

Face Detection and Identification Using a Hierarchical Feed-forward Recognition Architecture

Ingo Bax, Gunther Heidemann and Helge Ritter Neuroinformatics Group

University of Bielefeld

P.O. Box 10 01 31, D-33501 Bielefeld, Germany

E-mail: {ibax,gheidema,helge}@techfak.uni-bielefeld.de

Abstract— We apply a hierarchical feed-forward neural architecture to the problem of face recognition. The network is similar to the Neocognitron-approach and a two-layer variation of this architecture, which has previously been successfully applied to patch classification tasks. We extend this architecture to a three-layer one, which allows not only identification of image patches, but also detection in larger images. In the research area of face recognition a lot of expertise has been developed for the problem of either identification or detection, but approaches which deal with both problems simultaneously are rarely to be found. In this work, we apply the hierarchical approach to this problem and evaluate the performance on artificial datasets.

I. INTRODUCTION

Despite promising advances within the last years, visual recognition in unrestricted environments is still a major problem in computer vision research, because the input stimulus is subject to multiple sources of distortion like deformation, scaling, arbitrary viewpoints, sensor noise, changing illumination, etc. Fukushima's Neocognitron [1] is an early computational architecture for pattern recognition which is invariant to distortions of the input stimulus like rotation and scaling. This invariance is achieved by feed-forward processing in a multi-layer hierarchy, a principle inspired by physiological and psychophysical findings about the behavior of *simple cells* and *complex cells* in the mammalian visual cortex, discovered by Hubel and Wiesel [2].

Whereas the model was mostly applied to the recognition of artificial stimuli like paper-clip objects, more recently, Wersing and Körner [3] introduced a two-layer variation of the model, that can also be used for the recognition of more natural stimuli like objects and faces. This is achieved by incorporating an extension of Sparse Coding [4] as an un-supervised efficient coding scheme for obtaining receptive field profiles, that are tuned to the image domain. The approach was shown to have high classification performance on the COIL-100 object dataset [5] and also on ORL face dataset [6].

In this contribution, we describe an extension of the architecture proposed in [3], which uses an additional network layer, whose receptive field profiles are obtained by supervised learning on the outputs of a two-layer network, that is exposed to a labeled dataset containing different training views of faces. These profiles are called "View Tuned Units" (VTUs) [3], [7] and can be understood as "grand-mother cells", which are sensitive to different views of the same class of stimuli. We

apply the VTUs in the three-layer model to detect and identify faces in a larger image.

In contrast to other approaches, see e.g. [8], [9], [10], where the main task is detection, the approach presented here is capable of simultaneously detecting and identifying faces in images.

In the next section, we will provide a definition of the three-layer hierarchical model, and then describe the purpose of each processing layer. In Section III we present experimental results for (i) a classification task, where we use distorted images from the ORL face dataset to evaluate the robustness of the approach with respect to scaling, clutter and shift in position and (ii) a detection task, where artificial test images are generated from faces from the ORL face dataset, which are randomly placed in images of natural scenes taken from the ArtExplosion image library [11].

II. THE HIERARCHICAL MODEL

The model consists of alternating layers of simple and complex cell planes, each of which performs a hierarchical feature extraction using different types of receptive field profiles. First, we describe the topology of the model and how the activation of the simple cells and the complex cells is computed and then discuss, what kind of profiles are used in each of the three layers.

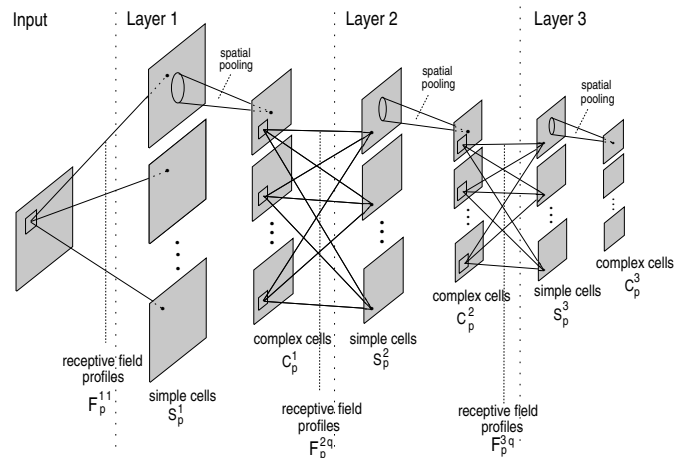


Fig. 1. The Hierarchical Model. The network consists of three alternating layers of simple and complex cell planes.

