

Face Recognition: A Convolutional Neural-Network Approach

Steve Lawrence, *Member, IEEE*, C. Lee Giles, *Senior Member, IEEE*, Ah Chung Tsoi, *Senior Member, IEEE*,
and Andrew D. Back, *Member, IEEE*

Abstract—Faces represent complex multidimensional meaningful visual stimuli and developing a computational model for face recognition is difficult. We present a hybrid neural-network solution which compares favorably with other methods. The system combines local image sampling, a self-organizing map (SOM) neural network, and a convolutional neural network. The SOM 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 (KL) transform in place of the SOM, and a multilayer perceptron (MLP) in place of the convolutional network. The KL transform performs almost as well (5.3% error versus 3.8%). The MLP 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 on the database considered as the number of images per person in the training database is varied from one to five. With five 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.

Index Terms—Face recognition, convolutional neural networks, self-organizing feature maps, Karhunen–Loève transforms, hybrid systems, access control, pattern recognition, image classification.

I. INTRODUCTION

THE REQUIREMENT for reliable personal identification in computerized access control has resulted in an increased interest in biometrics.¹ Biometrics being investigated

Manuscript received January 1, 1996; revised June 13, 1996. This work was supported in part by the Australian Research Council (ACT) and the Australian Telecommunications and Electronics Research Board (SL).

S. Lawrence is with the NEC Research Institute, Princeton, NJ 08540 USA. He is also with the Department of Electrical and Computer Engineering, University of Queensland, St. Lucia, Australia.

C. L. Giles is with the NEC Research Institute, Princeton, NJ 08540 USA. He is also with the Institute for Advanced Computer Studies, University of Maryland, College Park, MD 20742 USA.

A. C. Tsoi and A. D. Back are with the Department of Electrical and Computer Engineering, University of Queensland, St. Lucia, Australia.

Publisher Item Identifier S 1045-9227(97)00234-8.

¹Physiological or behavioral characteristics which uniquely identify us.

include fingerprints [4], speech [7], signature dynamics [36], and face recognition [8]. Sales of identity verification products exceed \$100 million [29]. Face recognition has the benefit of being a passive, nonintrusive system for verifying personal identity. The techniques used in the best face recognition systems may depend on the application of the system. We can identify at least two broad categories of face recognition systems.

- 1) We want 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 [34]. Often only one image is available per person. It is usually not necessary for recognition to be done in real-time.
- 2) We want to identify particular people in real-time (e.g., in a security monitoring system, location tracking system, etc.), or we want to allow access to a group of people and deny access to all others (e.g., access to a building, computer, etc.) [8]. Multiple images per person are often available for training and real-time recognition is required.

In this paper, we are primarily interested in the second case.² We are interested in recognition with varying facial detail, expression, pose, etc. We do not consider invariance to high degrees of rotation or scaling—we assume that a minimal preprocessing stage is available if required. 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 [37], [40], and [43].

The remainder of this paper is organized as follows. The data we used is presented in Section II and related work with this and other databases is discussed in Section III. The components and details of our system are described in Sections IV and V, respectively. We present and discuss our results in Sections VI and VII. Computational complexity is considered in Section VIII, Section IX lists avenues for further research, and we draw conclusions in Section X.

II. DATA

We have used the ORL database, which contains a set of faces taken between April 1992 and April 1994 at the Olivetti

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

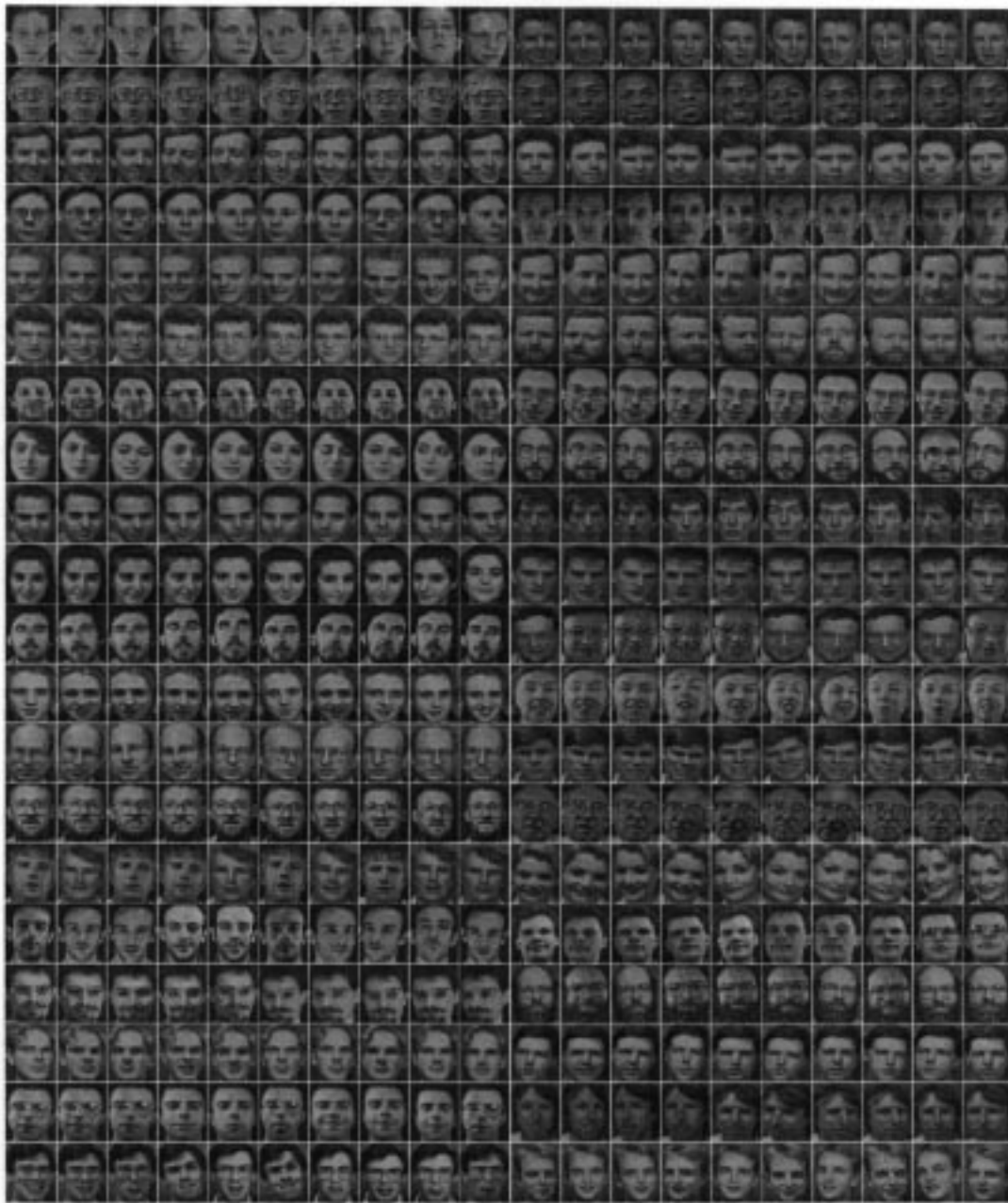


Fig. 1. The ORL face database. There are ten images each of the 40 subjects.

Research Laboratory in Cambridge, U.K.³ There are ten 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/nonsmiling), and facial details (glasses/no glasses). All 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 Fig. 1 and a larger set of images for one subject is shown in Fig. 2. The images are greyscale with a resolution of 92×112 .

