Local features and kernels for classification of texture and object categories: A comprehensive study

J. Zhang[1], M. Marszałek[1], S. Lazebnik[2] and C. Schmid[1],

[1] INRIA, GRAVIR-CNRS, 655, av. de l'Europe, 38330 Montbonnot, France

[2] Beckman Institute, University of Illinois, 405 N. Mathews Ave., Urbana, IL 61801, USA

Abstract

Recently, methods based on local image features have shown promise for texture and object recognition tasks. This paper presents a large-scale evaluation of an approach that represents images as distributions (signatures or histograms) of features extracted from a sparse set of keypoint locations and learns a Support Vector Machine classifier with kernels based on two effective measures for comparing distributions, the Earth Mover's Distance and the χ^2 distance. We first evaluate the performance of our approach with different keypoint detectors and descriptors, as well as different kernels and classifiers. We then conduct a comparative evaluation with several state-of-the-art recognition methods on four texture and five object databases. On most of these databases, our implementation exceeds the best reported results and achieves comparable performance on the rest. Finally, we investigate the influence of background correlations on recognition performance via extensive tests on the PASCAL database, for which ground-truth object localization information is available. Our experiments demonstrate that image representations based on distributions of local features are surprisingly effective for classification of texture and object images under challenging real-world conditions, including significant intra-class variations and substantial background clutter.

Keywords: image classification, texture recognition, object recognition, scale- and affine-invariant keypoints, support vector machines, kernel methods.

1 Introduction

The recognition of texture and object categories is one of the most challenging problems in computer vision, especially in the presence of intra-class variation, clutter, occlusion, and pose changes. Historically, texture and object recognition have been treated as two separate problems in the literature. It is customary to define texture as a visual pattern characterized by the repetition of a few basic primitives, or *textons* [27]. Accordingly, many effective texture recognition approaches [8, 31, 33, 57, 58] obtain textons by clustering local image features (i.e., appearance descriptors of relatively small neighborhoods), and represent texture images as histograms or distributions of the resulting textons. Note that these approaches are *orderless*, i.e., they retain only the frequencies of the individual features, and discard all information about their spatial layout. On the other hand, the problem of object recognition has typically been approached using *parts-and-shape* models that represent not only the appearance of individual object components, but also the spatial relations between them [1, 17, 18, 19, 60]. However, recent literature also contains several proposals to represent the "visual texture" of images containing objects using orderless *bag-of-features* models. Such models have proven to be effective for object classification [7, 61], unsupervised discovery of categories [16, 51, 55], and video retrieval [56]. The success of orderless models for these object recognition tasks may be explained with the help of an analogy to *bag-of-words* models for text document classification [40, 46]. Whereas for texture recognition, local features play the role of textons, or frequently repeated elements, for object recognition tasks, local features play the role of "visual words" predictive of a certain "topic," or object class. For example, an eye is highly predictive of a face being present in the image. If our visual dictionary contains words that are sufficiently discriminative when taken individually, then it is possible to achieve a high degree of success for *whole-image classification*, i.e., identification of the object class contained in the image without attempting to segment

or localize that object, simply by looking which visual words are present, regardless of their spatial layout. Overall, there is an emerging consensus in recent literature that orderless methods are effective for both texture and object description, and it creates the need for a large-scale quantitative evaluation of a single approach tested on multiple texture and object databases.

To date, state-of-the-art results in both texture [31] and object recognition [18, 23, 48, 61] have been obtained with local features computed at a sparse set of scale- or affine-invariant keypoint locations found by specialized *interest operators* [34, 43]. At the same time, Support Vector Machine (SVM) classifiers [54] have shown their promise for visual classification tasks (see [50] for an early example), and the development of kernels suitable for use with local features has emerged as a fruitful line of research [4, 13, 23, 37, 47, 59]. Most existing evaluations of methods combining kernels and local features have been small-scale and limited to one or two datasets. Moreover, the backgrounds in many of these datasets, such as COIL-100 [44] or ETH-80 [32] are either (mostly) uniform or highly correlated with the foreground objects, so that the performance of the methods on challenging real-world imagery cannot be assessed accurately. This motivates us to build an effective image classification approach combining a bag-of-keypoints representation with a kernel-based learning method and to test the limits of its performance on the most challenging databases available today. Our study consists of three components:

Evaluation of implementation choices. In this paper, we place a particular emphasis on producing a carefully engineered recognition system, where every component has been chosen to maximize performance. To this end, we conduct a comprehensive assessment of many available choices for our method, including keypoint detector type, level of geometric invariance, feature descriptor, and classifier kernel. Several practical insights emerge from this process. For example, we show that a combination of multiple detectors and descriptors usually achieves better results than even the most discriminative individual detector/descriptor channel. Also, for most datasets in our evaluation, we show that local features with the highest possible level of invariance do not yield the best performance. Thus, in attempting to design the most effective recognition system for a practical application, one should seek to incorporate multiple types of complementary features, but make sure that their local invariance properties do not exceed the level absolutely required for a given application.

Comparison with existing methods. We conduct a comparative evaluation with several state-of-the-art methods for texture and object classification on four texture and five object databases. For texture classification, our approach outperforms existing methods on Brodatz [3], KTH-TIPS [24] and UIUCTex [31] datasets, and obtains comparable results on the CUReT dataset [9]. For object category classification, our approach outperforms existing methods on the Xerox7 [61], Graz [48], CalTech6 [18], CalTech101 [15] and the more difficult test set of the PASCAL challenge [14]. It obtains comparable results on the easier PASCAL test set. The power of orderless bag-of-keypoints representations is not particularly surprising in the case of texture images, which lack clutter and have uniform statistical properties. However, it is not *a priori* obvious that such representations are sufficient for object category classification, since they ignore spatial relations and do not separate foreground from background features.

Influence of background features. As stated above, our bag-of-keypoints method uses both foreground and background features to make a classification decision about the image as a whole. For many existing object datasets, background features are not completely uncorrelated from the foreground, and may thus provide inadvertent "hints" for recognition (e.g., cars are frequently pictured on a road or in a parking lot, while faces tend to appear against indoor backgrounds). Therefore, to obtain a complete understanding of how bags-of-keypoints methods work, it is important to analyze the separate contributions of foreground and background features. To our knowledge, such an analysis has not been undertaken to date. In this paper, we study the influence of background features on the diverse and challenging PASCAL benchmark. Our experiments reveal that, while backgrounds do in fact contain some discriminative information for the foreground category, particularly in "easier" datasets, using foreground and background features together *does not* improve the performance of our method. Thus, even in the presence of background correlations, it is the features on the objects themselves that play the key role for recognition. But at the same time, we show the danger of training the recognition system on datasets with monotonous or highly correlated backgrounds—such a system does not perform well on a more complex test set.

For object recognition, we have deliberately limited our evaluations to the image-level classification task, i.e., classifying an entire test image as containing an instance of one of a fixed number of given object classes. This task must be clearly distinguished from *localization*, or reporting a location hypothesis for the object that is judged to be present. Though it is possible to perform localization with a bag-of-keypoints representation, e.g., by incorporating a probabilistic model that can report the likelihood of an individual feature for a given image and category [55], evaluation of localization accuracy is beyond the scope of the present paper. It is important to emphasize that we do not propose basic bag-of-keypoints methods as a solution to the general object recognition problem. Instead, we seek to demonstrate that, given the right implementation choices, simple orderless image representations with suitable kernels can be surprisingly effective on a wide variety of imagery. Thus, they can serve as good baselines for measuring the difficulty of newly acquired datasets and for evaluating more sophisticated recognition approaches that incorporate structural information about the object.

The rest of this paper is organized as follows. Section 2 presents existing approaches for texture and object recognition. The components of our approach are described in section 3. Results are given in section 4. We first evaluate the implementation choices relevant to our approach, i.e., we compare different detectors and descriptors as well as different kernels. We then compare our approach to existing texture and object category classification methods. In section 5 we evaluate the effect of changes to the object background. Section 6 concludes the paper with a summary of our findings and a discussion of future work.

2 Related work

This section gives a brief survey of recent work on texture and object recognition. As stated in the introduction, these two problems have typically been considered separately in the computer vision literature, though in the last few years, we have seen a convergence in the types of methods used to attack them, as orderless bags of features have proven to be effective for both texture and object description.

2.1 Texture recognition

A major challenge in the field of texture analysis and recognition is achieving invariance under a wide range of geometric and photometric transformations. Early research in this domain has concentrated on global 2D image transformations, such as rotation and scaling [6, 39]. However, such models do not accurately capture the effects of 3D transformations (even in-plane rotations) of textured surfaces. More recently, there has been a great deal of interest in recognizing images of textured surfaces subjected to lighting and viewpoint changes [8, 9, 33, 35, 57, 58, 62]. Distribution-based methods have been introduced for classifying 3D textures under varying poses and illumination changes. The basic idea is to compute a *texton histogram* based on a universal representative *texton dictionary*. Leung and Malik [33] constructed a *3D texton representation* for classifying a "stack" of registered images of a test material with known imaging parameters. The special requirement of calibrated cameras limits the usage of this method in most practical situations. This limitation was removed by the work of Cula and Dana [8], who used single-image histograms of 2D textons. Varma and Zisserman [57, 58] have further improved 2D texton-based representations, achieving very high levels of accuracy on the Columbia-Utrecht reflectance and texture (CUReT) database [9]. The descriptors used in their work are filter bank outputs [57] and raw pixel values [58]. Hayman et al. [24] extend this method by using support vector machine classifiers with a kernel based on χ^2 histogram distance. Even though these methods have been successful in the complex task of classifying images of materials despite significant appearance changes, their representations themselves are not invariant to the changes in question. In particular, the support regions for computing descriptors are fixed by hand; no adaptation is performed to compensate for changes in surface orientation with respect to the camera. Lazebnik et al. [31] have proposed a different strategy, namely, an intrinsically invariant image representation based on distributions of appearance descriptors computed at a *sparse* set of affine-invariant keypoints (in contrast, earlier approaches to texture recognition can be called *dense*, since they compute appearance descriptors at every pixel). This approach has achieved promising results for texture classification under significant viewpoint changes. In the experiments presented in this paper, we take this approach as a starting point and further improve its discriminative power with a kernel-based

learning method, provide a detailed evaluation of different descriptors and their invariance properties, and place it in the broader context of both texture and object recognition by measuring the impact of background clutter on its performance.

2.2 Object recognition

The earliest work on appearance-based object recognition has mainly utilized *global* descriptions such as color or texture histograms [45, 50, 53]. The main drawback of such methods is their sensitivity to real-world sources of variability such as viewpoint and lighting changes, clutter and occlusions. For this reason, global methods were gradually supplanted over the last decade by *part-based methods*, which became one of the dominant paradigms in the object recognition community. Part-based object models combine appearance descriptors of local features with a representation of their spatial relations. Initially, part-based methods relied on simple Harris interest points, which only provided translation invariance [1, 60]. Subsequently, local features with higher degrees of invariance were used to obtain robustness against scaling changes [18] and affine deformations [30]. While part-based models offer an intellectually satisfying way of representing many real-world objects, learning and inference problems for spatial relations remain extremely complex and computationally intensive, especially in a *weakly supervised* setting where the location of the object in a training image has not been marked by hand. On the other hand, orderless *bag-of-keypoints* methods [55, 61] have the advantage of simplicity and computational efficiency, though they fail to represent the geometric structure of the object class or to distinguish between foreground and background features. For these reasons, bag-of-keypoints methods can be adversely affected by clutter, just as earlier global methods based on color or gradient histograms. One way to overcome this potential weakness is to use feature selection [12] or boosting [48] to retain only the most discriminative features for recognition. Another approach is to design novel kernels that can yield high discriminative power despite the noise and irrelevant information that may be present in local feature sets [23, 37, 59]. While these methods have obtained promising results, they have not been extensively tested on databases featuring heavily cluttered, uncorrelated backgrounds, so the true extent of their robustness has not been conclusively determined. Our own approach is related to that of Grauman and Darrell [23], who have developed a kernel that approximates the optimal partial matching between two feature sets. Specifically, we use a kernel based on the *Earth Mover's Distance* [52], which solves the partial matching problem exactly. Finally, we note that our image representation is similar to that of [61], though our choice of local features and classifier kernel results in significantly higher performance.

