

# Structural Feature Extraction by Neural Networks

Rei-Yao Wu<sup>§</sup> and Wen-Hsiang Tsai<sup>\*</sup>

<sup>§</sup>Institute of Computer Science and Information Engineering

<sup>\*</sup>Department of Computer and Information Science

National Chiao Tung University

Hsinchu, Taiwan 300

Republic of China

## Abstract

A neural network system for structural feature extraction from thinned binary images is proposed. The system consists of two neural networks, called the line separation network (LSN) and the feature extraction network (FEN). The LSN includes several neuron planes which classify input image lines into groups of lines in distinct directions. And the FEN constructs structural features from the lines with known directions. The LSN is a recurrent network which collects information about neighbors in wide areas by information propagation through iterations. So, even curvy lines can be obtained. This in turn makes it easier for the FEN to extract features. Good experimental results show the effectiveness of the proposed neural network system.

## 1. Introduction

Intersection points, such as crosses, forks, corners, line ends, etc., in line images are useful for many applications, like optical character recognition, engineering drawing recognition, etc. They are especially suitable for the recognition of Chinese characters since most Chinese characters are composed of linear strokes.

Neural networks have been proved to be effective tools for pattern recognition [1-3]. In a neural network implementation of an image recognition system, the features fed into the neural network are usually extracted by algorithmic approaches using computers, not by the neural network itself. Nevertheless, Fukushima [4] successfully derived features from binary images and recognized numeral characters using the neocognitron neural network. Graf et al. [5] used a VLSI architecture to extract features by multiplying a binary input image with a stored matrix of weights.

Extracting features by neural networks is efficient since the processing elements in a network work in parallel. In this paper, a new neural network scheme is proposed for extracting the features of various line intersection and end points. Some examples of the features which can be extracted by the proposed neural network system are shown in Fig. 1. Because all the feature points come from the intersections or ends of one or more lines, it makes sense to extract these features from lines in different directions rather than to extract them directly from original object images. Based on this principle, lines in different directions are extracted from input images first in the proposed approach. Desired structural feature points are then derived from the extracted lines.

One way of extracting a line from the image is to label each pixel simply by considering the condition of its local neighbors, as is done by the neocognitron [4]. A problem of this approach is that bad line continuation will be produced when pixels on a curvy line are labeled (i.e. a curvy line will be broken into several pieces after the extraction process). Low tolerance of noise or distortion in line images will result from this problem when the features are extracted. To avoid this problem, the proposed approach iteratively gather more global information to extract lines, in a manner like the relaxation algorithm [9]. Lines with good continuation (like those strokes in Chinese characters) can be obtained by this approach.

Each image processed by the proposed approach is a line pattern with single line width, which comes from the thinning result of an input binary image. A one-pass parallel thinning algorithm, which was proposed by the authors, shown to have good thinning effect, and implemented by two neural networks [6-8], is used here to obtain thinned images.

To systematically extract desired structural features, lines in four different directions are first identified by a neural network called line separation network (LSN). The lines derived by the LSN are then used to extract various structural features by a neural network called feature extraction network (FEN).

<sup>§</sup> Also with Institute for Information Industry, Taipci, Taiwan 106, Republic of China.

<sup>\*</sup> To whom all correspondence should be sent.



The remainder of this paper is organized as follows. Section II describes the architecture of the LSN. Section III describes how to extract structural features by the FEN. Some examples of extracted lines and features are given in Section IV to illustrate the effectiveness of the proposed networks. Finally, Section V includes some conclusion remarks.

## II. Line Separation Network (LSN)

### A. Design principle

The objective of identifying lines by directions using the LSN is to obtain proper stroke segments as a preprocessing step of structural feature extraction. This task is merely to assign each pixel to one or more classes with each class collecting all the pixels on the lines in one of four directions (vertical, horizontal, and diagonal lines). Pixels being assigned to a class constitute all the line segments in an identical direction. Note that a line in a specific direction may be a freehand line, not necessarily a strictly straight line. A pixel might be assigned to more than one class. For example, when a pixel is located at the cross point of a vertical line and a horizontal line, it will be assigned to two classes. An ideal case is to obtain the line segments which match exactly the composing strokes of a character. But this is difficult to achieve if only a small local area is observed, since a stroke is not necessarily a strictly straight line. To achieve correct labeling, observing pixels in a properly wider area is necessary. However, processing pixels in a wider area is difficult and inefficient. Another approach is to use an iterative method. That is, for each pixel to be labeled, besides the pixel itself, the local neighbors of the pixel are also considered to determine which class the pixel belongs to. Useful information are then propagated wider and wider by gathering the neighboring information iteratively to correct improper initial labeling. By this way, for example, pixels above and below a pixel on a vertical line encourage the pixel to be considered as one on the vertical line. The proposed LSN has this type of iterative pixel labeling capability.

