one and zero, whereas the proposed NN can output all values equally.

In the proposed training, the correction of a weight between the unit in the hidden layer and the unit in the output layer is represented by

$$\begin{aligned}
\Delta W_m^O &= -\eta \frac{\partial E}{\partial O^O} \frac{\partial O^O}{\partial X} \frac{\partial X}{\partial W_m^O} \\
&= -\eta \frac{\partial E}{\partial O^O} f_I'(X) O_m^H \quad (7) \\
&= -\eta \frac{\partial E}{\partial O^O} O_m^H,
\end{aligned}$$

where O^O is the output of the unit in the output layer, X is an input value to the activation function, and f_I' is the derivative of an identity function. On the other hand, the correction of a weight in the conventional training is represented by

$$\begin{aligned}
\Delta W_m^O &= -\eta \frac{\partial E}{\partial O^O} f_S'(X) O_m^H \\
&= -\eta \frac{\partial E}{\partial O^O} O^O(1 - O^O) O_m^H,
\end{aligned} \quad (8)$$

where f_S' is the derivative of a sigmoid function. Comparing the two equations, we can find that the difference is just the derivatives of activation functions. Therefore, we can rewrite the right side of the above equation as the following equation, using η_S:

$$-\eta \frac{\partial E}{\partial O^O} O^O(1 - O^O) O_m^H = -\eta_S \frac{\partial E}{\partial O^O} O_m^H. \quad (9)$$

When the training proceeds, the output of the NN, O^O, should approach the teaching edge magnitude, f_E/G_M. Therefore, the learning rate of the conventional training can be approximated by

$$\eta_S = \eta \cdot O^O(1 - O^O) \approx \eta \cdot f_E/G_M(1 - f_E/G_M). \quad (10)$$

This equation shows that the learning rate of the conventional training is modulated by the derivative of a sigmoid function, which is 0.5 when the teaching edge magnitude is 0.5 and is zero when the teaching edge magnitude is zero or one. In other words, the learning rate of the proposed training corresponds to that of the conventional training before the modulation. Therefore, in the conventional training, the edge magnitudes of zero and one are never
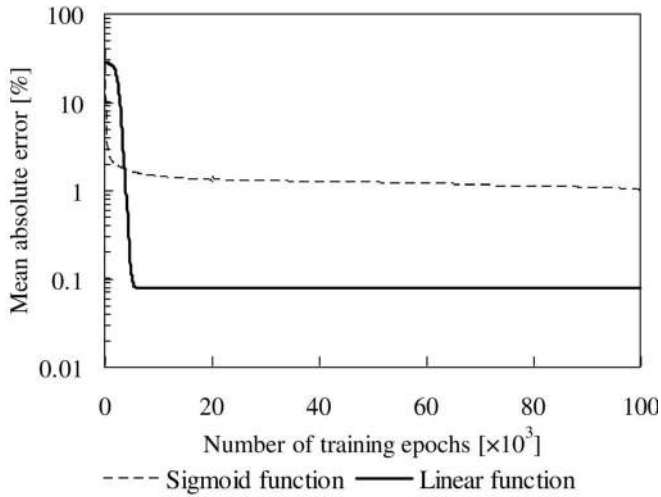
(a)
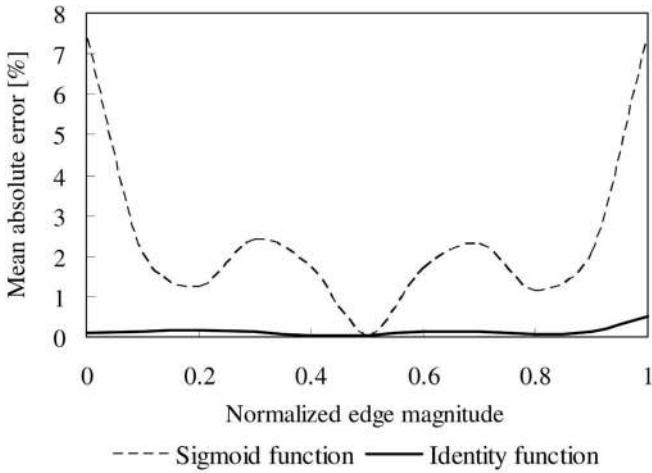
(b)

Fig. 2. Comparison of the property of the proposed NN with that of the conventional NN. (a) Learning curve. (b) Mean absolute errors as a function of the edge magnitudes of the teaching edges.

trained, and the training for the edge magnitude near zero and one converges more slowly. This would affect the convergence characteristic and the output characteristic.

An alternative way to cancel the derivative of a sigmoid function would be to use the cross-entropy energy function instead of the sum-of-square error function [61].

2.4.2 Experiment

In order to illustrate the basic property of the proposed NN, we carried out an experiment to enhance the edges of artificial patterns. A pulse pattern whose height increases in proportion to the horizontal position was generated by the following equation:

$$f(x,y) = \begin{cases} a[x/2k] & if\ [x/k]\ is\ odd \\ 0 & if\ [x/k]\ is\ even, \end{cases} \qquad (11)$$

where $[\ ]$ denotes the Gauss notation, k is the width of the pulse, and a is a gain parameter for determining the height of the pulse. The pulses were generated by use of the following parameters: $a = G_M/100$; $k = X_{max}/(2 \times 100 + 1)$,
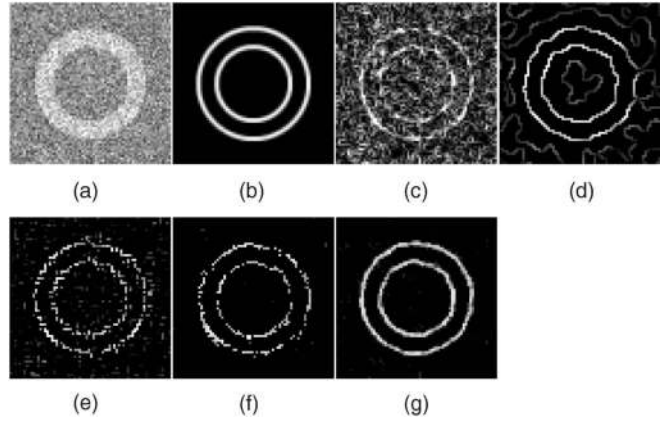

Fig. 3. Comparison of the edges enhanced by the NEE with those enhanced by the conventional edge detectors (SNR_T: 3 dB; edge contrast: medium). (a) Input image. (b) Teaching edges. (c) Sobel filter. (d) Optimized Marr-Hildreth operator. (e) Optimized Canny edge detector. (f) Optimized Hueckel operator. (g) Neural edge enhancer.

where X_{max} is the number of pixels in the horizontal axis, i.e., the pulse image included 100 pulses, and the maximum height of the pulse was G_M. The teaching edges were obtained from the pulse image by application of the Sobel filter. Five training samples were selected at $x = 20k$, $60k$, $100k$, $140k$, and $180k$. The input regions of the proposed and the conventional NNs consisted of three-by-three pixels, which correspond to the kernel of the Sobel filter. The numbers of the units in the input, hidden, and output layers were nine, five, and one, respectively. The NNs were trained on 100,000 epochs.

The learning curves of NNs are shown in Fig. 2a. The training of the proposed NN converged with a much smaller error than did the conventional NN. The mean absolute errors as a function of the edge magnitudes of the teaching edges are shown in Fig. 2b. These results lead to the conclusion that the proposed NN is suitable for applications involving continuous values such as edge enhancement.

3 EXPERIMENTS WITH NOISY ARTIFICIAL IMAGES

3.1 Synthesizing Artificial Images

In order to examine the performance on edge enhancement against noise, we prepared artificial images for testing as described in [62] for experiments. First, images including two concentric circles were generated as follows:

$$c_b(x_b, y_b) = \begin{cases} S_0 - S/2 & (0 \le r < r_1) \\ S_0 + S/2 & (r_1 \le r < r_2) \\ S_0 - S/2 & (r_2 \le r), \end{cases} \qquad (12)$$

where

$$r = \sqrt{(x_b - x_C)^2 + (y_b - y_C)^2}. \qquad (13)$$

x_b and y_b are the indices of spatial coordinates, x_C and y_C are the indices of coordinates of the central point of an image, r_1 and r_2 are the radii of the concentric circles, S_0 is a base gray level, and S is an edge contrast. The parameters were set according to [62] as $r_1 = 64$, $r_2 = 96$, $S_0 = 60$, and
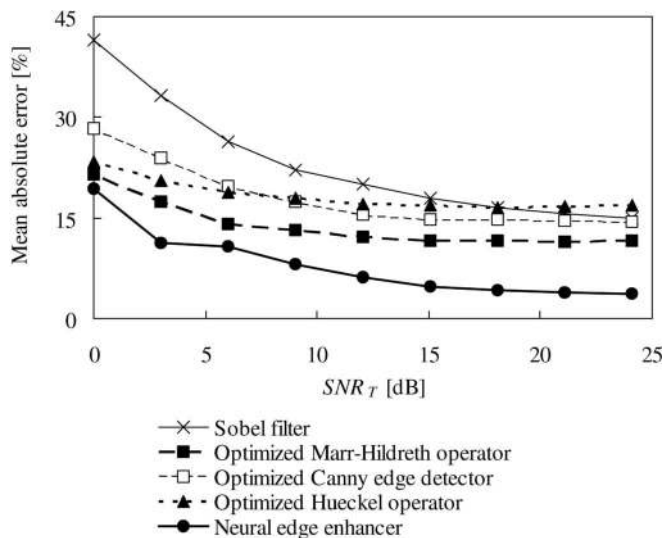
Fig. 4. Comparison of the performance of the NEE in robustness against noise with that of conventional edge detectors (edge contrast: medium).

