Available online at www.sciencedirect.com
SCIENCE DIRECT
Neural Networks 18 (2005) 958–966
Neural Networks
www.elsevier.com/locate/neunet

A robust classifier combined with an auto-associative network for completing partly occluded images

Takashi Takahashi[a,*], Takio Kurita[b]

[a]Department of Applied Mathematics and Informatics, Ryukoku University, Ootsu, Shiga 520-2194, Japan
[b]National Institute of Advanced Industrial Science and Technology (AIST), Tsukuba, Ibaraki 305-8568, Japan

Received 3 August 2004; revised 23 March 2005; accepted 23 March 2005

Abstract

This paper describes an approach for constructing a classifier which is unaffected by occlusions in images. We propose a method for integrating an auto-associative network into a simple classifier. As the auto-associative network can recall the original image from a partly occluded input image, we can employ it to detect occluded regions and complete the input image by replacing those regions with recalled pixels. By iterating this reconstruction process, the integrated network is able to classify target objects with occlusions robustly. To confirm the effectiveness of this method, we performed experiments involving face image classification. It is shown that the classification performance is not decreased, even if about 30% of the face image is occluded.

Keywords: Auto-associative network; Principal component analysis; Face recognition; Occlusion; Outlier; Recurrent data reconstruction

1. Introduction

In general situations, recognition target in a given image is often occluded by uninteresting objects (e.g. sunglasses on human faces). If a recognition system can automatically detect occluded regions in an image and estimate the original information corresponding to those regions, it is expected that the system will achieve improved recognition ability and extended applicability. One of the methods for realizing such functions is to incorporate a sort of auto-associative memory into the system. As the auto-associative memory can recall the whole image from partial data (Kohonen, 1989), we can use the recalled image to discriminate occluded pixels as outliers from those belonging to a target object. It is also possible to reconstruct a given image by replacing pixel values in the occluded regions with the recalled pixel values. In light of this principle, we investigate how to construct an auto-associative network which can complete partly occluded images and is suitable for combining with a conventional classifier.

In recent studies regarding view-based pattern recognition, dimensionality reduction techniques such as Principal Component Analysis (PCA) are often used before classification. The eigenface method (Kirby & Sirovich, 1990; Turk & Pentland, 1991) is typical. Such a method can be considered to have a feed-forward computation architecture as shown in Fig. 1(a). In this architecture, each input data vector is processed using one-way transformation by successively applying a dimensionality reduction algorithm and a classification algorithm. In contrast to this, we adopt an architecture which performs recurrent computation as shown in Fig. 1(b). The lower part of tile figure describes an auto-associative network that reconstructs an input vector as well as reduces dimensionality. If an input image is partly occluded or contaminated with outliers, the image is completed by repeating these two processes. As a result, dimensionally reduced feature components are extracted without being affected by outliers, in other words, by pixels in occluded regions. Accordingly, classification performance is improved in terms of robustness against occluded images.

To implement the above function of auto-associative network, PCA is employed. We can define a three-layer

* Corresponding author. Tel.: +81 77 543 7501; fax: +81 77 543 7524.
E-mail address: takataka@math.ryukoku.ac.jp (T. Takahashi).

0893-6080/$ - see front matter © 2005 Elsevier Ltd. All rights reserved.
doi:10.1016/j.neunet.2005.03.011

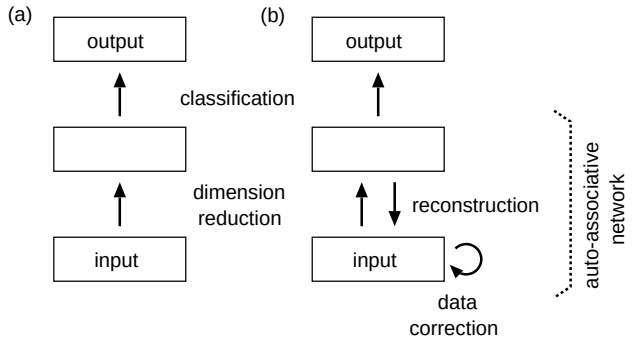

Fig. 1. A schematic diagram of (a) conventional classifier architecture and (b) the proposed architecture.

perceptron through PCA for given data. The perceptron consists of linear units and performs linear dimensionality reduction and reconstruction. The hidden-layer units extract the principal components of an input while the output-layer units reconstructs the input itself. It is known that such a perceptron can be obtained via learning to map each input vector onto itself (Baldi & Hornik, 1989; Diamantaras & Kung, 1996). Besides PCA, we also use kernel PCA (Schölkopf, Smola, & Müller, 1998). Kernel PCA can be considered a natural generalization of standard (linear) PCA and it enables us to extract non-linear feature components.

This paper is structured as follows. In Section 2, we briefly describe PCA and introduce a computation method called *Recurrent Data Reconstruction*. This is a procedure to detect pixels in occluded regions and to replace their values with estimated values by means of an auto-associative network. In Section 3, this method is extended to kernel PCA. Next, Section 4 shows some experimental results of the application of our method to a face recognition task. Section 5 discusses works related to our method, and Section 6 concludes the present study. Part of the results described in this paper have abstracted in two conference proceedings (Kurita, Takahashi, & Ikeda, 2002; Takahashi & Kurita, 2002).

2. Principal component analysis and recurrent data reconstruction

This section introduces a computation process called *Recurrent Data Reconstruction*. This is a method for modifying data vectors repeatedly using the outputs of an auto-associative network in order to remove outliers in data vectors and to improve the network's robustness. In this section, we describe a recurrent data reconstruction process for an auto-associative network created by applying Principal Component Analysis to data vectors. Such a network can be considered a three-layer perceptron consisting of linear units.

