A discriminative framework for texture and object recognition using local image features

Svetlana Lazebnik, Cordelia Schmid, Jean Ponce

A Discriminative Framework for Texture and Object Recognition Using Local Image Features

Svetlana Lazebnik[1], Cordelia Schmid[2], and Jean Ponce[1]

[1] Beckman Institute, University of Illinois
405 N. Mathews Avenue, Urbana, IL 61801, USA
{slazebni,jponce}@uiuc.edu
[2] INRIA Rhône-Alpes
665 Avenue de l'Europe, 38330 Montbonnot, France
cordelia.schmid@inrialpes.fr

Abstract. This chapter presents an approach for texture and object recognition that uses scale- or affine-invariant local image features in combination with a discriminative classifier. Textures are represented using a visual dictionary found by quantizing appearance-based descriptors of local features. Object classes are represented using a dictionary of composite *semi-local parts*, or groups of nearby features with stable and distinctive appearance and geometric layout. A discriminative maximum entropy framework is used to learn the posterior distribution of the class label given the occurrences of parts from the dictionary in the training set. Experiments on two texture and two object databases demonstrate the effectiveness of this framework for visual classification.

1 Introduction

By analogy with a text document, an image can be viewed as a collection of parts or "visual words" drawn from a "part dictionary." This parallel has been exploited in recent *bag-of-keypoints* approaches to visual categorization [6, 27], unsupervised discovery of visual "topics" [24], and video retrieval [23]. More generally, representations based on *local image features*, or salient regions extracted by specialized interest operators, have shown promise for recognizing textures [13], different views of the same object [9, 22], and different instances of the same object class [1, 7, 8, 26]. For textures, appearance-based descriptors of salient local regions are clustered to form characteristic texture elements, or *textons*. For objects, such clusters can also play the role of generic object parts. In our own previous work [15], we have introduced a more expressive representation based on composite *semi-local parts*, defined as geometrically stable configurations of multiple local regions that are robust against approximately rigid deformations and intra-class variations.

In this chapter, we present an approach to visual categorization that first constructs a texture or object representation based on a dictionary of textons or parts, and then learns a discriminative classifier that can effectively distinguish assemblies of parts or occurrence patterns of textons characteristic of different

classes. For the classification step, we adopt a discriminative *maximum entropy* framework, which has been used successfully for text document classification [3, 21] and image annotation [10]. This framework has several characteristics that make it attractive for visual categorization as well: It directly models the posterior distribution of the class label given the image, leading to convex (and tractable) parameter estimation; moreover, classification is performed in a true multi-class fashion, requiring no distinguished background class. Because the maximum entropy framework makes no independence assumptions, it offers a principled way of combining multiple kinds of features (e.g., keypoints produced by different detectors), as well as inter-part relations, into the object representation. While maximum entropy has been widely used in the computer vision for *generative* tasks, e.g., modeling of images as Markov random fields [28], where it runs into issues of intractability for learning and inference, it can be far more efficient for *discriminative* tasks. For example, Mahamud et al. [18] have used maximum entropy to combine multiple nearest-neighbor discriminators, and Keysers et al. [12] have applied it to digit recognition. In this chapter, we explore this framework in a part-based object categorization setting.

The rest of our presentation is organized as follows. We review in Section 2 the basics of *exponential models*, which arise from maximum entropy considerations. Sections 3 and 4 describe our approach to texture and object recognition, and Section 5 concludes with a summary and discussion of future directions. The research reported in this chapter has been previously published in [14].

2 The Maximum Entropy Framework

A discriminative maximum entropy approach seeks to estimate the posterior distribution of class labels given image features that matches the statistics of the features observed in the training set, and yet remains as uniform as possible. Intuitively, such a distribution properly reflects our uncertainty about making a decision given ambiguous or inconclusive image data. (By contrast, some generative methods, e.g., mixtures of Gaussians, tend to yield peaky or "overconfident" posterior distributions.) Suppose that we have defined a set of *feature functions* $f_k(I, c)$ that depend both on the image I and the class label c (the definitions of the specific feature functions used in our work will appear in Sections 3 and 4). To estimate the posterior of the class label given the features, we constrain the expected values of the features under the estimated distribution $P(c|I)$ to match those observed in the training set $\mathcal{T}$. The observed "average" value of feature f_k in the training set $\mathcal{T}$ is

$$\hat{f}_k = \frac{1}{|\mathcal{T}|} \sum_{I \in \mathcal{T}} f_k(I, c(I)) \,.$$

Given a particular posterior distribution $P(c|I)$, the expected value of f_k, taken with respect to the observed empirical distribution $P(I)$ over the training set, is

$$E[f_k] = \frac{1}{|\mathcal{T}|} \sum_{I \in \mathcal{T}} \sum_{c} P(c|I) f_k(I, c) \,.$$

We seek the posterior distribution that has the maximum *conditional entropy*

$$H = -\frac{1}{|\mathcal{T}|} \sum_{I \in \mathcal{T}} \sum_{c} P(c|I) \log P(c|I)$$

subject to the constraints $E[f_k] = \hat{f}_k$. It can be shown that the desired distribution has the *exponential form*

$$P(c|I) = \frac{1}{Z} \exp\left(\sum_k \lambda_k f_k(I, c) \right) , \tag{1}$$

where

$$Z = \sum_c \exp\left(\sum_k \lambda_k f_k(I, c) \right)$$

is the normalizing factor,[1] and the λ_k are parameters whose optimal values are found by maximizing the likelihood of the training data under the exponential model (1). This optimization problem is convex and the global maximum can be found using the improved iterative scaling (IIS) algorithm [3, 21]. At each iteration of IIS, we compute an update δ_k to each λ_k, such that the likelihood of the training data is increased. To do this, we bound $L(\lambda + \delta) - L(\lambda)$ from below by a positive function $F(\delta)$, and find the value of δ that maximizes this function. The derivation of updates is omitted here, but it can be shown [3, 21] that when the features are *normalized*, i.e., when $\sum_k f_k(I, c)$ is a constant S for all I and c, updates can be found efficiently in closed form:

$$\delta_k = \frac{1}{S} \left(\log \hat{f}_k - \log E_\lambda[f_k] \right) . \tag{2}$$

Because of the computational efficiency gained in this case, we use only normalized features in the present work.

