Applications of Artificial Neural Networks II,
Steven K. Rogers, Editor, Proc. SPIE 1469, Orlando, April 2-5, 1991, pp. 832-842.

Feature extractor giving distortion invariant hierarchical feature space

Jouko Lampinen

Lappeenranta University of Technology
Department of Information Technology
PO Box 20, SF-53851 Lappeenranta, Finland

E-mail: jouko@deneb.it.lut.fi

ABSTRACT

A block structured neural feature extraction system is proposed whose distortion tolerance is build up gradually by successive blocks in a pipeline architecture. The system consists of only feedforward neural networks, allowing efficient parallel implementation.

The feature extraction is based on distortion tolerant Gabor transformation and minimum distortion clustering by hierarchical self-organizing feature maps (SOFM). Due to unsupervised learning strategy there is no need for preclassified training samples or other explicit selection for training patterns during the training.

A subspace classifier implementation on top of the feature extractor is demonstrated. The current experiments indicate that the feature space has sufficient resolution power for small number of classes with rather strong distortions. The amount of supervised training required is very small, due to many unsupervised stages refining the data to be suitable for classification.

1 Introduction

Major stages in pattern recognition task are feature extraction and classification [7]. The feature extraction procedure transforms the preprocessed image into a representation that is more suitable for distinguishing different objects. A feature can be a complex description, e.g. a graph describing connected line segments and textured areas, or it can be simply a vector of real numbers, each component measuring some relevant quantity from the image.

Recent emergence of very powerful and general neural network classifiers, like learning vector quantization,LVQ, [10] and multilayer perceptron, MLP, with back-propagation learning [17], has greatly simplified construction of classifiers in pattern recognition systems. As non-parametric methods the neural classifiers require no model for class probability distributions, because they adaptively find class boundaries that correctly classify the training samples. But regardless of the alluring omnipotence of the neural classifiers, the need for efficient feature extraction still exists, since images of real world objects are often subject to various sources of distortions, like translation, scaling, rotation, perspective distortions, variations in lighting conditions, variations in background, partial occlusions by other objects and natural variations inside the classes, like human face with different expressions. All these distortions together make the object's class boundaries extremely complex in the gray scale space. To train a reasonably distortion tolerant classifier to work on pure gray scale data the training samples are required to span the whole space of objects with all possible distortions and their combinations. Clearly collecting and classifying this data is practically impossible. On the other hand, feature extraction based on distortion tolerant transforms and unsupervised learning reduces the amount of supervised training required, if the class boundaries in the feature space are simpler than in the original gray scale space.

Purpose of the proposed feature extractor

The first restriction to the feature extractor is unsupervised learning; it should use self-organization principles to adapt to the input data, requiring minimal human intervention or surveillance during learning.

This restriction disallows using 'holistic features' that pertain to the whole object, since it is impossible to locate and segment objects without any a priori information about their properties [7]. Holistic features, like principal components, could be taken into the feature model in some later stage when objects can be located. At the very first stage of feature extraction the features should relate to as small area in the image as possible, to increase probability that all the information collected for the feature comes from the same object and is thus relevant for classification.

The precise purpose of the system is to map each pixel in the image to a feature value that corresponds to the pattern (or detail) seen around the pixel, so that distance of feature values measures the dissimilarity of the images around the respective pixels. Similar mapping is done on several resolutions to produce a multiresolution representation of the image.

The whole feature extraction - classification system forms a pyramid, where number of neurons and connections decreases and connections become more adaptive in higher layers. 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 hierarchical self-organizing feature map.
3. **Supervised layer** carrying out the classification. The proposed classifier is based on linear PCA-networks and locally collected histograms of the previous level outputs.

In the following each block is described, and some experimental results are reported.

2 Producing the primary features with Gabor transformation

Role of the Gabor transformation in the system is explained earlier in [11] and is reviewed here. The 2D Gabor kernels are complex valued orientation specific linear band-pass filters. In the self-similar family of Gabor filters all the filters are dilations, translations and rotations of each others, with impulse response (centered to origin)

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

where

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

The central frequency of the pass band is f in orientation θ, σ determines the bandwidth of the filter and x and y are pixel coordinates in the image.