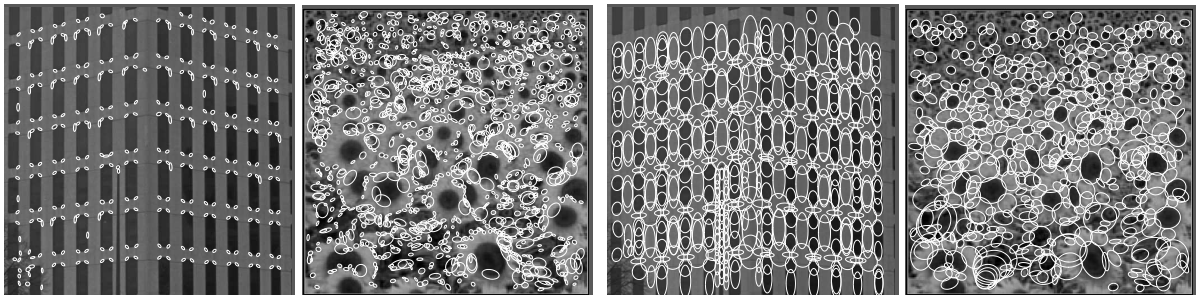
3 Components of the representation

This section introduces our image representation based on sparse local features. We first discuss scale- and affine-invariant local regions and the descriptors of their appearance. We then describe different image signatures and similarity measures suitable for comparing them.

3.1 Scale- and affine-invariant region detectors

In this paper, we use two complementary local region detector types to extract salient image structures: The *Harris-Laplace* detector [43] responds to corner-like regions, while the *Laplacian* detector [34] extracts blob-like regions (Fig. 1).

At the most basic level, these two detectors are invariant to scale transformations alone, i.e., they output circular regions at a certain characteristic scale. To achieve rotation invariance, we can either use rotationally invariant descriptors—for example, SPIN and RIFT [31], as presented in the following section—or rotate the circular regions in the direction of the dominant gradient orientation [36, 43]. In our implementation, the dominant gradient orientation is computed as the average of all gradient orientations in the region. Finally, we obtain affine-invariant versions of the Harris-Laplace and Laplacian detectors through the use of an *affine adaptation* procedure [21, 42]. Affinely adapted detectors output ellipse-shaped regions which are then *normalized*, i.e., transformed into circles. Normalization leaves a rotational ambiguity that can be eliminated either by using rotation-invariant descriptors or by finding the dominant gradient orientation, as described above.



Harris-Laplace detector Laplacian detector

Figure 1: Illustration of affine Harris and Laplacian regions on two natural images.

3.2 Descriptors

The normalized circular patches obtained by the detectors described in the previous section serve as domains of support for computing appearance-based descriptors. Many different descriptors have been presented in the literature (see [41] for an overview). In this paper we use three different descriptors: SIFT, SPIN and RIFT. The SIFT descriptor [36] has been shown to outperform a set of existing descriptors [41], while SPIN and RIFT, introduced by [31], have achieved good performance in the context of texture classification.

The SIFT descriptor computes a gradient orientation histogram within the support region. For each of 8 orientation planes, the gradient image is sampled over a 4×4 grid of locations, thus resulting in a $4 \times 4 \times 8 = 128$-dimensional feature vector for each region. A Gaussian window function is used to assign a weight to the magnitude of each sample point. This makes the descriptor less sensitive to the small changes in the position of the support region and puts more emphasis on the gradients that are near the center of the region.

The SPIN descriptor, based on *spin images* used for matching range data [26], is a rotation-invariant two-dimensional histogram of intensities within an image region. The two dimensions of the histogram are d, the distance of the center, and i, the intensity value. The entry at (d, i) is simply the probability of the occurrence of pixels with intensity value i at a fixed distance d from the center of the patch. We follow the same implementations of the spin images as [31], i.e., it is considered as a soft histogram. In our experiments, we use 10 bins for distance and 10 for intensity value, thus resulting in 100-dimensional feature vectors.

The RIFT descriptor is a rotation-invariant version of SIFT. An image region is divided into concentric rings of equal width, and a gradient orientation histogram is computed within each ring. To obtain rotation invariance, gradient orientation is measured at each point relative to the direction pointing outward from the center. We use four rings and eight histogram orientations, yielding a 32-dimensional feature vector.

To obtain robustness to illumination changes, our descriptors are made invariant to affine illumination transformations of the form $aI(x) + b$. For SPIN and RIFT descriptors each support region is normalized with the mean and standard deviation of the region intensities. For SIFT descriptors the norm of each descriptor is scaled to one [36].

Following the terminology of [31], we consider each detector/descriptor pair as a separate "channel." As explained in Section 3.1, our detectors offer different levels of invariance: scale invariance only (S), scale with rotation invariance (SR), and affine invariance (A). We denote the Harris detector with different levels of invariance as HS, HSR and HA and the Laplacian detector as LS, LSR and LA. Recall that HSR and LSR regions are obtained from HS and LS by finding the dominant gradient orientation, while for HS and LS, the dominant orientation is assumed to be horizontal for the purpose of computing SIFT descriptors. The combination of multiple detector/descriptor channels is denoted by (detector+detector)(descriptor+descriptor), e.g., (HS+LS)(SIFT+SPIN) means the combination of HS and LS detectors each described with SIFT and SPIN descriptors.

3.3 Comparing distributions of local features

After detecting salient local regions and computing their descriptors as described in the previous section, we need to represent their distributions in the training and test images. One method for doing this is to cluster the set of descriptors found in each image to form its *signature* $\{(p_1, u_1), \ldots, (p_m, u_m)\}$, where m is the number of clusters, p_i is the center of the ith cluster, and u_i is the relative size of the cluster (the number of descriptors in the cluster divided by the total number of descriptors extracted from the image). *Earth Mover's Distance* (EMD) [52] has shown to be very suitable for measuring the similarity between image signatures. The EMD between two signatures $S_1 = \{(p_1, u_1), \ldots, (p_m, u_m)\}$ and $S_2 = \{(q_1, w_1), \ldots, (q_n, w_n)\}$ is defined as follows:

$$D(S_1, S_2) = \frac{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}\, d(p_i, q_j)}{\sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij}}$$

where f_{ij} is a flow value that can be determined by solving a linear programming problem, and $d(p_i, q_j)$ is the *ground distance* between cluster centers p_i and q_j. We use Euclidean distance as the ground distance and extract 40 clusters with k-means for each image. Note that EMD is a cross-bin dissimilarity measure and can handle variable-length representation of distributions, i.e., m and n do not have to be the same.

An alternative to image signatures is to obtain a global *texton vocabulary* (or *visual vocabulary*) by clustering descriptors from a special training set, and then to represent each image in the database as a histogram of texton labels [8, 57, 58, 61]. Given a global texton vocabulary of size m, the ith entry of a histogram is the proportion of all descriptors in the image having label i. To compare two histograms $S_1 = (u_1, \ldots, u_m)$ and $S_2 = (w_1, \ldots, w_m)$, we use the χ^2 distance defined as

$$D(S_1, S_2) = \frac{1}{2} \sum_{i=1}^{m} \frac{(u_i - w_i)^2}{u_i + w_i}\,.$$

In our experiments, we extract 10 textons (clusters) with k-means for each class and then concatenate textons of different classes to form a global vocabulary.

3.4 Kernel-based classification

For classification, we use *Support Vector Machines* (SVM) [54]. In a two-class case, the decision function for a test sample x has the following form:

$$g(x) = \sum\nolimits_i \alpha_i y_i K(x_i, x) - b\,, \tag{1}$$

where $K(x_i, x)$ is the value of a *kernel function* for the training sample x_i and the test sample x, y_i the class label of x_i ($+1$ or -1), α_i the learned weight of the training sample x_i, and b is a learned threshold parameter. The training samples with weight $\alpha_i > 0$ are usually called *support vectors*. In the following, we use the two-class setting for binary detection, i.e., classifying images as containing or not a given object class. To obtain a detector response, we use the raw output of the SVM, given by Eq. (1). By placing different thresholds on this output, we vary the decision function to obtain Receiver Operating Characteristic (ROC) curves such as the ones in Figs. 18 to 21. For multi-class classification, different methods to extend binary SVMs tend to perform similarly in practice [54]. We use the one-against-one technique, which trains a classifier for each possible pair of classes. For each new test pattern, all binary classifiers are evaluated, and the pattern is assigned to the class that is chosen by the majority of classifiers. Experimental comparisons on our dataset confirm that the one-against-one and one-against-other techniques give almost the same results.

To incorporate EMD or χ^2 distance into the SVM framework, we use extended Gaussian kernels [5, 25]:

$$K(S_i, S_j) = \exp\big(-\frac{1}{A}\, D(S_i, S_j)\big)\,, \tag{2}$$

where $D(S_i, S_j)$ is EMD (resp. χ^2 distance) if S_i and S_j are image signatures (resp. vocabulary-histograms). The resulting kernel is the *EMD kernel* (or χ^2 *kernel*). A is a scaling parameter that can in principle be determined through cross-validation. We have found, however, that setting its value to the

mean value of the EMD (resp. χ^2) distances between all training images gives comparable results and reduces the computational cost. To combine different channels, we sum their distances, i.e., $D = \sum_i^n D_i$ where D_i is the similarity measure for channel i. We then apply the generalized Gaussian kernel, Eq. (2), to the combined distance.

The χ^2 kernel is a Mercer kernel [20]. We do not have a proof of the positive definiteness for the EMD-kernel; however, in our experiments, this kernel has always yielded positive definite Gram matrices. In addition, it must be noted that even non-Mercer kernels often work well in real applications [5].

4 Empirical Evaluation

4.1 Experimental setup

For our experimental evaluation, we use four texture and five object category datasets, described in detail in the following two sections. The texture datasets are UIUCTex [31], KTH-TIPS [24], Brodatz [3], and CUReT [9]. The object category datasets are Xerox7 [61], Graz [48], CalTech6 [18], CalTech101 [15] and Pascal [14].

