

Sharing visual features for animal categorization: an empirical study

Manuel J. Marín-Jiménez and Nicolás Pérez de la Blanca

Dpt. Computer Science and Artificial Intelligence, University of Granada
C/ Periodista Daniel Saucedo Aranda s/n,
Granada, 18071, Spain
mjmarin@decsai.ugr.es, nicolas@ugr.es

Abstract. The goal of this paper is to study the set of features that is suitable for describing animals in images, and for being able to categorize them in natural scenes. We propose multi-scale features based on Gaussian derivatives functions, that show interesting invariance properties. In order to build an efficient system, we will use classifiers based on the JointBoosting methodology, which will be compared with the well-known *one-vs-all* approach by using Support Vector Machines. Thirty five categories, containing animals, are selected from the challenging Caltech 101 object categories database to carry out the study.

1 Introduction

The Marr's theory [1] supports that in the early stages of the vision process, there are cells that respond to stimulus of primitive shapes, such as corners, edges, bars, etc. Young [2] models these cells by using Gaussian derivative functions. Riesenhuber & Poggio [3] propose a model for simulating the behavior of the Human Visual System (HVS), at the early stages of vision process. This model is named HMAX. It generates features that exhibit interesting invariance properties (illumination, translation, scale). More recently, Serre *et al.* [4], propose a new model for image categorization adding to the HMAX model a learning step and changing the original Gaussian derivative filter bank by a Gabor filter bank. They argue that the Gabor filter is much more suitable in order to detect local features. Nevertheless, Marin *et al.* [5] have empirically shown that features based on Gaussian derivatives, and generated with that model, behave as well as Gabor based features in the task of object categorization .

Different local feature based approaches are used in the field of object categorization in images. Serre *et al.* [4] use local features based on filter responses to describe objects, achieving a high performance in the problem of object categorization. On the other hand, different approaches using grey-scale image patches, extracted from regions of interest, to represent parts of objects has been suggested, Fei-Fei *et al.* [6], Agarwal *et al.* [7], Leibe [8]. Nevertheless, at the moment, there is not a clear advantage from any of these approaches. However, the non-parametric and simple approach followed by Serre *et al.* [4] in his learning step,

suggests that a lot of discriminative information can be learnt from the output of filter banks.

Unlike faces or cars, which are 'rigid' objects, animals are flexible as they are articulated, what makes more difficult to model their appearance. Hence, the most suitable approach to represent this kind of categories is local features [7],[9]. In this work we deal with those categories.

This paper is organized as follows: in section 2 we propose local features based on Gaussian derivatives filters to describe objects; in section 3 several experiments are carried out in order to evaluate the performance of the proposed features, and different classifiers for multicategorization are compared; and, finally, the paper is concluded in section 4.

2 Proposed scheme

In order to deal with the problem of multicategorization over non-rigid objects, we propose the combination of local features based on Gaussian derivatives, with JointBoosting classifiers [10]. In concrete, we have evaluated this scheme over categories representing animals.

2.1 Supporting Gaussian filters

Koenderink *et al.* [11] propose a methodology to analyze the local geometry of the images, based on the Gaussian function and its derivatives. Several optimization methods are available to perform efficient filtering with those functions [12]. Furthermore, steerable filters [13] [14] (oriented filters whose response can be computed as linear combination of other responses) can be defined in terms of Gaussian functions.

Yokono & Poggio [15] show, empirically, the excellent performance achieved by features created with filters based on Gaussian functions, applied to the problem of object recognition. In other published works, as Varma *et al.* [16], Gaussian filter banks are used to describe textures.

2.2 Local features based on Gaussian derivatives

Due to the existence of a large amount of works based on Gaussian filters, we propose to use filter banks compound by the Gaussian function and its oriented derivatives as local descriptors.

