Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Evolutionary Feature Design for Object Recognition with Hierarchical Networks

Georg Schneider, Heiko Wersing, Bernhard Sendhoff, Edgar Körner

2002

Preprint:

This is an accepted article published in Proceedings of the 9th International Conference on Neural Information Processing - ICONIP. The final authenticated version is available online at: https://doi.org/[DOI not available]

Proceedings of the 9th International Conference on Neural Information Processing, Singapore, November 2002.

EVOLUTIONARY FEATURE DESIGN FOR OBJECT RECOGNITION WITH HIERARCHICAL NETWORKS

Georg Schneider, Heiko Wersing, Bernhard Sendhoff and Edgar Körner

Honda R&D Europe (Deutschland) GmbH, Future Technology Research
Carl-Legien-Strasse 30, D-63073 Offenbach/Main, Germany
georg.schneider@hre-ftr.f.rd.honda.co.jp

ABSTRACT

A major problem in designing neural vision models is the large dimensionality of the search space for defining the needed networks. By using hierarchical vision models inspired by biology we narrow the space of possible architectures. We perform evolutionary optimization of remaining critical network parts e.g. the combination features, which are up to now mostly subject to manually determination. We show that the evolutionary approach leads to an optimized recognition system with respect to speed and performance, which is highly competitive with other state of the art systems.

1. INTRODUCTION

A critical problem in the application of artificial neural vision systems to object recognition tasks is the introduction of invariance properties. The correct recognition of presented visual objects should be robust under translation, scaling and rotation of the input stimuli. To incorporate these properties into a neural vision system a proper architectural design of the network is essential.

Evolution strategies provide a general and powerful method for system design optimization and their successful combination with neural networks has been shown in various applications [14]. In the work presented here we use evolution strategies to support the design process.

In order to apply evolutionary algorithms to the design of neural systems their structure and parameters must be represented or encoded. Most approaches use the so-called *direct coding* , e.g., via a *connection matrix*, where each entry represents a connection between two neurons. The main disadvantage of this method is the bad scaling property, since the representation scales quadratically with the number of neurons. In neural networks used for vision tasks, the number of neurons needed is immense and, therefore, the direct encoding is difficult to apply. Interesting approaches which focus on an indirect coding can be found in [3, 8]. Another way to prevent getting lost in an enormous search space of possible network architectures is to incorporate insights from neuroscience and in this way narrow the search space of possible network architectures. So inspired by the human visual cortex, we use a hierarchical vision system in which feature complexity is increasing from initial to later processing stages, and where invariance is achieved through pooling over increasing receptive fields [2, 7, 13]. One important part of a hierarchical model which still needs to be determined are the so-called *combination features*, which combine for example more elementary features like local edges into more complex patterns like corners and T-junctions. Methods which were proposed so far for the optimization of these features include unsupervised competitive learning combined with manually designed training patterns [2], supervised gradient-based optimization [4], enumeration heuristics [7], and sparse coding [13]. Few works use evolutionary methods to optimize hierarchical vision systems. Pan et. al. [6] optimize features with manually designed patterns as targets in intermediate stages of the vision architecture. Shi et. al. [12] on the contrary use genetic algorithms to build these patterns and use conventional supervised training methods of the neocognitron [2] to get the features.

In the work presented here we directly determine the combination feature filter bank using evolution strategies. We use a problem-specific direct coding and keep the dimensionality of the optimization sufficiently low by using a biologically inspired hierarchical architecture, described in the following section. In addition to the combination features, we also optimize important nonlinearities of the vision model architecture. The target value of the optimization is the classification performance of the vision network in an object recognition task. The details of the evolutionary optimization of the hierarchical vision system are described in Section 3. In Section 4 we demonstrate the generalization ability of the optimized feature set in object recognition tasks and compare the results to state-of-the-art algorithms.

We acknowledge partial funding of the LOKI project by the Bundesministerium für Bildung und Forschung (BMBF) grant 01IB001E.

In the last section, we summarize our results.