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

III. RELATED WORK

A. Geometrical Features

Many people have explored geometrical feature based methods for face recognition. Kanade [17] presented an automatic feature extraction method based on ratios of distances and reported a recognition rate of between 45–75% with a database of 20 people. Brunelli and Poggio [6] 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 *et al.* [9] have recently introduced a *mixture-distance* technique which achieves a recognition rate



Fig. 2. The set of ten images for one subject. Considerable variation can be seen.

of 95% using a query database of 95 images from a total of 685 individuals. 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.⁴ 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 provide a high degree of accuracy and require considerable computational capacity [41].

B. 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 (3-D) models to matched models [28]. However, Turk and Pentland [43] argue that it is likely that there is also a recognition process based on low-level two-dimensional (2-D) image processing. Their argument is based on the early development and extreme rapidity of face recognition in humans and on physiological experiments in a monkey cortex which claim to have isolated neurons that respond selectively to faces [35]. However, it is not clear that these experiments exclude the sole operation of the Marr paradigm.

Turk and Pentland [43] 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 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.* [33], [34] good results are reported on a large database (95% recognition of 200 people from a

database of 3000). It is difficult to draw broad conclusions as many of the images of the same people look very similar, and the database has accurate registration and alignment [30]. In Moghaddam and Pentland [30], very good results are reported with the FERET 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.

Swets and Weng [42] present a method of selecting discriminant eigenfeatures using multidimensional linear discriminant analysis. They present methods for determining the most expressive features (MEF) and the most discriminatory features (MDF). We are not currently aware of the availability of results which are comparable with those of eigenfaces (e.g., on the FERET database as in Moghaddam and Pentland [30]).

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.

C. Template Matching

Template matching methods such as [6] operate by performing direct correlation of image segments. Template matching is only effective when the query images have the same scale, orientation, and illumination as the training images [9].

D. Graph Matching

Another approach to face recognition is the well known method of graph matching. In [21], Lades *et al.* present a dynamic link architecture for distortion invariant object recognition which employs elastic graph matching to find the closest stored graph. Objects are represented with sparse graphs whose vertices are labeled with a multiresolution description in terms of a local power spectrum, and whose edges are labeled with geometrical distances. They present good results with a database of 87 people and test images composed of different expressions and faces turned 15°. The matching process is computationally expensive, taking roughly 25 s to compare an image with 87 stored objects when using

⁴A mugshot database typically contains side views where the performance of feature point methods is known to improve [8].

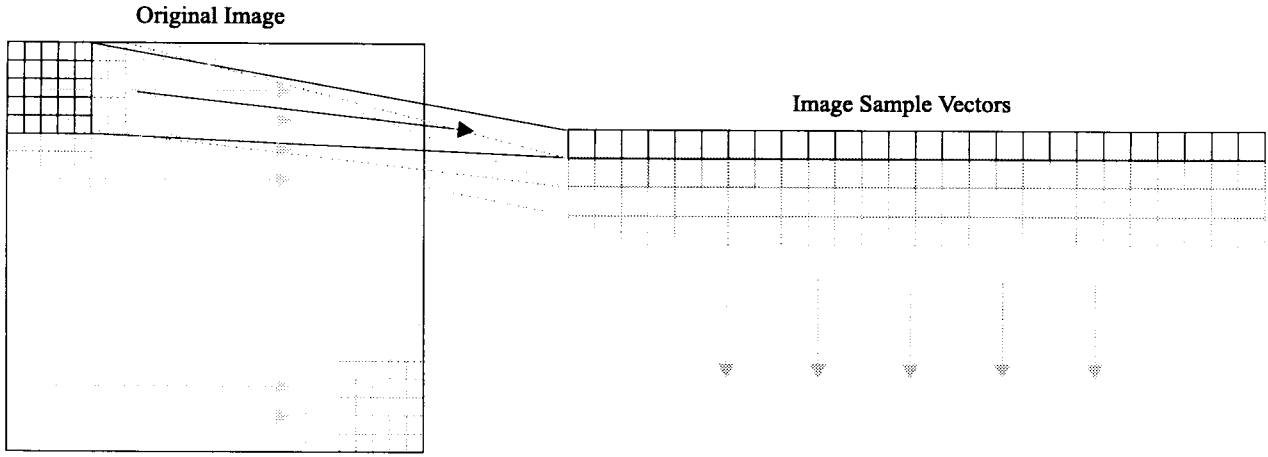


Fig. 3. A depiction of the local image sampling process. A window is stepped over the image and a vector is created at each location.

a parallel machine with 23 transputers. Wiskott *et al.* [45] use an updated version of the technique and compare 300 faces against 300 different faces of the same people taken from the FERET database. They report a recognition rate of 97.3%. The recognition time for this system was not given.

E. 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). We briefly describe a couple of approaches.

In [10] the first 50 principal components of the images are extracted and reduced to five dimensions using an autoassociative neural network. The resulting representation is classified using a standard multilayer perceptron (MLP). 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.

A hierarchical neural network which is grown automatically and not trained with gradient descent was used for face recognition by Weng and Huang [44]. They report good results for discrimination of ten distinctive subjects.

F. The ORL Database

In [39] a hidden Markov model (HMM)-based approach is used for classification of the ORL database images. The best model resulted in a 13% error rate. Samaria also performed extensive tests using the popular eigenfaces algorithm [43] on the ORL database and reported a best error rate of around 10% when the number of eigenfaces was between 175 and 199. We implemented the eigenfaces algorithm and also observed around 10% error. In [38] Samaria extends the top-down HMM of [39] with pseudo 2-D HMM's. The error rate reduces to 5% at the expense of high computational complexity—a single classification takes 4 min on a Sun Sparc II. Samaria notes that although an increased recognition rate was achieved the segmentation obtained with the pseudo 2-D HMM's appeared quite erratic. Samaria uses the same training and test set sizes as we do (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.

IV. SYSTEM COMPONENTS

A. Overview

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

B. Local Image Sampling

We have evaluated two different methods of representing local image samples. In each method a window is scanned over the image as shown in Fig. 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 i th column and the j th 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 $[x_{i-W,j-W}, x_{i-W,j-W+1}, \dots, x_{ij}, \dots, x_{i+W,j+W-1}, x_{i+W,j+W}]$.
- 2) The second method creates a representation of the local sample by forming a vector out of 1) the intensity of the center pixel x_{ij} , and 2) the difference in intensity between the center pixel and all other pixels within the square window. The vector is given by $[x_{ij} - x_{i-W,j-W}, x_{ij} - x_{i-W,j-W+1}, \dots, w_{ij}x_{ij}, \dots, x_{ij} - x_{i+W,j+W-1}, x_{ij} - x_{i+W,j+W}]$. 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} connected to the central intensity component.

C. The Self-Organizing Map

1) *Introduction:* Maps are an important part of both natural and artificial neural information processing systems [2]. Examples of maps in the nervous system are retinotopic maps in the visual cortex [31], tonotopic maps in the auditory cortex [18], and maps from the skin onto the somatosensory cortex [32]. The SOM, introduced by Kohonen [19], [20], 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).