### B. L-layer

The structure of the LSN is shown in Fig. 2. The LSN consists of two layers, namely, the line layer (L-layer) and the hidden layer (H-layer). There are four neuron planes in the L-layer with their sizes all identical to that of the image being processed. Each of them is used to collect the black points which are located on the lines in one of the four directions. Let them be called the V-plane, H-plane, W-plane, and E-plane for identifying vertical, horizontal, northwest-to-southeast diagonal and northeast-to-southwest diagonal lines, respectively. Each neuron in a

neuron plane is indexed by two integers  $i$  and  $j$  to indicate the row and column in which it is located. For example, the neuron in the  $i$ -th row and  $j$ -th column in the V-plane is denoted as  $V_{ij}$ . The value of each neuron in each of these planes is a confidence measure of whether the corresponding image pixel is located on a line in the direction which the neuron plane is responsible for. Initially, an input line image is fed into these four neuron planes by assigning to all of those neurons in each plane, which are spatially corresponding to the black pixels in the input image, a unique confidence value, and to all of those neurons, which are spatially corresponding to the white image pixels, the unique value of 0. After the operation of the LSN stabilizes, a neuron in a neuron plane with a high confidence value represents the fact that the corresponding pixel in the input image has high possibility to be a point on a line in the direction which the neuron plane is responsible for.

### C. H-layer

The other layer of the LSN is the H-layer. As shown in Fig. 2, corresponding to each neuron plane in the L-layer are two neuron planes in the H-layer also with their sizes identical to that of the input image. For example, neuron planes HV1-plane and HV2-plane in the H-layer correspond to neuron plane V-plane in the L-layer. The other neuron planes corresponding to the H-plane, W-plane, and E-plane are the HH1-plane and HH2-plane, HW1-plane and HW2-plane, and HE1-plane and HE2-plane, respectively. There are two planes for each direction, and either of them is called the dual plane of the other. Corresponding neurons in a pair of dual planes are called dual neurons. Each neuron in this layer receives inputs from the neighbors of the corresponding neuron in the L-layer and from the corresponding neurons in the other neuron planes of the L-layer. For example, neuron  $HV1_{ij}$  in the HV1-plane receives inputs from the neighbors of neuron  $V_{ij}$  in the V-plane and inputs from neurons  $H_{ij}$ ,  $W_{ij}$ , and  $E_{ij}$  in the other neuron planes of the L-layer. A neighbor of  $V_{ij}$ , which is possible to be located on a vertical line, together with  $V_{ij}$  will excite  $HV1_{ij}$ ; while a neighbor of  $V_{ij}$  without this possibility will inhibit  $HV1_{ij}$ . Neurons  $H_{ij}$ ,  $W_{ij}$ , and  $E_{ij}$  will also inhibit  $HV1_{ij}$ . The receptive field of each neuron in the other neuron planes can be described similarly.

Furthermore, the outputs of a pair of neurons, which are located respectively at identical positions in two dual planes, are fed back into the corresponding neuron which is located on the corresponding neuron plane in the L-layer. For example,  $HV1_{ij}$  and  $HV2_{ij}$  are fed back into  $V_{ij}$ . The role of a neuron  $M$  in the H-layer is to gather both positive and negative information about the confidence support from the L-layer and then feed this information back into its corresponding neuron  $N$  in the L-layer, which, together with the information coming from the dual

neuron of  $M_i$  is used for the evaluation of the new confidence value of  $N$ .

Confidence supports from two weight settings of the neighborhood are gathered independently. By setting proper weights, the neighborhood information of a neuron  $V_{ij}$  in the L-layer is collected into neurons  $HV1_{ij}$  and  $HV2_{ij}$ . A high value of either of the two neurons  $HV1_{ij}$  and  $HV2_{ij}$  then supports increment of the confidence value of neuron  $V_{ij}$ . Weight settings for lines in the other directions are proceeded similarly.

#### D. Descriptions of weight setting

At the beginning, a neuron in the four neuron planes in the L-layer is initialized with a positive constant value  $c$  if the corresponding pixel in the image is black. Let X-plane be one of the planes in the L-layer, i.e., let  $X$  be one of  $V$ ,  $H$ ,  $W$ , and  $E$ . The connection weight from a neighbor  $N$  of a neuron  $X_{ij}$  in a X-plane in the L-layer to neurons  $HX1_{ij}$  and  $HX2_{ij}$  in the H-layer are set to a strong weight  $W_s$ , a weak weight  $W_w$ , or an inhibitory negative weight  $W_h$  in the way discussed previously if both of the image pixels  $P_N$  and  $P_{ij}$  corresponding to  $N$  and  $X_{ij}$ , respectively, are black pixels in the input image. And the weights from neuron  $X_{ij}$  to  $HX1_{ij}$  and  $HX2_{ij}$  are set to another strong weight  $W_c$  if  $P_{ij}$  is a black pixel. For example, assume that  $P_{i-1,j}$ ,  $P_{ij}$ , and  $P_{i+1,j+1}$  are black pixels, then the connection weights from neurons  $V_{i-1,j}$ ,  $V_{ij}$ , and  $V_{i+1,j+1}$  to neuron  $HV2_{ij}$  are initialized to be  $W_s$ ,  $W_c$ , and  $W_w$ , respectively, the weights from neurons  $V_{i-1,j-1}$ ,  $V_{i-1,j+1}$ ,  $V_{i,j-1}$ , and  $V_{i,j+1}$  to neuron  $HV2_{ij}$  are set to  $W_h$ , and the weights from neurons  $V_{i+1,j-1}$  and  $V_{i+1,j}$  to neuron  $HV2_{ij}$  are set to 0