2. THE NEURAL VISION SYSTEM FOR OBJECT RECOGNITION

The used vision system for object recognition is based on a hierarchical feed-forward architecture with weight-sharing and a succession of feature-sensitive and pooling stages, see Fig.1. The first processing stage consists of a convolution with 4 differently oriented first-order Gabor filters, a Winner-Take-All mechanism between these features and a final threshold function. We adopt the notation, that vector indices run over the set of neurons within a particular feature plane of a particular layer. To compute the response $s_1^l(x,y)$ of a neuron in the first layer S1, responsive to feature type l at position (x,y), first the image vector $\mathbf{I}$ is multiplied with a weight vector $\mathbf{w}_1^l(x,y)$ (Gabor filter with orientation no. 1) characterizing the receptive field profile:

$$q_1^l(x,y) = |\mathbf{w}_1^l(x,y) * \mathbf{I}|. \qquad (1)$$

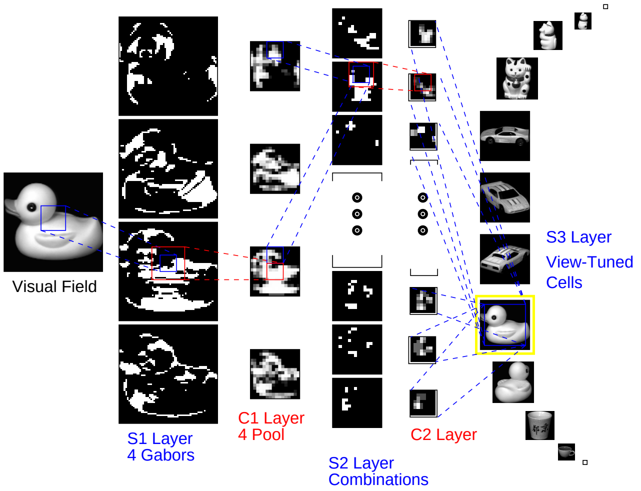

Figure 1: Sketch of the hierarchical network. The input image is presented as a 64×64 pixel image. The S1 layer consists of 4 Gabor feature planes at 4 orientations with a dimension of 64×64 each. The C1 layer subsamples by pooling down to a resolution of 16×16 for each of the 4 S1 planes. The S2 layer contains combination coding cells with possible local connections to all of the C1 cells. The C2 layer pools the S2 planes down to a resolution of 8 × 8. The final S3 cells are tuned to particular views, which are represented as the activity pattern of the C2 planes for an input image.

All neurons in a feature plane l have the same receptive field structure, given by $\mathbf{w}_1^l(x,y)$, but shifted receptive field centers. In a second step, a soft Winner-Take-All mechanism is performed with

$$r_1^l(x,y) = \begin{cases} 0 & \text{if } \frac{q_1^l(x,y)}{M} < \gamma_1 \text{ or } M = 0, \\ \frac{q_1^l(x,y) - M\gamma_1}{1-\gamma_1} & \text{else,} \end{cases} \qquad (2)$$

where $M = \max_k q_1^k(x,y)$ and $r_1^l(x,y)$ is the response after the WTA mechanism which suppresses sub-maximal responses and provides a model of latency-based competition. The parameter $0 < \gamma_1 < 1$ controls the strength of the competition. The activity is then passed through a simple threshold function with a common threshold θ_1 for all neurons in layer S1:

$$s_1^l(x,y) = \mathrm{H}\big(r_1^l(x,y) - \theta_1\big), \qquad (3)$$

where $\mathrm{H}(x) = 1$ if $x \geq 0$ and $\mathrm{H}(x) = 0$ else and $s_1^l(x,y)$ is the final activity of the neuron sensitive to feature l at position (x,y) in the S1 layer. The activities of the first layer of pooling C1-neurons are given by