2) *Algorithm:* We give a brief description of the SOM algorithm, for more details see [20]. 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 2-D SOM is shown in Fig. 4. A reference vector in the input space, $\mathbf{m}_i \equiv [\mu_{i1}, \mu_{i2}, \dots, \mu_{in}]^T \in \mathcal{R}^n$, is assigned to each node in the SOM. During training, each input vector, \mathbf{x} , is compared to all of the \mathbf{m}_i , obtaining the location of the closest match \mathbf{m}_c (given by $\|\mathbf{x} - \mathbf{m}_c\| = \min_i \{\|\mathbf{x} - \mathbf{m}_i\|\}$ where $\|\mathbf{a}\|$ denotes the norm of vector \mathbf{a}). The input point is mapped to this location in the SOM. Nodes in the SOM are updated according to

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h_{ci}(t)[\mathbf{x}(t) - \mathbf{m}_i(t)] \quad (1)$$

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

$$h_{ci} = \alpha(t) \exp\left(-\frac{\|\mathbf{r}_c - \mathbf{r}_i\|^2}{2\sigma^2(t)}\right) \quad (2)$$

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 [20]. The use of the neighborhood function means that nodes which are topographically close in the SOM structure are moved toward the input pattern along with the winning node. This creates a smoothing effect 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

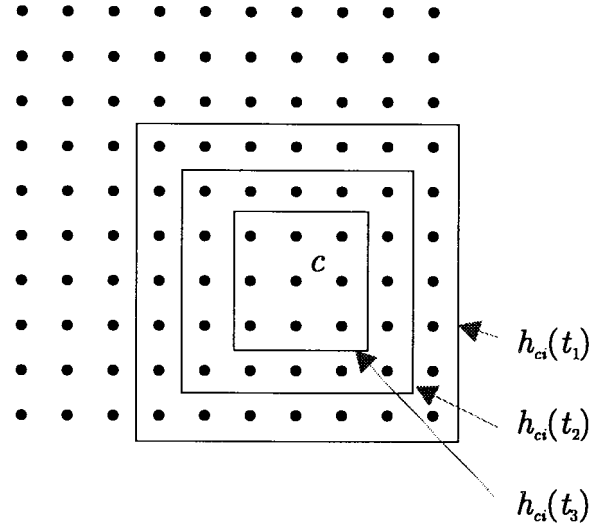


Fig. 4. A 2-D SOM showing a square neighborhood function which starts as $h_{ci}(t_1)$ and reduces in size to $h_{ci}(t_3)$ over time.

node. The SOM can be considered a nonlinear projection of the probability density, $p(\mathbf{x})$ [20].

3) *Improving the Basic SOM:* The original SOM is computationally expensive due to the following.

- 1) In the early stages of learning, many nodes are adjusted in a correlated manner. Luttrell [27] proposed a method, which is used here, that starts by learning in a small network, and doubles the size of the network 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 from the above node doubling strategy.⁵

D. KL Transform

The optimal linear method⁶ for reducing redundancy in a dataset is the KL transform or eigenvector expansion via principle components analysis (PCA) [12]. PCA generates a set of orthogonal axes of projections known as the principal components, or the eigenvectors, of the input data distribution in the order of decreasing variance. The KL transform is a well-known statistical method for feature extraction and multivariate data projection and has been used widely in pattern recognition, signal processing, image processing, and data analysis. Points in an n -dimensional input space are projected into an m -dimensional space, $m \leq n$. 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

⁵This assumes that the topological order is optimal prior to each doubling step.

⁶In the least mean squared error sense.

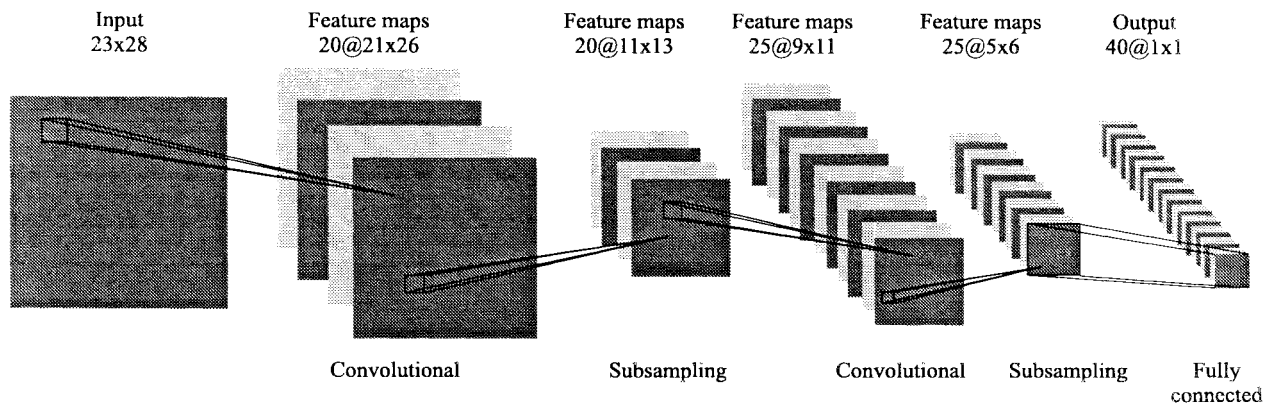


Fig. 5. A typical convolutional network.

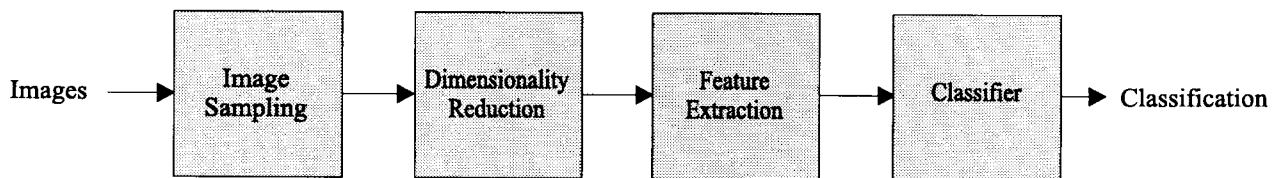


Fig. 6. A high-level block diagram of the system we have used for face recognition.

is used on the entire images, whereas it is only used on small local image samples in this work.

E. Convolutional Networks

The problem of face recognition from 2-D 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 2-D images as input, there is no invariance to translation or local deformation of the images [23].

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 [3], [5], [22]–[24].

A typical convolutional network is shown in Fig. 5 [24]. The network consists of a set of layers each of which contains one or more planes. Approximately centered and normalized images 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 1960's with the perceptron and Hubel and Wiesel's [15] discovery of locally sensitive orientation-selective neurons in the cat's visual system [23]. 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 two: $y_{ij} = (x_{2i,2j} + x_{2i+1,2j} + x_{2i,2j+1} + x_{2i+1,2j+1})/4$ 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 [13]. A connection strategy can be used to reduce the number of weights in the network. For example, with reference to Fig. 5, Le Cun *et al.* [24] connect the feature maps in the second convolutional layer only to one or two of the maps in the first subsampling layer (the connection strategy was chosen manually).

V. SYSTEM DETAILS

The system we have used for face recognition is a combination of the preceding parts—a high-level block diagram is shown in Figs. 6 and 7 shows a breakdown of the various subsystems that we experimented with or discuss.

