Int J Comput Vis (2008) 77: 175–198
DOI 10.1007/s11263-007-0091-7

Efficient Learning of Relational Object Class Models

Aharon Bar-Hillel · Daphna Weinshall

Received: 18 August 2005 / Accepted: 11 September 2007 / Published online: 17 November 2007

Abstract We present an efficient method for learning part-based object class models from unsegmented images represented as sets of salient features. A model includes parts' appearance, as well as location and scale relations between parts. The object class is generatively modeled using a simple Bayesian network with a central hidden node containing location and scale information, and nodes describing object parts. The model's parameters, however, are optimized to reduce a loss function of the training error, as in discriminative methods. We show how boosting techniques can be extended to optimize the relational model proposed, with complexity linear in the number of parts and the number of features per image. This efficiency allows our method to learn relational models with many parts and features. The method has an advantage over purely generative and purely discriminative approaches for learning from sets of salient features, since generative method often use a small number of parts and features, while discriminative methods tend to ignore geometrical relations between parts. Experimental results are described, using some bench-mark data sets and three sets of newly collected data, showing the relative merits of our method in recognition and localization tasks.

Keywords Object class recognition · Object localization · Generative models · Boosting · Weakly supervised learning

A. Bar-Hillel (✉)
Intel Research Israel, P.O.Box 1659, Haifa 31015, Israel
e-mail: aharon.bar-hillel@intel.com

D. Weinshall
Computer Science Department and the Center for Neural Computation, The Hebrew University of Jerusalem, Jerusalem 91904, Israel

1 Introduction

One of the important organization principles of object recognition is the categorization of objects into object classes. Categorization is a hard learning problem due to the large inner-class variability of object classes, in addition to the "common" object recognition problems of varying pose and illumination. Recently, there has been a growing interest in the task of object class recognition (Fergus et al. 2003, 2005; Agarwal et al. 2004; Opelt et al. 2004b; Csurka et al. 2004; Leibe et al. 2004; Feltzenswalb and Huttenlocher 2005; Fritz et al. 2005; Loeff et al. 2005; Dorkó and Schmid 2005) which can be defined as follows: given an image, determine whether the object of interest appears in the image. In many cases the localization of the object in the image is also sought.

Following previous work (Fergus et al. 2003; Vidal-Naquet and Ullman 2003), we represent an object using a part-based model (see illustration in Fig. 1). Such models can capture the essence of most object classes, since they represent both parts' appearance and invariant relations of location and scale between the parts. Part-based models are somewhat resistant to various sources of variability such as within-class variance, partial occlusion and articulation, and they are potentially convenient for indexing in a more complex system (Lowe 2001; Leibe et al. 2004).

Part-based approaches to object class recognition can be crudely divided into two types: (1) 'generative' methods which compute class models (Fergus et al. 2003, 2005; Leibe et al. 2004; Feltzenswalb and Huttenlocher 2005; Fei-Fei et al. 2003; Fritz et al. 2005; Loeff et al. 2005) and (2) 'discriminative' methods which do not compute class models (Opelt et al. 2004a, 2004b; Csurka et al. 2004; Serre et al. 2005; Viola and Jones 2001; Dorkó and Schmid 2005). In the Generative approach, a probabilistic model of

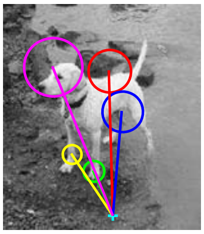


Fig. 1 (Color online) Dog image with our learnt part-based model drawn on top. Each *circle* represents a part in the model. The parts relative location and scale are related to one another through a hidden center

the object class is learnt by likelihood maximization. Afterwards, the likelihood ratio test is used to classify new images. The main advantage of this approach is the ability to naturally model relations between object parts. In addition, domain knowledge can be incorporated into the model's structure and priors. Discriminative methods do not learn explicit class models, and instead they seek a classification rule which discriminates object images from background images. The main advantage of discriminative methods is the direct minimization of a classification-based error function, which typically leads to superior classification results (NG and Jordan 2001). Additionally since these methods do not explicitly model object classes, they are usually computationally efficient.

In our current work, we suggest to combine the two approaches in order to enjoy the benefits of both worlds: the modeling power of the generative approach, with the accuracy and efficiency of discriminative optimization. We motivate this idea in Sect. 2 using general considerations, and as a solution to some problems encountered in related work. Our argument relies on two basic claims. The first is that feature relations are powerful cues for recognition, and perhaps indispensable cues for semantical recognition-related tasks like object localization or part identification. Clearly relations can be more naturally incorporated into an explicit generative model than an abstract discriminator. On the other hand, we argue that generative learning procedures are inadequate in the specific context of learning from unsegmented images, due essentially to computational and functional reasons. We therefore propose to replace maximum-likelihood optimization in the generative learning, by the discriminative optimization of the classifiers' parameters. The initial description of the main techniques and most of the recognition results has appeared in conference proceedings (Bar-Hillel et al. 2005a, 2005b).

Specifically, we suggest a novel learning method for classifiers based on a simple part based model. The model, described in Sect. 3, is a 'star'-like Bayesian network, with a central hidden node describing the objects location and scale. The location and scale of the different parts depend only on the central hidden variable, and so the parts are conditionally independent given this variable. Such a model allows us to represent part relations with low inference computational complexity. Models of similar topology are implicitly or explicitly considered in Lowe (2001), Leibe et al. (2004), Fritz et al. (2005), Fergus et al. (2005). While using a generative object model, we optimize its parameters by minimizing a loss over the training error, as done in discriminative learning. We show how a standard boosting approach can be naturally extended to learn such a model with conditionally independent parts. Learning time is linear in the number of parts and the number of feature extracted per image. Beyond this extension, we consider a wider family of gradient descent optimization algorithms, of which the extended boosting is a special case. Optimal performance is empirically achieved using algorithms from this family that are close to the extended boosting, but not identical to it. The discriminative optimization methods are discussed in Sect. 4.

Our experimental results are described in Sect. 5. We compare the recognition and localization performance of our algorithm to several state-of-the-art methods, using the benchmark data sets of Fergus et al. (2003) and Agarwal and Roth (2002). In the recognition task, our performance is somewhere in the middle. Our algorithm is usually better than generative methods which keep a 1-1 part-feature correspondence (Fergus et al. 2003, 2005), since it is able to learn larger models with selective features. It is also superior to plain boosting (Opelt et al. 2004b) which neglects spatial part relations. However, it is outperformed by Dorkó and Schmid (2005) which uses a clever mixture of interest detectors, and by Loeff et al. (2005) which allows a part to be implemented by many image features. These two alternative techniques, however, are inherently ill-suited for localization, given the fuzzy nature of location in their class models. In the localization task we use techniques introduced by Feltzenswalb and Huttenlocher (2005) to efficiently scan the image and find the exact location of one or more object instances. Our localization experiments are carried with the Caltech data (Fergus et al. 2003) and a localization benchmark (Agarwal and Roth 2002). The performance achieved is comparable to the best available methods.

In order to further investigate and challenge our method, we collected three more difficult data sets containing images of chairs, dogs and humans, with matching backgrounds (we have made this data publicly available online). We used these data sets to test the algorithm's performance under harder conditions, with high visual similarity between object and background, and large pose and scale variability. We investigated the relative contribution of the appearance, location and scale components of our model, and showed the importance of incorporating location relations between object parts. In another experiment we checked the contribution of using a large numbers of parts and features, and demonstrated their relative merits. We experimented with a generic interest point detector (Kadir and Brady 2001), as well as with a discriminative interest point detector (Gao and Vasconcelos 2004); our results show a small advantage for the latter. Finally, we showed that the classifiers learnt perform well against new, unseen backgrounds.

2 Why Mix Discriminative Learning with Generative Modeling: Motivation and Related Work

In this section we describe the main arguments for combining generative and discriminative methods in the context of learning from unsegmented images. In Sect. 2.1 we review the distinction between the generative and discriminative paradigms, and assess the relative merits of each approach in general. We next discuss the specific problem of learning from unsegmented images in Sect. 2.2, and characterize it as learning from unordered feature sets, rather than data vectors. In Sect. 2.3 we claim that relations between features, best represented in a generative framework, are useful in the context of learning from unordered sets, and are specifically important for semantical recognition-related tasks. In Sect. 2.4 we argue that generative maximum-likelihood learning is highly problematic in the context of learning from unsegmented images. Specifically, we argue that such learning suffers from inherent computational problems, and that it is likely to exhibit deficient feature pruning characteristics. To solve these problems while keeping the important information of feature relations, we propose to combine the generative treatment of relations with discriminative learning techniques. In Sect. 2.5 we briefly review how feature relations are handled in related discriminative methods.

2.1 Discriminative and Generative Learning

Generative classifiers learn a model of the probability $p(x|y)$ of input x given label y. They then predict the input labels by using Bayes rule to compute $p(y|x)$ and choosing the most likely label. With 2 classes $y \in \{-1, 1\}$, the optimal decision rule is the log likelihood ratio test, based on the statistic:

$$\log \frac{p(x|y=1)}{p(x|y=-1)} - \nu \qquad (1)$$

where ν is a constant threshold. The models $p(x|y=1)$ and $p(x|y=-1)$ are learnt in a maximum likelihood framework (or maximum-a-posteriori when a useful prior is available).

Discriminative classifiers do not learn probabilistic class models. Instead, they learn a direct map from the input space X to the labels. The map's parameters are chosen in a way that minimizes the training error, or a smooth loss function of it. With two labels, the classifier often takes the form $\mathrm{sign}(f(x))$, with the interpretation that $f(x)$ models the log likelihood ratio statistic.

There are several compelling arguments in the learning literature which indicate that discriminative learning is preferable to generative learning in terms of classification performance. Specifically, learning a direct map is considered an easier task than the reliable estimation of $p(x|y)$ (Vapnik 1998). When classifiers with the same functional form are learned in both ways, it is known that the asymptotic error of a reasonable discriminative classifier is lower or equal to the error achievable by a generative classifier (NG and Jordan 2001). In addition, discriminative methods are often simpler and faster then their generative counterparts (Ulusoy and Bishop 2005).

However, when we wish to design (or choose) the functional form of our classifier, generative models can be very helpful. When building a model of $p(x|y)$ we can use our prior knowledge about the problem's domain to guide our modeling decisions. We can make our assumptions more explicit and gain semantic understanding of the model's components. Specifically, the generative framework readily allows for the modeling of parts relations, while providing us with a rich toolbox of theory and algorithms for inference and relations learning. It is plausible to expect that a carefully designed classifier, whose functional form is determined by generative modeling, will give better performance than a classifier from an arbitrary parametric family.

These considerations suggest that a hybrid path may be beneficial. More specifically, choose the functional form of the classifier using a generative model of the data, then learn the model's parameters in a discriminative setting. While the arguments in favor of this idea as presented so far are very general, we next claim that when learning from images in particular, this idea can overcome several problems in current generative and discriminative approaches.

2.2 Learning from Features Sets

Our primary problem is object class recognition from unaligned and unsegmented images, which are binary labeled as to whether or not they contain an object from the class. A natural view of this problem is as a binary classification problem, where the input is a set of features rather than an ordered vector of features, as in standard learning problems. This is an important distinction: vector representation implicitly assumes that measurements of the 'same' quantities are made for all data instances and stored in corresponding indices of the data vectors. The 'same' features in different data vectors are assumed to have the same fixed, simple relation with the class label (the same 'role'). Such implicit correspondence is often hard to find in bottom up image representation, in particular when feature maps or local descriptors sets are detected with interest point detectors.

Thus we adopt the view of image representation as a set of features. Each feature has a location index, but unlike an element in a vector, its location does not imply a predetermined fixed 'role' in the representation. Instead, only relations between locations are meaningful. Such representations present a challenge to current learning theory and algorithms, which are well developed primarily for vectorial input.

A second inherent problem arises because the relevant feature set representations usually contain a large number of spurious features. The images are unsegmented, and therefore many features may not represent the object of interest at all (but background information), while many other features may duplicate each other. Thus feature pruning is an important part of the learning problem.

2.3 Semantics and Part Relations

The lack of feature correspondence between images can be handled in two basic ways: either try to establish correspondence, or give it up to begin with. Without correspondence, images are typically represented by some statistical properties of the feature set, without assigning roles to specific image features. A notable example is the feature histogram, used for example in Csurka et al. (2004), Chan et al. (2004), Thureson and Carlsson (2004) and most of the methods in Everingham et al. (2006). These approaches are relatively simple and in some cases give excellent recognition results. In addition they tend to have good invariance properties, as the use of invariant features directly gives invariant classifiers. Most of these approaches do not consider feature relations, mainly because of their added complexity (an exception is Thureson and Carlsson 2004). The main drawback of this framework is the complete lack of image semantics. While good recognition rates can be achieved, further recognition related tasks like localization or part identification cannot be done in this framework, as they require identifying the role of specific features.