The frequency response of the filter is

$$\Psi(f, \theta, \omega_x, \omega_y) = \exp\left(-\frac{\sigma^2}{2f^2}((\omega_x - f_x)^2 + (\omega_y - f_y)^2)\right) \tag{2}$$

where ω_x and ω_y are frequencies in x- and y-directions, respectively. To cover the whole frequency space we used $\sigma = 2\pi$ and

$$f = \frac{2\pi}{2^k}, \qquad k = 1, 2, ...$$

$$\theta = \frac{\pi t}{T}, \qquad T = 8, \quad t = 0, ...T - 1$$

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

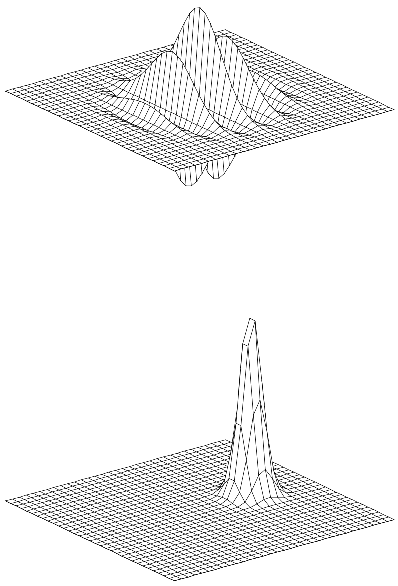

Figure. 1. Example of a Gabor filter's impulse response(upper figure) and frequency response (lower figure).

The Gabor filters are optimally localized band pass filters in terms of the bandwidth and extent of the impulse response [4], making them suitable for extracting frequency contents from as small area as possible.

The Gabor filters capture the whole frequency spectrum, both amplitude and phase. Thus real and imaginary parts of the convoluted image oscillate at the center frequency. We reject the phase information by computing and storing only the magnitude, which changes smoothly over the image. The amplitude of Gabor transformation is found to be robust, distortion tolerant space for pattern recognition in many studies, e.g. [2][3] [5].

To compute the Gabor transform on different resolutions we use only one set of filters at one frequency band and 8 orientations, and downsample the image to match the resolution. Thus the output of the first feature extraction stage is pyramid structure containing 8 real coefficients for each pixel in the image, on several resolutions.

Distribution of the Gabor coefficients on natural images is very compact [4], most of the coefficients are near zero and only few of them carry most of the relevant information. Thus it is reasonable to recode the data to more compact form. The next stage does this recoding by hierarchical self-organizing feature maps, while introducing minimal additional distortion.

3 Clustering the features with hierarchical SOFM

One of the most important features of the self-organizing feature map, SOFM [9], is the topology preserving mapping from high dimensional space to low dimensional map surface. SOFM is today used mostly as training strategy for vector quantizer, and in the final use the form of the map (order of the reference vectors) is not important. SOFM training gives, however, invaluable advantages in producing vector quantizer: the large neighbourhood in the beginning of training helps map to escape local minima of the minimized function, and updating several vectors for each training sample speeds up the convergence of minimization [16].

Direct advantage from the ordering of the map can be achieved in hierarchical systems of SOFM's, e.g. in a hierarchical vector quantization [12]. In the following a reasoning is given why the hierarchical SOFM is suitable for clustering also: it will produce "natural" clusters whos form is dynamically matched to the probability density structure of the input samples. If there are low dimensional structures in the data they will form long elongated clusters, while uniform data is mapped to circular clusters.

Let the SOFM lattice be D-dimensional set of reference vectors $\mathbf{m_r}$ indexed by a D-vector $\mathbf{r}$ and let the neighbourhood function (form of the "activity bubble" on the map [9]) be shift invariant over the lattice, $h : \mathbb{R}^{\mathrm{D}} \mapsto \mathbb{R}$, $h(\mathbf{r}, \mathbf{q}) = h(\mathbf{r} - \mathbf{q})$.

