A Hierarchical Feed-forward Network for Object Detection Tasks

Ingo Bax[a], Gunther Heidemann[a] and Helge Ritter[a]

[a]Neuroinformatics Group, Faculty of Technology, Bielefeld University,
P.O. Box 10 01 31, D-33501 Bielefeld, Germany

ABSTRACT

Recent research on Neocognitron-like neural feed-forward architectures, which have formerly been successfully applied to recognition of artifical stimuli like paperclip objects, is promising application to more natural stimuli. Several authors have shown high recognition performance of such networks with respect to translation, rotation, scaling and cluttered surroundings. In this contribution, we introduce a variation of existing hierarchical models, that is trained using a non-negative matrix factorization algorithm. In contrast to previous work, our approach can not only classify objects but is also capable of rapid object detection in natural scenes. Thus, the time-consuming and conceptually unsatisfying split-up into a localization stage (e.g. using segmentation) and a subsequent classification can be avoided. Though in principle an exhaustive search by classification of every sub-window of an image is performed, the process is nevertheless highly efficient. The network consists of alternating layers of simple and complex cell planes and incorporates nonlinear processing schemes that have been proposed in recent literature. Learning of receptive field profiles for the lower layers of the network takes place by unsupervised learning whereas a final classification layer is trained supervised. Detection is achieved by attaching an additional network layer, whose simple cell profiles are learned from the final classification units that were acquired during the training phase. We test the classification performance of the network on images of natural objects which are systematically distorted. To test the ability to detect objects, cluttered natural background is used.

Keywords: Neural Processing, Object Recognition, Object Detection, Neocognitron, Non-negative Matrix Factorization

1. INTRODUCTION

In recent years, advances have been made in solving problems of visual recognition in restricted environments like, e.g., detecting nonconforming products on an assembly line or recognizing isolated objects under fixed lighting conditions. However, as computer vision applications are about to enter also unrestricted environments as in the case of mobile vision systems, mobile robots, vehicle navigation aids or surveillance systems, tasks become more difficult because the recognition system has to cope with distortions caused by varying illumination, arbitrary view points, cluttered scenes, object deformations, partial occlusion, etc. Therefore, a major goal of modern vision research is finding approaches that allow recognition, which is *invariant* under distortion.

Since the visual systems of humans and animals seem to have few problems in solving difficult recognition tasks, it has become popular – and also quite successful – to take into account physiological and psychophysical findings about information processing principles in the brain to build artificial vision systems. Modern approaches that follow this paradigm often rely on the early findings by Hubel and Wiesel,[1] who determined receptive fields of *simple cells* and *complex cells* in the primary visual cortex of mammals, and by Barlow,[2] who analyzed the behavior of these cells and firstly suggested that their response properties might emerge from an efficient coding strategy in the sense of information theory.

Further author information: (Send correspondence to Ingo Bax)
Ingo Bax: E-mail: ibax@techfak.uni-bielefeld.de, Telephone: +49 521 106 6891
Gunther Heidemann: E-mail: gheidema@techfak.uni-bielefeld.de, Telephone: +49 521 106 6891
Helge Ritter: E-mail: helge@techfak.uni-bielefeld.de, Telephone: +49 521 106 6891

A computational model to account for the idea of efficient coding was introduced by Olshausen and Field,[3] who proposed the notion of *sparse coding* as a strategy of learning receptive fields from natural image data. The method produces results qualitatively similar to those obtained by Independent Component Analysis (ICA).[4] A recognition architecture that is based on a hierarchical organization of layers of simple and complex cell arrays was introduced by Fukushima,[5] called the *Neocognitron.* The network performs invariant recognition of simple visual stimuli like paperclip objects. More recently, Wersing and Körner[6] used a 2-layer variation of the Neocognitron architecture, which learns receptive fields using a special type of sparse coding algorithm with invariance constraints. The network achieves high invariance performance for classification of images patches that contain natural stimuli like objects or faces.

In the present work we use a similar architecture, that uses a Non-negative Matrix Factorization algorithm and show that such a recognition model can also be used for detection tasks, i.e. finding objects in a larger image. This is done by attaching an additional network layer, which uses features that are the result of supervised learning based on responses on the second network layer.

In the next section we introduce the hierarchical network model, in section 3, we describe experiments for classification of small image patches and object detection in large images, and discuss the results in section 4.

