

---

# Object categorization with SVM: Kernels for Local Features

---

**Jan Eichhorn**

Max Planck Institute for Biological Cybernetics  
Spemannstrasse 38  
72076 Tübingen, Germany  
jan.eichhorn@tuebingen.mpg.de

**Olivier Chapelle**

Max Planck Institute for Biological Cybernetics  
Spemannstrasse 38  
72076 Tübingen, Germany  
olivier.chapelle@tuebingen.mpg.de

## Abstract

In this paper, we propose to combine an efficient image representation based on local descriptors with a Support Vector Machine classifier in order to perform object categorization. For this purpose, we apply kernels defined on sets of vectors. After testing different combinations of kernel / local descriptors, we have been able to identify a very performant one.

## 1 Introduction

The performance of an object categorization system depends mainly on two ingredients. First, a suitable representation of the image and second a powerful classification algorithm on top of this representation. While the computer vision community has put much effort in working on the representation, there is only recently a growing interest in the algorithms that actually use the representation and the question how to combine them.

Local features [1, 2] have become a very powerful representation of images in categorization and recognition tasks, as could be seen e.g. on last years NIPS-tutorial of David Lowe. SVMs are standard techniques in machine learning and excel by their ability to control the regularization, but they need a suitable kernel in order to work well.

The basic problem here is that the local descriptors (evaluated at some interest points of the image) form an unordered set of vectors and there is no traditional kernel available to compare them.

However, three kernels have recently been proposed to compare such sets [3, 4, 5]. Our aim is to assess the performance of these kernels defined on local image descriptors in an object categorization task.

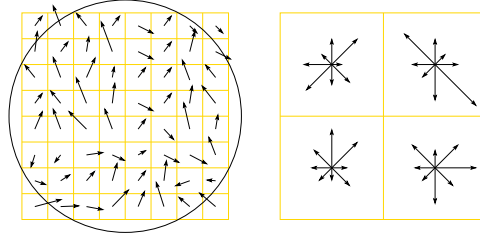


Figure 1: The SIFT keypoint descriptor

The paper is organized as follows: section 2.1 describes the local image descriptors that we will be using in our experiments, section 2.2 introduces the kernels defined on sets and finally section 3 presents detailed experimental results on the ETH80 database [6].

## 2 Local Features and Kernels

### 2.1 Local Features

Local image descriptors have shown very good performance in object recognition tasks [1, 2]. Much of this success is due to their distinctiveness and to the fact, that this type of image representation is robust to occlusion and affine transformations of the objects in the image. We will shortly describe the basic idea of this approach and how we implemented it.

At first an interest point detector (IPD) finds relevant points in the image at a given scale. A Harris corner detector [7] is a common choice here, but there are many other ways to find interest points. In a nutshell the Harris detector finds regions where the covariance matrix of the gradient vectors has two big eigenvalues. This corresponds to regions where the gradients are large in two directions, which is most likely the case for corners and other more complex structures. Consequently it is assumed that regions found by the Harris detector are especially relevant for a description of the object. The number of detected points can be adjusted by a threshold parameter. It has been shown that the performance of local descriptors depends only weakly on the type of IPD [8] and because the Harris detector is robust and easy to implement we applied it in all experiments throughout this study.

The second step of processing is the computation of a feature vector from the image region localized at the interest points. Several techniques have been developed for such a local description (see e.g. [8] for an extensive comparison), many of them involving gradients and higher order derivatives of the image. In the present study we chose two prominent representatives of advanced local descriptors, namely SIFT descriptors introduced by D. Lowe [2] and local JETs first proposed by C. Schmid [1]. Furthermore we implemented a simple approach as a baseline variant that just takes the image pixels of the region around the interest point as feature vector.