SOFM training [9] is a gradient descent method to minimize a cost function [12], [16].

$$D(\mathbf{m}) = \sum_{\mathbf{q}} \sum_{\mathbf{r}} h(\mathbf{r} - \mathbf{q}) \sum_{\mathbf{x} \in A_{\mathbf{q}}} p(\mathbf{x}) \|\mathbf{x} - \mathbf{m_r}\|^2 \tag{3}$$

where $A_{\mathbf{r}}$ is the set of points $\mathbf{x}$ that are mapped to unit $\mathbf{r}$. $A_{\mathbf{r}}$ is a convex volume limited by normal hyperplanes of nearest units in the space. In the following we denote set $\{\mathbf{x} | \mathbf{x} \in A_{\mathbf{r}}\}$ by $\mathbf{x_r}$. Let $p(\mathbf{x}) = 1/N$ where N is the number of training points.

Eq.3 means, that the map minimizes the mean distance from input vector to reference vectors that are in the $h()$-neighbourhood of $\mathbf{m_{r(x)}}$.

In a local minimum of the SOFM training the gradient for unit $\mathbf{q}$ is zero.

$$G_{\mathbf{q}} = \sum_{\mathbf{r}} h(\mathbf{q} - \mathbf{r}) \sum_{\mathbf{x_r}} (\mathbf{x_r} - \mathbf{m_q}) = \sum_{\mathbf{r}} h(\mathbf{q} - \mathbf{r})(\mathbf{C_r} - \mathbf{m_q}) = 0 \tag{4}$$

where $\mathbf{C_r}$ is the mean of the points mapped to $\mathbf{r}$, $\mathbf{C_r} = \frac{1}{N_{\mathbf{x_r}}} \sum_{\mathbf{x_r}} \mathbf{x_r}$. Let $\sum_{\mathbf{q}} h(\mathbf{q}) = 1$. Then

$$\mathbf{m_q} = \sum_{\mathbf{r}} h(\mathbf{r} - \mathbf{q}) \mathbf{C_r}$$

and

$$\mathbf{m} = \mathbf{h} * \mathbf{C} \tag{5}$$

where $\mathbf{h}$ is a vector containing the neighbourhood feedback values in the discrete lattice points and $*$ is the D-dimensional discrete convolution. To guarantee convergence $\mathbf{h}$ must be low-pass filter giving smooth $\mathbf{m}$. In the final stage of the SOFM training $h()$ approaches delta function and $\mathbf{m} \approx \mathbf{C}$ (and in the limit $\mathbf{m} \rightarrow \mathbf{C}$ and optimal vector quantization results). Then

$$\sum_{\mathbf{r}} h(\mathbf{q} - \mathbf{r})(\mathbf{C_r} - \mathbf{m_q}) \approx \sum_{\mathbf{r}} h(\mathbf{q} - \mathbf{r})(\mathbf{C_r} - \mathbf{C_q}) \tag{6}$$

So, for small neighbourhood the SOFM minimizes the weighted distance of points mapped to the neighbourhood. When the neighbourhood is smoothly decreased, map finds as compact set of points as possible to be mapped to each neighbourhood during the training. If the data contains compact structures (with small total volume of spheres required to contain the data points) then at some neighbourhood size the map can minimize the distance of points mapped to the neighbourhood by placing the units sequentially along the structure. This is how the map finds and preserves the local ordering relations of the training samples. On the other hand, if the data are uniformly distributed, then the units will end up close to each others on every neighbourhood size, turning the map to a space filling surface having the same structure on every scale [9].

3.1 Hierarchical SOFM

In a hierarchical SOFM a second layer map $\mathbf{m}^2$ quantizes the outputs $\mathbf{r}$ of the first layer map $\mathbf{m}^1$. Units of $\mathbf{m}^2$ divide the units in $\mathbf{m}^1$ into equiprobable convex areas corresponding approximately to some neighbourhood size used during training in $\mathbf{m}^1$.

Now the region mapped to one $\mathbf{m}^2$ unit is the set of all the regions of the corresponding units in $\mathbf{m}^1$, forming arbitraryly complex convex or concave volume limited by hyperplanes.

This clustering corresponds roughly to classical clustering methods (e.g. ISODATA) when the distance of a point from the cluster is defined as weighted distance from all the points in the cluster, but with HSOFM the clustering requires only two nearest neighbour searches instead of computing all the distances and the extent of weighting is adapted automatically to the data distribution ($h()$ is defined in the map coordinates).