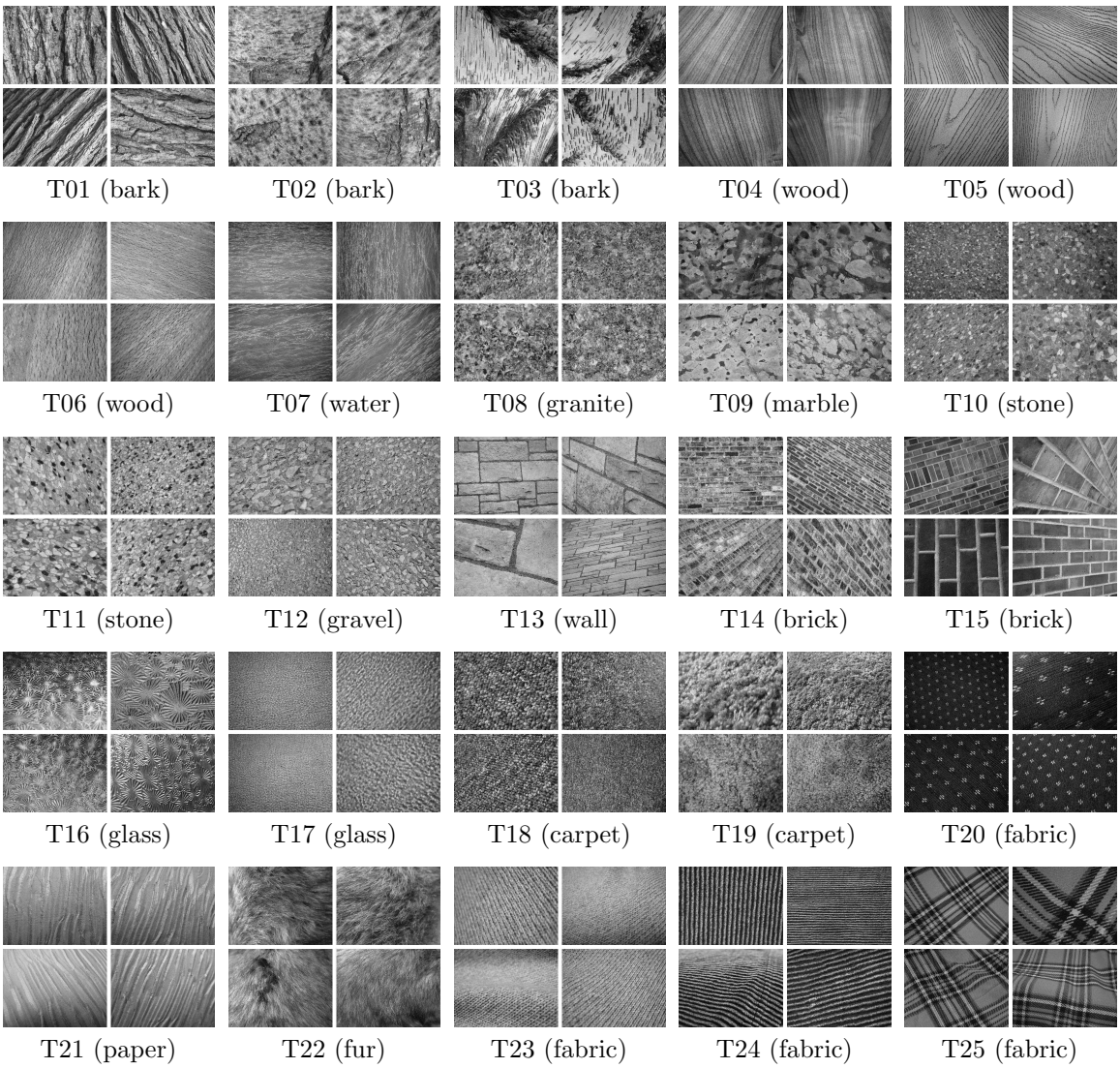


Figure 2: Four samples each of the 25 texture classes of the UIUCTex dataset. The database may be downloaded from `http://www-cvr.ai.uiuc.edu/ponce_grp/data/texture_database`.

4.1.1 Texture datasets

The **UIUCTex** dataset [31] contains 25 texture classes with 40 images per class. Textures are viewed under significant scale and viewpoint changes. Furthermore, the dataset includes non-rigid deformations, illumination changes and viewpoint-dependent appearance variations. Fig. 2 presents four sample images per class, each showing a textured surface viewed under different poses.

The **KTH-TIPS** dataset [24] contains 10 texture classes. Images are captured at nine scales spanning two octaves (relative scale changes from 0.5 to 2), viewed under three different illumination directions and three different poses, thus giving a total of 9 images per scale, and 81 images per material. Example images with scale and illumination changes are shown in Fig. 3. From this figure, we can see that scaling and illumination changes increase the intra-class variability and reduce the inter-class separability. For example, the sponge surface under scale S3 looks somewhat similar to the cotton surface under scale S3. This increases the difficulties of the classification task.

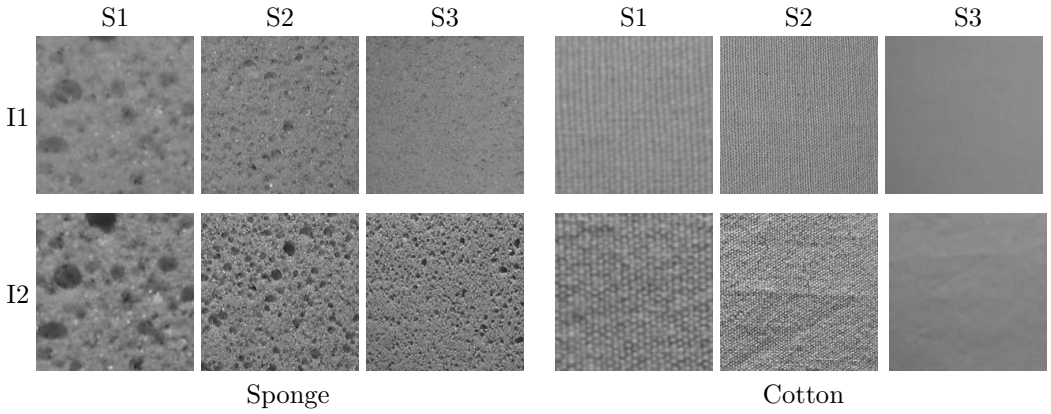


Figure 3: Image examples of the KTH-TIPS database. S1, S2 and S3 indicate different scales, i.e., the relative scales 0.5, 1 and 2.0 respectively. I1, I2 represent two different illuminations. The database may be downloaded from `http://www.nada.kth.se/cvap/databases/kth-tips`.

The **Brodatz** texture album [3] is a well-known benchmark dataset. It contains 112 different texture classes where each class is represented by one image divided into nine sub-images (cf. Fig. 4). Note that this dataset is somewhat limited, as it does not model viewpoint, scale, or illumination changes.

Figure 4: Image examples of the Brodatz textures. Each image is divided into 9 non overlapping sub-images for experiments.

For the **CUReT** texture database [9] we use the same subset of images as [57, 58]. This subset contains 61 texture classes with 92 images for each class. These images are captured under different illuminations with seven different viewing directions. The changes of viewpoint, and, to a greater extent, of the illumination direction, significantly affect the texture appearance, cf. Fig. 5.

For texture classification, we evaluate the dependence of performance on the number of training images per class. To avoid bias, we randomly select 100 different groups of n training images. The remaining

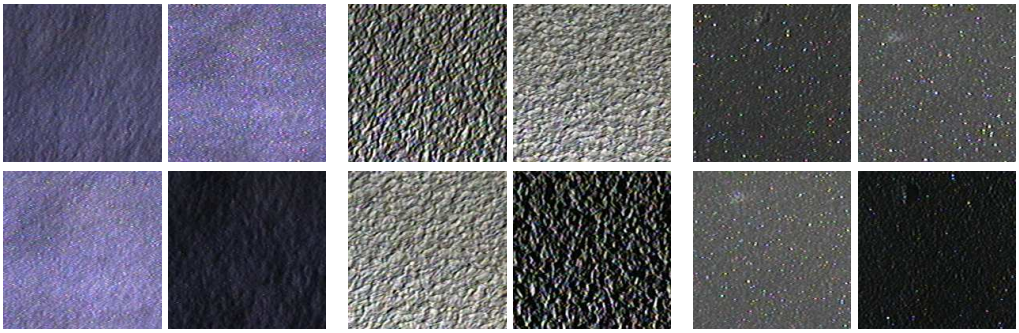


Figure 5: Image examples of CUReT textures under different illuminations and viewpoints.

images are used for testing. The results are reported as the average value and standard deviation over the 100 runs.

4.1.2 Object category datasets

The **Xerox7** dataset [61] consists of 1776 images of seven classes: bikes, books, buildings, cars, faces, phones and trees. This is a challenging dataset, as it includes images with highly variable pose and background clutter, and the intra-class variability is large. Some of the images are shown in Fig. 6. We use the same setup as in [61], i.e., we perform multi-class classification with ten-fold cross-validation and report the average accuracy.

Figure 6: Images of categories bikes, books, buildings, cars, faces, phones and trees of the Xerox7 dataset. Note that all of them are classified correctly with our approach.

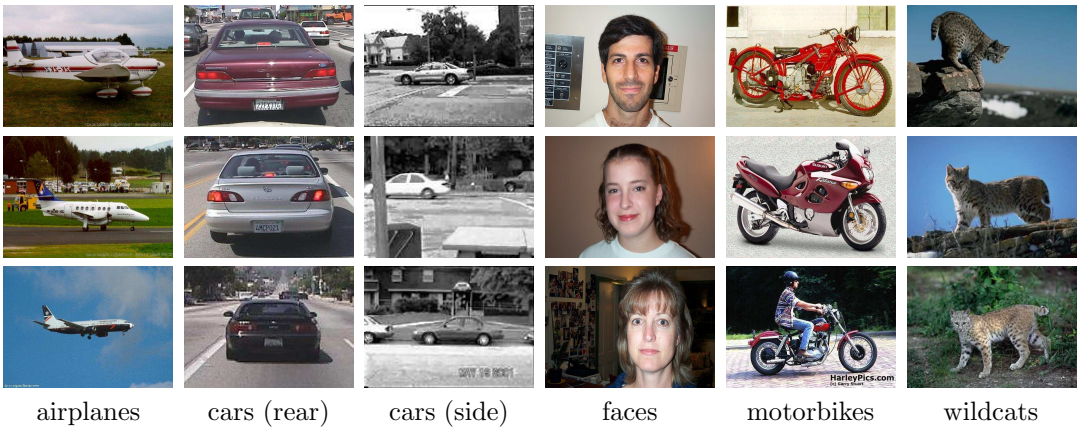


Figure 7: Image examples of the six categories of CalTech6 dataset. The dataset may be downloaded from `http://www.robots.ox.ac.uk/~vgg/data.html`.

The **CalTech6** database [18] contains airplanes (side) (1074 images), cars (rear) (1155 images), cars (side)[1] (720 images), faces (front) (450 images), motorbikes (side) (826 images), spotted cats (200 images), and a background set (900 images). The original category of spotted cats is from the Corel image library and contains 100 images. Here we flipped the original images to have the same set as used in [18]. We use the same training and test set for two-class classification (object vs. background) as [18]. Some images are presented in Fig. 7.

The **Graz** dataset [48] contains persons, bikes and a background class. Some of the images are shown in Fig. 8. We use the same training and test set for two-class classification as [48], for a total of 350 images.

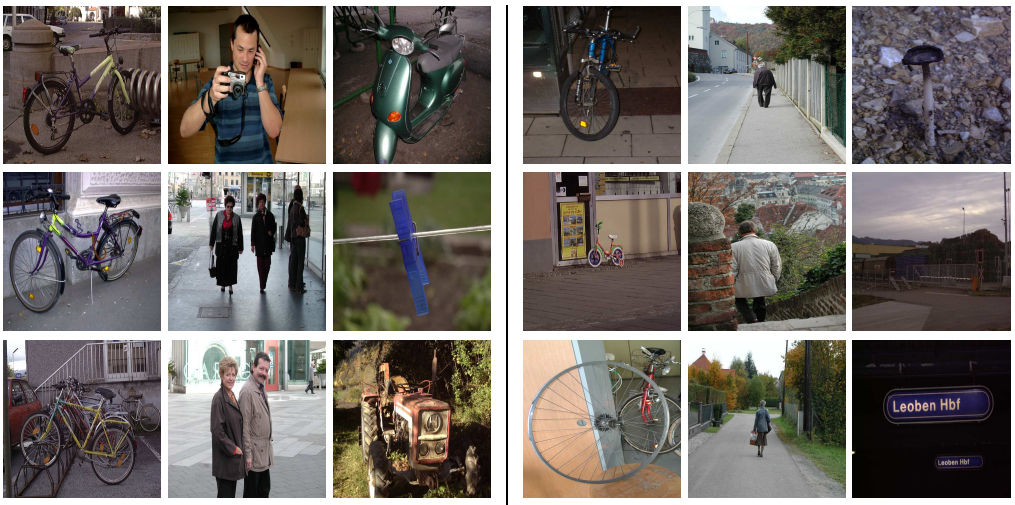


Figure 8: Image examples of the two categories and a background class of Graz dataset. The images on the left are correctly classified with our approach, the images on the right were misclassified. The dataset may be obtained from `http://www.emt.tugraz.at/~pinz/data`.