2.1. Principal component analysis

Principal Component Analysis is a widely used technique in data analysis. The purpose of PCA is to find the optimal linear transformation for extracting lower dimensional feature vectors which can adequately describe a set of high-dimensional data.

Consider an M-dimensional random variable $\boldsymbol{x}$. It is assumed that the mean of $\boldsymbol{x}$ is $E[\boldsymbol{x}]=0$ and the covariance matrix $C=E[\boldsymbol{x}\boldsymbol{x}^{\mathrm{T}}]$. In PCA, the feature vector $\boldsymbol{y}\in R^{H}(H<M)$ is computed as an orthogonal linear transformation of $\boldsymbol{x}$:

$$\boldsymbol{y} = U\boldsymbol{x}, \tag{1}$$

where U denotes an $H\times M$ matrix which satisfies $UU^{\mathrm{T}}=I$. Then $\boldsymbol{x}$ can be reconstructed from $\boldsymbol{y}$ as the projection onto the subspace spanned by column vectors of U:

$$\boldsymbol{z} = U^{\mathrm{T}}\boldsymbol{y} = U^{\mathrm{T}}U\boldsymbol{x}. \tag{2}$$

PCA seeks the optimal matrix U in terms of mean squared reconstruction error:

$$J_e = E[||\boldsymbol{x}-\boldsymbol{z}||^2]. \tag{3}$$

Let $\lambda_1, \lambda_2,\ldots,\lambda_N$ be the eigenvalues of C, and let their corresponding normalized eigenvectors be $\boldsymbol{e}_1, \boldsymbol{e}_2,\ldots,\boldsymbol{e}_N$. The eigenvalues are assumed to be arranged in decreasing order. It is then shown that the optimal matrix U minimizing J_e under the constraint $UU^{\mathrm{T}}=I$ has the form

$$U^{*} = T[\pm\boldsymbol{e}_1 \pm \boldsymbol{e}_2 \cdots \pm\boldsymbol{e}_H]^{\mathrm{T}}, \tag{4}$$

where T is any square orthogonal matrix (Diamantaras & Kung, 1996). It is also shown that the minimization of J_e is equivalent to the maximization of the variance of the feature vector $\boldsymbol{y}$:

$$J_v = E[\mathrm{tr}(\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}})]. \tag{5}$$

Hence, in regard to minimal reconstruction error and maximal variance, U^* gives the feature vector which describes the data's characteristics more accurately than does any other H-dimensional linear transformations.

In most applications, the covariance matrix is estimated by a sample covariance matrix. Consider a set of data

$$\{\boldsymbol{x}_i \in R^{M}\}_{i=1}^{N} \tag{6}$$

whose sample mean $\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i = 0$. The sample covariance matrix $\hat{C}=\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i\boldsymbol{x}_i^{\mathrm{T}}$. Then the matrix U is found by solving the eigenvalue problem of $\hat{C}$:

$$\hat{C}e = \lambda e. \tag{7}$$

2.2. Recurrent data reconstruction

In the field of pattern recognition, PCA has been widely applied in order to reduce the dimensions of the data. It enables us to extract a small number of feature components

called principal components which account for variation in high-dimensional data. Accordingly, it is useful in reducing the computational costs involved in treating real-world data. In case of face recognition, for instance, it is known that dozens of principal components of human face images are sufficient as inputs to a classifier for recognition (Turk & Pentland, 1991).

Besides computational costs, PCA-based dimensionality reduction reduces the detrimental influence of undesirable change in input data by discarding minor components. The principal components of face images are less affected by small levels of noise or minute changes of expression, lighting conditions, etc. than are raw images; therefore, they are effective for face recognition. However, most principal components are seriously affected by the contamination of some input components by outliers. They also have a great influence on reconstruction. Fig. 2 shows examples of a reconstruction from face images with outliers. Hence, in order to accomplish robust recognition using the PCA-based technique, it is necessary to improve the dimensionality reduction process.

As an approach to the above problem, we introduce a method composed of two steps: (1) detecting outlier components in each input vector, and (2) replacing each outlier component with some estimated value. In the first step, component-wise errors between an input and its reconstruction are used. If some input component is an outlier, it is expected to have a large error. In the second step, the detected outlier components are replaced with the corresponding components of the reconstruction. These steps can be applied repeatedly. If a partly occluded face image is inputted, the principal components should represent the features of the original unocculded face image and the reconstruction should be a accurate approximation of it after the repetition of these stops. We refer to this method as *Recurrent Data Reconstruction* (RDR).

The detailed procedure of recurrent data reconstruction is described as follows.

Step 0: The iteration parameter t is initialized as $t=0$ and some initial value is assigned to the input vector $\tilde{\boldsymbol{x}}_0$.

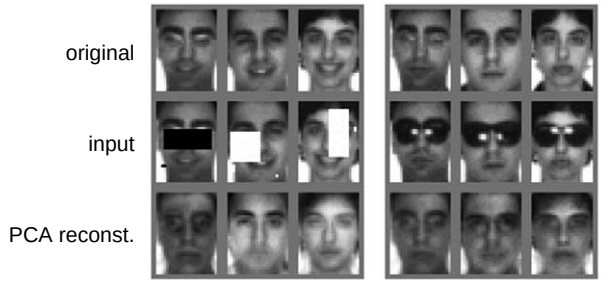

Fig. 2. Examples of occluded images and their reconstruction. Top row: original face images. Middle row: partly occluded images. Bottom row: their reconstruction via PCA.