The alternative research choice, which we adopt in the current paper, seeks to identify and correspond features with the same 'role' in different images. This is done explicitly in some generative modeling approaches (Fergus et al. 2003, 2005; Feltzenswalb and Huttenlocher 2005; Leibe et al. 2004), using the notion of a probabilistically modeled 'part'. The 'part' is an entity with a fixed role (probabilistically modeled), and its instantiation in each image is a single feature, to be chosen from the set of available image features. Discriminative part based methods (Opelt et al. 2004a, 2004b; Agarwal et al. 2004; Vidal-Naquet and Ullman 2003), as well as some generative models (Loeff et al. 2005), use a more implicit 'part' notion, and their degree of commitment to finding semantically similar features in images varies. The important advantage of identifying parts with fixed roles over the images is the ability to perform image understanding tasks beyond mere recognition.

When looking in images for parts with fixed roles, feature relations (mainly location and scale relations) provide a powerful, perhaps indispensable cue. Basing part identity on appearance criteria alone is possible, and in (Opelt et al. 2004a; Serre et al. 2005; Dorkó and Schmid 2005) it leads to very good recognition results. However, as reported in (Opelt et al. 2004a), the stability of correct part identification is low, and localization results are mediocre. Specifically, it was found that typically less than 50% of the instantiating features were actually located on the object. Instead, many feature rely on the difference in background context between object and non-object images. Conversely, good localization results are reported for methods based on generative models (Fergus et al. 2003, 2005; Leibe et al. 2004). In Agarwal et al. (2004) a detection task is considered in a discriminative framework. In order to achieve good localization, gross part relations are introduced as additional features into the discriminative classifier.

2.4 Learning in Generative Models

We now consider generative model learning when the input is a set of unsegmented images. In this scenario, the model is learnt from a set of object images alone, and its parameters are chosen to maximize the likelihood of the image set (sometimes under a certain prior over models). We describe two inherent problems of this maximum likelihood approach. In Sect. 2.4.1 we claim that such learning involves an essential tradeoff, where computational efficiency is traded for weaker modeling which allows repetitive parts. In Sect. 2.4.2 we review how this problem is handled in some current generative models. In Sect. 2.4.3 we maintain that generative learning is not well adjusted to feature pruning, and becomes problematic when rich image representations are used.

2.4.1 The Computational Problem

Assume that the image is represented as a set of features (see Sect. 2.2), that our generative model incorporates part relations, and that we are committed to a notion of 'part' instantiated by a single image feature, as discussed in Sect. 2.3. Likelihood evaluation and model learning under these conditions are hard. Denote the feature set of image I by $F(I)$, and the number of features in $F(I)$ by N. While the input is a feature set, the generative model typically specifies the likelihood $P(V|M)$ for an ordered part vector $V = (f_1, \dots, f_P)$ of P parts. The problem of learning from unordered sets is tackled by considering all the possible vectors V that can be formed using the feature set. Legitimate part vectors should have no repeated features, and there are $O(N^P)$ such vectors. Thus, the image likelihood $P(I|M)$ requires marginalization[1] over all such vectors. Assuming uniform prior over these vectors, we have

$$P(I|M) = \sum_{\substack{V=(x_1,\dots,x_P)\in F(I)^P \\ x_i\neq x_j \text{ if } i\neq j}} P(V|M). \tag{2}$$

Efficient likelihood computation in relational models is only possible via the decomposition of the joint probability using conditional independence assumptions, as done in graphical models. Such decomposition specifies the probability as a product of local terms, each depending on a small subset of parts. For a part vector $V = (f_1, \dots, f_P)$

$$P(V|M) = \prod_c \Psi_c(V|_{S_c}) \tag{3}$$

where $S_c \subset \{1, \dots, P\}$ are index subsets and $V|_S = \{f_i : i \in S\}$. Using dynamic programming, inference and marginalization are exponential in the 'induced width' g of the related graphical model, which is usually relatively low (note that for trees, $g = 2$ only).

The summation in (2) does not lend itself easily to such simplifications, however. We therefore make the following approximation, in which part vectors with repetitive features are allowed

$$\begin{aligned} P(I|M) &= \sum_{\substack{(x_1,\dots,x_P)\in F(I)^P \\ x_i\neq x_j \text{ for } i\neq j}} \prod_c \Psi_c(V|_{S_c}) \\ &\approx \sum_{(x_1,\dots,x_P)\in F(I)^P} \prod_c \Psi_c(V|_{S_c}). \end{aligned} \tag{4}$$

This approximation is essential to making efficient marginalization possible. If feature repetition is not allowed, global dependence emerges between the features assigned to the different parts (as they cannot overlap). As a result we get global constraints, and efficient enumeration becomes impossible.

The approximation in (4) may appear minor, which is indeed the case when a fixed, 'reasonable' part based model is applied to an image. In this case, typically, parts are characterized by different appearance and location models, and part vectors with repetitive parts have low insignificant probability. But during learning, approximation (4) brings about a serious problem: when vectors with feature repetitions are allowed, learning may result in models with many repetitive parts. In fact, standard maximum likelihood has a strong tendency to choose such models. This is because it can easily increase the likelihood by choosing the same parts with high likelihood, over and over again.

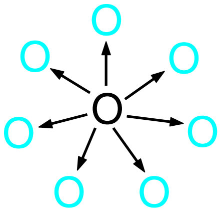


Fig. 2 (Color online) A "star" graphical model. Peripheral nodes, shown in *blue*, are related only via a hidden central node. Such a model is used in our work, as well as in Fergus et al. (2005). If (i) feature repetition is allowed (as in (4)), and (ii) model parameters are chosen to maximize the likelihood of the best object occurrence, then all the peripheral nodes are optimized to represent the same part

The intuition above can be made precise in the simple case in which a 'star' model is used (see Fig. 2) and the sum over all hypotheses is approximated by the single best features vector. In this extreme case, the maximal likelihood is achieved when all the peripheral parts models are identical. We specifically consider this model in Sect. 3 and prove the last statement in Appendix 1. The proof doesn't directly apply when a sum over all the feature vectors is used, but as this sum is usually dominated by only a few vectors, part repetition is likely to occur in this case too.

Thus, in conclusion, we see that in the ideal generative framework, one needs to choose between computational efficiency and the risk of part duplication. One way to escape this dilemma is by dropping the requirement that a part is instantiated in a single image feature, as done in (Loeff et al. 2005). This, however, leads to a vaguer 'part' notion, with lower semantic value. The alternative we suggest here keeps the 'part' notion intact, and gives up generative optimization instead.

2.4.2 *How is This Computational Problem Handled: Related Work*

Several recent approaches use generative modeling for object class recognition (Fergus et al. 2003, 2005; Fei-Fei et al. 2003; Holub et al. 2005; Feltzenswalb and Huttenlocher 2005; Loeff et al. 2005). In Fergus et al. (2003), Fei-Fei et al. (2003), Holub et al. (2005) a full relational model is used. The probability $P((f_1, \dots, f_P)|M)$ in this model cannot be decomposed into the product of local terms, due to the complex probabilistic dependencies between all of the model's parts (in graphical models terminology the model is a single large clique). As a result, both learning and recognition are exponential in the number of model parts, which limits the number of parts that can be used (up to 7 in Fergus et al.

[1] Alternatively, one may approximate the sum in (2) by a max operator, looking for the best model interpretation in the image. This does not affect the computation considerations discussed here.

2003, 4 in Fei-Fei et al. 2003, 3 in Holub et al. 2005), and the number of features per image ($N = 30, 20$, up to 100 respectively). In (Fergus et al. 2005) a decomposable model is proposed with a 'star'-like topology. This reduces the complexity of recognition (i.e., the likelihood evaluation of an existing model) significantly. However, learning remains essentially exponential, in order to avoid part repetition in the learnt model.

In contrast, the problem (as well as the feature pruning problem, discussed in the next section) is completely avoided in the case of learning from segmented images, as done in Feltzenswalb and Huttenlocher (2005). Here the input is a set of object images, with manually segmented parts and manual part correspondence between images. In this case learning is reduced to standard maximum likelihood estimation of vectorial data. As stated above, Loeff et al. (2005) avoid the computational problem by allowing for each part to be implemented in many image features.

2.4.3 Feature Pruning

We argued in Sect. 2.2 that feature pruning is necessary when learning from images. P, the number of parts in the model, is often much smaller than the number of features per image N. This is usually not the case in classical applications of generative modeling, in which data is typically described as a relatively small feature vector.

When $P \ll N$, maximum likelihood chooses to model only parts with high likelihood—often parts which are highly repetitive in images, with repetitive relations. This optimization policy has a number of drawbacks. On the one hand, it introduces a preference for simple parts, as these tend to have low variability through images, which gives rise to high likelihood scores. It also introduces preference for features which are frequent in natural images, whether they belong to the object or not. On the other hand, there is no explicit preference for discriminative parts, nor any preference for feature diversity. As a result, certain aspects of the object may be extensively described, while others are neglected. The problem may be intuitively summarized by stating that generative methods can describe the data, but they cannot choose what to describe. Additional task related signal, external to the data, is needed, and is most readily provided by labels.

In Fergus et al. (2003), Fei-Fei et al. (2003), initial feature pruning is obtained by using the Kadir and Bradey detector (Kadir and Brady 2001), which finds relatively diverse, high entropy regions in the image. Explicit preference is given to features with large scale, which tend to be more discriminative. In addition, they limit the number of features per image ($N = 20, 30$). To some extent, the burden of feature pruning is placed on the pre-learning feature detection mechanisms. However, with such a small number of features per image, objects do not always get sufficient coverage. In fact, learning is very sensitive to the fine tuning of the feature pruning mechanism.

In Fergus et al. (2005), where a 'star'-like decomposable model is used, more parts and features are used in the generative learning experiments. Surprisingly, the results do not show obvious improvement. Increasing the number of parts P and features N_f does not typically reduce the error rates, since many of the additional features turn out to be irrelevant, which makes feature pruning harder. In Sect. 5 we investigate the impact that P and N_f have on performance for models similar to those used by Fergus et al. (2005), but optimized discriminatively. In our experiments extra information (increased N_f) and modeling power (increased P) clearly lead to better performance.

2.5 Relations in Discriminative Methods

Many part based object class recognition methods are mostly discriminative (Opelt et al. 2004b; Vidal-Naquet and Ullman 2003; Ullman et al. 2002; Agarwal et al. 2004; Dorkó and Schmid 2005). In most of these methods, spatial relations between parts are not considered at all. While some of these systems exhibit state-of-the-art recognition performance, they are usually lacking in further, more semantical tasks as localization and part identification, as described in Sect. 2.3. In the 'fragment based' approach of (Vidal-Naquet and Ullman 2003; Ullman et al. 2002) relations are not used, but when the same approach is applied to segmentation, which requires richer semantics, fragment relations are incorporated (Borenstein at al. 2004).

One way to incorporate part relations into a discriminative setting is used by the object detection system of Agarwal et al. (2004). The task is localization, and it requires the exact correspondence and the identification of parts. To achieve this, qualitative location relations between fragment features are also considered as features, creating a very large and sparse feature vector. Discriminative learning in this very high dimensional space is then done using a specific feature-efficient learning algorithm. The relational features in this scheme are highly qualitative (for example, 'fragment a in to the left and bottom of fragment b'). Another problem with this approach is that supervised learning from high dimensional sparse vectors is a hard problem, often requiring dimensionality reduction to enable efficient learning.

In this context, our main contribution may be the design of a relatively simple and efficient technique for the introduction of relational information into the discriminative framework of boosting. As such, our work is related to the purely discriminative techniques used in Opelt et al. (2004a, 2004b). In spirit, our work has some resemblance to the work of (Torralba et al. 2004), in which relational context information is incorporated into a boosting process.

However, the techniques we use and the task we consider are quite different.

2.6 Similar Approaches to the Generative-Discriminative Combination

In our work, the generative-discriminative combination is aimed at solving a very specific problem: how to allow the efficient learning of part-based models with spatial part relations. But when viewed more broadly, it is an instance of a more general recent trend, trying to combine the representation advantage of generative models with the accuracy and goal-oriented nature of discriminative ones. In many cases, the combination is done by concatenating an initial generative stage, which provides the representation, with a second discriminative stage for the actual classification. In Holub et al. (2005) and Fritz et al. (2005), generative methods (previously presented in Fergus et al. 2003 and Leibe et al. 2004 respectively) are augmented with a discriminant SVM-based second stage. This approach is shown to considerably enhance recognition (Holub et al. 2005) and localization results (Fritz et al. 2005). In these two examples the generative models include spatial relations. Other approaches use a similar 2-stage procedure for a bag-of-features model (Li et al. 2005; Dorkó and Schmid 2005), and obtain excellent recognition results. In these approaches the set of object image features is represented using a Gaussian mixture model, followed by a discriminative procedure which selects informative Gaussian components and uses them for classification.

