A concurrent real-time biologically-inspired visual object recognition system

Andreas Holzbach[1] and Gordon Cheng[2]

Abstract— **In this paper, we present an biologically-motivated object recognition system for robots and vision tasks in general. Our approach is based on a hierarchical model of the visual cortex for feature extraction and rapid scene categorization. We modify this static model to be usable in time-crucial real-world scenarios by applying methods for optimization from signal detection theory, information theory, signal processing and linear algebra. Our system is more robust to clutter and supports object localization by approaching the binding problem in contrast to previous models. We show that our model outperforms the preceding model and that by our modifications we created a robust and fast system which integrates the capabilities of biological-inspired object recognition in a technical application.**

I. Introduction

Object recognition in technical systems is still confined to specific scenarios and very limited in performance outside their intended scope. In order to solve the problem of object recognition, it makes sense to follow the biological example for two reasons. First we don't have any other examples of an universal working vision system and second biological systems exceed the capabilities of any existing technical system by far. Humans are capable of detecting and recognizing objects under the most complex circumstances. They can easily identify objects under most lightning conditions, orientation, color or size. Even objects in clutter pose little problems, in contrast to state-of-the-art computer-based object recognition systems, which struggle to perform adequately under varying situations. Therefore, it only makes sense – and maybe is the only successful way – to analyse how the visual system in biological systems works and use that knowledge for modelling those mechanisms to build a more likely effective and robust object recognition system.

Only recently researchers began to look into possible architectures which process information similar to its biological prototype [1], [2], [3]. These models cover a sub-functionality of the vision processing performed by the brain; like visual attention, object recognition, tracking or learning. Especially in the area of object recognition, models have been built as a proof-of-concept with little effort in situating them in the real-world, mainly because they aim on biologically accurateness and the plausible modelling of neural processing. So naturally these models are slow, inefficient and hardly applicable in robotics. So far little effort has been put into modifying and enhancing those models to be usable in time-crucial applications in uncertain environments. With our work we contribute to solve this issue.

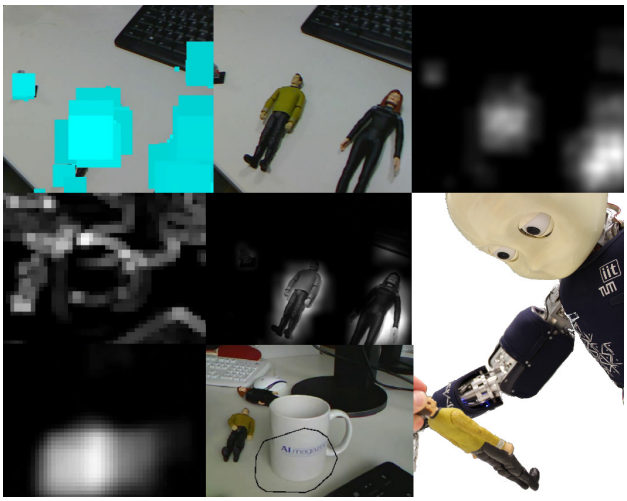


Fig. 1: Responses for entropy (top left and right), gabor filter (middle left) and object localization (bottom left and middle).

II. Related Work

In the last couple of years there has been an increase in biologically-inspired hierarchical models for object recognition, due to a deeper understanding of information processing in the brain [4], [5], [2]. Some of these models have also been applied to enhance common techniques like face recognition by using biologically-inspired features [6]. Some research draw more attention to active-vision systems, which have been used to solve different vision problems like: object recognition [7], [8], [9], [10], [11]; visual search [12], [13]; visual attention [14]; or visual tracking [15]. It has also been investigated how to integrate object recognition [16], [17] and visual attention also with a focus on the aspect of computational complexity [18]. Especially the HMAX model [19] has been investigated and modified in multiple publications [20], [21], [22], [23].

In this paper we specifically focus on the optimization of biologically-inspired object recognition for technical applications to encourage further investigations in this promising research field.

