

# Object Categorization via Local Kernels

Barbara Caputo  
NADA/CVAP, KTH  
Stockholm, Sweden  
caputo@nada.kth.se

Christian Wallraven  
MPI for Biological Cybernetics  
Tübingen, Germany  
wallraven@tuebingen.mpg.de

Maria-Elena Nilsback  
NADA/CVAP, KTH  
Stockholm, Sweden  
mel@nada.kth.se

## Abstract

*This paper considers the problem of multi-object categorization. We present an algorithm that combines support vector machines with local features via a new class of Mercer kernels. This class of kernels allows us to perform scalar products on feature vectors consisting of local descriptors, computed around interest points (like corners); these feature vectors are generally of different lengths for different images. The resulting framework is able to recognize multi-object categories in different settings, from lab-controlled to real-world scenes. We present several experiments, on different databases, and we benchmark our results with state-of-the-art algorithms for categorization, achieving excellent results.*

## 1 Introduction

Over the last three decades there has been significant progress in the performance of object recognition systems. Today it is possible to perform object identification in different poses [10, 8]; significant improvements have been achieved in identifying objects in the presence of clutter, occlusion and varying lighting conditions [11, 2, 7, 8]. Moreover, approaches for category detection (such as faces, cars, pedestrians and horses) have obtained remarkable results [1, 5, 14]. However, limited progress has been achieved on the more general task of multi-object categorization, on which relatively few efforts have been reported in the literature [14, 15, 6]. We argue that an effective algorithm for multi-object categorization must satisfy two main requirements:

**Robust representation** An effective representation for object categories must be able to extract the key visual information which is common to the objects. At the same time, it should be able to capture the visual variability of objects belonging to the same category. Finally, objects belonging to given categories should be recognized in real-world settings: an effective representation should be able to support all the robustness properties which are desirable for object identification (i.e. robustness to noise, occlusion and so on).  
**Robust classification** An effective classification algorithm

for categorization must face all the challenges described above for representation, and tackle them within an appropriate learning rule. A further challenge is the possible lack of control on the quality of training views, which means that robust classification should be possible even when learning is done on difficult training data containing noise and/or clutter.

The contribution of this paper<sup>1</sup> is an algorithm for multi-category recognition which employs local features for representation, and SVMs for classification. Local features have shown excellent results for robust object identification (see for instance [8, 7] and many others) and category detection [1, 14, 15]. SVMs are state-of-the-art large margin classifiers which have demonstrated remarkable performance in object recognition [9]. We combine these two successful approaches via local kernels [16]. The resulting algorithm satisfies our robustness requirements for representation and classification. We present several experiments on multi-category recognition and category detection. We investigate the performance of our algorithm (1) when training is performed on a limited amount of training data; (2) when training is done on images taken in controlled settings, and test on images taken in real-world scenes, and (3) when training and test is done on cluttered views. Benchmark with a probabilistic approach to categorization [14] compares very favourably to our method. To the best of our knowledge, this is the first discriminative algorithm for category recognition presented in the literature.

The paper is organized as follows: Section 2 reviews local representations and Section 3 SVMs and local kernels. Section 4 describes the experimental setup and the results obtained. The paper concludes with a summary discussion.

## 2 The Method

We present an appearance-based approach to object categorization. Categories are defined by training views of several instances of the category under consideration. For example, for the category car, the training set will consist of image views of different cars, taken under different viewpoints. Categories' appearance is described by lo-

<sup>1</sup>This work has been supported by the EU project "Cognitive Vision Systems" -IST-2000-29375 CogVis.

cal features, which constitute the input of an SVM. During the training stage, the algorithm’s parameters are tuned via model selection. Thus, both representation and classification can be designed so to capture at the best the categories’ appearance, and to achieve high discriminability. In the rest of the Section we describe in details the representation and classifier we used.

**Robust Representation: Local Features** The underlying philosophy in describing an image by local features is that once “interesting points” in the image are detected local descriptors are computed around these points. Such local descriptors should be discriminative in the sense that, if a point is detected again in a new image, the comparison of the descriptors computed around the points will allow them to match correctly. Local features have been shown to be remarkably successful for object identification and category detection in real-world settings [11, 7].

