

Distortion Tolerant Pattern Recognition Based on Self-Organizing Feature Extraction

Jouko Lampinen, *Member, IEEE*, and Erkki Oja, *Senior Member, IEEE*

Abstract—A generic, modular, neural network-based feature extraction and pattern classification system is proposed for finding essentially two-dimensional objects or object parts from digital images in a distortion tolerant manner. The distortion tolerance is built up gradually by successive blocks in a pipeline architecture. The system consists of only feedforward neural networks, allowing efficient parallel implementation. The most time and data-consuming stage, learning the relevant features, is wholly unsupervised and can be made off-line. The consequent supervised stage where the object classes are learned is simple and fast. The feature extraction is based on distortion tolerant Gabor transformations, followed by minimum distortion clustering by multilayer self-organizing maps. Due to the unsupervised learning strategy, there is no need for preclassified training samples or other explicit selection for training patterns during the training, which allows a large amount of training material to be used at the early stages. A supervised, one-layer subspace network classifier on top of the feature extractor is used for object labeling. The system has been trained with natural images giving the relevant features, and human faces and their parts have been used as the object classes for testing. The current experiments indicate that the feature space has sufficient resolution power for a moderate number of classes with rather strong distortions.

I. INTRODUCTION

A potential application for neural networks is machine vision and image understanding. The essential tasks are to locate and recognize individual objects and to compile a useful interpretation from the objects and their relations. Both of these subtasks have proven to be extremely difficult even in the case of two-dimensional machine vision. The classical approach of programming the *a priori* knowledge or models about the objects into the solution [1] and using artificial intelligence based search methods has severe limitations in computational complexity and in handling all the natural variations in images. In the case of rigid objects, such variations may be due to translation, scaling, rotation, perspective distortions, variations in lighting conditions, variations in the background, or partial occlusion by other objects. In many cases there are also inherent variations within the object classes like in handwritten character recognition or human face recognition. Therefore, using model-free neural networks seems a promising alternative.

Manuscript received December 22, 1992; revised April 28, 1994. This work was supported in part by Tekes (Finland) under the projects "Finsoft" and "Machine Vision in Industrial Automation."

J. Lampinen is with the Lappeenranta University of Technology, Department of Information Technology, P.O. Box 20, 53851 Lappeenranta, Finland. E. Oja is with the Helsinki University of Technology, Laboratory of Computer and Information Science, 02150 Espoo, Finland.

IEEE Log Number 9409151.

A large majority of the present-day neural network research and applications has focused on supervised learning networks like the multilayer perceptron network, the radial basis function network, or the LVQ network. Together with their powerful training algorithms, these networks provide highly efficient model-free methods to design arbitrarily complex nonlinear classifiers iteratively from examples. An important theoretical result motivating the current interest in neural networks is that such networks can, in principle, form class boundaries of any required complexity [10]. There is a great deal of practical sample cases showing the power of neural classifiers trained with the learning algorithms as compared to classical methods; for a review see, e.g., [18].

When the input dimensionality or the abstraction level of the classification increases, however, the shapes of the class regions become increasingly complex, finally requiring networks with huge numbers of parameters. It is well known [27] that even neural networks cannot escape the parameter estimation problem which means that the amount of training data must grow in proportion to the number of free parameters. Consequently, very large amounts of training data and training time are needed in highly complex problems to form the class boundaries. Collecting the samples would clearly be very expensive, if not impossible. This problem cannot be solved with some recent approaches of weight sharing [17] or constraining the number of units or parameters [20] if the large number of parameters is a true indication of the problem complexity.

In conventional pattern recognition, the answer to this problem is to divide the task in two parts: feature extraction, which maps the original patterns or images to a feature space of reduced dimensions, and complexity, followed by classification in this space. There is no well-developed theory for feature extraction; most features are very application oriented and often found by heuristic methods and interactive data analysis [24]. An important basic principle is that the features must be independent of class membership because, by definition, at the feature extraction stage the membership is not yet known. This implies that if any learning methods are used for developing the feature extractors, they should be unsupervised in the sense that the target class for each object is unknown.

Our approach is to likewise divide the object recognition problem to separate feature extraction and classification tasks, but to use only neural networks for both. The total system is then a multilayer collection of neural modules. To handle the large variability of natural scenes, the system must have a large

number of free parameters in the early stages, and estimating the parameters will require much data. This dilemma is now solved by using unsupervised learning techniques in the early feature extraction stages to reduce the number of degrees of freedom in the input data. Then the final supervised classifiers can have a much smaller number of free parameters and thus require a smaller amount of preclassified training samples. This scheme is especially suitable for general scene analysis, since it is fairly inexpensive to collect large amounts of image data to be used in unsupervised training, as long as the images need no manual analysis and classification.

The whole feature extraction-classification system forms a pyramid, where the number of neurons and connections decrease and the connections become more adaptive in higher layers. The major blocks of the system are:

- 1) Fixed layer doing Gabor transformation to the image on different resolutions using self-similar kernels.
- 2) Unsupervised layer clustering the Gabor coefficients to natural clusters while introducing minimal additional distortion. The clustering is accomplished by a multilayer self-organizing map (MSOM).
- 3) Supervised layer carrying out the classification. The classifier is based on subspace networks which operate on locally collected histograms of the previous level outputs.

Special care has been taken to choose the individual modules in such a way that the characteristics of the output of each module are optimally suited to the processing in the next stage. Thus MSOM is well adapted to find the nonconvex clusters given by the Gabor output vectors, and the subspace network is an ideal classifier for feature histograms.

In Section II, each block is described in turn. Experimental results using natural images are reported in Section III, followed by conclusions in Section IV.

