Recognition of paper currencies by hybrid neural network

Masahiro Tanaka
Okayama University

Fumiaki Takeda
Glory Corporation Limited

Kazuya Ohkouchi
Okayama University

Yasuyo Michiyuki
Okayama University

Recognition of Paper Currencies by Hybrid Neural Network

M. Tanaka, Dept. of Info. Tech., Okayama University, Okayama 700-8530, Japan
F. Takeda, Glory Co. Ltd., 1-3-1 Shimoteno, Himeji 670-8567, Japan
K. Ohkouchi and Y. Michiyuki, Okayama University, Okayama 700-8530, Japan

Abstract

For the recognition of paper currencies by image processsing, processing data by two steps can yield high performance. The two steps are so called "recognition" and "verification" steps. In the current recognition machine, a simple statistical test is used as the verification step, where univariate Gaussian distribution is employed. Here we propose using the probability density formed by a multivariable Gaussian function, where the input data space is transferred to a lower dimensional subspace. Due to the structure of this model, we call the total processing system as a hybrid neural network. Since the computation of the verification model is only to take inner product and square, the computational load is very small. In this paper, the method and the numerical experimental results are shown by using the real data and the recognition machine.

1. Introduction

It is an important task to classify the paper currencies at banks or large shops quickly and correctly. In [4], two images using high and low frequency filters were used as the input to the neural network for the classification of the paper currencies, but the classification success was less than 90%. One of the authors has utilized the multilayer perceptron (MLP) to classify the paper currencies and it has been used in the real time systems [1], [2], [3]. The successful classification system consists of two parts: recognition stage and verification stage. MLP has been used in the recognition stage. The key point of success is using the optimal mask to extract the feature vectors.

MLP is effective in this kind of classification of currency mainly because

- learning is very easy,
- computational load at the judgement is fairly small.

Moreover, the same system can be easily applied to various countries, where only the training of MLP of the currencies of the country is necessary. This is a plausible feature for a machine developer.

MLP works correctly among known classes because it is capable of making arbitrary shape of boundary surfaces. But it may yield false outputs if the unknown patterns are input to the system.

We could take several strategies to this problem. One way is to make many "dummy" patterns and train the MLP by setting all the outputs as zero. This method could work well for a very low dimensional system, but it is impossible for high dimensional spaces because the required dummy data amounts huge. To avoid the "curse of dimensionality", principle component analysis and/or the subspace recognition are widely used for pattern recognition [6], [7]. However, they are not directly related to the probability density function (PDF).

Takeda et al. [2] developed the verification stage by using a simple statistical approach. The method was to use the statistical test for each variable independently for every class. It has made a substantial success, but this verification was not sufficient enough because the the elements of the feature vectors are often heavily correlated each other. For a simple example, if the image is taken at a dark environment, the data point moves toward the origin with the phase of the vector fixed.

Frosini et al. [8] extensively studied the currency recognition problem by using MLP systems both in recognition and verification stages. The MLP in the recognition uses simple information extracted from the image, while the verification stage uses the pyramidal MLP as the autoassociator [9]. Their basic idea is that if the autoassociated pattern is similar to the input, it means that the data point was near the many training patterns and if the autoassociated pattern is much different from the input, it means that the data point is from the area where there were little training data.

Another method is to use the PDF. If the value of the PDF is small, this means that the sample is an unknown pattern. The PDF can be formed by several ways. It is possible to establish arbitrary shape of PDF by Parzen window. The drawback for Parzen window is that it requires storing all the training data in the memory. This is not appropriate for this kind of problems that require fast processing. A simple but useful model is to use the multivariable Gaussian PDF. This is not very powerful to the problems that need complicated

0-7803-4859-1/98 $10.00©1998 IEEE

boundaries because the boundary necessarily becomes the ellipsoid or the hyperplane. To overcome this drawback of Gaussian PDF, we propose a hybrid classification method where the MLP is used at the recoginition stage and the stochastic judgement is made at the verification stage. We call this "hybrid neural network" because of the structure. Previous work related to this problem is [2].

