Face Recognition: A Hybrid Neural Network Approach

Steve Lawrence[1,2]*, C. Lee Giles[1]†, Ah Chung Tsoi[2], Andrew D. Back[2]
{lawrence,act,back}@elec.uq.edu.au, giles@research.nj.nec.com

[1] NEC Research Institute, 4 Independence Way, Princeton, NJ 08540
[2] Electrical and Computer Engineering, University of Queensland, St. Lucia, Australia

Technical Report
UMIACS-TR-96-16 and CS-TR-3608
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742

April 1996 (Revised August 1996)

Abstract

Faces represent complex, multidimensional, meaningful visual stimuli and developing a computational model for face recognition is difficult (Turk and Pentland, 1991). We present a hybrid neural network solution which compares favorably with other methods. The system combines local image sampling, a self-organizing map neural network, and a convolutional neural network. The self-organizing map provides a quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, thereby providing dimensionality reduction and invariance to minor changes in the image sample, and the convolutional neural network provides for partial invariance to translation, rotation, scale, and deformation. The convolutional network extracts successively larger features in a hierarchical set of layers. We present results using the Karhunen-Loève transform in place of the self-organizing map, and a multilayer perceptron in place of the convolutional network. The Karhunen-Loève transform performs almost as well (5.3% error versus 3.8%). The multilayer perceptron performs very poorly (40% error versus 3.8%). The method is capable of rapid classification, requires only fast, approximate normalization and preprocessing, and consistently exhibits better classification performance than the eigenfaces approach (Turk and Pentland, 1991) on the database considered as the number of images per person in the training database is varied from 1 to 5. With 5 images per person the proposed method and eigenfaces result in 3.8% and 10.5% error respectively. The recognizer provides a measure of confidence in its output and classification error approaches zero when rejecting as few as 10% of the examples. We use a database of 400 images of 40 individuals which contains quite a high degree of variability in expression, pose, and facial details. We analyze computational complexity and discuss how new classes could be added to the trained recognizer.

Keywords: Convolutional Networks, Hybrid Systems, Face Recognition, Self-Organizing Map

* http://www.neci.nj.nec.com/homepages/lawrence, http://www.elec.uq.edu.au/~lawrence
†Also with the Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742.

1 Introduction

The requirement for reliable personal identification in computerized access control has resulted in an increased interest in biometrics[1]. Biometrics being investigated include fingerprints (Blue, Candela, Grother, Chellappa and Wilson, 1994), speech (Burton, 1987), signature dynamics (Qi and Hunt, 1994), and face recognition (Chellappa, Wilson and Sirohey, 1995). Sales of identity verification products exceed $100 million (Miller, 1994). Face recognition has the benefit of being a passive, non-intrusive system for verifying personal identity. The techniques used in the best face recognition systems may depend on the application of the system. There are at least two broad categories of face recognition systems:

1. The goal is to find a person within a large database of faces (e.g. in a police database). These systems typically return a list of the most likely people in the database (Pentland, Starner, Etcoff, Masoiu, Oliyide and Turk, 1993). Often only one image is available per person. It is usually not necessary for recognition to be done in real-time.
2. The goal is to identify particular people in real-time (e.g. in a security monitoring system, location tracking system, etc.), or to allow access to a group of people and deny access to all others (e.g. access to a building, computer, etc.) (Chellappa et al., 1995). Multiple images per person are often available for training and real-time recognition is required.

This paper is primarily concerned with the second case[2]. This work considers recognition with varying facial detail, expression, pose, etc. Invariance to high degrees of rotation or scaling are not considered – it is assumed that a minimal preprocessing stage is available if required (i.e. to locate the position and scale of a face in a larger image). We are interested in rapid classification and hence we do not assume that time is available for extensive preprocessing and normalization. Good algorithms for locating faces in images can be found in (Turk and Pentland, 1991; Sung and Poggio, 1995; Rowley, Baluja and Kanade, 1995).

The remainder of this paper is organized as follows. The data used is presented in section 2 and related work with this and other databases is discussed in section 3. The components and details of our system are described in sections 4 and 5 respectively. Results are presented and discussed in sections 6 and 7. Computational complexity is considered in section 8 and conclusions are drawn in section 10.

2 Data

The database used is the ORL database which contains photographs of faces taken between April 1992 and April 1994 at the Olivetti Research Laboratory in Cambridge, UK[3]. There are 10 different images of 40 distinct subjects. For some of the subjects, the images were taken at different times. There are variations in facial expression (open/closed eyes, smiling/non-smiling), and facial details (glasses/no glasses). All of the images were taken against a dark homogeneous background with the subjects in an upright, frontal position, with tolerance for some tilting and rotation of up to about 20 degrees. There is some variation in scale of up to about 10%. Thumbnails of all of the images are shown in figure 1 and a larger set of images for one subject is shown in figure 2. The images are greyscale with a resolution of 92×112.

[1] Physiological or behavioral characteristics which uniquely identify people.

[2] However, experiments have not been performed where the system was required to reject people that are not in a select group (important, for example, when allowing access to a building).

[3] The ORL database is available free of charge, see `http://www.cam-orl.co.uk/facedatabase.html`.

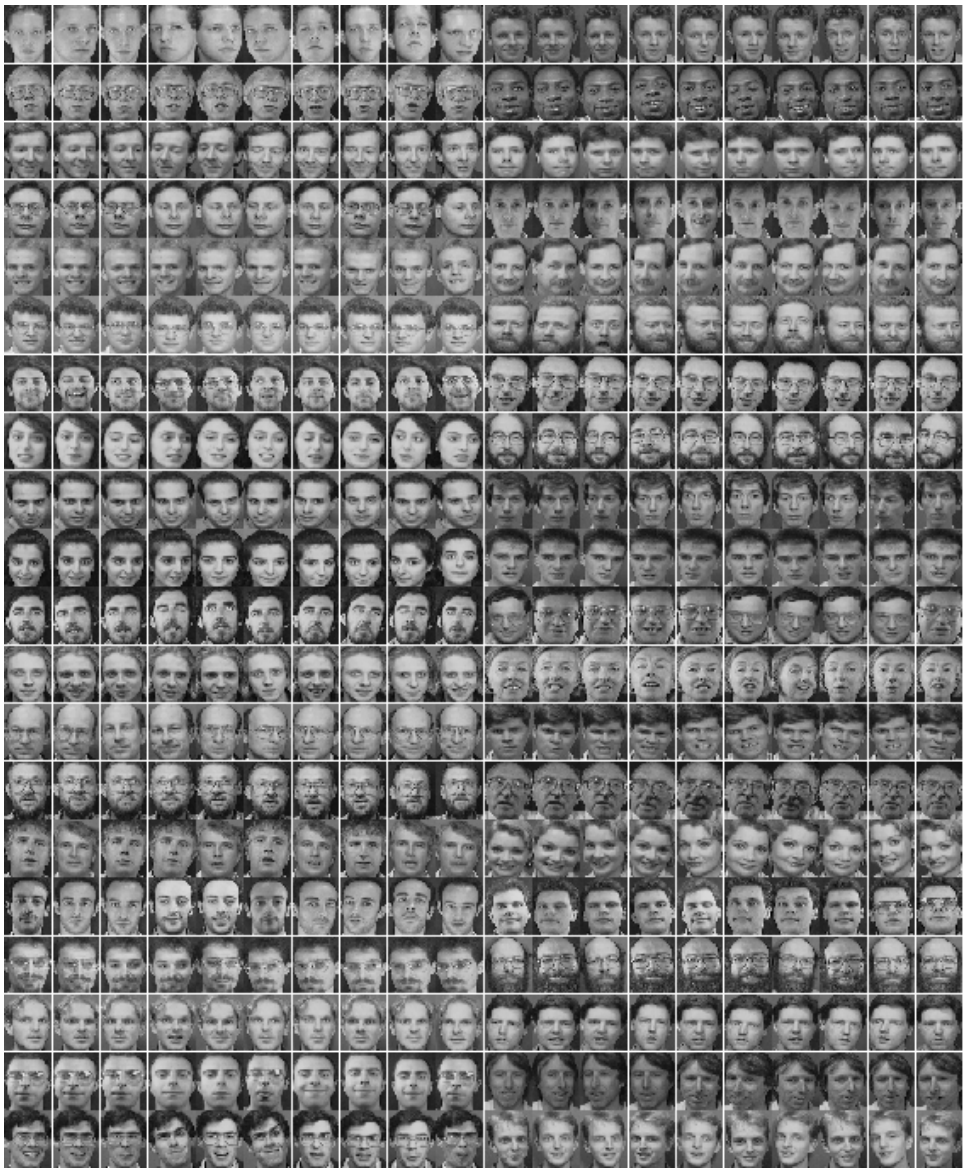

Figure 1. The ORL face database. There are 10 images each of the 40 subjects.

3 Related Work

This section summarizes related work on face recognition – geometrical feature based approaches, template matching, neural network approaches, and the popular *eigenfaces* technique.

Figure 2. The set of 10 images for one subject. Considerable variation can be seen.

3.1 Geometrical Features

Many people have explored geometrical feature based methods for face recognition. Kanade (1973) presented an automatic feature extraction method based on ratios of distances (between feature points such as the location of the eyes, nose, etc.) and reported a recognition rate of between 45-75% with a database of 20 people. Brunelli and Poggio (1993) compute a set of geometrical features such as nose width and length, mouth position, and chin shape. They report a 90% recognition rate on a database of 47 people. However, they show that a simple template matching scheme provides 100% recognition for the same database. Cox, Ghosn and Yianilos (1995) have recently introduced a *mixture-distance* technique which achieves a recognition rate of 95% using 95 test images and 685 training images (one image per person in each case). Each face is represented by 30 *manually* extracted distances.