2. THE HIERARCHICAL MODEL

The hierarchical model described in this section is related to several architectures proposed earlier.[5–7] It consists of alternating layers of simple and complex cell planes, each of which performs feature extraction using receptive field profiles that are *fixed* on the first layer, the result of *unsupervised learning* on the second layer and *supervised learning* on the third layer. In the next section we will first describe the feed-forward processing scheme, then discuss the underlying principles of *Weight Sharing* and *Spatial Pooling*, which lead to the robust recognition capabilities of the network, and finally describe, how receptive field profiles are obtained.

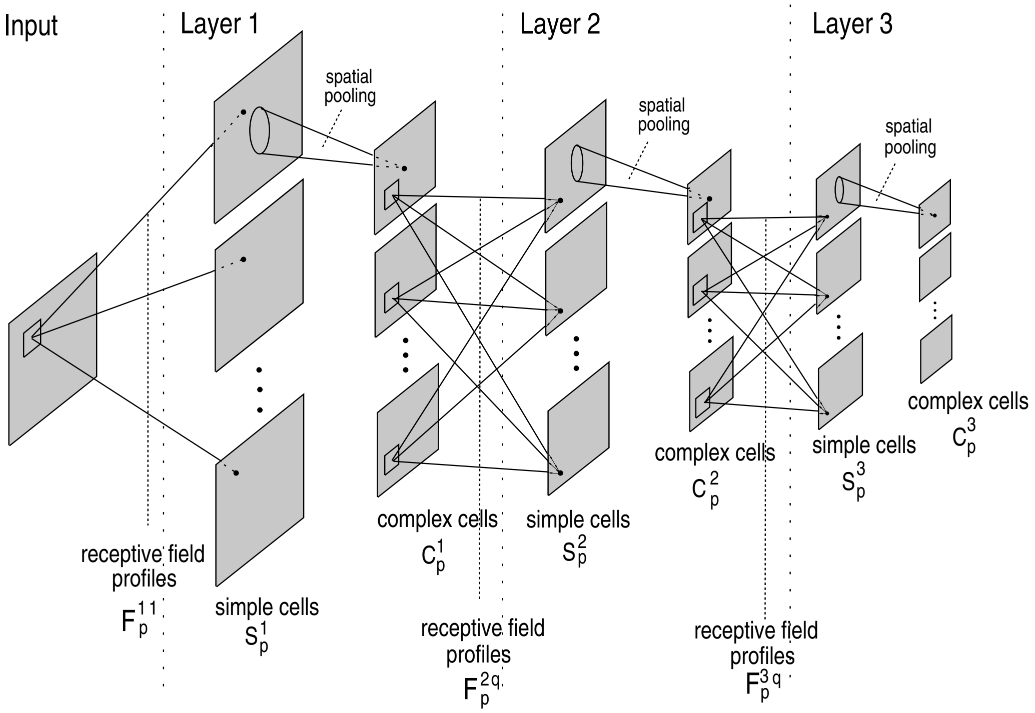

Figure 1. *The Hierarchical Model.* The network consists of three alternating layers of simple and complex cell planes. See text for explanation.

2.1. Feed-forward Processing

Figure 1 shows a diagram of the hierarchical model used for the experiments in this paper. It is characterized by the following properties:

- *Topology:* The model consists of $L = 3$ layers, indexed $l = 1 \dots L$ and each holding P_l planes of two types: *simple cell planes* S^l_p and *complex cell planes* C^l_p 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^0_1. An edge between a complex cell plane C^{l-1}_p and a simple cell plane S^l_p denotes a receptive field profile $F^{l\ p}_q$.

- *Computing simple cell plane activation:* The activation of simple cells in plane S^l_p is computed in two steps: First, we sum up the results of convolution of the activations of the complex cell planes of the previous layer C^{l-1}_q with corresponding receptive field profiles $F^{l\ q}_p$, $q = 1 \dots P_{l-1}$:

$$\hat{S}^l_p = \sum_{q=1}^{P_{l-1}} C^{l-1}_q \otimes F^{l\ q}_p, \tag{1}$$

  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^l_p, a "winner takes most" plane-wise competitive mechanism[6] is performed among all cells that are located at a position (x, y) in the planes $\hat{S}^l_p, p = 1 \dots P_l$:

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

  where $M = \max_p \hat{S}^l_p(x,y)$, 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[6] for a detailed discussion on this nonlinear step.