III. HMAX

The object recognition module presented in this paper is built on Serre *et al.*'s HMAX [24], which presents a feed-

*This work was supported (in part) by the DFG cluster of excellence Cognition for Technical systems CoTeSys of Germany, and also (in part) BMBF through the Bernstein Center for Computational Neuroscience Munich (BCCN-Munich).

Andreas Holzbach[1] and Gordon Cheng[2] are with the Institute for Cognitive Systems, Technische Universität München, Karlstr. 45/II, 80333 München, Germany. Email available at www.ics.ei.tum.de.

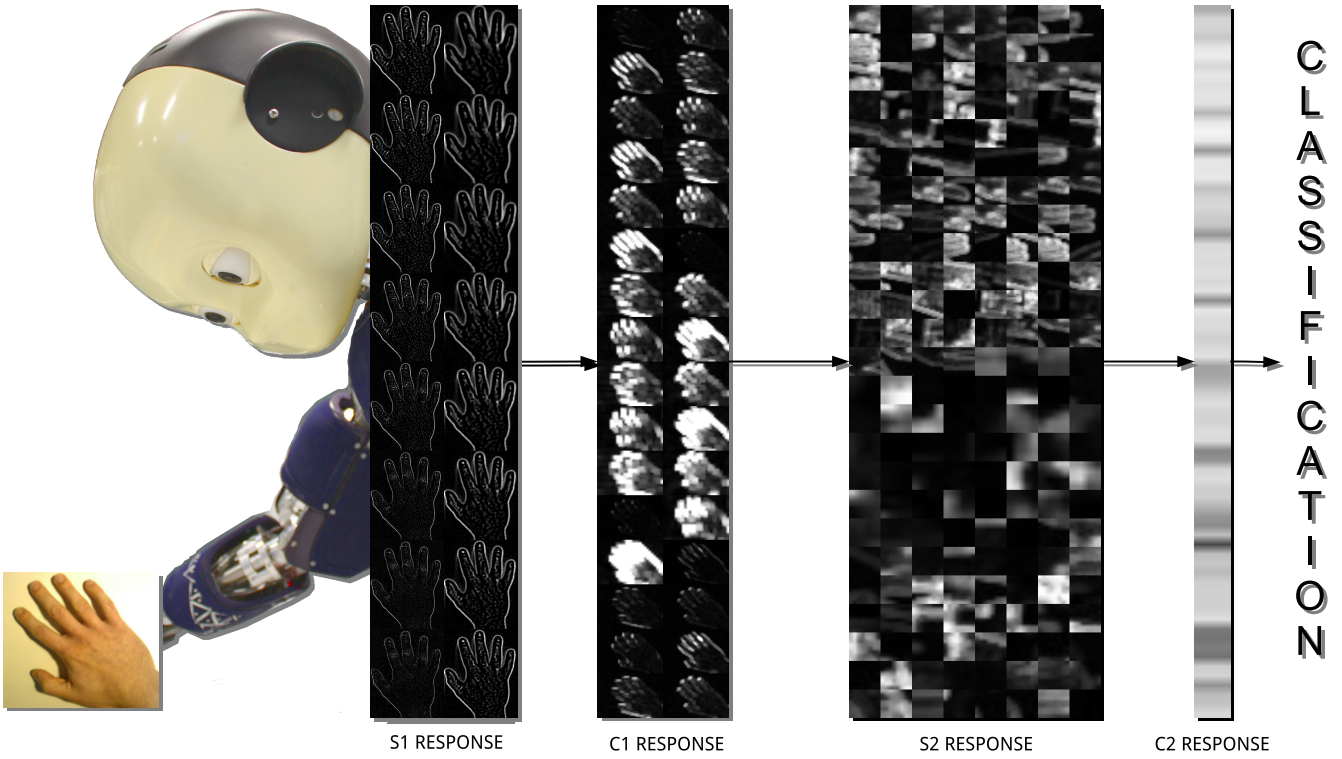


Fig. 2: Functional Overview of the architecture.

forward model of the visual cortex described by Riesenhuber and Poggio [19]. An overview is given in Figure 2. Each layer in the classical model consists of four alternating layers of simple cells (S1, S2) and complex cells (C1, C2) [25].