$$c_1^l(x,y) = \tanh\big(\mathbf{g}_1(x,y) * \mathbf{s}_1^l\big), \qquad (4)$$

where $\mathbf{g}_1(x,y)$ is a normalized Gaussian pooling kernel with width σ_1, identical for all features l, and $\tanh$ is the hyperbolic tangent function. The features in the intermediate layer S2 are sensitive to local combinations of the features in the planes of the previous layer, and are thus called *combination neurons* in the following. We introduce the layer activation vector $\bar{\mathbf{c}}_1 = (\mathbf{c}_1^1, \ldots, \mathbf{c}_1^K)$ and the layer weight vector $\bar{\mathbf{w}}_2^l = (\mathbf{w}_2^{l1}, \ldots, \mathbf{w}_2^{lK})$ with K=4. Here $\mathbf{w}_2^{lk}(x,y)$ is the receptive field vector of the S2 neuron of feature l at position (x,y), describing connections to the plane k of the previous $C1$ neurons. The combined linear summation over previous planes is then given by $q_2^l(x,y) = \bar{\mathbf{w}}_2^l(x,y) * \bar{\mathbf{c}}_1$. After the same WTA procedure with strength γ_2 as in (2), the activity in the S2 layer is given by $s_2^l(x,y) = H(r_2^l(x,y) - \theta_2)$ after thresholding with a common threshold θ_2. The step from S2 to C2 is identical to (4) and given by $c_2^l(x,y) = \tanh(\mathbf{g}_2(x,y) * \mathbf{s}_2^l)$, with Gaussian spatial pooling kernel $\mathbf{g}_2(x,y)$ with range σ_2.

Classification of an input image with C2 output $\bar{\mathbf{c}}_2$ is done by nearest neighbor match to previously stored template activations $\bar{\mathbf{c}}_2^v$ for each training view v. This can be realized e.g. by view-tuned units (VTU) in an additional S3 layer with a radial basis function characteristics according to $s_3^v = \exp(-||\bar{\mathbf{w}}_3^v - \mathbf{c}_2||^2)$ where $\bar{\mathbf{w}}_3^v = \mathbf{c}_2^v$ is tuned to the training C2 output of pattern v. Classification can then be performed by detecting the maximally activated VTU.

3. EVOLUTIONARY OPTIMIZATION OF THE NEURAL VISION SYSTEM

3.1. Evolution strategies

In evolution strategies (ES) the essential variations during the evolutionary search are mutations which are realized by adding normally distributed random numbers to the objective variables. The variances of the normal distribution are called the strategy parameters of the search process and their values determine the width of the search. The variances have to adapt during the process to the local topology of the search space. This process of *self-adaptation* is a key principle of evolution strategies. It relies on a “second-order” or indirect selection of the strategy parameters which are part of each individual. The strategy parameters are also subject to mutations. Thus, the chromosome of an individual consists of both the objective and the step-size vector, see e.g. Schwefel [10]. In the evolutionary optimization applied here we used a global step-size-adaptation with 2 different step sizes, which turned out to be sufficient for this optimization, one for the 6 nonlinearity parameters and one for the combination filter bank weights, described in more detail in the following sections. We used a discrete recombination for the 6 parameters and also discretely recombined whole combination filters as the smallest parts of a combination filter bank. The strategy parameters, i.e. the global-step-sizes, were recombined by a generalized intermediate recombination [1]. The “ES-typical” deterministic (μ, λ) selection was used in the experiment in Section 4.

3.2. Representation of system nonlinearities

In our vision model, we selected 6 parameters which efficiently characterize the quality of the nonlinearities of the system. These are the WTA selectivities γ_1, γ_2, which control the competition between the different features in the same layer, the threshold parameters θ_1, θ_2, which control the number of neurons firing, and the pooling ranges σ_1, σ_2, which control the sizes of the Gaussian pooling kernels used in layer C1 and C2. The parameters γ_1, γ_2, θ_1, θ_2, σ_1, σ_2 are coded as real values into the chromosome. The values are restricted to the following interval : $\gamma_1, \gamma_2 \in [0, 1]$, with a value of 1 meaning that only the output of the strongest features are transmitted, whereas 0 means, that all signals from all features are transmitted without any reduction in strength. The normalization of the gray values of the images and used filters results in $\theta_1 \in [0, 1]$ and $\theta_2 \in [0, 2]$. For an adequate receptive field size for pooling we set $\sigma_1, \sigma_2 \in [0.5, 5]$.

