[Trans. of the Society of Instrument and Control Engineers]
Vol. E-1, No.1, 171/186 (2001)

Rotation-Invariant Neural Pattern Recognition Systems with Application to Coin Recognition

Minoru Fukumi*, Sigeru Omatu**, Fumiaki Takeda*** and Toshihisa Kosaka****

This paper considers rotation-invariant pattern recognition systems using artificial neural networks. First, five neural pattern recognition systems are described. Next, they are applied to a variably rotated coin recognition problem to show those effectiveness. A 500 Japanese yen coin and a 500 Korean won coin classified in this paper have the same shape, size, and thickness, and have a similar pattern. In relation to such a fact, there was the report on misclassification between those coins in a newspaper.

From the results of computer simulation for coin recognition, considerations on computational complexity in hardware implementation of those systems and on modeling of our brains are described.

Key words: neural networks, rotation-invariance, back-propagation, coin recognition, training

1. Introduction

Humans can recognize any pattern easily even if it is transformed by scale-change, translation, rotation, and noise. However it is difficult for digital computers to recognize such patterns. In this background, artificial neural networks, which are models emulating a biological neuron network, are actively used to perform pattern recognition.

Up to now, artificial neural network invariant to any transform such as rotation, translation, scale, and deformation, have been presented. Fukushima [1] proposed the Neocognitron which is a model of a visual system in brains and showed invariance to translation and deformation of input patterns. However, the limitation to rotation and scale. has been reported [2][3] and it has not been applied to gray scale images. Widrow [4] presented a multi-layered ADALINE network which is insensitive to translation and rotation by 90 degree of input patterns.

* University of Tokushima, Minami-Josanjima, Tokushima
** University of Osaka Prefecture, Sakai, Osaka
*** Kochi University of Technology, Tosa-Yamada, Kochi
**** Glory Ltd. Simoteno, Himeji

However rotational invariance by any degree has not been presented. Koch [5] proposed a similar method. The authors extended the Widrow's system to be tolerant to noise [6]-[10] and presented systems to insensitive to rotation by any degree [11]-[15]. Reid [16] constructed the system insensitive to translation, rotation, and scale by using a higher order neural network in preprocessing. However, this causes a combinatorial explosion of an input pattern and is not practical.

There have been many pattern recognition systems insensitive to transform [17]-[20]. However a practical system invariant to rotation, translation, and scale in the same time has never been presented. Another approach to transform invariant pattern recognition is to utilize mathematical functions which can produce input signals insensitive to various transform. Then these signals are learned and recognized by classifiers as neural networks. This approach includes mathematical frameworks, such as Moment invariants [21]-[24], Fourier transform [25]-[27], and Hough transform [28]. Note that contour extraction is required for gray scale images before these mathematical transforms [19].

On the one hand, there are a few practical systems emulating a visual system in brains. To clarify brain functions, it is important to construct systems modeling the visual system in engineering. This paper therefore tries to construct neural net based systems as long as possible. As a result, this can be a cue to clarify brain functions.

In this paper neural network systems are utilized to perform rotation-invariant pattern recognition and applied to rotated coin recognition problems. If the sizes of coins to be classified are different, their classification can be easily done. If the coins have the same size, they should be recognized by their image pattern. A Japanese 500 yen and a Korean 500 won coins to be classified in this paper are very similar in size, weight, color, and pattern structure. Therefore misclassification of those coins often happens at toll gate in highway. There is no problem if they have the same value. However 500 won has only the value of about 80 yen in June 30 1992 [29] and this fact causes a social problem.

Gray scale images taken out of real coin classifiers in a bank are usually rotated by random degree in comparison to a standard image. In order to recognize such images, a pattern recognition architecture with rotational invariance is necessary.

Authors have considered different kinds of neural pattern recognition systems to recognize rotated coin images. These are effective for engineering problems but cause difficulty in computational complexity to implement a real hardware. In this paper therefore we introduce our work [6]-[15],[30]-[32] and the present systems which are easy to implement and have a fewer computation. Concretely speaking, size reduction of networks by a structural learning and decrease of computational complexity using a network with feedback connections are achieved.

The section 2 describes the architectures of rotation invariant systems. The sections 3 to 6 are devoted to show their definite structures. Finally the section 7 considers their systems by means of computer simulations.

2. Architectures of rotation-invariant neural pattern recognition systems

It is possible in principle to construct a system insensitive to translation, rotation, scale by designing an appropriate method for each of them. In image data processing (e.g. 256×256 pixels), however, it is impossible to perform computer simulations of such a system because the system with a preprocessing part becomes huge [20]. On the one hand, location and extraction of the center of images taken out of an image scanner are easy to be done. Furthermore, image data treated in this paper are taken in from the automatic coin classification machine and no change in size. In this paper therefore methods to form rotation invariant pattern recognition systems by using neural networks are described.

Neural network based rotation invariant systems include the following methods:

(A) Preprocessing (mathematical transform) + BP [12][26] ・・・Transform model

(B) Preprocessing (neural net) + BP [11]-[15] ・・・Slab model

(C) Preprocessing (edge detective neural net) + BP with link weight [30]-[32] ・・・Edge detection model

(D) BP with link weight [33][34] ・・・BP model

(E) BP with feedback connections ・・・ Feedback model

where BP means the error back-propagation method [33]. The Fourier transform is utilized as a mathematical transform in (A). An improved slab architecture proposed by Widrow is used as a preprocessing. The preprocessor in (C) extracts edge features of coin images. This can be regarded as a procedure of feature detection performed in the visual field of brains [35]-[39]. The BP with link weight in (C) and (D) has a structure of link weight which can be insensitive to rotation of input patterns and cannot change output values of its network.. The methods of (A) to (D) require relatively many computations. Therefore they need a kind of technique to reduce it. The section 6 presents a method to reduce a computational complexity for the transform model (A). The method of (E) is different approach from the others and tries to learn and recognize rotated coin images using a network with feedback connections. This network achieves reduction of computational quantity by using a simple architecture without any preprocessor.

(B) and (C) in the above mentioned systems finally aim at modeling brain functions. (A) is a system for the purpose of application to real machines in an engineering point of view. (E) is a system trying to achieve both.

This paper considers the methods of (A), (B), (C), and (E) and compares their effectiveness through computer simulations for coin recognition. Simulation results on the method (D) are described without explaining its detail.

The sections 3 and 4 describe the slab model and the edge detection model, respectively. The sections 5 and 6 present a recognition method using the feedback model and a method to reduce computational complexity using the transform model.

3. Rotation invariant systems with a slab architecture

3.1 Rotation-invariant pattern recognition systems

A pattern recognition system considered in this section is illustrated in Fig.1. The system consists of two parts, a fixed preprocessing network (Box of slabs) and a trainable a-CONE network, as shown in [4][6]-[10]. The conventional system is insensitive to rotation only by 90 degree but the present systems can be invariant to rotation by any degree..

3.2 A structure of the preprocessor

As shown in Fig.1, the preprocessor (Box of slabs) is composed of many slabs. Each slab includes many neuron and one majority vote taker. The circles labeled "N" in the preprocessor indicate the sigmoid neuron units and "M" is the majority vote taker. The majority in this paper, however, produces an analog output value, which is different from the conventional system [4]. The sigmoid voting can be invariant to change of input signals and noise tolerant. This sigmoid neuron unit has input connections with the same value.

Each slab in the preprocessor produces a single output, which is an input signal to the trainable multi-layered a-CONE neural network.. Therefore the number of input units in the a-CONE network is the same as the number of slabs in the preprocessor. The problem is how to determine weights of each neuron unit in order to obtain rotation invariant slab outputs. The rotational system with many slabs. Invariance by 90 degree has been already proposed by Widrow. This paper extends the Widrow's system to be invariant to rotation by any degree..