- *Computing complex cell plane activation:* The activation of a complex cell plane C^l_p (which is smaller in size than the simple cell planes in the same layer) is directly derived from its corresponding S^l_p plane. The activation of a cell C^l_p at position (x, y) is computed by weighted spatial pooling over a neighborhood of corresponding simple cells:

$$C^l_p(x,y) = \tanh\left(\sum_{(x',y') \in H_l(x,y)} G_l(x',y';x,y) * C^l_p(x',y') \right), \tag{3}$$

  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).

2.2. Processing Principles

There are two concepts utilized in the feed-forward processing scheme of the model that require comment, because they give rise to the invariant recognition capabilities of the network:

- *Weight Sharing:* The concept of weight sharing is often used to reduce the training complexity of neural networks. However, the type of weight sharing, that is present in the current hierarchical network is hidden in the calculation of the simple cell plane activations and has an additional purpose: Instead of having 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 type of weight sharing contributes to the the robustness: 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 to achieve robustness with respect to small spatial translations of local parts of the input stimulus.

- *Spatial Pooling:* One major functional property of complex cells in the visual cortex 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.[8] Several authors suggest, that in a computational model, this type of behavior can be resembled by a spatial pooling mechanism,[6,7] i.e. by combining the response of a number of simple cells within a neighboring region. In the definition above, this pooling is archived by Gaussian convolution and sub-sampling. Passing the response through a tanh transfer function implements a smooth spatial or-operation.[6]

2.3. Receptive Field Profiles

The choice of receptive field profiles for the three layers is motivated by the idea, that each network layer should perform 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 $F_1^{1\ p}$, $p = 1 \ldots 4$, first-order even Gabor kernels at 0, 45, 90 and 135 degrees[6] as *fixed* receptive field profiles. This choice is motivated by the fact that efficient coding on natural image patches yields Gabor like receptive fields.[3,4,9] 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.

In contrast, the profiles on the second layer are specialized to the image domain in the sense of extracting "typical" features. These profiles are obtained by efficient coding using a Non-negative Matrix Factorization algorithm with Sparseness Constraints (NMFSC), a method recently proposed by Hoyer.[9] It was shown to have better properties than other coding methods like Sparse Coding, standard NMF or ICA, because sparseness of both the feature matrix and the latent variables can be controlled explicitly. In the experiments in this paper we set $P_3 = 10$, which is sufficient for achieving high recognition rates on the chosen test datasets. The reader is referred to appendix A for more details.

Profiles on the third layer are specialized to objects. In the present work, we set the number of planes P_3 to the number of object classes and each unit can be thought of taking the role of a "grand-mother cell" being sensitive to all different views of one specific object. They correspond to the "view tuned units" used in related models.[6,7] The profiles are obtained by supervised learning of linear discriminator functions as explained in appendix B.

3. CLASSIFICATION AND DETECTION EXPERIMENTS

In this section we describe computer simulations of the model outlined above. We consider two experimental settings: In the first setting, a 2-layer network is used to test the invariance performance of the view tuned units for the problem of patch classification. This experiment is described in section 3.1. The second setup uses the complete 3-layer network for an object detection task. This experiment is described in section 3.2.

The datasets used for both experiments contain artificially generated test stimuli from two natural image databases. The first is the COIL-20 dataset,[10] an image library that contains 72 views of 20 different objects which were recorded using fixed lighting conditions and a turn table. Since the objects of the COIL-20 dataset appear in front of a black background, we can easily perform a figure-ground separation to embed the objects into a natural surrounding taken from the Art Explosion dataset.[11] This way, we can obtain large labeled datasets of stimuli, that contain image data which is sufficiently close to natural conditions to make predictions about the performance of network for applications in natural environments.