The functions used in this work are defined by the following equations:

a) First order Gaussian derivative:

$$G^1(x, y) = -\frac{y}{2\pi\sigma_x\sigma_y^3} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) \quad (1)$$

b) Second order Gaussian derivative:

$$G^2(x, y) = \frac{y^2 - \sigma_y^2}{2\pi\sigma_x\sigma_y^5} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right) \quad (2)$$

Figure 1 shows samples of anisotropic oriented filters based on first order Gaussian derivatives (top row) and second order Gaussian derivatives (bottom row). Classically, filters based on first order Gaussian derivatives are used to detect *edges*, and the second order ones to detect *bars*.

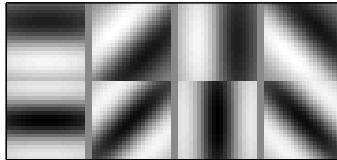


Fig. 1. Anisotropic oriented filters. Top: first order Gaussian derivatives. Bottom: second order Gaussian derivatives.

In order to compute multi-scale robust features, we will employ the biologically inspired HMAX model [3], but using in the first level, our proposed filter banks. To ease the comprehension of this paper, we include a brief description of the steps of the HMAX model to generate C2 features (see [4] for details):

1. Compute S1 maps: the target image is convolved with a bank of oriented filters with various scales.
2. Compute C1 maps: pairs of S1 maps (of different scales) are subsampled and combined, by using the max operator, to generate *bands*.
3. Only during training: extract *patches* P_i of various sizes $n_i \times n_i$ and all orientations from C1 maps, at random positions.
4. Compute S2 maps: for each C1 map, compute the correlation Y with the patches P_i : $Y = \exp(-\gamma \|X - P_i\|^2)$, where X are all the possible windows in C1 with the same size as P_i , γ is a tunable parameter.
5. Compute C2 features: compute the max over all positions and bands for each S2 _{i} map, obtaining a single value C2 _{i} for each patch P_i .

3 Experiments

We have chosen the Caltech 101-object categories ¹ [6] to perform the experiments. This database has become, nearly, the standard database for object categorization. It contains images of objects grouped into 101 categories, plus a background category commonly used as the negative set. This is a very challenging database because the objects are embedded in cluttered backgrounds and have different scales and poses. Since ground-truth is not currently provided for

¹ The Caltech-101 database is available at <http://www.vision.caltech.edu/>

The anisotropic first and second order Gaussian derivatives (with aspect-ratio equals 0.25) are oriented at 0, 45, 90 and 135. All the filter banks contain 16 scales (as [4]). The standard deviation used for the Gaussian-based filter banks is equal to a quarter of the filter-mask size. Table 1 shows the value of the parameters for the filter banks, where FS is the size (in pixels) of the 16 mask-filters and σ is the related standard deviation of the functions.

FS	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37
σ	1.75	2.25	2.75	3.25	3.75	4.25	4.75	5.25	5.75	6.25	6.75	7.25	7.75	8.25	8.75	9.25

Table 1. Filter mask size (FS) and filter width (σ) for Gaussian-based filter banks.

Local features (named C2) will be generated following the HMAX model and using the same empirical tuned parameters proposed by Serre *et al.* in [4]. The evaluation of the filters will be done following a strategy similar to the one used in [6]. From one single category, we draw 30 random samples for training, and 50 different samples for test, or less (the remaining ones) if there are not enough in the set. The training and test negative set are both compound by 50 samples, randomly chosen following the strategy previously explained. For each category and for each filter bank we will repeat 10 times the experiment.



Fig. 3. Sample filters from the Viola-like filter bank. From left to right: horizontal and vertical edge detectors, vertical bar detector and special diagonal detector.

For comparison, we have introduced a Viola-like filter bank, which has been successfully used by Viola&Jones for rapid object detection [17]. These filters can be understood as simplified versions of first and second order Gaussian derivatives. We have chosen four filters: two edge detectors, one bar detector and one special diagonal detector. In figure 3 we can see a sample of the filters.

During the patch² extraction process, we have always taken the patches from a set of prefixed positions in the images. Thereby, the comparison is straightforward for all filter banks. We have decided, empirically, to use 300 patches

² In this context, a *patch* is a piece of a filtered image (at C1 level [4]), extracted from a particular scale. It is three dimensional: for each point of the patch, it contains the responses of all the different filters, for a single scale.