Systems which employ precisely measured distances between features may be most useful for finding possible matches in a large mugshot database (a mugshot database typically contains side views where the performance of feature point methods is known to improve (Chellappa et al., 1995)). For other applications, automatic identification of these points would be required, and the resulting system would be dependent on the accuracy of the feature location algorithm. Current algorithms for automatic location of feature points do not consistently provide a high degree of accuracy (Sutherland, Renshaw and Denyer, 1992).

3.2 Eigenfaces

High-level recognition tasks are typically modeled with many stages of processing as in the Marr paradigm of progressing from images to surfaces to three-dimensional models to matched models (Marr, 1982). However, Turk and Pentland (1991) argue that it is likely that there is also a recognition process based on low-level, two-dimensional image processing. Their argument is based on the early development and extreme rapidity of face recognition in humans, and on physiological experiments in monkey cortex which claim to have isolated neurons that respond selectively to faces (Perret, Rolls and Caan, 1982). However, these experiments do not exclude the possibility of the sole operation of the Marr paradigm.

Turk and Pentland (1991) present a face recognition scheme in which face images are projected onto the principal components of the original set of training images. The resulting *eigenfaces* are classified by comparison with known individuals.

Turk and Pentland (1991) present results on a database of 16 subjects with various head orientation, scaling, and lighting. Their images appear identical otherwise with little variation in facial expression, facial details, pose, etc. For lighting, orientation, and scale variation their system achieves 96%, 85% and 64% correct classification respectively. Scale is renormalized to the eigenface size based on an estimate of the head size. The middle of the faces is accentuated, reducing any negative affect of changing hairstyle and backgrounds.

In Pentland et al. (1993; 1994) good results are reported on a large database (95% recognition of 200 people from a database of 3,000). It is difficult to draw broad conclusions as many of the images of the same people look very similar (in the sense that there is little difference in expression, hairstyle, etc.), and the database has accurate registration and alignment (Moghaddam and Pentland, 1994). In Moghaddam and Pentland (1994), very good results are reported with the US Army FERET database database – only one mistake was made in classifying 150 frontal view images. The system used extensive preprocessing for head location, feature detection, and normalization for the geometry of the face, translation, lighting, contrast, rotation, and scale.

In summary, it appears that eigenfaces is a fast, simple, and practical algorithm. However, it may be limited because optimal performance requires a high degree of correlation between the pixel intensities of the training and test images. This limitation has been addressed by using extensive preprocessing to normalize the images.

3.3 Template Matching

Template matching methods such as (Brunelli and Poggio, 1993) operate by performing direct correlation of image segments (e.g. by computing the Euclidean distance). Template matching is only effective when the query images have the same scale, orientation, and illumination as the training images (Cox et al., 1995).

3.4 Neural Network Approaches

Much of the present literature on face recognition with neural networks presents results with only a small number of classes (often below 20). For example, in (DeMers and Cottrell, 1993) the first 50 principal components of images are extracted and reduced to 5 dimensions using an autoassociative neural network. The resulting representation is classified using a standard multilayer perceptron. Good results are reported but the database is quite simple: the pictures are manually aligned and there is no lighting variation, rotation, or tilting. There are 20 people in the database.

3.5 The ORL Database and Application of HMM and Eigenfaces Methods

In (Samaria and Harter, 1994) a HMM-based approach is used for classification of the ORL database images. HMMs are typically used for the stochastic modeling of non-stationary vector time series. In this case, they are applied to images and a sampling window is passed over the image to generate a vector at each step. The best model resulted in a 13% error rate. Samaria also performed extensive tests using the popular eigenfaces algorithm (Turk and Pentland, 1991) on the ORL database and reported a best error rate of around 10% when the number of eigenfaces was between 175 and 199. Around 10% error was also observed in this work when implementing the eigenfaces algorithm. In (Samaria, 1994) Samaria extends the top-down HMM of (Samaria and Harter, 1994) with pseudo two-dimensional HMMs. The pseudo-2D HMMs are obtained

by linking one dimensional HMMs to form vertical superstates. The network is not fully connected in two dimensions (hence "pseudo"). The error rate reduces to 5% at the expense of high computational complexity – a single classification takes four minutes on a Sun Sparc II. Samaria notes that, although an increased recognition rate was achieved, the segmentation obtained with the pseudo two-dimensional HMMs appeared quite erratic. Samaria uses the same training and test set sizes as used later in this paper (200 training images and 200 test images with no overlap between the two sets). The 5% error rate is the best error rate previously reported for the ORL database that we are aware of.

4 System Components

4.1 Overview

The following sections introduce the techniques which form the components of the proposed system and describe the motivation for using them. Briefly, the investigations consider local image sampling and a technique for partial lighting invariance, a self-organizing map (SOM) for projection of the local image sample representation into a quantized lower dimensional space, the Karhunen-Loève (KL) transform for comparison with the self-organizing map, a convolutional network (CN) for partial translation and deformation invariance, and a multilayer perceptron (MLP) for comparison with the convolutional network.

4.2 Local Image Sampling

Two different methods of representing local image samples have been evaluated. In each method a window is scanned over the image as shown in figure 3.

1. The first method simply creates a vector from a local window on the image using the intensity values at each point in the window. Let x_{ij} be the intensity at the ith column, and the jth row of the given image. If the local window is a square of sides $2W + 1$ long, centered on x_{ij}, then the vector associated with this window is simply $[x_{i-W,j-W}, x_{i-W,j-W+1}, \ldots, x_{ij}, \ldots, x_{i+W,j+W-1}, x_{i+W,j+W}]$.
2. The second method creates a representation of the local sample by forming a vector out of a) the intensity of the center pixel x_{ij}, and b) the difference in intensity between the center pixel and all other pixels within the square window. The vector is given by $[x_{ij} - x_{i-W,j-W}, x_{ij} - x_{i-W,j-W+1}, \ldots, w_{ij}x_{ij}, \ldots, x_{ij} - x_{i+W,j+W-1}, x_{ij} - x_{i+W,j+W}]$. The resulting representation becomes partially invariant to variations in intensity of the complete sample. The degree of invariance can be modified by adjusting the weight w_{ij} (≥ 0) connected to the central intensity component.

4.3 The Self-Organizing Map

4.3.1 Overview

Maps are an important part of both natural and artificial neural information processing systems (Bauer and Pawelzik, 1992). Examples of maps in the nervous system are retinotopic maps in the visual cortex

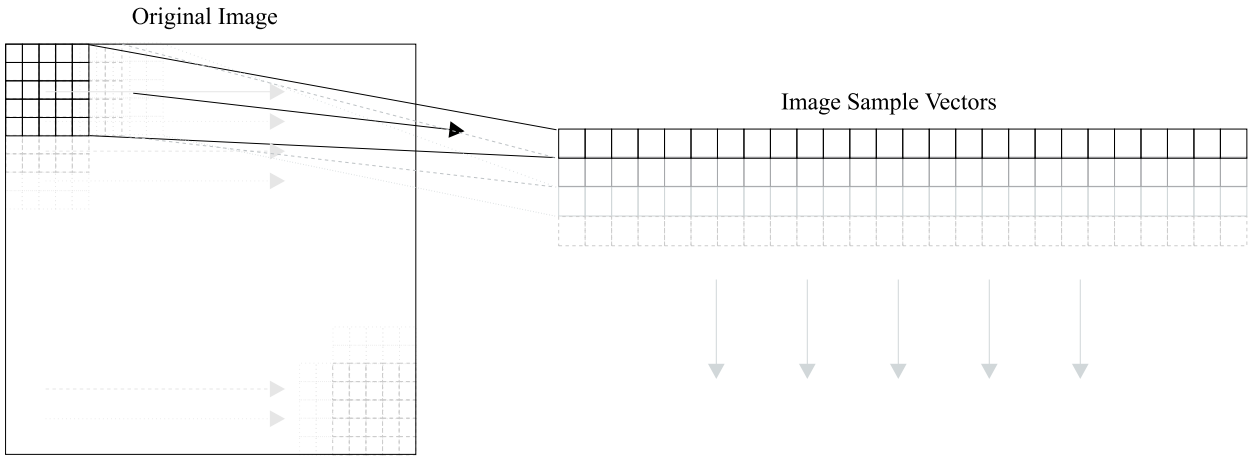

Figure 3. A depiction of the local image sampling process. A window is stepped over the image and a vector is created at each location.

(Obermayer, Blasdel and Schulten, 1991), tonotopic maps in the auditory cortex (Kita and Nishikawa, 1993), and maps from the skin onto the somatosensoric cortex (Obermayer, Ritter and Schulten, 1990). The self-organizing map, or SOM, introduced by Teuvo Kohonen (1990; 1995) is an unsupervised learning process which learns the distribution of a set of patterns without any class information. A pattern is projected from an input space to a position in the map – information is coded as the location of an activated node. The SOM is unlike most classification or clustering techniques in that it provides a topological ordering of the classes. Similarity in input patterns is preserved in the output of the process. The topological preservation of the SOM process makes it especially useful in the classification of data which includes a large number of classes. In the local image sample classification, for example, there may be a very large number of classes in which the transition from one class to the next is practically continuous (making it difficult to define hard class boundaries).