Step 1: The vector of the principal components $\boldsymbol{y}_t$ and the reconstructed input $\boldsymbol{z}_t$ is computed with respect to the input $\tilde{\boldsymbol{x}}_t$.

$$\boldsymbol{y}_t = U\tilde{\boldsymbol{x}}_t \tag{8}$$

$$\boldsymbol{z}_t = U^{\mathrm{T}}\boldsymbol{y}_t = U^{\mathrm{T}}U\tilde{\boldsymbol{x}}_t \tag{9}$$

Step 2: The $M\times M$ diagonal matrix $\boldsymbol{A}_t=\mathrm{diag}(\alpha_{1t}, \alpha_{2t},\ldots,\alpha_{Mt})$ is set as

$$\alpha_{jt} = \begin{cases} 0 & \text{if } |\boldsymbol{x}_j - \boldsymbol{z}_{jt}| \geq 2.5\sigma_j, \\ 1 & \text{otherwise} \end{cases} \tag{10}$$

where z_{jt} denotes the jth component of the reconstruction $\boldsymbol{z}_t$, and σ_j is the constant defined later.

Step 3: The new input $\tilde{\boldsymbol{x}}_{t+1}$ is computed from the original input $\boldsymbol{x}$ and the reconstruction $\boldsymbol{z}_t$ as

$$\tilde{\boldsymbol{x}}_{t+1} = A_t\boldsymbol{x} + (I - A_t)\boldsymbol{z}_t. \tag{11}$$

Step 4: If t reaches the specified value, the procedure is terminated. Otherwise, set $t\leftarrow t+1$ and we move to Step 1.

Eq. (10) means that the jth component of the original data, x_j, is identified as an outlier if the difference between x_j and the corresponding component of the tth reconstruction, z_{jt}, is large. Eq. (11) means that x_j is replaced as the new input for the next iteration by z_{jt} (see also Fig. 3).

The constants α_{jt} $(j=1,2,\ldots,M)$ represent the standard deviations of the errors. They are estimated by robust statistical method (Huber, 1981) based in advance on training data. When the training data set is given by Eq. (6) and each

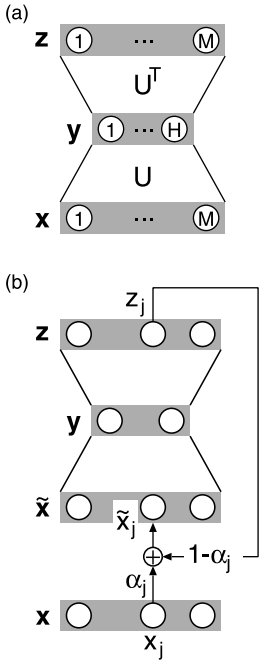

Fig. 3. Dimensionally reduction and reconstruction by PCA. (a) Conventional reconstruction and (b) Recurrent data reconstruction.

datum is reconstructed as $z_i = U^{\mathrm{T}}Ux_i$, σ_j is defined as:

$$\sigma_j = 1.4826\left(1+\frac{5}{N-1}\right)\underset{i}{\mathrm{med}}\,|x_{ij}-z_{ij}|. \tag{12}$$

Here med(x) denotes the median of $\{x\}$, and x_{ij} and z_{ij} are the jth component of $\boldsymbol{x}_i$ and $\boldsymbol{z}_i$, respectively.

In addition to the above, it is necessary to define the initial value $\tilde{\boldsymbol{x}}_0$. The simplest way of setting the initial value is to set it as $\tilde{\boldsymbol{x}}_0 = \boldsymbol{x}$. We confirmed in our preliminary experiments that this initial value produced robust results. However, the following method was found to give more robust results. First, the standard deviation of each component of $\boldsymbol{x}_i$ is estimated as

$$\sigma_j^0 = 1.4826\left(1+\frac{5}{N-1}\right)\underset{i}{\mathrm{med}}\,|x_{ij}-\bar{x}_j|, \tag{13}$$

where $\bar{x}_j = \frac{1}{N}\sum_{i=1}^{N} x_{ij}$. Next, $\tilde{\boldsymbol{x}}_0$ is computed in the same way in that $\tilde{\boldsymbol{x}}_{t+1}$ is computed by Eqs. (10) and (11) using σ_j, z_{ij}, and $\boldsymbol{z}_t$, but using σ_j^0, $\bar{x}_j$, and $\bar{\boldsymbol{x}} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{x}_i$.

2.3. Convergence property of recurrent data reconstruction

From Eqs. (9) and (11), the following relationship holds between $\boldsymbol{z}_t$ and $\boldsymbol{z}_{t+1}$

$$\boldsymbol{z}_{t+1} = U^{\mathrm{T}}U\tilde{\boldsymbol{x}}_{t+1} = U^{\mathrm{T}}U(A_t\boldsymbol{x}+(I-A_t)\boldsymbol{z}_t) \tag{14}$$

$$= \boldsymbol{z}_t + U^{\mathrm{T}}UA_t(\boldsymbol{x}-\boldsymbol{z}_t) \tag{15}$$

because $U^TU\boldsymbol{z}_t = \boldsymbol{z}_t$. Let us consider the case that $\boldsymbol{z}_t$ converges to a fixed point $\boldsymbol{z}_\infty$ as $t\to\infty$. Then

$$\boldsymbol{z}_\infty = \boldsymbol{z}_\infty + U^{\mathrm{T}}UA_\infty(\boldsymbol{x}-\boldsymbol{z}_\infty), \tag{16}$$

and hence

$$UA_\infty(\boldsymbol{x}-\boldsymbol{z}_\infty) = 0 \tag{17}$$

should hold. Eq. (17) indicates that the vector $A_\infty(\boldsymbol{x}-\boldsymbol{z}_\infty)$ is orthogonal to the principal subspace spanned by the first H eigenvectors $\boldsymbol{e}_1,\boldsymbol{e}_2,\ldots,\boldsymbol{e}_H$. Therefore, $A_\infty\boldsymbol{z}_\infty$, the vector obtained by substituting zero for the outlier components of $\boldsymbol{z}_\infty$, has the same H principal components as those of $A_\infty\boldsymbol{x}$. If no component is identified as an outlier, i.e., if $A_\infty = I$, the reconstruction becomes identical to that obtained by conventional principal component analysis. From these properties, we can expect that recurrent data reconstruction effectively extracts the principal feature components of the data when their outlier components are successfully identified.