Our system works as follows (we give complete details of dimensions etc. 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 Fig. 3 and local image samples are extracted at each step. At each step the window is moved by four pixels.
- 2) An SOM (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

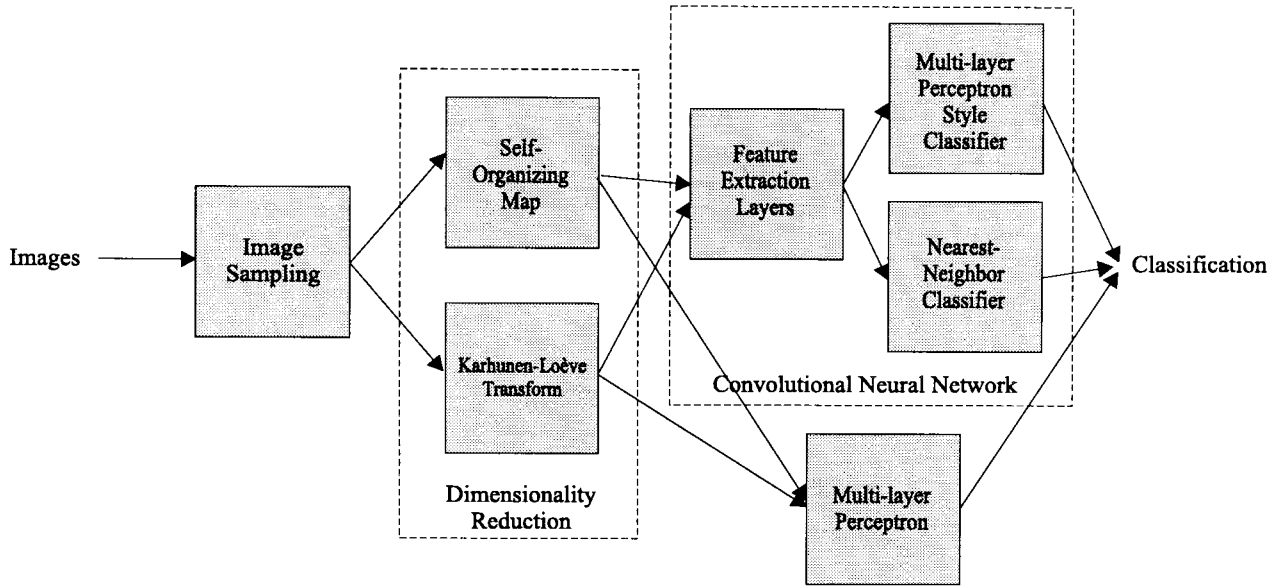


Fig. 7. A diagram of the system we have used for face recognition showing alternative methods which we consider in this paper. The top “MLP style classifier” represents the final MLP style fully connected layer of the convolutional network. We have shown this decomposition of the convolutional network 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 MLP shows that the entire convolutional network can be replaced with an MLP. We present results with either a SOM or the KL transform used for dimensionality reduction, and either a convolutional neural network or an MLP for classification.

ordered values. The three dimensions of the SOM can be thought of as three features. We also experimented with replacing the SOM with the KL transform. In this case, the KL transform projects the vectors in the 25-dimensional space into a 3-D 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 created by the SOM. (Each input image is now represented by three 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 four, the maps are 23×28).
- 4) A convolutional neural network is trained on the newly created training set. We also experimented with training a standard MLP for comparison.

A. Simulation Details

In this section we give the details of one of the best performing systems.

For the SOM, training is split into two phases as recommended by Kohonen [20]—an ordering phase and a fine-adjustment phase. In the first phase 100 000 updates are performed, and in the second 50 000 are performed. In the first phase, the neighborhood radius starts at two-thirds of the size of the map and reduces linearly to one. The learning rate during this phase is: $0.7 \times (1 - n/N)$ where n is the current update number, and N is the total number of updates. In the second phase, the neighborhood radius starts at two and is reduced to

one. The learning rate during this phase is $0.02 \times (1 - n/N)$.

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 I. The network was trained with backpropagation [13] 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 -1 to 1 . All nodes included a bias input which was part of the optimization process. The best of ten 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 [13]. Target outputs were -0.8 and 0.8 using the *tanh* output activation function.⁷ The quadratic cost function

⁷This helps avoid saturating the sigmoid function. If targets were set to the asymptotes of the sigmoid this would tend to: 1) drive the weights to infinity; 2) cause outlier data to produce very large gradients due to the large weights; and 3) produce binary outputs even when incorrect—leading to decreased reliability of the confidence measure.

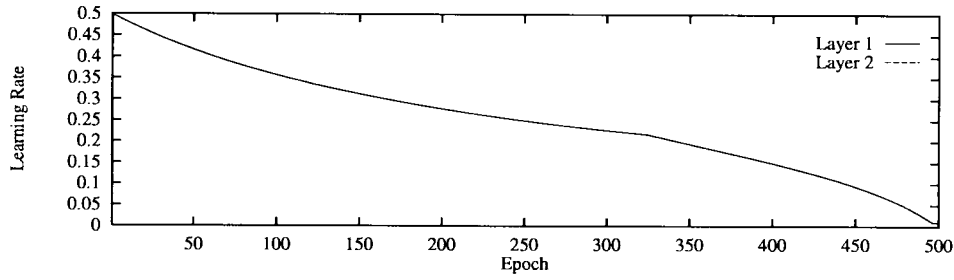


Fig. 8. The learning rate as a function of the epoch number.

TABLE I

DIMENSIONS FOR THE CONVOLUTIONAL NETWORK. THE CONNECTION PERCENTAGE REFERS TO THE PERCENTAGE OF NODES IN THE PREVIOUS LAYER TO WHICH EACH NODE IN THE CURRENT LAYER IS CONNECTED—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.* [24] 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 (21×26) IS EQUAL TO THE TOTAL NUMBER OF WAYS OF POSITIONING A 3×3 RECEPTIVE FIELD ON THE INPUT-LAYER PLANES (23×28)

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

was used. A search then converge learning rate schedule was used⁸

$$\eta = \frac{\eta_0}{\frac{n}{N/2} + \frac{c_1}{\max\left(1, \left(c_1 - \frac{\max(0, c_1(n - c_2N))}{(1 - c_2)N}\right)\right)}}$$

where η = learning rate, η_0 = initial learning rate = 0.1, N = total training epochs, n = current training epoch, $c_1 = 50$, and $c_2 = 0.65$. The schedule is shown in Fig. 8. Total training time was around four hours on an SGI Indy 100Mhz MIPS R4400 system.

VI. EXPERIMENTAL RESULTS

We performed various experiments and present the results here. Except when stated otherwise, all experiments were performed with five training images and five test images per person for a total of 200 training images and 200 test images. There was no overlap between the training and test sets. We note that a system which guesses the correct answer would be right one out of 40 times, giving an error rate of 97.5%. For the

⁸Relatively 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 (a simpler linear schedule also works well). We have found the use of learning rate schedules to improve performance considerably.

following sets of experiments, we vary only one parameter 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.⁹ We note that ideally we would like to have performed more simulations per reported result, however, we were limited in terms of computational capacity available to us. The constants used in each set of experiments were: number of classes: 40, dimensionality reduction method: SOM, dimensions in the SOM: three, number of nodes per SOM dimension: five, image sample extraction: original intensity values, training images per class: five. 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 II and Fig. 9 show the error rate of the system as the number of classes is varied from ten to 40. We made no attempt to optimize the system for the smaller numbers of classes. As we expect, performance improves with fewer classes to discriminate between.
- 2) *Variation of the dimensionality of the SOM*—Table III and Fig. 10 show the error rate of the system as the dimension of the SOM is varied from one to four. The best performing value is three dimensions.
- 3) *Variation of the quantization level of the SOM*—Table IV and Fig. 11 show the error rate of the system as the size of the SOM is varied from four to ten nodes per dimension. The SOM has three dimensions in each case. The best average error rate occurs for eight or nine nodes per dimension. This is also the best average error rate of all experiments.
- 4) *Variation of the image sample extraction algorithm*—Table V shows the result of using the two local image sample representations described earlier. We found that using the original intensity values gave the best performance. We investigated altering the weight assigned to the central intensity value in the alternative representation but were unable to improve the results.
- 5) *Substituting the SOM with the KL transform*—Table VI shows the results of replacing the SOM with the KL

⁹We ran multiple simulations in each experiment where we varied the selection of the training and test images (out of a total of $10!/5! = 30\,240$ possibilities) and the random seed used to initialize the weights in the convolutional neural network.