In addition, the system minimizes the total quantization error introduced by both the layers [12]. Among clustering methods this is a rather unique balance between preserving maximal information and collapsing the data to natural clusters.

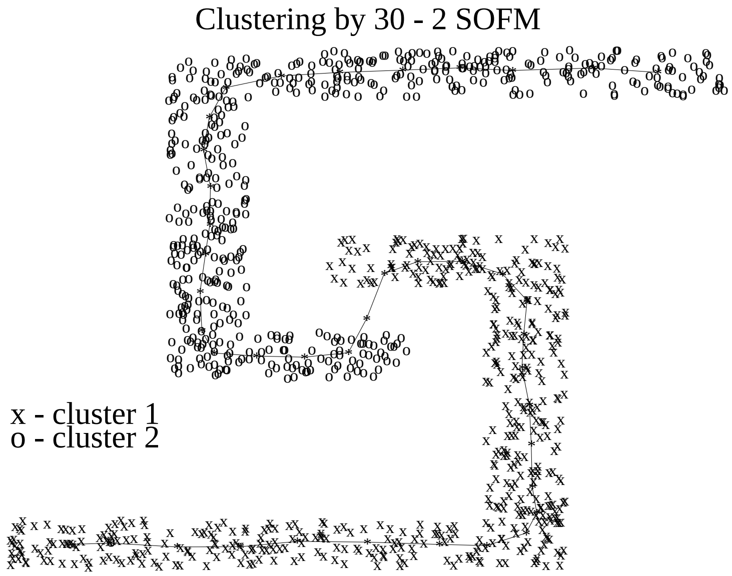

Figure 2. shows a very simple example about the capabilities of hierarchical SOFM in clustering. The input probability distribution contains continuous structures that are partially interleaved. Clustering by the HSOFM is based on continuous compact areas, and it thus easily finds the natural cluster boundary. The continuous line shows the first layer 30-unit map oriented to the data distribution. The second layer had only 2 units splitting the $\mathbf{m}^1$ to two halves as shown in Fig.1.

In the proposed system the output of the feature extraction stage is a "feature image" $c(\mathbf{p})$ where each image location **p** is mapped by the HSOFM to unit c corresponding to a cluster in the space of Gabor amplitudes in different orientations and resolutions.