3.3. Representation of the feature bank

Additionally to the system nonlinearities the weights $\bar{\mathbf{w}}_2^l = (\mathbf{w}_2^{l1}, \ldots, \mathbf{w}_2^{l4})$, which define the combination filter bank, are coded into the chromosome. Here $l = 1, ..., L$, where L is the number of S2 feature planes.

This coding is often referred to as the genotype-phenotype-mapping [11] and plays an important role in evolutionary optimization. One has to guarantee the completeness, i.e. all allowed and sensible phenotypes (i.e., network architectures respective combination filter bank in our case) have to be describable. No forbidden or senseless phenotypes should be describable in order not to unnecessarily enlarge the search space. The so-called strong causality condition should hold, i.e. the neighborhood relationship from the genotype space to the phenotype space should be conserved. The number of free parameters which describe the genotype should be kept as small as possible to keep the dimension of the search space low.

The coding of the combination filter bank is organized as follows: We define the size of one filter of the combination filter bank $\bar{\mathbf{w}}_2^l \in I\!R^{36=4\times3\times3}$. Each of the 4 planes of layer C1 corresponding to 4 different local orientations in the image are convoluted with a 3×3 filter. If we define w_{2i}^{lk}, with $k = 1, 2, 3, 4$, and $i = 1, ..., 9$ as the ith entry of $\mathbf{w}_2^{lk}$ we choose $w_{2i}^{lk}(\theta_i, a_i) = a_i h(\theta_i - k\pi/4 + \pi/2)$, with

$$h(\theta) = \begin{cases} 4\theta/\pi & \pi n \le \theta \le \pi n + \pi/4, \\ -4\theta/\pi + 2 & \pi n + \pi/4 \le \theta \le \pi n + \pi/2, \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

and $n \in I\!N_0$. For an easier understanding the function $h(\theta)$ together with the used 3 shifts are displayed in Fig. 2. The advantages of this coding are the following: Firstly, only sensible combination filter banks are describable, in the sense that entries in the same position of the 3×3 filters corresponding to orthogonal local orientations in the image are not allowed and between neighboring orientations there is a smooth transition. Secondly, through the cyclic coding unnecessary borders are avoided. Thirdly and most importantly, the number of free parameters could be reduced by half. The final optimization was carried out with $L = 9$ filters, which showed the best performance in preliminary optimization runs testing also L =6,7,8,10,14,20. With 9 filters 9×18=162 values have to be optimized. Thus the full optimization took place in a 162+6=168 dimensional search space.

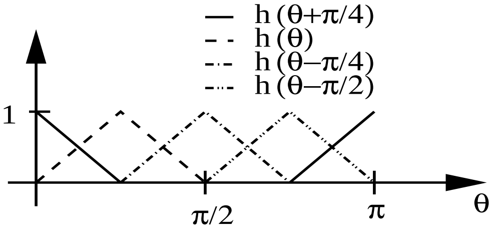

Figure 2: Periodic function $h(\theta)$ displayed in the interval $[0, 2\pi]$ together with 3 shifts. By shifting the function $h(\theta)$ in the definition of the combination filter bank we yield a soft transition from one to the other orientations brought into the system by the convolution with the 4 Gabor filters in the S1 layer.

4. RESULT

For the evolutionary optimization of the combination features and nonlinearity parameters we used the object database COIL20 [5]. This database contains 20 different objects with 72 images of varying angles of view, reaching from 0 to 360 degree in 5 degree steps. We train the vision system with 3 views (0, 120 and 240 degree) of each object. In the test phase the vision system has to recognize 24 remaining views, which are equally distributed between 0 and 360 degree, of all objects by matching them to the corresponding objects. The target of the optimization is the minimization of the misclassification rate.

To determine certain parameters of the optimization process itself we carried out several optimization runs with small population sizes for testing, using a (2,20)-strategy. After finding reasonable values for the initial and the minimal step size of the mutation operator and the number of combination filters L, we started 3 optimization runs with a (10,100)-strategy. The elitist which denotes the best individual was always stored but did not necessarily remain in the population.

In spite of the strict separation of training and test data the danger of over-fitting during the optimization is still there. This is caused by the evolutionary optimization loop in which the test data is used to compute the fitness of a single individual. The question is how strongly generalization is affected by this problem. Therefore, it is important to check the generalization ability of the final result again with a validation dataset. For this test we use the COIL100 [5] data base and the ORL test dataset [4], containing face images.