3. Recurrent data reconstruction for kernel PCA

In Section 2, we described the process of recurrent data reconstruction with the premise that dimensionality reduction and reconstruction is based on linear PCA. However, the process does not depend on the specific architecture of the auto-associative network. For our proposes, we can use a variety of network architectures such as five layer non-linear perceptron, kernel PCA, or a Hopfield network. In this section, we describe a method to combine recurrent data reconstruction with the data reconstruction process based on kernel PCA.

3.1. Kernel PCA

Kernel PCA can be derived using the known fact that PCA can be carried out on the dot product matrix instead of on the covariance matrix (Moerland, 2000; Schölkopf et al., 1998). Given a set of training data by Eq. (6), kernel PCA first maps the data into some feature space F by a function $\boldsymbol{\Phi}$: $R^M \to F$, and then performs standard PCA on the mapped data.

Defining the data matrix X by $X = [\boldsymbol{\Phi}(\boldsymbol{x}_1)\boldsymbol{\Phi}(\boldsymbol{x}_2)\ldots\boldsymbol{\Phi}(\boldsymbol{x}_N)]$, the sample covariance matrix in F becomes

$$\hat{C} = \frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Phi}(\boldsymbol{x}_i)\boldsymbol{\Phi}(\boldsymbol{x}_i)^{\mathrm{T}} = \frac{1}{N}XX^{\mathrm{T}}. \tag{18}$$

For simplicity, we assume that the mapped data are centered, i.e. $\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{\Phi}(\boldsymbol{x}_i) = 0$. This is not the case in general; however, all calculations can be reformulated to deal with centering (Schölkopf et al., 1998). Although it is in some cases intractable to carry out the direct eigen decomposition of $\hat{C}$, we can find the eigenvalues and eigenvectors of $\hat{C}$ via solving the eigenvalue problem:

$$\lambda\boldsymbol{u} = K\boldsymbol{u}. \tag{19}$$

The $N\times N$ matrix K is the dot product matrix defined by

$$K = \frac{1}{N}X^{\mathrm{T}}X, \tag{20}$$

where

$$K_{ij} = \frac{1}{N}\boldsymbol{\Phi}(\boldsymbol{x}_i)\cdot\boldsymbol{\Phi}(\boldsymbol{x}_j) = \frac{1}{N}k(\boldsymbol{x}_i,\boldsymbol{x}_j) \tag{21}$$

and $k(\boldsymbol{x},\boldsymbol{y}) = \boldsymbol{\Phi}(\boldsymbol{x})\cdot\boldsymbol{\Phi}(\boldsymbol{y})$ denote the kernel function which substitutes the dot product $\boldsymbol{x}\cdot\boldsymbol{y}$.

Let $\lambda_1 \geq \ldots \geq \lambda_P$ be the non-zero eigenvalues of K ($P\leq N$ and $P\leq M$), and $\boldsymbol{u}_1,\boldsymbol{u}_2,\ldots,\boldsymbol{u}_P$ the corresponding eigenvectors. Then $\hat{C}$ has the same eigenvalues and there is a one-to-one correspondence between the non-zero eigenvectors $\{\boldsymbol{u}_h\}$ of K and the non-zero eigenvectors $\{\boldsymbol{v}_h\}$ of $\hat{C}$

$$\boldsymbol{v}_h = r_hX\boldsymbol{u}_h, \tag{22}$$

where r_h is a constant for normalization (Moerland, 2000). If both of the eigenvectors have unit length, $r_h = 1/\sqrt{\lambda_h N}$. In the following discussion, we assume $||\boldsymbol{u}_h|| = 1/\sqrt{\lambda_h N}$ so that $r_h = 1$.

For a test data $\boldsymbol{x}$, its hth principal component y_h can be computed using kernel functions:

$$y_h = \boldsymbol{v}_h \cdot \boldsymbol{\Phi}(\boldsymbol{x}) = \sum_{i=1}^{N} u_{hi} k(\boldsymbol{x}_i, \boldsymbol{x}). \tag{23}$$

Then the $\boldsymbol{\Phi}$-image of $\boldsymbol{x}$ can be reconstructed from its projections onto the first $H(\leq P)$ principal components in F by using a projection operator P_H:

$$P_H \boldsymbol{\Phi}(\boldsymbol{x}) = \sum_{h=1}^{H} y_h \boldsymbol{v}_h. \tag{24}$$

The procedure of kernel PCA is equivalent to that of standard PCA on the mapped data. Hence, kernel PCA inherits some properties of standard PCA. For instance, the reconstruction error $\sum_i ||\boldsymbol{\Phi}(\boldsymbol{x}_i) - P_H \boldsymbol{\Phi}(\boldsymbol{x}_i)||^2$ is minimal among all H-dimensional projection operators in F.