Because of the form of (2), zero values of $\hat{f}_k$ cause the optimization to fail, and low values cause excessive growth of the weights. This is a symptom of one of the biggest potential pitfalls of the maximum entropy framework: overfitting. When the training set is small, the observed averages may deviate significantly from the "true" expectations, leading to a poor estimate of the posterior distribution. This problem can be alleviated by adding a zero-mean Gaussian prior on the weights [21]. However, in our experiments, we have achieved better results with a basic IIS setup where simple transformations of the feature functions are used to force expectations away from zero. Specifically, for all the feature functions defined in Sections 3 and 4, we use the standard Laplace smoothing, i.e.,

[1] Note that Z involves only a sum over the classes, and thus can be computed efficiently. If we were modeling the distribution of features given a class instead, Z would be a sum over the exponentially many possible combinations of feature values — a major source of difficulty for a generative approach. By contrast, the discriminative approach described here is more related to logistic regression. It is easy to show that (1) yields binary logistic discrimination in the two-class case.

adding one to each feature value and renormalizing. To simplify the subsequent presentation, we will omit this operation from all feature function definitions.

We close this section with a note concerning the technique we use to design feature functions. Instead of directly defining class-dependent features $f_k(I, c)$, it is much more convenient to obtain them from a common pool of *class-independent* features $g_k(I)$, as follows:

$$f_{d,k}(I, c) = \begin{cases} g_k(I) \text{ if } c = d, \\ 0 \quad \text{otherwise.} \end{cases}$$

Then we have

$$P(c|I) = \frac{1}{Z} \exp\left(\sum_{d,k} \lambda_{d,k} f_{d,k}(I, c)\right) = \frac{1}{Z} \exp\left(\sum_k \lambda_{c,k} g_k(I)\right) .$$

Thus, "universal" features g_k become associated with class-specific weights $\lambda_{c,k}$. All our feature functions will be defined in this way. Note, however, that the exponential framework also allows completely different features for representing each class.

3 Texture Recognition

In this section, we describe the application of the maximum entropy framework to texture recognition. Section 3.1 describes our texton-based representation, and Section 3.2 discusses experiments on two large collections of texture images, the Brodatz database [4] and the UIUC database [13].

3.1 Feature Functions

For texture recognition, we use the sparse representation introduced in our earlier work [13], where the locations and shapes of salient image regions are found by a specialized keypoint detector. We use either a scale- or an affine-invariant detector (returning circular and elliptical regions, respectively), depending on the degree of invariance required by a particular database. Next, the extracted regions serve as domains of support for computing appearance-based descriptors (the specific choices of detectors and descriptors used in our experiments are discussed in Section 3.2). After descriptors have been extracted from the training set, a texton dictionary is formed by clustering them, and associating each cluster center with a discrete texton label. Finally, each descriptor from a new image is assigned the label of the closest cluster center.

The next step is to define the feature functions for the exponential model. For text classification, Nigam et al. [21] use scaled counts of word occurrences in a document. By analogy, we define feature functions based on texton frequencies:

$$g_k(I) = \frac{N_k(I)}{\sum_{k'} N_{k'}(I)} ,$$

where $N_k(I)$ is the number of times texton label k occurs in the image I. To enrich the feature set, we also define functions $g_{k,\ell}$ that encode the probability of co-occurrence of pairs of labels at nearby locations. Let $k \diamond \ell$ denote the event that a region labeled ℓ is adjacent to a region labeled k. Specifically, we say that $k \diamond \ell$ if the center of ℓ is contained in the neighborhood obtained by "growing" the shape (circle or ellipse) of the kth region by a constant factor (4 in the implementation). Let $N_{k\diamond\ell}(I)$ denote the number of times the relation occurs in the image I, and define

$$g_{k,\ell}(I) = \frac{N_{k\diamond\ell}(I)}{\sum_{k',\ell'} N_{k'\diamond\ell'}(I)} \,.$$

An image model incorporating co-occurrence counts of pairs of adjacent labels is a counterpart of a *bigram language model* that estimates the probabilities of two-word strings in natural text. Just as in language modeling, we must deal with sparse probability estimates due to many relations receiving extremely low counts in the training set. Thus, we are led to consider smoothing techniques for probability estimates [5]. One of the most basic techniques, interpolation with marginal probabilities, leads to the following modified definition of the co-occurrence features:

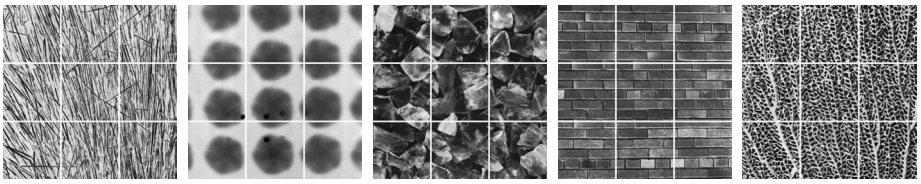
$$\tilde{g}_{k,\ell}(I) = (1-\alpha) g_{k,\ell}(I) + \alpha \Big(\sum_{\ell'} g_{k,\ell'}(I) \Big) \Big(\sum_{k'} g_{k',\ell}(I) \Big) \,,$$

where α is a constant (0.1 in our implementation). Informally, a co-occurrence relation $k \diamond \ell$ should have higher probability if both k and ℓ occur frequently in samples of the class, and if they each have many neighbors.

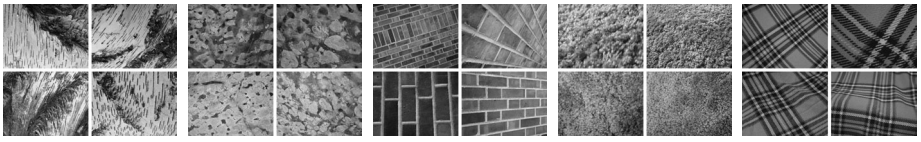
While smoothing addresses the problem of unreliable probability estimates, we are still left with millions of possible co-occurrence relations, and it is necessary to use feature selection to reduce the model to a manageable size. Possible feature selection techniques include greedy selection based on increase of likelihood under the exponential model [3], mutual information [7, 21] and likelihood ratio [7]. However, since more frequently occurring relations yield more reliable estimates, we have chosen a simpler likelihood-based scheme: For each class, we find a fixed number of relations that have the highest probability in the training set, and then combine them into a global "relation dictionary."