Fig. 6. Input image and its ideal edges, which were not used for training (nontraining image: teapot). (a) Noisy input image. (b) Ideal edges.
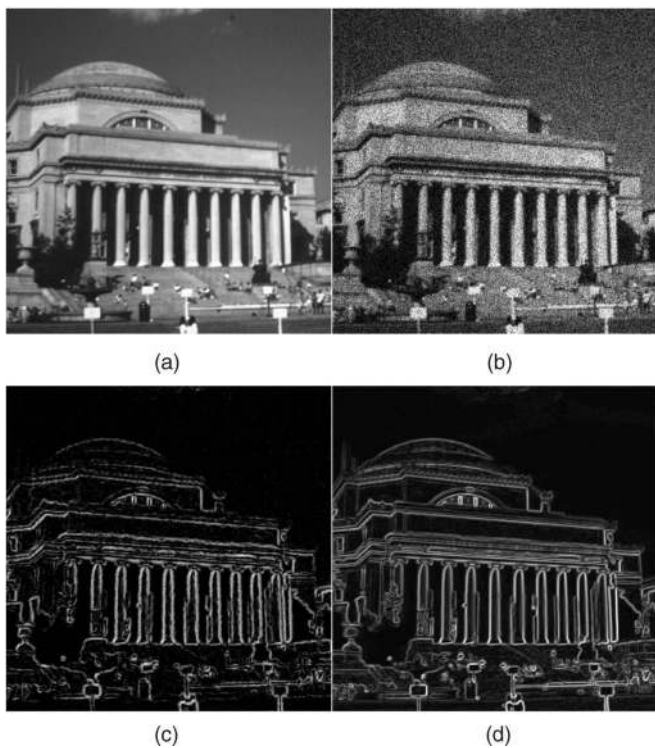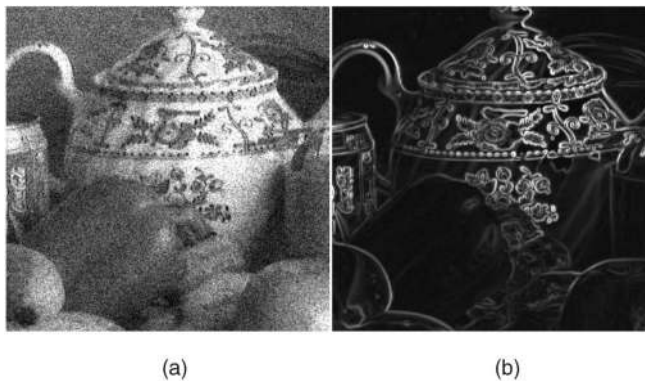

Fig. 5. Natural images used in our experiment (Columbia University). (a) Original image. (b) Noisy input image. (c) Edges enhanced by the NEE trained with the artificial images. (d) Ideal/teaching edges.

the image size was 512×512 pixels. The edges with various orientations can be evaluated by use of such concentric circles. Next, down-sampling based on averaging, the reduction rate of which is $1/4$, was performed on the above image as follows:

$$c(x,y) = \sum_{i,j \in R_{44}} c_b(4x - i, 4y - j)/16, \qquad (14)$$

where R_{44} is a region consisting of four by four pixels. Then, an image consisting of 128×128 pixels was obtained. A noisy artificial image was synthesized from $c(x,y)$ by adding white Gaussian noise as follows:
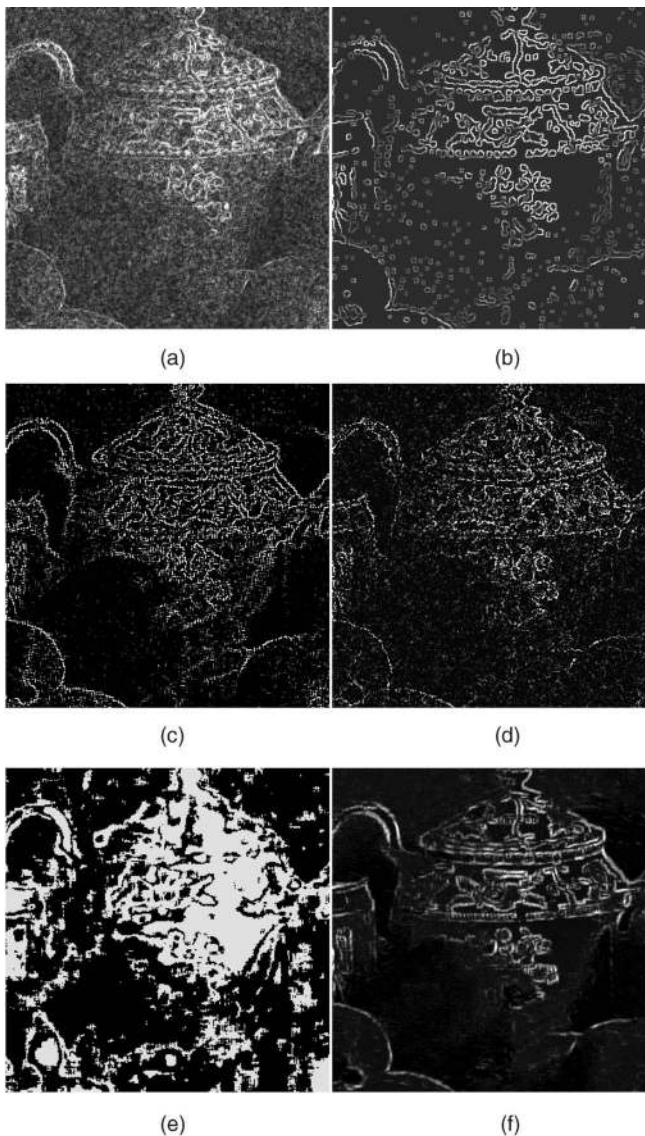

Fig. 7. Comparison of the edges enhanced by the NEE with those enhanced by the conventional edge detectors (nontraining image: teapot). (a) Sobel filter. (b) Optimized Marr-Hildreth operator. (c) Optimized Canny edge detector. (d) Optimized Hueckel operator. (e) Conventional edge detector based on a multilayer NN. (f) Neural edge enhancer.
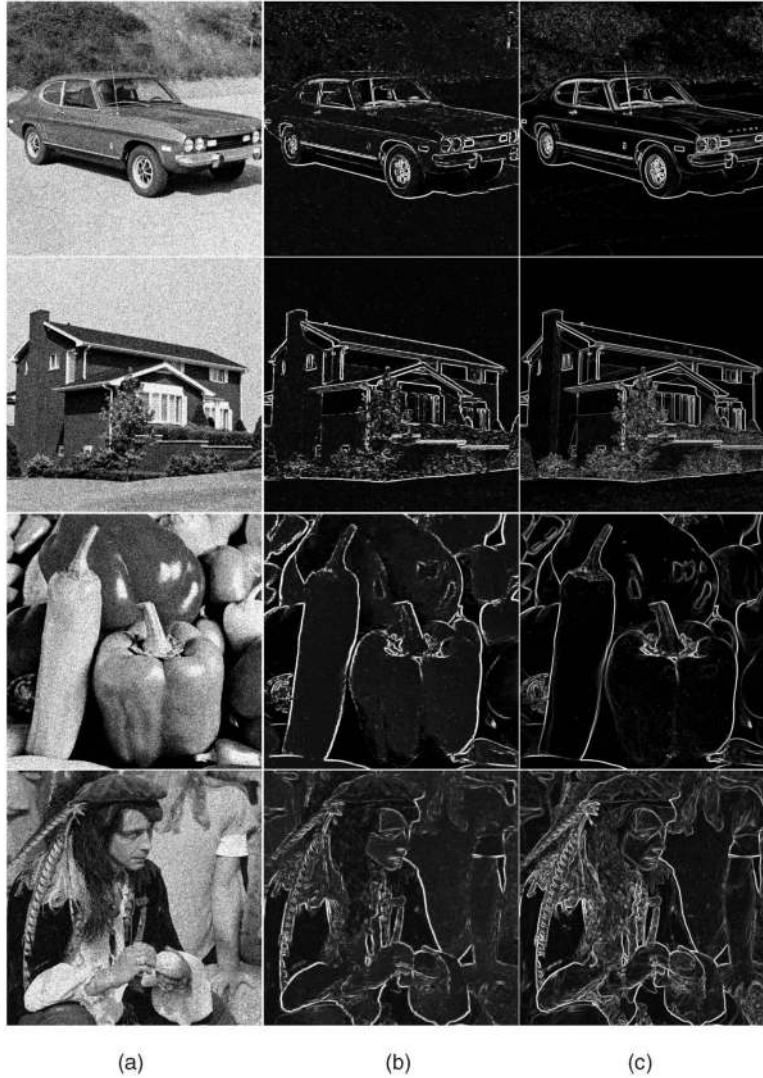
Fig. 8. Edges enhanced by the NEE for nontraining images. (a) Noisy input image. (b) Neural edge enhancer. (c) Ideal edges.

$$c_N(x, y) = c(x, y) + N(\sigma), \qquad (15)$$

where $N(\sigma)$ is white Gaussian noise with standard deviation σ. The signal-to-noise ratio of the artificial image is defined by

$$SNR_T = 10 \log_{10} \left(\frac{S}{\sigma} \right)^2. \qquad (16)$$

Several artificial images were synthesized by varying the SNR_T.