In Holub and Perona (2005), Holub et al. present an object recognition method which, like our proposed scheme, relies on discriminative optimization of a generative model based classifier. However, the proposed discriminative optimization does not solve the computational problem described in Sect. 2.4.1, and learning is even slower than the parallel generative learning procedure. The models learnt are hence limited to 3–4 parts. Note that the 2-stage methods described above (Holub et al. 2005; Fritz et al. 2005) do not solve the computational problem either. Specifically, the method of Holub et al. (2005) is also limited to 3–4 parts in practice, and the method of Fritz et al. (2005) learns from segmented or highly aligned images.

3 The Generative Model

We represent an input image using a set of local descriptors obtained using an interest point detector. Some details regarding this process are given in Sect. 3.1. We then define a classifier over such sets of features using a generative object model. The model and the resulting classifier are described in Sects. 3.2 and 3.3 respectively.

3.1 Feature Extraction and Representation

Our feature extraction and representation scheme mostly follows the scheme used in Fergus et al. (2003). Initially, images were rescaled to have a uniform horizontal length of 200 pixels. We experimented with two feature detectors: (1) Kadir & Brady (KB) (Kadir and Brady 2001), and (2) Gao & Vasconcellos (GV) (Gao and Vasconcelos 2004).[2] The KB detector is a generic detector. It searches for circular regions of various scales, that correspond to the maxima of an entropy based score in scale space. The GV detector is a discriminative saliency detector, which searches for features that permit optimal discrimination between the object class and the background class. Given a set of labeled images from two classes, the algorithm finds a set of discriminative filters based on the principle of Maximal Marginal Diversity (MMD). It then identifies circular salient regions at various scales by pooling together the responses of the discriminative filters.

Both detectors produce an initial set of thousands of salient candidates for a typical image (see example in Fig. 3a). As in Fergus et al. (2003), we multiply the saliency score of each candidate patch by its scale, thus creating a preference for large image patches, which are usually more informative. A set of N_f high scoring features with limited overlap is then chosen using an iterative greedy procedure. By varying the amount of overlap allowed between selected features we can vary the number of patches chosen: in our experiments we varied N_f between 13 and 513. After their initial detection, selected regions are cropped from the image and scaled down to 11×11 pixel patches. The patches are then normalized to have zero mean and variance of 1. Finally the patches are represented using their first 15 DCT coefficients (not including the DC).

To complete the representation, we concatenate 3 additional dimensions to each feature, corresponding to the x and y image coordinates of the patch, and its scale respectively. Therefore each image I is represented using an unordered set $F(I)$ of 18 dimensional vectors. Since our suggested algorithm's runtime is only linear in the number of image features, we can represent each image using a large number of features, typically in the order of several hundred features per image.

3.2 Model Structure

We consider a part-based model, where each part in a specific image I_i corresponds to a patch feature from $F(I_i)$. Denote the appearance, location and scale components of

[2] We thank Dashan Gao for making his code available to us, and providing useful feedback.

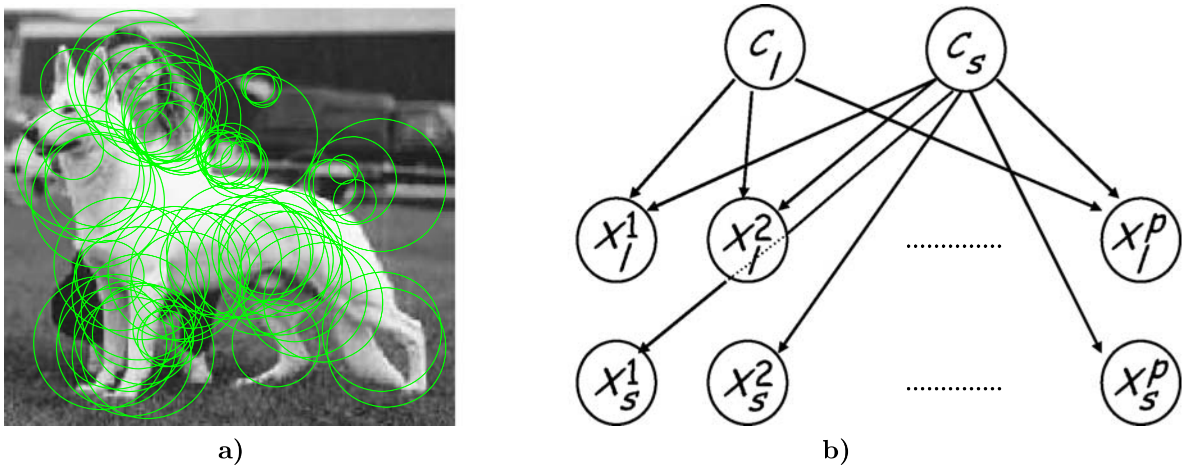


Fig. 3 (Color online) **a** Output of the KB interest point (or feature) detector, marked with *green circles*. **b** A Bayesian network specifying the dependencies between the hidden variables C_l, C_s and the parts scales and locations X_l^k, X_s^k for $k = 1, \ldots, P$. The part appearance variables X_a^k are independent, and so they do not appear in this network

each vector $x \in F(I)$ by x_a, x_l and x_s respectively (with dimensions 15, 2, 1), where $x = [x_a, x_l, x_s]$. We can assume that the appearance of different parts is independent, but this is obviously not the case with the parts' scale and location. However, once we align the object instances with respect to location and scale, the assumption of part location and scale independence becomes reasonable. Thus we introduce a 3-dimensional hidden variable $C = (C_l, C_s)$, which fixes the location of the object and its scale. Our assumption is that the location and scale of different parts is conditionally independent given the hidden variable C, and so the joint distribution decomposes according to the graph in Fig. 3b.

It follows that for a model with P parts, the joint probability of $\{X^k\}_{k=1}^P$ and C takes the form

$$\begin{aligned} &p(\{X^k\}_{k=1}^P, C|\Theta) \\ &\quad = p(C|\Theta) \prod_{k=1}^{P} p(X^k|C, \theta^k) \\ &\quad = p(C|\Theta) \prod_{k=1}^{P} p(X_a^k|\theta_a^k) p(X_l^k|C_l, C_s, \theta_l^k) p(X_s^k|C_s, \theta_s^k). \end{aligned} \tag{5}$$

We assume uniform probability for C and Gaussian conditional distribution for X_a, X_l, X_s as follows:

$$\begin{aligned} &P(X_a^k|\theta_a^k) = G(X_a^k|\mu_a^k, \Sigma_a^k), \\ &P(X_l^k|C_l, C_s, \theta_l^k) = G\left(\frac{X_l^k - C_l}{C_s} \,\middle|\, \mu_l^k, \Sigma_l^k\right), \\ &P(X_s^k|C_s, \theta_s^k) = G(\log(X_s^k) - \log(C_s)|\mu_s^k, \sigma_s^k), \end{aligned} \tag{6}$$

where $G(\cdot|\mu, \Sigma)$ denotes the Gaussian density with mean μ and covariance matrix Σ. We index the model components a, l, s as $1, 2, 3$ respectively, and denote the log of these probabilities by $LG(x_j|C, \mu_j, \Sigma_j)$ for $j = 1, 2, 3$.

3.3 A Model Based Classifier

As discussed in Sect. 2.4.1, the likelihood $P(I|M)$ is given by averaging over all the possible part vectors that can be assembled from the feature set $F(I)$ (see (2)). In our case, we should also average over all the possible values for the hidden variable C. Thus

$$P(I|M) = K_0 \sum_C \sum_{\substack{(x^1, \ldots, x^P) \in F(I)^P \\ x^i \neq x^j \textit{ for } i \neq j}} \prod_{k=1}^{P} P(x^k|C, \theta^k) \tag{7}$$

for some constant K_0.

In order to allow efficient likelihood assessment we make the following approximations

$$P(I|M) \approx K_0 \sum_C \sum_{(x^1, \ldots, x^P) \in F(I)^P} \prod_{k=1}^{P} P(x^k|C, \theta^k) \tag{8}$$

$$\approx K_0 \max_{C, (x^1, \ldots, x^P) \in F(I)^P} \prod_{k=1}^{P} P(x^k|C, \theta^k) \tag{9}$$

$$= K_0 \max_C \prod_{K=1}^{P} \max_{x \in F(I)} P(x|C, \theta^k). \tag{10}$$

Approximation (8) above was discussed earlier in a more general context (see (4)), and it is necessary in order to eliminate the global dependency between parts. In approximation (9), averages are replaced by the likelihood of the best feature vector and best hidden C. This approximation is compelling since as long as the image contains a single object from the class, there are rarely two different likely object interpretations. In addition, working with the best single vector uniquely identifies the object's location and scale, as well as the object's parts. Such unique identification is required for most semantical tasks beyond mere recognition. Finally, the approximated likelihood is decomposed into separate maxima over C and the different parts in (10).

The decomposition of the maximum achieved in (10) is the key to the efficient likelihood computation. We discretize the hidden variable C and consider only a finite grid of locations and scales, with a total of N_c possible values. Using this decomposition the maximum over the $N_c \cdot N_f^P$ arguments can be computed in $O(N_c N_f P)$ operations. However, we cannot optimizing the parameters of such a model by likelihood maximization. Since feature repetition is allowed, the ML solution will choose the same (best) part p times, as shown in Appendix 1.

The natural generative classifier is based on the comparison of the LRT statistic to a constant threshold, and it therefore requires a model of the background in addition to the object model. Modeling general backgrounds is clearly difficult, due to the diversity of objects and scenes that do not share simple common features. We therefore approximate the background likelihood by a constant. Our LRT based classifier thus becomes

$$\begin{aligned} f(I) &= \log P(I|M) - \log(I|BG) - \nu' \\ &= \max_C \sum_{k=1}^{P} \max_{x \in F(I)} \log p(x|C, \theta^k) - \nu \end{aligned} \tag{11}$$

for some constant ν.

4 Discriminative Optimization

Given a set of labeled images $\{I_i, y_i\}_{i=1}^N$, we wish to find a classifier $f(I)$ of the functional form given in (11), which minimizes the exponential loss

$$L(f) = \sum_{i=1}^{N} \exp(-y_i f(I_i)). \tag{12}$$

This is the same loss minimized by the Adaboost algorithm (Schapire and Singer 1999). Its main advantage in our context is that it allows for the determination of the classifier threshold using a closed form formula, as will be described in Sect. 4.1.

We have considered two possible techniques for the optimization of the loss in (12): Boosting and gradient descent. In the boosting context, we view the log probability of a part

$$\max_{x \in F(I)} \log p(x|C, \theta^k)$$

as a weak hypothesis of a specific functional form. However, the classifier form we use in (11) is rather different from the traditional classifiers built by boosting, which typically have the form $f(I) = \sum_{k=1}^{P} \alpha^k h^k(I)$. Specifically, the classifier (11) does not include part weights α^k, it has an extra threshold parameter ν, and it involves a maximization over C, which depends on all the 'weak' hypotheses. The third point is the most problematic, as it requires optimizing over parts with internal dependencies, which is much harder than optimization over independent parts as in standard boosting.

In order to simplify the presentation, we assume in Sect. 4.1 a simplified model with no spatial relations between the parts, and show how the problems of parts weights and threshold parameters are coped with, with minor changes to the standard boosting framework. In Sect. 4.2 we consider the problem of dependent parts, and show how boosting can be naturally extended to handle classifiers as in (11), despite the dependencies between parts due to the hidden variable C. Finally we consider the optimization from a more general viewpoint of gradient descent in Sect. 4.3. This allows us to introduce several enhancements to the pure boosting technique.

4.1 Boosting of a Probabilistic Model

Let us consider a simplified model with parts appearance only (see (6)). We show how such a classifier can be represented as a sum of weighted 'weak' hypotheses in Sect. 4.1.1. We then derive the boosting algorithm as an approximate gradient descent in Sect. 4.1.2. This derivation is slightly simpler than similar derivations in the literature, and provides the basis for our treatment of related parts, introduced in Sect. 4.2. In Sect. 4.1.3 we show how the threshold parameter in our classifier can be readily optimized.

4.1.1 Functional Form of the Classifier

When there are no relations between parts, the classifier (11) takes the following form

$$f(I) = \sum_{k=1}^{P} \max_{x \in F(I)} \log p(x_a|\theta_a^k) - \nu. \tag{13}$$

This classifier is easily represented as a sum of weak hypotheses $f(I)=\sum_{k=1}^{P} h^k(I)$ where

$$h^k(I)=\max_{x\in F(I)} \log G(x_a|\theta_a^k)-\nu^k \tag{14}$$

and $\nu=\sum_{k=1}^{P}\nu^k$. Weak hypotheses in this form can be viewed as soft classifiers.