In addition to the inputs from the neighborhood of  $X_{ij}$ , neurons  $HX1_{ij}$  and  $HX2_{ij}$  also receive inputs from the corresponding neurons in the other neuron planes in the L-layer with negative connection weights. For example, neuron  $HV1_{ij}$  receives inhibitory inputs from neurons  $H_{ij}$ ,  $W_{ij}$ , and  $E_{ij}$ . These inhibitory inputs reduce the confidence value of a neuron in the L-layer if the pixel corresponding to the neuron is not located on a line in the correct direction. The value of a neuron in the L-layer will converge to either the high value 1 or the low value 0 after a sufficiently large number of iterations.

In summary, the net input  $HX1_{ij}^I$  of a neuron  $HX1_{ij}$  in the H-layer is

$$HX1_{ij}^I = \sum_{\substack{i-1 \leq k \leq i+1 \\ j-1 \leq l \leq j+1}} WX_{ijk}^1 * X_{kl}^0 + \sum_{Y \in X} W_b * Y_{ij}^0 \quad (1)$$

where  $X$  denotes one of the symbols  $V$ ,  $H$ ,  $W$ , or  $E$ ;  $WX_{ijk}^1$  is the weight (one of  $W_c$ ,  $W_s$ ,  $W_w$ , and  $W_h$  mentioned previously) from neuron  $X_{kl}$  to neuron  $HX1_{ij}$ ;  $X_{kl}^0$  is the output of neuron  $X_{kl}$ ;  $Y$  is one of the symbols  $V$ ,  $H$ ,  $W$ , or  $E$ , and is not  $X$ ;  $W_b$  is the inhibitory connection weight from the corresponding neuron in each of the other neuron planes; and  $Y_{ij}^0$  is the output of  $Y_{ij}$ . The net input  $HX2_{ij}^I$  of neuron  $HX2_{ij}$  can be obtained similarly as follows:

$$HX2_{ij}^I = \sum_{\substack{i-1 \leq k \leq i+1 \\ j-1 \leq l \leq j+1}} WX_{ijk}^2 * X_{kl}^0 + \sum_{Y \in X} W_b * Y_{ij}^0 \quad (2)$$

where  $WX_{ijk}^2$  is the weight from neuron  $X_{kl}$  to neuron  $HX2_{ij}$ .

The outputs  $HX1_{ij}^0$  and  $HX2_{ij}^0$  of neurons  $HX1_{ij}$  and  $HX2_{ij}$  are

$$HX1_{ij}^0 = F_H(HX1_{ij}^I) \quad (3)$$

and

$$HX2_{ij}^0 = F_H(HX2_{ij}^I), \quad (4)$$

respectively, where  $F_H$  is a nonlinear function described by

$$F_H(x) = \begin{cases} 1 & \text{if } x \geq 1; \\ 0 & \text{if } x \leq 0; \\ x & \text{otherwise.} \end{cases}$$

Neuron  $X_{ij}$  in the L-layer receives inputs from neurons  $HX1_{ij}$  and  $HX2_{ij}$  in the H-layer. Its output  $X_{ij}^0$  is

$$X_{ij}^0 = F_L(HX1_{ij}^0, HX2_{ij}^0), \quad (5)$$

where  $F_L$  is a function described by

$$F_L(x1, x2) = \begin{cases} x1 & \text{if } x1 \geq x2; \\ x2 & \text{otherwise.} \end{cases}$$

### III. Feature Extraction Network (FEN)

#### A. Design principle



After lines are classified according to their directions by the LSN, the FEN extracts structural features both from the neuron planes in the L-layer of the LSN and from an image neuron plane, called I-plane, derived directly from the image whose neurons corresponding to black and white pixels are active and inactive, respectively. As shown in Fig. 3, the combination of the I-plane and all the neuron planes in the L-layer of the LSN is called the N-layer in the FEN. Features are extracted by the feature planes in another layer of the FEN, called the feature layer (F-layer), in a way of one feature one plane. That is, for each type of structural feature to be derived, the FEN uses a neuron plane to construct it. Specifically, the orthogonal cross points of vertical lines and horizontal lines are constructed by a neuron plane, and the upper-left corner points of rectangles are constructed by another feature plane, and so on. The number of required neuron planes in the F-layer is equal to the number of feature types to be extracted. After the operation of the FEN stabilizes, a neuron with a high value in a feature plane means that there is high possibility that a corresponding feature is located at a position corresponding to the neuron in the input image.