4.3.2 Algorithm

We give a brief description of the SOM algorithm, for more details see (Kohonen, 1995). The SOM defines a mapping from an input space $\mathcal{R}^n$ onto a topologically ordered set of nodes, usually in a lower dimensional space. An example of a two-dimensional SOM is shown in figure 4. A reference vector in the input space, $m_i \equiv [\mu_{i1}, \mu_{i2}, ..., \mu_{in}]^T \in \mathcal{R}^n$, is assigned to each node in the SOM. During training, each input, x, is compared to all of the m_i, obtaining the location of the closest match ($||x - m_c|| = \min_i\{||x - m_i||\}$). The input point is mapped to this location in the SOM. Nodes in the SOM are updated according to:

$$m_i(t+1) = m_i(t) + h_{ci}(t)[x(t) - m_i(t)] \tag{1}$$

where t is the time during learning and $h_{ci}(t)$ is the *neighborhood function*, a smoothing kernel which is maximum at m_c. Usually, $h_{ci}(t) = h(||r_c - r_i||, t)$, where r_c and r_i represent the location of the nodes in the SOM output space. r_c is the node with the closest weight vector to the input sample and r_i ranges over all nodes. $h_{ci}(t)$ approaches 0 as $||r_c - r_i||$ increases and also as t approaches ∞. A widely applied neighborhood function is:

$$h_{ci} = \alpha(t) \exp\left(-\frac{||r_c - r_i||^2}{2\sigma^2(t)}\right) \tag{2}$$

where $\alpha(t)$ is a scalar valued learning rate and $\sigma(t)$ defines the width of the kernel. They are generally both monotonically decreasing with time. The use of the neighborhood function means that nodes which are topographically close in the SOM structure activate each other to learn something from the same input x. A relaxation or smoothing effect results which leads to a global ordering of the map. Note that $\sigma(t)$ should not be reduced too far as the map will lose its topographical order if neighboring nodes are not updated along with the closest node. The SOM can be considered a non-linear projection of the probability density, $p(x)$ (Kohonen, 1995).

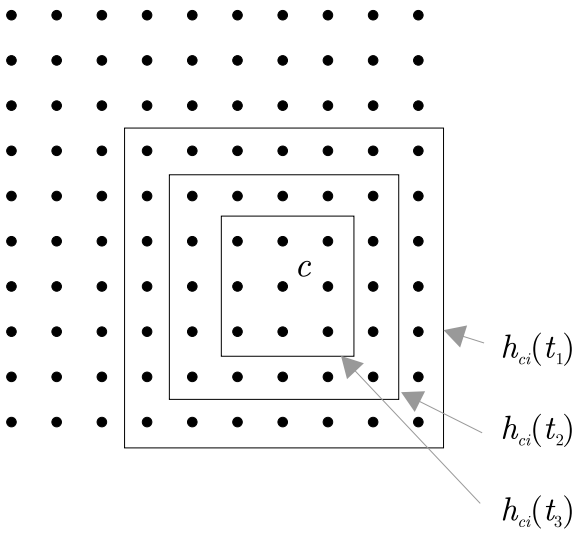

Figure 4. A two-dimensional SOM showing a square neighborhood function which starts as $h_{ci}(t_1)$ and reduces in size to $h_{ci}(t_3)$ over time.

4.3.3 Improving the Basic SOM

The original self-organizing map is computationally expensive because:

1. In the early stages of learning, many nodes are adjusted in a correlated manner. Luttrell (1989) proposed a method, which is used here, where learning starts in a small network, and the network is doubled in size periodically during training. When doubling, new nodes are inserted between the current nodes. The weights of the new nodes are set equal to the average of the weights of the immediately neighboring nodes.

2. Each learning pass requires computation of the distance of the current sample to all nodes in the network, which is $O(N)$. However, this may be reduced to $O(\log N)$ using a hierarchy of networks which is created using the above node doubling strategy[4]. This has not been used for the results reported here.

[4]This assumes that the topological order is optimal prior to each doubling step.

4.4 Karhunen-Loève Transform

The optimal linear method (in the least mean squared error sense) for reducing redundancy in a dataset is the Karhunen-Loève (KL) transform or eigenvector expansion via Principle Components Analysis (PCA) (Fukunaga, 1990). The basic idea behind the KL transform is to transform possibly correlated variables in a data set into uncorrelated variables. The transformed variables will be ordered so that the first one describes most of the variation of the original data set. The second will try to describe the remaining part of variation under the constraint that it should be uncorrelated with the first variable. This continues until all the variation is described by the new transformed variables, which are called principal components. PCA appears to be involved in some biological processes, e.g. edge segments are principal components and edge segments are among the first features extracted in the primary visual cortex (Hubel and Wiesel, 1962).

Mathematically, the KL transform can be written as (Dony and Haykin, 1995):

$$\mathbf{y} = \mathbf{W}\mathbf{x} \tag{3}$$

where $\mathbf{x}$ is an N dimensional input vector, $\mathbf{y}$ is an M dimensional output vector ($M \leq N$), and $\mathbf{W}$ is an $M \times N$ dimensional transformation matrix. The transformation matrix, $\mathbf{W}$, consists of M rows of the eigenvectors which correspond to the M largest eigenvalues of the sample autocovariance matrix, Σ (Dony and Haykin, 1995):

$$\Sigma = \left\langle \mathbf{x}\mathbf{x}^T \right\rangle \tag{4}$$

where $<>$ represents expectation.

The KL transform is used here for comparison with the SOM in the dimensionality reduction of the local image samples. The KL transform is also used in eigenfaces, however in that case it is used on the entire images whereas it is only used on small local image samples in this work.

4.5 Convolutional Networks

The problem of face recognition from 2D images is typically very ill-posed, i.e. there are many models which fit the training points well but do not generalize well to unseen images. In other words, there are not enough training points in the space created by the input images in order to allow accurate estimation of class probabilities throughout the input space. Additionally, for MLP networks with the 2D images as input, there is no invariance to translation or local deformation of the images (Le Cun and Bengio, 1995).

Convolutional networks (CN) incorporate constraints and achieve some degree of shift and deformation invariance using three ideas: local receptive fields, shared weights, and spatial subsampling. The use of shared weights also reduces the number of parameters in the system aiding generalization. Convolutional networks have been successfully applied to character recognition (Le Cun, 1989; Le Cun, Boser, Denker, Henderson, Howard, Hubbard and Jackel, 1990; Bottou, Cortes, Denker, Drucker, Guyon, Jackel, Le Cun, Muller, Sackinger, Simard and Vapnik, 1994; Bengio, Le Cun and Henderson, 1994; Le Cun and Bengio, 1995).

A typical convolutional network is shown in figure 5 (Le Cun, Boser, Denker, Henderson, Howard, Hubbard and Jackel, 1990). The network consists of a set of layers each of which contains one or more planes. Images which are approximately centered and normalized enter at the input layer. Each unit in a plane receives input from a small neighborhood in the planes of the previous layer. The idea of connecting units to

local receptive fields dates back to the 1960s with the perceptron and Hubel and Wiesel's (1962) discovery of locally sensitive, orientation-selective neurons in the visual system of a cat (Le Cun and Bengio, 1995). The weights forming the receptive field for a plane are forced to be equal at all points in the plane. Each plane can be considered as a feature map which has a fixed feature detector that is convolved with a local window which is scanned over the planes in the previous layer. Multiple planes are usually used in each layer so that multiple features can be detected. These layers are called convolutional layers. Once a feature has been detected, its exact location is less important. Hence, the convolutional layers are typically followed by another layer which does a local averaging and subsampling operation (e.g. for a subsampling factor of 2: $y_{ij} = (x_{2i,2j} + x_{2i+1,2j} + x_{2i,2j+1} + x_{2i+1,2j+1})/4$ where y_{ij} is the output of a subsampling plane at position i, j and x_{ij} is the output of the same plane in the previous layer). The network is trained with the usual backpropagation gradient descent procedure (Haykin, 1994).

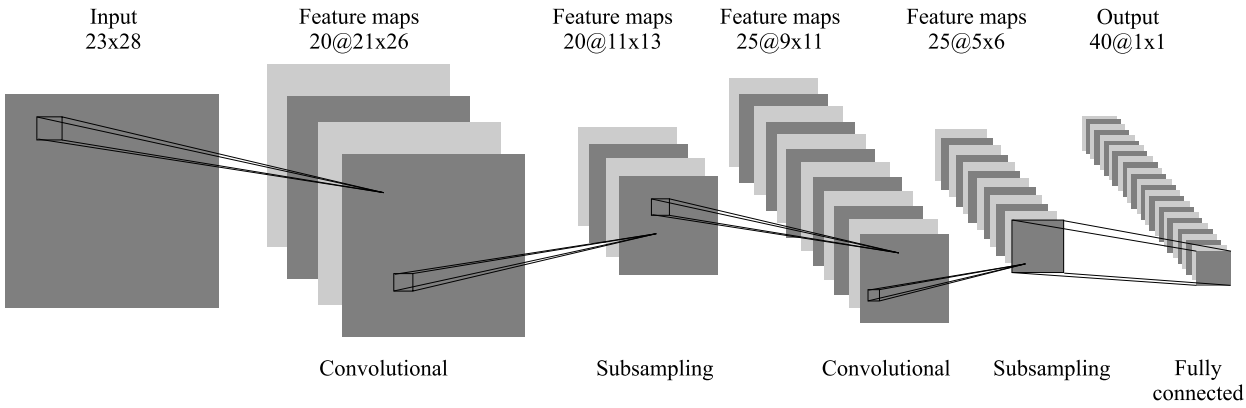

Figure 5. A typical convolutional network.

A connection strategy can be used to reduce the number of weights in the network. For example, with reference to figure 5, Le Cun, Boser, Denker, Henderson, Howard, Hubbard and Jackel (1990) connect the feature maps in the second convolutional layer only to 1 or 2 of the maps in the first subsampling layer (the connection strategy was chosen manually). This can reduce training time and improve performance (Le Cun, Boser, Denker, Henderson, Howard, Hubbard and Jackel, 1990).

Convolutional networks are similar to the Neocognitron (Fukushima, 1980; Fukushima, Miyake and Ito, 1983; Hummel, 1995) which is a neural network model of deformation-resistant pattern recognition. The Neocognitron is similar to the convolutional neural network. Alternating S and C-cell layers in the Neocognitron correspond to the convolutional and blurring layers in the convolutional network. However, in the Neocognitron, the C-cell layers respond to the most active input S-cell as opposed to performing an averaging operation. The Neocognitron can be trained using either unsupervised or supervised approaches (Fukushima, 1995).