3.2 Training the NEE

The noisy artificial image $c_N(x, y)$ was used as the input image to the NEE. The teaching edges were obtained from $c(x, y)$ by application of a desired edge enhancer, $\varphi(\cdot)$, as follows:

$$f_E(x, y) = \varphi\{c(x, y)\}. \qquad (17)$$

The Sobel filter was used as $\varphi(\cdot)$ in this experiment because it can enhance edges clearly from a noiseless image. The input images used for training were six images: Three images were made on the basis of (12) by varying S to be 15, 25, and 35 (here referred to as the edge contrast: low, medium, and high, respectively), and then two artificial

images were obtained from each of the three images by varying the SNR_T to be 3 dB and 18 dB. Examples of the input images and the teaching edges are shown in Figs. 3a and 3b. The region covering the two circles was used as the training region. Three-layered NEE (input region: 7×7 pixels; number of units in the hidden layer: 20; here referred to as 49-20-1) was trained on 200,000 epochs, and the training converged with an error E of 8.1 percent.

3.3 Conventional Edge Detectors

Conventional edge detectors can be classified into three broad classes:

1. gradient-based edge detectors [7], [11], [12];
2. algorithms dealing with the edge detection as the optimal filtering problem [13], [14], [15], [16], [18], [19], [20], [22], [23];
3. edge-model-based edge detectors [8], [9], [10].

A well-known representative in class 1 is the Sobel filter [7], those in class 2 are the Marr-Hildreth operator [14] and the Canny edge detector [16], and that in class 3 is the Hueckel operator [8], [9]. The Sobel filter is widely used because of a low computational cost. The Marr-Hildreth operator can

TABLE 1
Quantitative Evaluation of Performance by Use of the Error between the Ideal Edge and
the Edge Enhanced by Each Edge Enhancer/Detector (Training Image: Columbia)

Image	SNR_I [dB]	Mean absolute error [%]					
		SF	OMHO	OCED	OHO	EDMNN	NEE
Columbia	3.63	19.0	20.5	21.7	21.0	44.0	11.7
Teapot	5.53	18.1	19.1	20.4	20.1	38.1	14.9
Car	3.86	15.1	15.8	17.1	16.2	50.6	10.9
House	4.90	19.6	16.6	18.5	18.0	60.8	14.5
Peppers	7.19	13.6	15.2	15.9	15.3	50.9	10.7
Man	6.44	14.1	20.4	20.9	20.6	44.5	12.7
Airplane	4.36	19.8	17.4	19.1	18.8	88.1	14.5
Lena	4.49	14.5	15.0	15.4	15.2	52.7	12.1
Crowd	8.07	15.8	21.5	23.1	22.1	37.3	12.2
Lake	8.74	17.7	22.4	24.7	23.5	69.1	14.5
Bridge	7.50	22.4	32.1	34.7	33.5	51.2	21.3
Woman1	4.37	12.7	19.1	18.6	18.9	54.0	11.9
Couple	5.45	16.8	24.8	25.8	24.9	42.6	15.4
Girl	4.85	8.5	13.5	13.1	13.2	28.1	8.0
Average	5.83	16.1	19.5	20.6	20.0	51.4	13.3

SF: Sobel filter; OMHO: Optimized Marr-Hildreth operator;

OCED: Optimized Canny edge detector; OHO: Optimized Hueckel operator;

EDMNN: Conventional edge detector based on a multilayer NN in [51].

detect multiscale edges. The Canny edge detector is well-known as a good one. The Hueckel operator is well-known for robustness against noise.

3.4 Results of Edge Enhancement

We used the edge-enhancement parts of the conventional edge detectors for comparisons of the performance. For comparison of the NEE with the conventional edge detectors fairly, the parameters of the conventional edge detectors were optimized. Furthermore, because the magnitude of the enhanced edge is, generally, different from that of the teaching edge, a gray-scale transformation based on a linear function was performed. The optimization of the parameters and that of the gray-scale transformation were performed with the training images under the minimum-mean-square error criterion, defined as (5). Thus, each conventional edge detector had the highest performance against the training images.

The edges enhanced by each edge detector for the input image having an SNR_T of 3 dB are shown in Fig. 3. The edges enhanced by the NEE are continuous, and there is less false-positive detection (edges detected where no edge exists) than with others. In order to evaluate the performance quantitatively, we calculated the mean absolute error (MAE) between the teaching edges and the enhanced edges (normalization was performed with G_M). The results are shown in Fig. 4. The MAEs of the NEE are the smallest,

even in the case of nontraining images. However, the NEE was not very effective for the image having an SNR_T of 0 dB, which is less than the 3 dB used in training.

In order to investigate the versatility of the trained NEE, we applied the trained NEE to a natural image. The input image was synthesized from an original image (Columbia University, here referred to as Columbia; size: 480×480 pixels; maximum level of the gray scale: 256) by adding quantum noise on the basis of (1), where K_N was 1.2 percent of the maximum level of the gray scale. The ideal edges were synthesized from the original image by application of the Sobel filter. These images are shown in Fig. 5. The result of applying the NEE is shown in Fig. 5c. The result is similar to the ideal edges shown in Fig. 5d, which was synthesized from the original image by performing of the Sobel filter. This result indicated that the NEE trained with the artificial images was versatile. The NEE could learn the essential edge enhancement from the artificial images.

4 EXPERIMENTS WITH NOISY NATURAL IMAGES

4.1 Training the NEE

In order to carry out experiments to enhance the edges from noisy natural images, we used the images shown in Figs. 5b and 5d for training. To acquire the features in the entire image efficiently, the training data set was made by sampling of 10,000 points at random from the training

images. Three-layered NEE was adopted because it has been proven theoretically that any continuous mapping can be approximately realized by a three-layered NN [57], [58]. The training was performed on 1,000,000 epochs and converged with an error E of 1.76 percent. The execution time for the training was 340 hours on a workstation (UltraSPARC-II 300 MHz, Sun Microsystems).

4.2 Comparison with Conventional Edge Detectors

The conventional edge detectors were optimized in the same way as in the previous section. In addition, we compared the NEE with the conventional edge detector based on a multilayer NN [51]. In this edge detector, the histogram of gray levels in the input region is input to the multilayer NN, i.e., the inputs to the multilayer NN are frequencies of gray levels. The multilayer NN is employed as a classifier, i.e., the outputs are the classes, an edge or a background. In addition, the activation function of the unit in the output layer is a sigmoid function. Because this NN cannot handle continuous values such as edge magnitude, the teaching edges were transformed into a binary image by use of thresholding and then the binary image was used as the teaching image for this edge detector. The conventional edge detector based on the multilayer NN was trained under the same conditions as the NEE.

In order to evaluate their generalization ability, we applied the edge enhancers/detectors to the nontraining image shown in Fig. 6a. Fig. 6b shows the ideal edges made from an original noiseless image by application of the Sobel filter. The results are shown in Fig. 7. In the edges enhanced by the Sobel filter, much noise remains. In the edges enhanced by the Marr-Hildreth operator, the noise is enhanced in the form of small circles. The enhanced edges are rounded and different from the desired ones, although they are continuous. In the edges enhanced by the Canny edge detector and the Hueckel operator, much fine noise remains, and the enhanced edges are discontinuous. In the result of the conventional edge detector based on a multilayer NN, the edge is detected as a wide band, and there are many false positives. In contrast, in the edges enhanced by the NEE, there is less noise. The enhanced edges are continuous and clear and similar to the ideal ones. The edges enhanced by the NEE in the case of other nontraining images are shown in Fig. 8. These results indicate that the NEE can enhance the edges similar to the ideal ones for nontraining images.

The CPU execution time of each edge detector on a workstation was measured. The execution time for a 512×512-pixel image in seconds was as follows: the Sobel filter: 0.2, the Marr-Hildreth operator: 2.3, the Canny edge detector: 3.0, the Hueckel operator: 42.4, and the NEE: 6.4.