II. NEURAL NETWORK-BASED OBJECT RECOGNITION SYSTEM

A. Producing the Primary Features with Gabor Transformation

Gabor filters were introduced to image processing by Granlund [9] and analyzed by Daugman [5]. The two-dimensional Gabor filters are orientation and frequency sensitive band pass filters which are optimally localized in both spatial and frequency domains [4], making them suitable for extracting the orientation-dependent frequency contents, i.e., edge-like features, from as small an area as possible.

The forms of the Gabor kernels are spatial sinusoids localized by a Gaussian window. They operate directly on the digital image in spatial domain. The filters constitute a self-similar family where all the filters are dilations, translations, and rotations of each other, with the impulse response (centered at the origin) given by

$$\psi(f, \theta, x, y) = \exp\left(i(f_x x + f_y y) - \frac{f^2(x^2 + y^2)}{2\sigma^2}\right) \quad (1)$$

where

$$f_x = f \cos \theta, \quad f_y = f \sin \theta.$$

There x and y are the digital pixel coordinates in the image. The central frequency of the pass band is f , the spatial orientation is θ , and the parameter σ determines the bandwidth of the filter. Throughout this work, the value $\sigma = \pi$ was used.

In the early feature extraction module of the proposed system, directly operating on the original gray-scale images, several Gabor filters with different orientations and different resolutions were used. To cover the whole frequency and orientation ranges, the following central frequencies and orientations were used

$$f = f_k = \frac{\pi}{2^k}, \quad k = 0, 1, \dots$$

$$\theta = \theta_t = \frac{\pi t}{8}, \quad t = 0, \dots, 7$$

where k determines the frequency band (resolution of details extracted by the filters) and t enumerates the orientations.

The multiresolution representation of an image by Gabor filtering forms a pyramid of coefficients, a four-dimensional structure indexed by the image coordinates (x, y) , the resolution level k , and the orientation θ . Note that the resolution level k is related to the wavelength $\lambda_k = 2\pi/f_k$ of the filter, and thus to the spatial size, by

$$k = \log_2 \left(\frac{\lambda_k f_0}{2\pi} \right).$$

This means that the Gabor pyramid contains a somewhat similar geometric mapping as the log-polar transform, with respect to every pixel location. The log-polar transform $(x, y) \mapsto (\theta, \log r)$ maps the scaling and rotation of the image to translations. In the Gabor pyramid, scaling of the image shifts the response pattern along the discrete resolution axis k and rotation along the orientation axis θ , respectively.

The complex Gabor filters capture the whole frequency spectrum, both amplitude and phase. We reject the phase information by computing and storing only the amplitude of the convolution result, which changes smoothly over the image. The amplitude of the Gabor transformation has been found to be a robust, distortion tolerant space for pattern recognition in many studies, e.g., [2], [3], and [8].

Gabor filters with eight orientations were first applied to each pixel location (x, y) in the original image, giving eight scalar values (the outputs of the filters) at each location. The image was then downsampled by the usual method of combining every four pixels in a square into one pixel whose gray level was taken to be the discretized average of the four original pixels. From an $N \times N$ image, this produces a new $N/2 \times N/2$ image with half the resolution. The same eight Gabor filters were applied to this image, etc. Finally, the output of the Gabor feature extraction stage is a pyramid structure containing eight real coefficients for each pixel in the image on several resolutions. These feature vectors are similar to the Gabor Jets of [2] and [8]. This stage is schematically shown on the left part of Fig. 1.

B. Clustering the Features: Properties of the SOM

The self-organizing map (SOM) [13] is a neural network that can learn a topology preserving mapping from a high

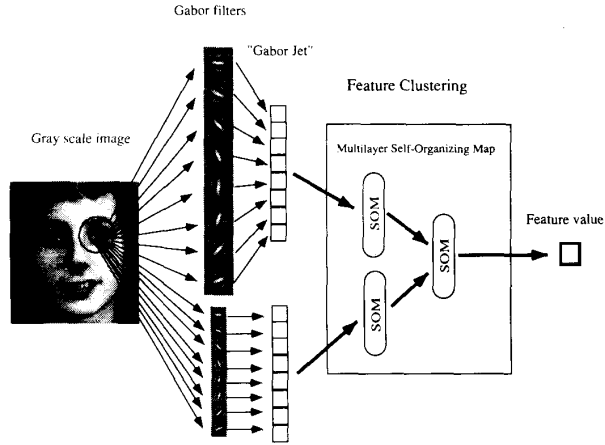


Fig. 1. Schematic drawing of the feature extraction system. Left part: eight-dimensional Gabor vectors at two resolutions are extracted from every pixel location in the 128×128 digital image. Right part: the two-layer SOM produces a feature value $c(p)$ for each pixel location p .

dimensional space to a lower dimensional map lattice. An essential advantage from the topological ordering of the map is achieved in hierarchical systems of SOM's, e.g., in hierarchical vector quantization [19]. In [16], the authors analyzed why the MSOM is suitable for clustering: it will produce "natural" clusters whose form is dynamically matched to the probability density structure of the input samples.

The probability density of the Gabor coefficients on natural images, produced in the previous feature extraction stage, is very sparse [5], with most of the coefficients being near zero and only a few of them carrying most of the relevant information. Thus it is reasonable to recode the data to a more compressed form. The coding should be tolerant to small distortions in the image, specifically changes of scale and orientation, while shift invariance can be taken into account by shift invariant transformations applied to the coded image.