3.2 Experimental Results

In this section, we show classification results on the Brodatz database (999 images: 111 classes, 9 samples per class) [4] and the UIUC database (1000 images: 25 classes, 40 samples per class) [13]. Figure 1 shows examples of images from the two databases. For the Brodatz database, we use a scale-invariant Laplacian detector [16], which finds salient blob-like circular regions in an image. This level of invariance is sufficient for the Brodatz database, which does not feature any significant geometric deformations between different samples from the same class. By contrast, the UIUC database contains arbitrary rotations, perspective



Brodatz texture database



UIUC texture database

Fig. 1. Examples of five classes each from the Brodatz database (top) and the UIUC database (bottom). The UIUC database is publicly available at `http://www-cvr.ai.uiuc.edu/ponce_grp/data`.

distortions and non-rigid deformations. This greater degree of geometric variability requires a greater degree of invariance in the low-level features. Therefore, we process the UIUC database with an affinely adapted version of the Laplacian detector, which returns elliptical regions. In both cases, the appearance of the detected regions is represented using SIFT descriptors [17]. The SIFT descriptor consists of gradient orientation histograms 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 128-dimensional feature vector. We have chosen to use SIFT descriptors because of their impressive performance in a recent comparative evaluation [20].

To form the texton dictionary, we run K-means clustering on a randomly selected subset of all training descriptors. To limit the memory requirements of the K-means algorithm, we cluster each class separately and concatenate the resulting textons. We find $K = 10$ and $K = 40$ textons per class for the Brodatz and the UIUC database, respectively, resulting in dictionaries of size 1110 and 1000. For co-occurrence relations, we select $10K$ features per class; because the relations selected for different classes sometimes coincide, the total number of $g_{k,\ell}$ features is slightly less than ten times the total number of textons.

Table 1 shows a comparison of classification rates obtained using various methods on the two databases. All the rates are averaged over 10 runs with different randomly selected training subsets; standard deviations of the rates are also reported. The training set consists of 3 (resp. 10) images per class for the Brodatz (resp. UIUC) database. The first row shows results for a popular baseline method using nearest-neighbor classification of texton histograms with the χ^2 distance (for an example of such an approach, see, e.g., [25]). The second row

	Brodatz database		UIUC database	
	Mean (%)	Std. dev.	Mean (%)	Std. dev.
χ^2	83.09	1.18	94.25	0.59
Naive Bayes	85.84	0.90	94.08	0.67
Exp. g_k	87.37	1.04	97.41	0.64
Exp. $g_{k,\ell}$	75.20	1.34	92.40	0.93
Exp. $g_k + g_{k,\ell}$	83.44	1.17	97.19	0.57
Exp. $\tilde{g}_{k,\ell}$	80.51	1.09	95.85	0.62
Exp. $g_k + \tilde{g}_{k,\ell}$	83.36	1.14	97.09	0.47

Table 1. Texture classification results (see text).

shows results for a Naive Bayes baseline using the *multinomial event model* [19]:

$$P(I|c) = \prod_k P(k|c)^{N_k(I)} ,$$

where $P(k|c)$ is given by the frequency of texton k in the training images for class c. The results for the two baseline methods on the Brodatz database are comparable, though Naive Bayes has a potential advantage over the χ^2 method, since it does not treat the training samples as independent prototypes, but combines them in order to compute the probabilities $P(k|c)$. This may help to account for the slightly better performance of Naive Bayes on the Brodatz database. The third and fourth rows show results for exponential models based on individual g_k (textons only) features and $g_{k,\ell}$ (relations only) features, respectively, and the fifth row shows results for the exponential model with both kinds of features combined. For both databases, the texton-only exponential model performs much better than the two baseline methods; the relations-only models are inferior to the baseline. Interestingly, combining textons and relations does not improve performance. To test whether this is due to overfitting, we compare performance of the $g_{k,\ell}$ features with the smoothed $\tilde{g}_{k,\ell}$ features (last two rows). While the smoothed features do perform better, combining them with textons-only features once again does not bring any improvement. Thus, texton-only features clearly supercede the co-occurrence relations.

To get a more detailed look at the performance of the exponential model, refer to Figure 2, which shows the histograms of classification rates achieved by the parts-only exponential model for individual classes. With this model, 100% recognition rate is achieved by 61 classes from the Brodatz database and by 8 classes from the UIUC database. The distribution of classification rates, in particular for the Brodatz database, suggests another reason (besides overfitting) for the lack of improvement afforded by co-occurrence features. Namely, most classes in the database can be represented quite well without taking texton co-occurrences into account, while a few are either extremely nonhomogeneous or

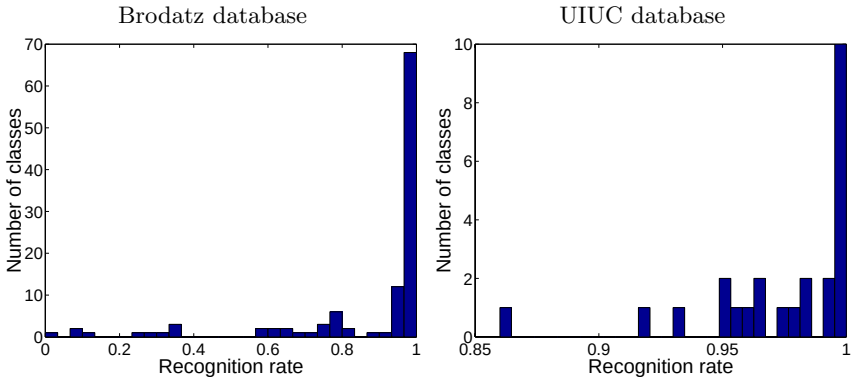


Fig. 2. Histograms of classification rates for the exponential parts-only model for the Brodatz database (left) and the UIUC database (right).

extremely perceptually similar to another class. Consequently, adding relations to the exponential model cannot improve the recognition of either the "easy" or the "difficult" classes.

Overall, the g_k exponential model performs the best for both texture databases. For the Brodatz database, our result of 87.37% is comparable to the rate of 87.44% reported in [13]. Note, however, that the result of [13] was obtained using a combination of appearance- and shape-based features. In our case, we use only appearance-based features, so we get as much discriminative power with a weaker representation. For the UIUC database, our result of 97.41% exceeds the highest rate reported in [13], that of 92.61%.

4 Object Recognition

In this section, we describe our approach to object recognition using *semi-local parts* and present results of experiments on two challenging datasets: the CalTech dataset [8] consisting of airplanes, cars, faces, and motorbikes; and a bird dataset that we have collected, consisting of images of six different species.

4.1 Semi-Local Parts

For our texture recognition experiments, Laplacian region detectors have proven to be successful. However, we have found them to be much less satisfactory for detecting object parts with complex internal structures, e.g., eyes, wheels, heads, etc. Instead, for object recognition, we have implemented the scale-invariant detector of Jurie and Schmid [11], which finds salient circular configurations of edge points, and is robust to clutter and texture variations inside the regions. Just as in Section 3, the appearance of the extracted regions is represented using SIFT descriptors.