Given a set of images  $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^m$ , the most general local feature vector for the image  $\mathbf{I}_i$  can be described as  $\mathbf{L}_i = \{\mathbf{l}_j(\mathbf{I}_i), \mathbf{p}_j(\mathbf{I}_i)\}_{j=1}^{n_i}$ , computed as follows: (1) An interest point detector (a popular choice is the Harris corner detector, [11, 7]) detects  $n_i$  points. In general, the number of interest points detected for each image  $\mathbf{I}_i$  will differ; (2)  $\mathbf{p}_j(\mathbf{I}_i)$  are the coordinates (in the image plane) of the  $j$ -th point; (3)  $\mathbf{l}_j(\mathbf{I}_i)$  is a feature vector computed locally around the  $j$ -th point (see for instance [11]). When one does not consider interest point coordinates, the local feature vector reduces to  $\mathbf{L}_i = \{\mathbf{l}_j(\mathbf{I}_i)\}_{j=1}^{n_i}$ .

In this paper we used jet features [11]. They are local grey value features computed at interesting points. The local characteristics are based on differential grey value invariants, which ensures invariance under the group of displacements within an image; a multi-scale approach makes them robust to scale changes. We performed detection of interest points using a standard Harris-type corner detector.

**Robust Classification: Local Kernels** Consider the problem of separating the set of training data  $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_m, y_m)$  into two classes, where  $\mathbf{x}_i \in \mathbb{R}^N$  is a feature vector and  $y_i \in \{-1, +1\}$  its class label. If we assume that the two classes can be separated by a hyperplane  $\mathbf{w} \cdot \mathbf{x} + b = 0$ , and that we have no prior knowledge about the data distribution, then the optimal hyperplane is the one which maximizes the margin [13]. The optimal values for  $\mathbf{w}$  and  $b$  can be found by solving a constrained minimization problem, using Lagrange multipliers  $\alpha_i (i = 1, \dots, m)$ . It results in a classification function

$$f(\mathbf{x}) = \text{sgn} \left( \sum_{i=1}^m \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x} + b \right), \quad (1)$$

where  $\alpha_i$  and  $b$  are found by using an SVC learning algorithm [13]. Those  $\mathbf{x}_i$  with nonzero  $\alpha_i$  are the “support vectors”. To obtain a nonlinear classifier, one maps the data

from the input space  $\mathbb{R}^N$  to a high dimensional feature space  $\mathcal{H}$  by  $\mathbf{x} \rightarrow \Phi(\mathbf{x}) \in \mathcal{H}$  (a Mercer kernel [13]), such that the mapped data points of the two classes are linearly separable in the feature space. Assuming there exists a kernel function  $K$  such that  $K(\mathbf{x}, \mathbf{y}) = \Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$ , then a nonlinear SVM can be constructed by replacing the inner product  $\mathbf{x} \cdot \mathbf{y}$  in the linear SVM by the kernel function  $K(\mathbf{x}, \mathbf{y})$

$$f(\mathbf{x}) = \text{sgn} \left( \sum_{i=1}^m \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b \right) \quad (2)$$

This corresponds to constructing an optimal separating hyperplane in the feature space.

Local features can be used as input for an SVM via a new class of Mercer kernels [16]:

$$K_L(\mathbf{L}_h, \mathbf{L}_k) = 1/2[\hat{K}(\mathbf{L}_h, \mathbf{L}_k) + \hat{K}(\mathbf{L}_k, \mathbf{L}_h)],$$

$$\hat{K}(\mathbf{L}_h, \mathbf{L}_k) = \frac{1}{n_h} \sum_{j_h=1}^{n_h} \max_{j_k=1, \dots, n_k} \{K_l(\mathbf{l}_{j_h}(\mathbf{L}_h), \mathbf{l}_{j_k}(\mathbf{L}_k)) \cdot \exp\{-(\mathbf{p}_{j_h}(\mathbf{L}_h) - \mathbf{p}_{j_k}(\mathbf{L}_k))^2 / 2\sigma^2\}.$$

It is possible to show that several matching techniques, used for state-of-the-art local features, are related to this class of kernels [16]. In this paper, we used the following kernel  $K_l$ :

$$K_l(\mathbf{x}_{j_h}, \mathbf{y}_{j_k}) = \frac{(\mathbf{x}_{j_h} - \boldsymbol{\mu}_x) \cdot (\mathbf{y}_{j_k} - \boldsymbol{\mu}_y)}{\|\mathbf{x}_{j_h} - \boldsymbol{\mu}_x\| \cdot \|\mathbf{y}_{j_k} - \boldsymbol{\mu}_y\|}, \quad (3)$$

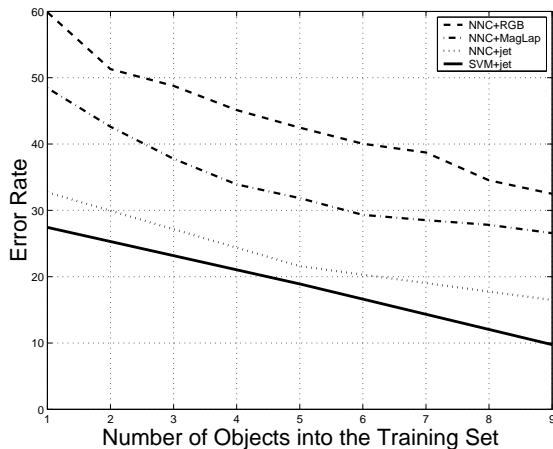
with  $\boldsymbol{\mu}_{x,y}$  mean value. When the local feature representation does not include position information, the exp function is neglected [16].