We first performed an optimization of the nonlinearity parameters alone, using a feature set of 50 combination features, that were obtained according to a local combination enumeration as suggested by [7]. For this feature set, after manual tuning of the nonlinearities Wersing and Körner [13] obtained a misclassification rate of 14.4 % (COIL20), see Fig. 3. After an optimization which at first included only the adjustment of the 6 nonlinearity parameters γ_1, γ_2, θ_1, θ_2, σ_1, σ_2 the misclassification rate could be lowered to a value of 8.7 %. Thereafter, we included the filter-bank in the optimization and the misclassification rate could be further reduced to a value of 5.8 %. Also important is the fact, that after evolutionary optimization of the filters only 9 are needed in contrary to 50 before. This minimizes the hardware resources needed quite considerably. Using a serial computer the recognition time of the neural vision system was halved to about 10 ms on a standard SUN blade 1000 workstation. In a parallel computing environment correspondingly less hardware would be needed. Also the generalization ability of the optimized result is good. The original vision system using enumeration features [13] reached on the COIL100 data base a misclassification rate of 29.0 % whereas the fully evolutionary optimized system reached a value of 26.6 %.

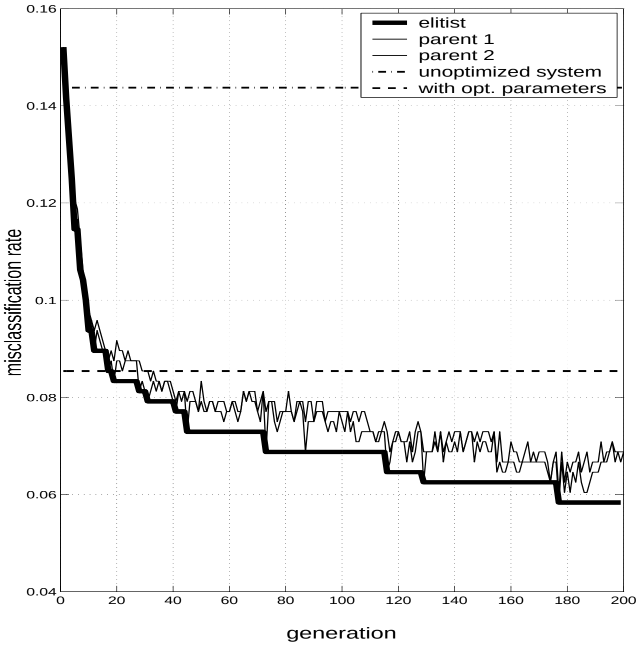

Figure 3: Typical optimization run of the hierarchical neural vision system. The misclassification rates of the best 2 parents are plotted over the simulated generations. Also displayed is the elitist over the generations and the classification rate of the system before any evolutionary optimization and after optimization of the parameters γ_1, γ_2, θ_1, θ_2, σ_1, σ_2 only.

To highlight the performance of the optimized recognition hierarchy we compared our model to recent results published on the COIL100 database using the SNoW method and a linear support vector machine [9]. For a fair comparison we replaced the simple nearest neighbor match operation, see Section 2, by supervised training of a single sigmoidal (tanh) linear discriminant for each object, based on the C2 layer outputs. Here, we performed stochastic gradient descent on the quadratic error, choosing target outputs as 0.9 and -0.9 for correct and incorrect objects, respectively. The results reveal a good generalization of the optimized system (see Fig. 4). Using an identical network feature and parameter setting, without new adaptation, the network achieves similar performance in a face classification task on the ORL face dataset (40 indivduals, 10 images each, courtesy of AT&T Research Labs, Cambridge) as a hybrid convolutional face classification approach [4], which is fully adapted to the task.