Figure 9: Image examples with ground truth object annotation of different categories of the PASCAL challenge. The dataset may be obtained from `http://www.pascal-network.org/challenges/VOC/voc/index.html`.

[1]The car (side) images are from the UIUC car dataset [1], `http://l2r.cs.uiuc.edu/~cogcomp/Data/Car`.

The **PASCAL** dataset [14] includes four categories: bicycles, cars, motorbikes and people. It has one training dataset (684 images) and two test sets (test set 1: 689 images, test set 2: 956 images). The goal of the challenge is to determine whether a given image contains an instance of a particular class, i.e., there are four independent binary classification tasks. Image examples from the training set and the two test sets of each category are shown in Fig. 9. In test set 1, expected to make an 'easier' challenge, images are taken from the same distribution as the training images. In test set 2, images are collected by Google search and thus come from a different distribution than the training data. This should make a 'harder' challenge. An additional complication is that many images in test set 2 contain instances of several classes. Note that in this dataset, ground truth annotations of each object are available. They are shown as yellow rectangles in Fig. 9. In Section 5, we will use this information to isolate the contributions of foreground and background features.

The **CalTech101** dataset [15] contains 101 object categories with 40 to 800 images per category. Some of the images are shown in Fig. 10.[2] Most of the images in the database contain little or no clutter. Furthermore, the objects tend to lie in the center of the image and to be present in similar poses. Furthermore, some images (see, e.g., the accordion and pagoda classes in Fig. 10) have a partially black background due to artificial image rotations. We follow the experimental setup of Grauman et al. [23], i.e., we randomly select 30 training images per class and test on the remaining images reporting the average accuracy. We repeat the random selection 10 times and report the average classification accuracy.

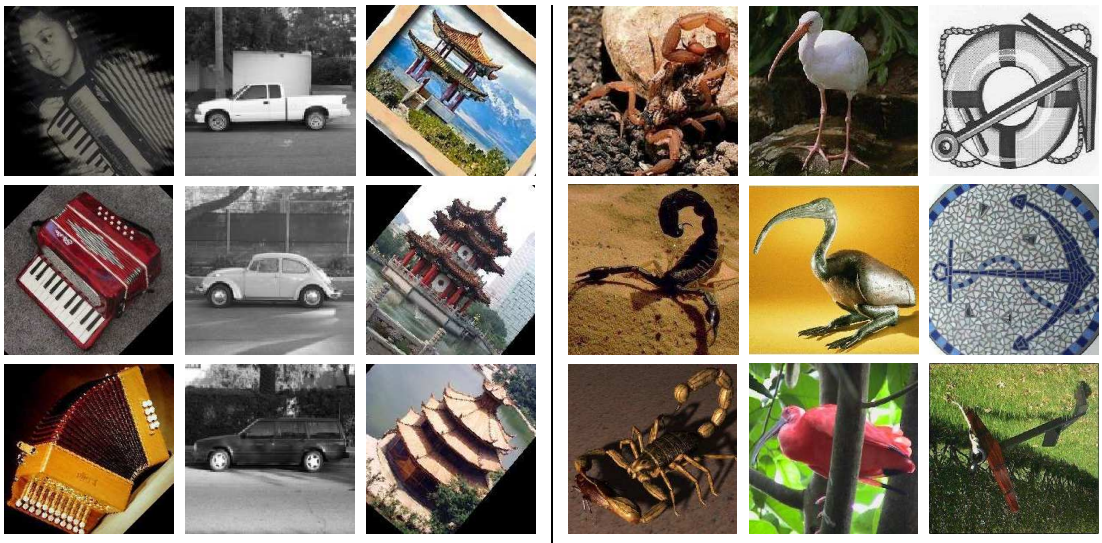


Figure 10: Image examples of the CalTech101 dataset. On the left the three classes with the best classification rates and on the right those with the lowest rates. The dataset may be downloaded from `http://www.vision.caltech.edu/Image_Datasets/Caltech101`.

For object category classification, we use the training and test settings specified above except in section 4.2. In this section we perform random splits for Graz and PASCAL to obtain statistically significant values. Note that we keep the original split proportions, i.e., the number of training and test images. As in the case of textures 100 different groups are selected and results are reported as the average value and standard deviation over the 100 runs.

4.2 Evaluation of implementation parameters

In this section we evaluate the main implementation choices of our approach, including the relative performance of different levels of invariance, different detector/descriptor channels, and different types of

[2]Note that the publicly available dataset contains two classes for face—called face and faces_easy—and that the zebra category cited in [15] is missing.

Databases	Scale Invariance			Scale and Rotation			Affine Invariance		
	HS	LS	HS+LS	HSR	LSR	HSR+LSR	HA	LA	HA+LA
UIUCTex	89.7 ± 1.5	91.2 ± 1.5	92.2 ± 1.4	97.1 ± 0.6	97.7 ± 0.6	**98.0** ± 0.5	97.5 ± 0.6	97.5 ± 0.7	**98.0** ± 0.6
KTH-TIPS	92.9 ± 1.6	**94.9** ± 1.6	94.4 ± 1.7	91.0 ± 1.7	92.5 ± 1.6	92.7 ± 1.6	87.6 ± 1.8	90.1 ± 1.8	90.0 ± 1.7
Brodatz	89.2 ± 1.0	**94.9** ± 0.7	94.4 ± 0.7	89.2 ± 1.0	94.1 ± 0.8	94.0 ± 0.9	84.7 ± 1.1	90.8 ± 0.9	91.3 ± 1.1
Xerox7	92.0 ± 2.0	93.9 ± 1.5	**94.7** ± 1.2	88.1 ± 2.1	92.4 ± 1.7	92.2 ± 2.3	88.2 ± 2.2	91.3 ± 2.1	91.4 ± 1.8
Graz bikes	90.9 ± 2.7	89.6 ± 2.6	**91.9** ± 2.6	89.4 ± 3.0	89.8 ± 2.6	91.3 ± 2.6	88.1 ± 3.4	89.8 ± 3.0	90.5 ± 3.0
PASCAL cars set 1	**92.5** ± 0.9	91.6 ± 0.9	92.4 ± 0.9	87.7 ± 1.1	87.4 ± 1.1	89.0 ± 1.1	87.1 ± 1.1	87.6 ± 1.2	88.7 ± 1.1

Table 1: Evaluation of different levels of invariance. We use the SIFT descriptor and the EMD-kernel. The number of training images per class are 20 for UIUCTex, 40 for KTH-TIPS, 3 for Brodatz. For Graz and PASCAL we keep the proportions of the original split. For Xerox7, we use ten-fold cross-validation.

SVM kernels. To give a complete picture of the functioning of our system, we conclude this section by reporting on the running time of different implementation components.

In the following we present results for 3 texture dataset and 3 object categories sets. The PASCAL dataset defines 8 sub-tasks. Due to space limitations we only present the results for the "cars set 1" task which is representative of all the other results. For Graz we use the "bikes" sub-task.

Evaluation of different levels of invariance. First, we show the results of evaluating different levels of invariance (S, SR, A) of our two keypoint detectors on several datasets. In this test, all regions are described with the SIFT descriptor and the EMD kernel is used for classification. Table 1 shows that pure scale invariance (S) performs best for the Brodatz, KTH-TIPS, Xerox7, Graz and PASCAL datasets, while for UIUCTex, rotation invariance (SR) is important. The reason is that Brodatz, KTH-TIPS, Xerox7, Graz and PASCAL have no rotation or affine changes (in the Xerox7 images for instance, no face is rotated by more than 45 degrees and no car is upside down), while UIUCTex has significant viewpoint changes and arbitrary rotations. Even in this case, affine-invariant features fail to outperform the scale- and rotation-invariant ones. Thus, somewhat surprisingly, affine invariance does not help even for datasets with significant viewpoint changes, such as UIUCTex.

There are two possible causes for the apparent advantage enjoyed by the detectors with lower levels of invariance. First, the normalization process necessary for obtaining rotation- and affine-invariant features may lose potentially discriminative information, resulting in weaker features in situations where such invariance is not necessary. Second, detectors with a high degree of invariance may be computationally less stable. However, independently of the cause, the practical choice is clear. Since using local features with the highest possible level of invariance does not yield the best performance for most datasets in our evaluation, an effective recognition system should not exceed the local invariance level absolutely required for a given application.

Evaluation of different channels. Next, we compare the performance of different detector/descriptor channels and their combinations. We use the EMD kernel for classification and report results for the level of invariance achieving the best performance for each dataset. Tables 2 to 7 show results for three texture datasets and three object datasets (the behavior of all the channels on the other datasets is similar). We can see that the Laplacian detector tends to perform better than the Harris detector. The most likely reason for this difference is that the Laplacian detector tends to extract four to five times more regions per image than Harris-Laplace, thus producing a richer representation. Using the two detectors together tends to further raise performance. SIFT performs better than SPIN and RIFT, while the performance rank between SPIN and RIFT depends on dataset. It is not surprising that RIFT performs worse than SIFT, since it averages gradient orientations over a ring-shaped region and therefore loses important spatial information. We have experimented with increasing the dimensionality of RIFT to 128, but this did not improve its performance. Combining SIFT with SPIN and RIFT with SPIN boosts the overall performance because the two descriptors capture different kinds of information (gradients vs. intensity values). As expected, however, combining RIFT with SIFT and SPIN results only in an insignificant improvement, as SIFT and RIFT capture the same type of information. Overall, the combination of Harris-Laplace and Laplacian detectors with SIFT and SPIN is the preferable choice in terms of classification accuracy, and this is the setup used in Sections 4.3 and 4.4. In Section 5, we drop the SPIN descriptor for computational efficiency. Finally, it is interesting to note that in the nearest-neighbor

Channels	SIFT	SPIN	RIFT	SIFT+SPIN	RIFT+SPIN	SIFT+SPIN+RIFT
HA	97.5 ± 0.6	95.5 ± 0.8	94.8 ± 0.8	97.9 ± 0.6	97.1 ± 0.7	98.1 ± 0.6
LA	97.5 ± 0.7	96.0 ± 0.9	96.4 ± 0.7	98.1 ± 0.6	97.8 ± 0.6	98.5 ± 0.5
HA+LA	98.0 ± 0.6	97.0 ± 0.7	97.0 ± 0.7	98.5 ± 0.5	98.0 ± 0.6	**98.7** ± 0.4
HSR	97.1 ± 0.6	93.9 ± 1.1	95.1 ± 0.9	97.4 ± 0.6	96.5 ± 0.8	97.8 ± 0.7
LSR	97.7 ± 0.6	93.9 ± 1.0	94.8 ± 1.0	98.2 ± 0.6	96.9 ± 0.8	98.4 ± 0.5
HSR+LSR	98.0 ± 0.5	96.2 ± 0.8	96.0 ± 0.9	98.3 ± 0.5	97.7 ± 0.7	98.5 ± 0.5

Table 2: Detector and descriptor evaluation on UIUCTex using 20 training images per class.

Channels	SIFT	SPIN	RIFT	SIFT+SPIN	RIFT+SPIN	SIFT+SPIN+RIFT
HS	89.2 ± 1.0	86.1 ± 1.1	82.7 ± 1.0	93.7 ± 0.8	89.8 ± 1.1	94.2 ± 0.9
LS	94.9 ± 0.7	87.9 ± 1.0	88.5 ± 0.9	94.7 ± 0.8	91.4 ± 0.9	95.2 ± 0.7
HS+LS	94.4 ± 0.7	90.2 ± 1.0	89.6 ± 1.0	95.4 ± 0.7	92.8 ± 0.8	**95.9** ± 0.6

Table 3: Detector and descriptor evaluation on Brodatz using 3 training images per class.