We next represent the classifier in an equivalent functional form in which the covariance scale is transformed to part weight. Now $f(I)=\sum_{k=1}^{P}\alpha^k h^k(I)$ where $h^k(I)$ takes the form

$$h^k(I)=\max_{x\in F(I)} \log G(x_a|\eta_a^k,\Sigma_a^k)-\nu^k,\quad |\Sigma_a^k|=1. \tag{15}$$

The equivalence of these forms is shown in Appendix 2.

4.1.2 Boosting as Approximate Gradient Descent

Boosting is a common method which learns a classifier of the form $f(x)=\sum_{k=1}^{P}\alpha^k h^k(x)$ in a greedy fashion. Several papers (Friedman et al. 2000; Mason et al. 2000) have presented boosting as a greedy gradient descent of some loss function. In particular, the work of Mason et al. (2000) has shown that the Adaboost algorithm (Freund and Schapire 1996; Schapire and Singer 1999) can be viewed as a greedy gradient descent of the exp loss of Eq. (12), in L^2 function space. In Friedman et al. (2000) Adaboost is derived using a second order Taylor approximation of the exp loss, which leads to repetitive least square regression problems. We suggest here another variation of the derivation, similar to Friedman et al. (2000) but slightly simpler. All three approaches lead to an identical algorithm (the discrete Adaboost Freund and Schapire 1996) when the weak learners are binary with the range $\{+1,-1\}$. For weak learners with a continuous output, our approach and the approach of Mason et al. (2000) culminates in the same algorithm, e.g. Adaboost with confidence intervals (Schapire and Singer 1999). However, our approach is simpler, and is later used to derive a boosting version for a model with dependent parts.

Specifically, we derive Adaboost by considering the first order Taylor expansion of the exp loss function. In what follows and throughout this paper, we use superscripts to indicate the boosting round in which a quantity is measured. At the pth boosting round, we wish to extend the classifier f by $f^p(x)=f^{p-1}(x)+\alpha^p h^p(x)$. We first assume that α^p is infinitesimally small, and look for an appropriate weak hypothesis $h^p(X)$. Since α^p is small, we can approximate (12) using the first order Taylor expansion.

To begin with, we differentiate $L(f)$ w.r.t. α^p

$$\frac{dL(f)}{d\alpha^p}=-\sum_{i=1}^{N}\exp(-y_i f(x_i))y_i h^p(x_i). \tag{16}$$

We denote $w_i=\exp(-y_i f(x_i))$, and derive the following Taylor expansion

$$L(f^p)\approx L(f^{p-1})-\alpha^p\sum_{i=1}^{N} w_i^{p-1}y_i h^p(x_i). \tag{17}$$

Assuming $\alpha^p>0$, the steepest descent of $L(f)$ is obtained for some weak hypothesis h^p which maximizes the weighted correlation score

$$S(h^p(x))=\sum_{i=1}^{N} w_i^{p-1}y_i h^p(x_i). \tag{18}$$

This maximization is done by a weak learner, getting as input the weights $\{w_i^{p-1}\}_{i=1}^N$ and the labeled data points. After the determination of $h^p(x)$, the coefficient α^p is determined by the direct optimization of the loss from Eq. (12). This can be done in closed form only for binary weak hypotheses with output in the range of $\{1,-1\}$. In the general case numeric methods are employed, such as line search (Schapire and Singer 1999).

4.1.3 Threshold Optimization

Maximizing the linear approximation (17) can be problematic when unbounded weak hypotheses are used. In particular, optimizing this criterion w.r.t. to the threshold parameter in hypotheses of the form (14) is ill-posed. Substituting (14) into criterion (17), we get the following expression to optimize:

$$\begin{aligned} S(h)&=\sum_{i=1}^{N} w_i y_i\Big(\max_{x\in F(I)}\log G(x_i|\mu_a,\Sigma_a)-\nu\Big)\\ &=C+\Big(\sum_{i:y_i=-1} w_i-\sum_{i:y_i=1} w_i\Big)\nu \end{aligned} \tag{19}$$

where C is independent of ν. If $\sum_{i:y_i=-1} w_i-\sum_{i:y_i=1} w_i\neq 0$, $S(h)$ can be increased indefinitely by sending ν to $+\infty$ or $-\infty$. Such a choice of ν clearly doesn't improve the original (exact) loss (12).

The optimization of the threshold should therefore be done by considering (12) directly. It is based on the following lemma:

Lemma 1 *Consider a function $f: I\to R$. We wish to minimize the loss (12) of the function $\tilde{f}=f-\nu$ where ν is a constant. Assume that there are both labels $+1$ and -1 in the data set.*

1. *An optimal ν^* exists and is given by*

$$\nu^*=\frac{1}{2}\log\left[\frac{\sum_{\{i:y_i=-1\}}^{N}\exp(f(I_i))}{\sum_{\{i:y_i=1\}}^{N}\exp(-f(I_i))}\right]. \tag{20}$$

2. *The optimal $\tilde{f}^* = f - \nu^*$ satisfies*

$$\sum_{\{i:y_i=1\}}^{N} \exp(-\tilde{f}^*(I_i)) = \sum_{\{i:y_i=-1\}}^{N} \exp(\tilde{f}^*(I_i)). \tag{21}$$

3. *The optimal loss $L(f - \nu^*)$ is*

$$2\left[\sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i)) \cdot \sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i))\right]^{\frac{1}{2}}. \tag{22}$$

The lemma is proved by direct differentiation of the loss w.r.t. ν, as sketched in Appendix 3.

We use this lemma to determine the threshold after each round of boosting, when $f^p(I) = f^{p-1}(I) + \alpha^p h^p(I)$. Equation (20) gives a closed form solution for ν once $h^p(I)$ and α^p have been chosen. Equation (22) gives the optimal score obtained, and it is useful when efficient numeric search for α^p is required. Finally, property (21) implies that after threshold update, the coefficient of ν in (19) is nullified (the slope of the linear approximation is 0 at the optimum). Hence optimizing the threshold before round p assures that the score $S(h^p)$ does not depend on ν^p. We optimize the threshold in our algorithm during initialization, and after every boosting round (see Algorithm 1). The weak learner can therefore effectively ignore this parameter when choosing a candidate hypothesis.

4.2 Relational Model Boosting

We now extend the boosting framework to handle dependent parts in a relational model of the form (11). We introduce part weights into the classifier by applying the transformation described in (15) to the three model ingredient described in (6), i.e. appearance, location and scale. The three new weights are summed into a single part weight, leading to the following classifier form

$$f(I) = \max_C \sum_{k=1}^{P} \alpha^k h^k(I,C) - \nu \tag{23}$$

where for $k = 1, \dots, P$

$$\begin{aligned}
&h^k(I,C) = \max_{x \in F(I)} g^k(I,C),\\
&g^k(I,C) = \sum_{j=1}^{3} \frac{\lambda_j^k}{\sum_{j=1}^{3} \lambda_j^k} LG(x_j^k | C, \mu_j^k, \Sigma_j^k),\\
&|\Sigma_j^k| = 1, \quad \lambda_j^k > 0,\ j = 1,2,3.
\end{aligned} \tag{24}$$

In this parametrization α^k is the sum of component weights and $\lambda_i / \sum_{j=1}^{3} \lambda_j$ measures the relative weights of the appearance, location and scale. Thus, given an image I, the computation of f requires the computation of the accumulated log-likelihood and its hidden center optimizer, denoted as follows

$$\begin{aligned}
&ll(I,C) = \sum_{k=1}^{p} \alpha^k h^k(I,C),\\
&C^* = \arg\max_C ll(I,C).
\end{aligned} \tag{25}$$

In order to allow tractable maximization over C, we discretize it and consider only a finite grid of locations and scales with N_c possible values. Under these conditions, the computation of ll and C^* amounts to standard MAP message passing, requiring $O(PN_f N_c)$ operations.

Our suggested boosting method is presented in Algorithm 1. We derive it by replicating the derivation of standard boosting in (16–18). For f of the form (23), the derivative of $L(f)$ w.r.t. α_p is now

$$\frac{dL(f)}{d\alpha^p} = -\sum_{i=1}^{N} w_i y_i h^p(I_i, C_i^*) \tag{26}$$

and using the Taylor expansion (17) we get

$$L(f^p) = L(f^{p-1}) - \alpha^p \sum_{i=1}^{N} w_i^{p-1} y_i h^P(I_i, C_i^{*,p-1}). \tag{27}$$

In analogy with the criterion (18), the weak learner should now get as input $\{w_i^{p-1}, C_i^{*,p-1}\}_{i=1}^{N}$ and try to maximize the score

$$S(h^p) = \sum_{i=1}^{N} w_i^{p-1} y_i h^P(I_i, C_i^{*,p-1}). \tag{28}$$

This task is not essentially harder than the weak learner's task in standard boosting, since the weak learner 'assumes' that the value of the hidden variable C is known and set to its optimal value according to the previous hypotheses. In the first boosting round, when $C^{*,p-1}$ is not yet defined, we only train the appearance component of the hypothesis. The relational components of this part are set to have low weights and default values.

Choosing α^p after the hypothesis $h^p(I,C)$ has been chosen is more demanding than in standard boosting. Specifically, α^p should be chosen to minimize

$$L(\max_C [ll^{p-1}(I,C) + \alpha^p h^p(I,C)] - \nu^*). \tag{29}$$

Algorithm 1 Relational model boosting

```
Given $\{(I_i, y_i)\}_{i=1}^N$, $y_i \in \{-1, 1\}$, initialize:
$ll(i, c) = 0, \quad i = 1, \dots, N$, c in a predefined grid
$\nu = \frac{1}{2} \log \frac{\#\{y_i = -1\}}{\#\{y_i = 1\}}$
$w_i = \exp(y_i \cdot \nu), \quad i = 1, \dots, N$
$w_i = w_i \Big/ \sum_{i=1}^N w_i$
For $k = 1, \dots, P$
    1. Use a weak learner to find a part hypothesis $h^k(I, C)$ which maximizes
       $S(h) = \sum_{i=1}^N w_i y_i h(I_i, C_i^*)$
       (see text for special treatment of round 1).
    2. Find optimal $\alpha^k$ by minimizing
       $\sum_{\{i:y_i=1\}}^N \exp(-f^0(I_i)) \cdot \sum_{\{i:y_i=-1\}}^N \exp(f^0(I_i))$
       where $f^0(I) = \max_C ll(I, C) + \alpha h^k(I, C))$.
       Update $ll$ and the optimal center $C^*$
       $ll(i, c) = ll(i, c) + \alpha^k h(i, c)$
       $[f^0(I_i), C_i^*] = \max, \arg\max_c ll(i, c)$
    3. Update $f(I_i)$ and the weights $\{w_i\}_{i=1}^N$
       $\nu = \frac{1}{2} \log \left[ \frac{\sum_{\{i:y_i=-1\}}^N \exp(f^0(I_i))}{\sum_{\{i:y_i=1\}}^N \exp(-f^0(I_i))} \right]$
       $f(I_i) = f^0(I_i) - \nu$
       $w_i = \exp(-y_i f(I_i))$
       $w_i = w_i / \sum_{i=1}^N w_i$
Output the final hypothesis
$f(I) = \max_C \sum_{k=1}^P \alpha_k h_k(I, C) - \nu$
```

Since the optimal value of C depends on α^p, its inference should be repeated whenever a different value is considered for α^p (although the messages $h^p(I, C)$ can be computed only once). After finding the maximum over C, the loss with the optimal threshold can be computed using (22). The search for the optimal α^p can be done using any line search algorithm, and we implement it using gradient descent as described next in Sect. 4.3.

4.3 Gradient Descent

In this section we combine the relational boosting from Sect. 4.2 with elements from a more general gradient descent perspective. In Sect. 4.3.1 we describe our implementation of Algorithm 1, in which the weak learner and the part weight optimization are gradient based. In Sect. 4.3.2 we suggest to supplement Algorithm 1 with feedback elements in the spirit of more traditional gradient descent algorithms. Algorithm 2 presents the resulting algorithm for part optimization.

4.3.1 Gradient-Based Implementation

Current boosting-based object recognition approaches use a version of what we call "selection-based" weak learners (Opelt et al. 2004b; Viola and Jones 2001; Murphy et al. 2003). The weak hypotheses family is finite, and hypotheses are based on a predefined feature set (Viola and Jones 2001) or on the set of features extracted from the training images (Opelt et al. 2004b; Murphy et al. 2003). The weak learner computes the weighted correlation for all the possible hypotheses and returns the best scoring one. Weak learners of this type, considered in the current paper, sample features from object images (exhaustive search is too expensive computationally); they build part hypotheses based on the feature and the current estimate of the hidden center C^* in the feature's image. However, as a single feature cannot reliably determine the relative weights of the different part components (the covariance scale of appearance, location and scale), several values of these parameters are considered for each feature.