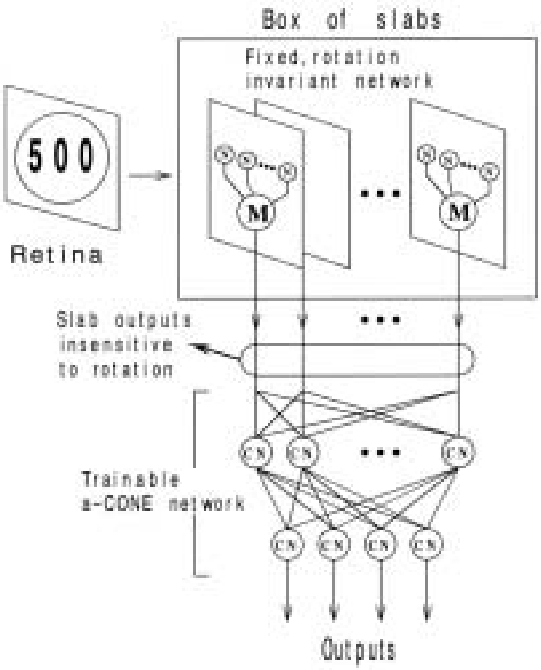
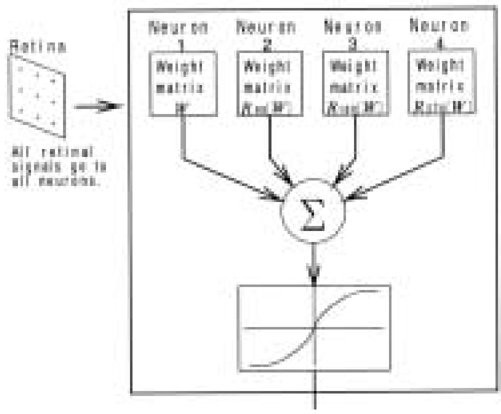


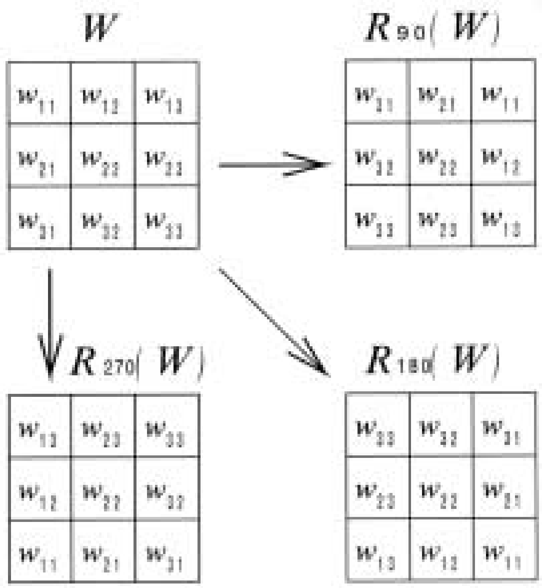
Fig.1 A rotation invariant neural pattern recognition

In implementing it, there are two ways in arrangement of neural weights of slabs, namely 2-dimensional grid and circular arrangements. The square array (grid arrangement) would be better to extend the conventional system [6]-[10] and to treat translation. However the circular arrangement of neural weights is better to consider only the rotational invariance and achieve high recognition accuracy. In this paper both arrangements are examined and compared by means of computer simulation.

The architecture of a preprocessor insensitive to rotation by 90 degree is briefly described on the basis of



(a) The structure to produce 90 degrees rotation-invariant output.



(b) Relationship between weight matrices

Fig.2 One slab with 90 degrees rotational invariance.

the reference [6]-[10]. Fig.2 (a) illustrates a slab structure achieving rotational invariance by 90 degree. Four neurons have weights with grid arrangement and the number of the weights is the same as the number of pixels on a retina. Pattern pixels on the retina are weighted by connection weights to compute the sum. It is passed through a nonlinear function to produce an output value of one of four neurons.

First, a weight matrix W of the neuron 1 is determined by random numbers. Next, the weight matrix W is rotated by 90 degrees, which is the weights $R_{90}(W)$ of the neuron 2. In the same way, 180 degrees rotated weights $R_{180}(W)$ and 270 degrees rotated weights $R_{270}(W)$ are formed, which are the weight matrices of the neurons 3 and 4, respectively. Relationship between the weights W and the others is illustrated in Fig2 (b). The figure shows the relation of 3 ×3 matrix case. They are the same except rotated by 90 to 270 degrees rotated.

A pattern on the retina is fed to every neuron of the slab. Each pixel of the pattern is weighted by a corresponding weight of a neuron unit in the slabs. The sum is the net input signal to the neuron and the neuron output is the output value of one of the neurons 1 to 4. Their outputs are weighted equally and are insensitive to every 90 degrees rotation because its rotation changes only the roles of the neurons 1 to 4. In the following, another rotation invariant preprocessor is described.

3.3 Square array of weights

Suppose that an input pattern has k×k pixels in a square array (grid) and slabs are constructed to be insensitive to rotation by every 30 degrees. Each neuron unit (labeled "N" in Fig.1) on slabs in the preprocessor has a k×k weight matrix. Fig.3 illustrates the right upper part (1/4) of a weight matrix of a neuron unit in k=12. In this case each weight is represented as a square. In Fig.3, $W_1(i_1,j_1)$ is the element of the *i*th row and *j*th column in the weight matrix W_1.

First, a 2-dimensional weight matrix W_1 with k×k is determined by random numbers. Next, the weight matrix $R_{30}(W_1)$ rotated by 30 degrees is determined. In Fig.3, the square bounded by a dotted line is the 30 degrees rotated $W_1(i_1,j_1)$. In this case, the centers of each square of W_1 and $R_{30}(W_1)$ do not have one-to-one correspondence. The conventional system cannot cause such a problem because it achieved only 90 degrees rotational invariance and the centers of each square of W_1 and $R_{30}(W_1)$ have one-to-one correspondence. In this paper such a point can be resolved as the following.

In making $R_{30}(W_1)$, not only the center of $W_1(i_1,j_1)$ but also many representative points, which are given in the square, are rotated by 30 degrees. In Fig.3, 25 representative points in the square of $W_1(i_1,j_1)$ are illustrated. Each representative point is regarded as having 1/25 of the $W_1(i_1,j_1)$ value. After rotation, the weight value of $W_1(i_1,j_1)$ is assigned to $R_{30}(W_1)$ (i_2,j_2) according to how many representative points of (i_1,j_1) are included in (i_2,j_2). Note that initial values of $R_{30}(W_1)$ are set as 0. For instance, Fig.3 shows that 19 representative points are included in the (i_2,j_2) element when 25 points are rotated by 30 degrees. In this case, 19/25 of the value of $W_1(i_1,j_1)$ is added to $R_{30}(W_1)$ (i_2,j_2). The other points of the (i_1,j_1) element are also assigned to their corresponding element according to the number of points included in each one after rotation. This procedure is performed for every weight of W_1. Weights which are not included in every element of $R_{30}(W_1)$ are ignored.

In the same way, $R_{60}(W_1)$, $R_{90}(W_1)$, ・・・, $R_{330}(W_1)$ are constructed in order. As a result, the following weight matrices are produced:

W_1 , $R_{30}(W_1)$, $R_{60}(W_1)$, ・ ・, $R_{330}(W_1)$.

Provided that these weights are assigned to neuron units "N" on slabs in Fig.1, slab outputs can be insensitive to every 30 degrees rotation of an input pattern. For the other

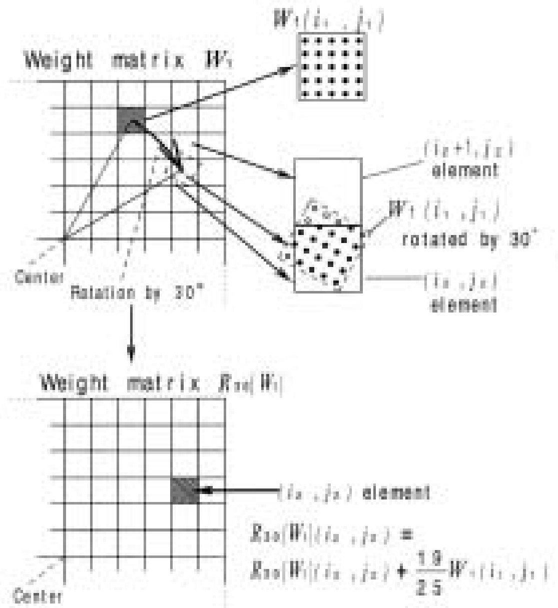


Fig.3 Rotation of weight matrix with square array of weights. Nineteen representative points of $W_1(i_1,j_1)$ are included in (i_1,j_1) element after rotation by 30 degrees. In this case, 19/25 times weight value of $W_1(i_1,j_1)$ is added to $R_{30}(W_1)$ (i_2,j_2).

slabs in the preprocessor rotation invariant outputs can be yielded by doing the same procedure using different weight sets W_i ,i=1,2, ・・・,*L*.