3.2. Recurrent data reconstruction using Gaussian kernels

In order to apply recurrent data reconstruction to kernel PCA, it is necessary to reconstruct the data in the input space R^M rather than in F. This can be achieved by seeking a vector $\boldsymbol{z}$ satisfying $\boldsymbol{\Phi}(\boldsymbol{z}) = P_H \boldsymbol{\Phi}(\boldsymbol{x})$. If such a $\boldsymbol{z}$ exists, it will be an accurate approximation of $\boldsymbol{x}$ in the input space. However, it will not always exist and it may not be unique even if it does exist. Gaussian kernels $k(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\|\boldsymbol{x} - \boldsymbol{y}\|^2/c)$ come under this case.

To settle the problem, Mika et al. (1999) proposed to approximate $\boldsymbol{z}$ by minimizing $\rho(\boldsymbol{z}) = \|\boldsymbol{\Phi}(\boldsymbol{z}) - P_H \boldsymbol{\Phi}(\boldsymbol{x})\|^2$. For kernels satisfying $k(\boldsymbol{x}, \boldsymbol{x}) \equiv$ constant for all $\boldsymbol{x}$, we can maximize the following expression instead of $\rho(\boldsymbol{z})$

$$\tilde{\rho}(\boldsymbol{z}) = \boldsymbol{\Phi}(\boldsymbol{z}) \cdot P_H \boldsymbol{\Phi}(\boldsymbol{x}) + \Omega = \sum_{i=1}^{N} w_i k(\boldsymbol{x}_i, \boldsymbol{z}) + \Omega, \tag{25}$$

where

$$w_i = \sum_{h=1}^{H} u_{hi} y_h \tag{26}$$

and Ω denotes the terms independent of $\boldsymbol{z}$. Employing standard gradient ascent methods for this, they derived an iteration scheme for computing optimal $\boldsymbol{z}$. Their iteration scheme is given as follows:

$$\boldsymbol{z}_{t+1} = \frac{\sum_{i=1}^{N} w_i k(\boldsymbol{x}_i, \boldsymbol{z}_t) \boldsymbol{x}_i}{\sum_{i=1}^{N} w_i k(\boldsymbol{x}_i, \boldsymbol{z}_t)}. \tag{27}$$

The recurrent data reconstruction procedure for kernel PCA is obtained by combining the above iteration scheme into the step reconstructing $\boldsymbol{z}$. Step 1 is modified as follows.

Step 1: The vector $\boldsymbol{y}_t$ consisting of the H principal components is computed with respect to the input $\tilde{\boldsymbol{x}}_t$ by

$$y_{ht} = \sum_{i=1}^{N} u_{hi} k(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_t). \tag{28}$$

Next the corresponding reconstruction $\boldsymbol{z}_t$ is computed by

$$w_{it} = \sum_{h=1}^{H} u_{hi} y_{ht} \tag{29}$$

$$\boldsymbol{z}_t = \frac{\sum_{i=1}^{N} w_{it} k(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_t) \boldsymbol{x}_i}{\sum_{i=1}^{N} w_{it} k(\boldsymbol{x}_i, \tilde{\boldsymbol{x}}_t)}. \tag{30}$$

The other steps are the same as those described in Section 2.2.

4. Experiment

In order to confirm the effectiveness of the proposed method regarding robustness against occluded images, we performed experiments using face images.

4.1. Conditions of the experiments

Face images were taken from the ARFace Database (Martinez & Benavente, 1998). The training data set consisted of $N = 171$ images (3 for each of 57 people) with 256 gray levels. The size of each image was normalized to $24 \times 32 (= 768)$. Examples of these face images are shown in the top row of Fig. 4.

PCA and kernel PCA were applied to these data. The dimensionality of the data was $M = 768$. For kernel PCA, we chose Gaussian kernels with $c = 0.08$ M. The parameter c was tuned by hand to attain lower error rates in the classification experiments described below. In the cases of both PCA and kernel PCA, the number of extracted principal components was set to $H = 25$. It was chosen to coincide with the dimensionality in which 92% of the total variance of the training data was preserved using PCA.

These H-dimensional vectors (principal components) were modified through recurrent data reconstruction in order to reduce the influence of outliers, and the resulting vectors were provided as inputs to a classifier. We adopted a multinomial logit model as the classifier (see Appendix).

The robustness of the proposed method was investigated in regard to two aspects: reconstruction error and classification error. In the reconstruction experiments, the proposed method was used to recall a person's face image in the training data based some test images.

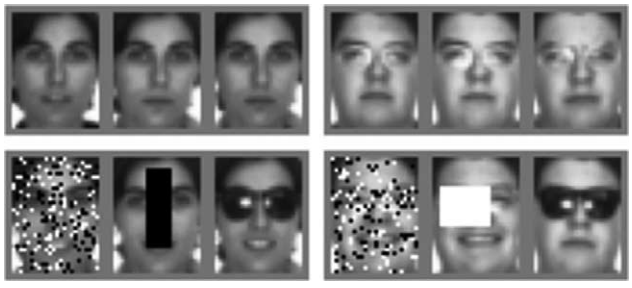

Fig. 4. Sample images in the training data set (top row) and in the test data set (bottom row).