(features) per category and filter bank. If those 300 patches were selected (from a huge pool) for each individual case, the individual performances would be better, but the comparison would be unfair.

In order to avoid a possible dependence between the features and the type of classifier used, we have trained and tested, for each repetition, two different classifiers: AdaBoost (with decision stumps) [18] and Support Vector Machines (linear kernel) [19] [20].

The results obtained for each filter bank, from the classification process, are summarized in table 2. For each filter bank, we have computed the average of all correct classification ratios, achieved for all the 35 categories, and the average of the confidence intervals (of the means). The top row refers to AdaBoost and the bottom row refers to Support Vector Machines. The performance is measured at *equilibrium-point* (when the miss-ratio equals the false positive ratio).

-	<i>Viola</i>	<i>First order</i>	<i>Second order</i>
<i>AdaBoost</i>	(79.6, 4.1)	(80.4, 4.0)	(80.6, 4.4)
<i>SVM</i>	(81.7, 3.1)	(81.8, 3.3)	(83.3, 3.5)

Table 2. Results of classification using three different filter banks: averaged performance and averaged confidence intervals. First row: AdaBoost with decision stumps. Second row: SVM linear. The combination of SVM with features based on second order Gaussian derivatives achieves the best mean performance for the set of animals.

Firstly, we can see that Support Vector Machines outperforms AdaBoost with both filter banks. Furthermore, the features generated with the filter bank based on second order Gaussian derivatives perform, on average, slightly better than the ones based on first order Gaussian derivatives. Note that the simple Viola-like filter bank offers similar results than the first order derivatives.

3.2 One-vs-all vs JointBoosting

In this section we are interested in comparing two methods to be used with our features in the task of multicategorization (we mean, to decide which is the category of the animal contained in the target image). The methods are *one-vs-all* and JointBoosting.

The *one-vs-all* approach consists of training N binary classifiers (as many as categories) where, for each classifier B_i , the positive set is compound by samples from class C_i and the negative set is compound by samples from all the other categories. When a test sample comes, it is classified by all the N classifiers, and the assigned label is the one belonging to the classifier with the greatest output. We have used Support Vector Machines (with linear kernel) [19] as the binary classifiers.

On the other hand, Torralba *et al.* have proposed a procedure, named JointBoosting [10], to generate boosting-based classifiers oriented to multiclass problems. The main idea is to train, simultaneously, several binary classifiers which share features between them, improving this way the global performance of the classification and reducing the computational cost. In the original formulation, each classifier has the form: $H(v, c) = \sum_{m=1}^M h_m(v, c)$. Where v is a vector of features, c is the class, M is the number of training rounds, and h_m are *weak classifiers*. We will use *decision stumps* as weak classifiers.

For this experiment, the training set is compound by the mixture of 20 random samples drawn from each category, and the test set is compound by the mixture of 20 different samples drawn from each category (or the remaining, if it is less than 20). Each sample is encoded by using 4075 patches, randomly extracted from the full training set. These features are computed by using the oriented second order Gaussian derivative filter bank.

Under this conditions, JointBoosting system achieves 32.8% of correct rate categorization, and *one-vs-all* approach achieves 28.7%. Note that for this set (35 categories), *chance* is below 3%. Regarding computation time, each experiment with JointBoosting has required seven hours, however each experiment with *one-vs-all* has needed five days, on a state-of-the-art desktop PC ³.

Hence, we decide to use JointBoosting for the remaining experiments of this paper.

3.3 Results by sharing features

Having chosen the scheme compound by second order Gaussian derivatives based features and JointBoosting classifiers, in this experiment we intend to study in-depth what this scheme can achieve in the problem of multicategorization on flexible object categories, in concrete, focused on categories of animals. Also, JointBoosting allows to understand how the categories are related by the shared features.

The basic experimental setup for this section is: 20 training samples per category, and 20 test samples per category. We will repeat the experiments 10 times with different randomly built pairs of sets.

Firstly, we will evaluate the performance of the system according to the number of features (patches) used to encode each image. We will begin with 100 features and we will finish with 4000 features.