S1 Layer: The first layer is based on a representation of simple cells which react to oriented edges and bars in the receptive field. The response of these cells are quite similar to Gabor filters. The Gabor filters are created using the function

$$G_{\lambda,\theta,\psi,\sigma,\gamma}\left(x',y'\right) = exp\left(-\frac{x'^2+y'^2\gamma^2}{2\sigma^2}\right)\cos\left(2\pi\frac{x'}{\lambda}+\psi\right) \tag{1}$$

with

$$x' = x\cos\theta + y\sin\theta \tag{2}$$

and

$$y' = -x\sin\theta + y\cos\theta \tag{3}$$

where θ controls the orientation of the filter, ψ the phase offset, σ the variance of the Gaussian, γ the spatial aspect ratio and λ represents the wavelength of the sine function. The edge-sensitive cells contribute to the rotation invariance of the recognition system by the sensitivity to edges and bars of different orientations.

C1 Layer: Complex cells have a larger receptive field than simple cells and add some degree of spatial invariance and shift tolerance to the system. S1 cells of same scale band, same orientation and adjacent filter size are connected to a complex cell. The functionality can be described as a kind of max pooling operation; The maximum value of two adjacent filters of different sizes is calculated by using a sliding window approach.

S2 Layer: In the third layer small patches are chosen from random positions in the receptive field of C1. Each patch set consists of 4 patches, assembled by taking each patch in the set from a C1 response of different orientation $(0°, 45°, 90°, 135°)$ but same position and same scale band. Serre et al. use different sizes of patch sets: patch sets which contain patches of size 4; patch sets with patches of size 8; with size 12 and with size 16. These patch sets are then used for two different cases

Before the training or classification case, a dictionary of patch sets needs to be built. In the standard HMAX system these patch sets are chosen randomly over multiple images.

The S2 cell response is similar to a Gaussian radial basis function and can be calculated as follows

$$r_{i,k} = \exp(-\beta||X_i - P_k||^2) \tag{4}$$

where β is the sharpness of the tuning. X_i is one of the patch sets created in the S2 layer and P_k is one of the "memorized" patch set in the earlier created dictionary. The radial basis function is calculated for all patches i in the set of patch sets of S2 and for all patch sets k in the dictionary.

C2 Layer: Like in C1, the complex cells in the C2 layer now again perform a max operation over all the responses. For each element in the dictionary the maximum response for equation 4 is calculated using all the RBF responses of the patch sets of equal size. Using equation 4 this leads to

$$f_k = \max(\exp(-\beta||X_i - P_k||^2)); \forall i \quad (5)$$

which builds the feature vector $F = \{f_0, f_1, \ldots, f_d\}$ for all k in the dictionary, with d being the length of the dictionary. The feature vector can now be further used for training a classifier. For comparison reasons we used a SVM classifier as Serre *et al.* with a radial basis function kernel [24].

IV. Improvements

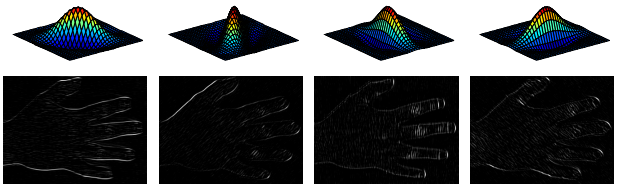
We enhanced the standard HMAX model to be applicable in real-world scenarios in terms of speed, object recognition performance and object localization (see figure 1).

A. Gabor Filter

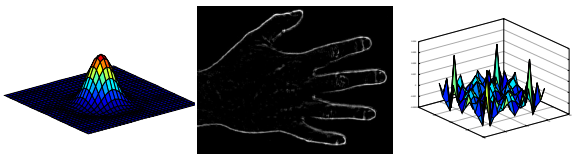
Gabor filters have been shown to provide a good estimate for the response of cortical simple cells and so they are used in all of the HMAX-like implementations. The model presented in [20] uses four different orientations with different sizes and parameters resulting in 64 different filters. Mutch and Lowe [22] use a slightly different approach by applying 12 different orientations but with a sparse representation to a pyramid-based model. The different orientations are supposed to contribute to the system's orientation invariance. However, those models create n-dimensional patches - with n being the number of different orientations - at stage S2 by sampling over random positions. These patches are used for creating a feature vector for classification by applying a radial basis function, which calculates the norm of the difference of the n-dimensional patches. Consequently the result of the RBF function is quite different if the patches are rotated, which indicates, that orientation invariance is in fact very limited. Therefore we argue, that Gabor filter of different orientations can be combined by creating an orientation-free Gabor filter:

$$G_{\lambda,\psi,\sigma,\gamma}(x,y) = exp\left(-\frac{x^2+y^2\gamma^2}{2\sigma^2}\right)\cos\left(2\pi\frac{\sqrt{x^2+y^2}}{\lambda}+\psi\right) \quad (6)$$

This approach creates a much finer representation of edges than ordinary Gabor filters, as all possible orientations are covered (see figure 3). In addition it reduces the computational cost of convolution from n dimensions to one - in our case from 64 to 12. Another benefit of a orientation-free Gabor filter is that it is separable, which would make it computationally more effective. But in the HMAX model the filter is only defined within a circular area as it is more accurate to a simple cells' anatomy, which makes it non-separable. We tested non-circular Gabor filters against circular ones and got better defined edges using the original approach. Using singular value decomposition (SVD) we are still able to factorize a circular Gabor filter into separable matrices. The SVD of the Gabor fitler matrix takes the following form:



(a) 0° (b) 45° (c) 90° (d) 135°



(e) Orientation-free Gabor filter (f) Error matrix SVD

Fig. 3: Modification of applied Gabor filters

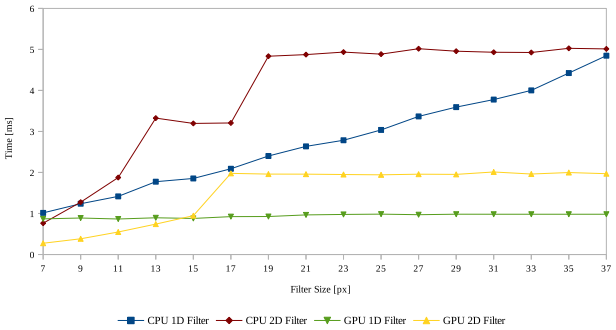


Fig. 4: Speed comparison between Image Filtering with non-separable and separable kernel using CPU and GPU for different kernel sizes.

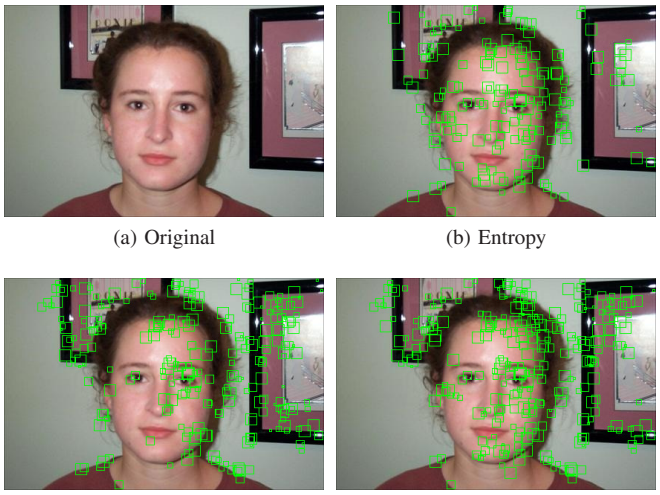
$$G = USV^T = \sum_{i=1}^{j} u_i s_i v_i^T \quad (7)$$

We can precalculate the separable filters and create the convolved image J from image I by using

$$J = \sum_{i=1}^{j} I * (u_i\sqrt{s_i}) + I * (v_i^T\sqrt{s_i}) \quad (8)$$

We achieve almost similar results for $j \geq 3$ compared to the original filter with an average error rate of $9.5 * 10^{-5}$ over the whole filter (see figure 3f) - and still are faster by applying the separable filtering for $j = 3$ than using the non-separable filter.

