A Maximum Entropy Framework for Part-Based Texture and Object Recognition

Svetlana Lazebnik, Cordelia Schmid, Jean Ponce

A Maximum Entropy Framework for Part-Based Texture and Object Recognition

Svetlana Lazebnik
Beckman Institute
University of Illinois, USA
slazebni@uiuc.edu

Cordelia Schmid
INRIA Rhône-Alpes
Montbonnot, France
cordelia.schmid@inrialpes.fr

Jean Ponce
Beckman Institute
University of Illinois, USA
ponce@cs.uiuc.edu

Abstract

This paper presents a probabilistic part-based approach for texture and object recognition. Textures are represented using a part dictionary found by quantizing the appearance of scale- or affine-invariant keypoints. Object classes are represented using a dictionary of composite semi-local parts*, or groups of neighboring keypoints 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 [4] and video retrieval [17]. More generally, image representations based on keypoints, or salient regions, have shown promise for recognizing textures [10] and object classes [1, 5, 6]. For textures, the appearance of local regions is clustered to form characteristic texture elements, or *textons*. For objects, such clusters can also play the role of generic object parts, though in our previous work [11], we have introduced a more expressive representation based on composite *semi-local parts*, defined as geometrically stable configurations of multiple keypoints that are robust against approximately rigid deformations and intra-class variations.

In the present work, our goal is to develop probabilistic learning and inference techniques for reasoning about object and texture models composed of multiple parts. To this end, we adopt a discriminative *maximum entropy* framework, which has been used successfully for text document classification [2, 16] and image annotation [7]. 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 [18], where it runs into issues of intractability for learning and inference, it can be far more efficient for *discriminative* tasks, e.g. [9, 14]. In this paper, we explore maximum entropy in a part-based setting. We begin in Section 2 by reviewing the basics of maximum entropy. Sections 3 and 4 describe our application of the framework to texture and object recognition, and Section 5 concludes with a summary and discussion of future directions.

2. The Maximum Entropy Framework

A discriminative maximum entropy approach seeks to estimate the posterior distribution of the class label given the 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 image data. 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 (specific definitions 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 $P(c|I)$ 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*

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

where $Z = \sum_c \exp\left(\sum_k \lambda_k f_k(I,c)\right)$ is the normalizing factor,[1] and 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 [2, 16]. 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. The derivation of updates is omitted here, but it can be shown [2, 16] 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:

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

In the present work we will use only normalized features.

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 tendency to overfit can be alleviated by adding a zero-mean Gaussian prior on the weights [16]. 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 Laplace smoothing, i.e., adding one to each feature value and renormalizing. To simplify the subsequent presentation, we will omit this operation from all feature function definitions.

In practice, it is often convenient to define feature functions based on *class-independent* features $g_k(I)$:

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

Then we have $P(c|I) = Z^{-1}\exp\left(\sum_{d,k}\lambda_{d,k}f_{d,k}(I,c)\right) = Z^{-1}\exp\left(\sum_k \lambda_{c,k}g_k(I)\right)$. Thus, "universal" features g_k become associated with class-specific weights $\lambda_{c,k}$. All our feature functions will be defined in this way.

3. Texture Recognition

3.1. Feature Functions

We use a sparse image representation [10] based on scale- or affine-invariant keypoints (regions shaped like circles and ellipses, respectively). A texton dictionary is formed by clustering appearance-based descriptors of keypoints, and each descriptor from a training or test image is then assigned the label of the closest cluster center. Implementation details of these steps will be given in Section 3.2.

[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 the 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.

In text classification, feature functions are typically based on scaled counds of word occurrences [16]. By analogy, we define feature functions using 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). Also, let $N_{k\diamond\ell}(I)$ denote the number of times the relation occurs in the image I. Then we set

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

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 [3]. 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). 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 [2], mutual information [5, 16] and likelihood ratio [5]. 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 to get 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) and the UIUC database [10] (1000 images: 25 classes, 40 samples per class). Figure 1 shows examples of images from the two databases. For the Brodatz database, we use a