Channels	SIFT	SPIN	RIFT	SIFT+SPIN	RIFT+SPIN	SIFT+SPIN+RIFT
HS	92.9 ± 1.6	90.8 ± 1.6	82.3 ± 2.0	94.2 ± 1.6	91.7 ± 1.6	94.1 ± 1.4
LS	94.9 ± 1.6	94.7 ± 1.2	86.5 ± 1.9	**96.1** ± 1.2	95.0 ± 1.3	**96.1** ± 1.1
HS+LS	94.4 ± 1.7	94.2 ± 1.4	86.7 ± 1.8	95.5 ± 1.3	94.3 ± 1.4	95.6 ± 1.2

Table 4: Detector and descriptor evaluation on KTH-TIPS using 40 training images per class.

Channels	SIFT	SPIN	RIFT	SIFT+SPIN	RIFT+SPIN	SIFT+SPIN+RIFT
HS	92.0 ± 2.0	83.0 ± 1.9	83.8 ± 3.1	91.4 ± 2.1	87.8 ± 2.4	92.0 ± 2.0
LS	93.9 ± 1.5	88.6 ± 2.0	89.1 ± 1.1	94.3 ± 0.9	90.8 ± 1.4	93.9 ± 1.5
HS+LS	94.7 ± 1.2	89.5 ± 1.4	89.3 ± 1.5	94.3 ± 1.1	91.5 ± 1.0	**94.7** ± 1.3

Table 5: Detector and descriptor evaluation on Xerox7 using ten-fold cross-validation.

Channels	SIFT	SPIN	RIFT	SIFT+SPIN	RIFT+SPIN	SIFT+SPIN+RIFT
HS	90.9 ± 2.7	83.3 ± 3.9	88.9 ± 3.6	91.0 ± 2.9	88.5 ± 3.2	91.5 ± 2.6
LS	89.6 ± 2.6	88.1 ± 3.5	88.0 ± 3.2	91.6 ± 2.5	90.3 ± 2.5	91.7 ± 2.1
HS+LS	91.9 ± 2.6	88.4 ± 3.2	90.6 ± 2.2	93.0 ± 2.3	90.9 ± 2.8	**93.1** ± 2.6

Table 6: Detector and descriptor evaluation on Graz bikes keeping original training set and test set proportions.

Channels	SIFT	SPIN	RIFT	SIFT+SPIN	RIFT+SPIN	SIFT+SPIN+RIFT
HS	92.5 ± 0.9	85.3 ± 1.1	87.0 ± 1.1	92.4 ± 0.9	89.6 ± 0.9	93.5 ± 0.8
LS	91.6 ± 0.9	84.3 ± 1.1	90.9 ± 0.9	91.5 ± 1.0	90.3 ± 1.0	93.0 ± 0.8
HS+LS	92.4 ± 0.9	86.4 ± 1.1	90.5 ± 1.0	92.6 ± 0.9	91.0 ± 0.9	**93.6** ± 0.8

Table 7: Detector and descriptor evaluation on PASCAL cars set 1 keeping original training set and test set proportions.

classification framework of [31], direct combination of different channels frequently results in a significant decrease of overall performance. However, in a kernel-based classification framework this problem is encountered less often. In cases when the distance estimates provided by one of the channels are much more noisy or unreliable than those of the others (i.e., when that channel is much less discriminative than the others), the noise degrades the performance of the nearest-neighbor classifier, but not of the SVM. This is probably due to the fact, that NN classifier simply compares the averaged channel distances, while SVM combines the number of distances to weighted training examples incorporating the distance values themselves. The robustness of the SVM classifier to noise and irrelevant information is also confirmed by our background evaluation of Section 5, where it is shown that a classifier trained on images containing both object and clutter features performs quite well on cleaner test images.

Evaluation of different kernels. The learning ability of a kernel classifier depends on the type of kernel used. Here we compare SVM with five different kernels, i.e, linear, quadratic, Radial Basis Function (RBF), χ^2, and EMD. As a baseline, we also evaluate EMD-NN, i.e., EMD with nearest-neighbor classification[3] [31]. For the signature-based classifiers (EMD-NN and EMD kernel), we use 40 clusters per image as before. For the other SVM kernels, which work on histogram representations, we create a global vocabulary by concatenating 10 clusters per class. For UIUCTex, KTH-TIPS, Brodatz, Xerox7, Graz and PASCAL the vocabulary sizes are 250, 100, 1120, 70, 20 and 40, respectively. Table 8 shows classification results for the LSR+SIFT channel, which are representative of all other channels. We can see that EMD-NN always performs worse than the EMD kernel, i.e., that a discriminative approach gives a significant improvement. The difference is particularly large for the Xerox7 database, which has wide intra-class variability. Among the vocabulary/histogram representations, the χ^2 kernel performs better than linear, quadratic, and RBF. Results for the EMD kernel and the χ^2 kernel are comparable. Either of the kernels seem to be a good choice for our framework, provided that a suitable vocabulary can be built efficiently. To avoid the computational expense of building global vocabularies for each dataset, we use the EMD kernel in the following experiments.

Databases	Vocabulary-Histogram				Signature	
	Linear	Quadratic	RBF	χ^2 kernel	EMD-NN	EMD-Kernel
UIUCTex	97.0 ± 0.6	84.8 ± 1.6	97.3 ± 0.7	**98.1** ± 0.6	95.0 ± 0.8	**97.7** ± 0.6
KTH-TIPS	91.9 ± 1.4	75.8 ± 1.9	94.0 ± 1.2	**95.0** ± 1.2	88.2 ± 1.6	**92.5** ± 1.5
Brodatz	96.1 ± 0.8	86.3 ± 1.5	**96.2** ± 0.7	96.0 ± 0.7	86.5 ± 1.2	**94.1** ± 0.8
Xerox7	79.8 ± 3.0	70.9 ± 2.4	86.2 ± 2.2	**89.2** ± 2.1	59.4 ± 4.1	**92.4** ± 1.7
Graz bikes	**83.9** ± 3.6	83.2 ± 3.5	83.2 ± 3.0	83.8 ± 2.0	84.6 ± 3.4	**89.8** ± 2.6
PASCAL cars set 1	76.8 ± 1.3	77.4 ± 1.3	77.0 ± 1.8	**80.6** ± 1.3	74.1 ± 1.6	**87.4** ± 1.1

Table 8: Classification accuracy of different kernels for LSR+SIFT. The number of training images per class are 20 for UIUCTex, 40 for KTH-TIPS, 3 for Brodatz. For Graz and PASCAL we keep the proportions of the original split. For Xerox7, we use ten-fold cross-validation.

Evaluation of different signature sizes. Figure 11 shows the influence of signature length on the classification results for PASCAL test set 1 using SVM with EMD kernel. The results are given for a (LS+HS)(SIFT+SPIN) image description with variable signature lengths. We can see that a signature length between 20 and 40 is a good choice for this dataset. We have observed a similar behavior for the other datasets. In general, very short signatures can lead to a significant performance drop, whereas a signature length above 40 does not improve performance, but increases the computational complexity significantly.

Evaluation of running time. The implementation of our recognition system consists of the following major stages: region detection, description computation, clustering, training (computing the Gram matrix based on EMD or χ^2 distances between each pair of training images, and learning the SVM parameters), and testing. Here we take the PASCAL dataset as an example to give a detailed evaluation of the computational cost of each stage. The size of the vocabulary is 1000 (250 clusters per class), which is sufficient to have good results. All components of our system are implemented in C and run on a computer with a 3 GHz Intel CPU and 1 GB of RAM. Tables 9 and 10 report average running times obtained by dividing the total running time of each stage by the number of images or comparisons. In

[3]We also tried k-nearest-neighbor with $k > 1$, but did not observe better performance.

		HS+SIFT	LS+SIFT	HS+SPIN	LS+SPIN
Region Detection	$\times(n+m)$	0.62s	0.96s	0.62s	0.96s
Descriptor Computation	$\times(n+m)$	1.39s	3.29s	5.67s	12.64s
Clustering (signature)	$\times(n+m)$	0.27s	1.28s	0.235s	1.18s
Training	$\times n(n-1)/2$	0.0024s	0.0024s	0.0026s	0.0026s
Testing	$\times(\sharp sv \cdot m)$	0.0023s	0.0021s	0.0024s	0.0024s

Table 9: Evaluation of computational cost on the PASCAL dataset using EMD kernel.

		HS+SIFT	LS+SIFT	HS+SPIN	LS+SPIN
Region Detection	$\times(n+m)$	0.62s	0.96s	0.62s	0.96s
Descriptor Computation	$\times(n+m)$	1.39s	3.29s	5.67s	12.64s
Clustering (vocabulary)	$\times c$	6.13m	6.13m	5.75m	6.25m
Histogramming	$\times(n+m)$	0.68s	2.18s	0.64s	1.93s
Training	$\times n(n-1)/2$	0.00022s	0.00022s	0.00020s	0.00022s
Testing	$\times(\sharp sv \cdot m)$	0.00034s	0.00034s	0.00031s	0.00032s

Table 10: Evaluation of computational cost on PASCAL dataset using χ^2 kernel. Note that the first two rows of this table are the same as those of Table 9.

the tables, $n = 684$ is the number of training images, $m = 1645$ is the number of test images (test sets 1 and 2 combined), $c = 4$ is the number of classes, and $\sharp sv$ is the number of support vectors. The range of $\sharp sv$ is 382 to 577 for the EMD kernel, and 268 to 503 for the χ^2 kernel. Note that in listing the running time of the training stage, we neglect the time for training the SVM, i.e., determining the parameters α_i and b of eq. (1), since it is dominated by the time for computing the Gram matrix.

We can see that the Laplacian channel is usually slower than the Harris-Laplace channel due to its much denser representation. Also, the computation of SPIN is a bit slower than SIFT because we implement SPIN as a soft histogram [31], which involves a large number of exponential computations. Furthermore, we can see that the bottleneck of the computation cost for the χ^2 method is the stage of forming the global texton vocabulary, whereas for the EMD the computation of Gram matrices (necessary during training and testing) is quite time-consuming. In our implementation, we use a standard k-means program with 8 different initializations, and select the one with the lowest error at convergence (the clustering times reported in the tables are the averages for one round).

Taking the (HS+LS)(SIFT+SPIN) combination of the channels, the total training time for $n = 684$ including 8 runs of k-means is 9h49m for the EMD kernel and 18h42m for the χ^2 kernel. The average time for classifying a test image of the PASCAL database with (HS+LS)(SIFT+SPIN) is 53.1s for the EMD kernel and 30.7s for the χ^2 kernel. Overall, the χ^2 kernel is slower than the EMD kernel when considering the vocabulary construction time. This is the main reason that we have preferred the EMD

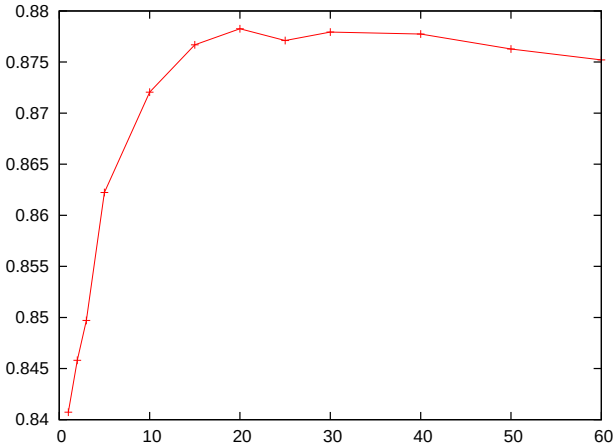


Figure 11: PASCAL test set 1 classification results influenced by signature length.

kernel for most of our evaluations. In the future, we plan to experiment with efficient clustering methods, such as x-means with kd-trees [49], to improve the speed of vocabulary construction.