2. Description of images

2.1. Corruption of currencies

New currencies are almost the same except at the small characters of numbers. In this kind of image processing systems, we can ignore such a small amount of differences. The differences of the image arise due to the following reasons.

1. The paper currency may come to the imaging system rotated with a small angle.
2. The paper currency is itself degraded by the wrinkles or stains.

We express the degradation process as follows. Let the original uncorrupted paper currency be expressed as U_0. This is degraded by a mapping for stains Φ_1 as

$$\Phi_1 : U_0 \rightarrow U_1$$

The worn-out currency is also corrupted by another mapping for wrinkles Φ_2 as

$$\Phi_2 : U_1 \rightarrow U_2$$

U_2 is the currency that we usually observe in our daily life.

Due to such reasons, the images vary in the feature space, but basically the vectors are somewhere around the original image. Figure 1 shows the corruption steps of paper currencies.

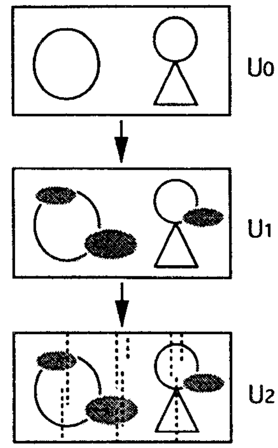

Figure 1. Corruption of Notes

2.2. Image processing system

Figure 2 shows the transition of data in the image processing system for paper currency classification. The currencies are transferred to imaging system, the image is first recorded as a gray scale image with 8 bits CCD camera with 45×128 mesh data. The supporting four edges are detected, and the inner small part of the image with center are taken for classification.

Now let us define the mathematical description of the images.

U_2 is transferred to the image capturing place under the camera, and it is recorded digitally. The recorded image is expressed here as D which is a two-dimensional digital data with 8 bits precision. D is too large in size to be used for pattern classification. Thus, D is aggregated into small number of blocks (6×8), and this block data is used for classification. The observed data D is obtained by the degradation through the rotation $\mathcal{R}$ and the imaging noise $\mathcal{N}$ as

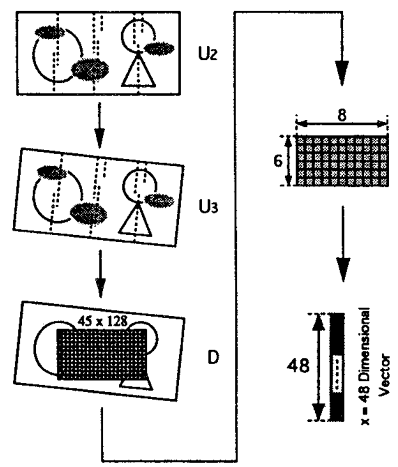

Figure 2. From Image to Data

$$\mathcal{R} : U_2 \rightarrow U_3$$

$$\mathcal{N} : U_3 \rightarrow D$$

The aggregated data $\boldsymbol{x}$ is given by

$$\mathcal{A} : D \rightarrow \boldsymbol{x}$$

3. Classification by hybrid neural network

3.1. Approaches for recognition of notes

A first approach for this problem should be the simple pattern matching after the calibration of the position and the rotation. Through various experiments, it has become clear that the simple pattern matching does not yield a good result because of the degradation of the currencies themselves. Thus more sophisticated pattern recognition methods have been considered to be necessary for this problem.

3.2. Recognition stage

MLP is used for this stage to find the candidates of the class.

Since the image data x has been considered too much redundant, the feature vector z was decided by using the mask $\mathcal{M}$ by

$$\mathcal{M} : x \rightarrow z$$

where the mask was optimized by the genetic algorithm. For each mask, the neural network was trained by using the training data.

Although this has made a big success, there still exists a problem for the unknown patterns. This problem will be discussed in the next subsection and a new method will be proposed.