5 System Details

The system used for face recognition in this paper is a combination of the preceding parts – a high-level block diagram is shown in figure 6 and figure 7 shows a breakdown of the various subsystems that are experimented with or discussed.

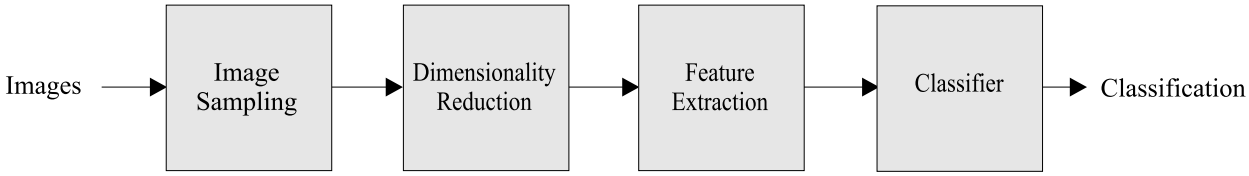

Figure 6. A high-level block diagram of the system used for face recognition.

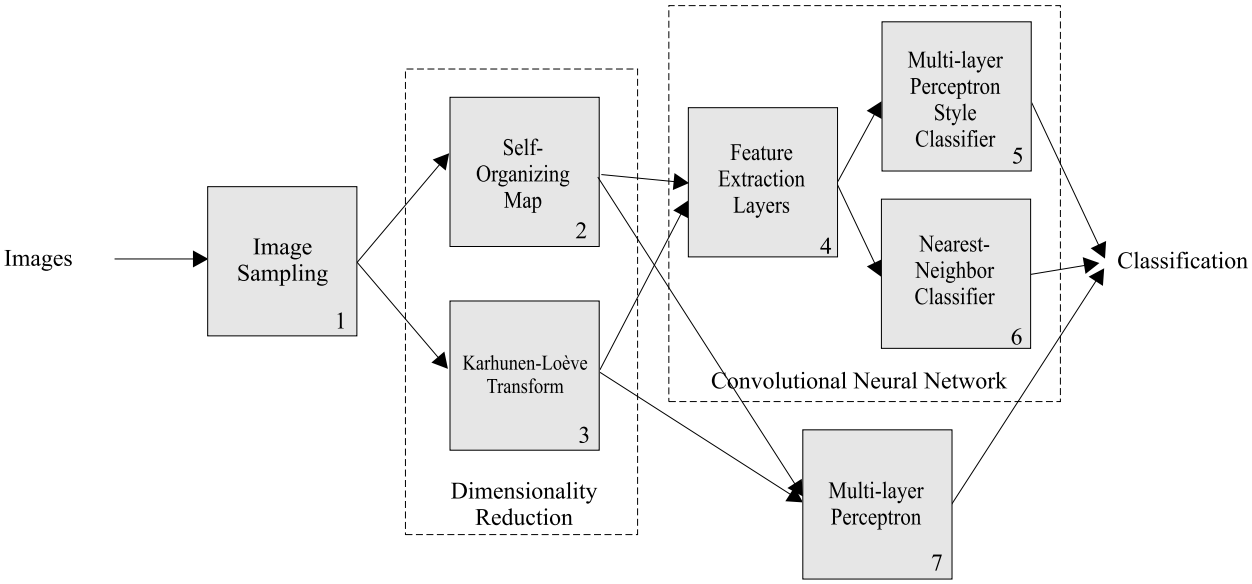

Figure 7. A diagram of the system used for face recognition showing alternative methods which are considered in this paper. The top "multilayer perceptron style classifier" (5) represents the final MLP style fully connected layer of the convolutional network (the CN is a constrained MLP, however the final layer has no constraints). This decomposition of the convolutional network is shown in order to highlight the possibility of replacing the final layer (or layers) with a different type of classifier. The nearest-neighbor style classifier is potentially interesting because it may make it possible to add new classes with minimal extra training time. The bottom "multilayer perceptron" (7) shows that the entire convolutional network can be replaced with a multilayer perceptron. Results are presented with either a self-organizing map (2) or the Karhunen-Loève transform (3) for dimensionality reduction, and either a convolutional neural network (4,5) or a multilayer perceptron (7) for classification.

The system works as follows (complete details of dimensions etc. are given later):

1. For the images in the training set, a fixed size window (e.g. 5×5) is stepped over the entire image as shown in figure 3 and local image samples are extracted at each step. At each step the window is moved by 4 pixels.

2. A self-organizing map (e.g. with three dimensions and five nodes per dimension, $5^3 = 125$ total nodes) is trained on the vectors from the previous stage. The SOM quantizes the 25-dimensional input vectors into 125 topologically ordered values. The three dimensions of the SOM can be thought of as three features. The SOM is used primarily as a dimensionality reduction technique and it is therefore of interest to compare the SOM with a more traditional technique. Hence, experiments were performed with the SOM replaced by the Karhunen-Loève transform. In this case, the KL transform projects the vectors in the 25-dimensional space into a 3-dimensional space.

3. The same window as in the first step is stepped over all of the images in the training and test sets. The local image samples are passed through the SOM at each step, thereby creating new training and test sets in the output space of the self-organizing map. (Each input image is now represented by 3 maps, each of which corresponds to a dimension in the SOM. The size of these maps is equal to the size of the input image (92×112) divided by the step size (for a step size of 4, the maps are 23×28).)

4. A convolutional neural network is trained on the newly created training set. Training a standard MLP was also investigated for comparison.

5.1 Simulation Details

Details of the best performing system from all experiments are given in this section.

For the SOM, training was split into two phases as recommended by Kohonen (1995) – an ordering phase, and a fine-adjustment phase. 100,000 updates were performed in the first phase, and 50,000 in the second. In the first phase, the neighborhood radius started at two-thirds of the size of the map and was reduced linearly to 1. The learning rate during this phase was: $0.7 \times (1 - n/N)$ where n is the current update number, and N is the total number of updates. In the second phase, the neighborhood radius started at 2 and was reduced to 1. The learning rate during this phase was: $0.02 \times (1 - n/N)$.

The convolutional network contained five layers excluding the input layer. A confidence measure was calculated for each classification: $y_m(y_m - y_{2m})$ where y_m is the maximum output, and y_{2m} is the second maximum output (for outputs which have been transformed using the *softmax* transformation: $y_i = \frac{\exp(u_i)}{\sum_{j=1}^{k} \exp(u_j)}$ where u_i are the original outputs, y_i are the transformed outputs, and k is the number of outputs). The number of planes in each layer, the dimensions of the planes, and the dimensions of the receptive fields are shown in table 1. The network was trained with backpropagation (Haykin, 1994) for a total of 20,000 updates. Weights in the network were updated after each pattern presentation, as opposed to batch update where weights are only updated once per pass through the training set. All inputs were normalized to lie in the range minus one to one. All nodes included a bias input which was part of the optimization process. The best of 10 random weight sets was chosen for the initial parameters of the network by evaluating the performance on the training set. Weights were initialized on a node by node basis as uniformly distributed random numbers in the range $(-2.4/F_i, 2.4/F_i)$ where F_i is the fan-in of neuron i (Haykin, 1994). Target outputs were -0.8 and 0.8 using the $\tanh$ output activation function[5]. The quadratic cost function was

[5]This helps avoid saturating the sigmoid function. If targets were set to the asymptotes of the sigmoid this would tend to: a) drive the weights to infinity, b) cause outlier data to produce very large gradients due to the large weights, and c) produce binary outputs even when incorrect – leading to decreased reliability of the confidence measure.

used. A search then converge learning rate schedule was used[6]: $\eta = \frac{\eta_0}{\frac{n}{N/2} + \frac{c_1}{\max\left(1, (c_1 - \frac{\max(0, c_1(n - c_2 N))}{(1-c_2)N}\right)}}$ where η = learning rate, η_0 = initial learning rate = 0.1, N = total training epochs, n = current training epoch, $c_1 = 50$, $c_2 = 0.65$. The schedule is shown in figure 8. Total training time was around four hours on an SGI Indy 100Mhz MIPS R4400 system.

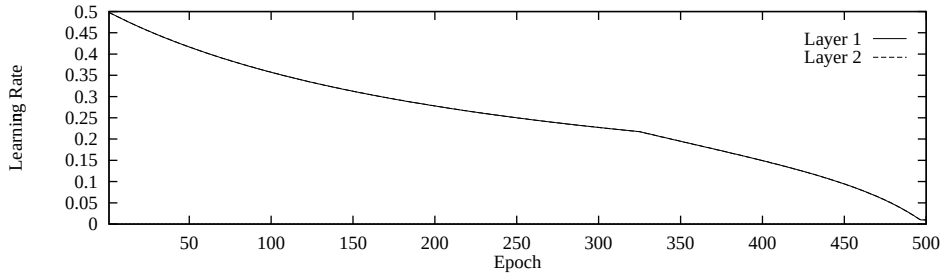

Figure 8. The learning rate as a function of the epoch number.

Layer	Type	Units	x	y	Receptive field x	Receptive field y	Connection Percentage
1	Convolutional	20	21	26	3	3	100
2	Subsampling	20	11	13	2	2	–
3	Convolutional	25	9	11	3	3	30
4	Subsampling	25	5	6	2	2	–
5	Fully connected	40	1	1	5	6	100

Table 1. Dimensions for the convolutional network. The connection percentage refers to the percentage of nodes in the previous layer which each node in the current layer is connected to – a value less than 100% reduces the total number of weights in the network and may improve generalization. The connection strategy used here is similar to that used by Le Cun et al. (1990) for character recognition. However, as opposed to the manual connection strategy used by Le Cun et al., the connections between layers 2 and 3 are chosen randomly. As an example of how the precise connections can be determined from the table – the size of the first layer planes (21x26) is equal to the total number of ways of positioning a 3x3 receptive field on the input layer planes (23x28).