4.3 Quantitative Evaluation

The signal-to-noise ratio of an input image [63], [64] can be defined as

$$SNR_I = 10 \log_{10} \frac{\sum_{x,y} \left\{ f(x,y) - \overline{f(x,y)} \right\}^2}{\sum_{x,y} \left\{ N(x,y) - \overline{N(x,y)} \right\}^2}, \quad (18)$$
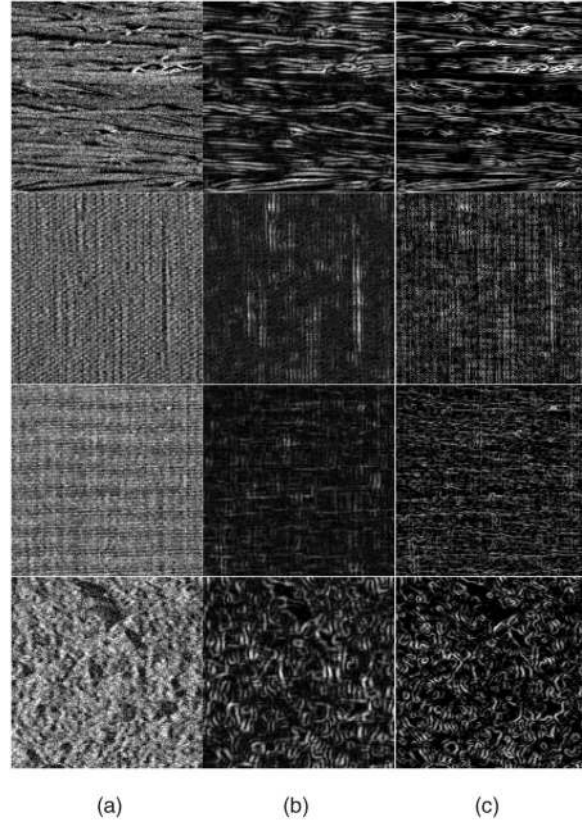
where

$$N(x,y) = f(x,y) - g(x,y), \quad (19)$$

Fig. 9. Results of applying the NEE trained with a natural image to texture images. (a) Input texture images with quantum noise (from top to bottom: straw, linen, cotton, and brown bread). (b) Edges enhanced by the neural edge enhancer. (c) Ideal edges.

and $\overline{f(x,y)}$ and $\overline{N(x,y)}$ denote means for $f(x,y)$ and $N(x,y)$, respectively. The MAE between the ideal edge and the enhanced edge was adopted as a metric for evaluation of the performance. The results of the evaluation are shown in Table 1. Most images are from the USC (University of Southern California) image database. The average values in the table were calculated from nontraining images. The MAEs of the NEE were the smallest of all. These results lead to the conclusion that the performance of the NEE is higher than that of the conventional edge detectors in terms of the similarity to the ideal edges.

4.4 Analysis of the Trained NEE

In order to gain insight into the properties of the trained NEE, we applied it to texture images. The texture images were obtained from the Columbia-Utrecht texture database. Quantum noise was added to the original texture images. The SNR_Is of all input texture images were 4.0 dB. The results is shown in Fig. 9. The NEE was effective for relatively coarse textures such as straw and brown bread, whereas it was not effective for fine textures such as linen and cotton. It should be noted that the results for the original texture images without noise were similar to those for the images with noise. This indicates that the trained NEE smoothes fine textures together with noise.

In order to gain insight into the structure of the trained NEE, we performed an analysis of the structure of the
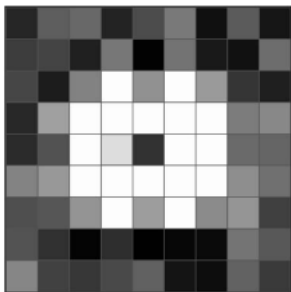
Fig. 10. Result of optimizing the units in the input layer of the NEE, which correspond to the receptive field. Each small square indicates an input unit of the NEE. A white square means a remaining unit after removal. A gray or black square means a removed unit. The order of the removal is expressed as a tone of gray. The darker square indicates a unit which has been removed earlier, i.e., an unnecessary unit.

trained NEE. A method for designing the optimal structure of an NN in [65], [66], [67] was applied to the trained NEE. The redundant units in the input layer and the hidden layer were removed based on the effect of removing each unit on the training error, and then the NEE was retrained to recover the potential loss due to this removal. The removal and the retraining were performed alternately, resulting in a reduced structure where redundant units were removed. As a result, the optimal structure, the smallest structure with the error that had been converged in the original training, was obtained. In other words, the minimum structure having the same performance as the original one was obtained. The number of units in the input layer and that in the hidden layer were determined to be 17 and 17, respectively. Use of this reduced structure would result in efficient training. The execution time for the training was 63 hours on a workstation. Because one unit in the hidden
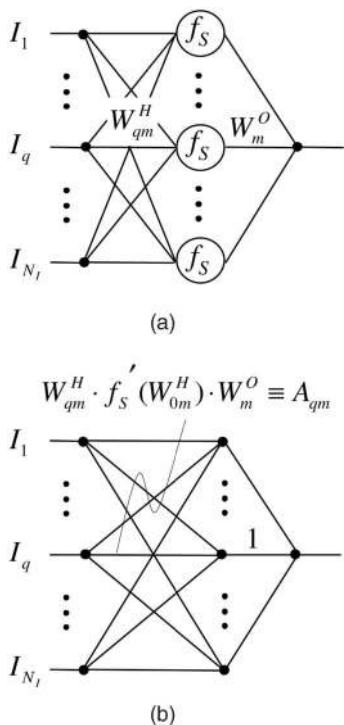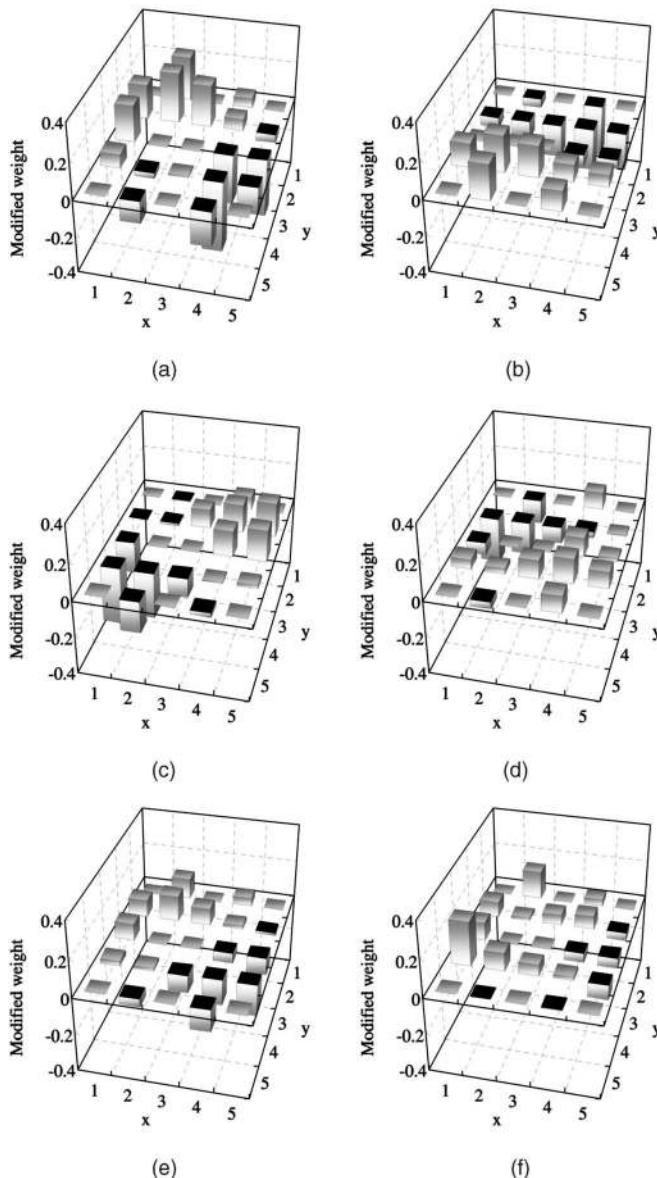

Fig. 12. The modified weights A_{qm} of the approximate network of the trained NEE. The six graphs correspond to the six networks that are connected to six units in the hidden layer.

layer corresponds to one feature calculated from the input pixel values, the result of optimization suggested that 17 features were used for the function of edge enhancement from noisy images in the trained NEE. The effective units in the input layer were within the square region consisting of five by five pixels, as shown in Fig. 10.