4.3 Texture classification

In this section, we present a comparative evaluation of our approach with four state-of-the-art texture classification methods: Lazebnik's method [31], VZ-joint [58], Hayman's method [24], and global Gabor filters [38]. Lazebnik's method uses (HA+LA)(SPIN+RIFT) for image description, and nearest neighbor classification with EMD. VZ-joint [58] uses $N \times N$ pixel neighborhoods as image descriptors and performs clustering on a dense representation. In our experiments we use $N = 7$, i.e., a 49-dimensional feature vector as suggested in [58]. Each pixel is labeled by its nearest texton center, and the representation is the distribution of all of the texton labels. χ^2 distance is used as similarity measure and combined with nearest-neighbor classification. Hayman's method [24] is an extension of the VZ approach. They use the VZ-MR8 (maximum filter response independent of orientation) descriptor [57] and SVM with χ^2 kernel for classification. In our implementation we use the VZ-joint descriptor instead of VZ-MR8, as VZ-joint has been shown to give better results [58]. Compared with the results for the KTH-TIPS database reported in [24], our implementation gives slightly higher classification accuracy for the same training and test set. Finally, global Gabor filters [38] is a "traditional" texture analysis method using global mean and standard deviation of the responses of Gabor filters. We use the same Gabor filters as in [38], i.e., 6 orientations and 4 scales. Classification is nearest neighbor based on the Mahalanobis distance.

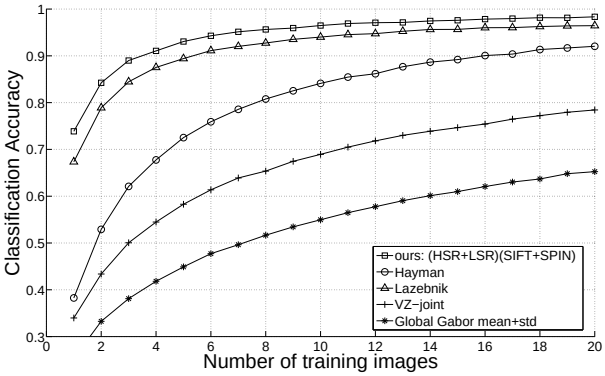


Figure 12: Comparison of different methods on the UIUCTex database.

Comparison on UIUCTex database. Fig. 12 shows the classification accuracy of the five different methods for a varying number of training images. We can observe that both our method and Lazebnik's method work much better than Hayman's method and VZ-joint, while Hayman's method works better than VZ-joint. Overall, the improved performance of our method over Lazebnik's and of Hayman over VZ-joint shows that discriminative learning helps to achieve robustness to intra-class variability. On this dataset, global Gabor features perform the worst, since they are not invariant and averaging the features

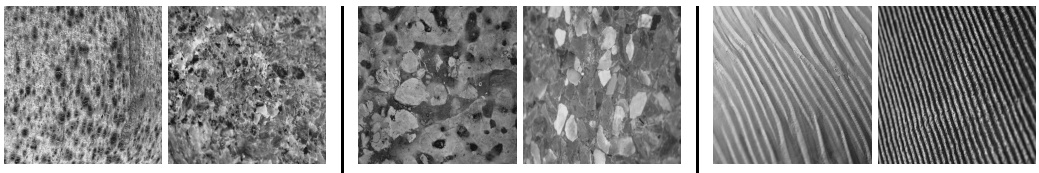


Figure 13: Example images of the most difficult texture classes of UIUC database and their confused classes. The left example of each pair shows the difficult texture class and its classification accuracy. The right example shows the most confused class and the confusion rate for the class on the left.

over all pixels loses discriminative information. Overall, the three non-invariant dense methods in our evaluation have relatively weak performance on this database, especially for smaller numbers of training images, where they cannot take advantage of multiple exemplars to learn intra-class variations that are not compensated for at the representation level. Finally, Fig. 13 shows three classes that get the lowest classification rates with our approach and the classes most often confused with those. We can see that perceptual similarity helps to account for this confusion.

Comparison on KTH-TIPS database. Fig. 14 shows the classification accuracy of the five different methods on the KTH-TIPS database for a varying number of training images. We can observe that our method works best, Hayman's comes second, and VZ-joint and Lazebnik's method are below them. Lazebnik's method performs worse on this database than on UIUCTex because its image representation is not invariant to illumination changes, and it does not incorporate a discriminative learning step to compensate for this weakness. Global Gabor filters come in last, though they still give good results and their performance is significantly higher for this database than for UIUCTex. This may be due to the relative homogeneity of the KTH-TIPS texture classes. Note the increase in performance of the global Gabor method between 10 and 40 training images, which confirms that a method with a non-invariant representation needs multiple exemplars to achieve high performance in the presence of significant intra-class variations due to lighting changes.

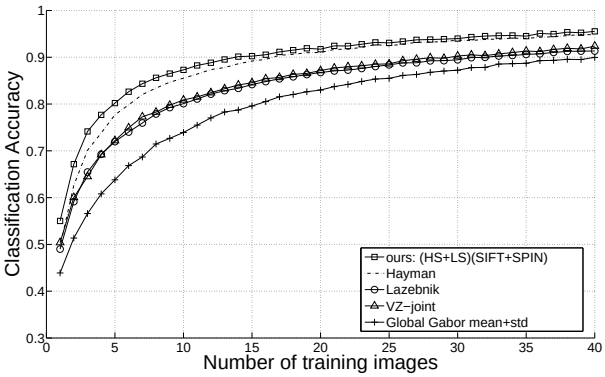


Figure 14: Comparison of different methods on the whole KTH-TIPS texture database

Comparison on Brodatz database. Table 11 shows results for one and three training images per class. Our method performs best, closely followed by Hayman's method. We can see that Hayman's method performs better than VZ-joint, and our method better than Lazebnik's method. This shows that kernel-based learning improves the performance over nearest neighbor classification.

methods	training images per class	
	1	3
ours: (HS+LS) (SIFT+SPIN)	**88.8** ± 1.0	**95.4** ± 0.3
Hayman	88.7 ± 1.0	95.0 ± 0.8
Lazebnik	80.0 ± 1.3	89.8 ± 1.0
VZ-joint	87.1 ± 0.9	92.9 ± 0.8
Global Gabor mean+std	80.4 ± 1.2	87.9 ± 1.0

Table 11: Comparison on the Brodatz database

Comparison on CUReT database. Fig. 15 shows that Hayman's method obtains the best results, followed by VZ-joint, our method, global Gabor filters, and Lazebnik's method. On this dataset, local

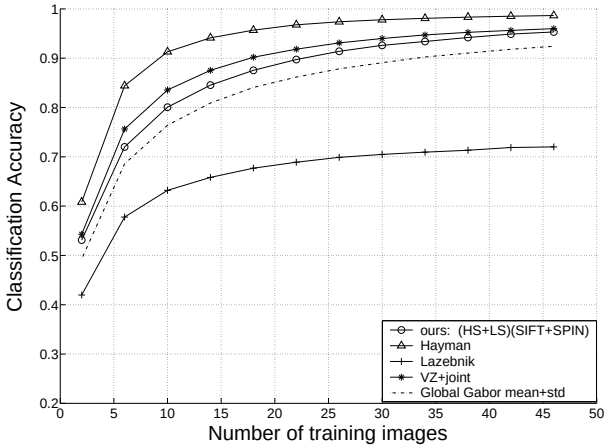


Figure 15: Comparison of different methods on the CUReT texture database

feature methods are at a disadvantage. Since most of the CUReT textures are very homogeneous and high-frequency, lacking salient structures such as blobs and corners, keypoint extraction does not produce very good image representations. A simple patch descriptor seems to be more appropriate.

Discussion. Our method achieves the highest accuracy on three texture databases and comparable results on the CUReT dataset. Its robustness to viewpoint and scale changes has been clearly demonstrated on the UIUCTex and the KTH-TIPS datasets. Our results show that for most datasets, combining geometric invariance at the representation level with a discriminative classifier at the learning level, results in a very effective texture recognition system. Note that even though impressive results are obtained using VZ-joint (patch descriptors) on the CUReT and Brodatz datasets, this method does not perform as well on the other datasets, thus showing its limited applicability. An important factor affecting the performance of local feature methods is image resolution, since keypoint extraction tends to not work well on low-resolution images. For example, CUReT images of size 200×200 have on average 236 Harris-Laplace regions and 908 Laplacian regions, while UIUCTex images of size 640×480 have an average of 2152 and 8551 regions, respectively. As in our earlier tests showing the advantage of the denser Laplacian detector over the sparser Harris-Laplace, extracting larger numbers of keypoints seems to lead to better performance.

4.4 Object category classification

In this section we evaluate our approach for object category classification and compare it to the results reported in the literature. In the following experiments, we use the combination of the Harris-Laplace and Laplacian detectors described with SIFT and SPIN unless stated otherwise. The EMD kernel and SVM are used for classification.

Comparison on Xerox7. Table 12 shows overall results for multi-class classification on the Xerox7 database. Our method outperforms the Xerox bag-of-keypoints method [61] in the same experimental setting. This is due to the fact that we use a combination of detectors and descriptors, a more robust kernel (EMD vs. linear, see the bottom line of table 8) and scale invariance as opposed to affine invariance (see table 1). Fig. 6 shows some images correctly classified by our method. Table 13 shows the confusion

	ours: (HS+LS) (SIFT+SPIN)	Xerox [61]
overall rate	**94.3**	82.0

Table 12: Classification accuracy on the Xerox7 database.

category	bikes	books	buildings	cars	faces	phones	trees	rate
bikes	122		1				2	97.6
books		116	1	9	12	4		81.7
buildings	1	5	123	5	10	1	5	82.0
cars			3	178	14	6		88.6
faces	1			1	787		3	99.4
phones		5	1	3	4	203		94.0
trees	1		1		2		146	97.3

Table 13: Confusion matrix for the Xerox7 dataset.

matrix and the classification rates for the individual categories[4]. The most difficult categories are books, buildings and cars. Fig. 16 shows some of the misclassified images for these categories. The first row of Fig. 16 shows book images misclassified as face and building. The second row shows building images misclassified as face and tree: there are trees in front of the buildings. The third row shows car images misclassified as building and phone. The image on the left does contain buildings. This shows the limitation of a whole-image classification method when a test image contains instances of multiple objects.

misclassified books (faces, faces, buildings)

misclassified buildings (faces, trees, trees)

misclassified cars (buildings, phones, phones)

Figure 16: Misclassified images of the Xerox7 dataset.

Comparison on Caltech6 and Graz. Results for two-class classification (object vs. background) on the CalTech6 and Graz databases are reported with the ROC equal error rate.[5] Table 14 compares our CalTech6 results to three of the state-of-the-art methods—the Xerox approach [61], Fergus et al. [18] and Deselaers et al. [11]. We can see that our method performs best on four out of six object classes and achieves comparable results on the remaining two. The results obtained by the other methods are also quite high, indicating the relatively low level of difficulty of the CalTech6 dataset. We also tested our method for two-class classification on the Graz dataset [48] (Table 15). Our method performs significantly better than Opelt et al. [48]. Fig. 8 shows some images correctly classified by our method and misclassified ones. Misclassified bikes are either observed from the front, very small, or only partially

[4]Note that the overall rate is the average over the individual rates weighted by the number of images in the category.
[5]Point on the ROC curves for which $p(TruePositives) = 1 - p(FalsePositives)$.

	ours: (HS+LS)(SPIN+SIFT)	Xerox [61]	Fergus [18]	Deselaers [11]
airplanes	**98.8**	97.1	90.2	98.6
cars (rear)	98.3	**98.6**	90.3	N/A
cars (side)	**95.0**	87.3	88.5	N/A
faces	**100**	99.3	96.4	96.3
motorbikes	98.5	98.0	92.5	**98.9**
spotted cats	**97.0**	N/A	90.0	N/A

Table 14: ROC equal error rates on the CalTech6 dataset.

	ours: (HS+LS) (SPIN+SIFT)	Opelt [48]
bikes	**92.0**	86.5
people	**88.0**	80.8

Table 15: ROC equal error rates on the Graz database.

visible. Misclassified people are either observed from the back, occluded, or very small.