A. Topology

Figure 1 shows a diagram of the hierarchical model. It consists of $L = 3$ layers, indexed $l = 1 \dots L$ and each holding P_l planes of two types: *simple cell planes* S_p^l and *complex cell planes* C_p^l with $p = 1 \dots P_l$. The network input is given as a gray value pixel image. For notational convenience, we set $P_0 = 1$ and refer to the input image as C_1^0 . An edge between a complex cell plane C_p^{l-1} and a simple cell plane S_p^l denotes a receptive field profile $F_q^{l,p}$.

B. Simple cells

The activation of simple cells in plane S_p^l is computed in two steps: First, we sum up the results of convolving the activations of the complex cell planes of the previous layer C_q^{l-1} with corresponding receptive field profiles $F_p^{l,q}$, $q = 1 \dots P_{l-1}$:

$$\hat{S}_p^l = \sum_{q=1}^{P_{l-1}} C_q^{l-1} \otimes F_p^{l,q}, \quad (1)$$

where \otimes denotes convolution. Note for the simple cells of the first layer the previous layer is simply the input image. Second, to compute the final (binary) activation of each cell in S_p^l , a “winner takes most” plane-wise competitive mechanism [3] is performed among all cells that are located at a position (x, y) in the planes $\hat{S}_p^l, p = 1 \dots P_l$:

$$S_p^l(x, y) = \begin{cases} 0 & \text{if } M = 0 \text{ or} \\ & \frac{\hat{S}_p^l(x, y)}{M} < \gamma_l \text{ or} \\ & \frac{\hat{S}_p^l(x, y) - \gamma_l M}{1 - \gamma_l} < \theta_l, \\ 1 & \text{else,} \end{cases} \quad (2)$$

where $M = \max_p \hat{S}_p^l(x, y)$, γ_l with $0 < \gamma_l < 1$ is the “competition strength”, and θ_l is the “activation threshold” common to all planes in layer l . See [3] for a detailed discussion on this nonlinear step.

The underlying processing principle of the first step can be understood as a type of *weight sharing*, that has the following effect: Instead of using a different weight for every spatial position, one and the same receptive field profile is applied to every position of the input plane by means of convolution. This contributes to the robustness in the sense, that a stimulus that leads to a certain local activation of a receptive field will cause the same activation in a neighboring cell under a minor change of position. This fact is exploited by the *spatial pooling* mechanism (see Section II-C) to achieve robustness with respect to small spatial translations of local parts of the input stimulus. The second nonlinear step is used to strengthen high responses of some cells, while discarding weak responses of others. This yields a segmentation of the input, i.e. the output of the previous layer, into regions, where a particular feature is dominant.

C. Complex cells

The activation of a complex cell plane C_p^l (which is usually chosen to be smaller in size than the simple cell planes in

the same layer) is directly derived from its corresponding S_p^l plane. The activation of a cell C_p^l at position (x, y) is computed by weighted spatial pooling over a neighborhood of corresponding simple cells.

$$C_p^l(x, y) = \sum_{(x', y') \in H_l(x, y)} G_l(x', y'; x, y) * C_p^l(x', y'), \quad (3)$$

where $H_l(x, y)$ is a neighborhood function for layer l , that returns a set of corresponding cell positions in S^l within a square of $\sigma_l \times \sigma_l$. $G_l(x', y'; x, y)$ is a Gaussian with variance σ_l , centered at the C^l cell position corresponding to (x, y) .

This spatial pooling mechanism is motivated by a major property of biological complex cells in the visual cortex, which is *position insensitivity*: Response rates of a complex cell are not much affected by small differences in the position of a stimulus on the retina [12]. Several authors suggest, that in a computational model, this type of behavior can be resembled by a spatial pooling mechanism [3], [7], i.e. by combining the responses of a number of simple cells within a neighboring region. In the definition above, this pooling is achieved by Gaussian convolution and sub-sampling.

In the following, we will describe, what kind of profiles are used in this work on each of the three layers in order to apply the network to the identification and detection tasks described in III.