3.4 Circular array of weights

A pattern recognition system considered here is the same as the system in Fig.1 except the weight arrangement of the neurons on slabs in the preprocessor. Therefore the problem is how to determine weights of neuron units in order to achieve rotational invariance similar to the square array. In this section the shape of a retina and input images in a preprocessor are circular.

30 degrees rotational invariance is considered here. Fig.4 shows the weight matrix (Retina) of a neuron unit in the r-θ coordinate system. It is divided into 12 equal segments along circumference direction and into 4 along radius direction. However its center element is not divided as shown in Fig.4. First, a weight matrix W_1 is determined by random numbers. Next, its 30 degrees rotated weights $R_{30}(W_1)$ is yielded from W_1.
Similarly, $R_{60}(W_1)$, ・・・, $R_{330}(W_1)$ are computed in order. As a result, the following weight matrices are obtained:

W_1 , $R_{30}(W_1)$, $R_{60}(W_1)$, ・ ・, $R_{330}(W_1)$

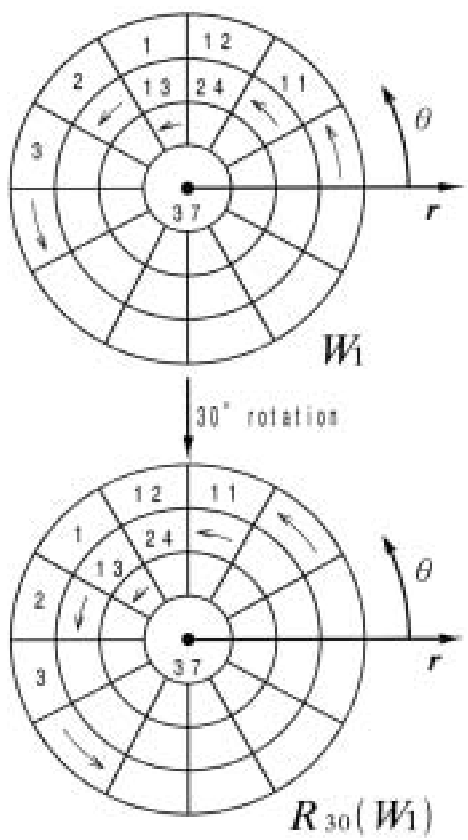


Fig.4 Rotation by 30 degrees for circular array of weights. The numbers denote the order of the weights.

Provided that these weights are assigned to neuron units "N", slab outputs can be insensitive to every 30 degrees rotation of an input pattern. For the other slabs in the preprocessor rotation invariant outputs can be produced by doing the same procedure using different weight sets W_i ,i=1,2, ・・・,*L*.

3.5 Trainable multi-layered network

A training layer has a multi-layered structure and performs a supervised training to classify signals from a preprocessing. Usually the back-propagation (BP) method is used in such a case. In this paper a multi-layered network with a-CONEs ("CN" in Fig.1) is utilized to achieve fast training.

4. Rotation invariant system with an edge detection structure

The system considered in this section is illustrated in Fig.5. It consists of a preprocessor (Feature Extraction Network) and a training layer (Trainable a-CONE Network). The preprocessor in this system is different from the former one [6]-[15], extracts edge features of an input pattern, and activates orientation specificity cells corresponding to edge directions. The activation pattern is an input signal to an input layer, which is composed of a set of orientation specificity cells. However this pattern is not insensitive to rotation. The rotational insensitivity is realized in the training layer.

4.1 Architecture of the preprocessing layer

The architecture of the preprocessor is illustrated in Fig..6. It consists of the edge detection layer and 3-dimensional structure of the orientation specificity cells. In Fig.6, Orientation specificity cells are illustrated as the OSC-layer on a plane. The square labeled "EN" indicates an edge detection network. Each EN handles a apart of an input pattern and computes edge intensity. in its area. Some edges with high intensity are detected and produced as outputs. In Fig.6, right and left direction and up and down direction on the edge detection- and the OSC-layers correspond to radial and circumference directions, respectively.

The orientation specificity cells are regularly arranged on the OSC-layer, which is a set of many cells. The OSC-layer consists of many planes, each of which includes cells selectively responding to an edge with the same direction. The number of cells on each plane in the OSC-layer is the same as the number of EN. Each cell on

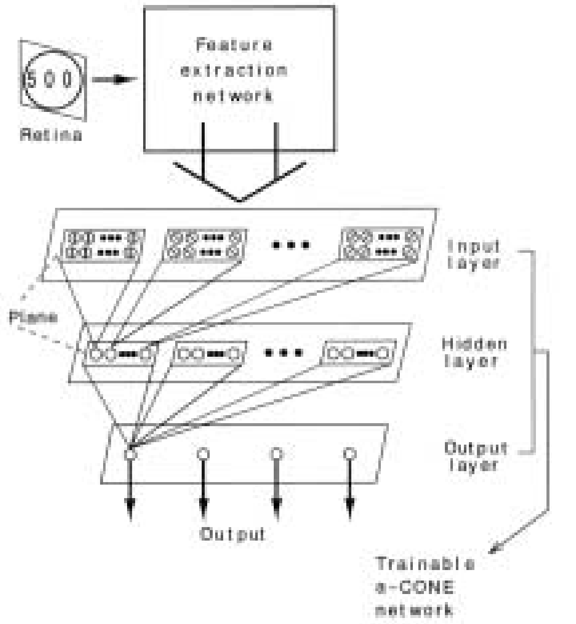


Fig.5 Rotation invariant neural network with an edge detection network.

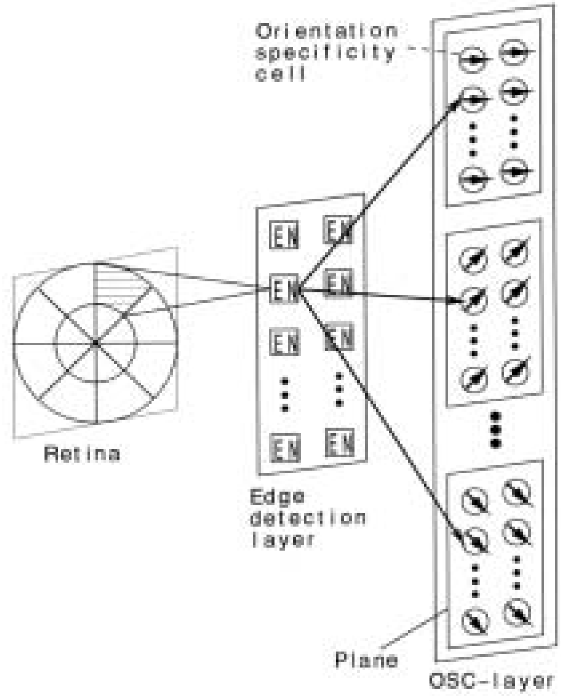


Fig.6 Feature extraction network. Each “EN” produce an edge direction and intensity.

the plane receives a signal only from corresponding EN. The number of planes depends on the number of divisions of a pattern on the retina along circumference direction. In Fig.6, the circumference is divided into eight and then the number of planes in the OSC-layer is eight. Selectively corresponding edge direction is different by 360/8=45° in neighboring planes.

Each EN yields edge directions with high intensity and activates cells at corresponding locations on planes nearest to their directions. In this paper the Sobel operator with size of 3x3 is used in edge detection but detail explanation omitted.

4.1 Structure of trainable network

An activity pattern on the OSC-layer is an input pattern to a trainable network. The trainable network is a multi-layered a-CONE network and the BP algorithm is used. In this case training with a binding condition is carried out by the BP algorithm with weight link similar to the reference [33][34].

When an input pattern is rotated, edge features to be activated are shifted on location to the circumference direction on a plane in the OSC-layer. Furthermore, edge direction of the input pattern is also rotated in this case. Therefore the plane of an orientation specificity cell to be activated varies corresponding to rotated degree.

The trainable layer has a three-layered structure and its hidden layer composed of a set of planes as shown in Fig.5. Each plane in the hidden layer includes sigmoid neuron units whose number is the same as the division number of a pattern on the retina. Each neuron on the plane possesses the same weights and a different receptive field. A different plane has different weights. Let weights on the ith plane $W_{i,k,m}$ m=1〜NTH, k=1〜NR, i=1〜NS, where NTH, NR are the number of divided elements along circumference and radial directions on the retina, respectively, and NS is the number of planes in the hidden layer. In this case each plane in the hidden layer includes NTH neurons. The subscript k and m are used to indicate weights to cells on each plane in the input layer. That is, the input layer has NTH planes, each of which possesses an activity pattern on orientation specificity cells with size of $NR \times NTH$ Let an activity pattern in the input layer be $S_{jj,k,mm}$ mm=1〜NTH, k=1〜NR, jj=1〜NTH, where jj indicates plane numbers.