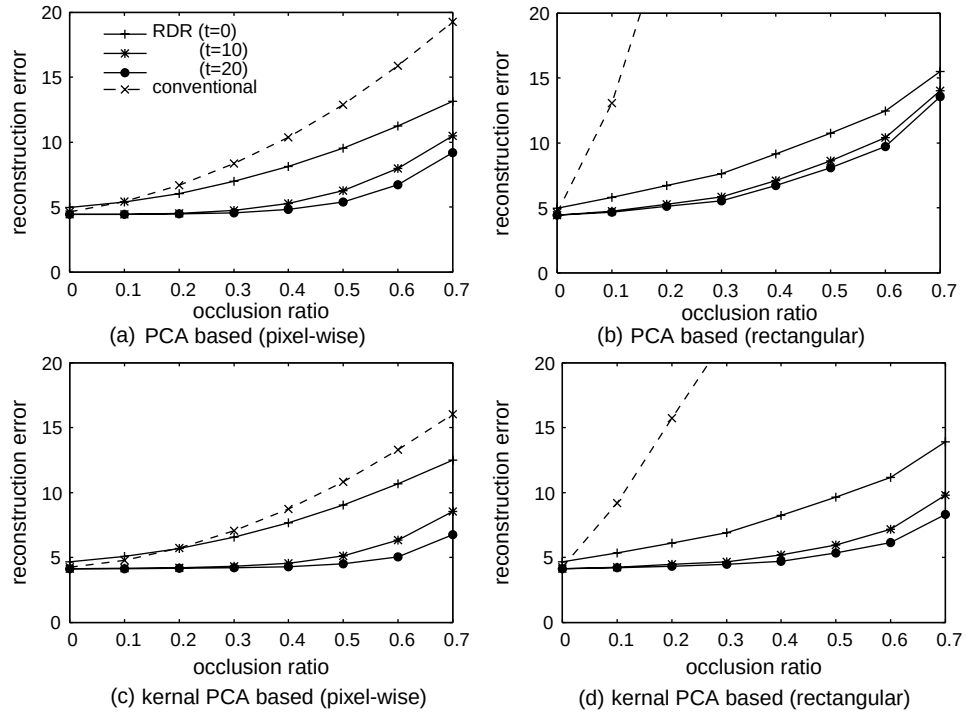

Fig. 5. Reconstruction errors for the test data with artificial occlusions.

Reconstruction performance was measured by mean squared errors between a reconstructed image and each of the three training images of the same face. On the other hand, in the classification experiments, the task was to classify test images into one of 57 classes according to the person's identities. Classification performance was measured by error rate, i.e. the ratio of misclassification to the total number of test data.

The test data set was composed of three types: *pixel-wise*, *rectangular*, and *sunglass*. The examples are shown on the bottom row of Fig. 4. The former two sets of data were generated by adding artificial occlusion to different images of the same 57 people (one for each person). The images in the pixel-wise data set were made by flipping each pixel to black or white with some constant probability, while those in the rectangular set were made by replacing some rectangular regions with black or white rectangles. On the other hand, the sunglass data set consisted of real-face images wearing sunglasses. The size of the pixel-wise and rectangular sets was 570 (10 different patterns of occlusion for each of 57 images). The sunglass set consisted of 57 images.

4.2. Reconstruction experiments

Fig. 5 and Table 1 show the reconstruction errors obtained by the proposed method. For quantitative comparison, the reconstruction errors obtained when not applying the proposed method (conventional reconstruction method by PCA) are also shown. It is noticed that the proposed recurrent data reconstruction method improves the robustness regarding occlusions with the repetition. It can also be seen that kernel PCA gives more robust results than does linear PCA.

Fig. 6 shows some examples of the test images and the reconstructed images. The occlusion ratios of the pixel-wise and rectangular test images were 0.4 and 0.2, respectively.

4.3. Classification experiments

Fig. 7 and Table 2 show the error rates of the classification experiments. Here we can see the same tendencies for robustness as in the reconstruction experiments. It is confirmed that the classification accuracy is significantly improved by applying the proposed recurrent data reconstruction method.

5. Discussion

The proposed method employs two computation processes based on PCA: the dimensionally reduction process

Table 1
Reconstruction errors for sunglass data

	PCA	kPCA
RDR		
$t=0$	7.17	6.53
$t=10$	7.58	5.10
$t=20$	7.73	5.00
Conventional	13.86	9.86

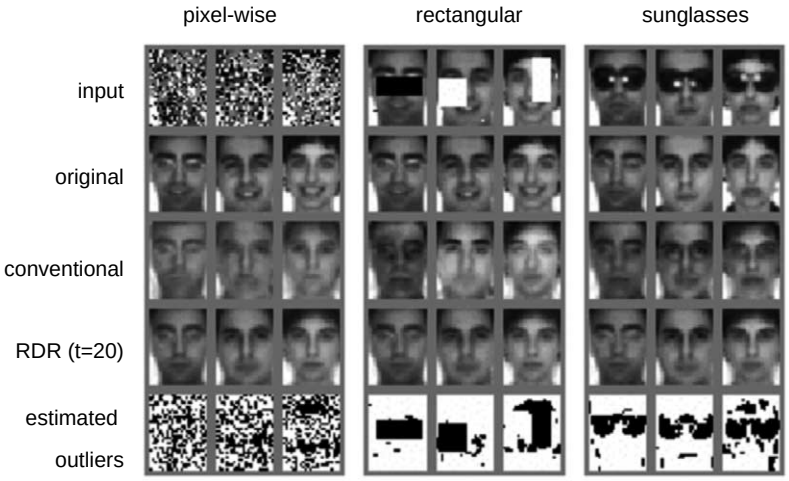

Fig. 6. Reconstruction from the occluded test images. *input:* test images. *original:* original images corresponding to the test images. *conventional:* reconstruction via conventional kernel PCA. *RDR* ($t=20$): reconstructed images after iterating the recurrent data reconstruction process 20 times. *estimated outliers:* black pixels correspond to estimated outliers pixels ($\alpha_{jt}=0$) while white pixels indicate $\alpha_{jt}=1$.