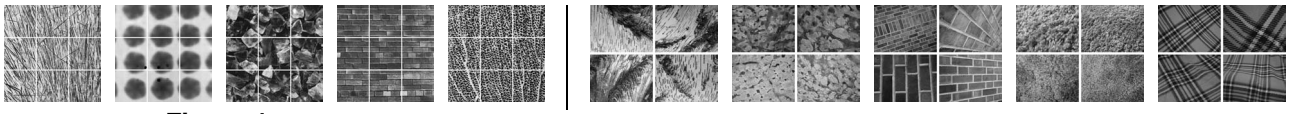


Figure 1. Examples of five classes from the Brodatz database (left) and the UIUC database (right).

scale-invariant Laplacian blob detector [12]. For the UIUC database, which contains perspective distortions and non-rigid deformations between samples of the same class, we use an affinely adapted version of the Laplacian detector. In both cases, the appearance of the detected regions is represented using SIFT descriptors [13].

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. The second row shows results for a Naive Bayes baseline using the *multinomial event model* [15]: $P(I|c) = \prod_k P(k|c)^{N_k(I)}$, 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 almost identical, 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 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 textons-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. With these features, 100% recognition rate is achieved by 61 classes from the Brodatz database and by 8 classes from the UIUC database.

	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).

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 [10]. Note, however, that the result of [10] 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 [10], that of 92.61%.

4. Object Recognition

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 [8], 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* [11], or groups of several neighboring keypoints 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 correspondence search. 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. Matching is accomplished efficiently with the help of strong appearance (descriptor similarity) and geometric consistency constraints (see [11] for details). Originally, we have introduced semi-local parts in conjunction with affine alignment; however, for the two databases of Section 4.2, scale invariance is sufficient. In the implementation, we still use linear least squares to estimate an affine aligning transformation between the regions in a hypothesized match, and then reject any hypothesis with too much distortion (skew, rotation, anisotropic scaling).

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 I as the number of regions in that instance. If several instances of the part are detected, we select the one with the highest repeatability; if no instance of part k is detected at all, we have $\rho_k(I) = 0$. Next, we compute a *validation score* for the part by taking the χ^2 distance between the histogram of repeatabilities of the part over the positive class and the histogram of its repeatabilities in all the negative images (for examples of these histograms, see Figures 3 (a) and 4 (a)). The 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 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

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

Note that it would be straightforward to define more elaborate relations that take into account the distance, relative scale, or relative orientations of the two parts [1]. However, such relations would have less geometric invariance (in particular, they would be unsuitable for non-rigid objects), and would require much more training data to learn reliably.

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 [6]. We have taken 300 images each from four classes: airplanes, rear views of cars, faces, and motorbikes (Figure 3). 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 4). 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 (candidate part computation, part selection, and testing) takes about a week.

Parts (a) of Figures 3 and 4 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. Interestingly, both airplanes and egrets are "thin" objects lacking characteristic texture, so that the keypoints that overlap the object also capture a lot of background, and the SIFT descriptors of these keypoints end up describing mostly clutter.

Tables 2 (a) and (b) 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 $P(I|c) = \prod_k P(\rho_k(I)|c)$. The distributions $P(\rho_k|c)$ are found by histogramming the repeatabilities of part k on all training images from class c. This takes into account the repeatability of parts on images from *all* classes, not only the class which they 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-

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

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 2. Classification rates for (a) CalTech (top) and (b) birds (bottom) using 20 parts per class (see text).

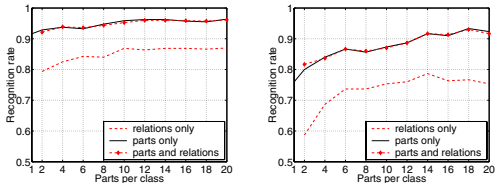


Figure 2. Classification rate (exp. parts) as a function of dictionary size: CalTech database (left), birds database (right). For the CalTech database, because three of the four classes have extremely strong and redundant parts, performance increases very little as more parts are added. For the bird database, diminishing returns set in as progressively weaker parts are added.

repeatability detections of parts from other classes. Note that 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 (a) and (b) 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. 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 2 shows a plot of the classification rate for the exponential model as a function of part dictionary size. Note that 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 [2].

Though we did not conduct a quantitative evaluation of localization accuracy, the reader may get a qualitative idea by examining parts (b) and (c) of Figures 3 and 4, 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.