We compared the computation speed for convolution with different filter sizes on CPU and GPU for the separable filter and the non-separable filter in figure 4 for a image size of 320×240. Using our separable filter approach we achieve a constant processing time on GPU of under 1 ms for $j = 3$ on all kernel sizes. The average computation time of the S1 layer using our approach with 16 orientation-free Gabor filters takes under 16 ms on GPU compared to about 256 ms for 64 filters on CPU with the standard system (see table I). This is a speed up of about 16.



(a) Original (b) Entropy

(c) Std Dev (d) Min Max

Fig. 5: Approximations for Entropy Calculation

B. Entropy

In [26] and [27] we enhanced the HMAX model by adding an information theoretic aspect of neural processing - the maximization of information along the pathway. Our system incorporates the information entropy in the S2 layer of the system. It is sensible in regard to the information a single patch carries and adaptively rejects patches which don't account for the overall information gain. We calculate the entropy of each patch by applying:

$$H(X) = -\sum_{m=1}^{M} p_m \log p_m \tag{9}$$

with p_m being the relative frequency of brightness value m within the patch. This approach filters out patches that show an almost plain distribution of intensities. In order to further reduce the computation time of the system we tested two additional approaches to approximate the entropy in a patch: 1. The standard deviation of the patch and 2. The difference of the maximum and minimum occurring intensity in the patch T:

$$H(X) \approx max(T) - min(T) \tag{10}$$

The intensity difference and the standard deviation approach were both equally fast but about $1.5\times$ faster than the entropy approach, with similar results (see figure 5). For our system we choose the intensity difference approach, because the threshold parameter is more intuitive than the other approaches.

C. Radial Basis Function

The feature vector is calculated using the radial basis function (see equation 4). This means the relative L_2-norm of the difference of two patches, the exponential function and an exponent has to be calculated. The computation time of this step highly depends on the number of sampled patches and the size of the dictionary. A dictionary size of 2000 and e.g. 500 sampled patches would require 1.000.000 RBF calls. We approximate the RBF function response by applying a simpler L_1-norm using:

$$r_{i,k} \approx 1 - \frac{||X_i - P_k||_{L_1}}{\theta} \tag{11}$$

with θ being the maximum possible value a L_1-norm can have for the specific patch size. Hereby we normalize r from a range from $[0; 1]$ with 1 meaning identical patches. This speeds up the computation by a factor of 2 over the normal approach.

D. Dictionary

In the standard HMAX implementation, the dictionary is created by randomly selecting patches as artificial neurons from a set of responses in C1. This approach bears the risk to select a non-optimal set with over-represented and redundant features. Especially in image data sets, where image categories are presented in clutter for training and testing it is uncertain if the applied algorithm actually classifies the object itself or just the surroundings. The category car in the Caltech101 database is for example such a case: The actual object only takes a fraction of the image, whereas objects like trees or houses take up most of the space. Therefore it is uncertain, if the presented algorithms actually recognize the class car or mainly the background, as the patches are randomly selected over the whole image.

To deal with this problem our method follows an approach, which is based on neural tuning. Cells in the brain selectively represent specific sensory patterns. Applying our orientation-free Gabor filter approach enables us to assign patches to specific object classes due to the higher complexity of the generated image after convolution. Each class is represented by an own sub-dictionary, that is created by keeping only patches which occur to a certain degree in all the training images. Hereby we want to achieve, that the created dictionary represents the actual object instead of it's surroundings. A car tire probably will appear in all images for example, however a tree might not, therefore patches containing the tree will most likely be filtered out.

After the sub-dictionaries are created, we apply an approach derived by lateral inhibition appearing in neural processing. For each patch in a sub-dictionary we calculate the response of each patch of each other sub-dictionary. If a patch exists, which reacts above a certain threshold to patches in all sub-dictionary, then these patches are completely removed. That way the sub-directories are even more confined to their specific class.

Mathematically, we can describe the set of sub-dictionaries as a partition of dictionary D

$$\bigcup_{D_i \in D} D_i = D \tag{12}$$

with

$$D_i = \{x | \forall x \in D_i : \nexists y \in D_j, i \neq j : r(x, y) > \theta\} \tag{13}$$

with θ being a threshold of the response of our approximated radial basis function r of Equation 11. Pseudo-Algorithm 1 displays how a sub-dictionary is created.