D. Layer One – General Feature Extraction

The choice of receptive field profiles for the three layers is motivated by the idea, that each network layer should perform a feature extraction at an increasing level of “specificity”. As a consequence, profiles on first layer are not specific at all, but perform a “general” feature extraction. In the experiments in this work we choose $P_1 = 4$ and use for the first layer, i.e. for $F_1^{1,p}$, $p = 1 \dots 4$, first-order even Gabor kernels at 0, 45, 90 and 135 degrees[3] as *fixed* receptive field profiles. This choice is motivated by the fact that efficient coding on natural image patches yields Gabor like receptive fields [13], [4], [14]. Together with the “winner takes most” nonlinearity, processing on the first layer yields a segmentation of the input stimulus based on four dominant edge orientations. Figure 2 shows an example of processing an image in the first layer. (Here, the input image is of dimension 64×64 , the size of the complex cell planes C^1 is set to 32×32 , the competition strength γ_1 is set to 0.9, the activation threshold θ_1 is set to 0.1 and the spatial pooling parameter σ_1 is set to 3.0.)

E. Layer Two – Domain Specific Feature Extraction

In contrast, the profiles on the second layer are specialized to the image domain in the sense of extracting “typical” features. As opposed to the work in [3], in our experiments, these profiles are obtained by efficient coding using a Non-negative Matrix Factorization algorithm with Sparseness Constraints (NMFSC), a method recently proposed by Hoyer [14]. It was shown to have better properties than other coding methods like Sparse Coding, standard NMF or ICA, because

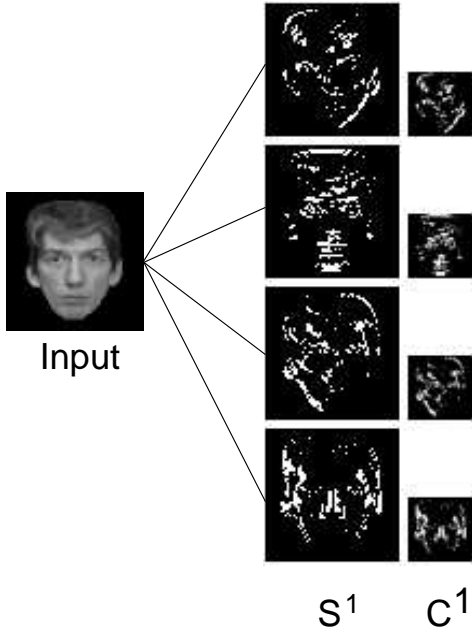


Fig. 2. Layer 1. An input image is convolved using first-order even Gabor kernels at four different orientations. The result is processed by the "winner takes most" nonlinearity (S^1) and by the spatial pooling mechanism (C^1).

sparseness of both the feature matrix and the latent variables can be controlled explicitly. Since this feature extraction is not the main focus of this paper, the reader is referred to the Appendix and to [14] for details. Figure 3 shows an example of processing the output of the first layer (see Fig. 2) in the second layer. (Here, the size of the complex cell planes C^2 is chosen to be identical to the first layer (32×32), the competition strength γ_2 is set to 0.9, the activation threshold θ_2 is set to 0.1 and the spatial pooling parameter σ_2 is set to 3.0.)

F. Layer Three – View Tuned Units

Profiles on the third layer are specialized to faces. In the present work, we set the number of planes P_3 to the number of classes (persons) and each unit can be thought of taking the role of a "grand-mother cell" being sensitive to all different views of one specific face. The profiles are obtained by supervised learning of linear discriminator functions as explained in the following: Given a labeled set D of training input images, where $\phi_{target}(I), I \in D$ stores a class index for every image. The indices are enumerated from 1 to $N_{classes}$. After passing all training examples through the first two network layers – assuming the profiles on the second layer are already trained (see last section) – and recording the complex cell activations of the second layer as $C^2(I)$ for each example, we obtain a set of $N_{classes}$ VTUs by minimizing the following error function with respect to F^3 :

$$E(F^3) = \sum_{I \in D} \sum_{p=1}^{N_{classes}} \delta(\phi_{target}(I), p) - \sum_{q=1}^{P_2} C_q^2(I) * F_p^3, \quad (4)$$