As an alternative, we have considered a second type of weak learners, which we call "gradient-based". A "gradient-based" weak learner uses a hypothesis supplied by the selection learner as its starting point, and tries to improve its score using gradient ascent. Unlike the selection based weak learner, the gradient-based weak learner is not limited to parts based on natural image features, as it searches in the continuum of all possible part models. The update rules are based on the relevant gradient, i.e. the derivative of the score $S(h^p)$ w.r.t. the part parameters, given by the weighted sum

$$\Delta\theta^p = \eta \frac{dS(h^p)}{d\theta^p} = \eta \sum_{i=1}^N w_i^{p-1} y_i \frac{dh^p(I_i, C_i^{*,p-1})}{d\theta^p}$$

$$= \eta \sum_{i=1}^N w_i^{p-1} y_i \frac{dg^p(I_i, C_i^{*,p-1}, x_i^{*,p})}{d\theta^p} \qquad (30)$$

where $\eta > 0$ is a step size and $x_i^{*,p}$ is the best part candidate in image i

$$x_i^{*,p} = \arg\max_{x \in F(I_i)} g^p(I_i, C_i^{*,p-1}).$$

Specifically, $\theta^p = \{\mu_j^p, \Sigma_j^p, \lambda_j^p\}_{j=1}^3$. In order to keep $\Sigma_j^p > 0$ during the gradient descent (for $j = 1, 2$) we re-parameterize $(\Sigma_j^p)^{-1} = (A_j^p)^t A_j^p$ and descend w.r.t. A_j^p. Dropping the superscript p from all the variables and parameters and denoting $g_i = g^p(I_i, C_i^{*,p-1}, x_i^{*,p})$, the gradients in (30) are given by

$$\begin{aligned} \frac{dg_i}{d\mu_j} &= -\lambda_j \Sigma_j^{-1}(z_{i,j}^* - \mu_j), \\ \frac{dg_i}{dA_j} &= -\lambda_j A_j (z_{i,j}^* - \mu_j)(z_{i,j}^* - \mu_j)^t, \\ \frac{dg_i}{d\lambda_j} &= 1\Big/\left(\sum_{j=1}^3 \lambda_j\right)[LG(x_{i,j}^* | C_i^*, \mu_j, \Sigma_j) \\ &\quad - \sum_{j=1}^3 \frac{\lambda_j}{\sum_{j=1}^3 \lambda_j} LG(x_{i,j}^* | C_i^*, \mu_j, \Sigma_j)]. \end{aligned} \tag{31}$$

Above $x_{i,j}^*$ stands for the jth component (appearance, location or scale) of x_i^* and

$$\begin{aligned} z_{i,1}^* &= x_{i,1}^*, \\ z_{i,2}^* &= (x_{i,2}^* - (C_l)_i^*)/(C_s)_i^*, \\ z_{i,3}^* &= \log(x_{i,3}^*) - \log((C_s)_i^*). \end{aligned}$$

The constraints of $|\Sigma_j| = 1$ and $\lambda_j \geq 0$ were enforced after each gradient step. Since the gradient depends on the best part candidates according to the current model, the gradient dynamics iterates between gradient steps in the parameters θ^p and the re-computation of the best part candidates $\{x_i^{*,p}\}_{i=1}^N$. Pseudo code is given in Step 1 of Algorithm 2.

We also use gradient descent dynamics to implement the line search for the optimal part weight α^p. This search method is based on slow, gradual changes in the value of α^p, and hence it allows us to experiment with a feedback mechanism (see Sect. 4.3.2). The gradient of the loss w.r.t. α^p is given in (26). Notice that the gradient depends on $\{C_i^*\}_{i=1}^N$ and $\{w_i\}_{i=1}^N$, and both are functions of α^p. Hence the gradient dynamics in this case iterates between gradient steps of α^p, inference of $\{C_i^*\}_{i=1}^N$, and updates of $\{w_i\}_{i=1}^N$. This loop is instantiated in Step 3 of Algorithm 2. The loop must be preceded by the computation of the messages $h(i, c)$ in Step 2.

Algorithm 2 Optimization of part p

```
Input : $F(I_i), y_i, w_i, C_i^*, \quad i = 1, \dots, N$
        $ll(i, c), \quad i = 1, \dots, N, \ c = 1, \dots, N_c$
initialize weak hypothesis using a selection learner :
Choose $\theta = \lambda_j, \mu_j, \Sigma_j, \quad j = 1, \dots, 3, \ \alpha = 0$
Set $[h(i, C_i^*), x^*(i)] = \max, \arg\max_{x \in F(I_i)} g(x, C_i^*)$
where $g(x, c) = \sum_{j=1}^3 \frac{\lambda_j}{\sum_{j=1}^3 \lambda_j} LG(x_j | c, \mu_j, \Sigma_j)$
Loop over 1, 2, 3 $K_1$ iterations
  1. Loop over a, b $K_2$ iterations ($\theta$ optimization)
     (a) Update weak hypothesis parameters
           $\theta = \theta + \eta \sum_{i=1}^N w_i y_i \frac{dg(x_i^*, c_i^*)}{d\theta}$
     (b) Update best part candidates for all images
           $[h(i, C_i^*), x_i^*] = \max, \arg\max_{x \in F(I_i)} g(x, C_i^*)$
  2. Compute for all $i, c$ $h(i, c) = \max_{x \in F(I_i)} g(x, c)$
  3. Loop over a, b, c $K_3$ iterations ($\alpha$ optimization)
     (a) Update $\alpha$: $\alpha = \alpha + \eta \sum_{i=1}^N w_i y_i h(i, C_i^*)$
     (b) Update hidden center for all images
           $[f^0(I_i), C_i^*] = \max, \arg\max_c ll(i, c) + \alpha h(i, c)$
     (c) Update $f(I_i)$ and the weights
           $\nu = \frac{1}{2} \log \left[ \frac{\sum_{\{i:y_i=-1\}}^N \exp(f^0(I_i))}{\sum_{\{i:y_i=1\}}^N \exp(-f^0(I_i))} \right]$
           $f(I_i) = f^0(I_i) - \nu$
           $w_i = \exp(-y_i f(I_i)), \quad w_i = w_i \Big/ \sum_{i=1}^N w_i$
Set $ll^p(i, c) = ll(i, c) + \alpha h(i, c)$
Return $\theta, w_i, C_i^*, ll^p(i, c), \quad i = 1, \dots, N, c = 1, \dots, N_c$.
```

4.3.2 Gradient-Based Extension

When the determination of both θ^p and α^p are gradient based, the boosting optimization at round p essentially makes a specific control choice for a unified gradient descent algorithm which optimizes α^p and θ^p together. A more traditional gradient descent algorithm can be constructed by (1) differentiating $L(f)$ directly instead of using its Taylor approximation, and (2) iterating small gradient steps on both α^p and θ^p in a single loop, instead of two separate loops as suggested by boosting. In boosting, the optimization of θ^p is done before setting α^p and there is no feedback between them. Such feedback is plausible in our case, since any change in α^p may induce changes in C^* for some images, and can therefore change the optimal part model of $h^p(I, C)$.

We considered the update steps required for gradient descent of the exact loss (12), without the Taylor approximation implied by the boosting strategy. The gradient of α^p (26) and its treatment remain the same, as α^p was optimized w.r.t. the exact loss in the boosting strategy as well. The gradient w.r.t. θ^p is

$$\frac{dL(f)}{d\theta^p} = \sum_{i=1}^{N} w_i^p y_i \frac{dh^p(I_i, C_i^{*,p})}{d\theta^p}. \qquad (32)$$

While this expression is very similar to (30), there is a subtle difference between them. In (32) w_i and C_i^* are no longer constant as they were in (30), but depend on θ^p and α^p. Exact gradient descent therefore requires the recomputation of w_i, C_i^* at each gradient iteration, which is computationally expensive.

We have experimented in the continuum between the 'boosting' and the 'gradient descent' flavors using Algorithm 2, which encloses the optimization loops of h^p and α^p in a third 'feedback' loop. Setting the outer loop counter K_1 to 1 we get the boosting flavor, i.e., Algorithm 2 implements an inner loop step in Algorithm 1. Setting K_1 to some large value and $K_2 = 1$, $K_3 = 1$, we get exact gradient descent flavor. We found that a good trade-off between complexity and performance is achieved with a version which is rather close to boosting, but still repeats the optimization of α^p and h^p several times to allow mutual cross-talk during the estimation of these parameters. Thus, our final optimization algorithm involves repeated, sequential calls of Algorithm 2.

5 Experimental Results

We tested our algorithm in recognition tasks using the Caltech datasets (Fergus et al. 2003), which are publicly available,[3] as well as three more challenging data sets we have collected specifically for this evaluation. The former are used as a common benchmark, while the latter are designed to measure the performance limits of the algorithm by challenging it with fairly hard conditions. Localization performance was evaluated using a common benchmark for this task (Agarwal et al. 2004).[4] The datasets are described in Sect. 5.1. In Sect. 5.2 we discuss the various algorithm parameters. Recognition results are presented in Sect. 5.3. In Sect. 5.4 we report the results of additional experiments, studying the contribution to recognition performance of several modeling factors in isolation. Finally, we report localization results in Sect. 5.5.

5.1 Datasets

We compare our recognition results with other methods using the Caltech datasets. Four datasets are used: Motorcycles (800 images), Cars rear (800), Airplanes (800) and Faces (435). These datasets contain relatively small variance in scale and location, and the background images do not contain objects similar to the class objects. In order to test recognition performance under harder conditions, we compiled 3 new datasets with matching backgrounds.[5] These datasets contain images of Chairs (800 images), Dogs (500) and Humans (593), and are briefly described below. The data sets mentioned above are not well suited for localization experiments, since the objects of interest are usually placed in the center of the image. We nevertheless present here localization results for the data sets which include a bounding-box information, which are Airplanes, Motorcycles and Faces. To better estimate the localization ability of our algorithm we used the UIUC cars side benchmark (Agarwal et al. 2004). The training set here is composed of 550 cars images and 500 background images. The test set includes 170 images, containing altogether 200 cars, with ground truth bounding boxes.

In the Chairs and Dogs datasets, the objects are seen roughly from the same pose, but include large inner class variability, as well as some variability in location and scale. For the Chairs dataset we compiled a background dataset of Furniture which contained images of tables, beds and bookcases (200,200,100 images respectively). When possible (for tables and beds), images were aligned to a viewpoint isomorphic to the viewpoint of the chairs. As background for the Dogs dataset, we compiled two animal datasets: 'Easy Animals' contains 500 images of animals not similar to Dogs; 'Hard Animals' contains 250 images from the 'Easy Animals' dataset, and an additional set of 250 images of four-legged animals (such as horses and goats) in a pose isomorphic to the Dogs.

The Humans dataset was designed to include large variability in location, scale and pose. The data contains images of 25 different people. Each person was photographed in 4 different scales (each 1.5 times larger than its predecessor), at various locations and with several articulated poses of the hands and legs. For each person there are several images in which s/he is partially occluded. For this dataset we created a background dataset of 593 images, showing the sites in which the Humans images were taken. Figure 4 shows several images from our datasets.

5.2 Algorithm Parameters

We have run a series of preliminary experiments, in order to tune the weak learners' parameters and compare the results

[3] http://www.robots.ox.ac.uk/∼vgg/data.

[4] http://www.pascal-network.org/challenges/VOC/#UIUC.

[5] The datasets are available at http://www.cs.huji.ac.il/~aharonbh/.

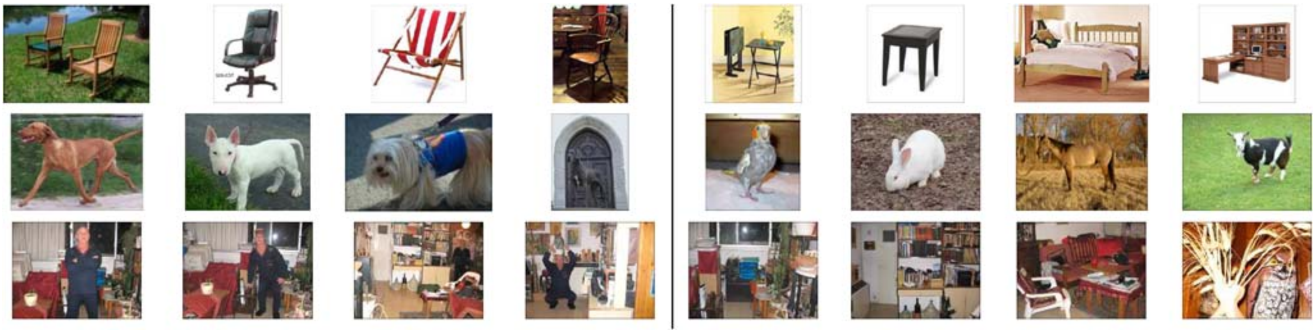


Fig. 4 (Color online) Images from the Chairs, Dogs and Humans datasets and their corresponding backgrounds. Object images appear on the *left*, background images on the *right*. In the *second row*, the *two leftmost* background *images* are of 'easy animals' and next are two 'hard animals' images. In the *third row*, the *two leftmost* object *images* belong to the easier image subset. The *next two images* are hard due to the person's scale and pose

when using selection-based vs. gradient-based weak learners. The parameters of the selection based weak learner include the number of image patches it samples, and the number of location/scale models used for each sampled patch. The parameters of the gradient based learner include the step size and stop condition. The gradient based learner is not limited to hypotheses based on object images, and in many cases it chooses exaggerated appearance and location models for the part in order to enhance discriminative power. In the exaggerated appearance models, the brightness contrast is enhanced and the mean patch looks almost like a Black&White mask (see examples in Fig. 5b). This tendency for exaggerated appearance model is enhanced when the weight of the location model is relatively weak.