5. Summary and Future Work

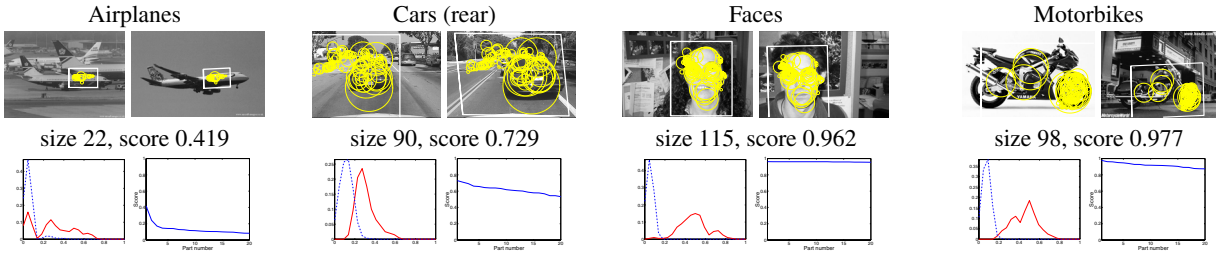
In this paper, we have presented a part-based approach to texture and object recognition using a discriminative maximum entropy framework. Our experiments have shown that the exponential model works well for both textures and objects. The classification rate achieved by our method on the UIUC database exceeds the state of the art [10], and our results on the four CalTech classes are comparable to others in recent literature [4, 5]. 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 [15, 16].

The most important negative result of this paper 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. For object recognition, the lack of improvement from relations can be ascribed, at least partly, to the strong geometric consistency constraints already captured by semi-local parts. For weaker "atomic" parts, relations have indeed been shown to improve performance [1]. We currently conjecture that combined with our semi-local part representation, overlap relations may be more useful for localization than for recognition. The key goal of our future experiments is to test this conjecture experimentally.

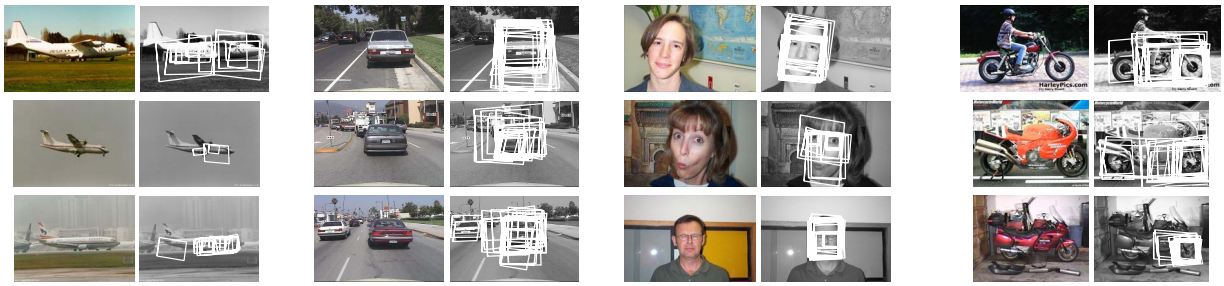
Acknowledgments. This research was supported by Toyota, National Science Foundation grants IIS-0308087 and IIS-0312438, the European project LAVA (IST-2001-34405), and the CNRS-UIUC Collaboration Agreement.

References

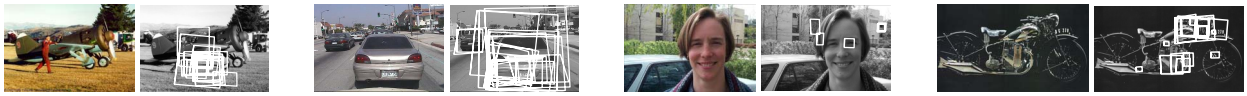
[1] S. Agarwal and D. Roth, "Learning a Sparse Representation for Object Detection," *ECCV* 2002, vol. 4, pp. 113-130.



(a) Modeling and part selection (see caption below).



(b) Correctly classified examples under the exponential parts-only model.



(c) Misclassified examples.

Figure 3. CalTech results. (a) First row: 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.) Second row, left of each column: 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. Second row, right of each column: plot of top 20 part scores following validation. (b) Three 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. Localization is poor for airplanes and very good for faces (notice the examples with closed eyes and a changed facial expression). For motorbikes, the front wheel is particularly salient. (c) Examples of misclassified images.

[2] A. Berger, S. Della Pietra, and V. Della Pietra, "A Maximum Entropy Approach to Natural Language Processing," *Computational Linguistics* 22(1):39–71, 1996.