Furthermore, in order to gain insight into the nonlinear kernel of the trained NEE, we analyzed it by approximating a sigmoid function to a linear function. The input pixel values to the NEE are rewritten as

$$\{g(x-i, y-j)/G_M | i, j \in R_S\} = \{I_1, I_2, \cdots, I_q, \cdots I_{N_I}\},$$

(20)

where q is a unit number in the input layer and N_I is the number of units in the input layer. The output of the mth unit in the hidden layer is represented by

Fig. 11. Modification of the network of the trained NEE for analysis. (a) Original network. (b) Approximate network.
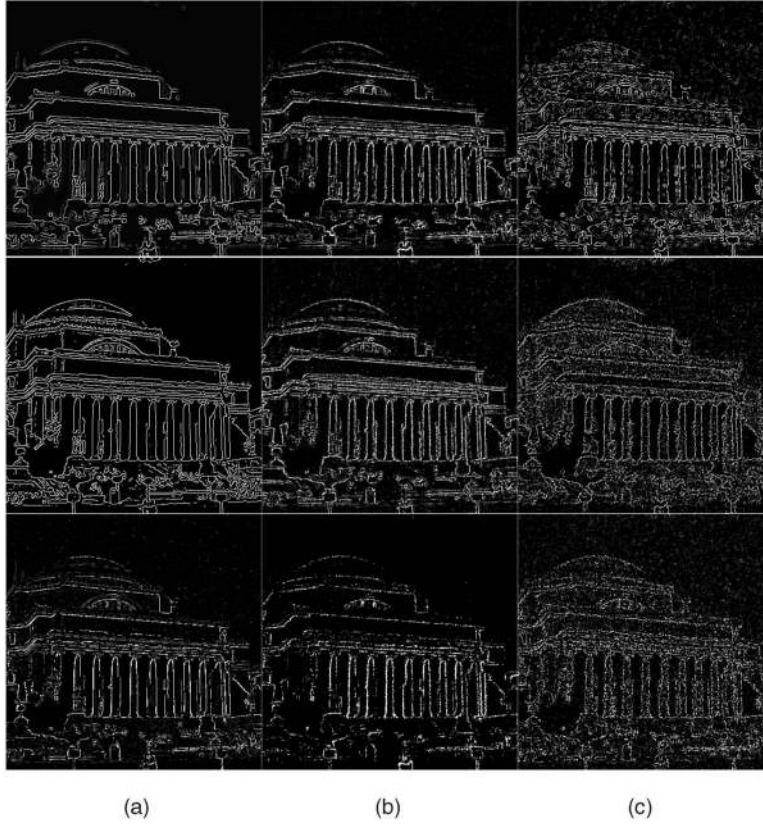
Fig. 13. Performance of the NEE on the change of a teaching edge type. The teaching edges were made by use of three target edge detectors. (Target edge detector: from top to bottom, the Marr-Hildreth operator, the Canny edge detector, and the Hueckel operator). (a) Teaching edges. (b) Edges enhanced by the neural edge enhancer for the noisy Columbia image. (c) Edges enhanced by the target edge detector for the noisy Columbia image.

$$O_m^H = f_S\left(\sum_{q=1}^{N_I} I_q \cdot W_{qm}^H + W_{0m}^H\right), \qquad (21)$$

where f_S is a sigmoid function, W_{qm}^H is a weight between the qth unit in the input layer and mth unit in the hidden layer, and W_{0m}^H is a bias. The output of the NEE is represented by

$$\hat{f}_E(x,y) = \sum_{m=1}^{N_H} O_m^H \cdot W_m^O + W_0^O, \qquad (22)$$

where N_H is the number of units in the hidden layer, W_m^O is a weight between the mth unit in the hidden layer and the unit in the output layer, and W_0^O is a bias.

Because the nonlinearity of a sigmoid function prevents us from analyzing the network easily, let us now approximate a sigmoid function with a linear function for analysis. The derivative of the sigmoid function at an offset can be approximated by the gain of the linear function. By use of this approximation, the output of the mth unit in the hidden layer can be represented by

$$O_m^H = f_S\left(\sum_{q=1}^{N_I} I_q \cdot W_{qm}^H + W_{0m}^H\right)$$
$$\approx f_S'\left(W_{0m}^H\right) \cdot \sum_{q=1}^{N_I} I_q \cdot W_{qm}^H. \qquad (23)$$

Hence, the output of the NEE can be represented by

$$\hat{f}_E(x,y) = \sum_{m=1}^{N_H}\left[\sum_{q=1}^{N_I}\left\{W_{qm}^H \cdot f_S'\left(W_{0m}^H\right) \cdot W_m^O \cdot I_q\right\}\right]$$
$$+ W_0^O \equiv \sum_{m=1}^{N_H}\sum_{q=1}^{N_I} A_{qm} \cdot I_q + W_0^O. \qquad (24)$$

The above modification of the network is shown in Fig. 11.

We modified the network of the NEE trained with a natural image after optimizing the structure. The modified weights A_{qm} of the approximate network are shown in Fig. 12. Because one unit in the hidden layer corresponds to one feature, each network that is connected to a unit in the hidden layer is shown separately in Fig. 12. The six graphs correspond to six networks. The six most effective networks were selected on the basis of the sum of the modified weights. The five-by-five matrices correspond to the input region of the trained NEE. The black square indicates the pixel having a negative weight. The pixels having the same sign correspond to a smoothing operation, whereas the pixels having the opposite sign correspond to an edge-enhancement operation. It is interesting to note that Fig. 12 is reminiscent of the receptive fields of various simple units in the cat and monkey cerebral cortex discovered by Hubel and Wiesel [68]. With cat and monkey, these neural filters are acquired during the critical period just after birth [69]. The modified weights in Fig. 12a indicate the operation for diagonal edge enhancement together with smoothing. The

TABLE 2
Quantitative Evaluation of the Performance of the NEEs Trained with Three Types of Teaching Edges (Training Image: Columbia)

Target one	Mean absolute error [%]					
	MHO		CED		HO	
	OMHO	NEE_{MHO}	OCED	NEE_{CED}	OHO	NEE_{HO}
Columbia	4.1	3.6	15.8	14.4	4.4	3.5
Teapot	2.8	3.7	14.9	7.1	4.2	3.1
Car	3.5	2.1	13.3	11.5	3.5	2.5
House	4.3	3.4	14.4	13.7	4.6	3.7
Peppers	2.9	2.0	13.4	11.3	3.4	2.6
Man	3.9	2.8	14.2	12.5	3.8	2.9
Airplane	3.8	2.6	15.5	11.3	4.7	3.5
Lena	4.5	2.4	13.9	11.0	3.4	2.3
Crowd	4.2	4.0	14.2	15.0	4.4	3.5
Lake	3.9	3.7	14.9	14.3	4.8	4.0
Bridge	5.5	4.9	16.2	16.6	4.9	4.1
Woman1	3.1	2.0	15.0	11.7	3.6	2.8
Couple	3.7	3.2	14.5	13.9	4.2	3.4
Girl	3.6	1.8	11.9	10.8	2.7	2.0
Average	3.8	3.0	14.3	12.4	4.0	3.1

OMHO: Optimized Marr-Hildreth operator;

OCED: Optimized Canny edge detector; OHO: Optimized Hueckel operator;

NEE: Neural edge enhancer. The suffix refers to each NEE's target edge detector.

modified weights in Figs. 12b and 12c function mainly as vertical edge enhancement together with horizontal smoothing and as edge enhancement with smoothing for another diagonal orientation, respectively. The modified weights in Fig. 12d are for diagonal edge enhancement with smoothing, but the scale is smaller. These features (functions) are integrated by summation in the approximate network. Because this analysis is based on a linear approximation, the integration actually contains some nonlinearity. The results of the analysis suggest that the trained NEE uses directional gradient operators with smoothing. These directional gradient operators with smoothing followed by integration with nonlinearity lead to robust enhancement against noise.

4.5 Changing Teaching Edge Types

In order to investigate whether the NEE can learn various types of edges, we carried out an experiment with various types of teaching edges. The NEEs were trained with three types of teaching edges. The teaching edges were made by the use of three target edge detectors, the Marr-Hildreth operator, the Canny edge detector, and the Hueckel operator. The three teaching edges are shown in Fig. 13a. Three NEEs were trained, and the trainings converged with errors E of 1.05 percent ~ 2.82 percent. The edges enhanced by the trained NEEs for the noisy Columbia image are shown in Fig. 13b. The edges enhanced by each NEE are similar to each of the teaching edges. In the results for the Marr-Hildreth operator and the Canny edge detector, the

edges are enhanced as thin lines similar to the teaching edges. In the results for the Hueckel operator, the edges of the columns of the structure are enhanced as strong edges. This result indicated that each NEE was able to acquire the features of each target edge enhancer.

Furthermore, in order to compare the performance of the NEE with that of the target edge detectors for noisy images, the target edge detectors were optimized with the training images. The results are shown in Fig. 13c. In the results of the target edge detectors, the noise is enhanced by mistake, and the edges are discontinuous. It should be noted that the results for nontraining images were similar to those for the training images. The results of the quantitative evaluation are shown in Table 2. The average values in the table were calculated from nontraining images. These results showed that the performance of each NEE was higher than that of the target edge detector.