Table 3 shows the evolution of the mean global performance (multicategorization) versus the number of used features. We can see figure 4.a for a graphical representation. Note that with only 100 features, performance is over 17% (better than chance, 3%).

Figure 4.b shows the confusion matrix (on average) for the 35 categories of animals, where the rows refers to the real category and columns to the assigned category. In figure 4.c we can see the histogram of the individual performances

³ Details: both methods programmed in C, PC with processor at 3 GHz and 1024 MB RAM

N features	100	500	1000	1500	2000	2500	3000	3500	4000
Performance	17.5	25.1	27.1	28.9	30.2	31.2	32	32.2	32.8

Table 3. Evolution of global performance. With only 100 features, performance is over 17% (note that chance is about 3%)

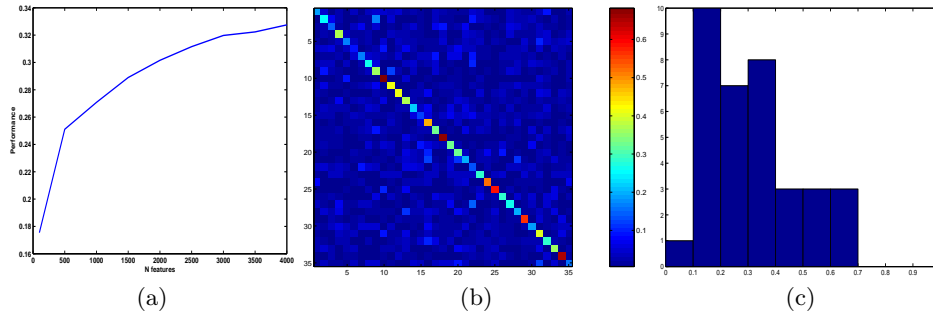


Fig. 4. Multicategorization results over the 35 categories of animals. (a) Performance (on average) vs number of patches. (b) Confusion matrix (on average). From top to bottom and left to right, categories are alphabetically sorted. (c) Histogram (on average) of individual performances.

achieved for the 35 object categories, in the multiclass task. Note, that more than 17 categories are over 30% correct classification ratio. If we study the results for each category, we notice that the hardest category is *cougar* (8.8%) and the easiest category is *dalmatian* (68.8%).

We know that the animals involved in the experiments have parts in common, and since we can know which features are shared by which categories, now we will focus on the relations established by the classifiers.

The first and second features selected by JointBoosting are used for describing the categories *tick* and *hawkbill*, respectively. Other shared features, or relations, are:

- *panda, stegosaurus, dalmatian.*
- *dalmatian, elephant, cougar body.*
- *dolphin, crocodile, bass.*
- *dalmatian, elephant, panda.*
- *kangaroo, panda, dalmatian, pigeon, tick, butterfly.*
- *dalmatian, stegosaurus, ant, octopus, butterfly, dragonfly, panda, dolphin.*
- *panda, okapi, ibis, rooster, bass, hawkbill, scorpion, dalmatian.*

For example, we notice that *panda* and *dalmatian* share several features. Also, it seems that *dolphin*, *crocodile* and *bass* have something in common.

In figure 5 we can see the six patches selected by JointBoosting in the first rounds of an experiment. There are patches of diverse sizes: 4x4, 8x8 and 12x12, all of them represented with their four orientations.

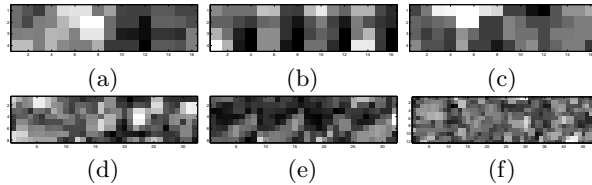


Fig. 5. Sample patches selected by JointBoosting, with their sizes: (a)(b)(c) 4x4x4, (d)(e) 8x8x4, (f) 12x12x4. For representational purposes, the four components (orientations) of the patches are joint. Lighter cells represent higher responses.