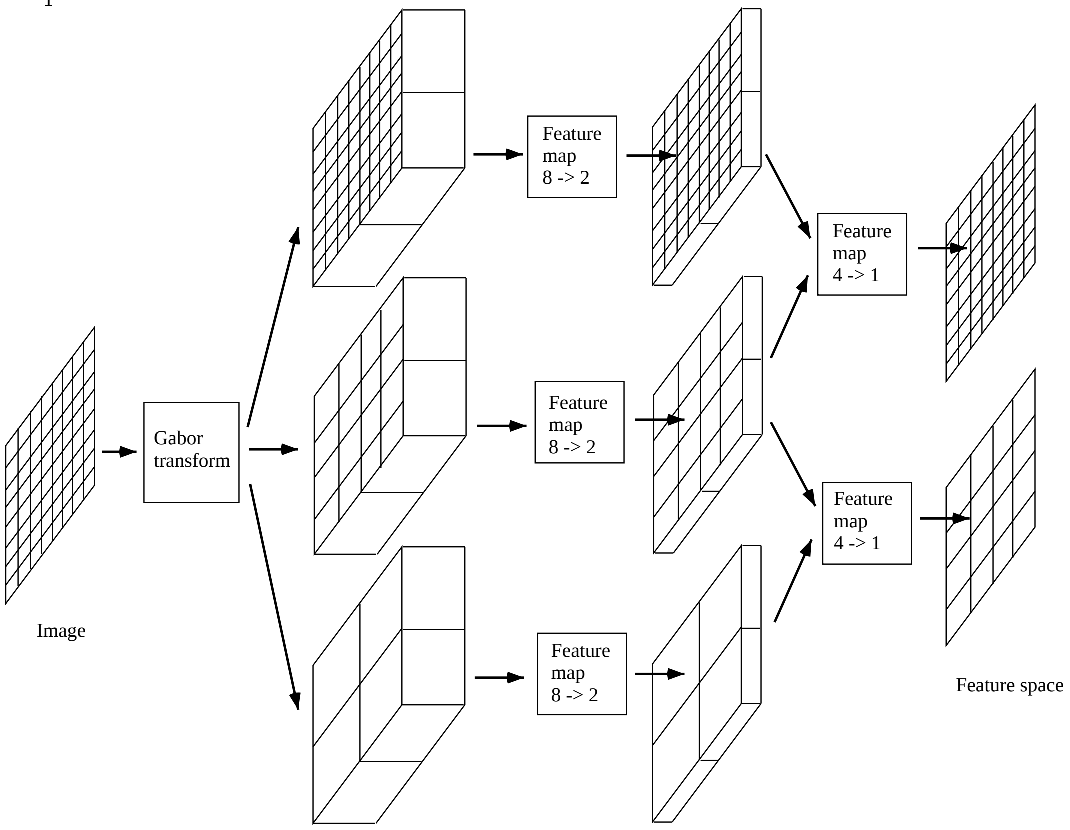

Figure 3. Schematic flowchart of the feature extractor

Fig.3. shows the principle of the HSOFM in the feature extractor. The squares represents pixels and the depth of the planes show dimensionality of the pixels, which is in the first plane 8 and in the second 2 in our configuration. The sizes of the maps were 10×10 and 100×1, respectively.

The whole mapping network could be implemented in purely neural way by having separate map with the same weights for each pixel to be mapped. Computationally, on sequential computers, it is of course simpler to scan the image and map each pixel in turn. The parallel maps would produce similar hypercolumns as found in the intermediate levels in the Neocognitron [6], suggesting the possibility to use this type of mapping as a preprocessor for Neocognitron networks.

In the resulting feature space any classifier could be used. In the next section we propose using subspace classifier [13], with an additional feature extracting step, that will increase tolerance to distortions, where distance relations of details of an object change.

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. One has to repeat the classification procedure on each resolution to capture objects of all sizes. The resolution hierarchy can be used in classification also, since the lower resolution features represent a *context* for the higher resolution details. It can also be used to locate the objects rapidly before classification (doing 'blob detection', or low-pass filtered template matching).

4 Distortion tolerant classification with histograms and PCA-networks

The feature extraction based on Gabor-filters is not shift invariant, but instead relates the features to as small area as possible increasing probability that the features come from only one object. 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 is not very prominent since the hierarchical clustering absorbs most small variations and expands the spatial support of clear details, see [11]. Consequently, good recognition results have been achieved by direct template matching in the feature space [11]. 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 these types of distortions

1. *Requirement that the important features must appear in correct order* based usually on editing distances [15].
2. *Planar graph matching algorithms.* See e.g. [3].
3. *Requirement that the important features must appear inside certain area* by a translation invariant transformation applied to the area. The transformation may be e.g. a histogram of the underlying features. Distortion tolerance of the Neocognitron [6] is based on this category, since a C-cell turns on if any of the pattern detecting S-cells in its receptive field is on, and whole C-layer thus produces histogram of the S-layer features, clipped to 1. According to recent results, by weighting the histogram with a smooth window the invariance property is enhanced [1].

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

4.1 Local feature histogram

The input vector for the classifier is a histogram of HSOFM codes weighted by a Gaussian window:

$$\mathbf{w}_i(\mathbf{p}) = \sum_{c(\mathbf{p}')=i} \Omega(\mathbf{p} - \mathbf{p}') \tag{7}$$

where $\mathbf{p}$ is a position vector in the feature image $c(\mathbf{p})$ and $\Omega()$ is the weight function $\Omega(\mathbf{p}) = \exp\left(-\frac{\mathbf{p}^T\mathbf{p}}{2R^2}\right)$ and R determines the size of the window.