3.3. Verification stage

The data of the same kind of money are considered to form a cluster. Figure 3 shows a concept of classification of two kinds of patterns.

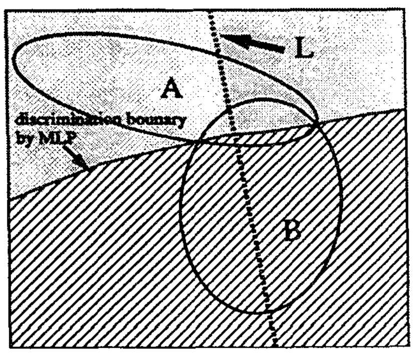

Figure 3. Classification of Two Patterns

Since the nonlinear property of the hidden unit is the sigmoid function, the output of the neural network should converge to 1 and 0 along the line L shown in the Figure 4.

If there are only instances from one of these classes, it works correctly. But it makes false output if the input pattern does not belong to either of these two classes. If the vector exists far in the lower directions, it is misjudged as the class B.

If there are q classes, it is natural to use q output units, and the neural network is trained by using an output vector $(0, 0, \cdots, 1, 0, \cdots, 0)$, where "1" appears at the k-th position if the training data is for the class

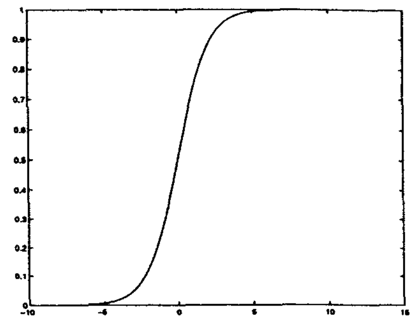

Figure 4. Output of MLP along the line L

k. We train the neural network in this way. The judgement whether the instance is from this class or not is to be made by using the stochastic classifier. Our strategy is to form a PDF for all the classes, and use it after the classification by the multilayer perceptron. The stochastic classifier also can be considered to be a kind of neural network because of its structure. Thus we call this total classification method Hybrid Neural Network.

The objective of stochastic classifier is to judge whether the test vector x can be accepted as a vector of the class i or it should be rejected. The Gaussian PDF of x for the class i is given by

$$p_i(x; m_i, M_i) = \frac{1}{(2\pi)^{(n/2)}|M|_i^{1/2}} \times \exp\left(-\frac{1}{2}(x - m_i)^T M_i^{-1}(x - m_i)\right) \quad (1)$$

Our problem is to estimate $m_i \in \mathbf{R}^n$ and $M_i \in \mathbf{R}^{n \times n}$.

Let the data of the class i be $x(1), \cdots, x(N)$, and suppose these are observed independently. Then the likelihood function $L(m_i, M_i)$ is given by

$$L(x(1), \cdots, x(N); m_i, M_i) = \prod_{k=1}^{N} p_i(x(k); m_i, M_i)$$

and the maximum likelihood estimates are given by

$$\hat{m}_i = \frac{1}{N}\sum_{k=1}^{N} x(k) \quad (2)$$

$$\hat{M}_i = \frac{1}{N}\sum_{k=1}^{N}(x(k) - \hat{m}_i)(x(k) - \hat{m}_i)^T \quad (3)$$

The judgement for x to be rejected to class i is made by

$$p_i(x) = \frac{1}{(2\pi)^{(n/2)}|\hat{M}_i|^{1/2}} \times \exp\left(-\frac{1}{2}(x - \hat{m}_i)^T \hat{M}_i^{-1}(x(i) - \hat{m}_i)\right) < \epsilon \quad (4)$$

where ϵ_i is a fixed threshold parameter, or equivalently

$$\exp\left(-\frac{1}{2}(\boldsymbol{x}-\hat{\boldsymbol{m}}_i)\hat{M}_i^{-1}(\boldsymbol{x}-\hat{\boldsymbol{m}}_i)\right) < S_i \tag{5}$$

Note that the judgement is made by the Mahalanobis distance between the center $\hat{\boldsymbol{m}}_i$ and the test vector $\boldsymbol{x}$ expressed as

$$(\boldsymbol{x}-\hat{\boldsymbol{m}}_i)\hat{M}_i^{-1}(\boldsymbol{x}-\hat{\boldsymbol{m}}_i) > -2\log S_i \tag{6}$$

Next we consider how to create a probability model from the observed data.