Neurons in different feature planes receive inputs from different planes in the N-layer. For example, the neurons in the feature plane used to extract cross points receive inputs both from the neurons in the V-plane and from those in the H-plane, while the neurons in the feature plane used to extract crosses of St. Andrew receive inputs both from the neurons in the W-plane and from those in the E-plane. As described previously, the I-plane is also included in the N-layer. It is used to inhibit simple features (through proper setting of negative weights) when a complex feature exists at that location. For example, in addition to receiving inputs from a line plane (i.e., from one of the V-, H-, E-, W-planes in the N-layer), the neurons in the feature plane used to extract end points from lines in a specific direction also receive inhibitory inputs from the I-plane to avoid yielding high values in the outputs of the neurons corresponding to fork points.

## B. Connection weight setting

A neuron in a feature plane in the F-layer receives inputs from the 3x3 neighborhoods of the corresponding neurons in two or more neuron planes in the N-layer. For a neuron  $R_{ij}$  in a feature plane R in the F-layer which receives inputs from a set of neuron planes S in the N-layer, its net input  $R_{ij}^I$  is

$$R_{ij}^I = \sum_{X \in S} \sum_{\substack{-1 \leq k \leq 1 \\ -1 \leq l \leq 1}} WRX_{kl} * X_{i+k, j+l}^O \quad (6)$$

where X is a neuron plane in S;  $X_{i+k, j+l}^O$  is the output of neuron  $X_{i+k, j+l}$ ; and  $WRX_{kl}$  is the connection weight from neuron  $X_{i+k, j+l}$  to neuron  $R_{ij}$ .

The output of neuron  $R_{ij}$  is

$$R_{ij}^O = F_R(R_{ij}^I) \quad (7)$$

where  $F_R$  is a nonlinear function described by

$$F_R(x) = \begin{cases} 1 & \text{if } x \geq 1; \\ 0 & \text{if } x \leq 0; \\ x & \text{otherwise.} \end{cases}$$

In addition to the selection of the neuron planes in the N-layer which should be connected to each neuron plane in the F-layer, another important issue in the extraction of a specific type of feature is the setting of the connection weights from the neurons in the N-layer to the neurons in the F-layer. The output value of a neuron in a specific feature plane in the F-layer indicates the degree of possibility that a corresponding feature point exists at that location, so all the neurons in a feature plane are treated identically except their locations. That is, the setting of the connection weights of the receptive fields is treated identically for all the neurons in a neuron plane in the F-layer. For each type of structural feature, the setting of the connection weights depends on the spatial location of the input neurons in the 3x3 receptive field as well as the plane they are located in, and reflects the importance of the existence of a line point at that location. Examples of the connection weights from neurons in a local area of the neuron planes in the N-layer to a neuron  $R_{ij}$  in the F-layer are shown in Table 1.

## IV. Experimental Results

The proposed neural networks have been implemented by software simulation. And experiments have been conducted to see the effects of the proposed neural networks for line identification and feature extraction. Images fed into the networks are of dimension 32x32. The initial confidence value  $c$  is 0.25 for all neurons in the L-layer corresponding to the black pixels in the image. The weights from the L-layer to the H-layer are 0.55, 0.51, 0.25, -0.21, and -0.05 for  $W_C$ ,  $W_S$ ,  $W_W$ ,  $W_H$ , and  $W_B$ , respectively. Fig. 4 shows the results of line classification for a Chinese character obtained by the LSN after 12 iterations. In the figure, symbol '.' marks the object skeleton of the input image; the decimal symbol '1' shows the confidence value of a neuron for the value range 0.1 to 0.2, symbol '2' for the range 0.2 to 0.3, and so on. It can be seen from the figure that line continuation is preserved for each line in each of the four directions. Also, high confidence values are possible to exist at the same position in two or more neuron planes. For example, the cross points have confidence values greater than 0.9 both in the V-plane and the H-

plane. Presence of a feature can be obtained by thresholding the output values of the F-layer. Fig. 5 shows the features extracted from the neuron planes shown in Fig. 4. Instead of showing all the feature planes, a plane consisting of all the features is shown. In the figure, symbol '·' is used to mark the object skeleton again, and the alpha-numerical symbols are used to mark the feature points and are coincident with the labels used in Fig. 1. Note that feature points are marked correctly though deformation appears near them. This shows the effectiveness of the iteration nature of the proposed approach.

## V. Conclusions

A neural network system for structural feature extraction has been proposed in this paper. The system consists of two neural networks, namely, the LSN and the FEN. The LSN classifies black pixels into different classes according to the directions of the lines they are located on. And the FEN extracts structural features from the results of the LSN.