Small variations in scale and orientation in the original image will have a characteristic effect on the geometry of the Gabor feature vectors because of the structure of the Gabor pyramid. When scale and orientation are smoothly changed, the feature vector corresponding to a given feature will move in the feature space along an elongated low dimensional trajectory. In coding, the objective is to give unique codes to such elongated nonconvex clusters. The approach taken here is to first span the clusters by the SOM and then combine several map elements to a larger cluster by an MSOM.

In this section, some basic properties of the SOM are reviewed. Specifically, it is shown that if the input data contains such compact elongated structures, the map will place neighboring units sequentially along the structure. This is the starting point for the application of MSOM for this recoding.

Assume a D -dimensional SOM lattice of neuron units with weight vectors $\mathbf{m}_r$, indexed by a D -vector $\mathbf{r}$. Assume that the neighborhood function h of the map algorithm is shift invariant over the lattice, $h(\mathbf{r}, \mathbf{q}) = h(\mathbf{r} - \mathbf{q})$. Shortly, the basic SOM training algorithm then randomly picks an input vector $\mathbf{x}$ from some unknown probability distribution, chooses

the weight vector $\mathbf{m}_b$ that is closest to $\mathbf{x}$ in Euclidean distance, and updates all weight vectors according to

$$\mathbf{m}_r := \mathbf{m}_r + \alpha h(\mathbf{r} - \mathbf{b})(\mathbf{x} - \mathbf{m}_r). \quad (2)$$

There α is a gain parameter slowly decreasing as training goes on.

A possible way to analyze where the weight vectors will converge is to consider algorithm (2) as gradient descent minimization of a cost function and solve for the extremal points of this function. For a continuous density of the input vectors $\mathbf{x}$, an exact cost function that would be minimized by SOM training is not known [14]. It has been shown by Erwin *et al.* [6], for the one-dimensional map, that the map algorithm is not a gradient descent step of any such cost function, although a set of cost functions, one for each neuron unit, can be defined. In clustering, however, the usual assumption is that the set to be clustered is a fixed finite set of vectors $\mathbf{x}$, each having an equal probability. Then the input probability distribution is discrete uniform, and for this case it has been shown by Ritter [26] (see also [19]) that, assuming the step size α is small, the energy or cost function that the map algorithm tries to minimize is

$$C = \sum_{\mathbf{q}} \sum_{\mathbf{r}} h(\mathbf{q} - \mathbf{r}) \sum_{\mathbf{x} \in V_r} \|\mathbf{x} - \mathbf{m}_q\|^2 \quad (3)$$

where $\mathbf{r}$ and $\mathbf{q}$ are vector indexes of the units on the map and V_r is the set of points $\mathbf{x}$ that are mapped to (closest to) the unit $\mathbf{r}$.

Geometrically, the criterion is the weighted sum of squared distances of all the input vectors $\mathbf{x}$ to all the map weight vectors $\mathbf{m}_q$. Each distance is weighted more heavily if the weight vector corresponds to the best matching unit for that input vector $\mathbf{x}$ or one of its closest neighbors in the neuron lattice. Thus the map finds as compact a set of points as possible to be mapped to each neighborhood during the training.

In a local minimum of (3), the gradient for the weight vector of unit $\mathbf{m}_q$ is zero

$$dC/d\mathbf{m}_q = -2 \sum_{\mathbf{r}} h(\mathbf{q} - \mathbf{r}) \sum_{\mathbf{x} \in V_r} (\mathbf{x} - \mathbf{m}_q) = 0. \quad (4)$$

If we denote by $N(V_r)$ the number of points in V_r and by $\mathbf{c}_r$ the mean of the points mapped to unit $\mathbf{r}$, i.e., $\mathbf{c}_r = \frac{1}{N(V_r)} \sum_{\mathbf{x} \in V_r} \mathbf{x}$, then we have from (4)

$$\sum_{\mathbf{r}} h(\mathbf{q} - \mathbf{r}) N(V_r) (\mathbf{c}_r - \mathbf{m}_q) = 0. \quad (5)$$

It has been shown by Erwin *et al.* [7], again for the one-dimensional case, that many stationary states satisfying (4) can exist, depending on the shape of the neighborhood function $h(\mathbf{q} - \mathbf{r})$. If the neighborhood function is a broad enough Gaussian function, or in general a convex function over the neuron lattice, then the only stationary states are the ordered states in the one-dimensional map. An analogy of this result for the general D -dimensional map is not known. It can be safely assumed, however, that the final state of the map after

convergence should be among the stationary states satisfying (5), hence

$$\mathbf{m}_q = \frac{\sum_r h(\mathbf{q} - \mathbf{r}) N(V_r) \mathbf{c}_r}{\sum_r h(\mathbf{q} - \mathbf{r}) N(V_r)}.$$

Assuming $h(\mathbf{q} - \mathbf{r})$ smooth like a broad Gaussian and $N(V_r)$ roughly equal, which usually happens in SOM training, this implies a certain smoothness to the map in the stationary state. This becomes especially clear if we assume that the units become equiprobable, i.e., each Voronoi set V_r contains an equal number of vectors. Under this assumption and assuming $\sum_q h(\mathbf{q}) = 1$, it follows further

$$\mathbf{m}_q = \sum_r h(\mathbf{q} - \mathbf{r}) \mathbf{c}_r. \quad (6)$$

This is a discrete convolution of the sequence h with the set of mean vectors. If the neighborhood function has a typical Gaussian shape, then the set of weight vectors are in fact obtained from the centroids $\mathbf{c}_r$ by low-pass filtering.