[3] S. Chen and J. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling," *Proc. Conf. of the Assoc. for Comp. Linguistics* 1996, pp. 310-318.

[4] G. Csurka, C. Bray, C. Dance, and L. Fan, "Visual Categorization with Bags of Keypoints," *ECCV Workshop on Statistical Learning in Computer Vision* 2004.

[5] G. Dorko and C. Schmid, "Selection of Scale-Invariant Parts for Object Class Detection," *CVPR* 2003.

[6] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *CVPR* 2003, vol. II, pp. 264-271.

[7] J. Jeon and R. Manmatha, "Using Maximum Entropy for Automatic Image Annotation," *Proc. Conf. on Image and Video Retrieval* 2004, pp. 24-32.

[8] F. Jurie and C. Schmid, "Scale-invariant Shape Features for Recognition of Object Categories," *CVPR* 2004.

[9] D. Keysers, F. Och, and H. Ney, "Maximum Entropy and Gaussian Models for Image Object Recognition," *DAGM Symposium for Pattern Recognition* 2002.

[10] S. Lazebnik, C. Schmid, and J. Ponce, "A Sparse Texture Representation Using Local Affine Regions," *IEEE Trans. PAMI* 27(8): 1265-1278, 2005.

[11] S. Lazebnik, C. Schmid, and J. Ponce, "Semi-local Affine Parts for Object Recognition," *BMVC* 2004.

[12] T. Lindeberg, "Feature Detection with Automatic Scale Selection," *IJCV* 30(2):77-116, 1998.

[13] D. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV* 60(2):91-110, 2004.

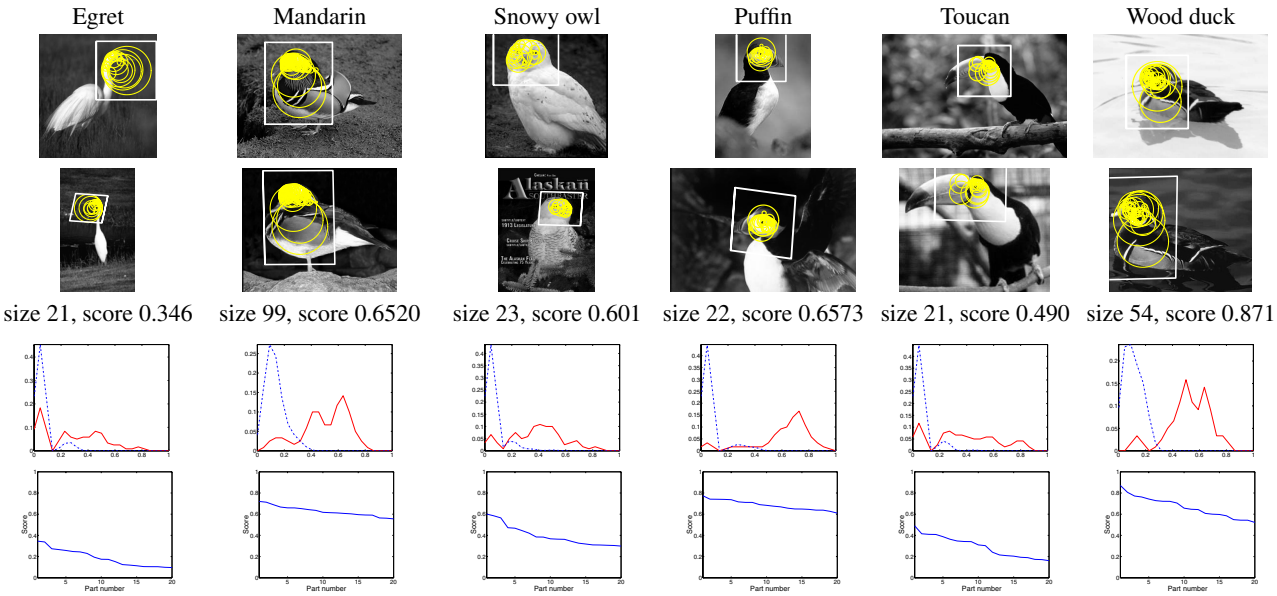
[14] S. Mahamud, M. Hebert, and J. Lafferty, "Combining Simple Discriminators for Object Discrimination," *ECCV 2002*.

