Artif Life Robotics (2004) 8:133–139
DOI 10.1007/s10015-004-0300-1

ORIGINAL ARTICLE

Ali Ahmadi · Sigeru Omatu · Toshihisa Kosaka
Toru Fujinaka

A reliable method for classification of bank notes using artificial neural networks

Received: June 13, 2003 / Accepted: October 27, 2003

Abstract We present a method based on principal component analysis (PCA) for increasing the reliability of bank note recognition machines. The system is intended for classifying any kind of currency, but in this paper we examine only US dollars (six different bill types). The data was acquired through an advanced line sensor, and after preprocessing, the PCA algorithm was used to extract the main features of data and to reduce the data size. A linear vector quantization (LVQ) network was applied as the main classifier of the system. By defining a new method for validating the reliability, we evaluated the reliability of the system for 1200 test samples. The results show that the reliability is increased up to 95% when the number of PCA components as well as the number of LVQ codebook vectors are taken properly. In order to compare the results of classification, we also applied hidden Markov models (HMMs) as an alternative classifier.

Key words Bank note recognition · Reliability · PCA · LVQ · HMM

1 Introduction

Neural networks have been widely applied for recognition of bank notes in automatic teller machines (ATMs) in past years, and a variety of approaches have been performed to improve the classification rate and reliability of the system.[1,2] Due to high risk of misclassification in such systems, the reliability of recognition becomes of high importance. Generally, the system must be able to classify not only different classes of different worth bills, but also the bills with different levels of fatigue, taint, and defects. Moreover, the system is expected to be robust in classification of shifted and rotated bills, and also be crucially sensitive to counterfeit notes.

Concerning the complexity of the bill structure, which is used to prevent bill counterfeiting, one major step in such systems is optimizing the data acquisition and feature extraction of data. The extracted features must be discriminative enough such that they help the classifier to recognize the various kinds of input bills and also to reject counterfeit ones. Furthermore, because these machines generally work under real conditions with a wide variance in input data, a major problem is how to improve the reliability of the system to cover all kinds of real data. Thus, all proposed approaches are mainly intended to address two main problems: first, definition of a criterion for validating the reliability, and second, providing new algorithms for improving the reliability.

The system proposed in this article is based on using principal component analysis (PCA) for feature extraction of data and linear vector quantization (LVQ) as the main classifier. Also, a new algorithm for evaluating the reliability of the classification is proposed, which is based on the assumption of Gaussian densities for data assigned to the codebook vectors of the LVQ, and then determination of the overlap zones.

In order to have a comparative study on the results of classification, we have also applied hidden Markov models (HMMs) as an alternative classifier, which is proper for sequential input data, instead of the LVQ. Here, by using a line sensor for reading the bill data, a set of sequential frames is generated which will be used as the HMMs observation data.

The experimental results obtained from different kinds of US dollar bills show that using PCA for feature extraction and LVQ as the main classifier can increase the reliability of the system significantly. The proposed method is intended for classifying different kinds of paper currency, although we have only examined US dollar bills.

A. Ahmadi (✉) · S. Omatu · T. Fujinaka
Osaka Prefecture University, Department of Computer and Systems Sciences, 1-1 Gakuen-cho, Sakai 599-8531, Japan
Tel. +81-72-254-9279; Fax +81-72-257-1788
e-mail: ahmadi@sig.cs.osakafu-u.ac.jp

T. Kosaka
Glory Ltd., Himeji, Japan

This work was presented, in part, at the 8th International Symposium on Artificial Life and Robotics, Oita, Japan, January 24–26, 2003

2 Data acquisition and preprocessing

Two kinds of sensors are used for reading data: point sensors and line sensors. The data generated by point sensors are used in LVQ classifier, while the line sensor data are used in HMMs because of its sequential shape which come as a sort of frames.