In practice, the neighborhood $h(\mathbf{q} - \mathbf{r})$ is smoothly decreased during SOM training. Equation (6) is an implicit constraint, imposing a certain "stiffness" to the map lattice after learning. If the data contains compact structures, then at some neighborhood size the map will place the units sequentially along the structure. This is how the map finds and preserves the local ordering relations of the training samples. This is exactly the kind of coding that is suitable for the low-dimensional structures representing distorted image features in the Gabor vectors.

C. Multilayer SOM

The MSOM is based on the fact that the index of the best matching unit b for input vector $\mathbf{x}$ can be considered as the numerical output from the SOM. This is in steep contrast to any biological models but can be used to an advantage in artificial applications of the SOM. In the MSOM, a second-layer map $\mathbf{m}^2$ simply takes as inputs the outputs b of the first-layer map $\mathbf{m}^1$ and clusters them. Now the region in the original input space which is mapped to one $\mathbf{m}^2$ unit is the union of all the regions of the corresponding units in $\mathbf{m}^1$. While a one-layer SOM always forms convex clusters given by the Voronoi cells, the clusters given by the two-layer SOM are unions of convex volumes. They can approximate any region in the input space to an arbitrary accuracy, given enough units in the first-layer SOM. This gives a certain analogy to the ability of a two-layer MLP network to form arbitrarily complex class regions as combinations of the simple half-spaces given by the first layer Perceptrons.

The clustering performed by the MSOM corresponds roughly to classical clustering methods (e.g., ISODATA) when the distance of a point from the cluster is defined as the weighted distance from all the points in the cluster [16]. With MSOM, however, the clustering of a given input vector $\mathbf{x}$ requires only two nearest neighbor searches among the weight vectors of the second- and first-layer maps, respectively, instead of computing all the distances between the input vectors themselves, and the extent of weighting is adapted

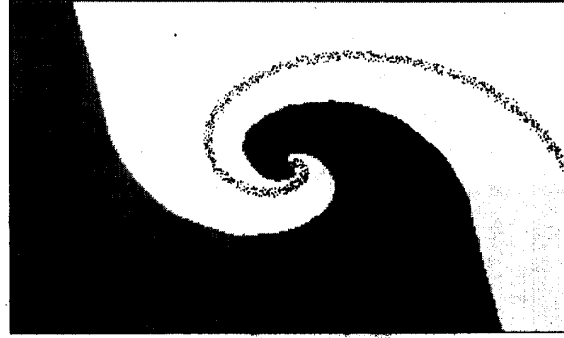


Fig. 2. Example of the clusters formed by the MSOM for a spiral shaped input data distribution. The two regions with different gray levels are mapped to the two neurons in the second-layer map.

automatically to the data distribution. In addition, the system minimizes the total quantization error introduced by both the layers as shown by Luttrell [19]. Among clustering methods this is a rather unique balance between preserving maximal information and collapsing the data to natural clusters.

Fig. 2 shows a very simple example about the capabilities of a two-layer SOM in clustering. The input probability distribution contains continuous spiral-shaped structures that are partially interleaved. The continuous line within the spirals shows the units and neighborhood connections of the first layer 30-unit map $\mathbf{m}^1$ adapted to the data distribution. The second layer had only two units splitting the $\mathbf{m}^1$ map to two halves. Clustering by the MSOM is based on continuous compact areas, and it thus easily finds the natural cluster boundary. This kind of clustering which can adapt itself to elongated clouds of input data is especially well suited for clustering the eight-dimensional Gabor vectors obtained at the first stage of our system.

The right-hand part of Fig. 1 shows the principle of the MSOM in the feature extractor. The eight-dimensional Gabor coefficient vectors at each resolution (in this case, two resolutions) are input to the first-layer SOM. The size of the first layer maps was 10×10 , and the output for each eight-dimensional input vector from each map is the two-dimensional location index of its best-matching unit. Two such indexes from the two maps comprise a four-dimensional vector of integer values, which is input to the second-layer map. The second-layer map was one-dimensional and had 100 units, and the output is again the one-dimensional index of the best-matching unit. Each unit on the second-layer map learns to code a certain combination of the codes given by the first-layer maps.

The output of the feature extraction stage is a "feature image" $c(\mathbf{p})$ where each image location $\mathbf{p} = (x, y)$ is mapped by the MSOM to a unit c of the second-layer map. This unit represents a cluster, possibly of a complicated nonconvex shape, in the original space of Gabor amplitudes in different orientations and resolutions. Because index c is an integer on the interval $[1, 100]$, it can be plotted in gray scale at each location $\mathbf{p}$, giving the feature image in the upper part of Fig. 3.

At the training phase, both the first- and the second-layer maps were trained with the usual unsupervised SOM algorithm

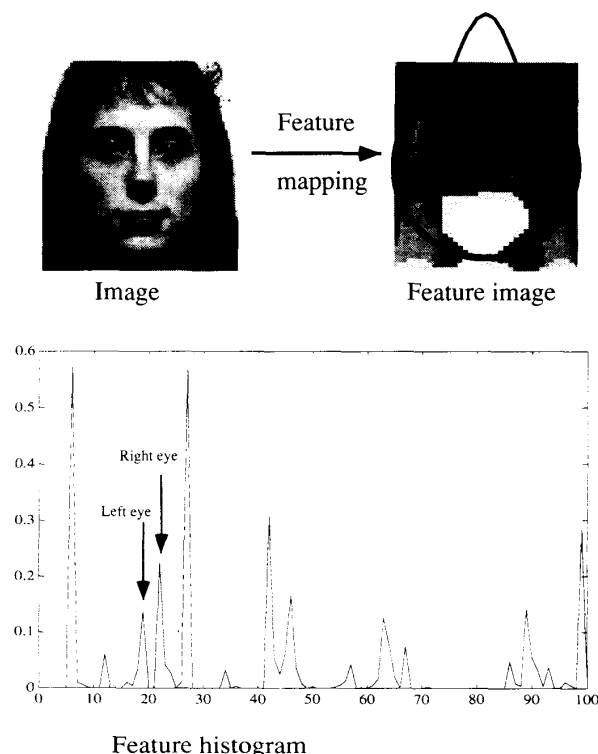


Fig. 3. Upper part: an example image and the feature image. The image was a part of a 128×128 image. The 100 feature values are represented by gray levels in the feature image. The circle gives the approximate face area to be used in computing the feature histogram. Lower part: the Gaussian weighted feature histogram. The Gaussian weight function had width $R = 50$ and was centered as shown by the circle of radius R in the feature image.