Although it looks enough to compute equation (6) for judgement, there are some problems related to that. The matrix $\hat{M}_i$ is not necessarily full rank and in this case, it is impossible to compute the inverse $\hat{M}_i^{-1}$ because of the low rank of $\hat{M}_i$. There are several reasons for this.

1. From the definition of $\hat{M}_i$, it is clear that the rank of $\hat{M}_i$ is less or equal to N, the number of samples. Hence, $\hat{M}_i$ is singular if the amount N of data is lower than the dimension n of the vector $\boldsymbol{x}$.
2. The mapping $\mathcal{R}$ basically increases the rank of the data little.

Furthermore, if some of the elements in $\boldsymbol{x}$ are formed as linear combinations of other elements, they don't contribute in increasing the rank. In the data used here, one element is the grand total of all the other elements. Thus the rank of $\hat{M}_i$ is never full.

Since $\hat{M}_i$ is a covariance matrix, it is positive semi-definite. Let $\lambda_1^{(i)} \geq \cdots \geq \lambda_n^{(i)} \geq 0$ are the eigenvalues of $\hat{M}_i$. Also, let the eigenvectors corresponding to those eigenvalues are $\boldsymbol{u}_1^{(i)}, \boldsymbol{u}_2^{(i)}, \cdots, \boldsymbol{u}_n^{(i)}$, where they are scaled to be unit vectors.

It is too much sensitive for noise if we consider based on the rank. Instead, it is much safer to treat the eigenvalues that are very close to zero as zero. Let C be the condition number that is used for this judgement. Then p_i is defined such as

$$\frac{\lambda_i(i)}{\lambda_1(i)} \geq \frac{1}{C}, \quad i = 1, \cdots, p_i$$

$$\frac{\lambda_i^{(i)}}{\lambda_1^{(i)}} < \frac{1}{C}, \quad i = p_i + 1, \cdots, n$$

is fulfilled. Thus

$$\hat{M}_i \boldsymbol{u}_j^{(i)} = \lambda_j^{(i)} \boldsymbol{u}_j^{(i)}, \quad j = 1, \cdots, p_i \tag{7}$$

$$\hat{M}_i U_i = U_i \Lambda_{p_i} \tag{8}$$

where

$$U_i = \{\boldsymbol{u}_1^{(i)}, \cdots, \boldsymbol{u}_p^{(i)}\}$$

$$\Lambda_p^{(i)} = \mathrm{diag}(\lambda_1^{(i)}, \lambda_2^{(i)}, \cdots \lambda_n^{(i)})$$

Since $U_i^T U_i = I$

$$U_i^T \hat{M}_i U_i = \Lambda_{p_i}^{(i)}$$

holds.

Here we define a p_i-dimensional vector $\boldsymbol{y}$ as

$$\boldsymbol{y} = U_i^T(\boldsymbol{x}-\hat{\boldsymbol{m}}_i) \tag{9}$$

Then obviously the PDF of $\boldsymbol{y}$ to class i comes to

$$p_i(\boldsymbol{y}) = \frac{1}{(2\pi)^{p_i/2}\sqrt{\lambda_1^{(i)}\cdots\lambda_{p_i}^{(i)}}} \exp\left(-\frac{1}{2}\boldsymbol{y}^T(\Lambda_{p_i}^{(i)})^{-1}\boldsymbol{y}\right) \tag{10}$$

$$\boldsymbol{y}^T(\Lambda_p^{(i)})^{-1}\boldsymbol{y} = \sum_{j=1}^{p_i}(\lambda_j^{(i)})^{-1}y_j^2 \tag{11}$$