2.1 Point sensors

Five point sensors are used for reading the bill. Each sensor uses two different wavelengths for generating two channels of data. Thus, there are a total of 10 channels with each of them containing 170 pixels (i.e., an 10×170 array). At first by using a linear function we generate a new channel of data based on the data of two channels for each sensor. Hence, 15 channels are obtained in total among which we select 6 main channels that represent the main characteristics of the data. These channels are the central channels with fewer similar pixels. In order to compress the 170 data pixels in each channel, we discard the first and last 10 pixels and take the average of every 6 neighbor pixels so that finally we have 25 pixels in each channel. Then a linear transformation is applied for normalization data as follows:

$$x_i = \frac{t_i - \bar{t}}{S_t} G + C \tag{1}$$

where t_i is the pixel value in each channel, $\bar{t}$ is the mean value of pixels, S_t is the standard deviation, and $G = 512$ and $C = 128$ are the coefficients of gain and offset, respectively, whose values are determined experimentally. Thus, a vector x of $6 \times 25 (= 150)$ elements is provided for the feature extraction step.

2.2 Line sensors

An advanced line sensor is used for reading the bill data in a width-wise fashion. The resolution of the sensor is around 4–10 pixels/mm, but after compression, there are 100 pixels in each line and a total of 50 lines for each US bill. It means that regarding the size of a US bill (156×66 mm), there is a resolution of 1.5 mm/line and 1.6 mm/pixel. Figure 1 shows a schematic of the line sensor reading process.

To make the data more effective, some additional features are extracted and added to the original pixels. For each frame, five main features are extracted based on its original pixels as follows:

1. Intensity, which is the percentage of black pixels within each frame
2. Number of white pixels
3. Mean value of pixels
4. Variance of pixels
5. Number of transition between black and white pixels

To identify a pixel as black or white, a threshold value for the gray level of a pixel is used. In order to reduce the size of pixels, we compress the original pixels of each frame to 25 pixels by averaging each 4 neighbor pixels.

Finally, by concatenating the five extracted features and 25 compressed pixels for each frame, we generate a data vector relating to each frame, and therefore, a total of 50 data vectors, each one containing 30 pixels, are generated as the observation data of the bill.

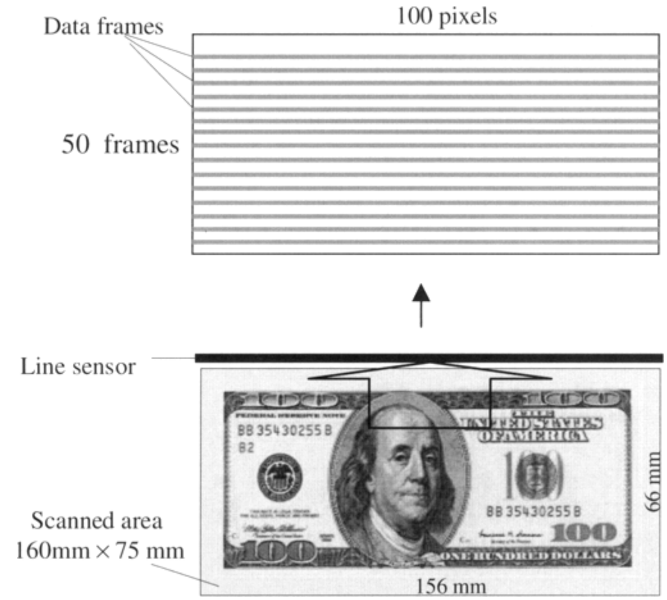


Fig. 1. The line sensor used to generate data pixels of the bill. For each bill, 50 data frames are generated

3 Principal component analysis feature extraction

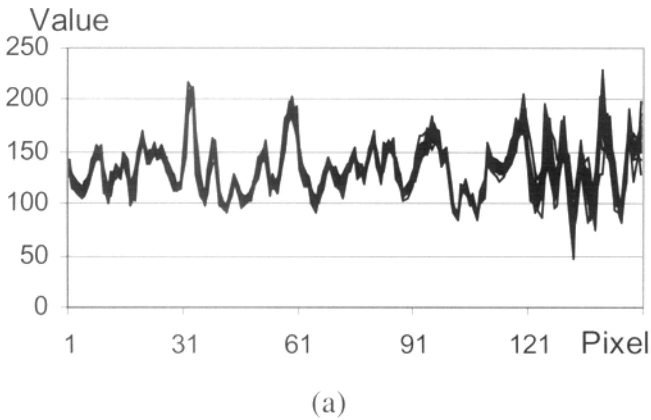
PCA is an essential technique for data compression and feature extraction, and it has been widely used in statistical data analysis and pattern recognition. The most common derivation of PCA is in terms of a standardized linear projection which maximizes the variance in the projected space.[3,4] For a set of observed d-dimensional data vectors $\{x_n\}, n \in \{1, \ldots, N\}$, the q principal axes $w_j j \in \{1, \ldots, q\}$, are those orthonormal axes onto which the retained variance under projection is maximal. It can be shown that the vectors w_j can be obtained by the q dominant eigenvectors corresponding to the covariance matrix of the data as follows:

$$\mathbf{S} w_j = \lambda_j w_j \tag{2}$$

where $\mathbf{S} = \Sigma_n (x_n - \bar{x})(x_n - \bar{x})^{\mathrm{T}}/N$ is the covariance matrix of data x_n, λ_j is the j-th eigenvalue of $\mathbf{S}$, and $\bar{x}$ is the data samples mean. Then $\mathbf{W} = (w_1, w_2, \ldots, w_q)$ is employed to transform the data vector x_n to a new space as

$$y_n = \mathbf{W}^{\mathrm{T}}(x_n - \bar{x}) \tag{3}$$

where y_n is a q-dimensional reduced representation of the observed vector x_n.



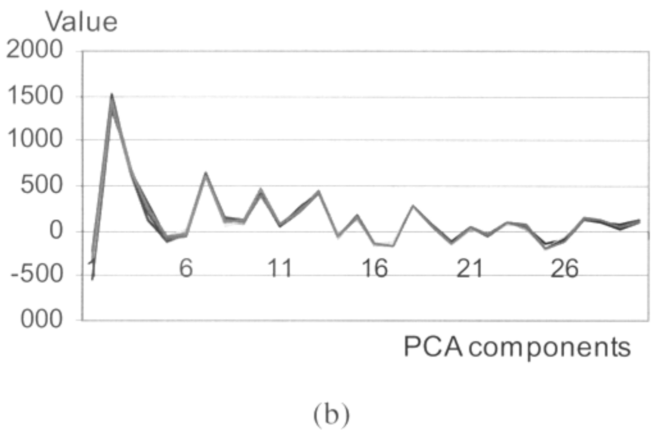


Fig. 2. a Distribution of data for US \$100 bill (taken from 30 sample bills). **b** Distribution of principal component analysis (PCA) components for the same data

The main property of the PCA is that only the projec-tion onto the principal subspace can minimize the squared reconstruction error $\Sigma \|x_n - \hat{x}_n\|^2$ where $\hat{x}_n$ is called optimal linear reconstruction of x_n and identified as: $\hat{x}_n = \mathbf{W}x_n + \bar{x}$.

Here, we take q as 30. It means by using PCA we extract 30 components of the input vectors. For some comparative experiments, we also applied other values for q between 10 and 50. Figure 2 shows the distribution of data for a \$100 bill before and after using PCA.

4 Classification

4.1 Learning vector quantization

Kohonen's learning vector quantization (LVQ) is a classification method based on competitive neural networks,[5,6] which allows us to define a group of categories on the space of input data by a supervised learning algorithm (Fig. 3). It basically models a discrimination function based on a set of labeled codebook vectors and then searches for the nearest neighborhood between codebooks and data. The training algorithm involves an iterative gradient update of the winner codebook. The direction of the gradient update depends on the correctness of the classification using a nearest-neighborhood rule in the Euclidean space. If a data sample is correctly classified (the labels of the winner unit and the data sample are the same), the codebook closest to the data sample is attracted toward the sample; if incorrectly classified, the data sample has a repulsive effect on the codebook. The update algorithm can be described as follows:

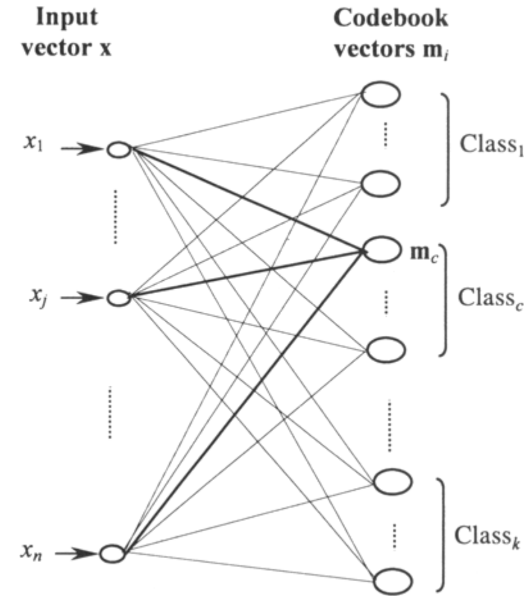


Fig. 3. Structure of learning vector quantization network

$$m_c(t+1) = m_c(t) \pm \alpha(t)[x(t) - m_c(t)] \tag{4}$$

where $x(t)$ is the input vector, $m_c(t)$ is the nearest codebook vector, and $\alpha(t) \in [0,1]$ is a learning rate which decreases monotonically with time. The sign ($\pm$) depends on whether the data sample is correctly classified (+) or misclassified (−). The training procedure is repeated iteratively until convergence. In the recognition phase, a data vector x_i is assigned to a class whose corresponding codebook has minimum distance to the vector.

Because the LVQ network is beneficial in classification of data with a large number of inputs and explanation of the misclassification, it is applied as the main classifier of the present system. As we consider six kinds of US bills including \$1, \$5, \$10, \$20, \$50, and \$100 and for each bill there exist four directions of rotation (Fig. 4), a total of 24 (6×4) output categories are considered for the classifier.

It should be noted that we consider four subclasses for each bill, because the data of four directions of each bill are quite different. However, in the reliability evaluation step we consider them as the same class.

The system is trained by taking a trial number of codebook vectors for each class looking for the best classification rate and maximum reliability. A total number of 200 codebook vectors (an average of 8 vectors per class) is found to be the best after the experiments. The number of iterations for each training epoch is taken to be 10000, while

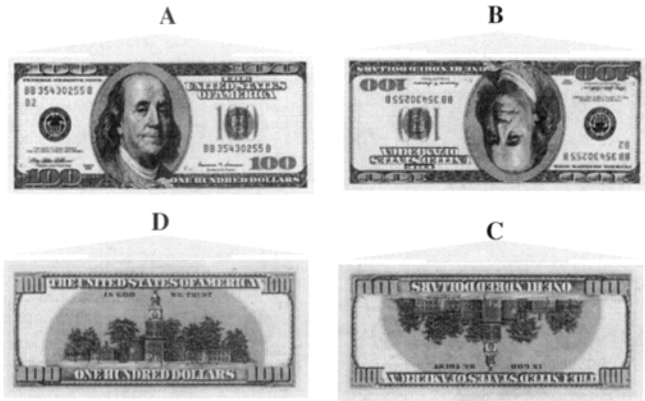


Fig. 4. Four different possible rotations for inserting a bill into the machine

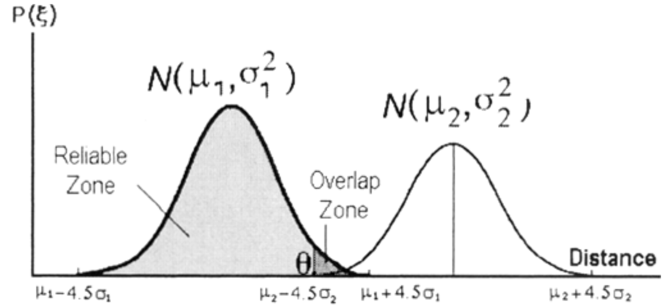


Fig. 5. The overlap between codebook densities of two nearby classes

a linear function $\alpha(t) = \alpha(0)\ (1.0 - t/T)$ is applied for learning where T is the number of iterations. Therefore, the LVQ classifier has 30 neurons (the number of extracted features) in the input layer and 200 neurons in the output.

4.2 Hidden Markov models

Considering the sequential nature of the data, we also apply HMMs as an alternative classifier to have comparative results.