Comparison on the PASCAL database. We also evaluate our approach for the object category classification task of the PASCAL challenge [14], sample images from which were shown in Fig. 9. Table 16 shows ROC equal error rates of our method for detecting each class vs. the others[6] as well as of the other best method reported in the PASCAL challenge. For test set 1 the best results, slightly better than ours, were obtained by Larlus [29]. This approach uses a dense set of multi-scale patches instead of a sparse set of descriptors computed at interest points. For test set 2 best results, below ours, were obtained by Deselaers et al. [10]. They use a combination of patches around interest points and patches on a fixed grid. A short description of all the participating methods may be found in [14].

	test set 1				test set 2			
	HS	LS	HS+LS	Larlus [29]	HS	LS	HS+LS	Deselaers [10]
bikes	87.7	89.4	90.3	**93.0**	67.3	**68.4**	68.1	66.7
cars	92.7	92.3	93.0	**96.1**	71.2	72.3	**74.1**	71.6
motorbikes	93.0	95.8	96.2	**97.7**	75.7	**79.7**	**79.7**	76.9
people	90.4	90.4	91.6	**91.7**	73.3	72.8	**75.3**	66.9

Table 16: ROC equal error rates for object detection on the PASCAL challenge using the combination of SIFT and SPIN descriptors and EMD kernel.

Comparison on the CalTech101. Table 17 shows the results for multi-class classification on CalTech101 dataset. Our approach outperforms Grauman et al. [23] for the same setup. The best results on this dataset (48%) are currently reported by Berg et al. [2]. However, these results are not comparable to ours, since they were obtained in a supervised setting with manually segmented training images. Fig. 10 presents the categories with the best and worst classification rates. We can observe that some of the lowest rates are obtained for categories that are characterized by their shape as opposed to texture, such as anchors.

	ours: (HS+LS) (SIFT+SPIN)	Berg [2]	Grauman [23]
avg.	**53.9**	48	43

Table 17: Classification accuracy on the CalTech101 dataset.

Discussion. Our method achieves the highest accuracy on Xerox7, Graz, CalTech6, CalTech101 and PASCAL test set 2. Slightly better results on PASCAL test set 1 were achieved using a dense method [29].

[6]Note that the results reported here differ slightly from those of the PASCAL challenge. Here we have used the same parameter settings as in the rest of the paper, which are not exactly the same as those in the submission to the PASCAL challenge.

Results for this method are officially published only for PASCAL test set 1, but a recent unpublished evaluation on PASCAL test set 2 reports results slightly worse than ours. It is also worth noting, that the complexity of the mentioned dense method is noticeably higher than ours.

We can observe varying levels of difficulty of the different datasets. Almost perfect results are achieved on the CalTech6, whereas significant room for improvement exists for PASCAL test set 2 and CalTech101.

5 Object category classification—influence of background

Figure 17: Image examples of the constant natural scene background. They are captured with lighting changes and the movement of clouds and trees.

Our method recognizes object categories in the presence of various backgrounds without segmentation. Thus, it takes both foreground and background features as input. In the following, we examine the roles of these features in discriminating the object categories from the PASCAL challenge. All of the experiments here are done using the signature/EMD kernel framework. Images are characterized with (HS+LS)(SIFT), SPIN is dropped for computational efficiency. Signature size is set to 40 per image.

PASCAL images are annotated with ground truth object regions, as shown in Fig. 9. For each image, we extract two sets of features: foreground features (FF) are those located within the object region, and background features (BF) are those located outside the object region. Note that many object categories have fairly characteristic backgrounds. In the case of cars, for example, most of the images contain a street, a parking lot, or a building. To determine whether this information provides additional cues for classification, we examine the change in classification performance when the original background features from an image are replaced by two specially constructed alternative sets: *random* and *constant natural scene* backgrounds (referred to as *BF-RAND* and *BF-CONST*, respectively). BF-RAND are obtained by randomly shuffling background features among all of the images in the PASCAL dataset. For example, the background of a face image may be replaced by the background of a car image. Note that the total number of features and the relative amount of clutter in an image may be altered as a result of this procedure. BF-CONST are background features extracted from images captured by a fixed camera observing a natural scene over an extended period of time, so they include continuous lighting changes and the movement of trees and clouds (Fig. 17).

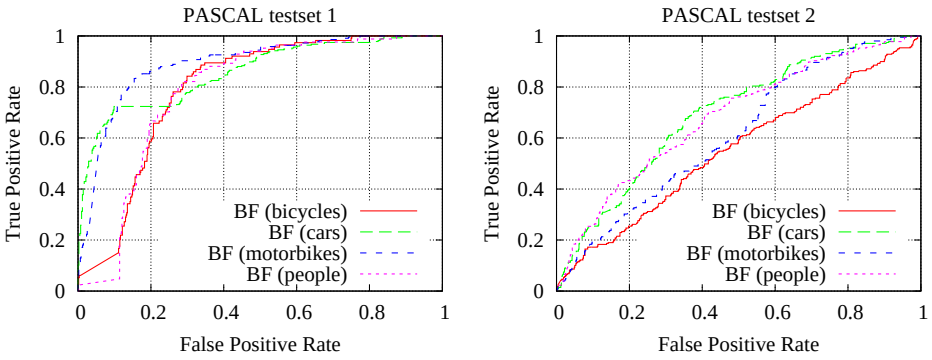


Figure 18: ROC curves of object classification on the PASCAL challenge obtained by training and testing on background features only. The left figure corresponds to test set 1, and the right one to test set 2.

Fig. 18 shows ROC curves obtained by training and testing categories on only the background features (BF) for all four classes in test sets 1 and 2. We can observe that the background features contain a lot of discrimination information for test set 1, i.e., using the background features alone is often sufficient to determine the category of the image. Background features are significantly less discriminant for test set 2. For example, the performance of background features for bicycles is close to chance level. Thus, one of the reasons why test set 2 is considered more difficult than test set 1, is the fact that its background features are much less correlated with the foreground. The performance of the BF-RAND and BF-CONST feature sets is at chance level as one would expect, since they do not contain any information about the foreground object class by construction.

Figs. 19, 20 and 21 evaluate combinations of foreground features with different types of background features. For the sake of brevity, we show only the results for test sets 1 and 2 of the people category, which is representative of the others. The rest of the curves may be found in our technical report [63]. AF denotes the features extracted from the original image, i.e., a combination of FF and BF; AF-RAND denotes the combination of foreground features with randomly selected background features, i.e., FF and BF-RAND; and AF-CONST denotes the combination of foreground features with identically distributed background features, i.e., FF and BF-CONST. Fig. 19 shows ROC curves for a situation where training and testing are performed on the same feature combination. In order of decreasing performance, these combinations are: FF, AF-CONST, AF, AF-RAND. FF always gives the highest results, indicating that object features play the key role for recognition, and recognition with segmented images achieves better performance than without segmentation. Mixing background features with foreground features *does not* give higher recognition rates than FF alone. For images with roughly constant backgrounds (AF-CONST), the performance is almost the same as for images with foreground features only. It is intuitively obvious that classifying images with fixed backgrounds is as easy as classifying images with no background clutter at all. Finally, the ROC curves for AF-RAND are the lowest, which shows that objects with uncorrelated backgrounds are harder to recognize.

Fig. 20 shows ROC curves for a setup where the training set has different types of backgrounds and the test set has its original background. We can observe that training on AF or AF-RAND and testing on AF gives the highest results. Thus, even under randomly changed training backgrounds, the SVM can find decision boundaries that generalize well to the original training set. Training on FF or AF-CONST and testing on AF gives lower results, most likely because the lack of clutter in FF set and the monotonous backgrounds in AF-CONST cause the SVM to overfit the training set. By contrast, varying the object background during training, even by random shuffling, results in a more robust classifier.

Finally, Fig. 21 shows ROC curves for a situation where the training set has the original backgrounds and the test set has different types of backgrounds. Testing on FF gives better results than when testing on the original dataset AF, while testing on AF-RAND gives much worse results. Thus, when the test set is "easier" than the training one, performance improves, and when it is "harder," the performance drops. This is consistent with the results of Fig. 20, where training on the "harder" sets AF or AF-RAND gave much better results than training on the "easier" sets FF and AF-CONST. Next, we can observe that the results of testing on AF-CONST are better than those of testing on AF.

Based on our evaluation of the role of background features in bag-of-keypoints classification, we can venture two general observations. First, while the backgrounds in most available datasets have non-negligible correlations with the foreground objects, using both foreground and background features for learning and recognition does not result in better performance for our method. In our experimental setting, the recognition problem is easier in the absence of clutter. This highlights the limitations as evaluation platforms of datasets with simple backgrounds, such as ETH-80 [32] and COIL-100 [44]: Based on the insights of our evaluation, high performance on these datasets would not necessarily mean high performance on real images with varying backgrounds. Second, when the statistics of the test set are unknown at training time, it is usually beneficial to pick the most difficult training set available, since the presence of varied backgrounds during training helps to improve the generalization ability of the classifier.

6 Discussion

In this paper we have investigated the performance of a kernel-based discriminative approach for texture and object category classification using local image features. Results on challenging datasets have

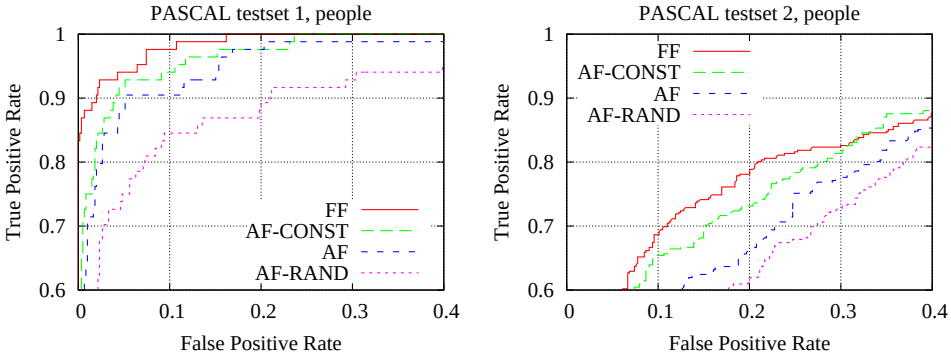


Figure 19: ROC curves of our method on the PASCAL challenge. The method is trained and tested with four combinations of the foreground features with different types of background. The same type of background is used for training and testing.

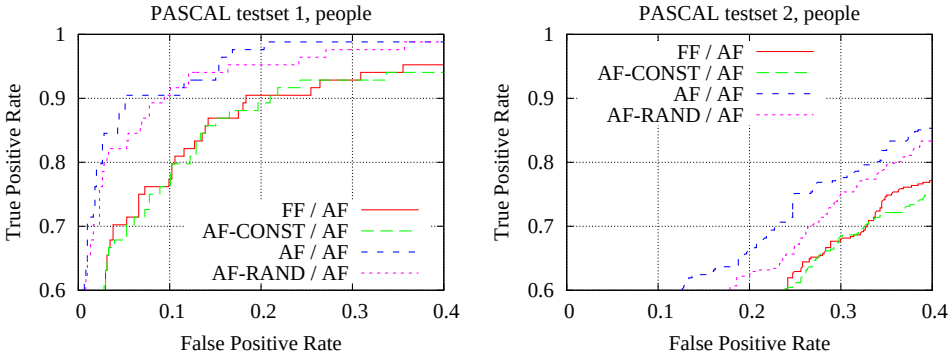


Figure 20: ROC curves of our method on the PASCAL challenge. The method is trained with four combinations of the foreground features with different types of background, and tested on the original test set of the PASCAL challenge.