For each object class, we construct a dictionary of composite *semi-local parts* [15], or groups of several nearby regions whose appearance and spatial configuration occurs repeatably in the training set. The key idea is that consistent occurrence of (approximately) rigid groups of simple features in multiple images is very unlikely to be accidental, and must thus be a strong cue for the presence of the object. Semi-local parts are found in a *weakly supervised* manner, i.e., from cluttered, unsegmented training images, via a direct search for visual correspondence.[2] The intractable problem of simultaneous alignment of multiple images is reduced to pairwise matching: *Candidate parts* are initialized by matching several training pairs and then *validated* against additional images.

The key operation of two-image matching is accomplished efficiently with the help of strong appearance (descriptor similarity) and geometric consistency constraints. Specifically, initial constraints on descriptor similarity are used to create a short list of *potential matches* for each region in the other image; semi-local neighborhood constraints [9, 23] reduce the set of all potential matches even further. Then, starting from the smallest possible *seed group* of nearby matches that allows us to estimate an aligning transformation, we conduct a greedy search for additional geometrically and photometrically consistent matches lying in the neighborhood of the current group. The aligning transformation can be scaling, similarity, or affine. Originally, we have introduced semi-local parts in the context of an affine alignment model [15]; however, for the two databases used in this chapter, scale and translation invariance are sufficient. Note that in the implementation, we treat all transformation groups within the same computational framework. Namely, we use linear least squares to estimate an affine alignment between the two groups of regions, and then enforce additional geometric constraints by rejecting any alignment that deviates too much from the desired model. In particular, for a scale-and-translation model, we reject transformations that include too much skew, rotation, and anisotropic scaling. The correspondence search terminates when the residual of the transformation grows too large, or when no further consistent matches can be found. Note that the number of regions in the correspondence (the size of the part) is determined automatically as a result.

In existing literature, similar procedures for growing groups of matches based on geometric and appearance consistency have been successfully applied to the recognition of the same object instance in multiple views [9]; one of the key insights of our earlier work [15] is that such procedures are also quite effective for building models of object classes with substantial intra-class variation. Because of the strong geometric and photometric consistency constraints that must be satisfied by semi-local parts, they are much more discriminative than atomic parts, and much less likely to give rise to false detections.

A detected instance of a candidate part in a validation image may have multiple regions missing because of occlusion, failure of the keypoint detector, etc. We define the *repeatability* $\rho_k(I)$ of a detected instance of part k in image

[2] See [2] for another recent approach to object recognition that shares our emphasis on geometric correspondence.

I as the number of regions in that instance normalized by the total number of regions in that part. If no instances of part k are detected at all, we have $\rho_k(I) = 0$, and if several instances are detected, we simply select the one with the highest repeatability. This implicitly assumes that an object can contain at most one instance of each part. In the future, we plan to improve our feature representation to allow for multiple detected instances of the same part. This would allow us to perform more accurate localization for classes such as cars (which have two wheels) or faces (which have two eyes).

After recording the repeatability values for a given part in all positive and negative validation images, we compute a *validation score* for the part by taking the χ^2 distance between h_p, the histogram of repeatabilities of the part over the positive class, and h_n, the histogram of its repeatabilities in all the negative images (for examples of these histograms, see Figures 5(b) and 7(b)). The χ^2 distance is defined as follows:

$$d(h_p, h_n) = \frac{1}{2} \sum_{b=1}^{B} \frac{\big(h_p(b) - h_n(b)\big)^2}{h_p(b) + h_n(b)} ,$$

where B is the number of bins (discrete repeatability levels) in the histograms, and $h_p(b)$ (resp. $h_n(b)$) is the proportion of all part detections in positive (resp. negative) images falling into the bin with index b. The validation score can range from 1, when the two histograms have no overlap at all, to 0, when they are identical. A fixed number of highest-scoring parts is retained for each class, and their union forms our dictionary.

Finally, for each part k and each training image I, we compute a normalized feature function based on its repeatability:

$$g_k(I) = \frac{\rho_k(I)}{\sum_{k'} \rho_{k'}(I)} .$$

Just as in our texture recognition experiments, we also investigate whether, and to what extent, incorporating relations into the object representation improves classification performance. To this end, we define *overlap* relations between pairs of parts that belong to the same class. Let $\omega_{k,\ell}(I)$ be the overlap between detected instances of parts k and ℓ in the image I, i.e., the ratio of the intersection of the two parts to their union. This ratio ranges from 0 (disjoint parts) to 1 (coincident parts). Then we define

$$g_{k,\ell}(I) = \frac{\omega_{k,\ell}(I)}{\sum_{k',\ell'} \omega_{k',\ell'}(I)} .$$

The overlap relations are very flexible — in effect, they enforce only spatial coherence. This flexibility potentially allows us to deal with non-rigid and/or articulated objects. In the future, we plan to experiment with more elaborate relations that take into account the distance, relative scale, or relative orientations of the two parts [1]. Finally, it is important to note that we currently do not use feature selection techniques to reduce the number of overlap relations within

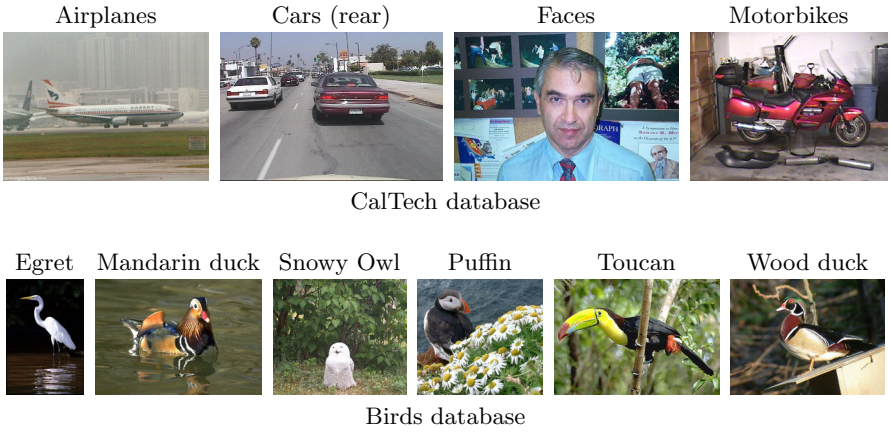


Fig. 3. One example image per class for the CalTech database (top) and the birds database (bottom). The birds database is publicly available at `http://www-cvr.ai.uiuc.edu/ponce_grp/data`.

the exponential model. Because of the small size of the part dictionaries used in the experiments presented in the next section (namely, 20 parts per class), the resulting number of overlap relations (190 per class) is quite manageable, unlike in our texture recognition experiments, where we had to contend with millions of potential co-occurrence relations.