6 Experimental Results

Various experiments were performed and the results are presented in this section. Except where noted, all experiments were performed with 5 training images and 5 test images per person for a total of 200 training

[6]Relatively high learning rates are typically used in order to help avoid slow convergence and local minima. However, a constant learning rate results in significant parameter and performance fluctuation during the entire training cycle such that the performance of the network can alter significantly from the beginning to the end of the final epoch. Moody and Darkin have proposed "search then converge" learning rate schedules. We have found that these schedules still result in considerable parameter fluctuation and hence we have added another term to further reduce the learning rate over the final epochs. We have found the use of learning rate schedules to improve performance considerably.

images and 200 test images. There was no overlap between the training and test sets. A system which guesses the correct answer would be right one out of forty times, giving an error rate of 97.5%. For the following sets of experiments, only one parameter is varied in each case. The error bars shown in the graphs represent plus or minus one standard deviation of the distribution of results from a number of simulations[7]. Ideally, it would be desirable to perform more simulations per result, however, the computational resources available were limited. The constants used in each set of experiments were: number of classes: 40, dimensionality reduction method: SOM, dimensions in the SOM: 3, number of nodes per SOM dimension: 5, local image sample extraction: original intensity values, training images per class: 5. Note that the constants in each set of experiments may not give the best possible performance as the current best performing system was only obtained as a result of these experiments. The experiments are as follows:

1. *Variation of the number of output classes* – table 2 and figure 9 show the error rate of the system as the number of classes is varied from 10 to 20 to 40. No attempt was made to optimize the system for the smaller numbers of classes. As expected, performance improves with fewer classes to discriminate between.

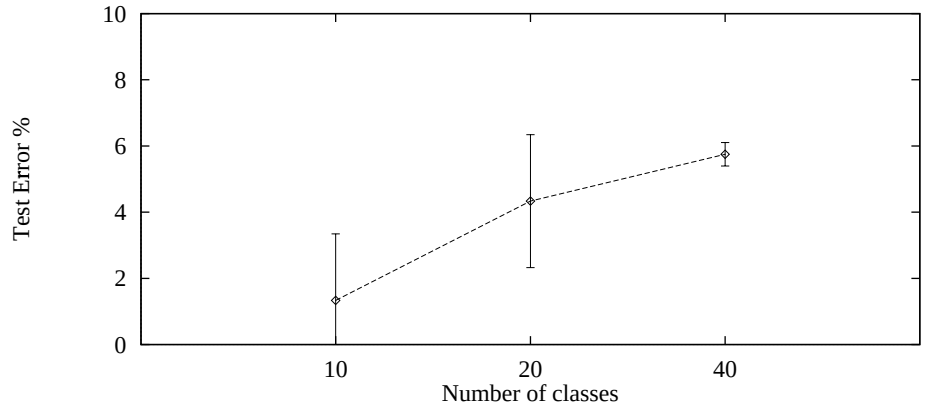

Figure 9. The error rate as a function of the number of classes. The network was not modified from that used for the 40 class case. The error bars represent plus and minus one standard deviation.

Number of classes	10	20	40
Error rate	1.33%	4.33%	5.75%

Table 2. Error rate of the face recognition system with varying number of classes (subjects). Each result is the average of three simulations.

2. *Variation of the dimensionality of the SOM* – table 3 and figure 10 show the error rate of the system as the dimension of the self-organizing map is varied from 1 to 4. The best performing value is three dimensions.

[7]Multiple simulations were performed in each experiment where the selection of the training and test images (out of a total of $10!/5! = 30240$ possibilities) and the random seed used to initialize the weights in the SOM and the convolutional neural network were varied.

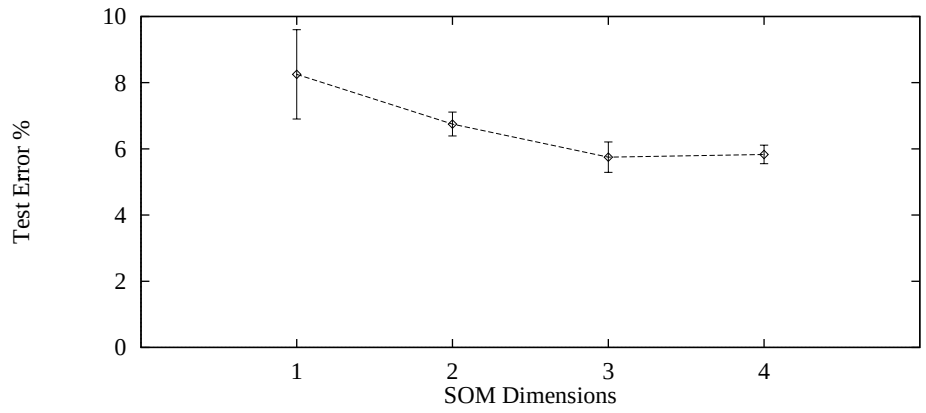

Figure 10. The error rate as a function of the number of dimensions in the SOM.

SOM Dimension	1	2	3	4
Error rate	8.25%	6.75%	5.75%	5.83%

Table 3. Error rate of the face recognition system with varying number of dimensions in the self-organizing map. Each result given is the average of three simulations.

3. *Variation of the quantization level of the SOM* – table 4 and figure 11 show the error rate of the system as the size of the self-organizing map is varied from 4 to 10 nodes per dimension. The SOM has three dimensions in each case. The best error rate occurs for 8 or 9 nodes per dimension. This is also the best error rate of all experiments.

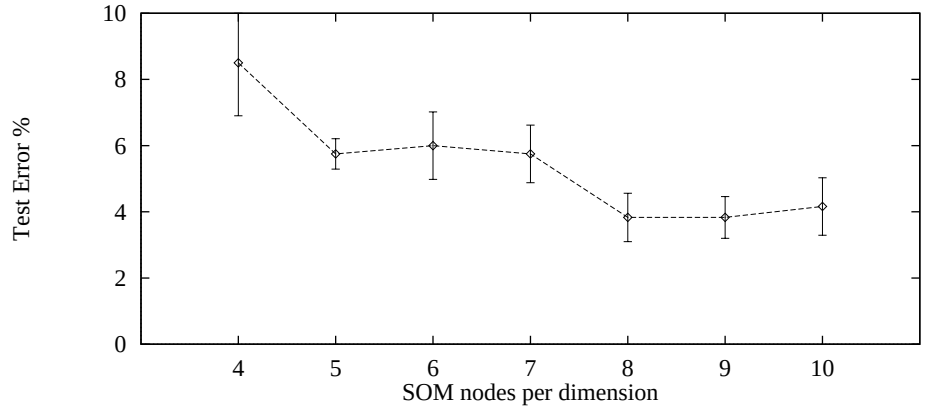

Figure 11. The error rate as a function of the number of nodes per dimension in the SOM.

SOM Size	4	5	6	7	8	9	10
Error rate	8.5%	5.75%	6.0%	5.75%	3.83%	3.83%	4.16%

Table 4. Error rate of the face recognition system with varying number of nodes per dimension in the self-organizing map. Each result given is the average of three simulations.

4. *Variation of the local image sample extraction algorithm* – table 5 shows the result of using the two local image sample representations described earlier. Using the original intensity values was found to give the best performance. Altering the weight assigned to the central intensity value in the alternative representation was investigated without success.

Input type	Pixel intensities	Differences w/base intensity
Error rate	5.75%	7.17%

Table 5. Error rate of the face recognition system with varying image sample representation. Each result is the average of three simulations.

5. *Substituting the SOM with the KL transform* – table 6 shows the results of replacing the self-organizing map with the Karhunen-Loève transform. Using the first one, two, or three eigenvectors was investigated. Surprisingly, the system performed best with only one eigenvector. The best SOM parameters that were tested produced slightly better performance. The quantization inherent in the SOM could provide a degree of invariance to minor image sample differences and quantization of the PCA projections may improve performance.

Dimensionality reduction	KL	SOM
Error rate	5.33%	3.83%

Table 6. Error rate of the face recognition system with KL and SOM feature extraction mechanisms. Each result is the average of three simulations.

6. *Replacing the CN with an MLP* – table 7 shows the results of replacing the convolutional network with a multilayer perceptron. Performance is very poor. This result was expected because the multilayer perceptron does not have the inbuilt invariance to minor translation and local deformation which is created in the convolutional network using the local receptive fields, shared weights, and spatial subsampling. As an example, consider when a feature is shifted in a test image in comparison with the training image(s) for the individual. The MLP is expected to have difficulty recognizing a feature which has been shifted in comparison to the training images because the weights connected to the new location were not trained for the feature.

   The MLP contained one hidden layer. The following hidden layer sizes were tested: 20, 50, 100, 200, and 500. The best performance was obtained with 200 hidden nodes and a training time of 2.5 days (on an SGI R4400 150Mhz machine). The learning rate schedule and initial learning rate were the same as for the original network. Note that the best performing KL parameters were used while the best performing SOM parameters were not. Note that it may be considered fairer to compare against an MLP

	KL	SOM
MLP	41.2%	39.6%
CN	5.33%	3.83%

Table 7. Error rate comparison of the various feature extraction and classification methods. Each result is the average of three simulations.

with multiple hidden layers (Haykin, 1996), however selection of the appropriate number of nodes in each layer and training is difficult (e.g. trying a network with two hidden layers containing 100 and 50 nodes respectively resulted in an error rate of 90%).

7. *The tradeoff between rejection threshold and recognition accuracy* – Figure 12 shows a histogram of the confidence of the system for the cases when the classifier is correct and when it is wrong for one of the best performing systems. From this graph it is expected that the classification performance will increase significantly if cases below a certain confidence threshold are rejected. Figure 13 shows the system performance as the rejection threshold is increased. It can be seen that by rejecting examples with low confidence it is possible to significantly increase the classification performance of the system. For a system which used a video camera to take a number of pictures over a short period, it may be possible to obtain a high level of performance with an appropriate rejection threshold.