Instead of deriving features directly from the image plane, the proposed scheme first identifies pixels on lines in specific directions using the LSN. Like the manner of a relaxation algorithm, the LSN propagates the support information along lines to tolerate line deformation. The intersection point of two lines can thus be successfully labeled by activating the neurons in the planes responsible for both lines regardless of the disturbance from each other. On the other hand, features cannot be correctly extracted by observing the local area around the feature points if deformation appears near the feature points. After the discovery of the directions of the lines in which pixels are located, this problem can be solved by considering the directions of the line patterns, and this is what the FEN performs. In short, the directions of lines derived by the LSN make it easier for the FEN to extract features.

Structural features are useful for further analysis or recognition of images, for example, for optical character

recognition (OCR). Recognition of image contents based on the structural features derived by this scheme is worthwhile for further studies. Note that the lines extracted according to their directions by the LSN are also good features for image recognition or analysis. This is also a good topic for future investigations.

## References

- [1] D. E. Rumelhart, and J. L. McClelland, *Parallel Distributed Processing: Explorations in The Microstructure of Cognition*, Volume 1, MIT Press, Cambridge, MA, 1986.
- [2] G. A. Carpenter, and S. Grossberg, A Massively Parallel Architecture for A Self-organizing Neural Pattern Recognition Machine. *Computer Visions, Graphics, and Image Processing*, 37, pp. 54-115, 1987.
- [3] D. L. Reilly, L. N. Cooper, and C. Elbaum, A Neural Model for Category Learning. *Biological Cybernetics*, 45, pp. 35-41, 1982.
- [4] K. Fukushima, Neocognitron: A Self-organizing Neural Network Model for A Mechanism of Pattern Recognition Unaffected by Shift in Position, *Biological Cybernetics*, 36, pp. 193-202, 1980.
- [5] H. P. Graf, L. D. Jackel, and W. E. Hubbard, VLSI Implementation of a Neural Network Model, *IEEE Computer*, 41-48, 1988.
- [6] R. Y. Wu and W. H. Tsai, A New One-Pass Parallel Thinning Algorithm for Binary Images, to appear in *Pattern Recognition Letters*, 13, pp. 715-723, 1992.
- [7] R. Y. Wu, and W. H. Tsai, Parallel Thinning by A Recurrent Neural Network, submitted.
- [8] R. Y. Wu, and W. H. Tsai, A Single-Layer Neural Network for Parallel Thinning, to be appeared in *International Journal of Neural System*.
- [9] A. Rosenfeld, and A. C. Kak, "Digital Picture Processing," Vol. 2, Academic Press, New York, 1982.



**Table 1. Weights from the N-layer to a neuron  $R_{ij}$  in the F-layer (the heading of each of columns 3 through 11 specify the indices of a neuron in the N-layer).**

Feature	From	$i-1,j-1$	$i-1,j$	$i-1,j+1$	$i,j-1$	$i,j$	$i,j+1$	$i+1,j-1$	$i+1,j$	$i+1,j+1$
+	V	0.09	0.18	0.09	-0.18	0.18	0.00	0.09	0.15	0.09
	H	0.09	-0.18	0.09	0.15	0.18	0.18	0.09	0.00	0.09
X	W	0.20	0.10	0.00	0.10	0.10	0.10	0.00	0.10	0.20
	E	0.00	0.10	0.20	0.10	0.10	0.10	0.20	0.10	0.00
└	V	0.00	0.20	0.00	-0.20	0.20	-0.20	-0.20	-0.30	-0.20
	H	0.00	0.00	0.00	0.20	0.20	0.20	0.10	0.00	0.10
/	W	0.20	0.00	0.00	0.00	0.20	-0.20	0.00	-0.20	-0.30
	E	0.00	0.00	0.20	0.00	0.20	0.10	0.20	0.10	0.00
┌	V	-0.50	-0.50	-0.50	-0.50	0.00	0.00	0.00	0.50	0.00
	H	-0.50	-0.50	0.00	-0.50	0.00	0.50	-0.50	0.00	0.00
^	W	-0.30	-0.30	-0.30	-0.30	0.30	0.30	-0.30	-0.15	0.30
	E	-0.30	-0.30	-0.30	-0.15	0.30	-0.30	0.30	0.15	-0.30
↑	V	0.00	0.00	0.00	0.00	0.50	0.00	0.25	0.50	0.25
	I	-0.75	-0.75	-0.75	-0.50	0.00	-0.50	-0.25	0.00	-0.25
↙	W	0.00	0.00	0.00	0.00	0.50	0.25	0.00	0.25	0.50
	I	-0.75	-0.75	-0.50	-0.75	0.00	-0.25	-0.50	-0.25	0.00