4.2 Experimental Results

This section presents recognition results obtained on two multi-class object databases. The first is a subset of the publicly available CalTech database [8]. We have taken 300 images each from four classes: airplanes, rear views of cars, faces, and motorbikes (Figure 3, top). The second database, which we collected from the Web, consists of 100 images each of six different classes of birds: egrets, mandarin ducks, snowy owls, puffins, toucans, and wood ducks (Figure 3, bottom). For the CalTech database, 50 randomly chosen images per class are used for creating candidate parts. Each image is paired up to two others, for a total of 100 initialization pairs. Of the several hundred candidate parts yielded by this matching process, the 50 largest ones are retained for training and selection. Candidate parts are then matched against every image from another training set, which also contains 50 randomly chosen images per class, and 20 highest-scoring parts per class are retained to form the part dictionary. The repeatability results of the selected parts on this training set are also used as training data to estimate the parameters of the exponential model. Finally, the remaining 200 images per class make up the test set. We follow the same protocol for the bird dataset, except that 20 images per class are used for finding candidate parts, another 30 for part selection, and the remaining 50 for testing. Unlike the texture

recognition results of Section 3.2, the results of this section are not averaged over multiple splits of the databases because of the considerably larger computational expense involved in computing semi-local parts. With our current unoptimized MATLAB implementation, a single run through an entire object database (both training and testing) takes about a week.

Figures 5 and 7 illustrate training and part selection. As can be seen from the plots of validation scores for all selected parts, the quality of part dictionaries found for different classes varies widely. Extremely stable, salient parts are formed for faces, motorbikes, and ducks. The classes with the weakest parts are airplanes for the CalTech database and egrets for the bird database. Both airplanes and egrets lack characteristic texture, and often appear against busy backgrounds that generate a lot of detector responses (buildings, people, and airport machinery in case of planes, or grass and branches in case of egrets). In addition, both egrets and airplanes are "thin" objects, so the local regions that overlap the object also capture a lot of background. Thus, the SIFT descriptors computed over these regions end up describing mostly clutter. To alleviate this problem, we plan to experiment with alternative descriptors that capture the shape of the edges close to the boundary of the scale-invariant regions [11], as opposed to the internal texture, as the SIFT descriptor does. Note that our part selection framework is suitable for choosing between semi-local parts based on different descriptors, since it abstracts from the low-level details of matching (i.e., how appearance similarity is computed, what aligning transformation is used, or how the correspondence search is performed), and looks only at the end result of the matching on the training set (i.e., how repeatable the resulting parts are, and whether they can be used to distinguish between positive and negative examples for a given class).

The parts obtained for classes other than airplanes and egrets have higher scores and capture much more salient object features. Interestingly, though, for cars, even the highest-scoring part includes spurious background detections along the horizontal line at the eye level of the image. This comes from the relative visual monotony of the car class: all the rear views of cars were apparently captured through the windshield by a person in the front seat. Thus, the "horizon" formed by the converging sides of the road is approximately in the same location in all the images, and the scenery at the roadside (trees, buildings) gives rise to a lot of features in stable positions that are consistently picked up by the matching procedure.

Tables 2 and 3 show classification performance of several methods with 20 parts per class. The first column of the tables shows the performance of a baseline Naive Bayes approach with likelihood given by

$$P(I|c) = \prod_k P\left(\rho_k(I)|c\right) \ .$$

The distributions $P\left(\rho_k|c\right)$ are found by histogramming the repeatabilities of part k on all training images from class c. Note that we take into account the repeatability of parts on images from *all* classes, not only the class which they

CalTech database	Naive Bayes	Exp. parts	Exp. relations	Exp. parts & relations
Airplanes	98.0	88.0	78.0	87.5
Cars (rear)	95.5	99.5	90.5	99.5
Faces	96.5	98.5	96.5	98.0
Motorbikes	97.5	99.5	83.0	99.5
All classes	96.88	96.38	87.0	96.13

Table 2. Classification rates for the CalTech database using 20 parts per class.

Birds database	Naive Bayes	Exp. parts	Exp. relations	Exp. parts & relations
Egret	68	90	72	88
Mandarin	66	90	66	90
Snowy owl	66	98	52	96
Puffin	88	94	94	94
Toucan	88	82	82	82
Wood duck	96	100	86	100
All classes	78.67	92.33	75.33	91.67

Table 3. Classification rates for the birds database using 20 parts per class.

describe. Roughly speaking, we expect $P(\rho_k(I)|c)$ to be high if part k describes class c and $\rho_k(I)$ is high, or if part k *does not* describe class c and $\rho_k(I)$ is low or zero. Thus, to conclude that an object from class c is present in the image, we not only have to observe high-repeatability detections of parts from class c, but also low-repeatability detections of parts from other classes. The exponential model, which encodes the same information in its feature functions, also uses this reasoning.

The second (resp. third, fourth) columns of Tables 2 and 3 show the classification performance obtained with exponential models using the g_k features only (resp. the $g_{k,\ell}$ only, g_k and $g_{k,\ell}$ combined). For the CalTech database, the Naive Bayes and the exponential parts-only models achieve very similar results, though under the exponential model, airplanes have a lower classification rate, which is intuitively more satisfying given the poor part dictionary for this class. Note that our classification accuracy of over 96% on the four CalTech classes is comparable to other recently published results [6, 7]. For the bird database, the exponential model outperforms Naive Bayes; for both databases, relations-only features alone perform considerably worse than the parts-only features, and combining parts-based with relation-based features brings no improvement. Figure 4

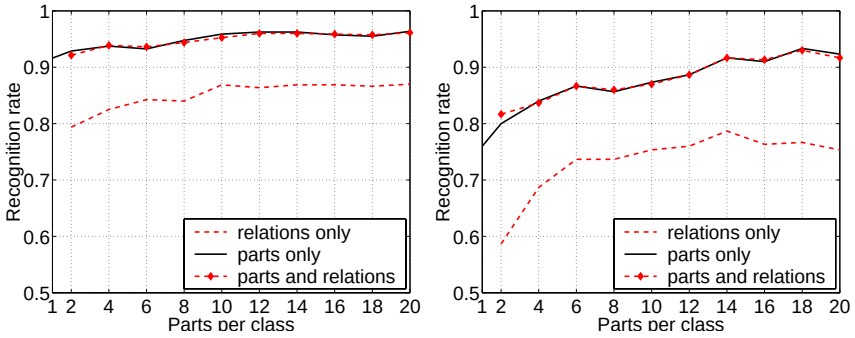


Fig. 4. Classification rate (exp. parts) as a function of dictionary size: CalTech database (left), birds database (right). For the CalTech database, because three of the four classes have extremely strong and redundant parts, performance increases very little as more parts are added. For the bird database, diminishing returns set in as progressively weaker parts are added.

shows a plot of the classification rate for the exponential model as a function of part dictionary size. Note that the curves are not monotonic — adding a part to the dictionary can decrease performance. This behavior may be an artifact of our scoring function for part selection, which is not directly related to classification performance. In the future, we plan to experiment with part selection based on increase of likelihood under the exponential model [3].