In exaggerated location models, parts are modeled as being much farther from the center than they are in real objects. An example is given in Fig. 7, showing a chair model where the tip of the chair's leg is located below its mean location in most images. Still, in most cases gradient based learners give lower error rates than their purely selection-based competitors. Some examples are given in Fig. 5a. We hence used gradient based learners in the rest of the recognition experiments.

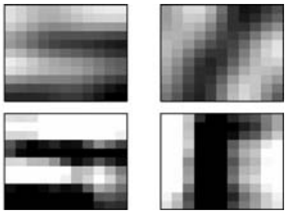
We have also experimented with the learning of covariance matrices for the appearance and location models. As stated in Sect. 4.3.1, we have implemented gradient dynamics for the square root of the covariance matrix. However, we have still observed too much over-fitting in the estimation of the covariance matrices in our experiments. These additional degrees of freedom tended not to improve the test results, while achieving lower training error. The problem was more serious with the appearance covariance matrix, where we have sometimes observed reduced performance, and the emergence of unstable models with covariance matrices close to singular. As a result, in the following experiments we fix the covariance matrices to σI. We only learn the covariance scale, which in our model determines the part and component weight parameters.

In the recognition experiments reported in Sect. 5.3, we constructed models with up to 60 parts using Algorithm 2, with control parameters of $K_1 = 60$, $K_2 = 100$, $K_3 = 4$. Each image was represented using at most $N_f = 200$ features (KB detector) or $N_f = 240$ features (GV detector). The hidden center location values were an equally spaced grid of 6×6 positions over the image. The hidden scale center had a single value, or 3 different values with a ratio of 0.63 between successive scales, resulting in a total of $N_c = 36, 108$ values respectively. We randomly selected half of the images from each dataset for training and used the remaining half for testing.

For the localization experiments reported in Sect. 5.5 we changed several important parameters of the learning process. Model accuracy is more important for this task, and we therefore learn smaller models with $P = 40$ parts, but using a finer location grid of 10×10 possible locations ($N_C = 100$) and $N_f = 400$ features extracted per image. As noted above, the dynamics of the gradient-based location model tends to produce 'exaggerated' models, in which parts are located too far from the objects center. This tendency dramatically reduces the utility of the model for localization. We therefore eliminated the gradient location dynamics in this context, and modified only the part appearance using gradient descent. We found experimentally that increasing the weight of the location component uniformly for all the parts improves the localization results considerably. In the experiments reported below, we multiply the location component weights λ_2^k, $k = 1, \dots, P$ (see (23) for their definition) by a constant factor of 10. Probabilistically, this amounts to smaller location covariance and hence to stricter demands on the accuracy of parts relative locations. Finally, parts without location component (when the location component weight is 0) are ignored; these parts do not convey localization information, and therefore add irrelevant 'noise' to the MAP score.

Data Name	Selection Learner	Gradient Learner
Motorbikes	7.2	6.9
Cars Rear	6.8	2.3
Airplanes	14.2	10.3
Faces	7.9	8.35

a)



b)

Fig. 5 **a** Comparison of error rates obtained by selection-based and gradient based weak learners on the Caltech data sets. The results presented were obtained for object models without a location component, i.e. the models are not relational and classification is based on part appearance alone. **b** Examples of parts from motorcycle models learnt using the selection-based learner (*top*) and the gradient-based learner (*bottom*). The images present reconstructions from the 15 DCT coefficients of the mean appearance vector. The parts presented correspond to motorcycles seat (*left*) and wheel (*right*). Clearly, the parts learnt by the gradient learner have much sharper contrasts

5.3 Recognition Results

As a general remark, we note that our algorithm tends to learn models in which most features have clear semantics in terms of object's parts. Examples of learnt models can be seen in Figs. 6 and 7. In the dog example we can clearly identify parts that correspond to the head, back, legs (both front and back), and the hip. Typically 40–50 out of the 60 parts are similar in quality to the ones shown. The location models are gross, and sometimes exaggerated, but clearly useful. Analysis of the part models shows that in many cases, a distinguished object part (e.g., a wheel in the motorcycle model, or an eye in the face model) is modeled using a number of model parts (12 for the wheel, 10 for the eye) with certain internal variation. In this sense our model seems to describe each object part using a mixture model.

In Table 1 we present our results on the Caltech database benchmark and compare them to several generative (Fergus et al. 2003, 2005; Loeff et al. 2005) and discriminative (Opelt et al. 2004b; Dorkó and Schmid 2005) approaches. All the methods compared learn object class recognition from unordered sets of features, obtained from an interest point detector. Following Fergus et al. (2003), the motorbikes, airplanes and faces datasets were tested against office background images, and the Cars rear dataset was tested against road background images. We used the exact train and test indices used by Fergus et al. (2003) and the same Kadir and Bradey (KB) feature detector. Comparison is hence easier between our method and methods (Fergus et al. 2003;[6] Opelt et al. 2004b;[7] Loeff et al. 2005) which use the same KB feature detector. The results reported were obtained without modeling scale, since it did not improve classification results when using the KB detector. This may be partially explained by noting that the Caltech datasets contain relatively small variance in scale. Error rates for our method were computed using the threshold learnt by our boosting algorithm.

In this recognition task our method seems to be superior to Fergus et al. (2003) and Opelt et al. (2004b), comparable to Fergus et al. (2005), and inferior to Loeff et al. (2005) and Dorkó and Schmid (2005). We believe that our advantage over the methods of Fergus et al. (2003) and Opelt et al. (2004b) can be attributed to the small number of parts in the former and the neglect of spatial part information by the latter. It seems that while the method presented in Fergus et al. (2005) is somewhat problematic w.r.t. its learning method (see Sect. 2.4.2), it compensates for it by using a rich combination of 3 different interest detectors. The methods (Loeff et al. 2005; Dorkó and Schmid 2005) are usually preferable to our suggested method in the binary object recognition problem. These methods, however, are less suitable for the localization task, and they were not tested on localization benchmarks.

We used the Chairs and Dogs datasets to test the sensitivity of the algorithm to visual similarity between object and background images. We trained the Chairs dataset against the Caltech office background dataset, and against the furniture dataset described above. The Dogs dataset was trained against 3 different backgrounds datasets: Caltechs 'office' background, 'Easy Animals' and 'Hard Animals'. The results are summarized in Table 2. As can be seen, our algorithm works well when there are large differences between the object and background images. However, it fails to discriminate, for example, dogs from horses.

We used the Humans dataset to test the algorithm's sensitivity to variations in scale and object articulations. In order to obtain reasonable results on this hard dataset, we had to reduce scale variability to 2 scales and restrict the variability in pose to hand gestures only—we denote this dataset by 'Humans restricted' (355 images). The results are shown in Table 2.

[6] The results reported in Fergus et al. (2003) (except for the cars data base) were achieved using manually scale-normalized images, while the our methods did not rely on any such rescaling.

[7] In Opelt et al. (2004a), this approach was reported to give better results using segmentation based features. We did not include these results here since we focus on comparing different learning algorithms using similar features. see also discussion in Sect. 6.

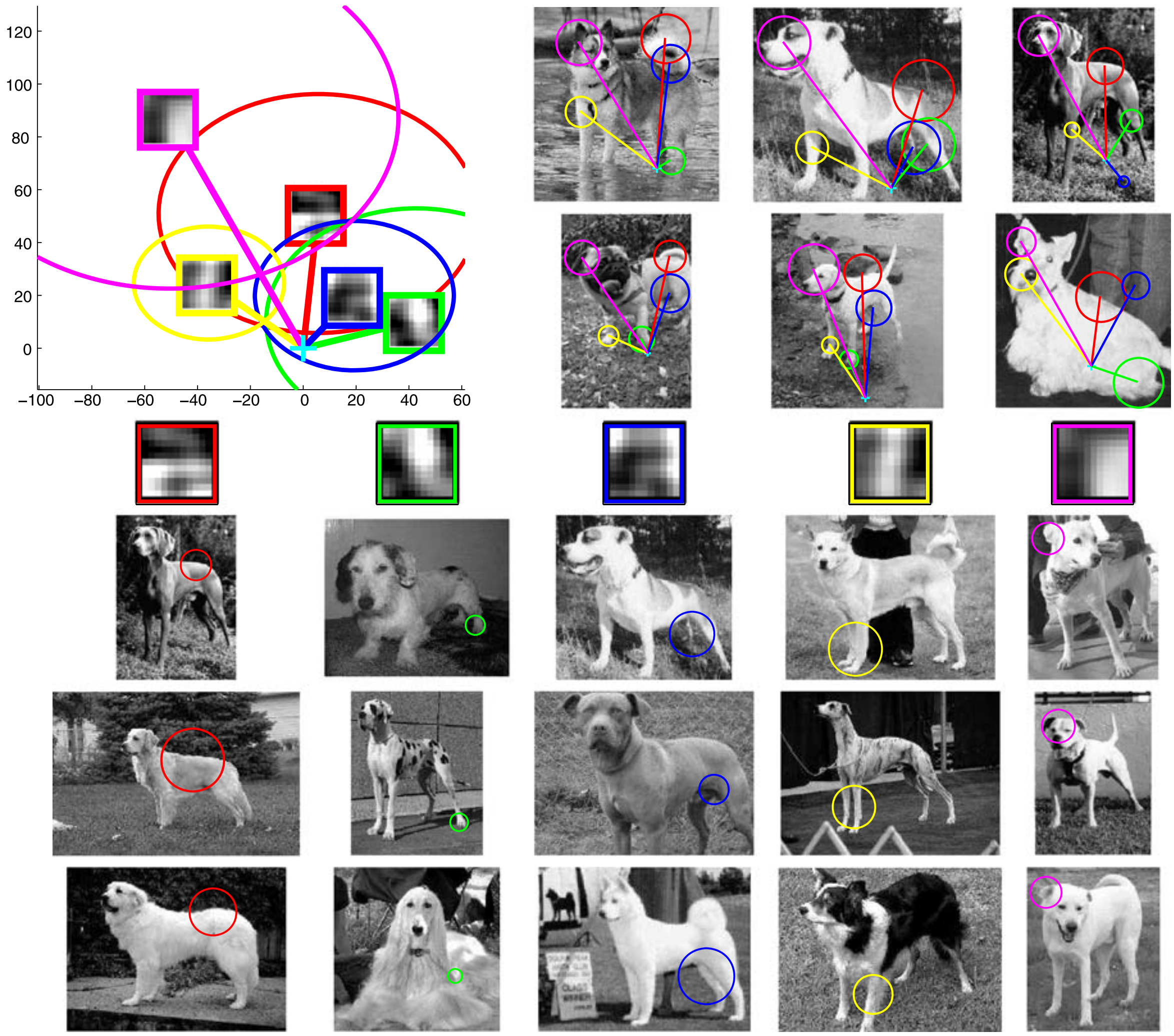


Fig. 6 (Color online) 5 parts from a dog model with 60 parts. The *top left drawing* shows the modeled locations of the 5 parts. Each part's mean location is surrounded by the 1 std line. The *cyan cross* indicates the location of the hidden 'center'. The *top right pictures* show dog test images with the model implementation found. All these dogs were successfully identified except for the one on the *right-bottom corner*. Below the location model, the parts' mean appearance patches are shown. The last three rows present parts implementations in the 3 test images that got the highest part likelihood. Each column presents the implementations of the part shown *above* the column. The parts have clear semantic meaning and repetitive locations on the dogs back, hind leg, joint of the hind leg, front leg and head. Most other parts behave similarly to the ones shown

The parameters in our model are optimized to minimize training error with respect to a certain background. One may worry that the learnt models describe the background just as well as they describe the object, in which case performance in classification tasks against different backgrounds is expected to be poor. Indeed, from a purely discriminative point of view, there is no reason to believe that the learnt classifier will be useful when one of the classes (the background) changes. To investigate this issue, we used the learnt models to classify object images against various background images not seen by the learning algorithm. We found that the learnt models tend to generalize well to the new classification problem, as seen in Table 3. These results show that the models have 'generative' qualities: they seem to capture the 'essence' of the object in a way that does not really depend on the background used.