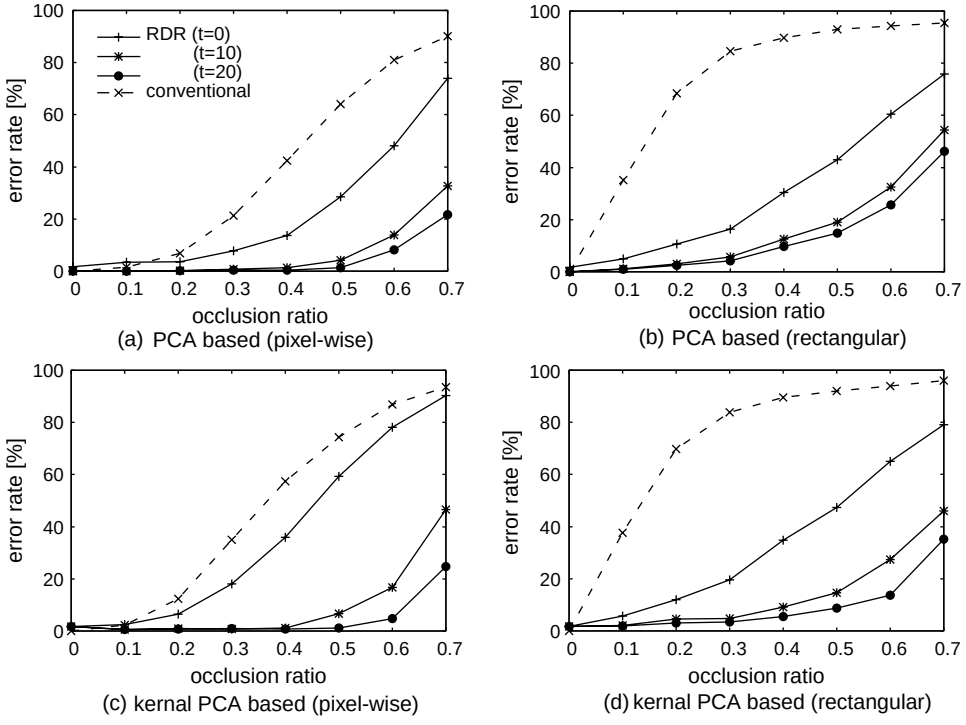

Fig. 7. Classification error rates for the test data with artificial occlusions.

and the reconstruction process. The dimensionality reduction computation extracts feature components from data while reducing the dimensionality of the data, and the reconstruction computation recovers the data while removing the influence of outliers. Thus, these processes play the most important role in the proposed method. There are several related works which focus on improving the robustness of standard PCA.

In the field of neural networks, it is well known that multi-layer networks can perform such PCA-like

Table 2
Classification error rates for sunglass data

	PCA (%)	kPCA (%)
RDR		
$t=0$	17.5 (10/57)	15.8 (9/57)
$t=10$	12.3 (7/57)	7.0 (4/57)
$t=20$	12.3 (7/57)	5.3 (3/57)
Conventional	60.0 (34/57)	61.4 (35/57)

computation. Xu and Yuille (1995) proposed a learning algorithm for robust PCA. By using their algorithm, the data vectors of outliers can be discriminated from regular data vector. Although their algorithm allows the complete rejection of each data vector, it does not work in a case in which some components of each data vector are outliers constituting occluded images.

Black et al. investigated a robust algorithm for PCA (Black & Jepson, 1996; De la Torre & Black, 2001). Their method uses M-estimation for discriminating outliers and seeking principal components. Sakaue and Shakunaga proposed a similar algorithm (Sakaue & Shakunaga, 2004). These related works aim to develop a method for removing outliers in data vectors, and therefore, their approaches are similar to ours. However, one unique characteristic distinguishes our method from theirs. Unlike these approaches based on linear PCA, our approach can employ a variety of computation algorithms other than linear projections. In the present paper, we adopted kernel PCA and showed experimentally that it outperformed the method based on linear PCA.

In addition to the above, the proposed method may be reasonable from a neuroscientific point of view. In the primate cerebral cortex, visual information is processed in the primary visual cortex (V1), and output signals from V1 are further processed in successive areas along the visual pathways. It is well known that reciprocal connections exist among these cortical areas. For instance, forward neuronal connections from V1 to the neighboring areas are always accompanied by backward connections. Many researchers have investigated the function of such reciprocal connectivity and proposed computational models. Fukushima (1987) proposed a model for selective attention with reciprocal neural connections. Okajima (1991) also showed by simulation experiments that a system backward connections can separate an object pattern from a background in a given image. Our neural network model can be considered a model that advocates the importance of recurrent computation in a biological visual system.

6. Conclusion

In this paper, we studied how to construct a classifier which is unaffected by occlusions in an image. We proposed a method to integrate an auto-associative network into a simple classifier. The auto-associative network is used to detect occluded regions and fill them with estimated original pixel values. By applying this process recursively, the integrated network can classify occluded images robustly. The effectiveness of the proposed method was confirmed by the result of experiments regarding face image classification. It was shown that stable classification performance was obtained even if 20–30% of the face images were occluded. We intend to apply this method to other tasks including face detection or moving object tracking.

Acknowledgements

This research was partially supported by the Ministry of Education, Science, Sports and Culture, Grant-in Aid for Young Scientists (B), No. 15700204.