This PDF can be used to judge for the membership of $\boldsymbol{x}$ to class i.

$$y_j = (\boldsymbol{u}_j^{(i)})^T(\boldsymbol{x}-\hat{\boldsymbol{m}}_i), \quad j = 1, \cdots, p_i \tag{12}$$

$$\sum_{j=1}^{p_i}(\lambda_j^{(i)})^{-1}y_j^2 \geq -2\log\tilde{S}_i \tag{13}$$

Note that this method has close relationship to the Principal Component Analysis (PCA). In the algorithm developed here, the rejection scheme is based on the position of the data mapped on the subspace.

First note that

$$(\lambda_j^{(i)})^{-1}y_j^2 \geq 0$$

We can see that the criterion

$$\sum_{j=1}^{p_i}(\lambda_j^{(i)})^{-1}y_j^2$$

consists of the same form of the quadratic term $(\lambda_j^{(i)})^{-1}y_j^2$. For a large j, $\lambda_j^{(i)}$ is very close to zero, i.e. $(\lambda_j^{(i)})^{-1}$ is very large. Thus, this term is very sensitive to the deviation in the direction of the component $\boldsymbol{u}_j^{(i)}$. This means that the vector that does not belong to this class i tends to gain a large value for this term. Thus, including the term with large j is useful for finding the vector of different class. However, this also gains a high value for a noise of this component direction. So, it is not always very effective to set C or p_i large for a very noisy system because it is too much sensitive for a vector of the same class that were not used in establishing the parameters $\hat{\boldsymbol{m}}_j$ and $\hat{M}_i$. Thus, it is rather effective to constrain the C at a relatively low value. This tuning should be made at the numerical experiments.

3.4. Algorithm

Set C before hand and execute the following step offline. Do the same for all the classes i.

1. Compute $\hat{\boldsymbol{m}}_i$, $\hat{M}_i$.
2. Compute the eigenvalues of $\hat{M}_i$ and let them $\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n \geq 0$. Also, compute the corresponding eigenvectors $\boldsymbol{u}_1^{(i)}, \boldsymbol{u}_2^{(i)}, \cdots, \boldsymbol{u}_n^{(i)}$.
3. Figure out p_i using the condition number C.
4. Let $l_1 = (\lambda_1^{(i)})^{-1}, l_2 = (\lambda_2^{(i)})^{-1}, \cdots, l_{p_i} = (\lambda_{p_i}^{(i)})^{-1}$.
5. Decide $\tilde{S}$ in order to maximize the classification performance and $\Gamma = -2\log\tilde{S}$.

Next is the online procedure. Suppose $\boldsymbol{x}$ is the test vector. Then

1. Compute
$$y_j = (\boldsymbol{u}_j^{(i)})^T(\boldsymbol{x} - \hat{\boldsymbol{m}}_i), \;\; j = 1, \cdots, p_i \tag{14}$$
2. Check whether the inequality
$$z = \sum_{j=1}^{p_i} l_j y_j^2 < \Gamma \tag{15}$$
holds. If (15) is not satisfied, this vector is considered not to belong to class i.

4. Currency recognition system

Here the hybrid neural network will be tested to the real currency. Figure 5 shows the recognition machine.