Though we did not conduct a quantitative evaluation of localization accuracy, the reader may get a qualitative idea by examining Figures 6 and 8, which show examples of part detection on several test images. A poorer part vocabulary for a class tends to lead to poorer localization quality, though this is not necessarily reflected in lower classification rates. Specifically, an object class represented by a relatively poor part vocabulary may still achieve a high classification rate, provided that parts for other classes do not generate too many false positives on images from this class. The second airplane example in Figure 6 is a good illustration of this phenomenon: only three airplane parts are detected in this image, yet the airplane is recognized correctly since the image does not contain enough clutter to generate false detections of parts from other classes.

Perhaps the most suprising finding of our experiments is that inter-part relations do not improve classification performance. From examining the part detection examples in Figures 6 and 8, it seems intuitively clear that the pattern of overlap of different part instances encodes useful information: the part detections that lie on the object tend to be clustered close together, while false detections are frequently scattered all over the image. At this stage, we conjecture that the overlap information may be more useful for localization than for recognition. We are currently in the process of hand-segmenting the bird database so as to be able to evaluate localization quantitatively.

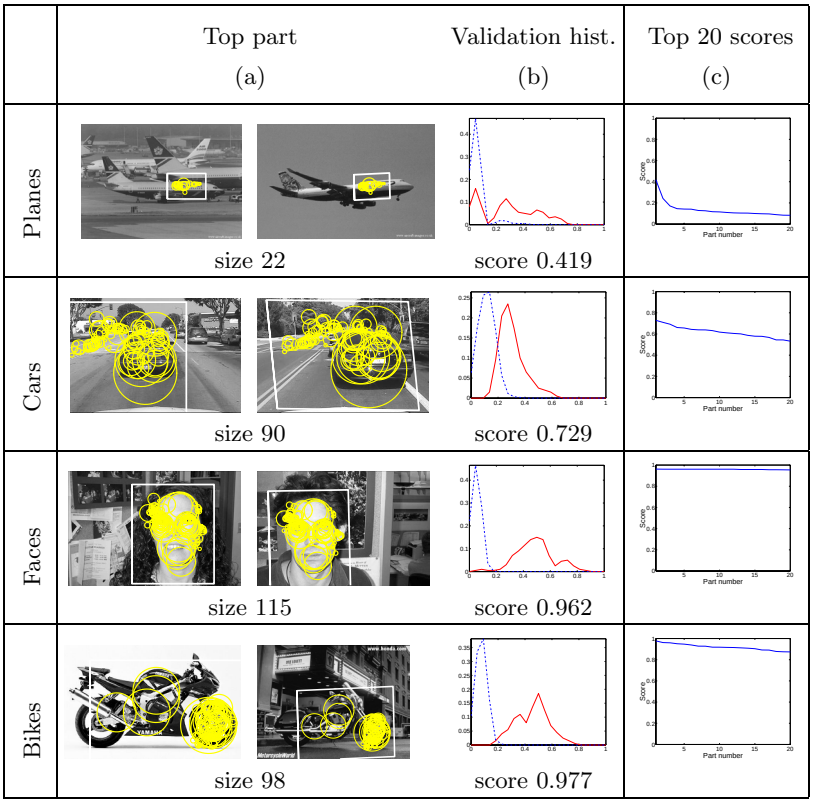


Fig. 5. Learning part vocabularies for the CalTech database. (a) The highest-scoring part for each class. The two training images that were originally matched to obtain the part are shown side by side, with the matched regions (yellow circles) superimposed. The aligning transformation between the two groups of matches is indicated by the bounding boxes: the axis-aligned box in the left image is mapped onto the parallelogram in the right image. (Recall that we use an affine alignment model and then discard any transformation that induces too much distortion.) (b) Repeatability histograms for the top part. The solid red line (resp. dashed blue line) indicates the histogram of repeatability rates of the part in all positive (resp. negative) training images. Recall that the validation score of the part is given by the χ^2 distance between the two histograms. (c) Plots of top 20 part scores following validation.

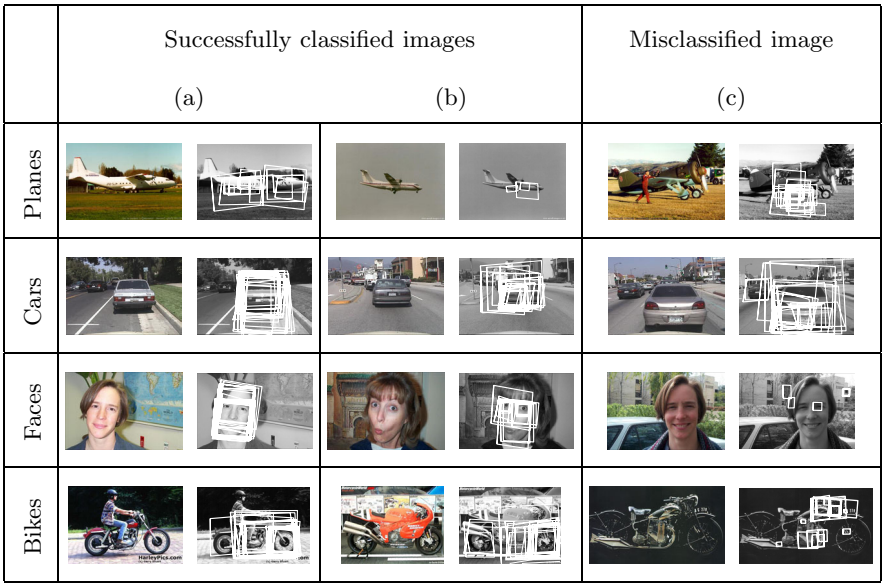


Fig. 6. CalTech results. (a), (b) Two examples of correctly classified images per class. Left of each column: original image. Right of each column: transformed bounding boxes of all detected part instances for the given class superimposed on the image. (c) Examples of misclassified images. Note that localization is poor for airplanes and very good for faces (notice the example a changed facial expression). For motorbikes, the front wheel is particularly salient. Out of the entire test set, only one bike image was misclassified, and it is one in which the front wheel is not properly visible.