[12], using as inputs to the first-layer map a large number of eight-dimensional Gabor vectors obtained from a number of images. After the first-layer map had been trained, the second-layer map was trained likewise taking the inputs from the first-layer map outputs. More details about the training material and results are given in Section III.

D. Distortion Tolerant Classification with Histograms and Subspace Networks

In object recognition, there are several invariance problems to be taken into account. The three basic ones are shift, rotation, and size invariance. For nonrigid objects and in three-dimensional object recognition, many other kinds of distortions are possible, too. The feature extraction method described above can tolerate certain variances, while others must be left to the classifier.

Notably, size invariance is taken into account by the multiresolution Gabor filters, since the feature space consists of several layers in resolution hierarchy. This helps size invariant recognition: according to our experiments the features are usually tolerant to size variations of less than a half octave, which is the largest possible change before the object appears on the next lower or higher resolution.

Feature extraction based on Gabor filters is not shift invariant; on the contrary, it relates the features to as small an area as possible. In the case of several objects in the image, this increases the likelihood that the features come from one object only and helps in classification. The distortion invariance of any method is always limited to the spatial support of its receptive field, and any distortions larger than that will affect the output.

In the feature extraction used here the problem of shift invariance is not very prominent because the hierarchical clustering absorbs most small variations and expands the spatial support of clear details; see [15]. Consequently, good recognition results have been achieved by direct template matching in the feature space [15]. But since the features are local, the problem of distortions in feature positions is there and should be taken into account in classification.

There are three main approaches to tackle this type of distortions:

- 1) Minimizing editing distances, based on the requirement that the important features must appear in correct order [25].
- 2) Planar graph matching algorithms; see, e.g., [3].
- 3) Translation invariant transformations, based on the assumption that the important features must appear inside a certain area. The transformation may be, e.g., a histogram of the underlying features.

The third approach was chosen in the current system because of most promising test results from the three categories and also because of the possibility of fast parallel implementation with only feedforward networks. Both the other categories require time consuming iterations to find the optimal match.

For a small number of objects in the training data, there will be specific units in the MSOM map for the details of each object, and the recognition can be based on histograms from the whole object. For larger or more complicated objects, more position information must be maintained. Then the histogram is collected from several locations with smaller windows, and the vectors are stacked together.

The input vector for the classifier at each position is now a local histogram of MSOM codes, which is weighted by a Gaussian window. Denote, again, by $\mathbf{p} = (x, y)$ the position coordinates in the feature image $c(\mathbf{p})$, which was the outcome of the previous feature extraction stage (see Fig. 3). We smooth the histogram by a weight function $\Omega(\mathbf{p} - \mathbf{p}') = \exp(-\frac{\|\mathbf{p} - \mathbf{p}'\|^2}{2R^2})$, with R determining the size of the window. Then the histogram of MSOM codes at image position $\mathbf{p}$ is given by the one-dimensional array

$$w_i(\mathbf{p}) = \sum_{\mathbf{p}': c(\mathbf{p}')=i} \Omega(\mathbf{p} - \mathbf{p}'), \quad i = 1, \dots, 100. \quad (7)$$

Using the Gaussian weighted histogram as the final feature vector complies well with using the Gabor filter amplitudes as the primary features, since they are basically histograms of certain frequencies inside a Gaussian window.

Fig. 3 shows an example of the feature image and the weighted histogram which was centered at the face. The central point $\mathbf{p}$ for computing the histogram is the center of the circle



Fig. 4. Images used for unsupervised training of the feature maps.

in the feature image. The radius of the circle is the standard deviation of the Gaussian, in this case $R = 50$ pixels.

Subspace methods [21] and [23] are efficient and computationally simple classification methods which have been developed for histogram and spectrum type data. Each class is defined as a linear subspace and represented by a set of orthogonal base vectors. The outputs of the net are the projections of the input vector onto the class subspaces, and an unknown input vector is classified according to the largest output.

The neural subspace classifier is based on the principal component analysis (PCA) network [22]. In [22] it is shown that with the learning rule

$$M_{k+1} = M_k + \alpha(I - M_k M_k^T) x_k x_k^T M_k \quad (8)$$

the columns of matrix M_k become orthonormal and span the same subspace as the most dominant eigenvectors of the autocorrelation matrix of the input data $E\{xx^T\}$, i.e., the space of the largest principal components of the inputs.

Now each class i is represented by its own weight matrix M_i , whose columns span the class subspace. The simplest approach for finding the class subspaces is to compute the largest principal components for each class separately using (8). Better results are obtained by the learning subspace method (LSM) [11] that finds subspaces minimizing classifications errors. The neural implementation of LSM [22] requires only a small change in the update rule of (8). If the classification label of an input is a and the correct class is b , then M_a is updated with a negative learning parameter α and M_b with a positive one.