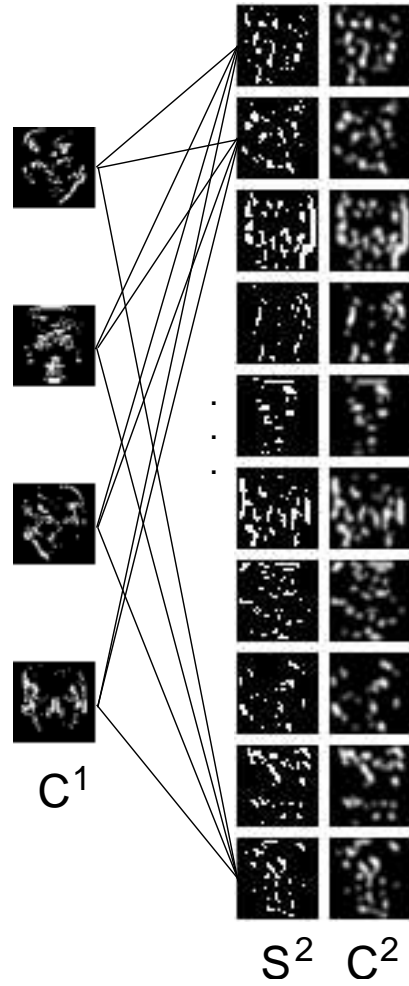


Fig. 3. Layer 2. The output of layer 1 (C^1) is convolved by the domain specific profiles that were obtained by the un-supervised NMFSC algorithm. The result is again processed by the "winner takes most" nonlinearity (S^2) and by the spatial pooling mechanism (C^2).

where $\delta(x, y)$ is set to 0.9 if $x = y$ and 0.1 else. After training, the obtained VTUs are used as the receptive field profiles of layer 3. This allows us to use the network to process a larger test image, and by convolution, the VTUs yield activation peaks on the final C^3 layer, which indicate the presence of a particular face at that position. Figure 4 shows an example image (as used in the experiments described in Section III-B) being processed in layer 3. (Here, the size of the input image is of dimension 256×256 , which means, that all other dimension parameters are adjusted accordingly. The dimension of the final complex cell planes C^3 is chosen to be 30×30 , the "winner takes most" mechanism is not applied at the third layer, and the spatial pooling parameter σ_3 is set to 3.0.)

III. EXPERIMENTAL RESULTS

In this section, we present experimental results on artificial datasets, that are generated using the ORL face dataset (10 frontal views of faces of each of 40 different persons) and the ArtExplosion Image Library [11] (high resolution images of natural scenes). For all experiments in this work, it was

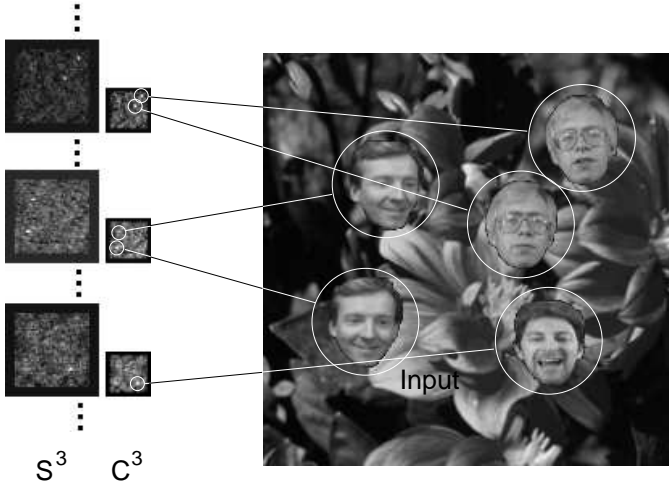


Fig. 4. Layer 3. The input test image is passed through the first two layers of the network and the final outputs of layer 2 (C^2) are convolved using the View Tuned Units obtained by supervised learning. The resulting maps (S^3) clearly exhibit local activation peaks, that indicate the presence of a particular face at that position. Here, only the planes, that correspond to the faces in the test image are displayed. The detection of local maxima is performed on the C^3 planes. (Note, that the “winner takes most” mechanism is not applied at layer 3).

necessary to manually segment the face images from the ORL dataset, i.e. replacing all background pixels by 0, which is needed for dataset generation as described below. In the first experiment, we will evaluate the classification performance of the two-layer network by applying the VTUs to the activation of the C^2 planes using a distorted test dataset. In the second experiment, we will apply the complete three-layer network for a detection and identification task.