4 Summary and conclusions

A scheme of multi-scale local features based on filter responses, combined with classifiers that share features (Joint-Boosting), has been proposed and evaluated for the task of animal categorization. Unlike cars, faces, bottles, etc., which are 'rigid' objects, animals are flexible as they are articulated. For example, there are many different profile views of a cat, depending on how the tail or the paws are. Therefore, learning these classes of objects is harder than the others whose different poses are invariants.

For our experiments, 35 categories of animals have been selected from the challenging Caltech 101-object categories. Firstly, local features based on first and second order Gaussian derivatives has been evaluated over this set, with AdaBoost and Support Vector Machines classifiers, in order to select the best filter bank to generate multi-scale features. As a result, the features based on second order derivatives have been selected. Afterwards, the classic *one-vs-all* approach for multicategorization has been compared with the recent JointBoosting procedure, using the previously selected features. In this experiment, JointBoosting has shown a significant better performance in the task. And, finally, the scheme consisting of features based on second order Gaussian derivatives and the Joint-Boosting classifiers, has been studied in-depth over the set of animals, achieving promising results in the multicategorization of these flexible objects.

Currently, none spatial relation between the local features is used for the process of categorization, but we think that a soft model of relative positions could improve the results, decreasing the confusion between some categories.

5 Acknowledgments

This work was supported by the Spanish Ministry of Education and Science (grant FPU AP2003-2405) and project TIN2005-01665.

References

1. David Marr. *Vision*. W. H. Freeman and Co., 1982.
2. Richard A. Young. The gaussian derivative model for spatial vision: I. Retinal mechanisms. *Spatial Vision*, 2(4):273–293, 1987.
3. M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.
4. T. Serre, L. Wolf, and T. Poggio. Object recognition with features inspired by visual cortex. In *IEEE CSC on CVPR*, June 2005.
5. M.J. Marín-Jiménez and N. Pérez de la Blanca. Empirical study of multi-scale filter banks for object categorization. In *IEEE ICPR*, August 2006.
6. F.F. Li, R. Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop of Generative Model Based Vision (WGMBV)*, 2004.
7. Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE PAMI*, 26(11):1475–1490, Nov. 2004.
8. Bastian Leibe. *Interleaved Object Categorization and Segmentation*. PhD thesis, ETH Zurich, October 2004.
9. Shimon Ullman, Michel Vidal-Naquet, and Erez Sali. Visual features of intermediate complexity and their use in classification. *Nature neuroscience*, 5(7):682–687, July 2002.
10. Antonio B. Torralba, Kevin P. Murphy, and William T. Freeman. Sharing features: Efficient boosting procedures for multiclass object detection. In *CVPR (2)*, pages 762–769, 2004.
11. J.J. Koenderink and A.J. van Doorn. Representation of local geometry in the visual system. *Biological Cybernetics*, 55:367–375, 1987.
12. L. van Vliet, I. Young, and P. Verbeek. Recursive gaussian derivative filters. In *14th Int'l Conf. on Pattern Recognition (ICPR-98)*, volume 1, pages 509–514. IEEE Computer Society Press, August 1998.
13. W.T. Freeman and E.H. Adelson. Steerable filters for early vision, image analysis and wavelet decomposition. In IEEE Computer Society Press, editor, *3rd Int. Conf. on Computer Vision*, pages 406–415, Dec 1990.
14. P. Perona. Deformable kernels for early vision. *IEEE PAMI*, 17(5):488–499, May 1995.
15. J. J. Yokono and T. Poggio. Oriented filters for object recognition: an empirical study. In *Proc. of the Sixth IEEE FGR*, May 2004.
16. M. Varma and A. Zisserman. Unifying statistical texture classification frameworks. *Image and Vision Computing*, 22(14):1175–1183, 2005.
17. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE CVPR*, volume 1, pages 511–518, 2001.
18. J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting. Technical report, Dept. of Statistics. Stanford University, 1998.
19. E. Osuna, R. Freund, and F. Girosi. Support Vector Machines: training and applications. Technical Report AI-Memo 1602, MIT, March 1997.
20. C. Chang and C. Lin. LIBSVM: a library for support vector machines, April 2005.