In the current problems, one- and two-dimensional class subspaces were found to be sufficient. In most of the experiments, even the principal component subspaces were able to

TABLE I
CLASSIFICATION RESULTS OF THE SUBSPACE NETWORK. THE IMAGES ARE GIVEN IN FIG. 5. THOSE MARKED WITH BOLDFACE ARE THE TRAINING IMAGES FOR THE CLASSIFIER AND THE OTHERS ARE TEST IMAGES. THE NUMBERS GIVE THE PROJECTIONS OF THE IMAGE HISTOGRAMS ONTO THE THREE CLASS SUBSPACES

T1	T2	T3	T4
0.9413	0.8687	0.9524	0.6623
0.2429	0.1074	0.1039	0.1402
0.2546	0.1182	0.0996	0.1447
T5	T6	T7	T8
0.3804	0.4751	0.7026	0.6575
0.3283	0.2040	0.3728	0.3631
0.2200	0.1682	0.4532	0.3650
K1	K2	K3	K4
0.2182	0.1671	0.4749	0.1100
0.6283	0.7861	0.5223	0.8200
0.5061	0.5802	0.4719	0.5644
J1	J2	J3	J4
0.1497	0.2235	0.1158	0.1661
0.5113	0.2236	0.6297	0.5488
0.7355	0.4563	0.8066	0.8380

separate the classes, but for better resolution the LSM was used.

III. EXPERIMENTAL RESULTS

The Gabor filter outputs are low level features, essentially representing edge-like structures of varying sharpness and orientation. Such features are expected to exist in practically any real-world scene. It is easy to collect a large number of Gabor coefficient vectors from suitable training images. Fig. 4 shows a typical collection of images, from which a number of eight-dimensional Gabor coefficient vectors were collected by random sampling.

In the unsupervised clustering block, the Gabor vectors were clustered by a 10×10 map separately on each resolution and then two successive resolutions were combined by a 100-unit one-dimensional map. The processing thus far needs no manual sampling or training and is fully automatic and unsupervised. Varying the training image from that of Fig. 4 to a different collection of natural images had only a marginal effect on the clusters found by the feature extraction stage. At this stage, the training material need not be especially adapted to the final object classes that we want to analyze. The reason is that different training images have many similar primitive details, like edges, corners, and textures, so that the 100 features will become tuned to such generic details. The balance between generalization (i.e., generic distortion tolerant features) and specialization (i.e., features related to certain objects) could be controlled by the number of features.

That was also the case here: Figs. 5–6 show some of the test images to be used in the supervised classification experiments. The test objects were human faces or their parts. The goal was to classify either the whole face to one of the “person” classes or, e.g., the eye region to either the “right eye” or the “left eye” class. The number of persons (classes) varied from three to 19. Very little was done to normalize the conditions in the images to allow fairly large natural variations. Face recognition was selected as a test case because human faces clearly contain distinct features that characterize the face independently of

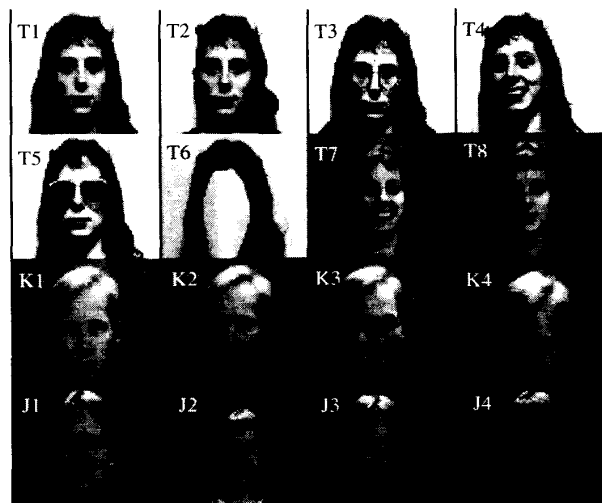


Fig. 5. Test images of three persons (T, P, and J) to be used for classification experiments. The results are given in Table I.

changes in imaging conditions, and the purpose was to verify whether the MSOM network can find such features. The classifier used in the experiments is fairly primitive since all location information is lost from the features, and only the relative frequencies of the features in the image are considered.

In the case of Fig. 5, two training images were selected for each person, and they were subjected to the Gabor transformations and the coding using the MSOM trained previously. In the experiments, the location of the face was detected with a simple template matching procedure: the correlation template consisted of the face area in a lower resolution feature image, where details, such as eyes or mouth, map to same features for different persons. The subspace classification was applied in the local maxima of the template correlation. In the images of Fig. 5, the error of the face location detector was less than 5% of the face size (error was less than three pixels while the size of the faces was about 60 pixels).

The feature histogram was formed from a Gaussian window with $R = 20$, which covered approximately the face region. Now the histogram of these features is a 100-dimensional vector that was passed over to the subspace classifier. Table I shows the network outputs or subspace projections for the classes T, K, and J, respectively. The training samples are denoted by bold face letters in the table. As seen from Table I, all the test images were correctly classified with relatively large margins. Even the partially occluded images (T5 and T6) were correctly classified because the histogram of the remaining features was closer to the correct class than to the others.