**SIFT** Figure 1 illustrates the computation of the SIFT descriptor. First the gradient magnitudes and orientations are sampled from a region around the interest point. The contributions of the points are weighted by a Gaussian centered at the descriptor window (indicated by the circle) in order to give less emphasis to points further away from the center and to make the descriptor robust to small changes in the window position. A schematic view of the descriptor itself is shown on the

right side of Figure 1. It contains the orientation histograms of the four  $4 \times 4$  sample regions with eight entries per histogram. To make the descriptor more robust against small shifts of the window or small rotations, the contribution of each sample point is distributed via trilinear interpolation to adjacent histogram bins. The image shows a  $2 \times 2$  array of histograms whereas in our implementation we used a  $4 \times 4$  array which results in a  $4 \times 4 \times 8 = 128$ -dimensional feature vector. Note that the original implementation of SIFT contains its own IPD. To make the results more comparable, we decided to use the Harris detector instead in order to ensure that all types of local descriptors are computed at the same locations.

**JET** The JET descriptor is computed from derivatives of the image region around the interest point. One can show that a complete set of differential invariants can be constructed that fully describes the image in a local neighborhood of the interest point. Furthermore, this description is invariant under rotation (more precisely under  $SO(2)$ -transformations). By considering derivatives up to third order, the JET descriptor contains nine differential invariants, making it a nine-dimensional feature vector.

**Image patch** As mentioned above, we implement a simple minded version of local descriptors in form of a  $5 \times 5$  pixels images patch centered at the interest point. To avoid effects on the image boundary, we simply neglect all interest points too close to the edge of the image.

In many of the applications of local descriptors, the computation is performed on different scales in order to achieve invariance to changes in object size. Because we do not need this invariance properties on our dataset and to keep computation simple all processing was performed at only one fixed scale.

Hence, by computing local descriptors we transformed the pixel representation of the image into a set of feature vectors. Here we have to point out explicitly that in all of following the spatial information about the interest points (i.e. their coordinates) will be neglected. It is very likely that by considering this information, categorization results could be improved substantially. However the focus of this work is to assay kernel functions that only use the features themselves. The question how to combine these kernels with spatial information is subject of ongoing research.

## 2.2 Kernels on sets

To apply the SVM algorithm to the local descriptor representation a kernel function is necessary that can compare two images by using these descriptor vectors. At a first glance it is not obvious how to achieve this function. In general we get a different number of interest points per image and the local descriptors themselves can not be easily ordered, especially given that we don't use the position of the interest points. Basically we have to deal with an unordered set of descriptors for each image and therefore we need a kernel on sets.

There has been a number of approaches to this problem. One common feature of the methods is that they all have to compute a similarity score between the elements of the sets. The choice of this measure can vary and we will call this function *minor kernel*. We will use three existing kernel functions for sets:

**Matching kernel** This “kernel” was proposed in [5] where it was applied to an object recognition task. Given two sets of local descriptors  $\mathbf{L}_1$  and  $\mathbf{L}_2$ , at

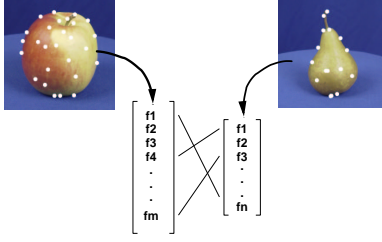


Figure 2: Matching features is not obvious

first a matrix of similarity scores between  $\mathbf{L}_1$  and  $\mathbf{L}_2$  is computed. Common choices for the similarity measure (the *minor kernel*) are the RBF-kernel:

$$k_{rbf}(f_1, f_2) = \exp\left(-\frac{\|f_1 - f_2\|^2}{2\sigma^2}\right)$$

or the so-called normalized cross correlation kernel:

$$k_{normCC}(f_1, f_2) = \exp\left(-\rho\left(1 - \frac{(f_1 - \mu_1) \cdot (f_2 - \mu_2)}{\|f_1 - \mu_1\| \cdot \|f_2 - \mu_2\|}\right)\right).$$

Finally the kernel value is the average over the best-match-scores of the elements in  $\mathbf{L}_1$  and  $\mathbf{L}_2$ :

$$K_{match}(\mathbf{L}_1, \mathbf{L}_2) = \frac{1}{2} \left[ \hat{K}(\mathbf{L}_1, \mathbf{L}_2) + \hat{K}(\mathbf{L}_2, \mathbf{L}_1) \right],$$

with

$$\hat{K}(\mathbf{L}_1, \mathbf{L}_2) = \frac{1}{|\mathbf{L}_1|} \sum_{i=1}^{|\mathbf{L}_1|} \max_{j=1 \dots |\mathbf{L}_2|} k(f_i, f_j).$$

Despite the claim in [5], it turns out that this function is actually not positive definite. For this reason, from a theoretical point of view, it is not safe to use it as a kernel function in an SVM. However, from a practical point of view, it might still achieve good performances.

