A system for various visual classification tasks based on neural networks

Gunther Heidemann Dirk Lücke Helge Ritter
AG Neuroinformatik, University of Bielefeld
Postfach 10 01 31, D-33501 Bielefeld, Germany
gheidema@techfak.uni-bielefeld.de

Abstract

A three stage recognition architecture that can be trained to different recognition or segmentation tasks is presented. It consists of an adaptive feature extraction based on vector quantization and local PCA. The features are classified by neural expert networks. It will be shown that the system can be applied to object classification, segmentation of partially occluded objects and classification of object parts without modifications in the architecture.

1. Introduction

Artificial neural networks (ANN) are well-suited for classification tasks in computer vision. Their adaptivity has two major advantages: (a), ANN acquire the necessary knowledge from examples and therefore can be adapted to different object domains, (b), because of the implicit representation ANN can store even knowledge that can hardly be modelled explicitly by a human designer. Since for the most tasks ANN cannot be applied directly to the raw pixel data, suitable features have to be extracted before the classification itself. However, the benefit of adaptivity gained by the application of ANN may vanish if the required feature extraction has to be designed from scratch for a new vision problem. Hence, feature extraction should have the same adaptivity as the neural classifier.

The recognition system presented here has three processing stages which can all be adapted to the recognition task by examples [2]. In the first level, the input data are pre-structured by a vector quantization. Then, features are extracted by a local principal component analysis which are classified by expert ANN in the last processing level. We will first describe the system, then the ability of the system to be adapted to different recognition tasks will be demonstrated.

2. The adaptive recognition architecture

The proposed system is designed for the classification of whole images or smaller image patches as well. Both types of input will be called “window” in the following. A window may show an object totally visible (in this case some kind of pre-segmentation is assumed) or smaller parts of an object which are either sampled randomly or gained by an attentional mechanism apart from the system outlined here. We refer to section 3 for examples.

The basic idea is to provide an adaptive feature extraction for the neural classification level. It should be possible to train the feature extraction by the same examples as the classifier. An established technique in computer vision is projecting the windows to their principal components. However, principal component analysis (PCA) leads to specific filters only if the training set itself is highly specific. An example are so called “eigenfaces” for face recognition [11]. In contrast, if the training windows are taken randomly from natural images, the principal components tend to be the same for all images and even throughout different scales [8, 1] which is due to the linearity of the method. Local PCA, however, can be viewed as a nonlinear extension of simple, global PCA [10] and leads to a much greater variety of highly specific filters. In the system proposed here, the input data are pre-structured by vector quantization (VQ). For each reference vector of the VQ, a locally valid PCA is performed (Fig. 1). So far, the system can be trained unsupervised. To classify the features extracted by the local PCA, to each reference vector an expert net of the local linear map (LLM) – type is attached, which is trained supervised. We will outline the system in more detail.

2.1. Vector quantization

The input data are roughly approximated by N_{vq} reference vectors $\vec{W}_i^{vq}$, $i \in [1, N_{vq}]$, which are positioned by the vector quantization learning rule

$$\Delta \vec{W}_{n_R}^{vq} = \epsilon(\vec{x} - \vec{W}_{n_R}^{vq}), \quad \epsilon \in]0,1[, \tag{1}$$

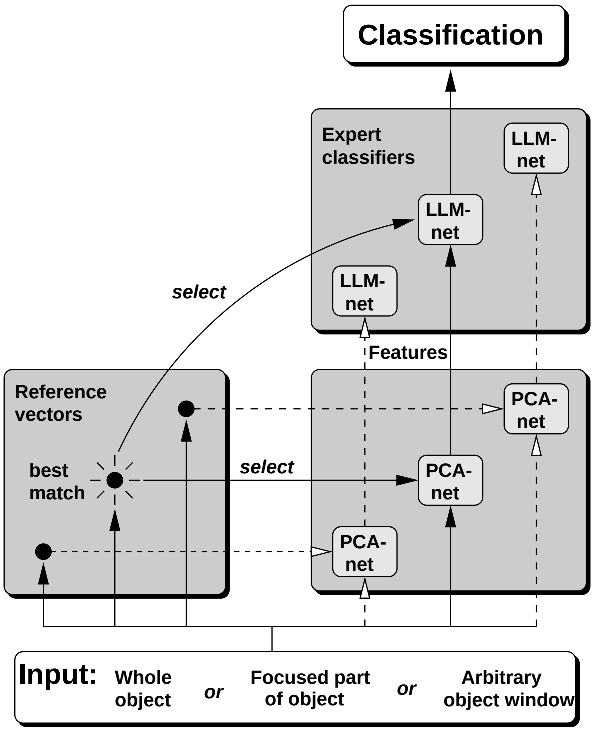

Figure 1. The neural classification system.

where n_R is the index of the best match reference vector. Since application of eq. (1) alone tends to be caught in local minima of the mean square error function for the approximation error, we enhanced the algorithm by a so called "Activity Equalization". In short, this method re-initializes the nodes if they seldom or never become the "winner", so the average node activities will be equalized during training. For details see [3, 2].