The Gabor filters are not scale invariant, since scaling of the image shifts the whole spectrum and the amount of energy in the filter pass-bands changes depending on the frequency contents on other bands. The system is merely scale tolerant; some features span, due to the MSOM, long clusters reaching the invariance over the one octave interval, and some features

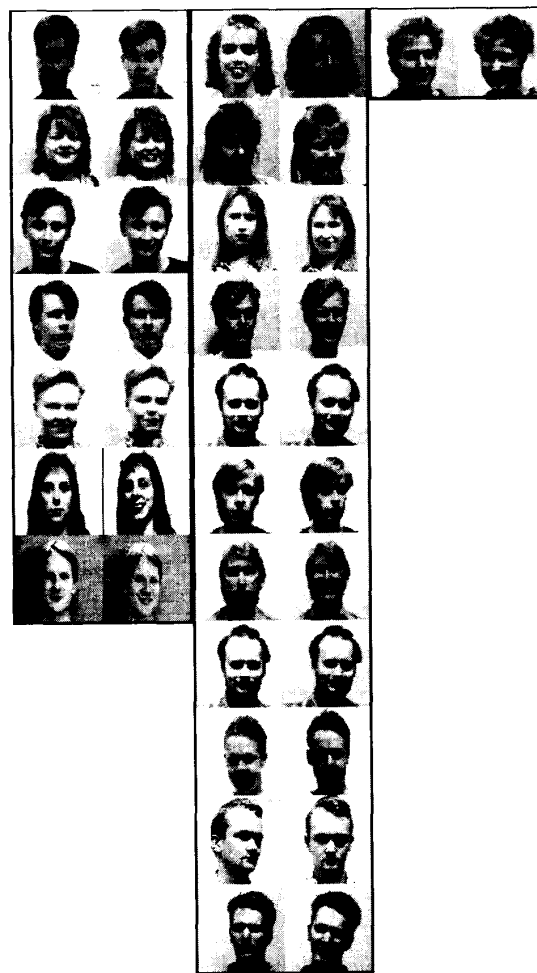


Fig. 6. Another set of 19 face images to be used for classification experiments.

are much more sensitive to scaling. Thus, and according to most experiments, scaling causes projections to all the classes to decrease but their ratios still facilitate correct classification (see, e.g., J2).

Fig. 6 shows a gallery of 19 test images. The feature extraction was again done exactly like before, using the fixed Gabor filters and the MSOM previously trained with Fig. 4. One of the images of each person was used as the training image for the classifier and the other for testing. Objects in the first column of Fig. 6 were classified 100% correct in a robust shift invariant manner, tolerating a large variation in the location of the feature histogram window. Objects in the middle column required that the feature histogram was calculated from the same relative location in both training and test images, thus the recognition was not shift invariant. The minimum size of the area giving correct classification was two pixels for the bottom image in Fig. 6. Successful recognition then requires either that the face locations can be found to the two-pixel accuracy, which is difficult in practice, or that

the histogram is computed and classified at several locations with a maximum spacing of two pixels. Since the histogram computation and subspace classification are both rather light operations, the latter method was used to locate the faces in these experiments. The face in the third column in Fig. 6. was mapped to exclusive features and could not be classified correctly.

IV. CONCLUSION

A pipelined neural network based pattern recognition system was discussed. Each stage, except the lowest level, used learning. The main emphasis was to separate the feature extraction and classification stages so that no supervised training would be needed in feature extraction, allowing a large amount of virtually unprocessed training image material to be used in the first automatic training stage. It was shown that this material need not be the same as that used in training the classifier, as long as similar elementary features like edges are present in both.

To learn the basic features, a fixed self-similar family of Gabor filters with varying resolutions and orientations were applied at every pixel location in the training images containing natural and artificial scenes. Then an MSOM was trained to map continuums in the input data space of the multidimensional Gabor feature vectors to clusters, while causing minimal additional quantization error. This provides a feature space of cluster codes that is distortion tolerant in small scale and offers features specific enough to facilitate classification based on local feature histograms. A learning subspace network classifier was shown to be well suited for such histogram data. Images of faces were chosen as test material for the classification experiments, with good classification results.

Whether this scheme can provide sufficient recognition capability to serve as a universal set of features in general scene analysis, and how the MSOM maps should be structured, will require further analysis. For a limited number of object classes, the feature space seems to be adequate for reliable and very fast recognition with a minimal amount of supervised learning.

The current system has no feedback in any stage. Adding feedback to control the focusing of attention when proceeding from coarse levels to higher resolutions and finer details will be a future extension.