A. Classification Performance of the Two-layer Network

For this experiment, we divide the ORL dataset into a training and a test set, both containing five views of each person. This gives us 200 images in each set. All test images are randomly scaled by $\pm 10\%$, randomly shifted by ± 5 pixels in x and y direction, and clutter is added as background. Also, we generate 200 additional images that only contain clutter. Clutter images are obtained by randomly cropping windows from images of the ArtExplosion library. Fig. 5 shows some examples of the training and the test images. From the training set we use an increasing number of views of each person’s face (1...5) to obtain five sets of 40 View Tuned Units as described in section II-F. We then pass the test images through the network and apply all View Tuned Units to the resulting C^2 activations. The index of the maximally active unit then yields the classification result:

$$\phi(I) = \arg \max_p \sum_{q=1}^{P_3} C_q^2 * F_p^{3,q}, p = 1 \dots P_3. \quad (5)$$

An input image is rejected as an unknown pattern, if the activation of the “winner” unit is below a threshold θ_3 , i.e.



Fig. 5. Example images for experiment 1. Top: Manually segmented training images. Middle: Distorted test images. Bottom: Clutter images to be rejected as ‘unknown’. (Note, that manual segmentation of the training images is not generally required, but only needed for dataset generation in the experiments in this work.)

$$\sum_{q=1}^{P_3} C_q^2 * F_{\phi(I)}^{3,q} < \theta_3 \quad (6)$$

Varying the threshold parameter θ_3 , the following quantities are counted:

- *True Positives (TP)*: The test image contains a face and the correct unit has maximum activation above threshold.
- *False Positive (FP)*: The test image does not contain a face, but not all units have activation below the threshold, or, the test image does contain a face, but an incorrect unit has maximum activation above threshold.
- *True Negative (TN)*: The test image does not contain a face and all units have activation below threshold.
- *False Negative (FN)*: The test image contains a face, but no unit has activation above threshold.

Figure 6 shows the ROC curves for using 1,3 and 5 views, plotting sensitivity ($\frac{TP}{TP+FN}$) vs. 1-specificity ($1 - \frac{TN}{FP+TN}$). As one might expect, the classifier performs better, the more views are used for training. Assuming equal importance of sensitivity and specificity, we can say, that using five views, a performance of approx. 88% can be achieved.

B. Detection Performance of the Three-layer Network

For the object detection experiment, we use the full three-layer network, which is first trained on small image patches of size 64×64 , and then exposed to larger test images of size 256×256 . The test images are generated from random ArtExplosion images, into which manually segmented face images of the ORL face dataset are placed at random positions. For every image these positions are memorized as ground truth for evaluation of the detection results. One example of a test image is shown in Fig. 4.

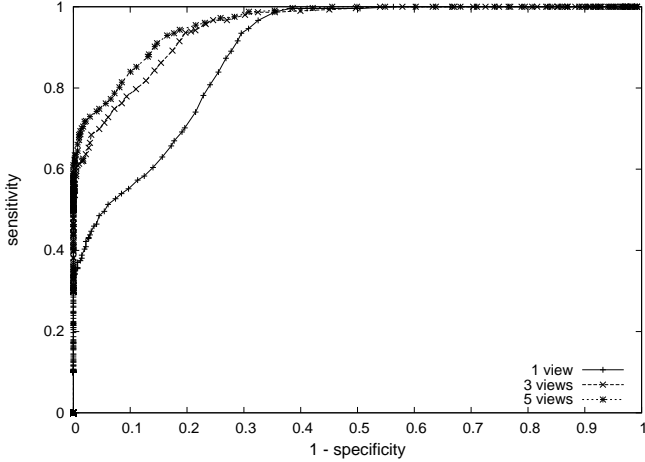


Fig. 6. Results of experiment 1. Classification performance of the two-layer network using distorted test data and 1, 3, and 5 views of each person’s face for training.

After passing a test image through the network architecture, we obtain the detection result by local maxima detection on the final C^3 complex cell planes. If a maximum is found, an object is detected if (i) there is no higher maximum within a radius of $h = 3$ cell positions on a different plane and (ii) the activation at this position is above a threshold θ_3 . The class index is derived from the plane index. For a given set of test images, we then count the following quantities:

- *True positives (TP)*: A detected face is present at that location and it is classified correctly.
- *False positives (FP)*: A detected face is either not present or classified incorrectly.
- *False negatives (FN)*: A face is present, but it has not been detected.