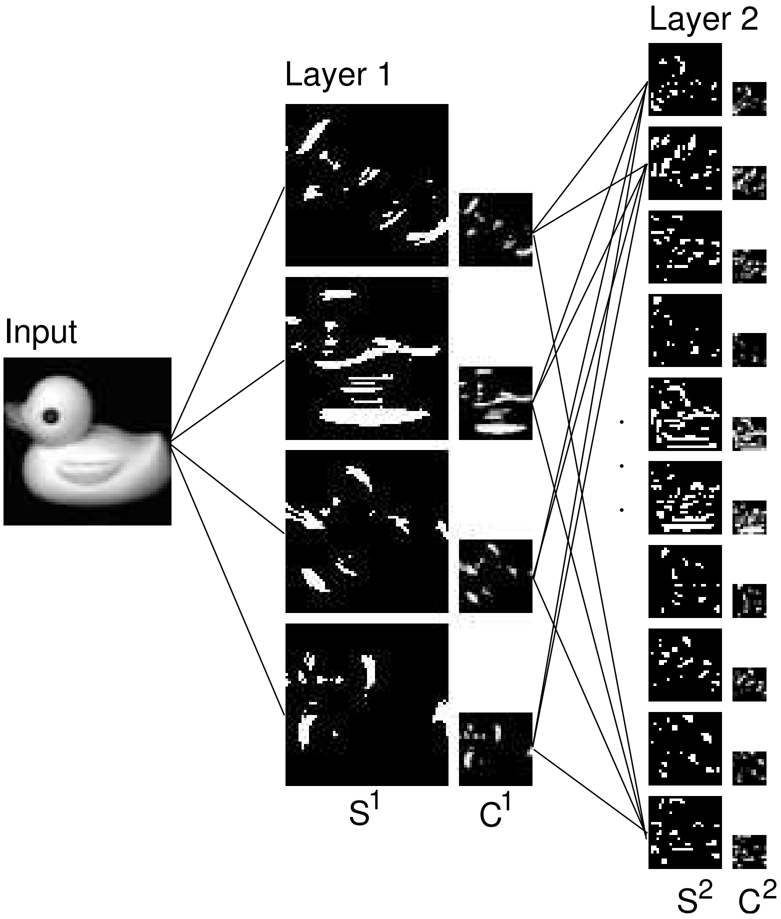

Figure 2. *Processing example of a 2-layer network: The gray value input image is convolved using Gabor kernels at 4 different orientations. The result is subject to a "winner takes most" nonlinearity. The final binary activation of the simple cells in the first layer (S^1) is input to a "spatial pooling" mechanism, whose result is displayed in the complex cells of the first layer (C^1). This serves as input for processing in the second layer. All planes are convolved using receptive field profiles and the result (S^2) is again subject to the "winner takes most" and the "spatial pooling" mechanisms. The output of the 2-layer network is the activation of the complex cell planes of the second layer (C^2)*

3.1. Patch Classification Experiment

Figure 2 shows an example of passing an image patch I through the 2-layer architecture. The activity on the second complex cell layer is the output of the network. From the activations, the classification answer $\phi(I)$ is determined by computing the inner products with all view tuned units. The index of the unit with maximum activity yields the classification result:

$$\phi(I) = \arg\max_{p} \sum_{q=1}^{P_2} C_q^2 * F_p^{3\ q}, p = 1 \dots P_3. \tag{4}$$

We can also reject a stimulus as an unknown pattern, if the activity of the "winner" view tuned unit is below a threshold θ_3, i.e.

$$\sum_{q=1}^{P_3} C_q^2 * F_{\phi(I)}^{3\ q} < \theta_3 \tag{5}$$

For the experiment, we use the COIL-20 dataset,[10] an image library, which contains 72 views of each of 20 different objects. We divide the database into a training set (all odd-numbered views) and a test set (all even-numbered views). For training, we vary the number of views from 1-36, that are used to obtain the profiles on the second layer (F^2) and the view tuned units on the third layer (F^3). For testing, we always use the whole test set, which is additionally distorted by scaling (+/− 10%) and by placing natural clutter in the background of the objects. The clutter is randomly taken from images of the Art Explosion database.[11] Figure 3 (left and middle) shows some examples of the images used in this experiment. The result is show in the left diagram of Fig 4. As one might expect, it can be seen, that an increasing amount of views used for training also leads to a higher rate of correct classification. Saturation is reached at about 10-20 views, an over-fitting effect can be observed if the number of views becomes larger than 20.

In order to test the rejection capability of the network, in a second experiment we double the size of the test set by adding images that only contain clutter from the Art Explosion test set. Figure 3 (right) shows examples of these additional test images. The result is shown in ROC curves for 1, 20 and 36 training views in Fig. 4 (varying the threshold parameter θ_3 from Eq. 5). The best performance is reached for 20 training views. Also, the over-fitting effect for 36 object views resembles for the rejection experiment.