An input vector $\vec{x} \in \mathbb{R}^{w^2}$ represents an image window of size $w \times w$, or, alternativly, $\vec{x} \in \mathbb{R}^{3 \cdot w^2}$ if three colour channels are evaluated. $\vec{x}$ is mapped by the VQ to an integer number $n_R = 1 \ldots N_{vq}$ (n_R denotes the best match).

2.2. Local PCA

Sanger [8] proposed a single layer feed forward network for the successive calculation of the principal components (PCs) of training vectors. The nodes have a linear activation function

$$V_i = \sum_{j=1}^{d} W_{ij}^{pca} x_j, \quad i = 1 \ldots N_{pca},$$

where $\vec{W}_i^{pca}$ are the input weight vectors of the nodes and $\vec{x}$ the input (see last section). After training by Sanger's rule

$$\Delta W_{ij}^{pca} = \epsilon V_i \left[\left(x_j - \sum_{k=1}^{i-1} V_k W_{kj}^{pca} \right) - V_i W_{ij}^{pca} \right]$$

the weight vectors approximate the PCs in the order of their eigenvalues, beginning with the largest. The output $\vec{V} \in \mathbb{R}^{N_{pca}}$ of the network is the projection of the input $\vec{x}$ to the first N_{pca} PCs with the largest eigenvalues.

2.3. LLM-nets

The Local Linear Map is related to the self-organizing map [4] and the GRBF approach (e.g. [6]). It can be trained to approximate a nonlinear function by a set of locally valid linear mappings, for details see e.g. [2]. For the classification task, a mapping from the N_{pca}-dimensional input vector $\vec{V}$ to an N_{cl}-dimensional output vector $\vec{y}$ is required (N_{cl} = number of classes). The target training vector for class c has the form $y_j^{(\alpha)} = \delta_{jc}, j = 1 \ldots N_{cl}$, where α denotes the number of the training example.

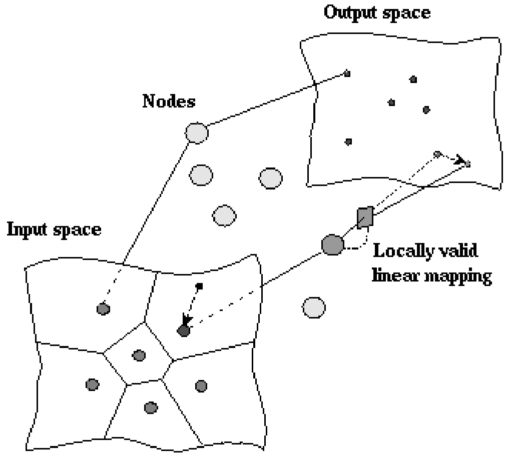

Figure 2. The LLM-net approximates a nonlinear mapping by several locally valid linear mappings.

An LLM-node i has an *input weight vector* $\vec{W}_i^{llm,in} \in \mathbb{R}^{N_{pca}}$ and a linear function to compute the output, which consists of the *output weight vector* $\vec{W}_i^{llm,out} \in \mathbb{R}^{N_{cl}}$ and a matrix $\mathbf{A}_i \in \mathbb{R}^{N_{cl} \times N_{pca}}$. The network output is calculated from the input vector $\vec{V} \in \mathbb{R}^{N_{pca}}$ by a search for the best match node k, which is determined by

$$k = \arg \min_{i=1 \ldots N_{llm}} (\|\vec{V} - \vec{W}_i^{llm,in}\|),$$

where N_{llm} is the number of LLM-nodes. The output vector $\vec{y} \in \mathbb{R}^{N_{cl}}$ is then

$$\vec{y} = \vec{W}_k^{llm,out} + \mathbf{A}_k(\vec{V} - \vec{W}_k^{llm,in}). \qquad (2)$$