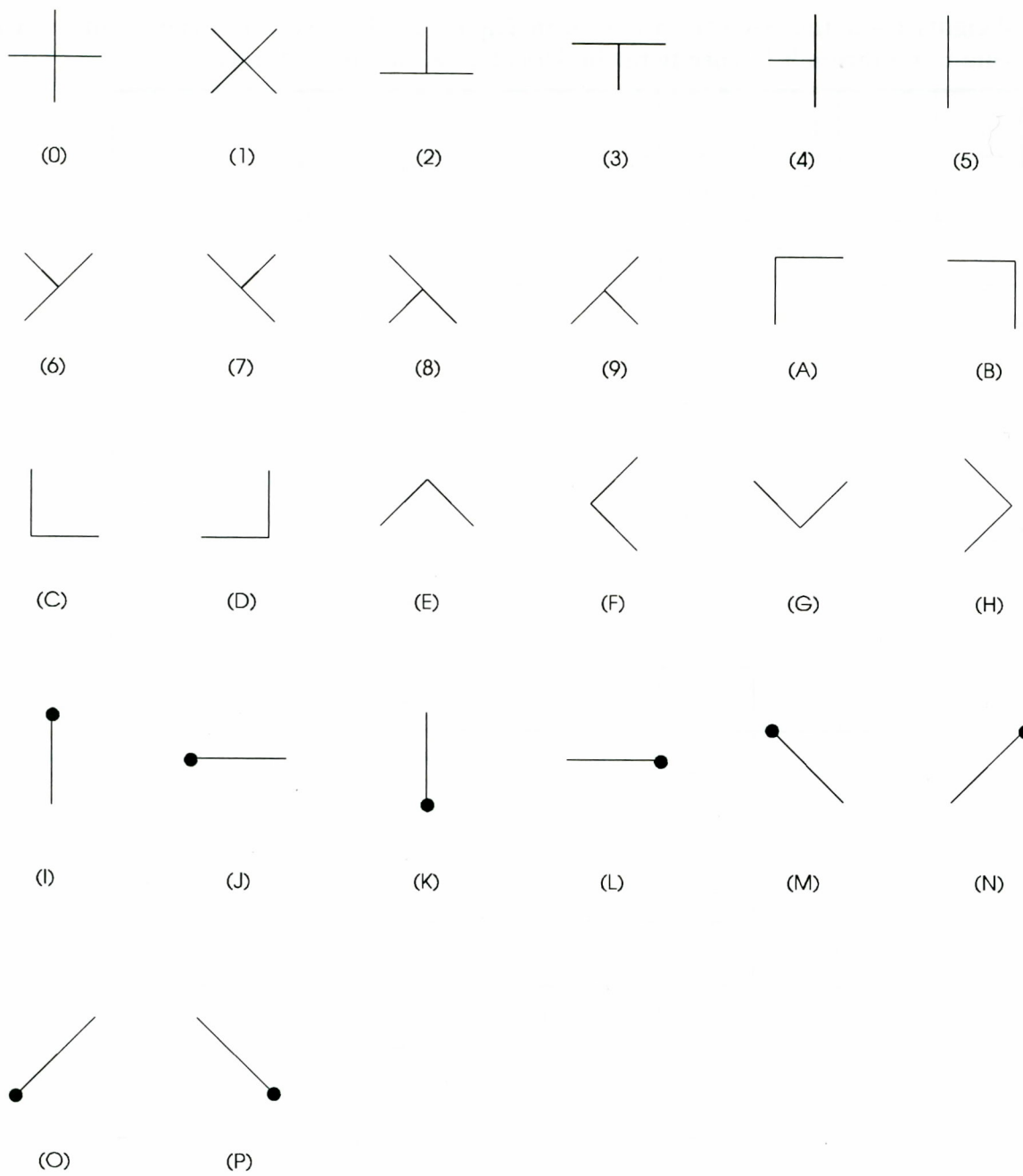


Fig. 1. Examples of structural features which can be extracted by proposed neural network system.