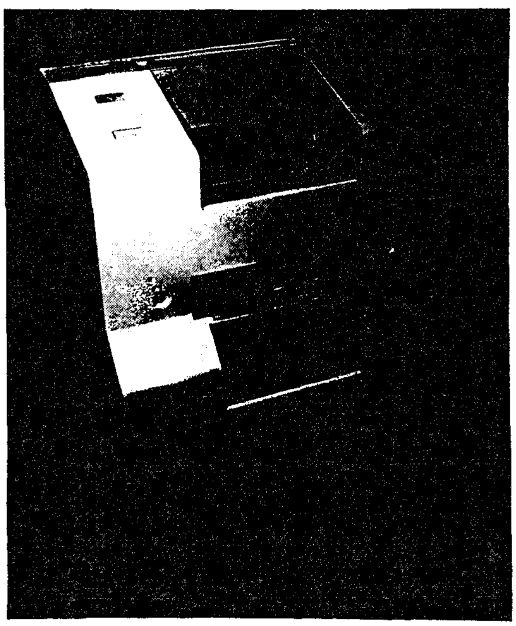

Figure 5. Recognition Machine

Currency image data resembles each other among different kinds. Especially among the currencies of the same country, the basic feature is often the same.

In experimental study here, the problem is to classify each of the Italian currencies and others.

4.1. Data description

We will deal with the 8 kinds (1000, 2000, 5000, 10000, (N)50000, (O)50000, (N)100000, (O)100000) of Italian Lire. (N) denotes the new type and (O) denotes the old type.

Considering the direction of a currency when tested by the machine, there are 4 directions (let A is the normal direction, B is 180-degree rotation of A, C is up surface down of B but doesn't move the location of shorter edge, D is 180-degree rotation of C). To make the problem simple, we treat these 4-directional images as different kinds of currencies.

For each kind, we prepared 80 instances of data, among which 60 instances were used to estimate the parameters, and the remaining 20 instances were used as the test data. As a test data set, we used various currencies from 16 countries including Italian currencies. The other kinds of currencies are listed as follows: Australia (100 Dollar), Belgium (5000 Franc), Canada (100 Dollar), Denmark (500 Krone), France (100 Franc), Germany (200 Mark), Korea (10000 Won), Netherlands (100 Gulden), Norway (500 Krone), Spain (10000 Pesetas), Sweden (500 Krone), Switzerland (100 Franc), Taiwan (1000 Dollar), UK (50 Pound) and US (100 Dollar).

4.2. Numerical results

In this numerical example, we mainly show the results for the stochastic classifier since the MLP has already been shown to yield very good performances for the known patterns [2].

The value of z of (15) for the Italian 5000 Lire (A direction) is plotted in Figure 6. The x axis denotes the value of p_i, and the y axis of the figure denotes the value of z. Since each term in (15), it increases as the new term is added. Solid lines denotes the result for the training data and dotted lines indicate the result for the test data of the same kind, and the chained line denotes the result for the data of foreign currencies.

From this graph, we observe as follows.

1. Value z of (15) takes smaller value for training data, and does not increase significantly as p_i increases. This is because the axes at minor components are those where the training data didn't include much information for these axes.
2. Test data takes larger value when we use over about 40 axes. It is very sensitive to noisy value that arises in minor components. On the other hand, the value at the middle range still keeps low value, which means the test data also has the similar density as the training data in the primary axes.

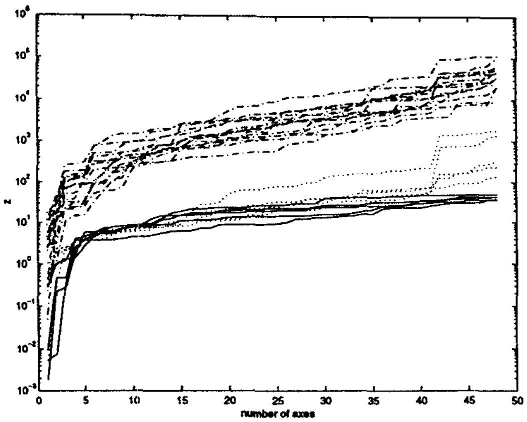

Figure 6. Values of z for (O) 5000 Lire (A direction)

3. For other currency data, z takes large values because they have different primary components.

We got stable results using $p_i \in [5, 40]$. We put $C = 10^3$, but it does not need to be so large. We selected the axes supposing the condition number $T = 10^3$. On this condition number, p, the dimension of y varies for different kind of bill, and the number was 22-23.