4.6 Edge Localization Method

We propose an edge localization method for the NEE, which is a method for finding the edge locations from the edges enhanced by the NEE. An edge localization method produces a binary image where each pixel is assigned to an edge or a background. A labeling algorithm [70] may be applied on the binary image to determine which edges are connected to each other. The proposed edge localization method consists of 1) nonmaximum suppression and 2) hysteresis thresholding. In the nonmaximum suppression, the pixel value is set to zero except for the pixel having a local maximum. First, we assume
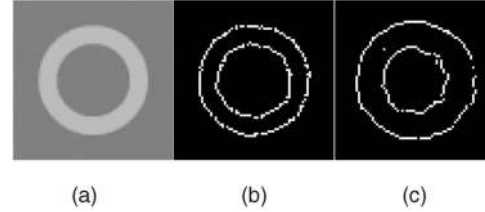
Fig. 15. Comparison of edge detection of the NEE with that of a leading edge detector for an artificial image with constant additive white Gaussian noise. (a) Artificial image without noise. (b) Edges detected by the NEE with the edge localization method. (c) Edges detected by the Elder edge detector.

(a)   (b)

Fig. 14. Results of the detection of edge locations by use of the proposed edge localization method (applies to the edges enhanced by the NEE where the target edge enhancer is the Sobel filter). (a) After nonmaximum suppression. (b) After hysteresis thresholding.

that the edges form a thin line, the width of which is one pixel. If an eight-connected line which does not branch off crosses a local region ($u \times u$ pixels) and one of the pixels of the line is on the center pixel (x, y), the descending order of the edge magnitude at the center pixel in the local region should be within $u + 1$th. Therefore, the local maximum of the edge magnitude can be extracted by use of the following equation:

$$f_S(x, y; n) = \vartheta_n\Big\{\hat{f}_E(x - i, y - j)|i, j \in R_{uu}\Big\}, \qquad (25)$$

where $\vartheta_n\{\cdot\}$ is an operator extracting the nth greater value, and R_{uu} is the local region. If the edge magnitude of the object pixel is greater than the uth greater edge magnitude in the local region, zero is put in the object pixel as follows:

$$f_{LM}(x, y) = \begin{cases} \hat{f}_E(x, y) & if \ \hat{f}_E(x, y) \geq f_S(x, y; u) \\ 0 & otherwise. \end{cases} \qquad (26)$$

Using this method, we can set the pixel value to zero except for the pixel having the local maximum. Next, hysteresis thresholding is applied to the results of the nonmaximum suppression. We adopted the hysteresis thresholding in [16]. The results of edge localization in the case that $u = 3$

are shown in Fig. 14. These images are the results for the edges enhanced by the NEEs when the Sobel filter is the target edge enhancer. These results indicate that the edge locations could be identified effectively by use of the proposed edge localization method.

In order to compare the NEE with a leading edge detector, we carried out an experiment on edge detection. The performance of the NEE was compared with the Elder edge detector [71]. Because the Elder edge detector assumes that the noise in images is constant additive white Gaussian noise, images with this type of noise were used in this experiment. The parameters of the Elder edge detector were determined such that the highest performance was obtained, according to a method described in [71]. The NEE trained with the image with quantum noise was applied to noisy images. The result for an artificial image is shown in Fig. 15. The SNR_T of the input noisy image was 3 dB. In the result of the Elder edge detector, there are few false positives and the detected edges are clear. However, the locations of the edges are slightly different from the true locations, whereas the locations of the edges detected by the NEE with the edge localization method correspond better to the true locations. This effect of the Elder edge detector would be improved by use of an accurate method for subpixel localization [72]. The results for natural images are shown in Fig. 16. The SNR_I of the Columbia image, that of the teapot image, and that of the man image were 3.52 dB, 4.59 dB, and 5.80 dB, respectively. The Elder edge detector produces clear, continuous edges. However, the edges are warped slightly. The edges detected by the NEE with the edge localization method are similar to the results for the images with quantum noise. This result shows that the performance of the NEE did not depend very much on the type of noise.

5 CONCLUSIONS

In this paper, a new supervised edge enhancer based on a modified multilayer NN, called a neural edge enhancer (NEE), is proposed for enhancing the desired edges clearly from noisy images. By comparison with conventional edge enhancers/detectors, the following was demonstrated: the NEE was robust against noise, was able to enhance clear, continuous edges from noisy images, and was superior to the conventional edge detectors in similarity to the desired edges. An experiment to investigate the performance of the NEE for various teaching edge types was performed. The results showed that the NEEs were able to learn the various types of teaching edges. In order to gain insight into the nonlinear kernel of the NEE, we performed analyses on the trained NEE. The results suggested that the trained NEE

(a)   (b)

Fig. 16. Comparison of edge detection of the NEE with that of a leading edge detector for natural images with constant additive white Gaussian noise. (a) Edges detected by the NEE with the edge localization method. (b) Edges detected by the Elder edge detector.

used directional gradient operators with smoothing. Furthermore, we introduced the edge localization method for the NEE, and we presented the results of edge detection. In our experiments, the NEE was proven to be useful for enhancing edges from noisy images.

Although the performance of the NEE was superior to that of the conventional edge detectors, the execution time for training was long (63 hours on a workstation, UltraSPARC-II 300 MHz, Sun Microsystems, for a reduced structure, 17-17-1). Existing acceleration methods for training [73] would be useful for improving the efficiency. However, once the training is finished, the execution time for edge enhancement itself is short (6.4 seconds for 512×512 pixels). For real-time applications, the development of an efficient algorithm for the NEE is needed. The efficient realization method for the NFs in [30], [34] can be applied to the NEE. By use of the efficient realization method, the trained NEE can be represented by an efficient approximate network.

We recently extended the concept of the NEE and developed an NN-based scheme for distinction between specific patterns and other patterns in medical images [74]. We plan to extend the scheme to accommodate various

tasks in general images. We plan to study a combination of the NEE and a leading model-based edge/contour localization method, such as SNAKES or a dictionary-based edge labeling method [75], and to evaluate the performance. We also plan to study improving the performance of the NEE by simulating the early visual systems of humans.

References

[1]   A. Rosenfeld and A.C. Kak, *Digital Picture Processing,* second ed., vol. 2, pp. 84-112. Academic Press, 1982.
[2]   W.K. Pratt, *Digital Image Processing,* second ed., pp. 491-556. John Wiley & Sons, 1991.
[3]   J.C. Russ, *Image Processing Handbook,* second ed., pp. 225-262. Fla.: CRC Press, 1995.
[4]   J.R. Parker, *Algorithm for Image Processing and Computer Vision,* pp. 1-66. John Wiley & Sons, 1997.
[5]   B. Jahne, *Digital Image Processing,* fourth ed., pp. 322-340. Berlin: Springer-Verlag, 1997.
[6]   M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l J. Computer Vision,* vol. 1, no. 3, pp. 321-331, 1988.
[7]   R.O. Duda and P.E. Hart, *Pattern Classification and Scene Analysis,* pp. 267-272. Wiley, 1971.
[8]   M.H. Hueckel, "An Operator which Locates Edges in Digitized Pictures," *J. ACM,* vol. 18, no. 1, pp. 113-125, Jan. 1971.
[9]   M.H. Hueckel, "A Local Visual Operator which Recognizes Edges and Lines," *J. ACM,* vol. 20, no. 4, pp. 634-647, 1973.
[10] L. Mero and Z. Vassy, "A Simplified and Fast Version of the Hueckel Operator for Finding Optimal Edges in Pictures," *Proc. Int'l Joint Conf. Artificial Intelligence,* pp. 650-655, 1977.
[11] G.S. Robinson, "Edge Detection by Compass Gradient Masks," *Computer Graphics and Image Processing,* vol. 6, pp. 492-501, 1977.
[12] W. Frei and C. Chen, "Fast Boundary Detection: A Generalization and a New Algorithm," *IEEE Trans. Computers,* vol. 26, no. 10, pp. 988-998, Oct. 1977.
[13] K.S. Shanmugan, F.M. Dickey, and J.A. Green, "An Optimal Frequency Domain Filter for Edge Detection in Digital Pictures," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 1, no. 1, pp. 37-49, Jan. 1979.
[14] D. Marr and E. Hildreth, "Theory of Edge Detection," *Proc. Royal Soc. London,* vol. B207, pp. 187-21, 1980.
[15] W.H.H.J. Lunsher, "The Asymptotic Optimal Frequency Domain Filter for Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, pp. 678-680, 1983.
[16] J.F. Canny, "A Computational Approach to Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, pp. 679-698, 1986.
[17] V. Torre and T.A. Poggio, "On Edge Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, pp. 147-163, Mar. 1986.
[18] W.H.H.J. Lunsher and M.P. Beddoes, "Optimal Edge Detector Design I: Parameter Selection and Noise Effects," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, pp. 164-177, Mar. 1986.

[19] M. Petrou and J. Kittler, "Optimal Edge Detector for Ramp Edges," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 5, pp. 483-491, May 1991.