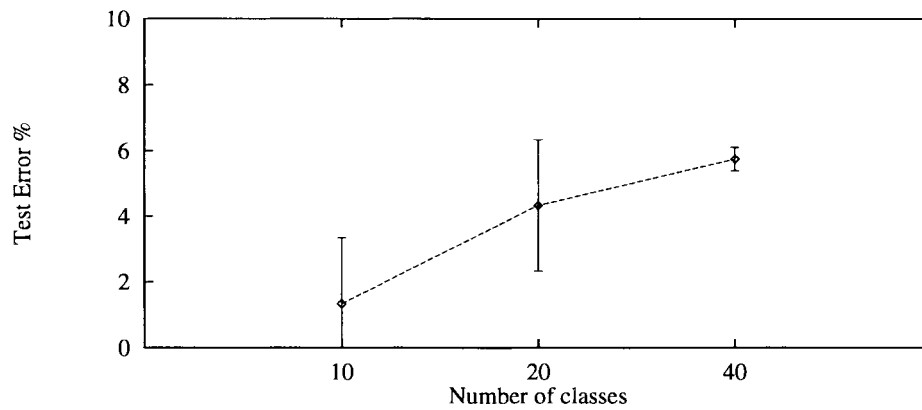


Fig. 9. The error rate as a function of the number of classes. We did not modify the network from that used for the 40 class case.

transform. We investigated using the first one, two, or three eigenvectors for projection. Surprisingly, the system performed best with only one eigenvector. The best SOM parameters we tried 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.

- 6) *Replacing the CN with an MLP*—Table VII shows the results of replacing the convolutional network with an MLP. Performance is very poor. This result was expected because the MLP 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. We expect the MLP 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. We investigated the following hidden layer sizes for the MLP: 20, 50, 100, 200, and 500. The best performance was obtained with 200 hidden nodes and a training time of two days. 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. We note that it may be considered fairer to compare against an MLP with multiple hidden layers [14], however selection of the appropriate number of nodes in each layer is difficult (e.g., we have tried a network with two hidden layers containing 100 and 50 nodes, respectively, which resulted in an error rate of 90%).

- 7) *The tradeoff between rejection threshold and recognition accuracy*—Fig. 12 shows a histogram of the recognizer's confidence for the cases when the classifier is correct and when it is wrong for one of the best performing systems. From this graph we expect that classification performance will increase significantly if we reject cases below a certain confidence threshold.

Fig. 13 shows the system performance as the rejection threshold is increased. We can see that by rejecting examples with low confidence we can significantly increase the classification performance of the system. If we consider a system which used a video camera to take a number of pictures over a short period, we could expect that a high performance would be attainable with an appropriate rejection threshold.

- 8) *Comparison with other known results on the same database*—Table VIII shows a summary of the performance of the systems for which we have results using the ORL database. In this case, we used a SOM quantization level of eight. Our system is the best performing system¹⁰ and performs recognition roughly 500 times faster than the second best performing system—the pseudo 2-D HMM's of Samaria. Fig. 14 shows the images which were incorrectly classified for one of the best performing systems.
- 9) *Variation of the number of training images per person*. Table IX shows the results of varying the number of images per class used in the training set from one to five for PCA+CN, SOM+CN and also for the eigenfaces algorithm. We implemented two versions of the eigenfaces algorithm—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 [43]. However, we found that using separate training vectors for each training image resulted in better performance. We found that using between 40 to 100 eigenfaces resulted in similar performance. We can see that the PCA+CN and SOM+CN methods are both superior to the eigenfaces technique even when there is only one training image per person. The SOM+CN method consistently performs better than the PCA+CN method.

VII. DISCUSSION

The results indicate that a convolutional network can be more suitable in the given situation when compared with a

¹⁰The 4% error rate reported is an average of multiple simulations—individual simulations have given error rates as low as 1.5%.

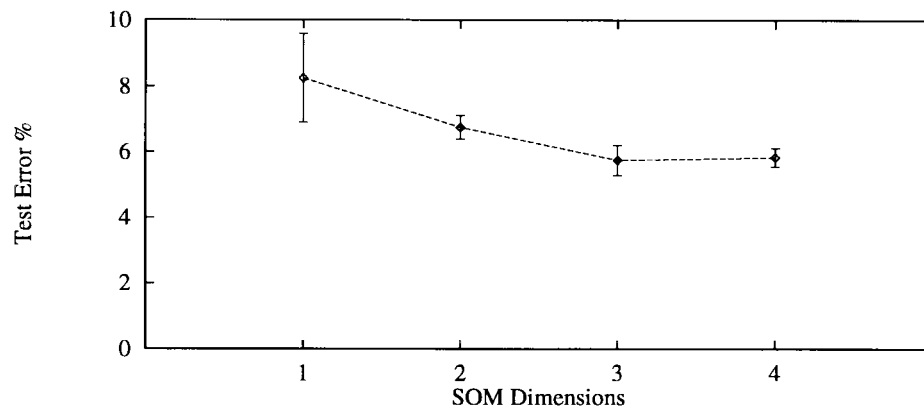


Fig. 10. The error rate as a function of the number of dimensions in the SOM.

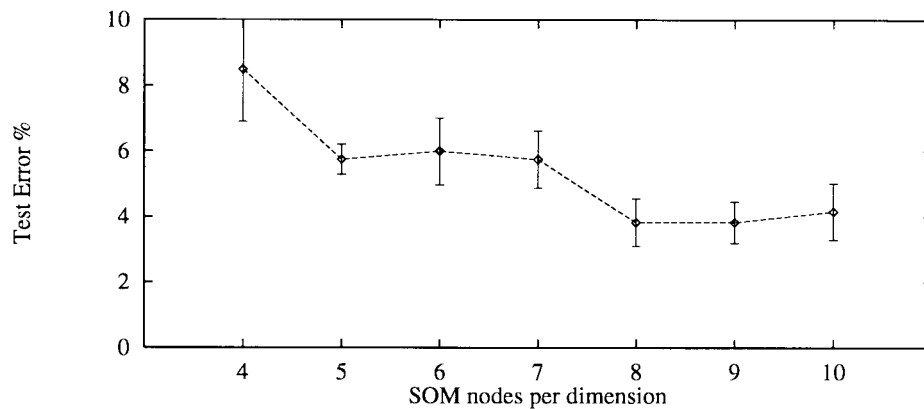


Fig. 11. The error rate as a function of the number of nodes per dimension in the SOM.

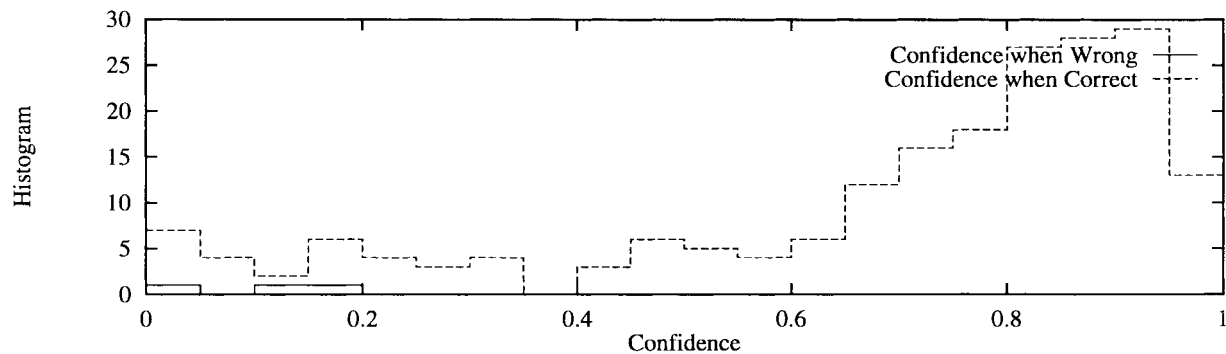


Fig. 12. A histogram depicting the confidence of the classifier when it turns out to be correct, and the confidence when it is wrong. The graph suggests that we can improve classification performance considerably by rejecting cases where the classifier has a low confidence.