HMM is a double stochastic process that can efficiently model the generation of sequential data[7] and is generally identified as a triple $\lambda = (\pi, A, B)$ where

$$\pi_i = P(q_1 = S_i) \tag{5}$$

$$a_{ij} = P(q_t = S_j | q_{t-1} = S_i) \tag{6}$$

$$b_{j,k} = b_j(v_k) = P(O_t = v_k | q_t = S_j) \tag{7}$$

Here, π_i is the probability of the initial state S_i, a_{ij} is the transition probability between states i and j, and $b_{j,k}$ is the probability of the observation v_k when the model is in the state j.

We use a discrete HMM with 20 states for modeling each class of bills. Therefore, 24 HMMs are taken in total. Each line of data scanned by the line sensor is considered as an observation data vector. Thus, we have 50 observation vectors for each bill, each vector contains 25 pixels. In order to limit the number of observation vectors and quantize them into a map of codebook vectors, a self-organizing map (SOM) clustering is used. The size of the map is taken as 10×10.

Prior to recognition, each HMM is initialized and trained independently by the Baum-Welch algorithm[7] to maximize the probability of the observation sequences. Then for each input test sample represented as a sequence of T observations, a Viterbi algorithm[7] is performed to give a list of class models which maximize the probability of observations.

5 Reliability validation criterion

There exist numerous methods for estimating the accuracy of classifiers, but they cannot be easily generalized. Two popular methods that are commonly used in classification problems are cross-validation and bootstrap.[8] Both of them are based on resampling of the test data for estimating the accuracy of the classification. This requires the system to be retrained many times, and therefore, the speed of the training is significantly lowered.

In this article we propose a simple but effective algorithm for validating the reliability, which is well adjusted to the LVQ classifier and is based on determining the probability densities of data around codebook vectors and then calculating the overlaps between densities of different classes.

After training the LVQ classifier and determining the codebook vectors, the test data set is used to estimate the probability density function (pdf) for each codebook vector supposing a Gaussian distribution as follows:

$$N(\mu_i, \sigma_i) = 1\Big/\left(\sqrt{2\pi}\sigma_i\right)\exp\left(-(\xi-\mu_i)^2/2\sigma_i^2\right) \tag{8}$$

where ξ $(= \| x_i - m_i \|)$ is the distance between data vector and codebook vector, and μ_i and σ_i are the mean and variance of the density corresponding to the codebook vector i, respectively. Considering the Gaussian density function for each codebook vector, the interval $[\mu_i - 4.5\sigma_i, \mu_i + 4.5\sigma_i]$ can be supposed as an area that covers almost 100% of probabilities $(= 100 - 5.122 \times 10^{-5})$ If all pdfs of codebook vectors within a class have no overlap within this interval with densities of other classes, the reliability for this class is supposed to be 100%, but in the case that this area is overlapped with the distribution of other classes, the reliability for each codebook density can be calculated as

$$\mathrm{RM} = \int_{\mu_i - 4.5\sigma_i}^{\theta_i} N(\mu_i, \sigma_i)\, d\xi \tag{9}$$

where θ is the cross point of the density with the interval boundary of the nearest density belonging to the different class (Fig. 5).

Consequently, the overall reliability of the class can be considered as the minimum reliability values of the codebook vectors. Finally, the total reliability rate of the system is determined by averaging these class reliability values.

6 Experimental results and discussion

The classification method and the reliability validating algorithm discussed above, have been experimented by using a data set of 3570 samples from six kinds of US dollar bills, including \$1, \$5, \$10, \$20, \$50, and \$100 (taking four directions of rotation for each bill). The bills we used were of various levels of fatigue and were issued in different years. The order of inputting data was quite random in both learning and testing phases.

6.1 Learning vector quantization

Among the input samples, 2400 samples were used for learning the LVQ classifier and the remaining data containing both normal and slightly shifted data, were taken for evaluating the system. The data were acquired through the point sensors, and after the preprocessing steps explained earlier and feature extraction by PCA, a set of 30 dimensional feature vectors was provided for classification.

We have tried different numbers of codebook vectors from three to ten per class looking for the best result. On the other hand, we have also tried different number of components from 10 to 50, seeking the best choice that maximizes the classification and reliability rate. The results of classification rate and reliability are shown in Table 1. The reliability is evaluated through the algorithm we defined in Sect. 5.