The sum $Net(L)$ of NTH neurons on ith plane are given by the following algorithm:

For L=1 to NTH
For k=1 to NR
For m=1 to NTH
  $mm=m+(L-1)$
  if ($mm>NTH$) THEN $mm=mm-NTH$
  $Net(L)=Net(L)+W_{i,k,m}\times S_{L,k,mm}$

NEXT

where $W_{i,k,m}$ is weights connecting the first neuron on thr ith plane in the hidden layer to the first plane in the input blayer. Then the second neuron on the ith plane in the hidden layer connects the same weights $W_{i,k,m}$ to the second plane in the input layer and their weights are shifted one on the index m. Similarly, the third neuron connects $W_{i,k,m}$ to the second plane in the input layer and their weights are shifted two on the index m. This shift on m corresponds to rotation of edge features yielded by rotation of an input pattern. This system is different in this shift procedure from methods by the reference [33][34].

Network weights are trained by the BP-type algorithm. The quantity of weight change $\triangle W_{i,k,m}$ is then computed as average over all neurons on the ith plane in the hidden layer and is given by

$$\Delta\, W_{i,k,m} \sum_{L=1}^{NTH} (\Delta W_{i,i,k,m})_L \;/\; NTH.$$

As to weights connecting a hidden layer to an output layer, weights $W_{n,i}$ connecting neurons on the ith plane in the hidden layer to the nth output unit are restricted to be the same on the neurons. Therefore the number of weights in each output neuron is the same as that of planes in the hidden layer. In the same way as mentioned above the quantity of weight change is given by

$$\Delta\, W_{n,i} \sum_{L=1}^{NTH} (\Delta W^{i}_{n,i})_L \;/\; NTH.$$

By using the above mentioned training algorithm with restriction, rotational invariance can be achieved. After training, if a rotated training pattern is given to the retina, edge features yielded in the preprocessor vary according to its rotation angle. However role of neurons on each plane in the hidden layer are interchanged and those output responses are maintained to be the same..

5. Recognition by networks with feedback connections

5.1 Network with feedback connections

Recently, researches on architectures of multi-layered networks with feedback connections have been actively done [41]-[47]. This is very important not only in engineering applications but also in elucidation of information processing mechanisms in brains and nerve systems.

Elman [43] and Jordan [44] presented simple recurrent networks with feedback connections, in which outputs in hidden and output units are fed to its input layer. It is shown that they can be trained by the BP algorithm for various problems. These networks fix their feedback connection weights and their training algorithm is called the direct BP method. Furthermore fully-connected recurrent networks are also proposed. However they have disadvantages in learning time and memory capacity [45][46]. Therefore in order to reduce computational complexity necessary for pattern classification, a coin image recognition method using a simple network structure is considered, which can be trained by the direct BP method [48].

5.2 Coin image recognition method

The network structure utilized in this paper is illustrated in Fig.7. The Elman network copies output values in a hidden layer to a context layer and the values of the context layer are regarded as input signals at the next time index. Therefore the usual BP method can be applied directly [49][50].

First a coin image is changed into one in the circular coordinate system in order to learn and recognize it with a recurrent network. Next, the transformed image is fed to the network in turn and the network tries to predict the next image value. A preprocessing procedure to extract image features cannot be performed. Such image features are learned directly by the recurrent network.

First, circular one dimensional data are extracted from coin images, as shown in Fig.8.This data is regarded as a time series data and trained by the Elman network. Then recognition accuracy for such one dimensional data is not very good. Therefore as shown in Fig.8, its radial direction is divided into M segments and such two dimensional data are used for simulations. For simplicity it is described as one dimensional data in the following.

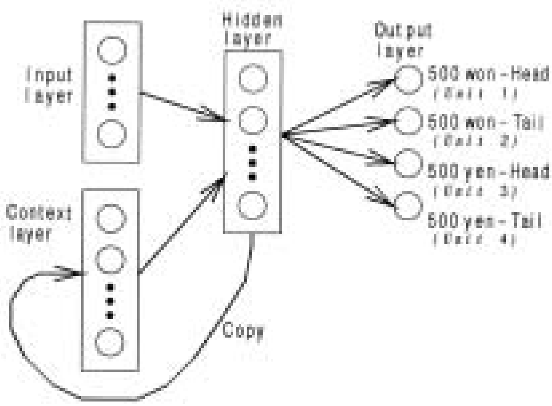


Fig.7 Simple recurrent (Elman) network.

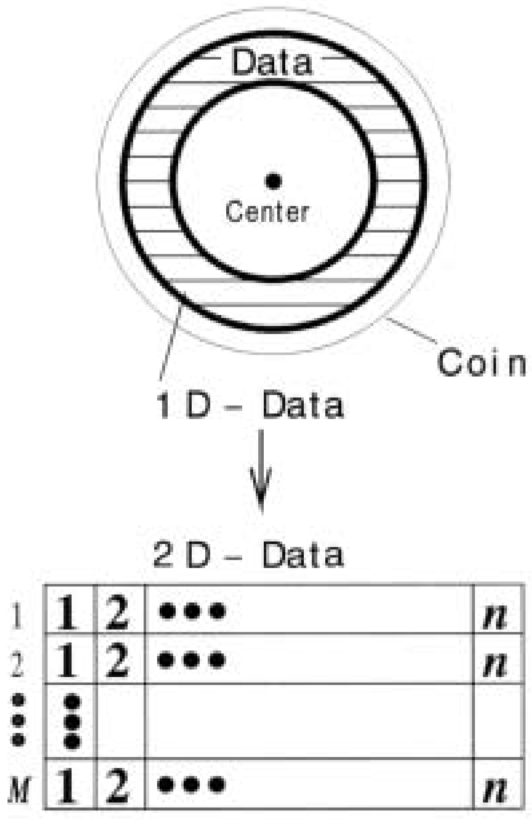


Fig.8 Simple recurrent (Elman) network.

Fig.7 illustrates a network structure for one dimensional data.

There are four patterns to be classified because an obverse and a reverse sides are recognized. A single output unit is necessary for each pattern. Therefore its network possesses an input and four output units in case of no time delay. In other words, an input signal at time t is fed to the network and a signal at time (t+1) is predicted. The signal at time (t+1) is its teacher signal. An output unit corresponding to each pattern learns only a single pattern independently. A hidden and a context layers are commonly trained. In training, the learning policy of an output unit to learn another coin except the present input pattern is given by:

(I) learn nothing.

(II) learn to produce bigger error

(III)learn zero or random numbers

This paper adopts the method (I) for simple learning and no learning is done for signals which have no relation to the output unit. For instance, a unit to learn the obverse side of 500 won coin is the output unit labeled “500 won-Head” and the other units cannot learn anything.

An initial value of coin data is random because randomly rotated coins are taken in. Therefore in order to reduce initial dependence the same coin data are given to the network several times and weight change is done for the second or later.

After training the network by training samples, test samples are fed to the network and the output error of each output unit is computed as

$$e_j = \sum_{m=1}^{NTH} (d_m - o_m)^2,$$

where d_m is a target signal, o_m is a real output value, NTH is a division number along the circumference, and j is the number of a output unit in Fig.7. Teacher signals in classification are common in four output units and are signals at the next time (t+1) if the present time is indicated by t. An output unit whose prediction error e_j is minimum on input data is a class to be recognized. It is given by

$$k = \arg\min_j e_j$$

where k indicates a class. If k=2, the result shows its class is the reverse side of 500 won coin.

6. Reduction of computational complexity of transform model

In this paper Fourier transform is adopted as a feature extraction method of images in Transform model as shown in Fig.9. When the discrete Fourier Transform is applied to circular coin images, first coin images are divided into a round shape and its circumference is divided into equal segments as the polar coordinate system. As coin images used in this paper are the same size, its radius is divided into equal length which is different from the complex-log mapping [25][26]. Next, circular coin images are changed into two-dimensional images in the Cartesian coordinate system. Then the discrete Fourier transform is applied to these images. In this case, as well known, amplitude spectra after transform are invariant to shift along each axis in Cartesian coordinate system. Therefore each amplitude spectrum is invariant to rotation of coins.