### 3 Experiments

This section presents experiments showing the effectiveness of our method for object categorization. We performed experiments on different databases, representing categories in controlled lab settings and in real-world scenes. Object views were represented by jet features consisting of 115 feature points, computed over 7 different scales, each resulting in a 9-dimensional vector. All SVM experiments were run using the LibSVM software [3], adding the local kernel in the kernel library. For all the experiments, the kernel parameters were selected via cross-validation on the test set; C was set to 100 [3]. For the multi-category experiments, we implemented a 1-against-the-rest scheme for training and a winner-takes-all strategy for test [13].

#### 3.1 Results: Categorization in Homogeneous Background

In the first series of experiments, we tested the capability of our method to categorize object views taken in a controlled



**Figure 1.** Categorization results for an increasing number of objects in the training set, for different methods. We report the error rates.

lab setting. For this purpose, we used the CogVis-ETH80 database, which contains 80 objects [6] from 8 different categories (apples, tomatos, pears, toy cows, toy horses, toy dogs, toy-cars and cups). For each object, we took 16 views around the equator, from different orientations. Views were represented by jet features with position information. For each category, objects were divided into training and test, with the number of training objects increasing for different experiments. For example, in the first set of experiments we trained on one object for each category, and tested the categorization performance on 9 previously unseen objects. In the second set of experiments we trained on 5 objects for each category, and tested the performance on 5 new objects; and so on. We ran experiments with 1, 5 and 9 objects in the training set, and the remaining objects were placed in the test set. Each experiment was performed on 10 different partitions, and then the results were averaged. We benchmarked our results with a Nearest Neighbor Classifier (NNC) using jet features and the local kernel (3) as the similarity measure. We benchmarked also with a NNC using  $\chi^2$  as similarity measure, and color (RGB,  $16 \times 16 \times 16$ , [12]) and Gaussian derivatives (*MagLap*,  $\sigma_{1,2,3} = 1, 2, 4$ , [10]) histograms as representation. Results are reported in Figure 1.

We see that, for each set of experiments, SVM + jet features achieves the best performance. The second best performance is obtained using NNC + jet features. This result confirms the effectiveness of local features for categorization. It also underlines that the performance of our method is not due only to local features, but also from their combination with SVMs via local kernels. Results obtained with 9 objects in the training set can be compared with those re-



**Figure 2.** Cars and cows in real-world settings.

ported in [6]<sup>2</sup>. The best result reported there, using a single cue, is a recognition rate of 86.40 %, whereas we obtain a recognition rate of 90.25 %. We can conclude that SVM combined with jet features via local kernels are very effective for multi-object categorization in controlled settings.

### 3.2 Results: Categorization in Heterogeneous Background

Of course real-world scenes are far from controllable and the real challenge is to recognize object categories in every day settings. We expect that, due to the use of local representations, our method will be able to perform this task. For this purpose, we performed two experiments. In the first we trained the model on object views taken in a controlled setting, and tested the performance on a collection of pictures of the same category of objects found on the web. This experiment tests also the capability of our model to generalize from toy-objects to real ones. In the second experiment both training and testing was performed on images collected in every day scenes. Note that in this way we test also the capability of our model of learning to recognize categories from cluttered views; this issue has been raised more than once in the literature [14, 15].

**First Experiment:** We considered three categories from the eight used in the previous experiment: cars, cows and cups. Note that cars and cows are toy objects. Training was done on 10 objects from each category, 16 views per object. Testing was performed on 135 views of real cars and 104 views containing cows. The category cup was used as a distractor<sup>3</sup>. Examples of test images are shown in Figure 2. In this experiment we used jet features without position information. We benchmarked with NNC +jet features as described for the previous experiment. Results are reported in Table 1. We see that SVM+jet features gives a better performance, as was to be expected from theory and results reported in Figure 1. It is remarkable to note the very good performance obtained for cars and cows, considering that the training was done on toy objects.

<sup>2</sup>Although in [6] the authors use 41 views per object and define training and test set differently, the statistical significance should be preserved.