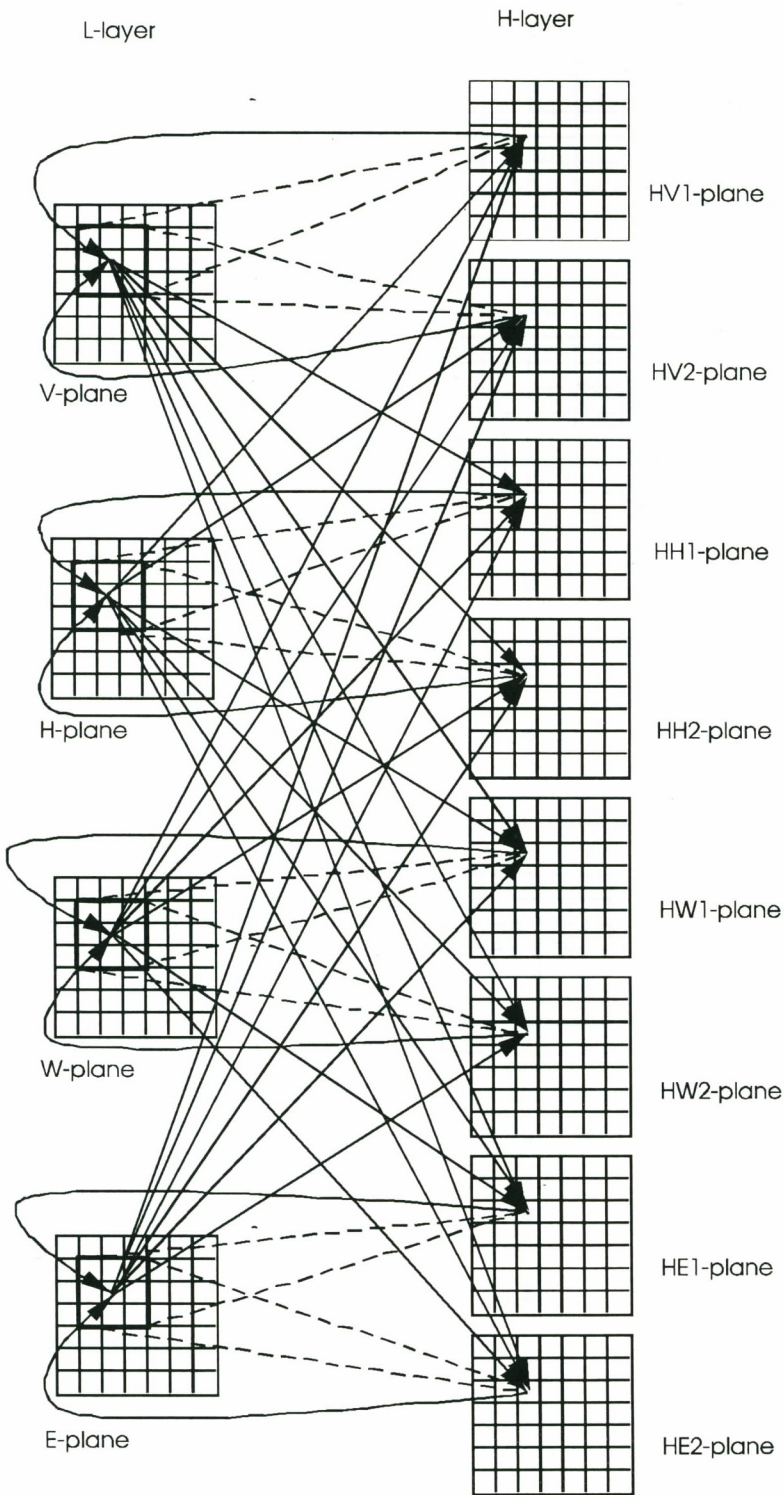


Fig. 2. Structure of the LSN.