Algorithm 1: Create Object Specific Dictionary

```
Data: Sub-Dictionary D_i; Set of training images T; Set
      of patches C; Threshold θ
Create New Set Of patches(T_1, D_i);
forall s > 1 do
    Create New Set Of patches(T_s, C);
    forall k do
        forall p do
            if f(D_{i_p}, C_k) < θ then
                delete(D_{i_p});
                break;
            end
        end
    end
end
```

E. Object Localization

Biologically-inspired computational models have mostly applied a simple sliding window approach to localize specific objects in an image, which makes the system rather inefficient, especially in a fast-changing environment. The patches in the sub-directories are object-specific enough that they allow us to deduce the object location to a certain degree using the patches maximum response occurrences in the image (see figure 6). This approach requires no additional calculation, as the maximum responses are anyway needed to be calculated by the system in order to create the feature vector for the classifier. We create a saliency map by adding the maximum response values for each patch in the sub-dictionary to the location in the saliency map where the patch from the test image was sampled that created this highest response.

V. Results

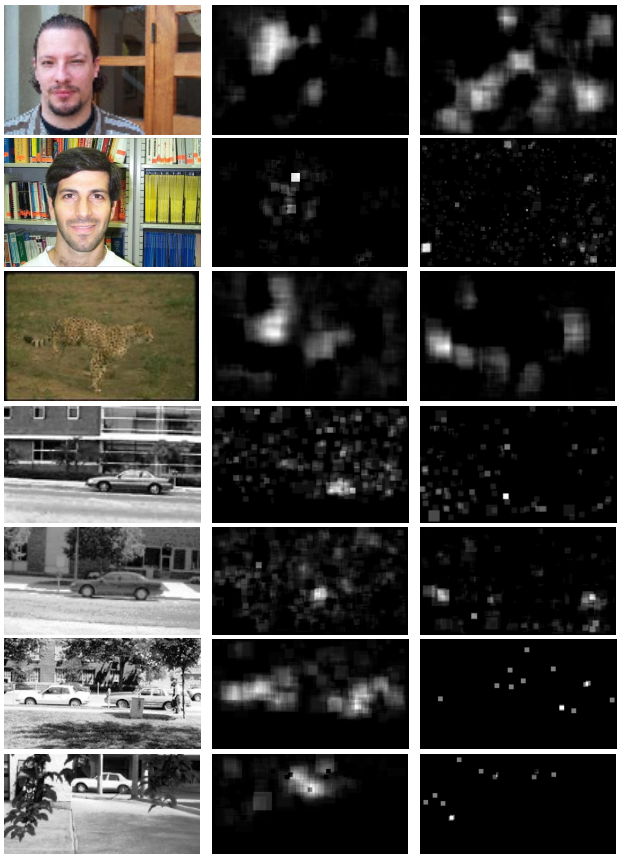
A. Processing Speed

As already shown in figure 4, we were able to speed up the gabor filtering by a factor of 4. Compared to our CPU implementation of the standard HMAX model with non-separable gabor filters, our system speeds up the computation using GPUs and separable orientation-free gabor filters by a factor of $\approx \mathbf{16.8}$ (see table I).

TABLE I: Processing speed of S1 layer in HMAX vs our system (averaged over 100 cycles; CPU: i7, GPU: Geforce 670 GTX).

	HMAX		Our System	
	CPU	GPU	CPU	GPU
Non-separable filter	252 ms	98 ms	63 ms	24 ms
Separable filter	177 ms	60 ms	44 ms	15 ms

In table II we show the computation speed for the next layer C1. Again we compared the speed of the original HMAX system against ours.



(a) Input Image (b) Saliency Map for Object Subdirectory (c) Saliency Map for different Subdirectory

Fig. 6: Object Localization. A saliency map of maximum responses to the object subdirectories. The map which belongs to the object in a) is shown in b); c) shows the response of a different object subdirectory. First three images were taken from the Caltech101 database, the others were taken from the UIUC car dataset.