5.4 Recognition Performance Analysis

In this section we analyze the contribution to performance of several important modeling factors. Specifically, we consider the contribution of modeling part location and scale, and of increasing the number of model parts and features extracted per image.

5.4.1 Location and Scale Models

The relational components of the model, i.e. the location and scale of the parts, clearly complicate learning considerably,

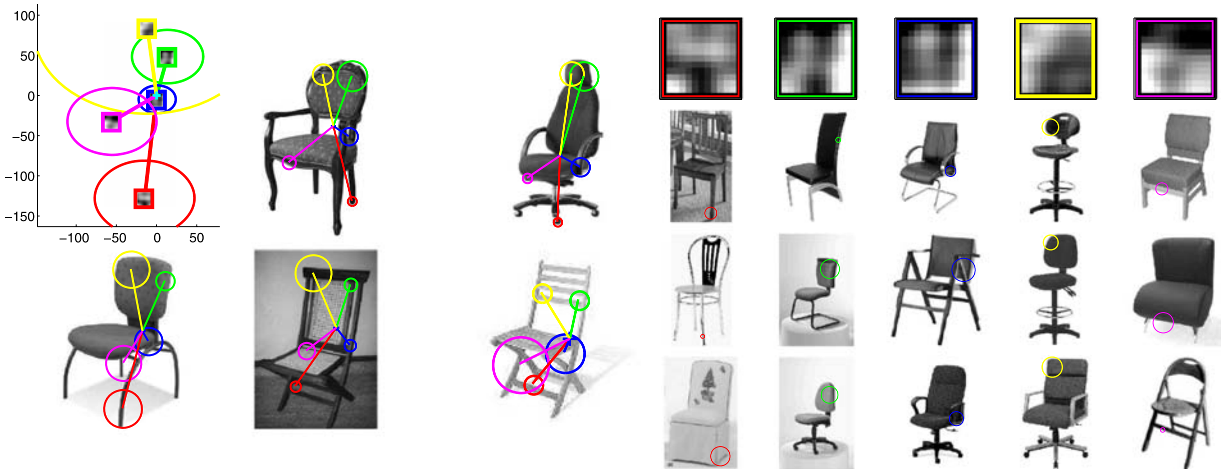


Fig. 7 (Color online) 5 parts from a chair model. The model is presented in the same format as Fig. 6. Model parts represent the tip of the chairs leg (first part), edges of the back (second and forth parts), the seat corner (third) and the seat edge (fifth). The location model is exaggerated: The tip of the chairs leg is modeled as being far below its real mean position in object images

Table 1 Test error rates over the Caltech dataset, showing the results of our method in 2 conditions—using 7 or 50 parts compared to several other methods. The algorithm's parameters were held constant across all experiments

Data name	Our model 50 parts	Fergus et al. (2003)	Opelt et al. (2004b)	Fergus et al. (2005)	Loeff et al. (2005)	Dorkó and Schmid (2005)
Motorbikes	4.9	7.5	7.8	4.0	3.0	0.5
Cars Rear	0.6	9.7	8.9	12.3	2.0	–
Airplanes	6.7	9.8	11.1	6.8	3.0	1.5
Faces	6.3	3.6	6.5	11.9	1.3	0.9

Table 2 Error rates with the new datasets of Chairs, Dogs and Humans. Results were obtained using the KB detector (see text for more details)

Data	Background	Test error
Chairs	Office	2.23
Chairs	Furniture	15.53
Dogs	Office	8.61
Dogs	Easy Animals	19.0
Dogs	Hard Animals	34.4
Humans	Sites	34.3
Humans (resticted)	Sites	25.9

and it is important to understand if they give any performance gain. Table 4 shows comparative results varying the model complexity. Specifically we present results when using only an appearance model, and when adding location and scale models, using the GV detector (Gao and Vasconcelos 2004).[8] We can see that although the appearance model produces very reasonable results, adding a location model significantly improves performance. The additional contribution of the scale model is relatively minor. Additionally, by comparing the results of our full blown model (A+L+S) to those presented in Tables 1 and 2, we can see that the discriminative GV detector usually provides somewhat better results than those obtained using the generic KB detector.

5.4.2 *Large Numbers of Parts and Features*

When hundreds of features are used per image, many features lie in the background of the image, and learning good parts implicitly requires feature pruning. Figure 8 gives error rates as a function of the number of parts and features. Significant performance gains are obtained by scaling up

[8] Similar experiments with the KB detector yielded similar results, but showed no significant improvement with scale modeling.

these quantities, indicating that the algorithm is able to find good part models even in the presence of many clutter features. This behavior should be contrasted with the generative learning of a similar model in Fergus et al. (2005), where increasing the number of parts and features does not usually lead to improved performance. Intuitively, maximum likelihood learning chooses to model features which are frequent in object images, even if these are simple clutter features from the background, while discriminative learning naturally tends to selects more distinctive parts.

Table 3 Generalization results of some learnt models to new backgrounds. Each row describes results of a single class model trained against a specific background and tested against other backgrounds. Test errors were computed using a sample of 100 images from each test background. The classifiers based on learnt models perform well in most of the new classification tasks. There is no apparent connection between the difficulty of the training background and successful generalization to new backgrounds

Data	Original BG	Motorcycles BG	Airplanes BG	Sites BG
Cars	Road (0.6)	3.0	2.2	6.8
Cars	Office (1.6)	1.0	0.8	6.4
Chairs	Office (2.2)	8.0	1.4	6.2
Chairs	Furniture (15.5)	17.4	4.2	8.4
Dogs	Office (8.6)	10.3	4.0	12.3
Dogs	Easy animals (19.0)	15.7	5.7	7.7

5.5 Localization Results

Locating an object in a large image is much harder than the binary present/absent detection task. The latter problem is tackled in this paper using a limited set of image features, and a crude grid of possible object locations. For localization we use a similar framework in learning, but turn to a more exhaustive search at the test phase. While searching we do not select representative features, but consider instead as part candidates all the possible image patches at every location and several scales. Object center candidates are taken from a dense grid of possible image locations. To search efficiently, we use the methods proposed in Fergus et al. (2005), Feltzenswalb and Huttenlocher (2005), which allow such an exhaustive search in a relatively low computational cost.

Table 4 Errors rates using models of varying complexity. (A) Appearance model alone. (A + L) Appearance and location models. (A + L + S) Appearance, location and scale models. The algorithm's parameters were held constant across all experiments

Data name	A	A + L	A + L + S
Motorbikes	8.1	3.2	3.5
Cars Rear	4.0	1.4	0.6
Airplanes	15.1	15.1	12.1
Faces	6.1	5.2	3.8
Chairs	16.3	10.8	10.9

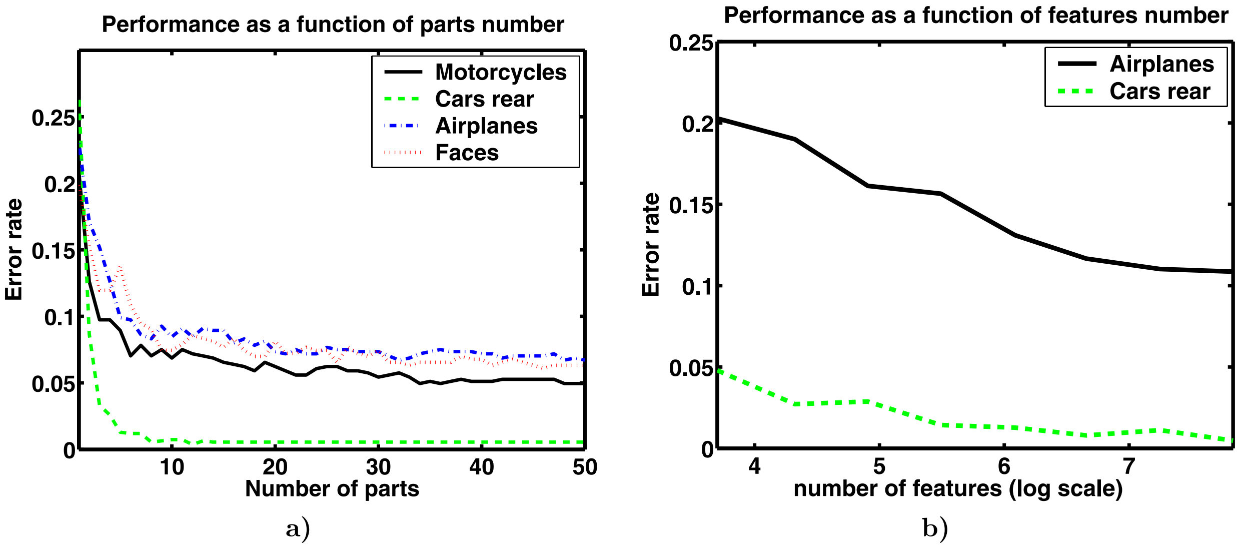


Fig. 8 **a**) Error rate as a function of the number of parts P in the model on the Caltech datasets for $N_f = 200$. **b**) Error rate as a function of the number of image features N_f on the Cars rear (easy) and Airplanes (Relatively hard) Caltech datasets, with $P = 30$. In **b**), the X axis varies between 13 and 228 features in log scale, base 2. All the results were obtained using the KB detector

The model is applied to an image following a three stage protocol:

1. Convolve the image with the first 15 filters of the DCT base at N_s scales, yielding $N_s \times 15$ coefficient 'activity maps'. We use $N_s = 5$, spanning patch sizes between 5 and 30 pixels.
2. Compute $P \times N_s$ appearance maps by applying the parts appearance models to the vector of DCT coefficients at every image location. The coordinate values (x, y) in map (k, j) contain the log probability of part k with scale j in location (x, y).
3. Apply the relational model to the set of appearance maps, yielding a single log probability map for the 'hidden center' node. To this end, the N_s appearance maps of each part are merged into a single map by choosing at each coordinate the most likely part scale. We then compute P part message maps, corresponding to the messages $h^k(I, C)$ defined in (24), by applying the distance transform (Feltzenswalb and Huttenlocher 2005) to the merged appearance maps. Finally the 'hidden center' map is formed as a weighted sum of parts message maps.

5.5.1 *Caltech Data Sets*

3 data sets used in the recognition experiments are supplied with bounding box information: Airplanes, Motorcycles and Faces. However, only in the latter objects have significant location variance, specifically in the X-axis. On these data sets we measure localization accuracy using the distance between the true and the detected object center, normalized by the bounding box size. These measurements are collected separately for the x and y axis (normalized by the x and y sizes of the bounding box respectively), and for the two dimensional Euclidean distance (normalized by the bounding box diagonal). Median scores for the three data sets are

Table 5 Median Localization scores for Caltech data sets. The scores are distances between true and detected object centers, normalized by the bounding box length in the relevant dimension. In the parenthesis the same scores are given for the null hypothesis, placing the object at the image center. We can see that the inaccuracies in median detection are not larger than 10% of the bounding box size for all categories

Data Name	x axis	y axis	$\sqrt{(x^2 + y^2)}$
Motorbikes	0.0251 (0)	0.0577 (0)	0.0463 (0)
Airplanes	0.0516 (0.0281)	0.1048 (0.0938)	0.0699 (0.0495)
Faces	0.0743 (0.2018)	0.0325 (0.0313)	0.0583 (0.1313)