However the above mentioned method, which uses two dimensional Fourier transform as a preprocessor and its amplitude spectra are input signals to a network, the number of input units becomes huge. For instance, suppose that the number of divisions along its radius and circumference direction is 32. In this case the number of

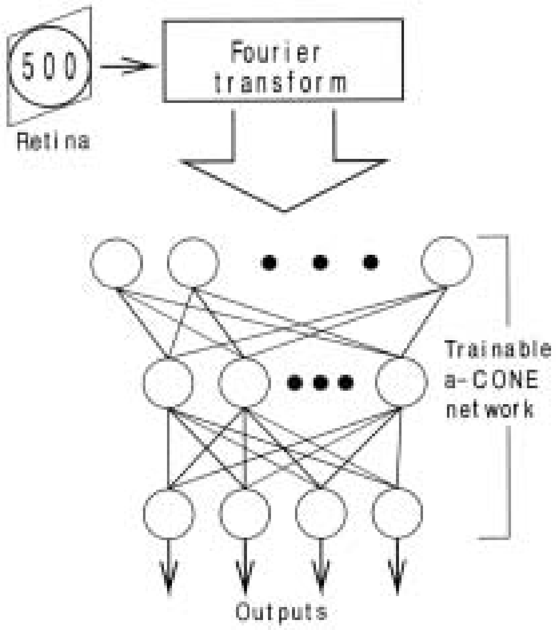


Fig.9 Neural network with Fourier transform followed by a three-layered neteork.

amplitude spectra yielded by the discrete Fourier transform is 32×32 =1,024. If the number of hidden units is 50, the total number of connecting weights is 51,400 and an issue on hardware implementation is caused.

In this paper we consider to decrease the number of connection weights in order to reduce computational complexity. For this purpose a structural learning method with forgetting of weight [51] is introduced in network learning. This learning method uses as a criterion function the following, in which an output error function *J* in the BP algorithm and sum of absolute values of connection weights are added:

$$J_f = J + \varepsilon' \sum_{i,j} | w_{ij} |$$
$$= \sum_i (d_i - o_i)^2 + \varepsilon' \sum_{i,j} | w_{ij} |$$

where o_i is an output value of the *i*th unit and d_i is its teacher signal. Then the quantity of weight change $\triangle w_{ij}'$ is given as

$$\Delta w_{ij}' = -\eta \frac{\partial J_f}{\partial w_{ij}} = \Delta w_{ij} - \varepsilon \mathrm{sgn}(w_{ij})$$

where $\triangle w_{ij}$ is the quantity by the BP learning and ε is a forgetting factor at every iteration. The above equation means a weight value is decreased by a constant ε per iteration cycle. Ishilawa showed that this learning method could find regularities included in training data and an obvious network structure emerged [51]. In this paper, we try to reduce a network size by using this structural learning method.

7. Applications to coin recognition

7.1 Teacher pattern in training

In classification of 500 yen and 500 won coins are classified and also its reverse and obverse sides of each coin are recognized. Therefore the number of output units in trainable network for the method (A)~(D) mentioned in the section *2* is four. Teacher signals are given as

500 won, obverse side: (1,0,0,0)
500 won, reverse side (0,1,0,0)
500 yen ,obverse side (0,0,1,0)
500 yen, reverse side (0,0,0,1).

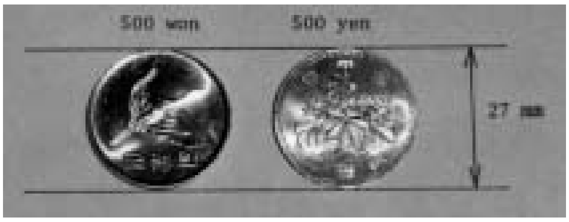
In this case, the number of pattern used for training is one, then it is linearly separable and a network without its hidden layer is desirable. However there is variance in gray scale level in every kind and many training samples are used for four kind. A three-layered network with a hidden layer is used for learning. The method (E) has four output units and a teacher signal at time *t* is an input signal at time (*t*+1) of time series data used for learning. The supervised learning BP method is used and the neuron model a-CONE (See appendix).

7.2 Normalization of coin images

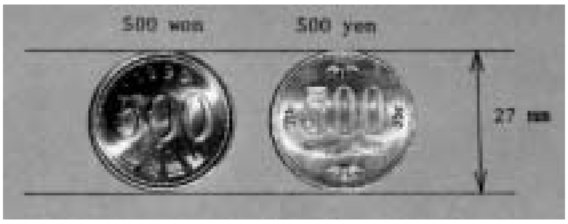
As shown in Fig.10, the diameter of coin size is 27mm. The 32mm square including coins is taken in as 256×256 dots and 256 gray scale levels. A preprocessing procedure extracts a coin part from this image and detects its coin center.. Coin extraction by using neural networks could be done [53] but omitted in this paper.

The extracted coin consists of 200 × 200 dots (radius=100 dots). This is transformed an image suitable for each system. For the slab model with square array of weights and the edge detection model, coin images are transformed as follows. Coin image data are normalized as 0 to 1 value by averaging 5×5 dots and yielding a pixel with the value of the average. Such a pattern is an input pattern to the retina. This procedure is necessary to reduce computational complexity and to detect precise edge features. As a result, input patterns consists of 40×40 pixels.

Next, input patterns for the slab model with circular array of weights are required to be transformed into circular images. Therefore the shape of its retina is circular. Suppose that the retina is divided into 4 and 12 along its radius and circumference, respectively, in the same way as



(a) Obverse model



(b) Reverse side

Fig.10 Coins with diameter of 27 millimeter.

that of Fig.4. In this case the number of segment pixels is 37 because the center is not divided along its circumference direction. Each neuron on a slab in the preprocessor also has 37 weights. Coin images are regarded as the same size as the retina and divided similarly. In this procedure, all pixel values included in each segment pixel which is the same shape as the circular array shown in Fig.4 are summed in gray scale and its sum is a gray scale value in each segment pixel. 37 circular segment pixels are produced in this case.

Next, input images for the feedback model are transformed into time series data as mentioned in section 5. In the transform model they are transformed into square array of data in the Cartesian coordinate system.

The above mentioned patterns are produced whose number is 50. A part of their samples is used for training and the rest is used as samples for test of recognition accuracy. Note that every image is taken in at random rotation and their gray scale values are also different.

7.3 Coin recognition results and considerations

(1) Transform model

The case using 1,024 Fourier spectra was examined. The number of input units is 1,024 and the number of hidden units is 25. Fig.11 shows recognition accuracy after error convergence in the BP learning. If the number of training samples is 10, network size achieving 100 % recognition accuracy is described in Table 1. From Table 1, it is shown that 100 % accuracy can be carried out only by using 8 % of weights and 72 % of input signals. This fact can contribute to reduction of computational complexity and simplify hardware implementation. Note that if weights less than 0.6 in absolute value are set as 0, recognition accuracy gradually decreases.

A random search method was utilized to do the same structure learning [54]. As a result, recognition accuracy is the same as that of 10 training samples case. However the number of weights capable of being eliminated is 1/3 of that of the BP case and variance of weights is large. The random search method is poor in generalization ability compared with the BP. The combination of a gradient method and a random search method would be effective as a learning method for complicated problems.

(2) Slab models

Results obtained using the slab model with the square array of weights are shown in Fig.12. Results obtained using one with the circular array of weights are shown in Fig.13. From these results, it is shown the system with circular array of weights can achieve 100 % accuracy using smaller number of training samples. It would be due to bad extensible performance in the system with square array as it is a system which can be extended to be invariant to translation, rotation, and scale. Both systems could not 100 % recognition accuracy only by using a single training sample because of variable gray scale values in coin images. The improvement on recognition accuracy has a bound even if the number of slabs in the preprocessor and as a result the number of input signals is increased. In these simulations, the number of slabs is 15 to 25 and the number of segments along circumference

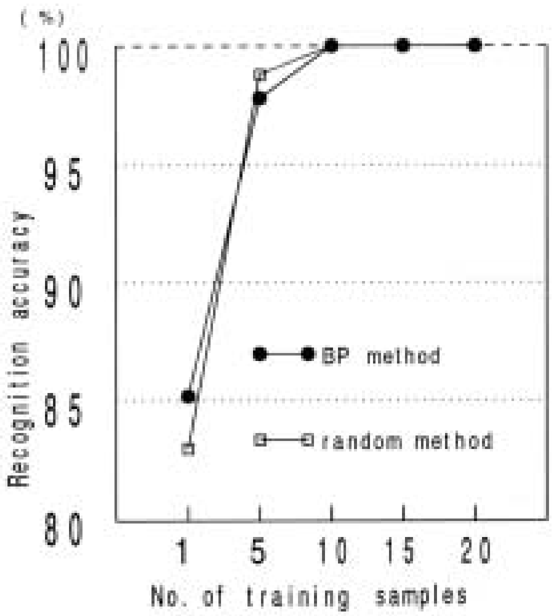


Fig.11 Recognition results obtained using varying number of training samples for training with forgetting.