**Bhattacharyya kernel** This second kernel was introduced in [3]. The Bhattacharyya affinity [9] is defined between two probability distributions  $p$  and  $p'$  as

$$k_{bhattach}(p, p') = \int \sqrt{p(x) \cdot p'(x)} dx.$$

For each set of vectors, it was suggested in [3] to fit a Gaussian distribution to those vectors. Then, the kernel value between two sets of vectors is defined as the Bhattacharyya affinity between the two corresponding Gaussian distributions, which can be computed in a closed form.

Since a Gaussian distribution captures only a limited part of the statistics in the empirical distribution of the vectors, the authors propose to first map those vectors in a feature space via the minor kernel. Then the Bhattacharyya affinity is computed in the feature space. Doing so allows to capture more structure of the empirical distribution.

**Kernel Principal Angles (KPA)** As a third means of comparing two sets  $\mathbf{L}_1$  and  $\mathbf{L}_2$  we use the Kernel Principal Angles measure introduced in [4]. The

principal angles  $\{\theta_i\}_{i=1,\dots,N}$  between two subspaces  $\mathbf{U}_{\mathbf{L}_1}$  and  $\mathbf{U}_{\mathbf{L}_2}$  spanned by the elements of  $\mathbf{L}_1$  and  $\mathbf{L}_2$  are iteratively defined as:

$$\begin{aligned} \cos \theta_k &= \max_{u \in \mathbf{U}_{\mathbf{L}_1}} \max_{v \in \mathbf{U}_{\mathbf{L}_2}} \langle u, v \rangle \\ \text{subject to: } & \langle u, u \rangle = \langle v, v \rangle = 1 \quad \text{and} \quad \langle u, u_i \rangle = \langle v, v_i \rangle = 0 \quad \forall i = 1, \dots, k-1 \end{aligned} \quad (1)$$

In [4] this method is augmented with the kernel trick to compute the principal angles for  $\phi(\mathbf{L}_1)$  and  $\phi(\mathbf{L}_2)$ , the mappings of the sets into some feature space via a minor kernel. Furthermore the authors prove that

$$k_{KPA}(\mathbf{L}_1, \mathbf{L}_2) = \prod_{i=1}^N (\cos \theta_i)^2$$

is a positive definite kernel for sets if the dimensions of  $\mathbf{U}_{\mathbf{L}_1}$  and  $\mathbf{U}_{\mathbf{L}_2}$  are equal on the whole dataset. However, this constraint turns out to be a substantial drawback in our setting as we have in general different numbers of interest points per image.

To overcome this problem we had to pad each descriptor set with random entries to ensure a constant size of the sets. The pool of random entries was fixed in the beginning, such that padding always uses the same entries. This procedure ensures adding identical subspaces to sets that are smaller than the maximum number of descriptors and thereby introducing only zero angles and consequently factors of one to the product  $\prod_{i=1}^N (\cos \theta_i)^2$ .

Even with this workaround, this kernel performed poorly as we will see in the experimental section.

### 3 Experiments

We conducted two series of experiments. First we combined all types of local descriptors with all the kernels and different minor kernels to see which approach is the most promising. In a second step we chose the best feature-kernel-pairing and analyzed the influence of the parameters on the performance of the system.

As dataset for the experiments we used the ETH80-database that was proposed for object categorization [6]. It contains 80 objects from eight different classes (apples, pears, tomatoes, cows, dogs, horses, cups, cars). Images of each object were taken from 41 different viewpoints and with uniform blue background. We worked with two different subsets of the database containing five views of each object. An example for each set is shown in Figure 3 where set ‘‘A’’ contains views from very different positions whereas set ‘‘B’’ contains images from consecutive views around the equator.