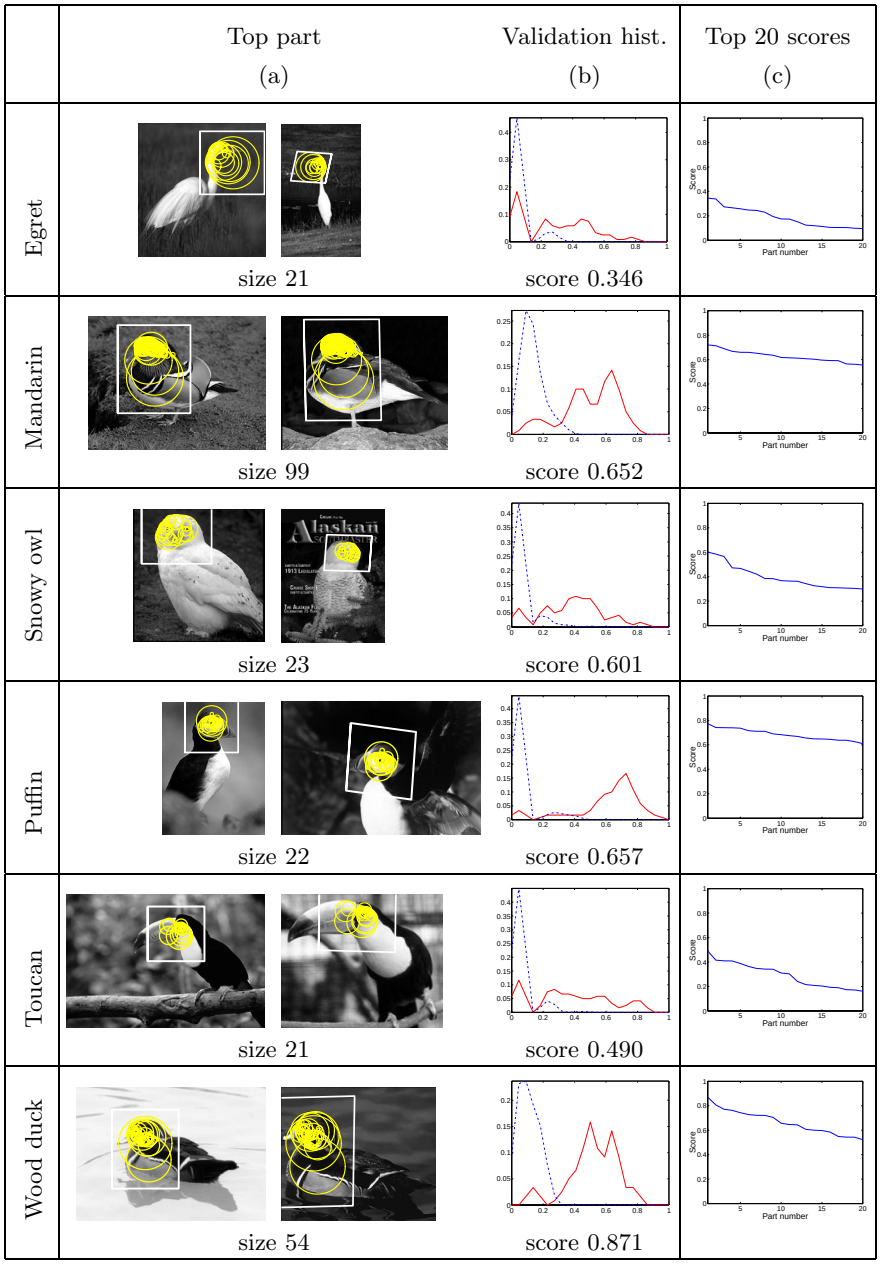


Fig. 7. Learning part vocabularies for the birds database. (a) The highest-scoring part for each class superimposed on the two original training images. (b) Validation repeatability histograms for the top parts. (c) Plots of validation scores for the top 20 parts from each class.

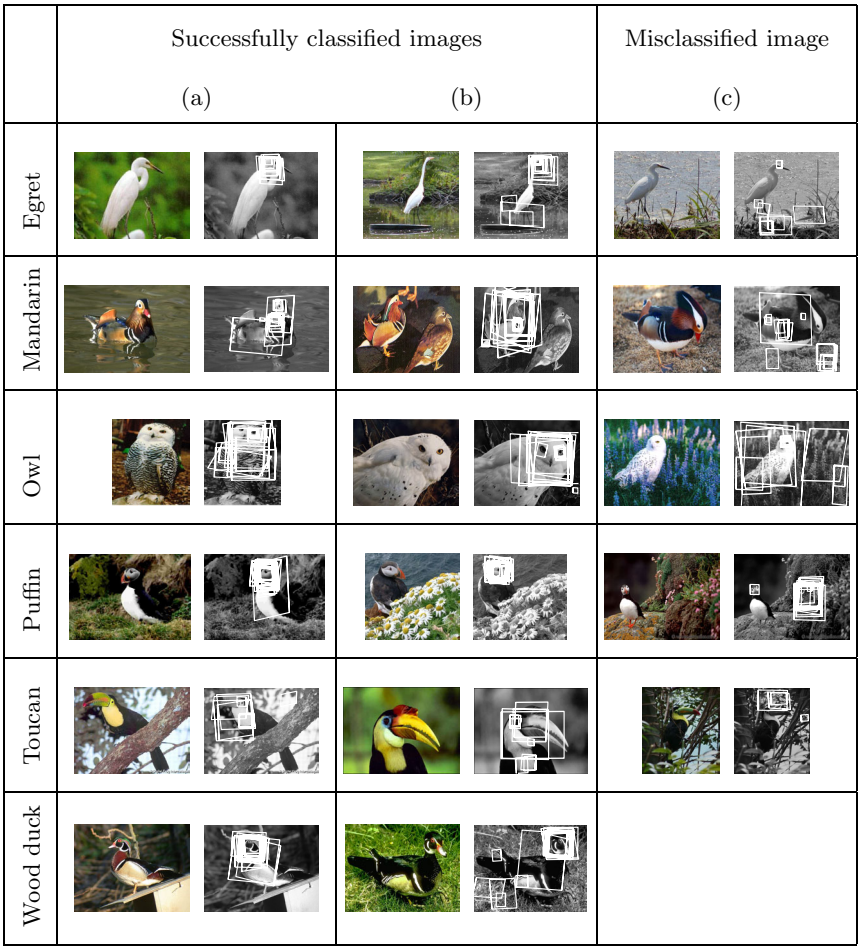


Fig. 8. Birds database results. (a), (b) Two examples of successfully classified images per class. The original test image is on the left, and on the right is the image with superimposed bounding boxes of all detected part instances for the given class. Notice that localization is fairly good for mandarin and wood ducks (the head is the most distinctive feature). Though owl parts are more prone to false positives, they do capture salient characteristics of the class: the head, the eye, and the pattern of the feathers on the breast and wings. (c) Misclassified examples. The wood duck class has no example because it achieved 100% classification rate.