Table 1 Comparison of the number of weights and input signals.

Network size after training(ε =10^{-6})			
Method	original	Random method	BP method
No. of input signals	1,024	1,024	738
No. of weights	25,700	17,793	2,076
Value of weights eliminated	0	$\|w_{ij}\| \leqq 0.2$	$\|w_{ij}\| \leqq 0.6$

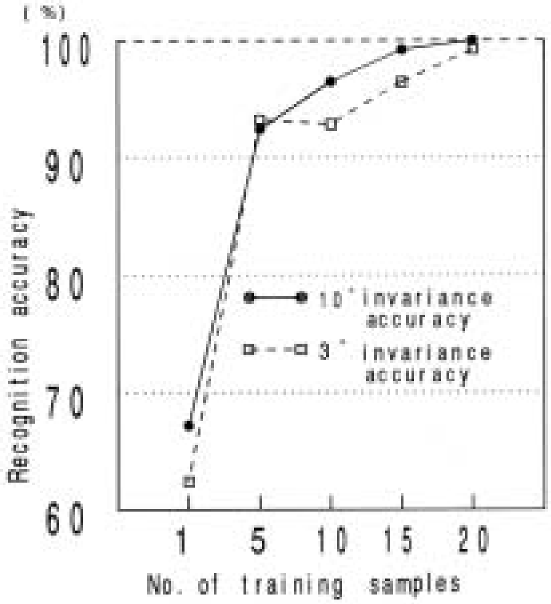


Fig.12 Recognition results obtained using varying number of training samples with square array of weights in the slab model. The number of slabs is 15.

direction is 36 to 72 were appropriate.

(3) Edge detection model

Figure 14 shows relationship between the number of training samples and recognition accuracy and rotational invariance by every 20 ° (The circumference is divided into 18). Comparison result obtained using varying number (1 to 3) of edge features on a edge detection network in the preprocessor is illustrated. When the number of edge increases its result becomes better. Four more edge features could not improve accuracy. The suitable number of planes in the hidden layer is 12 to 16 to yield good results.

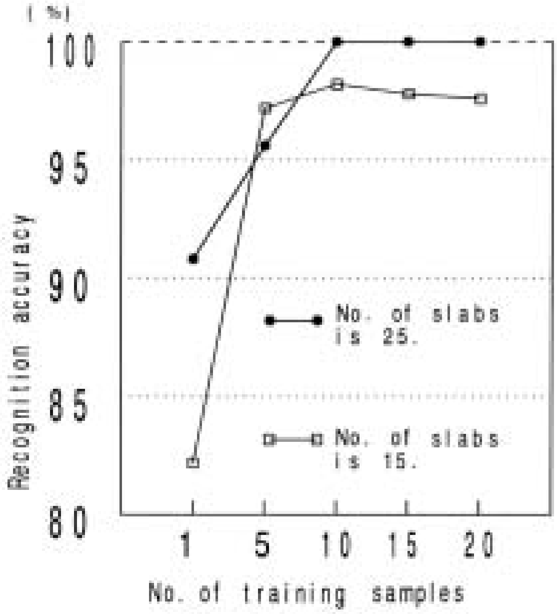


Fig.13 Recognition results obtained using varying number of training samples with circular array of weights in the slab model.. The number of neuron units is 72.

As shown in the figure, one sample training using a single one per class is only to perform 84 % accuracy and could not achieve 100 % accuracy. This is because the system could not learn delicate difference in edge feature between training and unknown samples. Such a fact would be due to variation in coin stain and in edge feature depending on a light condition for rotated coins on an coin classifying machine.

(4) BP model

For reference, the method in [34] is extended to be invariant to various rotation angles and is applied to the coin images used in this paper. This corresponds to the method (D) described in section 2. As a result of computer simulations, it is difficult the system diminishes training error using more than two samples per class. This fact shows that the BP algorithm could not converge the error for a simple feed-forward network by using the coin images, in order to achieve rotational invariance. Recognition accuracy obtained using one sample per class.

(5)Feedback model

Finally, simulation results for the feedback model are described. There are results obtained using M=4, described in section 5.2. We examined two cases: one is a case of 15×M input signals with time delay units and the other is a case of M inputs without time delay. The number

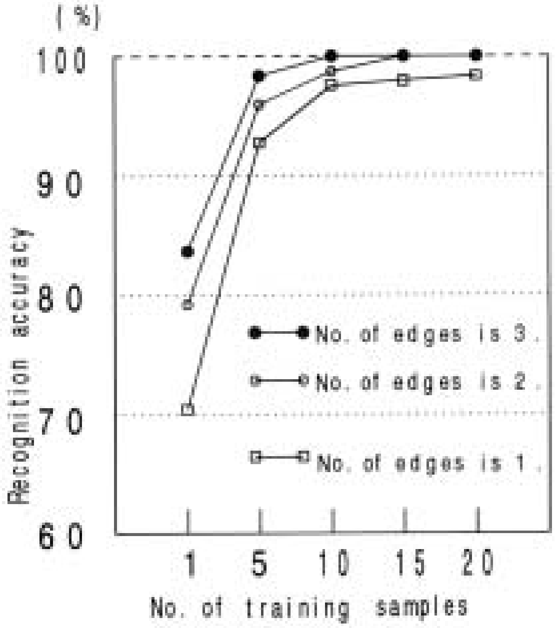


Fig.14 Recognition results obtained using varying number of training samples with the edge detection model. *NS* and *NTH* are set as 16 and 18, respectively.

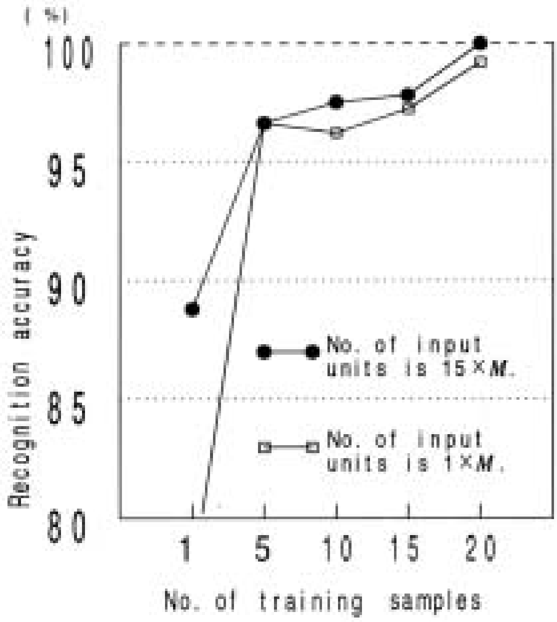


Fig.15 Recognition results obtained using varying number of training samples with the feedback model. The number of hidden units is 16.

of hidden and context layers units is 16. The number of output units is 4×M. The number of segments along circumference in coins is 36. As shown in Fig.15, when the number of training samples is 20 per class, 100 % accuracy is achieved in case with time delay terms.

For reference, we examined a feed-forward network without feedback term. When the number of input and hidden units are 25×M and 50, respectively, about 95 % accuracy carried out in 20 training samples.

(6)Considerations

As mentioned above, it is shown that each system is effective for engineering applications. Table 2 compares rough computational complexity of the systems. However it does not include the procedures of image transformation described in section 7.2. The methods (A) and (E) have advantages in computational complexity and memory capacity. Note that the method with the minimum variance of outputs is (A) using Fourier transform as a preprocessor. Therefore this method would be most suitable for hardware implementation. The edge detection model is a system considering to emulate brains. It is relatively good for coin recognition but its computational algorithm is complicated compared with (A). This fact is not very suitable for hardware implementation to real systems. In hardware implementation, it is important to reduce computational complexity and to decrease the number of

Table 2 Comparison of the number of multiplications

Multiplications H: Hundreds, T: Tens of thousands, M: Million		
Method	Preprocessor	Trainable network
A	T	T
B	M	H
C	T	T
D	*	M
E	*	T

input connections to neuron units in order to make small-sized systems. This would be a key to achieve hardware implementation of the transform model.