[20] J. Shen and S. Castan, "An Optimal Linear Operator for Step Edge Detection," *Computer Vision, Graphics, and Image Processing: Graphical Models and Understanding,* vol. 54, pp. 112-133, 1992.

[21] S.M. Bhandarkar, Y. Zhang, and W.D. Potter, "An Edge Detection Technique Using Genetic Algorithm-Based Optimization," *Pattern Recognition,* vol. 27, no. 9, pp. 1159-1180, 1994.

[22] J. Yoo, E.J. Coyle, and C.A. Bouman, "Dual Stack Filters and the Modified Difference of Estimates Approach to Edge Detection," *IEEE Trans. Image Processing,* vol. 6, pp. 483-491, Dec. 1997.

[23] T. Lindeberg, "Edge Detection and Ridge Detection with Automatic Scale Selection," *Int'l J. Computer Vision,* vol. 30, pp. 117-154, 1998.

[24] K. Arakawa and H. Harashima, "A Nonlinear Digital Filter Using Multi-Layered Neural Networks," *Proc. IEEE Int'l Conf. Comm.,* vol. 2, pp. 424-428, 1990.

[25] L. Yin, J. Astola, and Y. Neuvo, "A New Class of Nonlinear Filters—Neural Filters," *IEEE Trans. Signal Processing,* vol. 41, no. 3, pp. 1201-1222, Mar. 1993.

[26] Z.Z. Zhang and N. Ansari, "Structure and Properties of Generalized Adaptive Neural Filters for Signal Enhancement," *IEEE Trans. Neural Networks,* vol. 7, no. 4, pp. 857-868, July 1996.

[27] L. Yin, J. Astola, and Y. Neuvo, "Adaptive Multistage Weighted Order Statistic Filters Based on the Back Propagation Algorithm," *IEEE Trans. Signal Processing,* vol. 42, pp. 419-422, Feb. 1994.

[28] K. Suzuki, I. Horiba, N. Sugie, and M. Nanki, "A Recurrent Neural Filter for Reducing Noise in Medical X-Ray Image Sequences," *Proc. Int'l Conf. Neural Information Processing,* vol. 1, pp. 157-160, Oct. 1998.

[29] K. Suzuki, I. Horiba, N. Sugie, and M. Nanki, "Noise Reduction of Medical X-Ray Image Sequences Using a Neural Filter with Spatiotemporal Inputs," *Proc. Int'l Symp. Noise Reduction for Imaging & Comm. Systems,* pp. 85-90, Nov. 1998.

[30] K. Suzuki, I. Horiba, and N. Sugie, "Efficient Approximation of a Neural Filter for Quantum Noise Removal in X-Ray Images," *Neural Networks for Signal Processing IX,* Y.-H. Hu et al., eds., pp. 370-379, Aug. 1999.

[31] K. Suzuki, I. Horiba, and N. Sugie, "Signal-Preserving Training for Neural Networks for Signal Processing," *Proc. IEEE Int'l Symp. Intelligent Signal Processing and Comm. Systems,* vol. 1, pp. 292-297, Nov. 2000.

[32] K. Suzuki, I. Horiba, and N. Sugie, "Neural Filter with Selection of Input Features and Its Application to Image Quality Improvement of Medical Image Sequences," *Proc. IEEE Int'l Symp. Intelligent Signal Processing and Comm. Systems,* vol. 2, pp. 783-788, Nov. 2000.

[33] K. Suzuki, I. Horiba, and N. Sugie, "Training under Achievement Quotient Criterion," *Neural Networks for Signal Processing X,* B. Widrow et al., eds., pp. 537-546, 2000.

[34] K. Suzuki, I. Horiba, and N. Sugie, "Efficient Approximation of Neural Filters for Removing Quantum Noise from Images," *IEEE Trans. Signal Processing,* vol. 50, no. 7, pp. 1787-1799, July 2002.

[35] K. Suzuki, I. Horiba, N. Sugie, and M. Nanki, "Neural Filter with Selection of Input Features and Its Application to Image Quality Improvement of Medical Image Sequences," *IEICE Trans. Information and Systems,* vol. E85-D, no. 10, pp. 1710-1718, Oct. 2002.

[36] I.N. Aizenberg, "Processing of Noisy and Small-Detailed Gray-Scale Image Using Cellular Neural Networks," *J. Electronic Imaging,* vol. 6, no. 3, pp. 272-285, 1997.

[37] C. Rekeczky, A. Schultz, I. Szatmari, T. Roska, and L.O. Chua, "Image Segmentation and Edge Detection via Constrained Diffusion and Adaptive Morphology: A CNN Approach to Bubble/Debris Image Enhancement," *Proc. Int'l Symp. Nonlinear Theory and Application,* pp. 209-212, 1997.

[38] C. Guzelis and S. Karamahmut, "Recurrent Perceptron Learning Algorithm for CNNs with Application to Edge Detection," *Proc. IEEE Int'l Conf. Image Proc.,* pp. 1134-1139, 1997.

[39] I.N. Aizenberg, N.N. Aizenberg, and J. Vandewalle, "Precise Edge Detection: Representation by Boolean Functions, Implementations on the CNN," *Proc. IEEE Int'l Workshop Cellular Neural Networks and Their Applications,* pp. 301-306, Apr. 1998.

[40] Cs. Rekeczky, T. Roska, and A. Ushida, "CNN-Based Difference-Controlled Adaptive Nonlinear Image Filters," *Int'l J. Circuit Theory and Application,* vol. 26, pp. 375-423, 1998.

[41] D.L. Vilarino, D. Cabello, M. Balsi, and V.M. Brea, "Image Segmentation Based on Active Contours Using Discrete-Time Cellular Neural Networks," *Proc. IEEE Int'l Workshop Cellular Neural Networks and Their Applications,* pp. 331-336, 1998.

[42] I.N. Aizenberg, N.N. Aizenberg, T. Bregin, C. Butakov, and E. Farberov, "Image Processing Using Cellular Neural Networks Based on Multi-Valued and Universal Binary Neurons," *Proc. Neural Networks for Signal Processing X,* B. Widrow et al., eds., pp. 557-566, 2000.

[43] I.N. Aizenberg, N.N. Aizenberg, and J. Vandewalle, *Multi-Valued and Universal Binary Neurons–Theory, Learning and Applications.* Kluwer Academic, 2000.

[44] H. Nagai, Y. Miyanaga, and K. Tochinai, "An Edge Detection by Using Self-Organization," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 2749-2752, 1998.

[45] L. Guan, S.W. Perry, R. Romagnoli, H. Wong, and H. Kong, "Neural Vision System and Applications in Image Processing and Analysis," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing,* pp. 1245-1248, 1998.

[46] P.J. Toivanen, J. Ansamaki, S. Leppajarvi, and J.P.S. Parkkinen, "Edge Detection of Multispectral Images Using the 1-D Self Organizing Map," *Proc. Int'l Conf. Artificial Neural Networks,* pp. 737-742, 1998.

[47] M.S. Bhuiyan, M. Sato, H. Fujimoto, and A. Iwata, "Edge Detection by Neural Network with Line Process," *Proc. Int'l Joint Conf. Neural Networks,* pp. 1223-1226, 1993.

[48] M.S. Bhuiyan, M. Sato, H. Fujimoto, and A. Iwata, "An Improved Neural Network Based Edge Detection Method," *Proc. Int'l Conf. Neural Information Processing,* vol. 1, pp. 620-625, Oct. 1994.

[49] H. Iwata, T. Agui, and H. Nagahashi, "Boundary Detection of Color Images Using Neural Networks," *Proc. IEEE Int'l Conf. Neural Networks,* pp. 1426-1429, 1995.

[50] M. Muneyasu, K. Hotta, and T. Hinamoto, "Image Restoration by Hopfield Networks Considering the Line Process," *Proc. IEEE Int'l Conf. Neural Networks,* pp. 1703-1706, 1995.

[51] G. Ohashi, A. Ohya, M. Natori, and M. Nakajima, "Boundary Estimation Method for Ultrasonic 3-D Imaging," *Proc. SPIE Medical Imaging, Image Processing,* vol. 1898, pp. 480-486, 1993.

[52] G. Coppini, R. Poli, and G. Valli, "Recovery of the 3-D Shape of the Left Ventricle from Echocardiographic Images," *IEEE Trans. Medical Imaging,* vol. 14, no. 2, pp. 301-317, June 1995.

[53] S.W. Lu and J. Shen, "Artificial Neural Networks for Boundary Extraction," *Proc. IEEE Int'l Conf. Systems, Man, and Cybernetics,* vol. 3, pp. 2270-2275, 1996.

[54] Z. He and M.Y. Siyal, "Edge Detection with BP Neural Networks," *Proc. Int'l Conf. Signal Processing,* vol. 2, pp. 1382-1384, 1998.

[55] K. Suzuki, I. Horiba, N. Sugie, and M. Nanki, "Computer-Aided Diagnosis System for Coronary Artery Stenosis Using a Neural Network," *Proc. SPIE Medical Imaging: Image Processing,* vol. 4322, pp. 1771-1782, Feb. 2001.