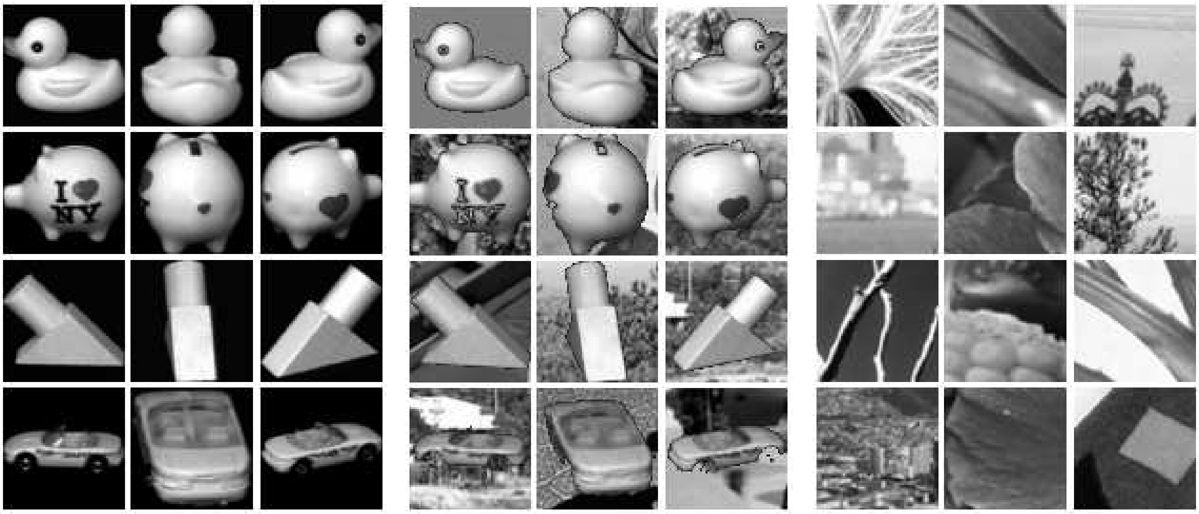

Figure 3. *Left: Examples of training images. Middle: Examples of distorted test images. Right: Examples of test images to be rejected as 'unknown'.*

3.2. Object Detection Experiment

For the object detection experiment, we use the full 3-layer network, which is first trained on small image patches of size 64 × 64, and then exposed to larger test images of size 512 × 512. The test images are generated from random Art Explosion images, into which images of the COIL-20 library are placed at random positions. For every image these positions are memorized as ground truth for evaluation of the detection results. Three examples of test images are shown in Fig. 5.

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 the experiment, we use an increasing number of object views (1-10) and create for each setting 100 test images. For evaluation of the detection results, we count the following quantities:

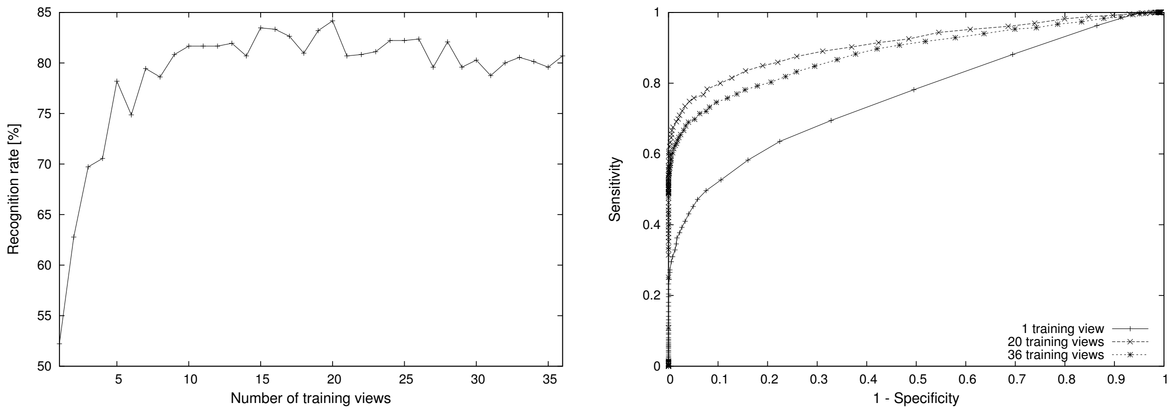

Figure 4. *Left: Result of the classification experiment: The recognition rate increases the more object views are used for training. Right: Result of the rejection experiment: ROC curves for different numbers of training views.*