In order to judge whether a detected face position matches the ground truth (see above), we allow for an inaccuracy of ± 5 pixels (This inaccuracy occurs, because the positions on the final C^3 layer have to be super-sampled to match the dimensions of the input image).

For a given set of test images, we can then plot *sensitivity* ($\frac{TP}{TP+FN}$) vs. *positive predictive value (PPV)* ($\frac{TP}{TP+FP}$) under a varying threshold θ_3 . An example of such a plot is shown in Fig. 7, where the VTUs were trained using five views of each person and the test set of 100 images was generated using the same five views. Assuming equal importance of sensitivity and PPV, we can say that the best performance is achieved for the threshold value at the intersection of the sensitivity and the PPV curves. Therefore, a detection performance of approx. 86% can be reached for this test set.

Fig. 8 shows the performance for other sets. The x-axis denotes the number of views (1...5), that are used for training. The dark-colored bars show the performance for test sets, that are generated using the same views as used for training. As one might expect, the performance decreases, the more views are used for training and testing. The light-colored bars show the performance against "unseen" views, where the test images are generated from five views of each person, that

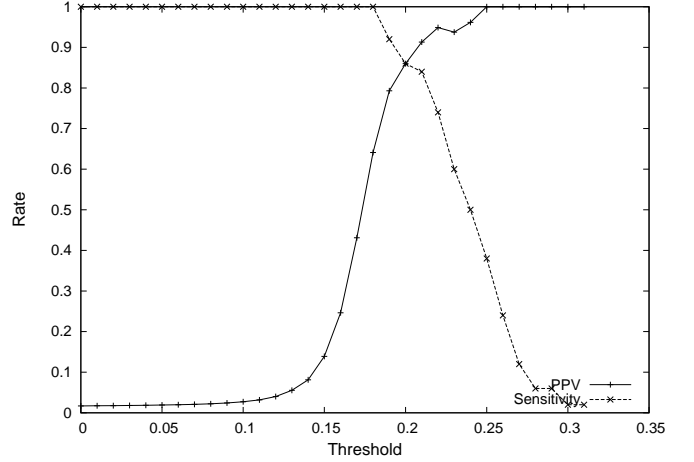


Fig. 7. Example plot for experiment 2. Detection performance of the three-layer network, which was trained using five views of each person and applied to 100 test images, that were generated using the same five views.

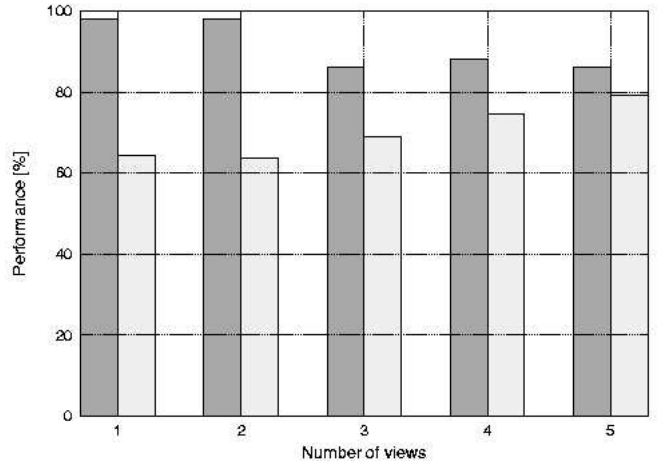


Fig. 8. Results of experiment 2. Detection performance of the three-layer network. The x-axis denotes the number of views of each face, that were used for training. The dark-colored bars show the best achievable performance for using the same views for generating test images. The light-colored bars show the performance for test images that were generated using five 'unseen' views, that were not used for training.

Here, a performance of approx. 79% can be reached when using five training views.

IV. SUMMARY AND CONCLUSION

In this contribution we applied a hierarchical feed-forward recognition architecture to the problem of face recognition. The advantage of the approach is, that it can simultaneously perform detection and identification. This was achieved by extending a recently proposed two-layer model for patch classification to a three-layer model, that allows detection and identification. We evaluated the performance on artificial test datasets, that were generated from images of natural faces and natural cluttered background and showed, that the network achieves high performance for a patch classification task with distorted test data, as well as for a detection and identification

task in cluttered surround.

We believe, that the approach is promising to be applied in real-world computer vision applications, such as person identification. Future work will be concerned with testing the approach in such environments.