REFERENCES

- [1] D. Ballard and C. Brown, *Computer Vision*. Englewood Cliffs, NJ: Prentice-Hall, 1979.
- [2] J. Buhman, J. Lange, and C. von der Malsburg, "Distortion invariant object recognition by matching hierarchically labeled graphs," in *Proc. IEEE IJCNN-89*, Washington DC, June 18-22, 1989, pp. 155-159.
- [3] J. Buhman, M. Lades, and C. von der Malsburg, "Size and distortion invariant object recognition by hierarchical graph matching," in *Proc. IEEE IJCNN-89*, San Diego, CA, Jan. 1989, pp. 411-416.
- [4] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency and orientation optimized by 2D visual cortical filters," *J. Opt. Soc. Am. (A)*, vol. 2, no. 7, pp. 1160-1169, 1985.
- [5] ———, "Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression," *IEEE Trans. ASSP*, vol. 36, no. 7, pp. 1169-1179, Jul. 1988.
- [6] E. Erwin, K. Obermayer, and K. Schulten, "Self-organizing maps: Ordering, convergence properties and energy functions," *Biol. Cyb.*, vol. 67, no. 1, pp. 47-55, 1992.
- [7] ———, "Self-organizing maps: Stationary states, metastability and convergence rate," *Biol. Cyb.*, vol. 67, no. 1, pp. 35-45, 1992.
- [8] K. A. Flator and S. T. Toborg, "An approach to image recognition using sparse filter graphs," in *Proc. IEEE IJCNN-89*, Washington DC, June 18-22, 1989, pp. 313-320.
- [9] G. H. Granlund, "In search of a general picture processing operator," *Comp. Graph. Image Proc.*, vol. 8, pp. 155-173, 1978.
- [10] K. Hornik, M. Stinchcombe, and H. White, "Multi-layer feedforward networks are universal approximators," *Neural Networks*, vol. 2, pp. 359-366, 1989.
- [11] T. Kohonen, H. Riittinen, M. Jalanko, E. Reuhkala, and S. Haltsonen, "A thousand-word recognition system based on the learning subspace method and redundant hash addressing," in *Proc. 5 ICPR*, Miami Beach, FL, Dec. 1-4, 1980, pp. 158-165.
- [12] T. Kohonen, "Self-organized formation of topologically correct feature maps," *Biol. Cyb.*, vol. 43, pp. 59-69, 1982.
- [13] ———, *Self-Organization and Associative Memory*. Berlin: Springer-Verlag, 1984.
- [14] ———, "Self-organizing maps: Optimization approaches," in *Proc. ICANN-91*, Espoo, June 24-28, 1991, pp. 981-990.
- [15] J. Lampinen and E. Oja, "Distortion tolerant feature extraction with Gabor functions and topological coding," in *Proc. INNC-90*, Paris, July 9-13, 1990, pp. 301-304.
- [16] ———, "Clustering properties of hierarchical self-organizing maps," *J. Math. Imaging and Vision*, vol. 2, pp. 261-272, 1992.
- [17] Y. LeCun, B. Boser, J. Denker, D. Henderson, R. Howard, W. Hubbard, and L. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541-551, 1989.
- [18] R. P. Lippman, "Pattern classification using neural networks," *IEEE Commun. Mag.*, pp. 47-64, Nov. 1989.
- [19] S. P. Luttrell, "Self-organization: A derivation from first principles of a class of learning algorithms," in *Proc. IEEE IJCNN-89*, Washington DC, June 18-22, 1989, pp. 495-498.
- [20] S. Nowlan and G. Hinton, "Simplifying neural networks by soft weight-sharing," *Neural Computation*, vol. 4, pp. 473-493, 1992.
- [21] E. Oja, *Subspace Methods of Pattern Recognition*. Research Studies Press Ltd., 1983.
- [22] ———, "Neural networks, principal components and subspaces," *Int. J. Neural Systems*, vol. 1, no. 1, pp. 61-68, 1989.
- [23] E. Oja and T. Kohonen, "The subspace learning algorithm as a formalism for pattern recognition and neural networks," in *Proc. IEEE 1988 ICNN*, San Diego, CA, July 24-27, 1988, pp. 277-284.
- [24] T. Pavlidis, *Structural Pattern Recognition*. New York: Springer-Verlag, 1977.
- [25] E. Reuhkala, "Recognition of strings of discrete symbols with special application to isolated word recognition," *Acta Poly. Scand. Ma 38. Dr. Tech. Dissertation*, Helsinki Univ. of Tech. 1983.
- [26] H. Ritter and K. Schulten, "Kohonen's self-organizing maps: Exploring their computational capabilities," in *Proc. IEEE IJCNN-88*, San Diego, vol. 1, July 24-27, 1988, pp. 109-116.
- [27] B. Widrow and M. Lehr, "30 years of adaptive neural networks: Perceptron, madaline, and backpropagation," *Proc. IEEE*, vol. 78, no. 9, pp. 1415-1442, 1990.



Jouko Lampinen (S'89-M'93) received the M.Sc. degree in applied physics and electronics from University of Kuopio, Finland, in 1988 and the Dr. Tech. degree in computer science from Lappeenranta University of Technology, Finland, in 1993. Dr. Lampinen is currently an acting Associate Professor in the Department of Information Technology at the Lappeenranta University of Technology, Finland. His present research interests are in applying neural networks in computer vision and modeling and controlling nonlinear processes with neural networks.



Erkki Oja (S'76-M'77-SM'90) was born in Helsinki, Finland, in March, 1948. He received the M.Sc. degree and the Dr. Tech. degree (with distinction) from Helsinki University of Technology in 1972 and 1977, respectively.

He was a visiting Scientist at Brown University, Providence, RI, in 1977-78, and at Tokyo Institute of Technology, Japan, in 1983-84. For the academic year 1990-91 he held the Toshiba Visiting Professor's Chair at Tokyo Institute of Technology, Department of Computer Science. From 1987-1993

he was Professor of Computer Science at Lappeenranta University of Technology, Finland. Presently he is Professor of Computer Science at the Laboratory of Computer and Information Science, Helsinki University of Technology, Finland. His present research interests are in applying neural networks to computer vision and the study of subspace, principal component analysis, and self-organizing networks.

Dr. Oja has served in the scientific and organization committees of a number of conferences, recently including IEEE ICNN-93, WCNN-93, IJCNN-93, ECAI-94, and ICPR-94. He was general chairman of the Sixth Scandinavian Conference on Image Analysis in 1989. He is member of ACM, INNS, Finnish Academy of Sciences, and Finnish Academy of Technical Sciences, past chairman of the Finnish Pattern Recognition Society, past vice president of the IEEE Finland Section, member of the Governing Board of the International Association of Pattern Recognition, and member of the Council of the European Neural Network Society. He is member of the editorial boards of *Neural Networks*, *Neural Computation*, and IEEE TRANSACTIONS ON NEURAL NETWORKS. He is the author of a number of journal papers and book chapters on pattern recognition, computer vision, and neural computing, and the book *Subspace Methods of Pattern Recognition*, which has been translated into Chinese and Japanese.