Problems awaiting to be solved are summarized in the viewpoint of modeling of brains as follows:

- For the edge detection model, a system considering not only edge directions but more complicated features.
- For the slab and the edge detection models, the extension to generalized systems capable of treating translation and scale.
- Evolution of network structure and design of advanced system using genetic algorithms.

Then knowledge in psychology and physiology [35]-[38] should be considered. From the viewpoint of applications to real machines, the following points are given:

- Verification of effectiveness in real hardware implementation using the transform model
- For the transform and the feedback models, design of recognition methods whose computational complexity is low and accuracy is high, and of reject procedures in case that coins could not be recognized.

8. Conclusions

This paper designs rotation invariant neural pattern recognition systems with application to coin recognition. In this paper, several systems are considered and their effectiveness is compared by means of computer simulations. As a result of such considerations, a Fourier transform model is suitable for hardware implementation in the viewpoint of computational complexity. From the viewpoint of elucidation of brain functions, an edge detection and a slab models should be improved to consider knowledge in psychology and physiology and to extend them to more generalized systems.

References

[1] K.Fukushima, S.Miyake, and T.Ito: Numeral Character Recognition by the Algorithm of the Neocognitron, IEICE Trans. Vol.J58-D, No.2, pp.206-213 (1983) in Japanese

[2] C.H.Ting: Rotation Invariant Neocognitron, Proc. Of Intl. Joint Conf. On Neural Networks, 3, pp.2216-2221 (1991)

[3] T.Nagano: Neural network to extract scale invariant features, Computrol, Vol.29, pp.26-31 (1990) in Japanese

[4] B.Widrow, R.Winter, and R.Baxter: Layered Neural Nets for Pattern Recognition, IEEE Trans. Acoust., Speech, and Signal Process., ASSP-36-7, pp.1109-1118 (1988)

[5] M.W.Koch, M.W.Roberts and S.W.Aiken: A Vision Architecture for Scale, Translation, and Rotation Invariance, Proc. Of Intl. Joint Conf. on Neural Networks, II, pp.393-396 (1990).

[6] M.Hosokawa, S.Omatu, and M.Fukumi: A New Approach for Pattern Recognition by Neural Networks with Scramblers, Proc. of Intl. Joint Conf. On Neural Networks, I, pp.183-188 (1989)

[7] M.Fukumi and S.Omatu: A New Back-Propagation Algorithm with Coupled Neuron and Its Application to Pattern Recognition, Proc. of the 28th SICE Annual Conf., pp.1327-1330 (1989)

[8] M.FUkumi, M.Hosokawa, and S.Omatu: Pattern Recognition System Insensitive to Translation and Rotation by Neural Network, Trans. IEE of Japan, Vol.110-C, No.3, pp.148-155 (1990)

[9] M.Fukumi, S.Omatu, and M.Teranishi: A New Neuron Model "CONE" with Fast Convergence Rate and Its Application to Pattern Recognition, IEICE Trans., Vol.J73-D-II, No.4, p.648-653 (1990) in Japanese

[10] M.Fukumi, S.Omatu, and M.Teranishi: A New Neuron Model "CONE" with Fast Convergence Rate with Application to Pattern Recognition, Proc. of the First Autonomous Decentralized System symp., pp.107-110 (1990) in Japanese

[11] M.Fukumi and S.Omatu: Rotation Invariant Pattern Recognition System by Using Neural Networks, Proc. of the 13th Intelligent System Symp., pp.53-56 (1991) in Japanese

[12] M.Fukumi, S.Omatu, F.Takeda, T.Kosaka: Neural Pattern Recognition System Invariant of Input Pattern and Its Application to Coin Recognition, ISCIE Trans., Vol.5, No.1, pp.9-17 (1992) in Japanese

[13] M.Fukumi, S.Omatu, F.Takeda, T.Kosaka: Rotation Invariant Pattern Recognition System Using Neural Networks, Proc. of System, Intelligent, and Neural Network Symp., pp.139-144 (1991) in Japanese

[14] M.Fukumi, S.Omatu, F.Takeda, T.Kosaka: Rotation Invariant Pattern Recognition System Using a Neural Network, Proc. of the 31th SICE Annual Conf., pp. 625-626 (1991) in Japanese

[15] M.Fukumi, S.Omatu, F.Takeda, and T.Kosaka: Rotation-Invariant Neural Pattern Recognition System with Application to Coin Recognition, IEEE Trans. On Neural Networks, Vol.3, No.2, pp.272-279 (1992)

[16] M.B.Reid, L.Spirkovska and E.Ochoa: Rapid Training of Higher-Order Neural Networks for Invariant Pattern Recognition, Proc. of Intl. Joint Conf. On Neural Networks, Vol.1, pp.689-692 (1989)

[17] K.Imai, K.Gouhara, and Y.Uchikawa: Architecture of Pattern Recognition System By Three Layered BP Model, IEICE Tech. Note, Vol.NC89-37, pp.27-32 (1989) in Japanese

[18] S.Lee and Y.Choi: Robust Recognition of Handwritten Numerals Based on Dual Cooperative Networks, Proc. of Intl. Joint Conf. On Neural Networks, Vol.III, pp.760-767 (1992)

[19] G.A.Carpenter and S. Grossberg:: The ART of

Adaptive Pattern Recognition by a Self-Organizing Neural Network, IEEE Computer, Vol.21, No.3, pp77-88 (1988)

[20] V.Cruz, G.Cristobal, T.Michaux and S.Barquin: Invariant Image Recognition Using a Multi Network

[21]K.H.Hu: Visual Pattern Recognition by Moment Invariants, IRE Trans. On Inform. Theory, Vol.IT-8, pp.179-187 (1962)

[22] C.The and R.T.Chin: On Image Analysis by the Methods of Moments, IEEE Trans. On Pattern Anal. And Machine Intell., Vol.PAMI-10, No.4, 00.496-513 (1988)

[23] A. Khotanzad and J.Lu: Classification of Invariant Image Representation Using a Neural Network, IEEE Trans. on Acoust. Speech and Signal Process., Vol. ASSP-38, No.6, pp.1028-1038 (1990)

[24] A.E.Grace and M.Spann: A Comparison between Fourierr-Mellin Descriptors and Moment Based Features for Invariant Object Recognition Using Neural Networks, Pattern Recognition Letters, Vol.12, pp.635-643 (1991)

[25] H.Wechsler and G.L.Zimmerman: 2-D Invariant Object Recognition Using Distributed Associative Memory, IEEE Trans. on Pattern Annal. and Machine Intell., Vol.PAMI-10, No.6, pp.811-821 (1988)

[26] Kageyu, N.Onishi, and N.Sugie: Character Recognition Invariant to Rotation and Scale, Tech. Note IEICE, Vol.NC90, pp.31-37 (1991) in Japanese

[27] D.Casasent and D.Psaltis: Position, Rotation, and Scale Invariant Optical Correlation, Applied Optics, Vol.15, No.7, pp. 1795-1799 (1976)

[28] D.G.Elliman and R.N.Banks: Shift Invariant Neural Net for Machine Vision, IEE Proc. Vol.137, Pt.I, pp.183-187 (1990)

[29] Foreign Coins in Toll Change Cause Mix-ups, Anger Drivers, The daily Yomiuri, June 30 (1992)

[30] M.Fukumi, S.Omatu, F.Takeda, T.Kosaka : Rotation Invariant Pattern Recognition System Using Neural Networks, Proc. of 15th Intelligent System Symp., pp.231-236 (1992) in Japanese

[31] M.Fukumi and S.Omatu: Rotation Invariant Neural Pattern Recognition System, Proc. of the 31th SICE Annual Conf., pp.301-302 (1992) in Japanese

[32] M.Fukumi, S.Omatu, F.Takeda, T.Kosaka: Rotation Invariant Neural Network with an Edge Detection Network, Trans. IEE of Japan, Vol.112-C, No.8, pp.457-464 (1992) in Japanese

[33] D.E.Rumelhart et al.: Parallel Distributed Processing , Vol.1, Cambridge, MA, MIT Press (1986) in Japanese