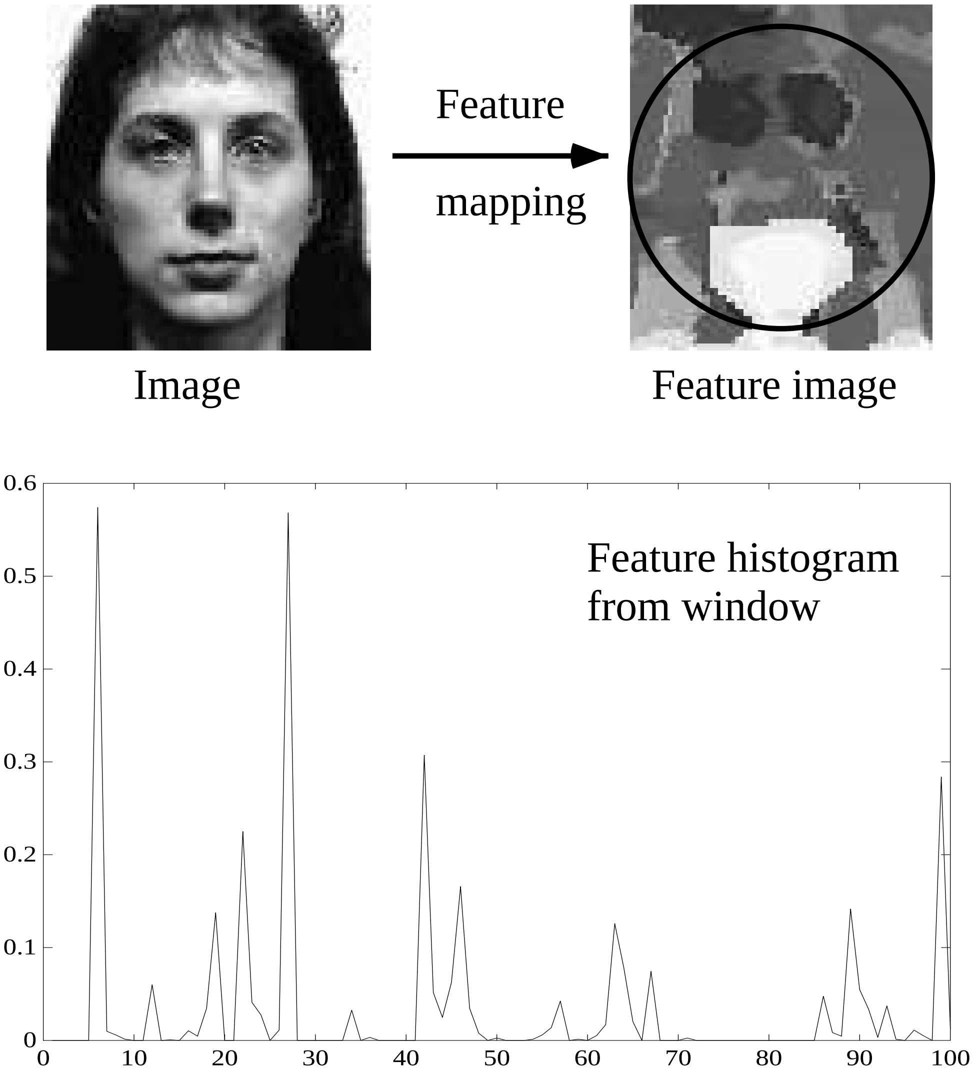

Figure 4. Example of feature image and weighted histogram

In purely neural implementation of this feature collection there is one unit for each possible feature value, connected with Gaussian receptive field to every corresponding feature detecting unit of every map.

For small number of objects there will be specific units in the HSOFM map for 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 input vector is collected from several locations with smaller windows and the vectors are stacked together.

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

A similar approach for distortion tolerance has been developed for handling 1D time dependent data by SOFM [8]: for each time step the input is quantized by a SOFM, and the history of the data is stored as exponentially weighted histogram over time of the outputs of the SOFM. Then these histories are learned by another SOFM.

4.2 Classification with PCA-networks

Subspace methods [13] are efficient and computationally simple classification methods for histogram and spectrum type data. Each class is defined as a linear subspace and represented by a set of orthogonal base vectors. Output of the net is the projection of the input vector onto the class subspace, and an unknown input vector is classified according to the largest output.