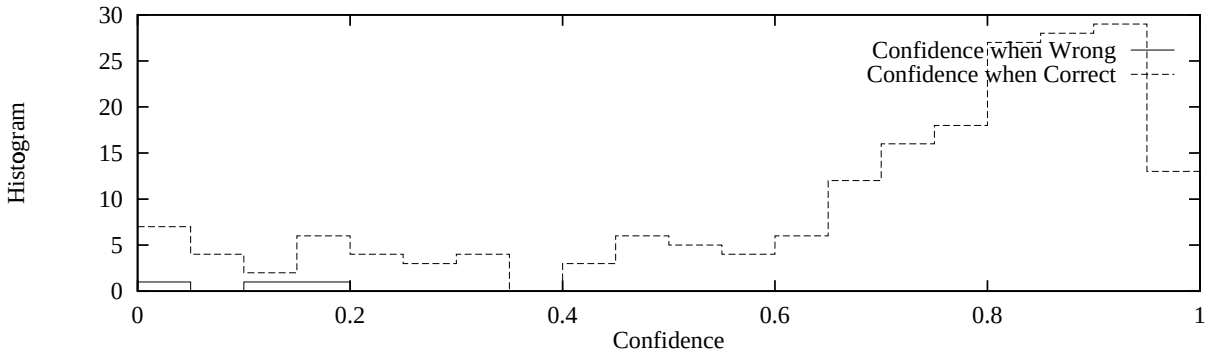

Figure 12. A histogram depicting the confidence of the classifier when it turns out to be correct, and the confidence when it is wrong. The graph suggests that it is possible to improve classification performance considerably by rejecting cases where the classifier has a low confidence (because the cases where the classifier is wrong have low confidence).

8. *Comparison with other known results on the same database* – Table 8 shows a summary of the performance of the systems for which results are available using the ORL database. A SOM quantization level of 8 is used in this case. The SOM+CN system presented here is the best performing system[8] and performs recognition roughly two orders of magnitude faster than the second best performing system – the pseudo 2D-HMMs of Samaria. Figure 14 shows the images which were incorrectly classified for one of the best performing systems.

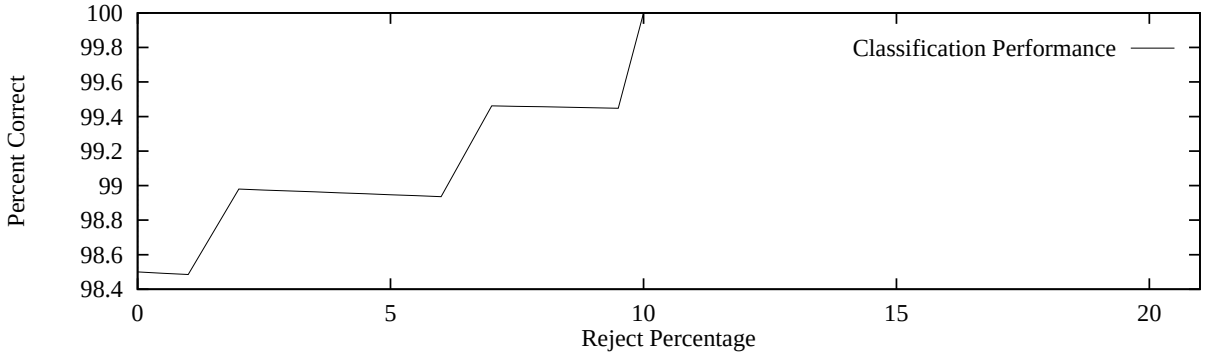

Figure 13. The test set classification performance as a function of the percentage of samples rejected. Classification performance can be improved significantly by rejecting cases with low confidence.

Figure 14. Test images. The images with a thick white border were incorrectly classified by one of the best performing systems.

9. *Variation of the number of training images per person.* Table 9 and figure 15 show the results of varying the number of images per class used in the training set from 1 to 5 for PCA+CN, SOM+CN and also for the eigenfaces algorithm. Two versions of the eigenfaces algorithm were implemented. The first version creates vectors for each class in the training set by averaging the results of the eigenface representation over all images for the same person. This corresponds to the algorithm as described by Turk and Pentland (1991). However, it was found that using separate training vectors for each training image resulted in better performance. Using between 40 and 100 eigenfaces resulted in similar performance. It can be observed that the PCA+CN and SOM+CN methods are both superior to the eigenfaces technique even

[8]The 3.83% error rate reported is an average of multiple simulations – individual simulations have given error rates as low as 1.5%.

System	Error rate	Classification time
Top-down HMM	13%	n/a
Eigenfaces	10.5%	n/a
Pseudo 2D-HMM	5%	240 seconds[1]
SOM+CN	3.8%	< 0.5 seconds[2]

Table 8. Error rate of the various systems. [1] On a Sun Sparc II. [2] On an SGI Indy MIPS R4400 100Mhz system. According to the SPECint92 and SPECfp92 ratings at `http://hpwww.epfl.ch/bench/SPEC.html` the SGI machine is approximately 3 times faster than the Sun Sparc II, making the SOM+CN approximately 160 times faster than the Pseudo 2D-HMMs for classification.

when there is only one training image per person. The SOM+CN method consistently performs better than the PCA+CN method.

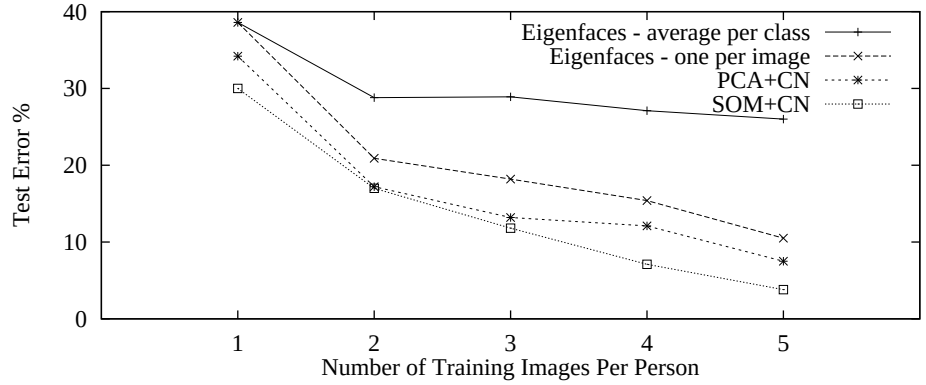

Figure 15. The error rate of the face recognition system and eigenfaces as the number of images per person is varied. Averaged over two different selections of the training and test sets.

Images per person	1	2	3	4	5
Eigenfaces – average per class	38.6	28.8	28.9	27.1	26
Eigenfaces – one per image	38.6	20.9	18.2	15.4	10.5
PCA+CN	34.2	17.2	13.2	12.1	7.5
SOM+CN	30.0	17.0	11.8	7.1	3.8

Table 9. The error rate for the eigenfaces algorithm and the SOM+CN as the size of the training set is varied from 1 to 5 images per person. Averaged over two different selections of the training and test sets. 100 eigenfaces were used in the eigenfaces algorithm. The architecture of the CN was as given in table 1. The SOM dimensionality was 3, the number of nodes per dimension was 8, and the PCA reduced the data to 3 dimensions.

7 Discussion

Convolutional networks have traditionally been used on raw images without any preprocessing. Without the preprocessing used in this work (the local image sampling and SOM or KL transform stages), the resulting convolutional networks are larger, more computationally intensive, and have not performed as well in our experiments (e.g. using no preprocessing and the same CN architecture except initial receptive fields of 8×8 resulted in approximately two times greater error (for the case of five images per person)).

Figure 16 shows the randomly chosen initial local image samples corresponding to each node in a two-dimensional SOM, and the final samples which the SOM converges to. Looking across the rows and columns it can be seen that the quantized samples represent smoothly changing shading patterns. This is the initial representation from which successively higher level features are extracted using the convolutional network. Figure 17 shows the activation of the nodes in a sample convolutional network for a particular test image.

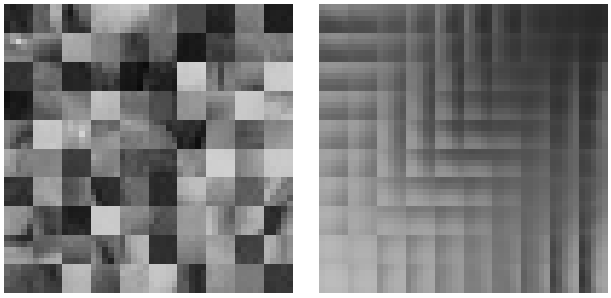

Figure 16. SOM image samples before training (a random set of image samples) and after training.

Figure 18 shows the results of sensitivity analysis in order to determine which parts of the input image are most important for classification. Using the method of Baluja and Pomerleau as described in Rowley et al. (1995), each of the input planes to the convolutional network was divided into 2×2 segments (the input planes are 23×28). Each of 168 (12×14) segments was replaced with random noise, one segment at a time. The test performance was calculated at each step. The error of the network when replacing parts of the input with random noise gives an indication of how important each part of the image is for the classification task. From the figure it can be observed that the eyes, nose, mouth, chin, and hair regions are all important to the classification task.