All images were converted to gray values for simplicity.

The SVM classifier was used in a one-versus-rest multi-class setting and we report the leave-one-out performance of the classifier. More precisely this means that the test set contains all five images of one object and the training set contains the images of the remaining 79 objects. The performance is the percentage of correct class prediction on the test set and is averaged over all 80 possible combinations of training and test set.

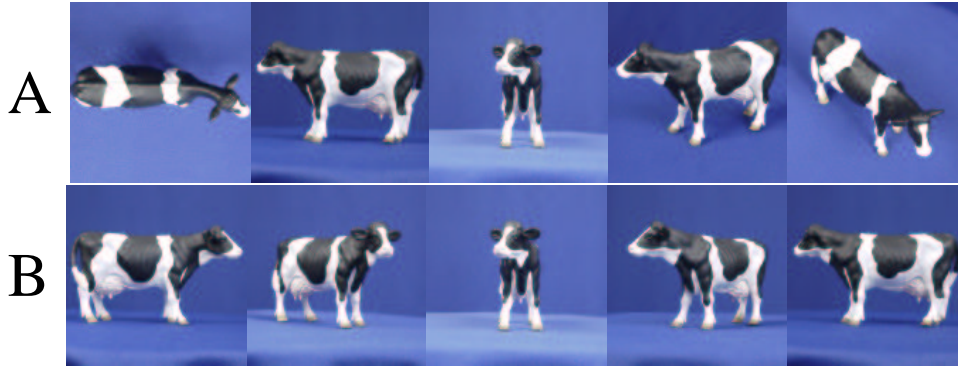


Figure 3: Subset “A” with high and subset “B” with low variability in pose

### 3.1 Finding the winning team

In a set of comprehensive tests we evaluated every combination of features and kernels and present the results in Table 1. The threshold of the Harris-detector was fixed to a value such that it produces in average 40 interest points per image. The regularization parameter of the SVM was set to  $C = 10^4$  and all tests were performed on subset “A” to make the task harder and identify efficient methods more easily. As minor kernels we applied the RBF-kernel and the normalized cross-correlation kernel mentioned above.

The performance in Table 1 is the maximum value that we achieved by varying the hyper-parameters of the kernels.

Table 1: Best performance for each combination of kernel, descriptor type and minor kernel

<i>Kernel</i>	<i>Minor Kernel</i>	<b>SIFT</b>	<b>JET</b>	<b>Image Patch (6x6)</b>
Matching	RBF	72%	43%	46%
	NormCC	74%	58%	43%
Bhattacharyya	RBF	74%	70%	74%
	NormCC	73%	69%	37%
KPA	RBF	27%	24%	23%
	NormCC	23%	23%	25%

As can be seen from Table 1 SIFT descriptors are in average the best of the three image representations. Of the kernel variants the Bhattacharyya kernel with an RBF minor kernel has the best average performance. Moreover, it is consistently the best kernel on the three different kinds of image representation. Therefore we chose this pairing for further investigation in the next section.

By looking at the results in Table 1 it is clear that the Kernel Principal Angles approach as we implemented it is not applicable to our setting. We were not able to achieve satisfactory results with different variations of this kernel.

### 3.2 Bhattacharyya kernel with SIFT descriptors

In the second series of experiments we varied several parameters of the setting to study their influence on the performance. Here images were always represented