Given correct input-output pairs of the form $(\vec{V}^{(\alpha)}, \vec{y}^{(\alpha)})$, the best match node k of the network is adapted supervised with the adaptation step sizes $\epsilon_{in}, \epsilon_{out}, \epsilon_A \in]0,1[$:

$$\begin{aligned}
\Delta\vec{W}_k^{llm,in} &= \epsilon_{in} \;\; (\vec{V}^{(\alpha)} - \vec{W}_k^{llm,in}), \\
\Delta\vec{W}_k^{llm,out} &= \epsilon_{out} \; (\vec{y}^{(\alpha)} - \vec{y}(\vec{V}^{(\alpha)})) + \; \mathbf{A}_k \Delta\vec{W}_k^{llm,in}, \\
\Delta\mathbf{A}_k &= \epsilon_A \;\; (\vec{y}^{(\alpha)} - \vec{y}(\vec{V}^{(\alpha)})) \cdot \frac{(\vec{V}^{(\alpha)} - \vec{W}_k^{llm,in})^T}{\|\vec{V}^{(\alpha)} - \vec{W}_k^{llm,in}\|^2}.
\end{aligned}$$

2.4. Training procedure

Given a training set T, the neural architecture shown in Fig. 1 is adapted in three stages:

1. VQ of the input space, then divide T into subsets $T_1 \ldots T_{N_{vq}}$ which contain the best match examples for the obtained reference vectors $\vec{W}^{vq}$.

2. Train one PCA-net for each subset T_i. Compute the sets T_i' of the (input-) projections of all examples in T_i to the first N_{pca} PCs of the corresponding PCA-nets.

3. Train one LLM-net for each training set T_i'.

The combination of the VQ with subsequent PCA-nets leads to a local PCA within the Voronoi tesselation cells in input space.

3. Results

The system was tested on three different types of problems. The recognition of complete, centered objects was tested using the Columbia Object Image Library (COIL). From the same library, images with partial object occlusion were generated artificially. Here, object segmentation was carried out by use of much smaller windows. Last, the system was tested in combination with a data driven attentional mechanism for the recognition of facial features.

3.1. Recognition of complete objects

To test the system at first independently of the “where problem”, we used images of 40 different objects of the COIL-100. This image collection shows single objects in a normalised position, so the images can directly be fed to the networks. COIL-100 is available at `http://www.cs.columbia.edu/CAVE` and described in [5]. As shown in Fig. 3, the subset of 40 objects was chosen in the way to maximize classification difficulty by selecting groups of high similarity (e.g. all toy cars). Each object is rotated on a turntable at pose intervals of 5 degrees, so there are 72 images of each object. The resolution was subsampled to 64 × 64. Moreover, only the grey value information was used, so the input to the classification system was $\vec{x} \in \mathbb{R}^{64^2}$. For training, 18 images of each object were used (at 20 degrees pose intervals), the remaining 54 for testing. A recognition rate (percentage of correct classifications) of 96.5% could be reached for $N_{vq} = 6$, $N_{pca} = 12$ and $N_{llm} = 40$. The dependence of the recognition rate on the choice of parameters proved to be “well behaved” in earlier studies [2], i.e., performance increases for larger N_{vq}, N_{pca} and N_{llm} until saturation (about 97.1%) is reached. Using more training examples (36 images of each object at 10 degrees pose intervals) leads to a better recognition rate of 98.2%, however, for applications the first result is more relevant since it demonstrates the good generalization properties.

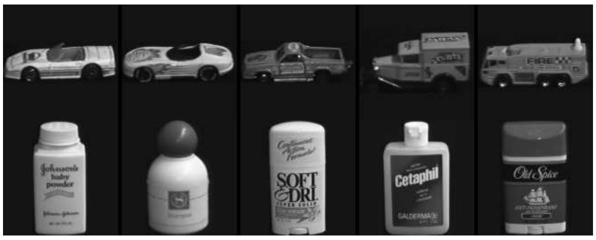

Figure 3. Ten of the 40 objects of COIL used for testing the recognition of complete objects in a normalized position.

3.2. Object segmentation by small windows

As in real applications objects are often partially occluded, classification of smaller object parts is a key ability. This can be done either for special, pre-defined object parts only (section 3.3), or continuously for the whole object area. Here, the system is applied to segment partially occluded objects by continuously scanning the scene with a window that is small compared to the objects but large enough to evaluate colour texture features [9]. 20 objects of COIL-100 were used in their RGB-version for the test. The system was trained with 18 images of each of the un-occluded objects, from which training windows of size 17 × 17 were sampled from the object area. For comparison, the images themselves have resolution 128 × 128. Since the RGB-values are used, the input is $\vec{x} \in \mathbb{R}^{3 \cdot 17^2}$. The system was tested on images which were artificially generated from the remaining 54 images of each object in the way that in each new image two different objects are partly overlapping, see

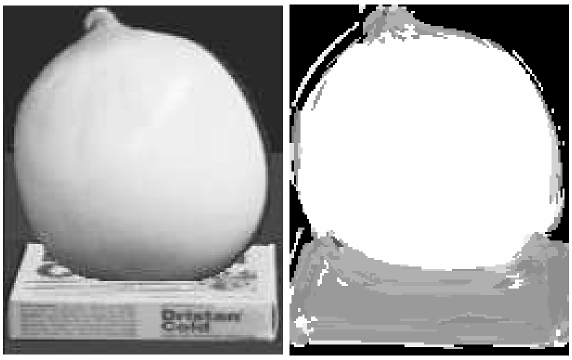

Figure 4. Object segmentation of partially occluding objects (on artificially generated images of COIL) by classification of windows with size 17×17.