8 Computational Complexity

The SOM training process is relatively slow. This may not be a major drawback of the approach however, because it may be possible to extend the system to cover new classes without retraining the SOM. All that

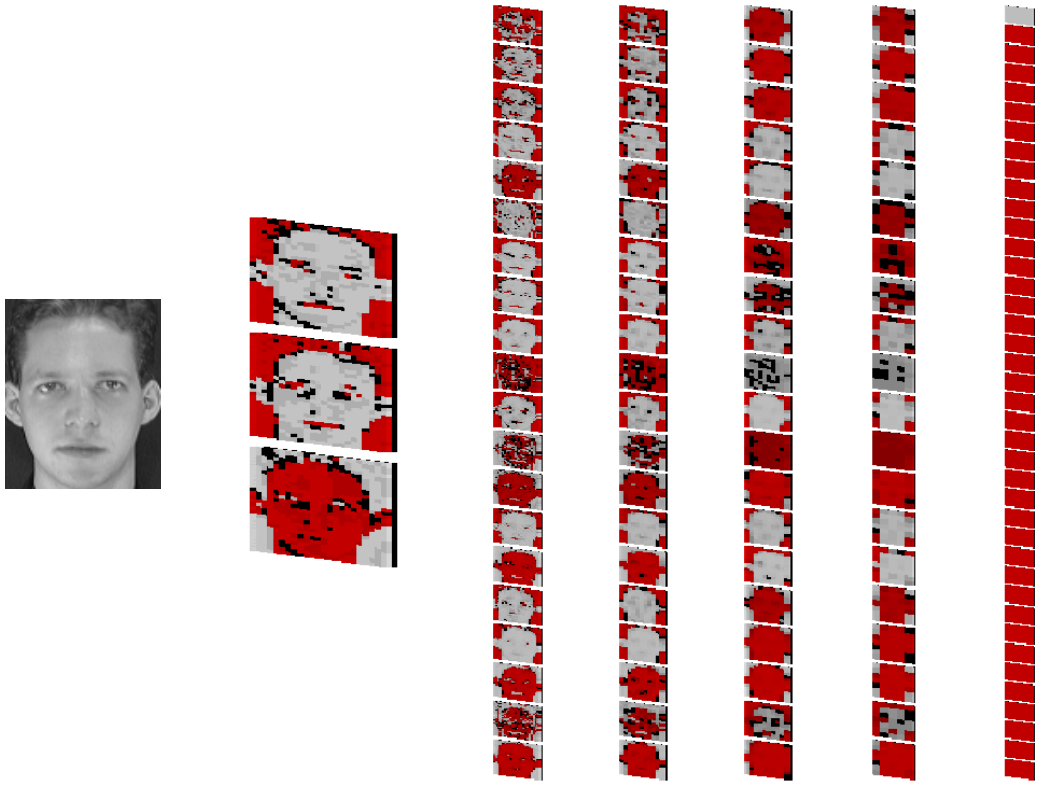

Figure 17. A depiction of the node maps in a sample convolutional network showing the activation values for a particular test image. The input image is shown on the left. In this case the image is correctly classified with only one activated output node (the top node). From left to right after the input image, the layers are: the input layer, convolutional layer 1, subsampling layer 1, convolutional layer 2, subsampling layer 2, and the output layer. The three planes in the input layer correspond to the three dimensions of the SOM.

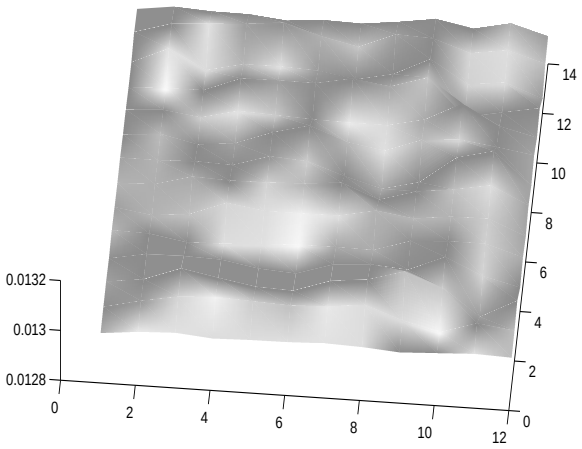

Figure 18. Sensitivity to various parts of the input image. It can be observed that the eyes, mouth, nose, chin, and hair regions are all important for the classification. The z axis corresponds to the mean squared error rather than the classification error (the mean squared error is preferable because it varies in a smoother fashion as the input images are perturbed). The image orientation corresponds to upright face images.

is required is that the image samples originally used to train the SOM are sufficiently representative of the image samples used in new images. For the experiments reported here, the quantized output of the SOM is very similar if it is trained with only 20 classes instead of 40. In addition, the Karhunen-Loève transform can be used in place of the SOM with a relatively minor impact on system performance.

The convolutional network training process is also relatively slow, how significant is this? The convolutional network extracts features from the image. It is possible to use fixed feature extraction. Consider if the convolutional network is divided into two parts: the initial feature extraction layers and the final feature extraction and classification layers. Given a well chosen sample of the complete distribution of faces to recognize, the features extracted from the first section may also be useful for the classification of new classes. These features could then be considered fixed features and the first part of the network may not need to be retrained when adding new classes. The point at which the convolutional network is broken into two would depend on how well the features at each stage are useful for the classification of new classes (the larger features in the final layers are less likely to be a good basis for classification of new examples). It may be possible to replace the second part with another type of classifier – e.g. a nearest-neighbor classifier. In this case the time required for retraining the system when adding new classes would be minimal (the extracted feature vectors are simply stored for the training images).

The following variables will be used to give an idea of the computational complexity of each part of the system:

N_c	The number of classes
N_s	The number of nodes in the self-organizing map
N_{w1}	The number of weights in the convolutional network
N_{w2}	The number of weights in the classifier
N_{tr}	The number of training examples
N_n	The number of nodes in the neighborhood function
N_{nn1}	The total number of next nodes used to backpropagate the error in the CN
N_{nn2}	The total number of next nodes used to backpropagate the error in the MLP classifier
N_{od}	The output dimension of the KL projection
N_{id}	The input dimension of the KL projection
N_{is}	The number of local image samples per image
N_{sm}	The number of training samples for the SOM or the KL projection

Tables 10 and 11 show the approximate complexity of the various parts of the system during training and classification. The complexity is shown for both the SOM and KL alternatives for dimensionality reduction and for both the neural network (MLP) and a nearest-neighbor classifier (as the last part of the convolutional network – not as a complete replacement, i.e. this is not the same as the earlier multilayer perceptron experiments). Note that the constant associated with the log factors may increase exponentially in the worst case (cf. neighbor searching in high dimensional spaces (Arya and Mount, 1993)). The approximations aim to show how the computational complexity scales according to the number of classes, e.g. for the training complexity of the MLP classifier, although $N_{w2} + N_{nn2}$ may be larger than N_c, both N_{w2} and N_{nn2} scale roughly according to N_c.

With reference to table 11, consider, for example, the main SOM+CN architecture in recognition mode. The complexity of the SOM module is independent of the number of classes. The complexity of the CN scales according to the number of weights in the network. When the number of feature maps in the internal layers is constant, the number of weights scales roughly according to the number of output classes (the number of weights in the output layer dominates the weights in the initial layers).

Section	Training complexity
KL	$O((2 + N_{id}^2)N_{sm} + 3N_{od}^3) \approx O(N_{id}^2 + N_{od}^3)$
SOM	$O(k_1 N_{SM} N_n k_2 \log N_s) \approx O(N_{sm} N_n \log N_s)$ (N_n varies)
CN	$O(k_3 N_{tr}(N_{w1} + N_{nn1})) \approx O(N_{tr} N_{w1})$
MLP Classifier	$O(k_3 N_{tr}(N_{w2} + N_{nn2})) \approx O(N_{tr} N_c)$
NN Classifier	$O(N_{tr})$

Table 10. Training complexity. k_1 and k_3 represent the number of times the training set is presented to the network for the SOM and the CN respectively.

Section	Classification complexity
KL	$O(N_{is} N_{id} N_{od})$
SOM	$O(N_{is} k_1 \log N_s) \approx O(N_{is} \log N_s)$
CN	$O(k_2 N_{w1}) \approx O(N_{w1})$
MLP Classifier	$O(N_{w2}) \approx O(N_c)$
NN Classifier	$O(k_4 \log N_{tr}) \approx O(\log N_c)$

Table 11. Classification complexity. k_2 represents the degree of shared weight replication.

In terms of computation time, the requirements of real-time tasks varies. The system presented should be suitable for a number of real-time applications. The system is capable of performing a classification in less than half a second for 40 classes. This speed is sufficient for tasks such as access control and room monitoring when using 40 classes. It is expected that an optimized version could be significantly faster.

9 Further Research

The following topics for further research could improve performance:

1. More careful selection of the convolutional network architecture, e.g. by using the Optimal Brain Damage algorithm (Le Cun, Denker and Solla, 1990) as used by Le Cun et al. (1990) to improve generalization and speed up handwritten digit recognition.

2. More precise normalization of the images to account for translation, rotation, and scale changes. Any normalization would be limited by the desired recognition speed.

3. The various facial features could be ranked according to their importance in recognizing faces and separate modules could be introduced for various parts of the face, e.g. the eye region, the nose region, and the mouth region (Brunelli and Poggio (1993) obtain very good performance using a simple template matching strategy on precisely these regions).

4. An ensemble of recognizers could be used. These could be combined by using simple methods such as a linear combination based on the performance of each network, or via a gating network and the Expectation-Maximization algorithm (Drucker, Cortes, Jackel, Le Cun and Vapnik, 1994; Jacobs, 1995).

Examination of the errors made by networks trained with different random weights and by networks trained with the SOM data versus networks trained with the KL data shows that a combination of networks should improve performance (the set of common errors between the recognizers is often significantly smaller than the total number of errors).

5. Invariance to a group of desired transformations could be enhanced with the addition of pseudo-data to the training database – i.e. the addition of new examples created from the current examples using local deformation, etc. Leen (1991) shows that adding pseudo-data can be equivalent to adding a regularizer to the cost function where the regularizer penalizes changes in the output when the input goes under a transformation for which invariance is desired.