[34] S.Araya, Suzaki, H.Asoh: Neural Networks to train and Recognize Rotated Pattern, Trans. IEE of Japan, Vol.111-C, No.5, pp.202-208 (1991) in Japanese

[35] N.Tsukahara, Information Processing in Brains, Asakura-soten (1984) in Japanese

[36] Iriki, Sotoyam : Phisiology, Bunko-do (1986) in Japanese

[37] Yazawa: Newest Brain Science, Gakushu-Kenkyusha (1988) in Japanese

[38] Yazawa: Newest Theory of Brain machine, Gakushu-Kenkyu-sha (1992) in Japanese

[39] K.Fukushima: Neural Circuits and Self-Organization, Kyouritu-shuppan (1979) in Japanese

[40] M.Fukumi and S.Omatu: A New Back-Propagation Algorithm with Coupled Neuron, IEEE Trans. on Neural Networks, Vol.2, No.5, pp535-538 (1991)

[41] F.J.Pineda: Generalization of Back-Propagation to Recurrent Neural Networks, Phys. Rev. Lett., Vol.59, No. 19, pp.2229-2232 (1987)

[42] B.A.PearlMutter: Learning State Space Trajectories in Recurrent Neural Networks, NeuralComputation, Vol.1, No.2, pp.263-269 (1989)

[43]J.L.Elman: Finding Structure in Time, CRL Technical Report 8801, pp.1-31 (1988)

[44] M.I.Jordan: Supervised Learning and Systems with Excess Degrees of Freedom, COINS Technical Report, Vol.88, No.27, pp.1-41 (1988)

[45] R.J.Williams andD.Zipser: A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, Neural Computation, Vol.1, No.2, pp.270-280 (1989)

[46] K.Doya and S.Yoshizawa: Neural Network Model to learn Temporal Pattern, IEICE Trans., Vol.J73-D-II, No.8, pp.1150-1157 (1990) in Japanese

[47] K.Doya: Learning Algorithm for Recurrent Networks, J.of SICE, Vol.30, No.4, pp.296-301 (1991) in Japanese

[48] M.Fukumi, S.Omatu: Coin Image Recognition By Neural Networks, Proc. of Automatic Control Associ., pp.451-452 (1992) in Japanese

[49] A. Cleeremans, D.Servan-schreiber and J.L.McClelland: Finite State Automata and Simple Recurrent Networks, Neural Computation, Vol.1, pp.372-381 (1989)

[50] T.W.Karjala: Data Rectification Using Recurrent (Elman) Neural Networks, Proc. of Intl. Joint Conf. On Neural Networks, Vol.11, pp.901-906 (1992)

[51] M.Ishikawa: Structural Learning Algorithm of Connectionist Model Using Forgetting, Proc. of

Artificial Intelli., Vol.5, No.5, pp.595-603 (1990) in Japanese

[52] Y.Hayashi, M.Sakata and S.I.Gallant: Multi-Layer Versus Single-Layer Neural Networks and An Application to Reading Hand-s¥Stamped Characters, Proc. of Intl. Neural Network Conf., Vol.2, pp.781-784 (1990)

[53] M.Teranishi, M.Fukumi, S.Omatu: Pattern Recognition System by Neural Networks, Proc. of 34th ISCIE Annual Conf.,pp.105-106 (1990) in Japanese

[54]F.J.Solis, et al.: Minimization by Random Search Techniques, Mathematics of Operations Research, Vol.6, pp.19-30 (1981)

[55] S.Amari: Theory of Training and Classification, J of .ISCIE, Vol50, pp.1272-1279 (1967) in Japanese

[56] N.Baba: A Hybrid Algorithm for Finding the Global Minimum of Error Function of Neural Networks, Vol.1, pp.585-588 (1990)

Appendix

In this paper, the a-CONE (analog-Coupled Neuron) model shown in Fig.16 is used as a neuron model in a trainable multi-layered neural network. Neuron units in a training layer are a-CONEs. A learning method for a-CONE is called a CNR algorithm [9][10][15][40]. This learning algorithm is basically the same as that of the BP method, which is given as

$$\Delta W_{ij}(k) = \eta\delta_i x_j + \alpha\Delta W_{ij}(k-1) \qquad \text{(A1)}$$

where W_{ij} is the weight connecting the jth unit to ith unit, η is a step size, α is a momentum coefficient, and x_j is an input signal to the jth unit. δ_i is a error estimation term, which is given in the same way as BP in the following.

$$\delta_i = \begin{cases} (d_i - O_i)O_i' : & \text{Output units} \\ O_i' \sum_{m} W_{mi}\delta_m : & \text{Hidden units} \end{cases}$$

(A2)

where d_i is a teacher signal and O_i' is the derivative of the ith unit which is given as

$$O_i' = \eta_1' f_1' + \eta_2' f_2' \qquad \text{(A3)}$$

where f_i, i=1,2 correspond to a sigmoid function shown in Fig.16 and η_i, i=1,2 are set as 1. By using this (A3) as a derivative, a state with nearly 0 derivative can be escaped and fast learning is carried out. The detail explanations are referred to [9][10][11][15][40].

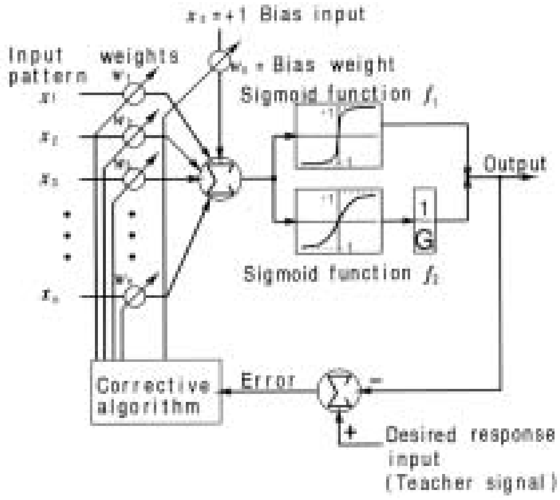


Fig.16 A neuron model "analog output neuron (a-CONE)"

Minoru Fukumi (Member)

Minoru Fukumi received the B.E. and M.E. degrees from University of Tokushima, in 1984 and 1987, and the doctor degree from Kyoto University, in 1996, respectively.. Since 1987, he joined Department of Information Science and Systems Engineering, University of Tokushima. Since 1996, he has been an Associate Professor at the Department of Information Science and Intelligent Systems, University of Tokushima. He received a Best Paper Award from the SICE in 1995. His research interests include neural networks, evolutionary algorithms, and Image processing,. He is a member of IEEE, IEEJ, ISCIE, and IEICE.

Sigeru Omatu (Member)

Sigeru Omatu received the B.E. degree in electrical engineering from University of Ehime, Japan, in 1969, and the M.E. and Ph.D. degrees in electronics engineering from University of Osaka Prefecture, Japan, in 1971 and 1974, respectively.

From 1974 to 1975 he was a Research Associate, from 1975 to 1980 a Lecturer, from 1980 to 1988 an Associate Professor, and since 1988 a Professor at University of Tokushima, Japan. Since 1995, he has been a Professor in

University of Osaka Prefecture.

Dr. Omatu Received a Best Paper Award from the Institute of Electrical Engineers of Japan in 1991 and the SICE in 1995. His interests center on neural networks and distributed parameter system theory.

Fumiaki Takeda (Member)

Fumiaki Takeda received the B.E. and M.E. degrees in mechanical engineering from the Nagoya Institute Technology, Japan, in 1982 and 1984, and the Ph.D. degree from university of Tokushima, in 1994, respectively. From 1974 to 1986, he was with Toyota Central Research and Development and in 1986 he joined Glory Ltd, Himeji, Japan. Since then, he has been engaged in research and development work on a recognition system for paper currency. Since 1998, he has been a Professor in Kochi University of Technology. His research interests include neural networks and expert systems as applied to paper currency recognition.

Toshihisa Kosaka (Member)

Toshihisa Kosaka received the B.E. degree in electronics engineers from Osaka Institute of Technology, Japan, in 1970 and the Ph.D. degree from University of Osaka prefecture, in 1999.

In 1070, he joined Glory Ltd. Himeji, Japan. Since then he has been engaged in research and development work on automatic machines for transactions involving paper currency and coins. Currently he is a senior engineer in the company. His research interests include neural networks and statistical analysis with applications to paper currency recognition.

Translated from J. of the SICE, Vol. 33, No.2, 151/165 (1994)