Fig. 4, left. The output of the system is the object class or an additional class for the background (in total 21 classes). Fig. 4, right, shows an example of the segmentation result when the whole image was scanned by the classifier. The correctly classified pixels rate about 82%.

3.3. Classification of focused windows

The last test was carried out on the Yale Face Library (http://giskard.eng.yale.edu/yalefaces/yalefaces.html), which contains 150 grey value images of 15 subjects, each with differing face expression, changing illumination and with glasses in one image. The task was to identify eyes, nose and mouth in each image. In contrast to section 3.2, the images were not scanned continuously. Instead, an attentional mechanism similar to the symmetry based saliency map proposed in [7] was used to generate so called *focus points*, for details see [2]. The focus points are shown by markers in Fig. 5. Only 17 × 17-windows centered at the focus points were fed to the classification system (the images were subsampled to 160 × 121). The task is to classify eyes, nose and mouth against the other focus points. For training and testing in each case half of the database was used. 92% correct classifications could be achieved for $N_{vq} = 3$, $N_{pca} = 15$ and $N_{llm} = 30$.

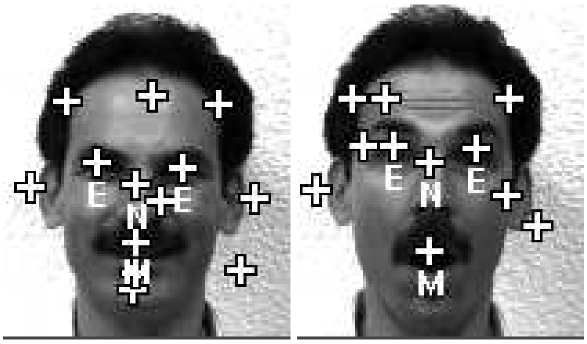

Figure 5. Classification results of windows around the focus points (markers). E,N,M indicate eye, nose and mouth, focus points without labels belong to the rejection class.

4. Conclusion

The proposed neural system could be applied to three different types of visual classification tasks due to its adaptive feature extraction and -classification. A limitation of the current realization is that for the recognition of complete objects or part of objects the “where problem” must be solved by other mechanisms. In future work, we will try to use the segmentation application (section 3.2) as attentional guidance for the recognition system for complete objects.

This work was supported by the Deutsche Forschungsgemeinschaft in the collaborative research project SFB 360.

References

[1] P. Hancock, R. Baddeley, and L. Smith. The principal components of natural images. *Network*, 3:61–70, 1992.

[2] G. Heidemann. *Ein flexibel einsetzbares Objekterkennungssystem auf der Basis neuronaler Netze*. PhD thesis, Univ. Bielefeld, Techn. Fak., 1998. Infix, DISKI 190.

[3] G. Heidemann and H. Ritter. Efficient Vector Quantization using the WTA-rule with Activity Equalization. *Neural Processing Letters*, 13(1):17–30, 2001.

[4] T. Kohonen. Self-organization and associative memory. In *Springer Series in Information Sciences 8*. Springer-Verlag Heidelberg, 1984.

[5] S. Nene, S. Nayar, and H. Murase. Columbia object image library: Coil-100. Technical Report CUCS-006-96, Dept. Computer Science, Columbia Univ., 1996.

[6] T. Poggio and S. Edelman. A network that learns to recognize three dimensional objects. *Nature*, pages 263–266, 1990.

[7] D. Reisfeld, H. Wolfson, and Y. Yeshurun. Context-free attentional operators: The generalized symmetry transform. *Int'l J. Computer Vision*, 14:119–130, 1995.

[8] T. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.

[9] D. Strotmann. Lokale Klassifikation als Basis für die Erkennung teilverdeckter Objekte. Master's thesis, Univ. Bielefeld, Technische Fakultät, 1999.

[10] M. E. Tipping and C. M. Bishop. Mixtures of probabilistic principal component analyzers. *Neural Computation*, 11(2):443–482, 1999.

[11] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3:71–86, 1991.