5 Summary and Future Work

In this chapter, we have presented an approach to texture and object recognition that uses a visual dictionary of textons or object parts in combination with a discriminative maximum entropy framework. Our experiments have shown that the approach works well for both textures and objects. The classification rate achieved by our method on the UIUC database exceeds the state of the art [13], and our results on the four CalTech classes are comparable to others in recent literature [6, 7]. Interestingly, while all our recognition experiments used small training sets (from 3 to 50 images per class), no overfitting effects were observed. In addition, we have found that the Naive Bayes method, which we used as a baseline to evaluate the improvement provided by the exponential model, can be quite powerful in some cases — a finding that is frequently expresssed in the document classification literature [19, 21]. Specifically, for the Brodatz database, Naive Bayes outperforms the other baseline, histograms with χ^2 distance; for the CalTech database, it performs as well as the exponential model.

The most important negative result of this chapter is the lack of performance improvement from co-occurrence and overlap relations. Once again, this is consistent with the conventional wisdom in the document classification community, where it was found that for document-level discrimination tasks, a simple orderless "bag-of-words" represententation is effective. For textures, we expect that co-occurrence features may be helpful for distinguishing between different textures that consist of local elements of similar appearance, but different spatial layouts. To investigate this further, it is necessary to collect larger-scale, more difficult texture databases that include a wider variety of classes. For object recognition, the lack of improvement can be ascribed, at least partly, to the weakness of our overlap relations, especially compared to the strong geometric consistency constraints encoded within semi-local parts. In the future, we plan to investigate geometric relations that capture more discriminative information, and to test their behavior for classification on additional object databases.

Acknowledgments. This research was supported by Toyota, NSF grants IIS-0308087 and IIS-0312438, the European project LAVA (IST-2001-34405), and the CNRS-UIUC Collaboration Agreement.

References

1. S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," In *Proc. ECCV* 2002, vol. 4, pp. 113-130.
2. A. Berg, T. Berg, and J. Malik, "Shape Matching and Object Recognition Using Low-Distortion Correspondence," In *Proc. CVPR* 2005.
3. A. Berger, S. Della Pietra, and V. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics* 22(1):39–71, 1996.
4. P. Brodatz, *Textures: A Photographic Album for Artists and Designers*, Dover, New York, 1966.
5. S. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," In *Proc. Conf. of the Association for Computational Linguistics* 1996, pp. 310-318.

6. G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual Categorization with Bags of Keypoints," In *ECCV Workshop on Statistical Learning in Computer Vision* 2004.
7. G. Dorko and C. Schmid, "Selection of Scale-Invariant Parts for Object Class Recognition," In *Proc. ICCV* 2003, vol. I, pp. 634-640.
8. R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," In *Proc. CVPR* 2003, vol. II, pp. 264-271.
9. V. Ferrari, T. Tuytelaars, and L. Van Gool, "Simultaneous object recognition and segmentation by image exploration," In *Proc. ECCV* 2004.
10. J. Jeon and R. Manmatha, "Using Maximum Entropy for Automatic Image Annotation," In *Proc. Conf. on Image and Video Retrieval* 2004, pp. 24-32.
11. F. Jurie and C. Schmid, "Scale-invariant Shape Features for Recognition of Object Categories," In *Proc. CVPR* 2004.
12. D. Keysers, F. Och, and H. Ney, "Maximum Entropy and Gaussian Models for Image Object Recognition," *DAGM Symposium for Pattern Recognition* 2002.
13. S. Lazebnik, C. Schmid, and J. Ponce, "A Sparse Texture Representation Using Local Affine Regions," *IEEE Trans. PAMI* 27(8): 1265-1278, 2005.
14. S. Lazebnik, C. Schmid, and J. Ponce, "A Maximum Entropy Framework for Part-Based Texture and Object Recognition," In *Proc. ICCV* 2005, to appear.
15. S. Lazebnik, C. Schmid, and J. Ponce, "Semi-local Affine Parts for Object Recognition," In *Proc. BMVC* 2004.
16. T. Lindeberg, "Feature Detection with Automatic Scale Selection," *IJCV* 30(2):77-116, 1998.
17. D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV* 60(2):91-110, 2004.
18. S. Mahamud, M. Hebert, and J. Lafferty, "Combining Simple Discriminators for Object Discrimination," *ECCV 2002*.
19. A. McCallum and K. Nigam, "A Comparison of Event Models for Naive Bayes Text Classification," *AAAI-98 Workshop on Learning for Text Categorization* 1998, pp. 41-48.
20. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," In *Proc. CVPR* 2003, vol. 2, pp. 257-263.
21. K. Nigam, J. Lafferty, and A. McCallum, "Using Maximum Entropy for Text Classification," *IJCAI Workshop on Machine Learning for Information Filtering* 1999, pp. 61-67.
22. F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints," *IJCV*, 2005, to appear.
23. J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," In *Proc. ICCV* 2003, pp. 1470-1477.
24. J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman, "Discovering objects and their location in images," In *Proc. ICCV* 2005, to appear.
25. M. Varma and A. Zisserman, "Texture Classification: Are Filter Banks Necessary?" In *Proc. CVPR* 2003, vol. 2, pp. 691-698.
26. M. Weber, M. Welling, and P. Perona, "Unsupervised learning of models for recognition," In *Proc. ECCV* 2000, vol. 1, pp. 18–32.
27. J. Willamowski, D. Arregui, G. Csurka, C. R. Dance, and L. Fan, "Categorizing nine visual classes using local appearance descriptors," In *International Workshop on Learning for Adaptable Visual Systems*, 2004.
28. S.C. Zhu, Y.N. Wu, and D. Mumford, "Filters, Random Fields, and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling," *IJCV* 27(2):1-20, 1998.