TABLE II
ERROR RATE OF THE FACE RECOGNITION SYSTEM WITH VARYING NUMBER OF CLASSES (SUBJECTS). EACH RESULT IS THE AVERAGE OF THREE SIMULATIONS

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

TABLE III
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

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

standard MLP. This correlates with the common belief that the incorporation of prior knowledge is desirable for MLP style networks (the CN incorporates domain knowledge regarding the relationship of the pixels and desired invariance to a degree of translation, scaling, and local deformation).

Convolutional networks have traditionally been used on raw images without any preprocessing. Without the preprocessing we have used, 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

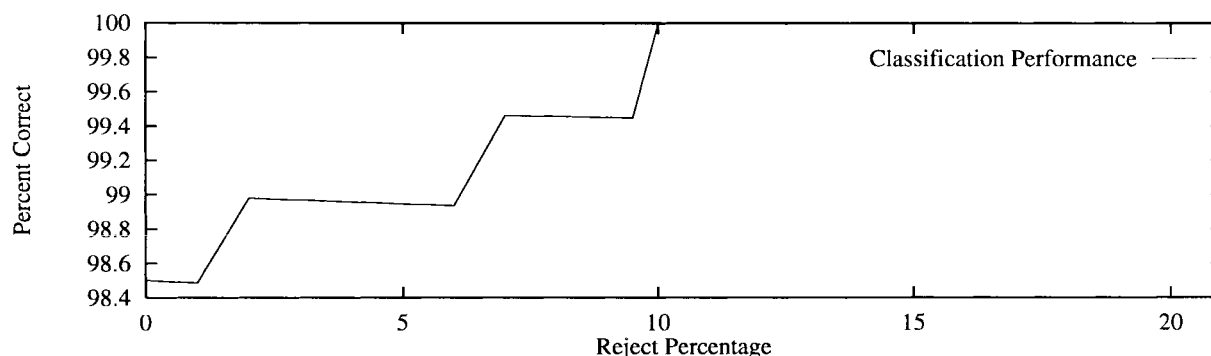


Fig. 13. The test set classification performance as a function of the percentage of samples rejected. Classification performance can be improved significantly by rejecting cases with low confidence.



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

TABLE IV
ERROR RATE OF THE FACE RECOGNITION SYSTEM WITH VARYING
NUMBER OF NODES PER DIMENSION IN THE SOM. EACH
RESULT GIVEN IS THE AVERAGE OF THREE SIMULATIONS

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 V
ERROR RATE OF THE FACE RECOGNITION SYSTEM WITH VARYING IMAGE SAMPLE
REPRESENTATION. EACH RESULT IS THE AVERAGE OF THREE SIMULATIONS

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

TABLE VI
ERROR RATE OF THE FACE RECOGNITION SYSTEM WITH LINEAR
PCA AND SOM FEATURE EXTRACTION MECHANISMS.
EACH RESULT IS THE AVERAGE OF THREE SIMULATIONS

Dimensionality reduction	Linear PCA	SOM
Error rate	5.33%	3.83%

TABLE VII
ERROR RATE COMPARISON OF THE VARIOUS FEATURE
EXTRACTION AND CLASSIFICATION METHODS. EACH
RESULT IS THE AVERAGE OF THREE SIMULATIONS

	Linear PCA	SOM
MLP	41.2%	39.6%
CN	5.33%	3.83%

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)].

Fig. 15 shows the randomly chosen initial local image samples corresponding to each node in a 2-D SOM, and the final samples to which the SOM converges. Scanning across the rows and columns we can see 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. Fig. 16 shows the activation of the nodes in a sample convolutional network for a particular test image.

TABLE VIII

ERROR RATE OF THE VARIOUS SYSTEMS. ¹On a Sun Sparc II. ²On 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 THREE TIMES FASTER THAN THE SUN SPARC II, MAKING THE SOM+CN APPROXIMATELY 160 TIMES FASTER THAN THE PSEUDO 2-D HMM'S FOR CLASSIFICATION

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 IX

ERROR RATE FOR THE EIGENFACES ALGORITHM AND THE SOM+CN AS THE SIZE OF THE TRAINING SET IS VARIED FROM ONE TO FIVE IMAGES PER PERSON. 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

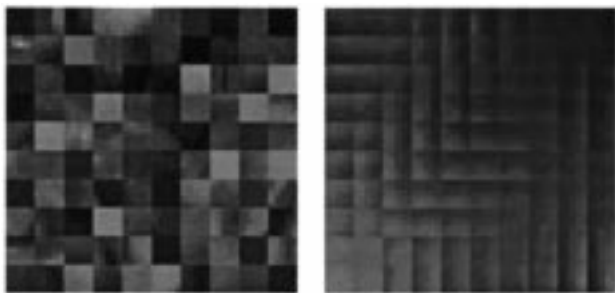


Fig. 15. SOM image samples before training (a random set of image samples) and after training.

Fig. 17 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 [37], 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, as expected, the eyes, nose, mouth, chin, and hair regions are all important to the classification task.

Can the convolutional network feature extraction form the optimal set of features? The answer is negative—it is unlikely that the network could extract an optimal set of features for all images. Although the exact process of human face recognition is unknown, there are many features which humans may use but our system is unlikely to discover optimally—e.g., 1) knowledge of the 3-D structure of the face; 2) knowledge of the nose, eyes, mouth, etc.; 3) generalization to glasses/no glasses, different hair growth, etc.; and 4) knowledge of facial expressions.

VIII. COMPUTATIONAL COMPLEXITY

The SOM takes considerable time to train. This is not a drawback of the approach however, as the system can be extended to cover new classes without retraining the SOM. All that 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 we have reported here, the quantized output of the SOM is very similar if we train it with only 20 classes instead of 40. In addition, the KL transform can be used in place of the SOM with a minimal impact on system performance.

It also takes a considerable amount of time to train a convolutional network; how significant is this? The convolutional network extracts features from the image. It is possible to use fixed feature extraction. Consider if we separate the convolutional network 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 which we want to recognize, the features extracted from the first section could be expected to 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). We note that it may be possible to replace the second part with another type of classifier—e.g., a nearest-neighbors classifier. In this case the time required for retraining the system when adding new classes is minimal (the extracted feature vectors are simply stored for the training images).

To give an idea of the computational complexity of each part of the system we define:

N_c	the number of classes;
N_s	the number of nodes in the SOM;
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;
$N_{samples}$	the number of training samples for the SOM or the KL projection;
N_{window}	the number of local image samples per image.

Tables X and XI show the approximate complexity of the various parts of the system during training and classification. We show the complexity for both the SOM and KL alternatives for dimensionality reduction and for both the neural network (MLP) and a nearest-neighbors classifier (as the last part of the convolutional network—not as a complete replacement, i.e., this is not the same as the earlier MLP experiments). We note



Fig. 16. 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.

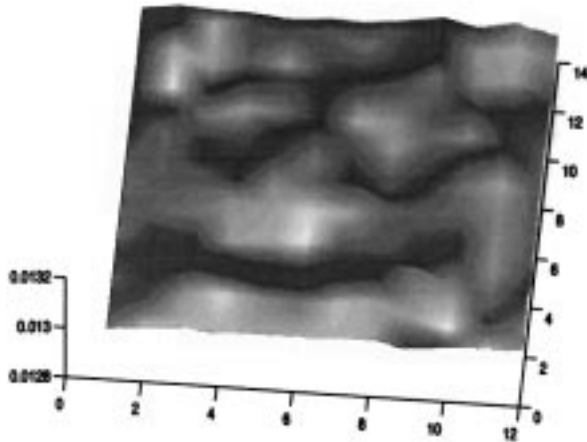


Fig. 17. Sensitivity to various parts of the input image. It can be observed that the eyes, mouth, nose, chin, and hair regions are all important for the classification. The z axis corresponds to the mean squared error rather than the classification error (the mean squared error is preferable because it varies in a smoother fashion as the input images are perturbed). The image orientation corresponds to upright face images.