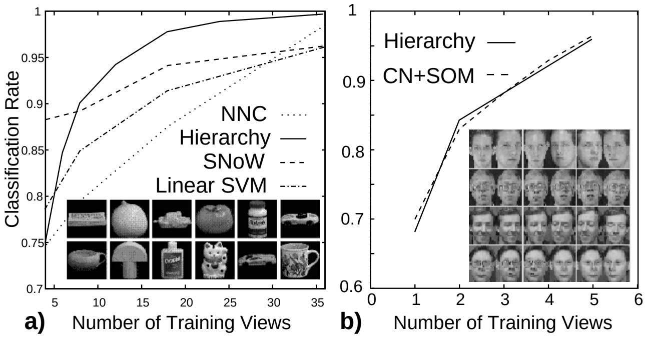

Figure 4: Comparison of classification rates. Part a) compares the classification rates on the COIL100 dataset for our evolutionary optimized hierarchy to results obtained by Roth et. al. using their SNoW model, a linear support vector machine, and direct image nearest neighbor classifier (NNC). In a wide regime of sufficient recognition task difficulty (compare NNC), our feature hierarchy achieves best results with high generalization. Part b) shows results for a face datase, using identical features and nonlinearity parameters. The results match the performance of the hybrid convolutional face classification approach of Lawrence et. al. (CN+SOM).

5. CONCLUSION

In this paper, we demonstrated how evolution strategies help to design critical architectural parts as nonlinearities and combination features, which are mostly built up manually. The system optimized in this way has shown an improved generalization ability and at the same time reduced the needed hardware resources. The results have been compared to state-of-the-art algorithms and exhibited a superior performance in most parts. We also showed that the optimized system is capable of working successfully across domains. The same architecture worked for COIL objects as well as for face images. In the future, we will extend the representation of the neural vision system to increase the degree of freedom for the evolutionary structuring process.

6. REFERENCES

[1] T. Bäck, D. B. Fogel, and Z. Michalewicz, editors. *Evolutionary Computation 1: Basic Algorithms and Operators*. Institute of Physics Publishing, Bristol, 2000.

[2] K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cyb.*, 39:139–202, 1980.

[3] H. Kitano. Designing neural networks using genetic algorithms with graph generation system. *Complex Systems*, 4:461–476, 1990.

[4] S. Lawrence, C. L. Giles, A. C. Tsoi and A. D. Back. Face recognition: A convolutional neural-network. *IEEE Trans. Neur. Netw.*, 8(1):98–113, 1997.

[5] S. Nene, S. Nayar, and H. Murase. Columbia object image library, 1996.

[6] Zhengjun Pan, Theo Sabisch, Rod Adams, and Hamid Bolouri. Staged training of neocognitron by evolutionary algorithms. In Peter J. Angeline, Zbyszek Michalewicz, Marc Schoenauer, Xin Yao, and Ali Zalzala, editors, *Proceedings of the Congress on Evolutionary Computation*, volume 3, pages 1965–1972, Washington D.C., USA, 6-9 July 1999. IEEE Press.

[7] M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

[8] Edmund T. Rolls and Simon M Stringer. On the design of neural networks in the brain by genetic evolution. *Progress in Neurobiology*, 61:557-579, 2000.

[9] D. Roth, M. H. Yang and N. Ahuja. Learning to recognize 3D objects. *Neural Computation*, to appear, 2002.

[10] Hans-Paul Schwefel and Günter Rudolph. Contemporary evolution strategies. In F. Morán, A. Moreno, J. J. Merelo, and P. Chacón, editors, *Proceedings of the Third European Conference on Artificial Life : Advances in Artificial Life*, volume 929 of *LNAI*, pages 893–907, Berlin, June 1995. Springer Verlag.

[11] Berhard Sendhoff, Martin Kreutz, and Werner von Seelen. A condition for the genotype-phenotype mapping: Causality. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA'97)*, pages 73-80, Morgan Kauffmann, San Francisco, 1997.

[12] Daming Shi, Dong Chunlei, and Yeung Daniel S. Neocognitron's parameter tuning by genetic algorithms. *International Journal of Neural Systems*, 9:497–509, 1999.

[13] H. Wersing and E. Körner. Unsupervised learning of combination features for hierarchical recognition models. *Int. Conf. Artif. Neur. Netw. ICANN*, 2002. accepted.

[14] Xin Yao. Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447, 1999.