The neural subspace classifier is based on the principal component network [14]. In [14] it is shown that with the learning rule

$$\frac{dM}{dt} = \alpha(\mathbf{I} - MM^T)\mathbf{x}\mathbf{x}^T M \tag{8}$$

the columns of matrix M become orthonormal and span the same subspace as the k largest eigenvalues of the autocorrelation matrix of the input data (E$\{\mathbf{x}\mathbf{x}^T\}$) i.e. the space of the k largest principal components of the inputs.

Now each class i is represented by a subspace M_i. The most simple approach for finding the class subspaces is to compute the largest principal components for each class separately using Eq.8.

Better results are obtained by the Learning Subspace Method LSM [14] that finds subspaces minimizing classifications errors. LSM requires only a small change in update rule of Eq.8 (compare to LVQ vs. SOFM). If classification of an input is a and the correct class is b then update M_a with negative learning parameter and M_b with positive one.

In the current problems 1 and 2 dimensional class subspaces where found to be sufficient. In most of the experiments even the principal component subspaces were able to separate the classes but for better resolution the LSM was used.

5 Experimental results

In the unsupervised clustering block the 8 Gabor filter responses were quantized by a 10×10 map separately on each resolution and then two successive resolutions were combined by a 100-unit 1D map. Now the histogram of these features is a 100 dimensional vector that was passed over to the subspace classifier. The width of the Gaussian window Ω was about 50 pixels ($R = 20$) for face recognition.

Fig. 5. shows some of the test images. For each person two sample images were selected and a PCA-network was trained with the samples. Table 1. shows the network outputs for the classes T,K and J, respectively The training samples are denoted by bold face letters in the table.

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 HSOFM, long clusters reaching the invariance over the 1 octave interval, and some features 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 in Fig.2).

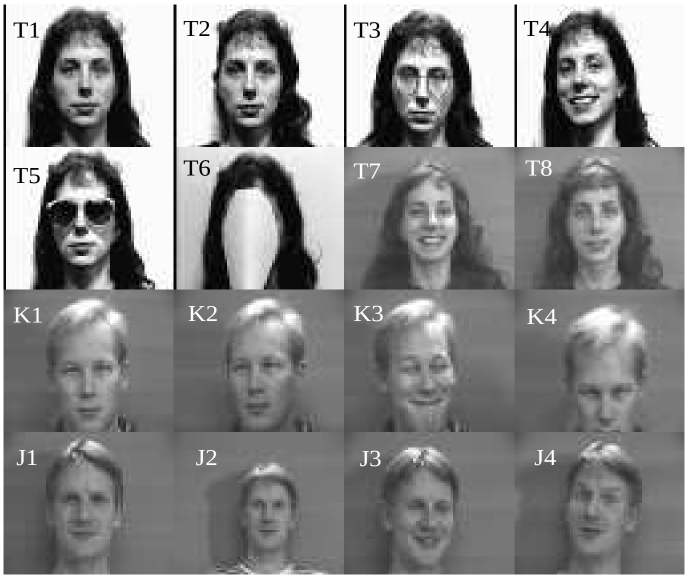

Figure. 5. Some test images (each one 128 $\times$ 128 pixels)

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

Table 1. Projections of test images onto class subspaces

The system is not shift invariant since moving the window Ω changes the histogram, but to the extent of Ω it is shift tolerant, small shifts causing small changes in projections. In Fig.6. three outputs of the PCA networks are plotted along a horizontal line in an image containing figures T8, K1 and J1 in a row, respectively. The shift tolerance is very good and classification is correct even near the midplanes of two objects.

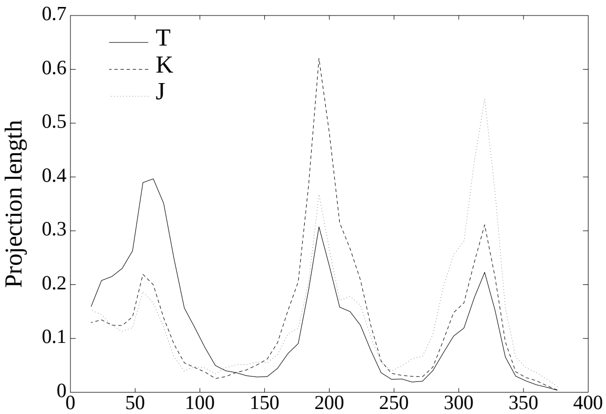

Figure 6. Class projections as functions of histogram window location.