APPENDIX FEATURE CODING USING NMFSC

In this appendix we describe the use of Non-negative Matrix Factorization with Sparseness Constraints (NMFSC) [14] for feature coding on the second network layer: To obtain a training set for the feature coding procedure in layer 2, we first apply layer 1 of the network to a set of training images. Patches of size $d_{F^2} \times d_{F^2}$ are extracted at random positions from the activation of C^1 cell planes. Concatenating these sample patches yields vectors of dimension $d_{F^2} * d_{F^2} * P_1$. The vectors are used as the columns of a data matrix V which is subsequently decomposed using the NMFSC algorithm [14].

The algorithm solves the problem $V \approx WH$, where W denotes the feature matrix and H the latent matrix. The inner dimension of WH is set to P_2 . The solution is obtained by minimizing the MSE between WH and V under explicit sparseness constraints $0 < W_s < 1$ (the sparseness of columns of W) and $0 < H_s < 1$ (the sparseness of rows of H), and the additional constraints of non-negativity for matrices W and H . The algorithm also allows us to omit W_s or H_s causing the standard NMF learning rules [15] to be used (refer to [14] for details). After decomposition, each column p of W is normalized and the values are used to obtain the receptive field profiles F_q^{2p} , for $p = 1 \dots P_2$ and $q = 1 \dots P_1$. Figure 9 shows an example of 10 profiles learned with NMFSC.

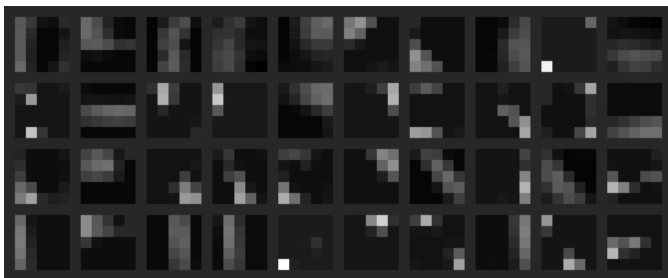


Fig. 9. Feature coding using NMFSC. Example of 10 receptive field profiles of dimension $5 \times 5 \times 4$ learned with the NMFSC algorithm. The fields are arranged column-wise.

ACKNOWLEDGMENTS

This work was conducted within the project VAMPIRE (Visual Active Memory Processes and Interactive REtrieval) which is part of the IST program (IST-2001-34401).

REFERENCES

- [1] K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position." in *Biol. Cybern.*, 1980, pp. 36:193–202.
- [2] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *Journal of Physiology*, vol. 148, pp. 574–591, 1959.
- [3] H. Wersing and E. Körner, "Learning optimized features for hierarchical models of invariant object recognition," *Neural Comp.*, vol. 15, no. 7, pp. 1559–1588, 2003. [Online]. Available: citeseer.ist.psu.edu/wersing02learning.html
- [4] B. A. Olshausen and D. J. Field, "Emergence of simple cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, 1996.
- [5] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library: COIL-100," Dept. Computer Science, Columbia Univ., Tech. Rep. CUCS-006-96, 1996.
- [6] *The Database of Faces*, AT&T Laboratories Cambridge, 2002. [Online]. Available: <http://www.uk.research.att.com/facedatabase.html>
- [7] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in visual cortex," *Nature*, vol. 2, no. 11, pp. 1019–1025, 1999.
- [8] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 1, pp. 23–38, 1998.
- [9] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, pp. 71–86, 1991.
- [10] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. Conf. Computer Vision and Pattern Recognition CVPR*, 2001.
- [11] *Art Explosion Photo Gallery*, Nova Development Corporation, 23801 Calabasas Road, Suite 2005 Calabasas, California 91302-1547, USA.
- [12] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture of the cat's visual cortex," *Journal of Physiology*, vol. 160, pp. 106–154, 1962.
- [13] A. J. Bell and T. J. Sejnowski, "The independent components of natural images are edge filters," *Vision Research*, vol. 37, no. 27, pp. 3327–3338, 1997.
- [14] P. O. Hoyer, "Non-negative Matrix Factorization with Sparseness Constraints," *Machine Learning Research*, vol. 5, no. 37, pp. 1457–1469, 2004.
- [15] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, 2000, pp. 556–562. [Online]. Available: citeseer.ist.psu.edu/lee01algorithms.html