[15] 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.

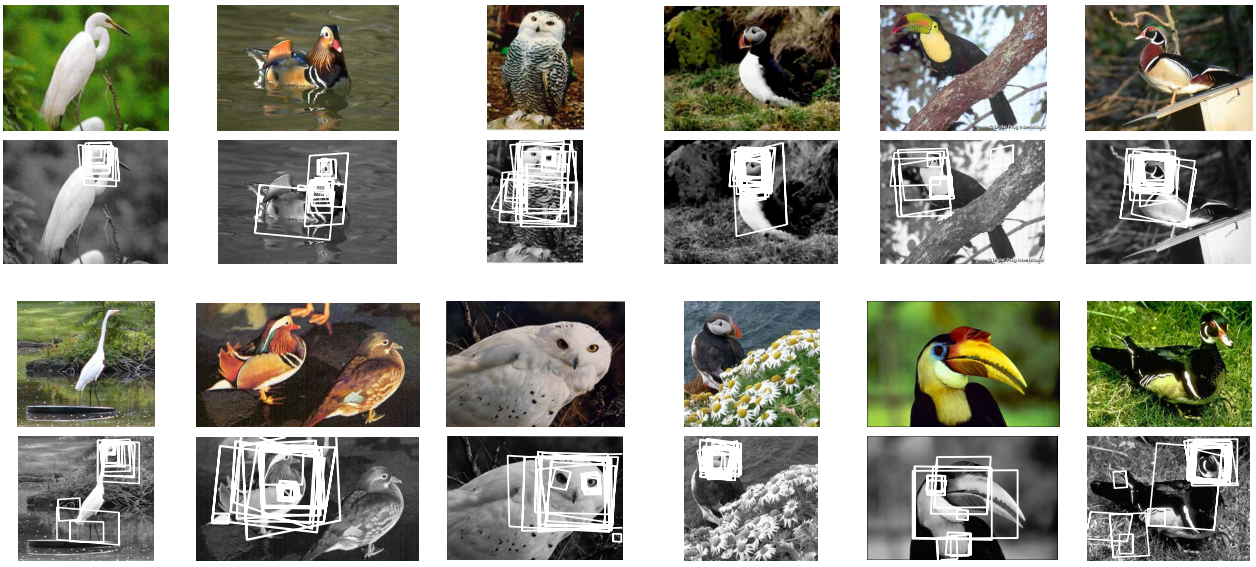
[16] K. Nigam, J. Lafferty, and A. McCallum, "Using Maximum Entropy for Text Classification," *IJCAI Workshop on Machine Learning for Information Filtering* 1999, pp. 61-67.

[17] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *ICCV* 2003.

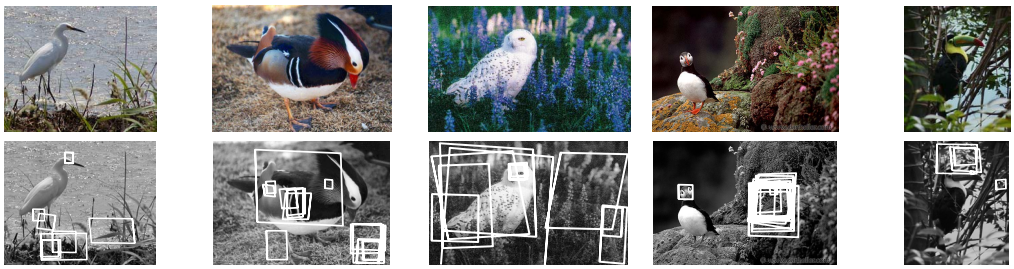
[18] S.C. Zhu, Y.N. Wu, and D. Mumford, "Filters, Random Fields, and Maximum Entropy (FRAME): Towards a Unified Theory for Texture Modeling," *IJCV* 27(2):1-20, 1998.



(a) Modeling and part selection (see caption below).



(b) Correctly classified examples under the exponential parts-only model.



(c) Misclassified examples.

Figure 4. Birds database results. (a) First and second rows: highest-scoring part for each class superimposed on the two original training images. Third row: validation repeatability histograms for the top parts. Fourth row: plots of validation scores for the top 20 parts from each class. (b) Two examples of successfully classified images per class. The original test image is on top, and below it 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.