Appendix. Multinomial logit model

The multinomial logit model is a special case of generalized linear models (McCullaph & Nelder, 1983), and it can be regarded as one of the simplest neural network models for solving a multi-way classification problem. This network model consists of only two layers: an input layer and an output layer. The output values of the network are computed through softmax competition among the output layer units. Let us consider a problem involving the classification of each L-dimensional vector into one of the K classes $\{C_1,\ldots,C_K\}$. The input and the output, layer consist of L and $K-1$ units, respectively. For a given input vector $\boldsymbol{\xi}\in R^{\boldsymbol{L}}$, the output values of the output layer units, $p_{\boldsymbol{k}}(k=1,2,\ldots,K-1)$, are computed as

$$p_k = \frac{\exp(\eta_k)}{1+\sum_{k'=1}^{K-1}\exp(\eta_{k'})}. \tag{31}$$

The value $\eta_{\boldsymbol{k}}$ represents a weighted sum of the input components

$$\eta_k = \boldsymbol{\omega}_k^{\mathrm{T}}\boldsymbol{\xi}, \tag{32}$$

where $\boldsymbol{\omega}_k$ denotes the weight vector between the input layer and the kth output unit. Each value p_k can be considered an estimate of the probability that $\boldsymbol{\xi}$ belongs to the kth class, and p_K is given by

$$p_K = \frac{1}{1+\sum_{k'=1}^{K-1}\exp(\eta_{k'})}. \tag{33}$$

Thus

$$\sum_{k=1}^{K} p_k = 1.$$

For this probabilistic model, the learning process of the network can be regarded as the maximum likelihood estimation of the parameters $\{\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_{\mathrm{K}-1}\}$. Consider a set of training data with teacher signals $\{({}^{(i)}\boldsymbol{\xi},{}^{(i)}\boldsymbol{t})\}_{i=1}^{N}$, where $\boldsymbol{t}=({}^{(i)}t_1,\ldots,{}^{(i)}t_K)^{\mathrm{T}}\in\{0,1\}^K$ denotes a vector composed of binary teacher signals: ${}^{(i)}t_{\boldsymbol{k}}=1$ if ${}^{(i)}\boldsymbol{\xi}$ should be classified into C_k, otherwise ${}^{(i)}t_{jk}=0$. Then the likelihood of the classifier for the training data is given by

$$P(\boldsymbol{t}|\boldsymbol{\xi}) = \prod_{i=1}^{N}\prod_{k=1}^{K}({}^{(i)}p_k)^{{}^{(i)}t_{\mathrm{k}}}, \tag{34}$$

and hence the log-likelihood becomes

$$\ell_C = \sum_{i=1}^{N}\sum_{k=1}^{K-1} (\mathrm{i})\mathrm{t}_k\,(\mathrm{i})\eta_k - \sum_{i=1}^{N}\log\left(1+\sum_{k=1}^{K-1}\exp({}^{(i)}\eta_k)\right). \quad (35)$$

Learning rules for the network are derived by considering the maximization of ℓ_C. By employing a standard gradient method, we can derive the updating rules of the network's weights by taking partial derivatives of ℓ_C

$$\Delta\omega_{kj} = \alpha\sum_{i=1}^{N}({}^{(i)}t_k - {}^{(i)}p_k)^{(i)}\xi_j, \quad (36)$$

where α is a constant which controls the learning rate.

References

Baldi, P., & Hornik, K. (1989). Neural networks and principal component analysis: Learning from examples without local minima. *Neural Networks*, *2*, 53–58.

Black, M. J., & Jepson, A. D. (1996). *Eigentracking: Robust matching and tracking of articulated objects using a view-based representation In proceeding ECCV* (pp. 329–342).

De la Torre, F., & Black, M. J. (2001). Robust principal component analysis for computer vision. *In proceedings ICCV* .

Diamantaras, K. I., & Kung, S. Y. (1996). *Principal component neural networks*. New York: Wiley.

Fukushima, K. (1987). Neural network model for selective attention in visual pattern recognition and associative recall. *Applied Optics*, *26*(3), 4985–4992.

Huber, P. J. (1981). *Robust statistics*. New York: Wiley.

Kirby, M., & Sirovich, L. (1990). Application of the Karhunen-Loéve procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysisi and Machine Intelligence*, *12*(1), 103–108.

Kohonen, T. (1989). *Self-organization and associative memory* (3rd ed.). New York: Springer-Verlag.

Kurita, T., Takahashi, T., & Ikeda, Y. (2002). *A neural network classifier for occluded images In proceedings of international conference on pattern recognition (ICPR2002)*, Vol. III (pp. 45–48).

Martinez, A. M., & Benavente, R. (1998). The AR face database. *CVC technical report* , 24.

McCullaph, P., & Nelder, J. A. (1983). *Generalized Linear Models*. London: Chapman and Hall.

Mika, S., Schölkopf, B., Smola, A., Müller, K.-R., Scholz, M., & Rätsch, G. (1999). Kernel PCA and de-noising in feature spaces. *In advances in neural information processing systems*, 11.

Moerland, P. (2000). An on-line EM algorithm applied to kernel PCA. Technical report, IDIAP.

Okajima, K. (1991). A recurrent system incorporating characteristics of the visual system: A model for the function of backward neural connections in the visual system. *Biological Cybernetics*, *65*, 235–241.

Sakaue, F., & Shakunaga, T. (2004). Robust projection onto normalized eigenspace using relative residual analysis and optimal partial projection. *IEICE Transactions Information and Systems*, *E87-D*(1), 31–41.

Schölkopf, B., Smola, A. J., & Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299–1319.

Takahashi, T., & Kurita, T. (2002). *Robust De-noising by kernel PCA In Artificial Neural Networks-ICANN2002*. Berlin: Springer (pp.739–744).

Turk, M. A., & Pentland, A. P. (1991). *Face recognition using eigenfaces. In proceedings of IEEE conference on computer vision and pattern recognition* (pp. 586–591).

Xu, L., & Yuille, A. A. (1995). Robust principal component analysis by self-organizaing rules based on statistical physics approach. *IEEE Transactions on Neural Networks*, *6*(1), 131–143.