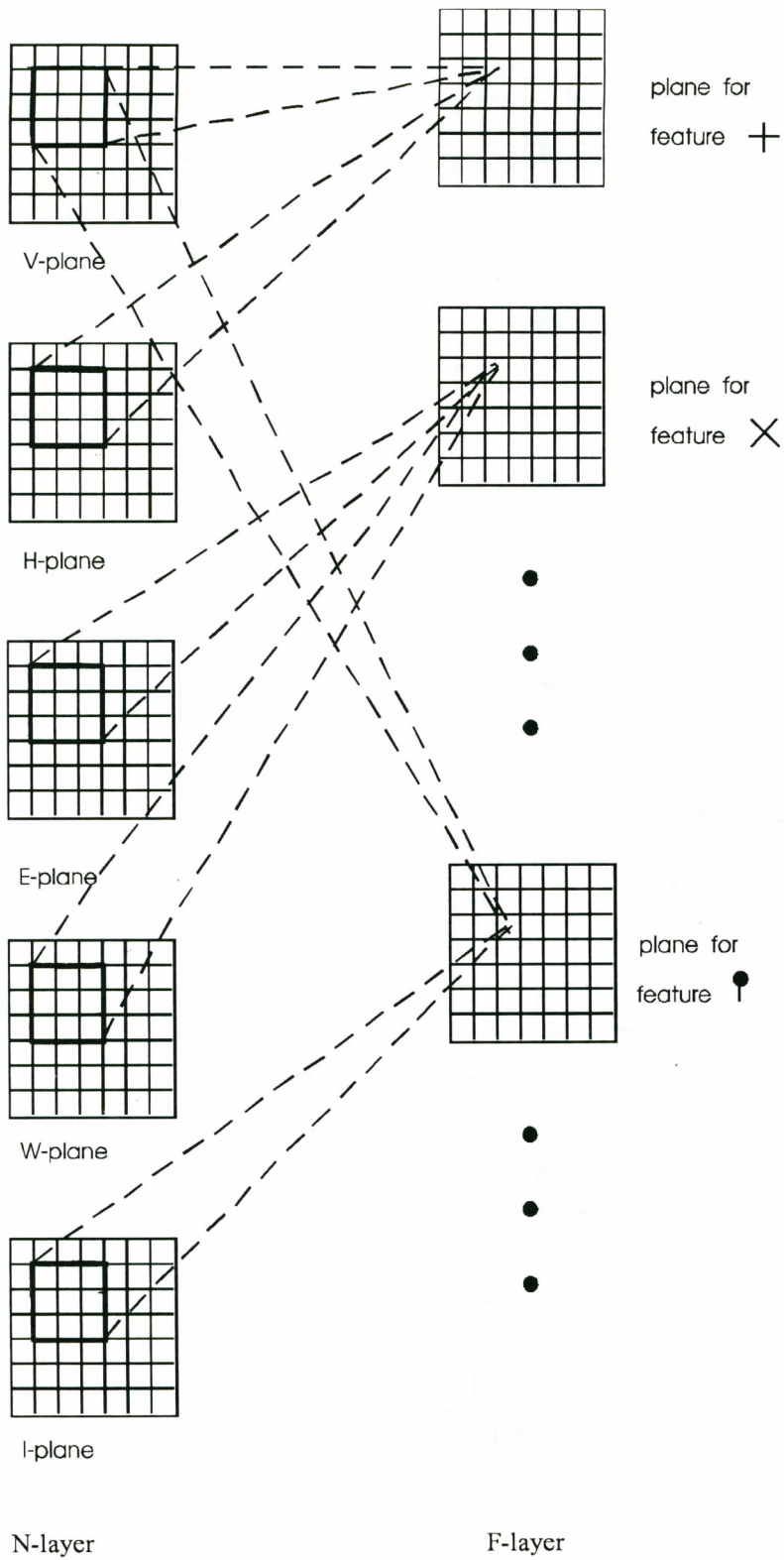


Fig. 3. Structure of FEN.

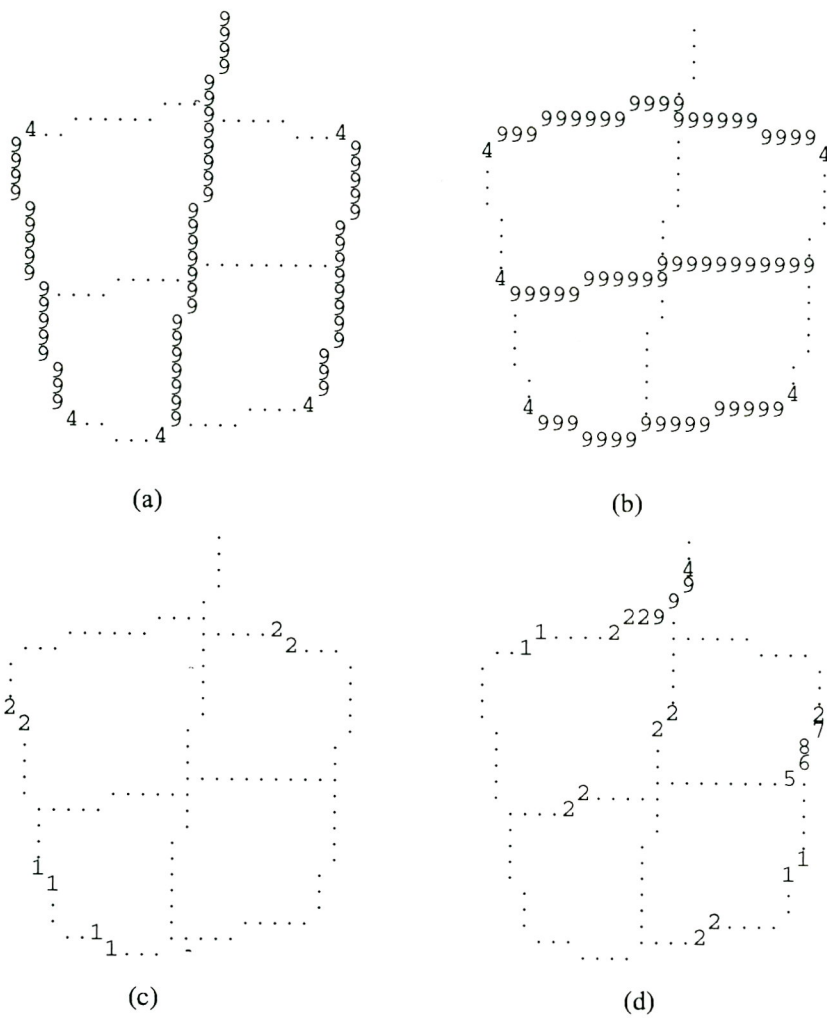


Fig. 4. Results of line separation. (a) Vertical lines obtained by the V-plane. (b) Horizontal lines obtained by the H-plane. (c) Diagonal lines obtained by the W-plane. (d) Diagonal lines obtained by the E-plane.

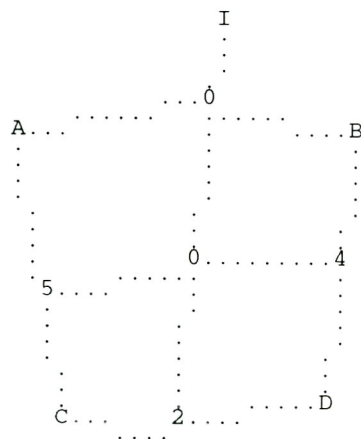


Fig. 5. Result of feature extraction. The alpha-numerical symbols used to mark feature points are coincident with the labels used in Fig. 1.