TABLE X

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	Training complexity
KL	$O((2 + N_{id}^2)N_{samples} + 3N_{od}^3) \approx O(N_{id}^2 + N_{od}^3)$
SOM	$O(k_1 N_{samples} N_n k_2 \log N_s) \approx O(N_{samples} N_n \log N_s)$ (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 XI

CLASSIFICATION COMPLEXITY. k_2 REPRESENTS THE DEGREE OF SHARED WEIGHT REPLICATION

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

that the constant associated with the log factors may increase exponentially in the worst case (cf. neighbor searching in high-dimensional spaces [1]). We have aimed 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 XI, 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).

In terms of computation time, the requirements of real-time tasks varies. The system we have 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.

IX. FURTHER RESEARCH

We can identify the following avenues for improving performance.

- 1) More careful selection of the convolutional network architecture, e.g., by using the optimal brain damage algorithm [25] as used by Le Cun *et al.* [24] to improve generalization and speedup 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 [6] 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 via simple methods such as a linear combination based on the performance of each network, or via a gating network and the expectation-maximization algorithm [11], [16]. Examination of the errors made by networks trained with different random seeds 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 much 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 translation, etc. Leen [26] shows that adding pseudodata 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.

X. CONCLUSIONS

We have presented a fast, automatic system for face recognition which is a combination of a local image sample representation, an SOM network, and a convolutional network for face recognition. The SOM 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, 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 KL transform for the SOM 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 [43] on the database considered as the number of images per person in the training database is varied from one to five. With five 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 have presented avenues for further improvement.

There are no explicit 3-D models in our system, however we have 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. In comparison with the eigenfaces approach, we believe that the system presented here is able to learn more appropriate features in order to provide improved generalization. The system is partially invariant to changes in the local image samples, scaling, translation and deformation by design.

ACKNOWLEDGMENT

The authors would like to thank I. Cox, S. Haykin, and the anonymous reviewers for helpful comments, and the Olivetti Research Laboratory and Ferdinando Samaria for compiling and maintaining the ORL database.

REFERENCES

- [1] S. Arya and D. M. Mount, "Algorithms for fast vector quantization," in *Proc. DCC 93: Data Compression Conf.*, J. A. Storer and M. Cohn, Eds. Piscataway, NJ: IEEE Press, pp. 381–390.
- [2] H.-U. Bauer and K. R. Pawelzik, "Quantifying the neighborhood preservation of self-organizing feature maps," *IEEE Trans. Neural Networks*, vol. 3, pp. 570–579, 1992.
- [3] Y. Bengio, Y. Le Cun, and D. Henderson, "Globally trained handwritten word recognizer using spatial representation, space displacement neural networks, and hidden Markov models," in *Advances in Neural Information Processing Systems 6*. San Mateo, CA: Morgan Kaufmann, 1994.
- [4] J. L. Blue, G. T. Candela, P. J. Grother, R. Chellappa, and C. L. Wilson, "Evaluation of pattern classifiers for fingerprint and OCR applications," *Pattern Recognition*, vol. 27, no. 4, pp. 485–501, Apr. 1994.
- [5] L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, L. Jackel, Y. Le Cun, U. Muller, E. Sackinger, P. Simard, and V. N. Vapnik, "Comparison of classifier methods: A case study in handwritten digit recognition," in *Proc. Int. Conf. Pattern Recognition*. Los Alamitos, CA: IEEE Comput. Soc. Press, 1994.
- [6] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 1042–1052, Oct. 1993.
- [7] D. K. Burton, "Text-dependent speaker verification using vector quantization source coding," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-35, pp. 133, 1987.
- [8] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," *Proc. IEEE*, vol. 83, pp. 5, pp. 705–740, 1995.
- [9] I. J. Cox, J. Ghosn, and P. N. Yianilos, "Feature-based face recognition using mixture-distance," in *Computer Vision and Pattern Recognition*. Piscataway, NJ: IEEE Press, 1996.