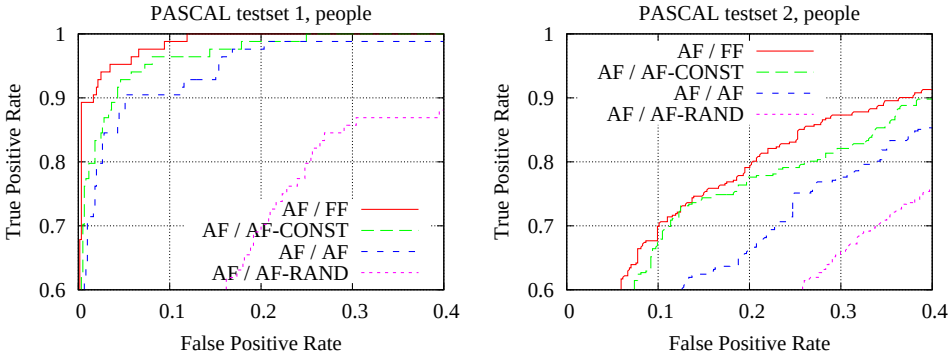


Figure 21: ROC curves of our method on PASCAL challenge. The method is trained on the original training set of PASCAL challenge, and tested on four combinations of the foreground features with different types of background.

shown that surprisingly high levels of performance can be achieved with an image representation that is essentially an orderless histogram. This is true not only for texture images, which are clutter-free and relatively statistically homogeneous, but also for object images, even in the case of completely uncorrelated backgrounds.

One of the contributions of our paper is a comprehensive evaluation of multiple keypoint detector types, levels of geometric invariance, feature descriptors, and classifier kernels. This evaluation has revealed several general trends, which should prove useful for computer vision practitioners designing high-accuracy recognition systems for real-world applications. For example, we show that to achieve the best possible performance, it is necessary to use a combination of several detectors and descriptors together with a classifier that can make effective use of the complementary types of information contained in them. Also, we show that using local features with the highest possible level of invariance usually does not yield the best performance. Thus, a practical recognition system should seek to incorporate multiple types of complementary features, as long as their local invariance properties do not exceed the level absolutely required for a given application.

In testing our method on four texture and five object databases, we have followed an evaluation regime far more rigorous than that of most other comparable works. In fact, our evaluation of multiple texture recognition methods highlights the danger of the currently widespread practice of developing and testing a recognition method with only one or two databases in mind. For example, methods tuned to achieve high performance on the CUReT database (e.g., the VZ method) have weaker performance on other texture databases, such as UIUCTex, and vice versa, methods tuned to UIUCTex and Brodatz (e.g., the Lazebnik method) perform poorly on CUReT.

Another contribution of our paper is the evaluation of the influence of background features. It shows the pitfalls of training on datasets with uncluttered or highly correlated backgrounds, since this causes overfitting and yields disappointing results on test sets with more complex backgrounds. On the other hand, training a method on a harder dataset typically improves the generalization power of a classifier and does not hurt performance even on a clutter-free dataset.

Future research should focus on designing improved feature representations. We believe that significant performance gains are still to be realized from developing more effective detectors and descriptors, for example for representing shape. Another promising area is the development of hybrid sparse/dense representations. For example, the recent successes of the novel feature extraction schemes of [10, 28] suggest that increasing the density and redundancy of local feature sets may be beneficial for recognition. Additional research directions include designing kernels that incorporate geometrical relations between local features (see [22] for preliminary work along these lines) and feature selection methods that can separate foreground from background. In the longer term, successful category-level object recognition and localization is likely to require more sophisticated models that capture the 3D shape of real-world object categories as well as their appearance. In the development of such models and in the collection of new datasets, simpler bag-of-keypoints methods can serve as effective baselines and calibration tools.

Acknowledgments

This research was supported by the French project MoViStaR under the program "ACI Masse de données", the European project LAVA, the European Network of Excellence PASCAL, and the UIUC-CNRS-INRIA collaboration agreement. J. Zhang was funded by an ACI postdoctoral fellowship and M. Marszałek by the INRIA student exchange program and a grant from the European Community under the Marie-Curie project VISITOR. S. Lazebnik was funded in part by Toyota and National Science Foundation grants IIS-0308087 and IIS-0312438. We also thank J. Ponce for discussions, M. Varma and A. Zisserman for providing the subset of the CUReT dataset used in their paper, E. Hayman for explaining the implementation details of their method, and Y. Rubner for making his implementation of EMD publicly available on the web.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *European Conference on Computer Vision*, volume 4, pages 113–130, 2002.

[2] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 26–33, 2005.

[3] P. Brodatz. *Textures: A Photographic Album for Artists and Designers*. Dover, New York, 1966.

[4] B. Caputo, C. Wallraven, and M.-E. Nilsback. Object categorization via local kernels. In *International Conference on Pattern Recognition*, volume 2, pages 132–135, 2004.

[5] O. Chapelle, P. Haffner, and V. Vapnik. Support vector machines for histogram-based image classification. *IEEE Transactions on Neural Networks*, 10(5):1055–1064, 1999.

[6] F.S. Cohen, Z. Fan, and M.A.S. Patel. Classification of rotated and scaled textured images using Gaussian Markov field models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):192–202, 1991.

[7] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[8] O.G. Cula and K.J. Dana. Compact representation of bidirectional texture functions. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 1041–1047, 2001.

[9] K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink. Reflectance and texture of real world surfaces. *ACM Transactions on Graphics*, 18(1):1–34, 1999.

[10] T. Deselaers, D. Keysers, and H. Ney. Discriminative training for object recognition using image patches. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 157–162, 2005.

[11] T. Deselaers, D. Keysers, and H. Ney. Improving a discriminative approach to object recognition using image patches. In *DAGM*, pages 326–333, 2005.

[12] G. Dorkó and C. Schmid. Object class recognition using discriminative local features. Technical Report RR-5497, INRIA - Rhône-Alpes, February 2005.

[13] J. Eichhorn and O. Chapelle. Object categorization with SVM: kernels for local features. Technical report, Max Planck Institute for Biological Cybernetics, Tuebingen, Germany, 2004.

[14] M. Everingham, A. Zisserman, C. Williams, L. Van Gool, et al. The 2005 PASCAL visual object classes challenge. In F. d'Alche Buc, I. Dagan, and J. Quinonero, editors, *Selected Proceedings of the first PASCAL Challenges Workshop*. LNAI, Springer, 2006. http://www.pascal-network.org/challenges/VOC/voc/index.html.

[15] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop on Generative-Model Based Vision*, 2004.

[16] L. Fei-Fei and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 524–531, 2005.

[17] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

[18] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 264–271, 2003.

[19] M. Fischler and R. Elschlager. The representation and matching of pictorial structures. *IEEE Transactions on Computers*, 22(1):67–92, 1973.

[20] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nyström method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):1–12, 2004.

[21] J. Gårding and T. Lindeberg. Direct computation of shape cues using scale-adapted spatial derivative operators. *International Journal of Computer Vision*, 17(2):163–191, 1996.

[22] K. Grauman and T. Darrell. Efficient image matching with distributions of local invariant features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 627–634, 2005.

[23] K. Grauman and T. Darrell. Pyramid match kernels: Discriminative classification with sets of image features. In *International Conference on Computer Vision*, volume 2, pages 1458–1465, 2005.

[24] E. Hayman, B. Caputo, M. Fritz, and J.-O. Eklundh. On the significance of real-world conditions for material classification. In *European Conference on Computer Vision*, volume 4, pages 253–266, 2004.

[25] F. Jing, M. Li, H.-J. Zhang, and B. Zhang. Support vector machines for region-based image retrieval. In *IEEE International Conference on Multimedia and Expo*, 2003.

[26] A. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3D scenes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(5):433–449, 1999.

[27] B. Julesz. Textons, the elements of texture perception and their interactions. *Nature*, 290:91–97, 1981.

[28] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition. In *International Conference on Computer Vision*, volume 1, pages 604–610, 2005.

[29] D. Larlus, G. Dorkó, and F. Jurie. Création de vocabulaires visuels efficaces pour la catégorisation d'images. In *Reconnaissance des Formes et Intelligence Artificielle*, 2006.

[30] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *British Machine Vision Conference*, volume 2, pages 959–968, 2004.

[31] S. Lazebnik, C. Schmid, and J. Ponce. A sparse texture representation using local affine regions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1265–1278, 2005.

[32] B. Leibe and B. Schiele. Analyzing appearance and contour-based methods for object categorization. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 409–415, 2003. http://www.mis.informatik.tu-darmstadt.de/Research/Projects/categorization/eth80-db.html.

[33] T. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. *International Journal of Computer Vision*, 43(1):29–44, 2001.

[34] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):79–116, 1998.

[35] X. Llado, J. Marti, and M. Petrou. Classification of textures seen from different distances and under varying illumination direction. In *IEEE International Conference on Image Processing*, volume 1, pages 833–836, 2003.

[36] D. Lowe. Distinctive image features form scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[37] S. Lyu. Mercer kernels for object recognition with local features. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 223–229, 2005.

[38] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):837–842, 1996.

[39] J. Mao and A. Jain. Texture classification and segmentation using multiresolution simultaneous autoregressive models. *Pattern Recognition*, 25(2):173–188, 1992.

[40] A. McCallum and K. Nigam. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, pages 41–48, 1998.

[41] K. Mikolajczyk and C.Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, 2005.

[42] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *European Conference on Computer Vision*, volume 1, pages 128–142, 2002.

[43] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[44] S.A. Nene, S.K. Nayar, and H. Murase. Columbia object image library (COIL-100). Technical Report CUCS-006-96, Columbia University, 1996. http://www1.cs.columbia.edu/CAVE/research/softlib/coil-100.html.

[45] W. Niblack, R. Barber, W. Equitz, M. Fickner, E. Glasman, D. Petkovic, and P. Yanker. The QBIC project: Querying images by content using color, texture and shape. In *SPIE Conference on Geometric Methods in Computer Vision II*, 1993.

[46] K. Nigam, J. Lafferty, and A. McCallum. Using maximum entropy for text classification. In *IJCAI Workshop on Machine Learning for Information Filtering*, pages 61–67, 1999.

[47] M.-E. Nilsback and B. Caputo. Cue integration through discriminative accumulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 578–585, 2004.

[48] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. In *European Conference on Computer Vision*, volume 2, pages 71–84, 2004.

[49] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *International Conference on Machine Learning*, pages 727–734, 2000.

[50] M. Pontil and A. Verri. Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(6):637–646, 1998.

[51] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica, T. Tuytelaars, and L. Van Gool. Modeling scenes with local descriptors and latent aspects. In *International Conference on Computer Vision*, volume 2, pages 883–890, 2005.

[52] Y. Rubner, C. Tomasi, and L. Guibas. The Earth Mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

[53] B. Schiele and J. Crowley. Recognition without correspondence using multidimensional receptive field histograms. *International Journal of Computer Vision*, 36(1):31–50, 2000.

[54] B. Schölkopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, Cambridge, MA, 2002.

[55] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering objects and their location in images. In *International Conference on Computer Vision*, volume 1, pages 370–378, 2005.

[56] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, volume 2, pages 1470–1477, 2003.

[57] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. In *European Conference on Computer Vision*, volume 3, pages 255–271, 2002.

[58] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 691–698, 2003.

[59] C. Wallraven, B. Caputo, and A. Graf. Recognition with local features: the kernel recipe. In *International Conference on Computer Vision*, volume 1, pages 257–264, 2003.

[60] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2101–2109, 2000.

[61] J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan. Categorizing nine visual classes using local appearance descriptors. In *ICPR Workshop on Learning for Adaptable Visual Systems*, 2004.

[62] J. Wu and M. J. Chantler. Combining gradient and albedo data for rotation invariant classification of 3D surface texture. In *International Conference on Computer Vision*, volume 2, pages 848–855, 2003.

[63] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classifcation of texture and object categories: An in-depth study. Technical Report RR-5737, INRIA Rhône-Alpes, November 2005. http://lear.inrialpes.fr/pubs/2005/ZMLS05.