[56] K. Suzuki, I. Horiba, K. Ikegaya, and M. Nanki, "Recognition of Coronary Arterial Stenosis Using Neural Network on DSA System," *Systems and Computers in Japan,* vol. 26, no. 8, pp. 66-74, Aug. 1995.

[57] K. Funahashi, "On the Approximate Realization of Continuous Mappings by Neural Networks," *Neural Networks,* vol. 2, pp. 183-192, 1989.

[58] A.R. Barron, "Universal Approximation Bounds for Superpositions of a Sigmoidal Function," *IEEE Trans. Information Theory,* vol. 39, no. 3, pp. 930-945, Mar. 1993.

[59] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Representations of Back-Propagation Errors," *Nature,* vol. 323, pp. 533-536, 1986.

[60] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, "Learning Internal Representations by Error Propagation," *Parallel Distributed Processing,* vol. 1, chapter 8, pp. 318-362, MIT Press, 1986.

[61] C.M. Bishop, *Neural Networks for Pattern Recognition,* pp. 230-240. Oxford Univ. Press, 1995.

[62] I.E. Adbou and W.K. Pratt, "Quantitative Design and Evaluation of Enhancement/Thresholding Edge Detectors," *Proc. IEEE,* vol. 67, no. 5, pp. 753-763, May 1979.

[63] J.C. Brailean, R.P. Kleihorst, S. Efstratiadis, A.K. Katsaggelos, and R.L. Lagendijk, "Noise Reduction Filters for Dynamic Image Sequences: A Review," *Proc. IEEE,* vol. 83, no. 9, pp. 1270-1291, Sept. 1995.

[64] M.R. Banham and A.K. Katsaggelos, "Digital Image Restoration," *IEEE Signal Processing Magazine,* vol. 14, no. 2, pp. 24-41, Mar. 1997.

[65] K. Suzuki, I. Horiba, and N. Sugie, "Designing the Optimal Structure of a Neural Filter," *Neural Networks for Signal Processing VIII,* M. Niranjan et al., eds., pp. 323-332, Aug. 1998.

[66] K. Suzuki, I. Horiba, and N. Sugie, "An Approach to Synthesize Filters with Reduced Structures Using a Neural Network," *Quantum Information II,* T. Hida and K. Saito, eds., pp. 205-218, 2000.

[67] K. Suzuki, I. Horiba, and N. Sugie, "A Simple Neural Network Pruning Algorithm with Application to Filter Synthesis," *Neural Processing Letters,* vol. 13, no. 1, pp. 43-53, Feb. 2001.

[68] D.H. Hubel and T.N. Wiesel, "Receptive Fields, Binocular Interaction and Functional Architecture in the Cat's Visual Cortex," *J. Physiology (London),* vol. 160, pp. 106-154, 1962.

[69] C. Blakemore and G.F. Cooper, "Development of the Brain Depends on the Visual Environment," *Nature,* vol. 228, pp. 477-478, 1970.

[70] K. Suzuki and I. Horiba, and N. Sugie, "Linear-Time Connected-Component Labeling Based on Sequential Local Operations," *Computer Vision and Image Understanding,* vol. 85, no. 1, pp. 1-23, Jan. 2003.

[71] J.H. Elder and S.W. Zucker, "Local Scale Control for Edge Detection and Blur Estimation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 7, pp. 699-716, July 1998.

[72] J.H. Elder and S.W. Zucker, "Space Scale Localization, Blur, and Contour-Based Image Coding," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 27-34, June 1996.

[73] S. Hayk, *Neural Networks—A Comprehensive Foundation,* second ed., pp. 233-234. Prentice Hall, 1999.

[74] K. Suzuki, S.G. Armato III, F. Li, S. Sone, and K. Doi, "Massive Training Artificial Neural Network (MTANN) for Reduction of False Positives in Computerized Detection of Lung Nodules in Low-Dose CT," *Medical Physics,* vol. 30, no. 7, pp. 1602-1617, July 2003.

[75] E.R. Hancock and J. Kittler, "Edge-Labeling Using Dictionary-Based Relaxation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 2, pp. 165-181, Feb. 1990.

Kenji Suzuki received the BS and MS degrees, both with highest honors, in electrical and electronic engineering from Meijo University, Nagoya, Japan, in 1991 and 1993, respectively, and the PhD degree in information engineering by a thesis from Nagoya University, Nagoya, Japan, in 2001. From 1993 to 1997, he was with the Research and Development Center at Hitachi Medical Corporation, Kashiwa, Japan, as a researcher. He was engaged in research and development of intelligent medical imaging systems, including a digital angiography system and a digital radiography system. In 1997, he joined Aichi Prefectural University, Aichi, Japan, where he assisted on founding the Faculty of Information Science and Technology. From 1998 to 2002, he was a research associate in the Faculty of Information Science and Technology at Aichi Prefectural University. During the period of 2001 to 2002, he was on leave with the Kurt Rossmann Laboratories for Radiologic Image Research, the Department of Radiology, the Division of Biological Sciences, The University of Chicago, Illinois, as a research associate. From 2002 to 2003, he was a research associate in the Kurt Rossmann Laboratories for Radiologic Image Research, the Department of Radiology, the Division of Biological Sciences, the University of Chicago. Since 2003, he has been a research assiciate (instructor) at the same university. He received the Paul C. Hodges Award from the Department of Radiology, The University of Chicago in 2002. He was a member of the research group organized by the Aichi Science and Technology Foundation, Japan, from 1997 to 1998. He was an investigator of the research project promoted by the Ministry of Education, Science, Sports, and Culture of Japan, Quantum Information Theoretical Approach to Life Science, in the Frontiers of Science and Technology at Meijo University from 1997 to 2002. His research interests include neural networks for image processing and pattern recognition, computer-aided diagnosis, and image processing suggested by the human visual systems. He is a member of the IEEE, IEICE, IEEJ, IPSJ, JNNS, and JCS.

Isao Horiba received the BS and PhD degrees in electrical engineering from Nagoya University, Nagoya, Japan, in 1974 and 1985, respectively. From 1974 to 1987, he was with the Research and Development Center at Hitachi Medical Corporation, Kashiwa, Japan, as a senior researcher. He was engaged in research and development of medical imaging systems, including a computed topography system, a magnetic resonance imaging system, a digital subtraction angiography system, and a digital radiography system. In 1987, he joined the Faculty of Science and Technology at Meijo University, Nagoya, Japan, as an assistant professor and then an associate professor. Since 1998, he has been a professor on the Faculty of Information Science and Technology at Aichi Prefectural University, Aichi, Japan. He received the Best Technology Award from the Japan Society of Medical Electronics and Biological Engineering and the Best Patent Award from the Japan Institute of Invention and Innovation in 1988 and 1990, respectively. He has published more than 100 papers and is the coauthor of three books in the area of medical engineering and image understanding for intelligent transportation systems. He was a senator of IPSJ from 1997 to 1999. His research interests include intelligent image processing, image processing for medical systems, and image understanding for intelligent transportation systems. He is a member of the IEICE, IPSJ, JSMIT, JSUM, and JMEBE.

Noboru Sugie received the BS degree in electrical engineering from Nagoya University, Nagoya, Japan, in 1957, and the PhD degree in engineering from the University of Tokyo, Tokyo, Japan, in 1970. From 1957 to 1979, he was with the Government Electrotechnical Laboratory, Tsukuba, Japan. During the period of 1962 to 1964, he was on leave with the Department of Electrical Engineering, McGill University, Montreal, Canada, as an NRC Postdoctoral Fellow. He had been the director of the sections of both bionics, and human and computer vision at the Government Electrotechnical Laboratory, from 1970 to 1977 and from 1977 to 1978, respectively. From 1979 to 1994, he was a professor in the Department of Information Engineering and the Graduate School of Engineering at Nagoya University. In the meantime, from 1990 to 1994, he was the director of Nagoya University Computation Center. He is now a professor emeritus of Nagoya University. From 1994 to 2000, he was a professor in the Department of Electrical and Electronic Engineering and the Graduate School of Science and Technology at Meijo University, Nagoya, Japan. From 2000 to 2002, he was the founding chairman of the Department of Information Sciences at Meijo University. Since 2000, he has been a professor in the Department of Information Sciences. He received the Technological Achievement Award from the Society of Instrument and Control Engineers in 1996. He has published more than 300 papers and is an author or editor of more than 50 books in the area of computer vision, human vision, biocybernetics, and natural language processing. He was an executive director of Japanese Society for AI, Japan Society for Simulation Technology, and Society of Biomechanisms Japan, a director of IPSJ and ITE, and an advisor of JMEBE and Japanese Neural Network Society. He is a fellow of the IEICE and Robotics Society of Japan. He served as an editor or an associate editor of many journals and a member of organizing committees of many international conferences. His research interests include computer vision, human vision, biocybernetics, and natural language processing. He is a member of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** http://computer.org/publications/dlib.