- [10] D. DeMers and G. W. Cottrell, "Nonlinear dimensionality reduction," in *Advances in Neural Information Processing Systems 5*, S. J. Hanson, J. D. Cowan, and C. Lee Giles, Eds. San Mateo, CA: Morgan Kaufmann, 1993, pp. 580–587.
- [11] H. Drucker, C. Cortes, L. Jackel, Y. Le Cun, and V. N. Vapnik, "Boosting and other ensemble methods," *Neural Comput.*, vol. 6, pp. 1289–1301, 1994.
- [12] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. Boston, MA: Academic, 1990.
- [13] S. Haykin, *Neural Networks, A Comprehensive Foundation*. New York: Macmillan, 1994.
- [14] ———, private communication, 1996.
- [15] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction, and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, pp. 106–154, 1962.
- [16] R. A. Jacobs, "Methods for combining experts' probability assessments," *Neural Comput.*, vol. 7, pp. 867–888, 1995.
- [17] T. Kanade, "Picture processing by computer complex and recognition of human faces," Ph.D. dissertation, Kyoto Univ., Japan, 1973.
- [18] H. Kita and Y. Nishikawa, "Neural-network model of tonotopic map formation based on the temporal theory of auditory sensation," in *Proc. World Congr. Neural Networks, WCNN 93*, vol. II. Hillsdale, NJ: Lawrence Erlbaum, pp. 413–418.
- [19] T. Kohonen, "The self-organizing map," *Proc. IEEE*, vol. 78, pp. 1464–1480, 1990.
- [20] ———, *Self-Organizing Maps*. Berlin: Springer-Verlag, 1995.
- [21] M. Lades, J. C. Vorbrüggen, J. Buhmann, J. Lange, C. von der Malsburg, Rolf P. Würtz, and W. Konen, "Distortion invariant object recognition in the dynamic link architecture," *IEEE Trans. Comput.*, vol. 42, pp. 300–311, 1993.
- [22] Y. Le Cun, "Generalization and network design strategies," Dep. Comput. Sci., Univ. Toronto, Canada, Tech. Rep. CRG-TR-89-4, 1989.
- [23] Y. Le Cun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*, M. A. Arbib, Ed. Cambridge, MA: MIT Press, 1995, pp. 255–258.
- [24] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Handwritten digit recognition with a backpropagation neural network," in *Advances in Neural Information Processing Systems 2*, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, pp. 396–404.
- [25] Y. Le Cun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Neural Information Processing Systems*, vol. 2, D. S. Touretzky, Ed. San Mateo, CA: Morgan Kaufmann, 1990, pp. 598–605.
- [26] T. K. Leen, "From data distributions to regularization in invariant learning," *Neural Comput.*, vol. 3, no. 1, pp. 135–143, 1991.
- [27] S. P. Luttrell, "Hierarchical self-organizing networks," in *Proc. 1st IEEE Conf. Artificial Neural Networks*. London: British Neural Network Soc., 1989, pp. 2–6.
- [28] D. Marr, *Vision*. San Francisco, CA: W. H. Freeman, 1982.
- [29] B. Miller, "Vital signs of identity," *IEEE Spectrum*, pp. 22–30, Feb. 1994.
- [30] B. Moghaddam and A. Pentland, "Face recognition using view-based and modular eigenspaces," in *Automat. Syst. Identification Inspection of Humans*, vol. 2257, 1994.
- [31] K. Obermayer, G. G. Blasdel, and K. Schulten, "A neural-network model for the formation and for the spatial structure of retinotopic maps, orientation, and ocular dominance columns," in *Artificial Neural Networks*, T. Kohonen, K. Mäksä, O. Simula, and J. Kangas, Eds. Amsterdam, Netherlands: Elsevier, 1991, pp. 505–511.
- [32] K. Obermayer, H. Ritter, and K. Schulten, "Large-scale simulation of a self-organizing neural network: Formation of a somatotopic map," in *Parallel Processing in Neural Systems and Computers*, R. Eckmiller, G. Hartmann, and G. Hauske, Eds. Amsterdam, Netherlands: North-Holland, 1990, pp. 71–74.
- [33] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, 1994.
- [34] A. Pentland, T. Starner, N. Etcoff, A. Masoiu, O. Oliyide, and M. Turk, "Experiments with eigenfaces," in *Proc. Looking at People Wkshp, Int. Joint Conf. Artificial Intell.*, Chamberry, France, 1993.
- [35] D. J. Perret, E. T. Rolls, and W. Caan, "Visual neurones responsive to faces in the monkey temporal cortex," *Experimental Brain Res.*, vol. 47, pp. 329–342, 1982.
- [36] Y. Y. Qi and B. R. Hunt, "Signature verification using global and grid features," *Pattern Recognition*, vol. 27, no. 12, pp. 1621–1629, Dec. 1994.
- [37] H. A. Rowley, S. Baluja, and T. Kanade, "Human face detection in visual scenes," School Comput. Sci., Carnegie Mellon Univ., Pittsburgh, PA, Tech. Rep. CMU-CS-95-158, July 1995.
- [38] F. S. Samaria, "Face recognition using hidden Markov models," Ph.D. dissertation, Trinity College, Univ. Cambridge, Cambridge, U.K., 1994.
- [39] F. S. Samaria and A. C. Harter, "Parameterization of a stochastic model for human face identification," in *Proc. 2nd IEEE Wkshp. Applicat. Comput. Vision*, Sarasota, FL, 1994.
- [40] K.-K. Sung and T. Poggio, "Learning human face detection in cluttered scenes," in *Computer Analysis of Images and Patterns*, G. Goos, J. Hartmonis, and J. van Leeuwen, Eds. New York: Springer-Verlag, 1995, pp. 432–439.
- [41] K. Sutherland, D. Renshaw, and P. B. Denyer, "Automatic face recognition," in *Proc. 1st Int. Conf. Intell. Syst. Eng.* Piscataway, NJ: IEEE Press, 1992, pp. 29–34.
- [42] D. L. Swets and J. J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Trans. Pattern Anal. Machine Intell.*, to appear, 1996.
- [43] M. Turk and A. Pentland, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, pp. 71–86, 1991.
- [44] J. Weng, N. Ahuja, and T. S. Huang, "Learning recognition and segmentation of 3-D objects from 2-D images," in *Proc. Int. Conf. Comput. Vision, ICCV 93*, 1993, pp. 121–128.
- [45] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg, "Face recognition and gender determination," in *Proc. Int. Wkshp. Automat. Face Gesture Recognition*, Zürich, Switzerland, 1995.



Steve Lawrence (M'96) received the B.Sc. and B.Eng. degrees *summa cum laude* in 1993 from the Queensland University of Technology, (QUT) Australia. He submitted his Ph.D. thesis in August 1996 at the University of Queensland, Australia.

He is presently working as a Scientist at the NEC Research Institute in Princeton, NJ. His research interests include neural networks, pattern recognition, intelligent agents, and information retrieval.

His awards include a university medal, a QUT award for excellence, ATERB and APA priority scholarships, QEC and Telecom Australia Engineering prizes, and three prizes in successive years of the Australian Mathematics Competition.



C. Lee Giles (S'80–M'80–SM'95) received the B.A. degree from Rhodes College, Memphis, TN, the B.S. degree in engineering physics from the University of Tennessee, Knoxville, in 1969, the M.S. degree in physics from the University of Michigan, Ann Arbor, in 1974, and the Ph.D. degree in optical sciences from the University of Arizona, Tucson, in 1980.

Previously, he was a Program Manager at the Air Force Office of Scientific Research in Washington, D.C. where he initiated and managed research programs in neural networks and in optics in computing and processing. Before that, he was a Research Scientist at the Naval Research Laboratory, Washington, D.C., and an Assistant Professor of Electrical and Computer Engineering at Clarkson University. During part of his graduate education he was a Research Engineer at Ford Motor Scientific Research Laboratory. He is a Senior Research Scientist in Computer Sciences at NEC Research Institute, Princeton, NJ, and an Adjunct Associate Professor at the Institute for Advanced Computer Studies at the University of Maryland, College Park, MD. His research interests include neural networks, machine learning, and artificial intelligence; hybrid systems and integrating neural networks with intelligent systems and agents; applications of intelligent methods to signal processing, communications, computing, networks, time series, pattern recognition, and optical computing and processing. He has published more than 80 journal and conference papers and book chapters in these areas.

Dr. Giles serves on many related conference program committees and has helped organize many related meetings and workshops. He has been an advisor and reviewer to government and university programs in both neural networks and in optical computing and processing. He has served or is currently serving on the editorial boards of *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, *IEEE TRANSACTIONS ON NEURAL NETWORKS*, *Journal of Parallel and Distributed Computing*, *Neural Networks*, *Neural Computation*, *Optical Computing and Processing*, *Applied Optics*, and Academic Press. He is a member of AAAI, ACM, INNS, and OSA.



Ah Chung Tsoi (S'70–M'72–M'84–SM'90), was born in Hong Kong. He received the Higher Diploma in electronic engineering, Hong Kong Technical College, in 1969, and the M.Sc. degree in electronic control engineering and the Ph.D. degree in control engineering from the University of Salford in 1970 and 1972, respectively.

From 1972 to 1974 he worked as Senior Research Fellow at the Inter-University Institute of Engineering Control, University College of North Wales, Bangor, Wales. From 1974 to 1977, he worked as a Lecturer at the Paisley College of Technology, Renfrewshire, Scotland. From 1977 to 1984, he worked as a Senior Lecturer at the University of Auckland, New Zealand. From 1985 to 1990, he worked as a Senior Lecturer at the University College, University of New South Wales. He is presently a Professor in Electrical Engineering at the University of Queensland, Australia. His research interests include aspects of neural networks and their application to practical problems, adaptive signal processing, and adaptive control.



Andrew D. Back (M'96) received the B.Eng. degree with distinction from Darling Downs Institute of Advanced Education (now the University of Southern Queensland), Australia, in 1985, and the Ph.D. degree from the University of Queensland, Australia, in 1992.

He was a Research Fellow with the Defence Science and Technology Organization (DSTO) from 1988 to 1992, and a Research Scientist with the DSTO in 1993. In 1995 he was a Visiting Scientist at the NEC Research Institute, Princeton, NJ. He is currently a Frontier Researcher with the Brain Information Processing Group in the Frontier Research Program at the Institute of Physical and Chemical Research (RIKEN) in Japan. He has published about 30 papers. His research interests include nonlinear models for system identification, time series prediction, and blind source separation.

Dr. Back was awarded an Australian Research Council (ARC) Postdoctoral Research Fellowship in the Department of Electrical and Computer Engineering at the University of Queensland for the period from 1993–1996. He has organized workshops for the Neural Information Processing Systems Conference in 1994, 1995, and 1996 in area of neural networks for signal processing.