In Table 1, α plays an important role. α is the maximal value of z for the 20 test data of the same kind of currency. This means that all the instances gave $z \leq \alpha$ for each class. Also, β is the minimal value of z for other currencies computed with the same parameters of the class i. If $\alpha < \beta$, all the test data can be accepted or rejected correctly without the MLP. There was a very small amount of misclassification in the Table 1. These misclassification arose only between the new and the old currencies of the same price. Note that these currencies look like quite similar among the same prices.

Using the hybrid neural network, it was possible to classify these 4 currencies because they were classified into the correct classes in the recognition stage by the MLP.

5. Conclusions

The hybrid neural network has the advantages such as 1) complicated boundaries can be learned by the MLP, 2) unknown patterns can be distinguished from the known patterns. It may be possible to use multiple Gaussian functions for a class, but the learning will need some sophisticated technique for this, for example, EM algorithm [5]. However, for the data of paper currencies, Gaussian PDF model for each class seems sufficient for the verification.

Table 1
Classification result

species	α	β	notes
1000(A)	112	198	
1000(B)	89	229	
1000(C)	69	302	
1000(D)	135	506	
2000(A)	81	323	
2000(B)	147	1011	
2000(C)	209	449	
2000(D)	98	181	
5000(A)	95	447	
5000(B)	348	458	
5000(C)	94	453	
5000(D)	51	263	
10000(A)	56	395	
10000(B)	94	194	
10000(C)	130	348	
10000(D)	140	281	
(N)50000(A)	75	268	
(N)50000(B)	123	215	
(N)50000(C)	67	78	
(N)50000(D)	77	106	
(O)50000(A)	614	132	(N)50000(A)
(O)50000(B)	58	68	
(O)50000(C)	68	106	
(O)50000(D)	138	237	
(N)100000(A)	139	308	
(N)100000(B)	296	322	
(N)100000(C)	53	168	
(N)100000(D)	164	208	
(O)100000(A)	362	249	(N)100000(A)
(O)100000(B)	439	754	
(O)100000(C)	216	136	(N)100000(C)
(O)100000(D)	414	152	(N)100000(D)

References

[1] F. Takeda and S. Omatu. Bank note recognition system using neural network with random masks, *Proc. World Cong. Neural Net.*, Portland, OR, 1: 241-244, 1993.

[2] F. Takeda and S. Omatu. High speed paper currency recognition by neural networks, *IEEE Trans. Neural Networks*, 6(1): 73-77, 1995.

[3] F. Takeda, S. Omatu and S. Onami. Mask optimization by genetic algorithm for a neuro-pattern recognition machine with masks, *Trans. Inst. of Syst. Contr. and Inform. Engineers*, 8(5): 196-203, 1995.

[4] S. Pattaramalai, P. Pimsen and K. Chamnongthai. A Thai note recognition by using two frequency bands and neural network, *Proc. IASTED Int. Conf. Modelling, Simulation and Optimization*, 16-19, 1997.

[5] R. L. Streit and T. E. Luginbuhl. Maximum likelihood training of probabilistic neural networks, *IEEE Trans. Neural Network*, 5(5): 764-783, 1994.

[6] M. Prakash and M. Narasimha Murty. Growing subspace pattern recognition methods and their neural-network models, *IEEE Trans. Neural Network*, 8(1): 161-168, 1997.

[7] C.M. Bishop. *Neural Netowork for Pattern Recognition*, Clarendon Press,Oxford, 1995.

[8] A. Frosini, M. Gori and P. Priami. A neural network-based model for paper currency recognition and verification, *IEEE Trans. Neural Networks*,7(6): 1482-1490, 1996.

[9] M. Bianchini, P. Frasconi and M. Gori. Learning in multilayered networks used as autoassociators, *IEEE Trans. Neural Networks*, 6(2): 512-515, 1995.