TABLE II: Processing speed C1

	HMAX		Our System	
	CPU	GPU	CPU	GPU
MAX Operation	140 ms	37 ms	35 ms	9.25 ms

Table III shows processing speed for a dictionary of size 2000 with a sampling rate of 200 patches per patch size per C1 layer. Our system speeds up the overall processing for the S2 Layer by a factor of ≈ 8.6. As our system creates a very efficient representation of an object within the sub-directories, we already achieved good results with a sub-directory size of about 100.

B. Classification Performance

We tested our system against the Caltech-101 database. For each run, we randomly chose a training and testing image set and computed results with different numbers of positive training examples (1, 3, 15, 30 and 40) and 50 negative training examples. Our approach outperforms the original system in regard to the classification accuracy (e.g. for the

TABLE III: Processing speed S2 (For a dictionary size of 2000 and a sample rate of 200 per layer

	HMAX		Our System	
Patch Size	RBF	Approx.RBF	RBF	Approx.RBF
4	1.06s	0.53s	0.26s	0.13s
8	1.59s	0.79s	0.41s	0.19s
12	2.17s	1.09s	0.55s	0.26s
16	2.86s	1.25s	0.71s	0.31s
Sum	7.68s	3.66s	1.92s	0.89s

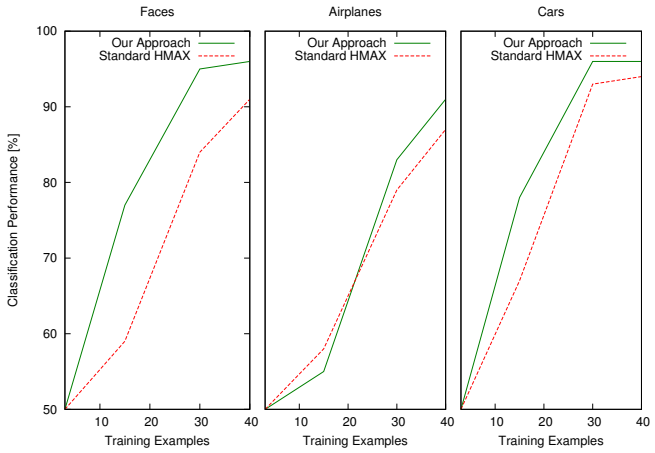


Fig. 7: Comparison of classification results for faces, airplanes and cars of the Caltech image database between the standard HMAX and our approach.

airplanes dataset 92% compared to 86%; faces: 96% to 90%, see figure7; cars 96% to 94%) or is at least of equal result.

VI. Conclusion

In this paper, we have presented a biologically-inspired object recognition system, which applies methods for optimization from signal detection theory, information theory, signal processing and linear algebra. With our modifications we were able to speed up the computation time while outperforming the original classification performance, which creates a system that integrates the potential of biologically-inspired hierarchical models into a technical application. We also enhanced the model to be object location sensitive without performance loss by making use of object subdirectories, which adds a crucial aspect to a vision system.

References

[1] N. Kruger, P. Janssen, S. Kalkan, M. Lappe, A. Leonardis, J. Piater, A. Rodriguez-Sanchez, and L. Wiskott, "Deep hierarchies in the primate visual cortex: What can we learn for computer vision?" *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 8, pp. 1847–1871, 2013.

[2] T. Poggio, "The Computational Magic of the Ventral Stream," *Nature Precedings*, 2012.

[3] Q. V. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. S. Corrado, J. Dean, and A. Y. Ng, "Building high-level features using large scale unsupervised learning," *arXiv preprint arXiv:1112.6209*, 2011.

[4] M. Thomure, W. Landecker, and M. Mitchell, "Random prototypes in hierarchical models of vision," *Learning*, no. 1998, p. 2010, 2010.

[5] T. Serre, G. Kreiman, M. Kouh, C. Cadieu, U. Knoblich, and T. Poggio, "A quantitative theory of immediate visual recognition." *Progress in brain research*, vol. 165, pp. 33–56, Jan. 2007.