Currently the face recognition experiments have contained only less than 10 persons, with results comparable to those in table 1. In another experiment different computer screens and terminals were used as objects, giving similar results. The experiments with larger number of classes are in progress, and they will show whether the size of the maps should be related to number of classes or will there be a limit where the system has developed a sufficient set of feature detectors for general usage.

6 Conclusions

Using hierarchical SOFM to map continuums in data space to clusters while causing minimal additional quantization error provides a feature space that is distortion tolerant in small scale and offers specific enough features to facilitate classification based on local feature histograms.

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

The current system has no feedback in any stage. Adding feedback to control focusing of attention when proceeding from coarse levels to higher resolutions will be subject to future study.

7 Acknowledgements

The author wishes to thank prof. Erkki Oja for helpfull discussions throughout the work and especially concerning the subspace classifier.

References

[1] E. Barnard, D. Casasent, Shift Invariance and the Neocognitron, Neural Networks, Vol. 3, 1990 , pp. 403-410.

[2] J. Buhman, J. Lange and C. von der Malsburg, Distortion Invariant Object Recognition by Matching Hierarchically Labeled Graphs, Proc. IEEE IJCNN-89 Vol. 1. 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, Proc. IEEE IJCNN-90 Vol. 2. San Diego, CA, Jan. 1989, pp. 411-416.

[4] J.G. Daugmann, Relaxation Neural Network for Complete Discrete 2-D Gabor Transforms, Proc. 6-SCIA, Oulu, Finland, June 19-22, 1989, pp. 33-46.

[5] K.A. Flator and S.T. Toborg, "An Approach To Image Recognition Using Sparse Filter Graphs", Proc. IEEE IJCNN-89 Vol. 1. Washington DC, June 18-22, 1989, pp. 313-320.

[6] K. Fukushima, Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by a shift in position, Biological Cybernetics, 36, 1980, pp.193-202.

[7] A.K. Jain, Fundamentals of digital image processing, Prentice-Hall, Inc., New Jersey, 1989.

[8] J. Kangas, Time dependent self-organizing maps for speech recognition, *To appear in* Proc. ICANN-91, Espoo, June 24-28, 1991.

[9] T. Kohonen, Self-Organization and Associative Memory, Springer-Verlag, Berlin, Heidelberg, 1989.

[10] T. Kohonen, An introduction to neural computing, Neural Networks, Vol. 1, 1988, pp. 3-16.

[11] J. Lampinen and E. Oja, Distortion Tolerant Feature Extraction with Gabor Functions and Topological Coding, Proc. INNC-90, Vol. 1., Paris, July 9-13, 1990, pp. 301-304.

[12] S.P. Luttrell, Self-Organisation: a derivation from first principles of a class of learning algorithms, Proc. IEEE IJCNN-89 Washington DC, Vol.2. June 18-22, 1989, pp.495-498.

[13] E. Oja, Subspace Methods of Pattern Recognition, Research Studies Press Ltd., 1983.

[14] E. Oja, Neural Networks, Principal Components and Subspaces, Int. Jou. of Neural Systems, Vol. 1., No. 1., 1989, pp 61-68.

[15] E. Reuhkala, Recognition of strings of discrete symbols with special application to isolated word recognition, Acta Polyt. Scand. Ma 38. Dr. Tech . dissertation, Helsinki Univ. of Tech. 1983.

[16] H. Ritter and K.Schulten, Kohonen's Self-Organizing Maps: Exploring their Computational Capabilities, Proc. IEEE IJCNN-88 SanDiego, Vol.1. July 24-27, 1988, pp.109-116.

[17] D.E. Rumelhart, G.E. Hinton and R.J. Williams, Learning internal representations by error propagation, in Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1., MIT Press, Cambridge, 1986, pp. 318-362.