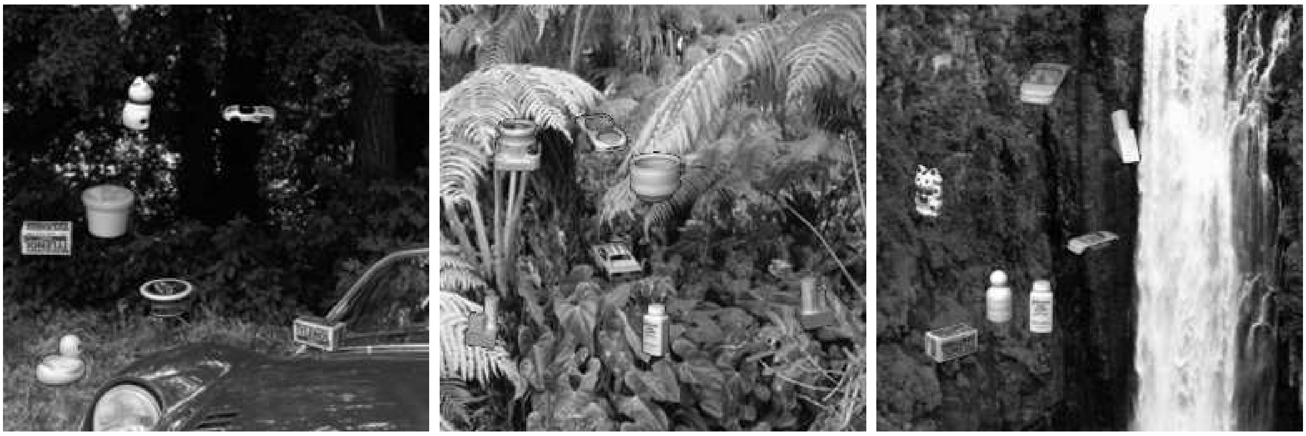

Figure 5. *Examples of test images used for the detection experiment. See text for further explanation.*

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

In order to judge whether a detected object position matches the ground truth (see above), we allow for an inaccuracy of +/− 3 pixels. The detection results for using 1, 5 and 10 object views are shown in Fig. 6, where *sensitivity* ($\frac{TP}{TP+FN}$) and *positive predictive value* ($\frac{TP}{TP+FP}$) are plotted for a varying threshold θ_3. Assuming equal importance of sensitivity and PPV, we can say that for using only one object view, a detection performance of approx. 76% can be reached. Of course, this value decreases substantially as we make the dataset more difficult by using more object views. This shortcoming could be avoided by using multiple view tuned units for each object, like also reported by Wersing and Körner[6] for a patch classification task. Incorporating this idea for object detection will be subject of future work.

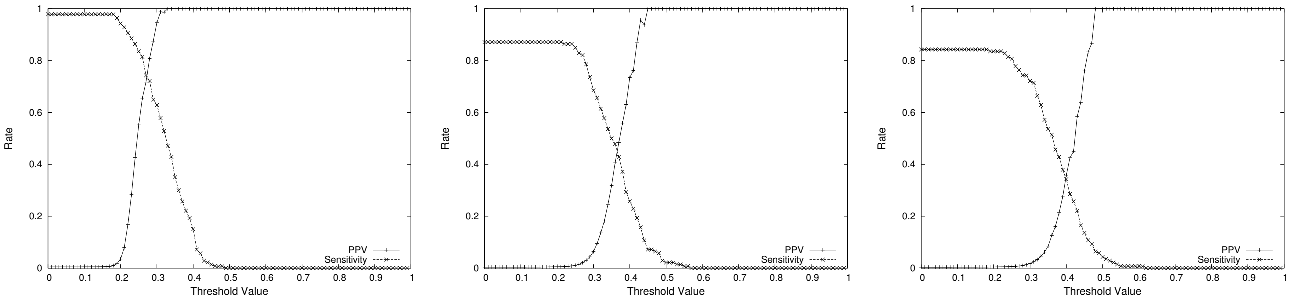

Figure 6. *Results of the detection experiment. Left: Only one object view is used. Middle: 5 object views. Right: 10 object views.*