As can be seen in Table 1, the recognition rate is 100% for all different values of PCA components when the number of codebook vectors is taken as 120 or 200. In other words, it is more affected by the number of codebooks rather than PCA components. However, the reliability rate is influenced by the number of both codebook vectors and PCA components. Codebook vectors have a direct impact on the reliability rate, that is, increasing the number of codebook vectors will increase the reliability rate. In the case of PCA components, we have an inverse ratio, that is, decreasing the number of PCA components causes an increase in the reliability rate. When the number of components drops below a threshold value, the reliability starts to decrease. Figures 6 and 7 clearly indicate the relation between reliability and number of codebook vectors as well as the number of PCA components.

It is worth noting that the PCA, however, increases the variance within the new components space, but at the same time reduces the overlaps between the densities of codebook vectors. This is the key point for improving the reliability as we have already explained in Sect. 5.

We found experimentally that a choice of 30 components is the best and the reliability of the system can be extended up to 95% over the test data space, when the number of codebook vectors is taken to be 200.

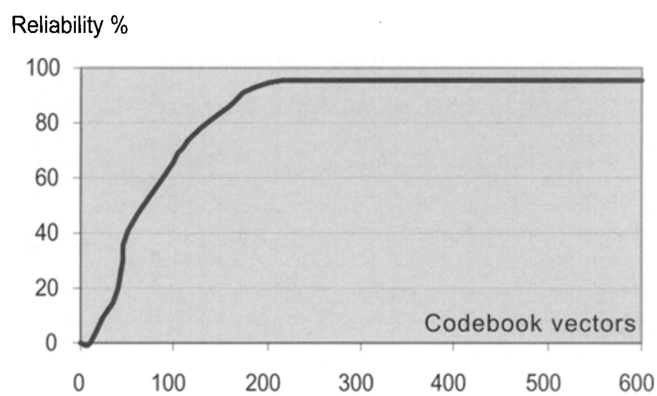


Fig. 6. The relation between reliability and number of codebooks vectors

Table 1. The results of classification before and after using principal component analysis (PCA)

No. of codebook	No. of components	Recognition rate (%)	Reliability rate (%)
80	150 (original data)	82	70
	50 (PCA)	88	75
	40 (PCA)	88	75.2
	30 (PCA)	90	77
	15 (PCA)	87	74
120	150 (original data)	94	80.3
	50 (PCA)	98	82.1
	40 (PCA)	98	84.2
	30 (PCA)	98	86
	15 (PCA)	98	85
200	150 (original data)	100	85.6
	50 (PCA)	100	89.4
	40 (PCA)	100	93
	30 (PCA)	100	95
	15 (PCA)	100	91.2
400	150 (original data)	100	82.1
	50 (PCA)	100	85
	40 (PCA)	100	87
	30 (PCA)	100	91.5
	15 (PCA)	100	90.5

6.2 Hidden Markov models

For the HMMs classifier, the same data set as in LVQ was applied. We used data frames of all the samples taken by the line sensor, in order to initialize the 24 HMMs and estimate their parameters (i.e., total of 3570 × 50 frames). A single 10 × 10 SOM was used for clustering the data frames and reducing the amount of data. The initial neighborhood radius for training of the SOM was taken to be ten and the initial learning rate was taken to be 0.2, selected through trial and error. Then, 1200 samples were used to test the system and measure the classification rate of HMMs. The same SOM was used for data quantizing in the test phase. The Viterbi algorithm was employed at this step to give a list of class models that maximize the probability of observations. Table 2 shows the results of the classification for all 24 classes. It should be noted that we did not evaluate the reliability and the results are only due to the classification rate. The Top(k) recognition rate means the percentage of samples whose corresponding class is among the first k candidates of HMMs. By taking higher values of k, the recognition rate takes a higher value. However, this decreases the accuracy of the measurement. The highest recognition rate Top(3) is 81.7%, which is significantly lower than the LVQ when applying PCA and taking 200 codebook vectors.