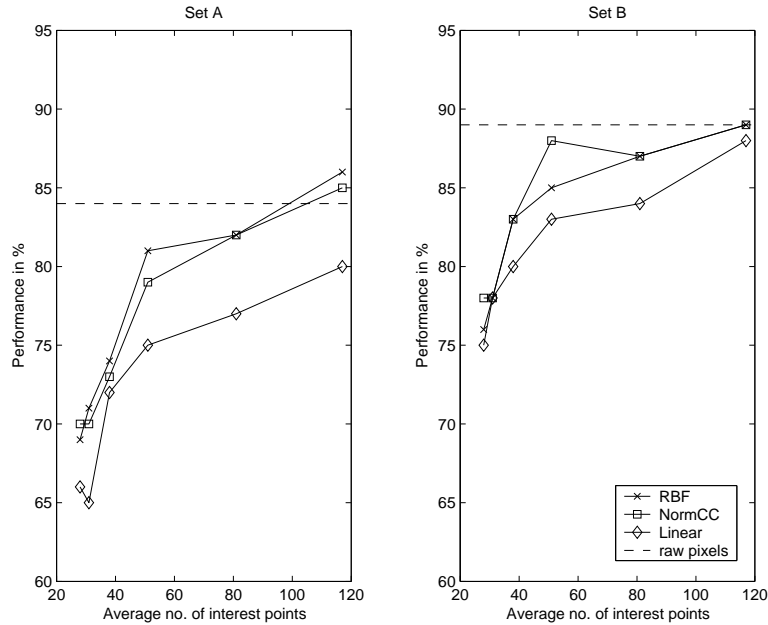


Figure 4: The performance as a function of the number of interest points, the minor kernel and the variability of the dataset. The dashed line indicates the performance of an SVM with RBF kernel on a raw pixel representation

by SIFT descriptors and the Bhattacharyya kernel was applied to this representation. The hyperparameters of the kernel were set to the values where they reached their maximum performance in the previous experiment (see Table 1). Again the SVM parameter  $C$  was fixed to  $10^4$ . Experiments show, that performance is hardly influenced by  $C$  and decreases when  $C$  becomes smaller than 10.

First we changed the number of interest points per image by adjusting the threshold of the Harris corner detector. Results are presented in Figure 4. The performance increases when we take more interest points into consideration. At the same time one can observe that the difference in performance between the usage of a non-linear minor kernel and a linear one increases also with the number of interest points. This effect indicates that non-linear feature mappings are more effective when many data points are available.

Note that the dashed line in figure 4 shows the performance of an SVM with an RBF kernel on the raw pixel representation with a resolution  $32 \times 32$ . Given that the images in this dataset were centered, on uniform background and did not contain occlusions or some kind of deterioration that one would expect in a real image, we believe that this SVM trained on the raw pixels is one of the best possible systems for this task.

For this reason, having been able to achieve comparable results with only local descriptors and without any spatial information appeared as a spectacular result.

## 4 Conclusion

The problem of image representation has always been one of the core issues in the application of machine learning to computer vision. The local image descriptors are a good method to obtain an invariant representation of the image while retaining enough discriminative power. We presented here a very promising solution for incorporating these descriptors in a kernel based learning system.

Further work includes the use of spatial information (i.e. the relative coordinates of the interest points). This kind of additional spatial information was for instance tested in [5, 1] and was shown to improve greatly the performances.

From a computational point of view, evaluating the Bhattacharyya kernel is cubic in the number of interest points and that is reason why we did not use more interest points in figure 4. However, we are in the process of implementing a fast approximation of this kernel based on the incomplete Cholesky decomposition [10].

## References

- [1] C. Schmid and R. Mohr. Local greyvalue invariants for image retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(5):530–534, 1997.
- [2] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [3] R. Kondor and T. Jebara. A kernel between sets of vectors. In *Proceedings of the ICML*, 2003.
- [4] L. Wolf and A. Shashua. Learning over sets using kernel principal angles. *Journal of Machine Learning Research (JMLR)*, pages 913–931, 2003.
- [5] C. Wallraven, B. Caputo, and A.B.A. Graf. Recognition with local features: the kernel recipe. In *ICCV 2003 Proceedings*, volume 2, pages 257–264. IEEE Press, 2003.
- [6] B. Leibe and B. Schiele. Analyzing contour and appearance based methods for object categorization. In *Proceedings of International Conference on Computer Vision and Pattern Recognition (CVPR'03)*, 2003.
- [7] C. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, Manchester, UK, 1988.
- [8] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.
- [9] A. Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math Soc.*, 35:99–110, 1943.
- [10] S. Fine and K. Scheinberg. Efficient svm training using low-rank kernel representations. *Journal of Machine Learning Research*, 2:243–264, 2001.