10 Conclusions

A fast, automatic system for face recognition has been presented which is a combination of a local image sample representation, a self-organizing map network, and a convolutional network. The self-organizing map provides quantization of the image samples into a topological space where inputs that are nearby in the original space are also nearby in the output space, which results in invariance to minor changes in the image samples, and the convolutional neural network provides for partial invariance to translation, rotation, scale, and deformation. Substitution of the Karhunen-Loève transform for the self-organizing map produced similar but slightly worse results. The method is capable of rapid classification, requires only fast, approximate normalization and preprocessing, and consistently exhibits better classification performance than the eigenfaces approach (Turk and Pentland, 1991) on the database considered as the number of images per person in the training database is varied from 1 to 5. With 5 images per person the proposed method and eigenfaces result in 3.8% and 10.5% error respectively. The recognizer provides a measure of confidence in its output and classification error approaches zero when rejecting as few as 10% of the examples. There are no explicit three-dimensional models in the system, however it was found that the quantized local image samples used as input to the convolutional network represent smoothly changing shading patterns. Higher level features are constructed from these building blocks in successive layers of the convolutional network. The system is partially invariant to changes in the local image samples, scaling, translation, and deformation by design.

Acknowledgments

We would like to thank Ingemar Cox for helpful comments and the Olivetti Research Laboratory and Ferdinando Samaria for compiling and maintaining the ORL database.

References

Arya, S. and Mount, D. (1993), Algorithms for fast vector quantization, *in* J. A. Storer and M. Cohn, eds, 'Proceedings of DCC 93: Data Compression Conference', IEEE Press, pp. 381–390.

Bauer, H.-U. and Pawelzik, K. R. (1992), 'Quantifying the neighborhood preservation of Self-Organizing Feature Maps', *IEEE Transactions on Neural Networks* **3**(4), 570–579.

Bengio, Y., Le Cun, Y. and Henderson, D. (1994), Globally trained handwritten word recognizer using spatial representation, space displacement neural networks and hidden Markov models, *in* 'Advances in Neural Information Processing Systems 6', Morgan Kaufmann, San Mateo CA.

Blue, J., Candela, G., Grother, P., Chellappa, R. and Wilson, C. (1994), 'Evaluation of pattern classifiers for fingerprint and OCR applications', *Pattern Recognition* **27**(4), 485–501.

Bottou, L., Cortes, C., Denker, J., Drucker, H., Guyon, I., Jackel, L., Le Cun, Y., Muller, U., Sackinger, E., Simard, P. and Vapnik, V. (1994), Comparison of classifier methods: A case study in handwritten digit recognition, *in* 'Proceedings of the International Conference on Pattern Recognition', IEEE Computer Society Press, Los Alamitos, CA.

Brunelli, R. and Poggio, T. (1993), 'Face recognition: Features versus templates', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **15**(10), 1042–1052.

Burton, D. K. (1987), 'Text-dependent speaker verification using vector quantization source coding', *IEEE Transactions on Acoustics, Speech, and Signal Processing* **35**(2), 133.

Chellappa, R., Wilson, C. and Sirohey, S. (1995), 'Human and machine recognition of faces: A survey', *Proceedings of the IEEE* **83**(5), 705–740.

Cox, I. J., Ghosn, J. and Yianilos, P. N. (1995), Feature-based face recognition using mixture-distance, Technical report, NEC Research Institute, Princeton, NJ.

DeMers, D. and Cottrell, G. (1993), Non-linear dimensionality reduction, *in* S. Hanson, J. Cowan and C. L. Giles, eds, 'Advances in Neural Information Processing Systems 5', Morgan Kaufmann Publishers, San Mateo, CA, pp. 580–587.

Dony, R. and Haykin, S. (1995), 'Neural network approaches to image compression', *Proceedings of the IEEE* **83**(2), 288–303.

Drucker, H., Cortes, C., Jackel, L., Le Cun, Y. and Vapnik, V. (1994), 'Boosting and other ensemble methods', *Neural Computation* **6**, 1289–1301.

Fukunaga, K. (1990), *Introduction to Statistical Pattern Recognition, Second Edition*, Academic Press, Boston, MA.

Fukushima, K. (1980), 'Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position', *Biological Cybernetics* **36**, 193–202.

Fukushima, K. (1995), Neocognitron: A model for visual pattern recognition, *in* M. A. Arbib, ed., 'The Handbook of Brain Theory and Neural Networks', MIT Press, Cambridge, Massachusetts, pp. 613–617.

Fukushima, K., Miyake, S. and Ito, T. (1983), 'Neocognitron: A neural network model for a mechanism of visual pattern recognition', *IEEE Transactions on Systems, Man, and Cybernetics* **13**.

Haykin, S. (1994), *Neural Networks, A Comprehensive Foundation*, Macmillan, New York, NY.

Haykin, S. (1996), 'Personal communication'.

Hubel, D. and Wiesel, T. (1962), 'Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex', *Journal of Physiology (London)* **160**, 106–154.

Hummel, J. (1995), Object recognition, *in* M. A. Arbib, ed., 'The Handbook of Brain Theory and Neural Networks', MIT Press, Cambridge, Massachusetts, pp. 658–660.

Jacobs, R. (1995), 'Methods for combining experts' probability assessments', *Neural Computation* **7**, 867–888.

Kanade, T. (1973), Picture Processing by Computer Complex and Recognition of Human Faces, PhD thesis, Kyoto University.

Kita, H. and Nishikawa, Y. (1993), Neural network model of tonotopic map formation based on the temporal theory of auditory sensation, *in* 'Proc. WCNN 93, World Congress on Neural Networks', Vol. II, Lawrence Erlbaum, Hillsdale, NJ, pp. 413–418.

Kohonen, T. (1990), 'The self-organizing map', *Proceedings of the IEEE* **78**, 1464–1480.

Kohonen, T. (1995), *Self-Organizing Maps*, Springer-Verlag, Berlin, Germany.

Le Cun, Y. (1989), Generalisation and network design strategies, Technical Report CRG-TR-89-4, Department of Computer Science, University of Toronto.

Le Cun, Y. and Bengio, Y. (1995), Convolutional networks for images, speech, and time series, *in* M. A. Arbib, ed., 'The Handbook of Brain Theory and Neural Networks', MIT Press, Cambridge, Massachusetts, pp. 255–258.

Le Cun, Y., Boser, B., Denker, J., Henderson, D., Howard, R., Hubbard, W. and Jackel, L. (1990), Handwritten digit recognition with a backpropagation neural network, *in* D. Touretzky, ed., 'Advances in Neural Information Processing Systems 2', Morgan Kaufmann, San Mateo, CA, pp. 396–404.

Le Cun, Y., Denker, J. and Solla, S. (1990), Optimal Brain Damage, *in* D. Touretzky, ed., 'Advances in Neural Information Processing Systems', Vol. 2, (Denver 1989), Morgan Kaufmann, San Mateo, pp. 598–605.

Leen, T. K. (1991), 'From data distributions to regularization in invariant learning', *Neural Computation* **3**(1), 135–143.

Luttrell, S. P. (1989), Hierarchical self-organizing networks, *in* 'Proc. 1st IEE Conf. of Artificial Neural Networks', British Neural Network Society, London, UK, pp. 2–6.

Marr, D. (1982), *Vision*, W. H. Freeman, San Francisco.

Miller, B. (1994), 'Vital signs of identity', *IEEE Spectrum* pp. 22–30.

Moghaddam, B. and Pentland, A. (1994), Face recognition using view-based and modular eigenspaces, *in* 'Automatic Systems for the Identification and Inspection of Humans, SPIE', Vol. 2257.

Obermayer, K., Blasdel, G. G. and Schulten, K. (1991), A neural network model for the formation and for the spatial structure of retinotopic maps, orientation and ocular dominance columns, *in* T. Kohonen, K. Mäkisara, O. Simula and J. Kangas, eds, 'Artificial Neural Networks', Elsevier, Amsterdam, Netherlands, pp. 505–511.

Obermayer, K., Ritter, H. and Schulten, K. (1990), Large-scale simulation of a self-organizing neural network: Formation of a somatotopic map, *in* R. Eckmiller, G. Hartmann and G. Hauske, eds, 'Parallel Processing in Neural Systems and Computers', North-Holland, Amsterdam, Netherlands, pp. 71–74.

Pentland, A., Moghaddam, B. and Starner, T. (1994), View-based and modular eigenspaces for face recognition, *in* 'IEEE Conference on Computer Vision and Pattern Recognition'.

Pentland, A., Starner, T., Etcoff, N., Masoiu, A., Oliyide, O. and Turk, M. (1993), Experiments with eigenfaces, *in* 'Looking at People Workshop, International Joint Conference on Artificial Intelligence 1993', Chamberry, France.

Perret, Rolls and Caan (1982), 'Visual neurones responsive to faces in the monkey temporal cortex', *Experimental Brain Research* **47**, 329–342.

Qi, Y. and Hunt, B. (1994), 'Signature verification using global and grid features', *Pattern Recognition* **27**(12), 1621–1629.

Rowley, H. A., Baluja, S. and Kanade, T. (1995), Human face detection in visual scenes, Technical Report CMU-CS-95-158, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.

Samaria, F. (1994), Face Recognition using Hidden Markov Models, PhD thesis, Trinity College, University of Cambridge, Cambridge.

Samaria, F. and Harter, A. (1994), Parameterisation of a stochastic model for human face identification, *in* 'Proceedings of the 2nd IEEE workshop on Applications of Computer Vision', Sarasota, Florida.

Sung, K.-K. and Poggio, T. (1995), Learning human face detection in cluttered scenes, *in* 'Computer Analysis of Images and Patterns', pp. 432–439.

Sutherland, K., Renshaw, D. and Denyer, P. (1992), Automatic face recognition, *in* 'First International Conference on Intelligent Systems Engineering', IEEE Press, Piscataway, NJ, pp. 29–34.

Turk, M. and Pentland, A. (1991), 'Eigenfaces for recognition', *J. of Cognitive Neuroscience* **3**, 71–86.