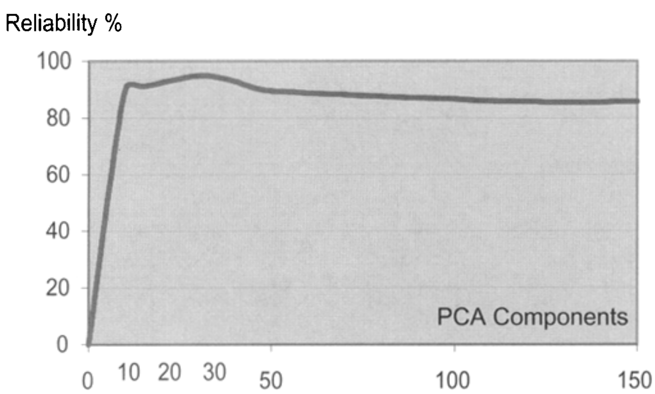


Fig. 7. The relation between reliability and number of PCA components

Next, we tried a 7 × 7 SOM for data frames of each class instead of taking a single SOM for entire data frames. In this case, the recognition rate increased considerably. The results are shown in Table 3.

As is displayed, there is still a significant difference between HMM classification rate and that of the LVQ classifier. The low classification rate of HMMs seems to be mainly due to the data quantization which we have adopted for reducing the data size and easing the classification process.

The results show that although the LVQ classifier works as an off-line model and may not be suitable for on-line performance applications like the current system, its classification rate is still higher and more reliable when compared with HMMs.

7 Conclusions

In this article, we applied an LVQ classifier for classification of paper currency. A PCA-based feature extraction was performed on the data to select the most effective compo-

Table 2. The recognition rate after using a single 10 × 10 self-organizing map (SOM) for data frames of all classes

Class	Recognition rate				
	Test samples	Top(1) (%)	Top(2) (%)	Top(3) (%)	Mis (%)
$1	50	40	55	75	25
$5	50	55	75	90	10
$10	50	45	65	75	25
$20	50	55	65	80	20
$50	50	50	60	80	20
$100	50	55	80	90	10
Total	1200	50	66.7	81.7	18.3

Mis, denotes misclassification rate

Table 3. The recognition rate after using a 7 × 7 SOM for data frames of each class

Class	Recognition rate				
	Test samples	Top(1) (%)	Top(2) (%)	Top(3) (%)	Mis (%)
$1	50	50	60	75	25
$5	50	75	100	100	0
$10	50	60	75	90	10
$20	50	55	75	95	5
$50	50	65	80	100	0
$100	50	75	90	100	0
Total	1200	63.3	80	93.3	6.7

nents. A new method is presented for validating the reliability of classification by using the densities of codebook vectors and determining the overlap zones. The method is simple and easy to use in classifiers that are based on codebooks.

Compared with HMMs, which we have been applied as an alternative classifier, the results of classification and reliability in LVQ have been found to be more robust. However, in order to increase the reliability rate of the LVQ classifier closer to 100%, it seems that we still need an additional clustering stage prior to LVQ to model the complexity of the real data and to give a clear estimation of data density.

HMMs are preferred for on-line classification and generally show faster performance in recognition of sequential data. However, in the current system it seems there are still some difficulties due to generating a proper sequence of data features. In addition, because we used quantized data rather than continuous densities, the classification rate was significantly reduced. Further work is in progress to make some new modifications in the HMMs training method, such as smoothing the HMMs parameters by using the neighborhood function of SOM, to make the classification more robust.

References

1. Frosini A, Gori M, Priami P (1996) A neural network-based model for paper currency recognition and verification. IEEE Trans Neural Network 7:1482–1490
2. Kosaka T, Taketani N, Omatu S (1999) Classification of Italian bills by a competitive neural network. Trans Inst Elec Eng Jpn 119-C:948–954
3. Fukunaga K (1972) Introduction to statistical pattern recognition. Academic, New York
4. Tipping ME, Bishop CM (1999) Probabilistic principal component analysis. J Roy Stat Soc B 61:611–622
5. Haykin S (1999) Neural networks. Prentice Hall, New Jersey
6. Kohonen T (1995) Self-organization maps. Springer, Berlin Heidelberg New York
7. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of IEEE 77:257–286
8. Kohavi R (1995) A study of cross-validation and bootstrap for accuracy estimation and model selection. Proceedings of the 14th International Joint Conference on Artificial Intelligence IJCAI, Montreal, Canada, pp 1137–1145