[6] E. Meyers and L. Wolf, "Using Biologically Inspired Features for Face Processing," *International Journal of Computer Vision*, vol. 76, no. 1, pp. 93–104, July 2007.

[7] S. Chen, Y. Li, and N. M. Kwok, "Active vision in robotic systems: A survey of recent developments," *The International Journal of Robotics Research*, vol. 30, no. 11, pp. 1343–1377, 2011.

[8] A. Andreopoulos, S. Hasler, H. Wersing, H. Janssen, J. K. Tsotsos, and E. Korner, "Active 3d object localization using a humanoid robot," *Robotics, IEEE Transactions on*, vol. 27, no. 1, pp. 47–64, 2011.

[9] C. Goerick, H. Wersing, I. Mikhailova, and M. Dunn, "Peripersonal space and object recognition for humanoids," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*. IEEE, 2005, pp. 387–392.

[10] H. Wersing and E. Körner, "Learning optimized features for hierarchical models of invariant object recognition." *Neural computation*, vol. 15, no. 7, pp. 1559–88, July 2003. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/12816566

[11] "Learning optimized features for hierarchical models of invariant object recognition." *Neural computation*, vol. 15, no. 7, pp. 1559–88, July 2003.

[12] B. Rasolzadeh, M. Björkman, K. Hübner, and D. Kragic, "An active vision system for detecting, fixating and manipulating objects in the real world," *The International Journal of Robotics Research*, vol. 29, no. 2-3, pp. 133–154, 2010.

[13] T. Halverson and A. J. Hornof, "A computational model of active vision for visual search in human–computer interaction," *Human–Computer Interaction*, vol. 26, no. 4, pp. 285–314, 2012.

[14] C. Siagian and L. Itti, "Rapid biologically-inspired scene classification using features shared with visual attention," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 2, pp. 300–312, 2007.

[15] V. Mahadevan and N. Vasconcelos, "Biologically inspired object tracking using center-surround saliency mechanisms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 3, pp. 541–554, March 2013.

[16] A. Ude, D. Omrčen, and G. Cheng, "Making object learning and recognition an active process," *International Journal of Humanoid Robotics*, vol. 5, no. 02, pp. 267–286, 2008.

[17] A. Ude, C. Gaskett, and G. Cheng, "Support vector machines and gabor kernels for object recognition on a humanoid with active foveated vision," in *Proceedings. IEEE/RSJ*, vol. 1, 2004, pp. 668–673.

[18] A. Ude, V. Wyart, L.-H. Lin, and G. Cheng, "Distributed visual attention on a humanoid robot," in *Humanoid Robots, 2005 5th IEEE-RAS International Conference on*, 2005, pp. 381–386.

[19] M. Riesenhuber and T. Poggio, "Hierarchical models of object recognition in cortex." *Nature neuroscience*, vol. 2, no. 11, Nov. 1999.

[20] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms." *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 411–26, 2007.

[21] P. Moreno and M. J. Mar, "A comparative study of local descriptors for object category recognition : SIFT vs HMAX," *Pattern Recognition*, no. June, pp. 1–8, 2007.

[22] J. Mutch and D. Lowe, "Multiclass Object Recognition with Sparse, Localized Features," *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1 (CVPR'06)*, pp. 11–18, 2006.

[23] C. Theriault, N. Thome, and M. Cord, "HMAX-S : Deep Scale Representation for biologically inspired Image Categorization," *Image (Rochester, N.Y.)*, pp. 3–6, 2011.

[24] T. Serre, L. Wolf, and T. Poggio, "Object recognition with features inspired by visual cortex," in *Computer Vision and Pattern Recognition, CVPR 2005*, vol. 2. Ieee, 2006, pp. 994–1000.

[25] D. Hubell and T. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of Physiology*, vol. 148, no. 3, 1959.

[26] A. Holzbach and G. Cheng, "Enhancing Object Recognition for Humanoid Robots through Time-Awareness." in *Humanoid Robots, 2013, 13th IEEE-RAS International Conference*, October 2013.

[27] ——, "An information theoretic approach to an entropy-adaptive neurobiologically inspired object recognition model," *Frontiers in Computational Neuroscience*, no. 135, 2011.