4. SUMMARY AND CONCLUSION

In this contribution we have shown that Neocognitron-like hierarchical feed-forward recognition models can be utilized for object detection tasks in natural environments. We extended recently proposed models, that rely on efficient coding methods. These type of models are used to resemble the properties of receptive fields in the mammalian visual cortex. While it has been shown before, that this *domain specific* feature extraction greatly aids in image patch *classification* tasks where stimuli are subject to distortions such as deformation and clutter, we utilized this property also for the task of object *detection.* We believe that the approach is promising to be applied for computer vision systems, that operate in natural and unrestricted environments, like e.g. mobile robots or automated vehicles. Future work will be concerned with testing the approach in such environments.

APPENDIX A. FEATURE CODING USING NMFSC

In this appendix we describe the use of Non-negative Matrix Factorization with Sparseness Constraints (NMFSC)[9] 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.[9]

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 to omit W_s or H_s causing the standard learning rules[12] to be used (refer to[9] for details). After decomposition, each column p of W is normalized and the values are used to obtain the receptive field profiles $F_q^{2\ p}$, for $p = 1 \dots P_2$ and $q = 1 \dots P_1$. Figure 7 shows an example of profiles learned with NMFSC.

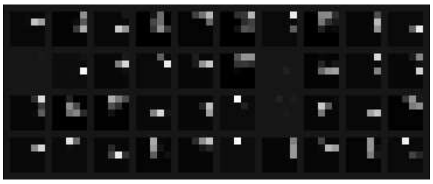

Figure 7. *Example receptive field profiles learned with the NMFSC algorithm.*

APPENDIX B. SUPERVISED LEARNING OF VIEW TUNED UNITS

In this appendix we describe how the receptive field profiles on the third network layer are obtained by supervised learning: Given a labeled set D_{train} of training input images, where $\phi_{target}(I), I \in D_{train}$ 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 Appendix A) – and recording the complex cell activations of the second layer as $C^2(I)$ for each example, we obtain a set of $N_{classes}$ view tuned units by minimizing the following error function with respect to F^3:

$$E(F^3; D_{train}) = \sum_{I \in D_{train}} \sum_{p=1}^{N_{classes}} \delta(\phi_{target}(I), p) - \sum_{q=1}^{P_2} C^2_q(I) * F^3_p{}^q, \tag{6}$$

where $\delta(x, y)$ is set to 0.9 if $x = y$ and 0.1 else. This corresponds to learning linear discriminator functions based on C^2 outputs.

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. D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurons in the cat's striate cortex," *Journal of Physiology* **148**, pp. 574–591, 1959.
2. H. B. Barlow, "Possible principles underlying the transformation of sensory messages," *Sensory Communication* , pp. 217–234, 1961.
3. B. A. Olshausen and D. J. Field, "Emergence of simple cell receptive field properties by learning a sparse code for natural images," *Nature* **381**, pp. 607–609, 1996.
4. A. J. Bell and T. J. Sejnowski, "The independent components of natural images are edge filters," *Vision Research* **37**(27), pp. 3327–3338, 1997.
5. K. Fukushima, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position.," in *Biol. Cybern.*, pp. 36:193–202, 1980.
6. H. Wersing and E. Körner, "Learning optimized features for hierarchical models of invariant object recognition," *Neural Comp.* **15**(7), pp. 1559–1588, 2003.
7. M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition on visual cortex," *Nature* **2**(11), pp. 1019–1025, 1999.
8. D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture of the cat's visual cortex," *Journal of Physiology* **160**, pp. 106–154, 1962.
9. P. O. Hoyer, "Non-negative Matrix Factorization with sparseness constraints," *Machine Learning Research* **5**(37), pp. 1457–1469, 2004.
10. S. A. Nene, S. K. Nayar, and H. Murase, "Columbia Object Image Library (COIL-20)," Tech. Rep. CUCUS-006-96, Dept. Computer Science, Columbia Univ. New York, N.Y. 10027, 1996.
11. Nova Development Corporation, 23801 Calabasas Road, Suite 2005 Calabasas, California 91302-1547, USA, *Art Explosion Photo Gallery.*
12. D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*, pp. 556–562, 2000.