<sup>3</sup>We are currently collecting views of cups and other categories like horses and fruits in several real-world settings.

SVM+jet (%)			NNC+jet (%)				
	car	cow	cup		car	cow	cup
car	86.1	11.7	2.2	car	78.1	17.5	4.4
cow	1.0	97.1	1.9	cow	2.9	90.4	6.7

**Table 1.** Recognition results for SVM+ jet features (left) and NNC+ jet features (right). Training was done on views in homogeneous background and in the case of cars and cows, on toy-objects. Testing was performed on views of real objects taken in real-world settings.

**Second Experiment:** In a second set of experiments for category detection we used 3 object categories from the Caltech database [14, 15], namely cars (rear), leaves and faces, taken in real world scenes and at different scales. The training and test sets consisted of 400 images for cars, 93 images for leaves, 218 for training and 217 for testing for faces. In analogy with the experiments reported in [14, 15], we trained our algorithm for category detection against the background class. Training and test sets for background consisted of the same number of views of the category under consideration, as described in [14, 15]. Table 2, left, reports the results obtained with our method. Table 2, right, reports the results obtained with the probabilistic method described in [14]. We see that, once again, results are favorable to our method. The cars (rear) and face databases were used also in [15], but the authors there report their recognition results using ROC curves and ROC equal error rates, making it very difficult to compare our results to theirs, apart from a qualitative manner. Both algorithm seems to perform well on these databases; we plan in the future to run more experiments for a quantitative comparison. We can conclude that SVM combined with jet features, via local kernels, is an effective approach for multi-object categorization.

## 4 Conclusions

We proposed a new method for multi-object categorization. It consists of a SVM combined with local features via a new class of local Mercer kernels. We presented results on different databases, showing that: (1) Our approach is able to perform multi-object categorization with limited amounts of training data; (2) Our approach is able to perform multi-category recognition in real-world scenes, in the case when the training is done on views taken in controlled settings (and possibly on toy objects, like toy cows), and recognition is performed on real object categories; (3) Our approach is able to perform multi-category recognition in real world settings, in the case when the training is done on cluttered views. In all cases, our method achieved very good results. We plan to extend this work in two directions:

SVM+jet			Weber <i>et al</i> , [14]		
cars	faces	leaves	cars	faces	leaves
97.88 %	92.4 %	91 %	84 %	87 %	84 %

**Table 2.** Results for category detection using SVM+jet features (left), and the probabilistic method described in [14] (right).

firstly, we will modify our local kernel so that position information is invariant to affine transformations, and to allow for local or semi-local position constraints; secondly, we will combine our method with virtual SVM [4], so as to obtain robustness to scale and light changes.

## References

- [1] S. Agarwal, D. Roth, "Learning a sparse representation for object detection", *ECCV02*.
- [2] H. Bishof, H. Wildenauer, A. Leonardis, "Illumination insensitive eigenspaces", *ICCV01*.
- [3] C. Chang, C. Lin. LIBSVM: a library for support vector machines. 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [4] D. Decoste, B. Schoelkopf, "Training invariant support vector machines", *Machine Learning*, 46, 161-190, 2002.
- [5] D. Forsyth, M. Fleck, "Body plans", *CVPR97*.
- [6] B. Leibe, B. Schiele, "Analyzing appearance and contour based methods for object categorization", *CVPR03*.
- [7] D. Lowe, "Object recognition from local scale invariant features", *ICCV99*.
- [8] R. Nelson, A. Selinger "A cubist approach to object recognition", *ICCV98*.
- [9] D. Roobaert, M. Zillich, J. O. Eklundh, "A pure learning approach to background invariant object recognition using pedagogical support vector learning", *CVPR01*.
- [10] B. Schiele, J. L. Crowley, "Recognition without correspondence using multidimensional receptive field histograms", *IJCV*, 31-50, 2000.
- [11] C. Schmid, R. Mohr, "Combining greyvalue invariants with local constraints for object recognition", *CVPR96*.
- [12] M. J. Swain, D. H. Ballard, "Color Indexing", *IJCV*, 11-32, 1991.
- [13] V. Vapnik, *Statistical learning theory*, Wiley and son, NY, 1998.
- [14] M. Weber, M. Welling, P. Perona, "Unsupervised learning of object models for recognition", *ECCV00*.
- [15] R. Fergus, P. Perona, A. Zisserman, "Object class recognition by unsupervised scale-invariant learning", *CVPR03*.
- [16] C. Wallraven, B. Caputo, A. Graf, "Recognition with local features: the kernel recipe", *ICCV03*.