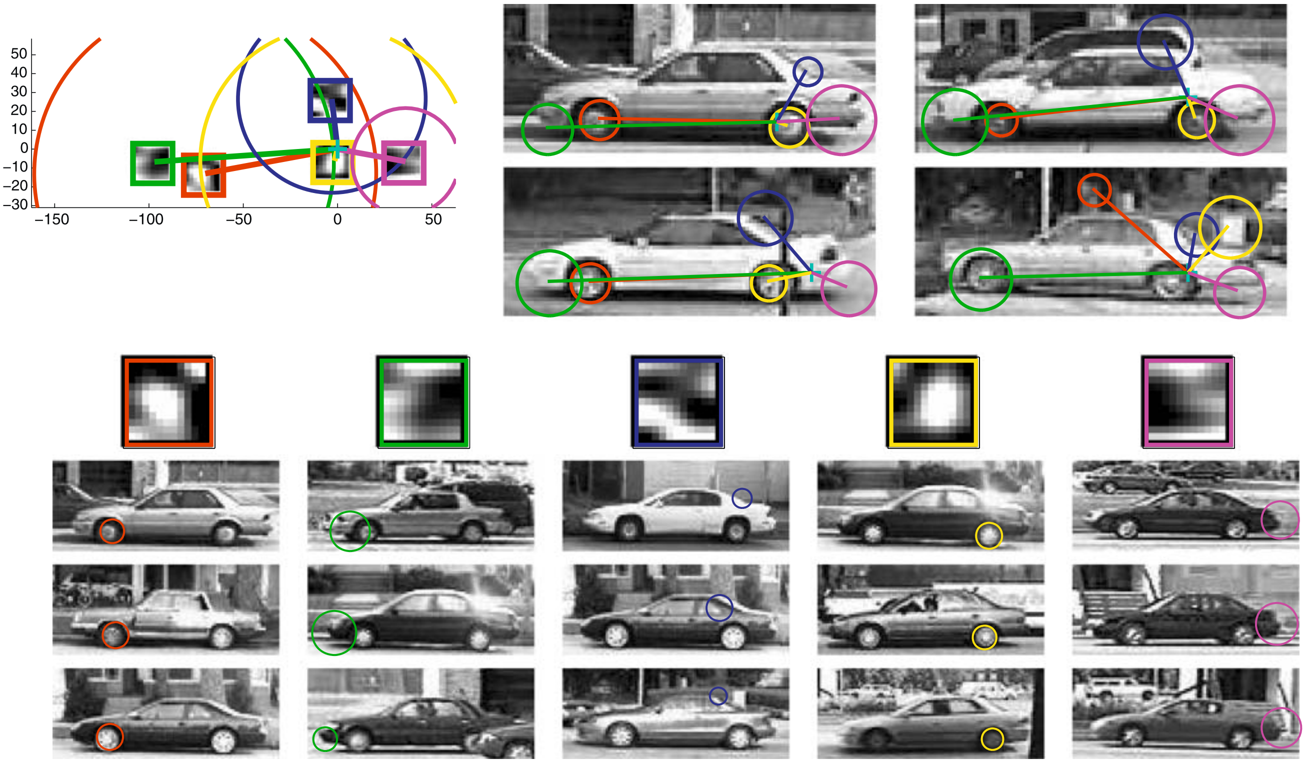


Fig. 9 (Color online) 5 parts from the car side model used in the localization task. The parts shown correspond to the two wheels, front and rear ends, and the top-rear corner. The complete model includes 38 parts, most of them with clear semantics. While the model is not symmetric w.r.t. to the x axis, it is not far from being so. It hence happens that a car is successfully detected, but its direction is not properly identified

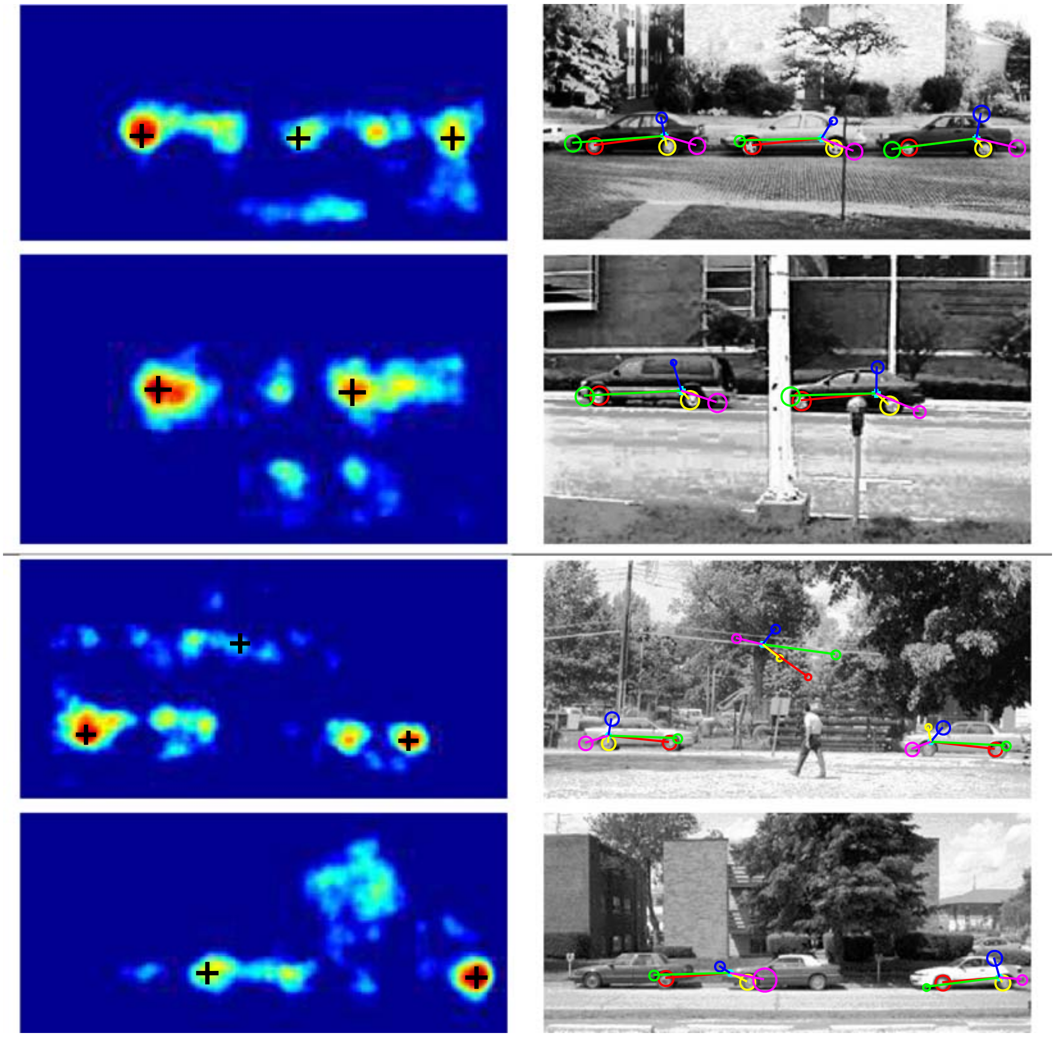
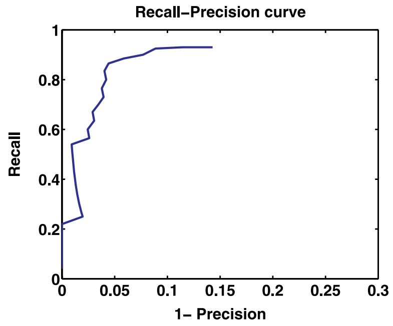


Fig. 10 (Color online) Hidden center probability maps and car detections. In each image pair, the *left image* shows the probability map for the location of reference point C. The *right image* shows the 5 parts from Fig. 9 superimposed on the detected cars. *Top* Successful detections. Notice that the middle of a gap between two cars tends to emerge as a probable car candidate, as it gains support from both cars. *Bottom* Problematic detections. The third example includes a spurious detection and a car detected using the model of the 'wrong' direction. The *bottom* example includes a spurious 'middle car' between two real cars. Values in the probability maps were thresholded and linearly transformed for visualization

reported in Table 5, as well as median scores for the null hypothesis, stating that the object is located at the image center. Clearly the final localization is quite accurate in these experiments. Note, however, that these results are only significant for the faces data set—the only one (in the CalTech data) with any meaningful variation in object location, as can be seen in the scores of the null hypothesis. Similar effects were noticed in Loeff et al. (2005).

5.5.2 UIUC Cars Side Data

The data includes cars facing both directions (i.e. left-to-right and right-to-left). We therefore flip the training images prior to training, such that all cars face the same direction. At the test phase we run the exhaustive search for the learnt model and its mirror image. We detect local maxima in the hidden center map, sort them according to likelihood, and prune neighboring maxima in a way similar to the neighborhood suppression technique suggested in Agarwal et al. (2004). Figure 10 presents some probability maps and detected cars, illustrating typical successful and problematic detections. Each detection is labeled as hit or miss using the criterion used in Agarwal and Roth (2002) (which is slightly different from the one used in Agarwal et al. (2004)), to allow for a fair comparison with other methods. Figure 11 presents a precision-recall curve and a comparison of the achieved localization performance to several recently suggested methods. Our results are comparable to those obtained by the best methods, and are inferior only to classi-



a)

Method	Equal error rate
Roth et al. 2002	0.21
Fergus et al. 2003	0.115
Leibe et. al. 2005	0.114
Fergus et al. 2005	0.078
Our method	0.076

b)

Fig. 11 **a** Recall-Precision curve for cars side detection, using the model shown in Fig. 9. **b** Error rates (recall = 1 − precision) obtained on the cars side data by several recent methods. Our performance is comparable to the best methods of weakly supervised learning

fiers learnt using 'strong' supervision, in the form of images with parts segmentations (e.g. the stronger classifiers in Leibe et al. 2004). In such a method part identities are not learnt but chosen manually, and so the learning task is simpler.

6 Discussion

We have presented a method for object class recognition and localization, based on discriminative optimization of a relational generative model. The method combines the natural treatment of spatial part relations, typical to generative classifiers, with the efficiency and pruning ability of discriminative methods. Efficient, scalable learning is achieved by extending boosting techniques to a simple relational model with conditionally dependent parts. In a recognition task, our method compares favorably with several purely generative or purely discriminative systems recently proposed. In a localization task its performance is comparable to the best available methods.

While our recognition results are good, (Opelt et al. 2004a; Serre et al. 2005) report better results obtained using discriminative methods which ignore geometric relations and focus instead on feature representation. Specifically, in Opelt et al. (2004a) segmentation based features are used, while in Serre et al. (2005) features are based on flexible exhaustive search of 'code book' patches. The recognition performance of these approaches relies on better feature extraction and representation, compared with our simple combination of interest point detection and DCT-based representation. We regard the advances offered by these methods as orthogonal to our main contribution, i.e. the efficient incorporation of geometrical relations. The advantages can be combined by combining better part appearance models and better feature extraction techniques with the relational boosting technique suggested here. We intend to continue our research along these lines.

The complexity of our suggested learning technique is linear in the number of parts and features per image, but it may still be quite expensive. Specifically, the inference complexity of the hidden center C is $O(N_c N_f P)$ where N_c is the number of considered center locations, and this inference is carried for each image many times during learning. This limits us to a relatively crude grid of possible center locations in the learning process, and hence limits the accuracy of the location model learnt. A possible remedy is to consider less exhaustive methods for inferring the optimal hidden center, based on part voting or mean shift mode estimation, as done in (Leibe et al. 2004). Such 'heuristic' inference solutions may offer enhanced scalability and a more biologically plausible recognition mechanism.

Finally, leaving technical details aside, we regard this work as a contribution to an important debate in learning from unprocessed images, regarding the choice of generative vs. discriminative modeling. We demonstrated that combining generative relational modeling with discriminative optimization can be fruitful and lead to more scalable learning. However, this combination is not free of problems. Our technical problems with covariance matrix learning and the tendency of our technique to produce 'exaggerated' models are two examples. The method proposed here is a step towards the required balance between the descriptive power of generative models and the task relatedness enforced by discriminative optimization.

Acknowledgements We are indebted to Tomer Hertz, who has been involved in the early stages of this research, for his support and his help in the empirical validation of the method.

Appendix 1: Feature Repetition and ML Optimization in a Star Model

Allowing feature repetitions, we derived the likelihood approximation (10) for our star model. For a set of object images $\{I_j\}_{j=1}^n$. This approximation entails the following total data likelihood

$$\begin{aligned}\sum_{j=1}^n \log P(I_j|\Theta) &= n \log K_0 + \sum_{j=1}^n \log\left[\max_C \prod_{K=1}^P \max_{x\in F(I_j)} P(x|C,\theta^k)\right] \\ &= n \log K_0 + \sum_{j=1}^n \max_C \sum_{K=1}^P \max_{x\in F(I_j)} \log P(x|C,\theta^k). \qquad (33)\end{aligned}$$

The maximum likelihood parameters $\Theta = (\theta_1, \ldots, \theta_P)$ are chosen to maximize this likelihood. In this maximization, we can ignore the constant term $n \log K_0$. To simplify further notation let us denote parts' conditional log likelihood terms by $g_j(C, \theta_k) = \max_{x\in F(I_j)} P(x|C,\theta_k)$. Also denote the vector of the hidden center variables in all images by $\vec{C} = (C_1, \ldots, C_n)$.

$$\begin{aligned}\max_\Theta\left[\sum_{j=1}^n \max_C \sum_{K=1}^P g_j(C,\theta^k)\right] &= \max_{(C_1,\ldots,C_n)}\left[\max_\Theta \sum_{j=1}^n \sum_{K=1}^P g_j(C_j,\theta^k)\right] \\ &= \max_{\vec{C}} \sum_{K=1}^P \max_{\theta^k}\left[\sum_{j=1}^n g_j(C_j,\theta^k)\right]. \qquad (34)\end{aligned}$$

For any fixed centers vector $\vec{C}$, and any $1 \le k \le P$, the optimal θ^k is determined as $\theta^k = \arg\max_\theta G(\theta, \vec{C})$ where $G(\theta, \vec{C}) = \sum_{j=1}^n g_j(C_j, \theta)$. Hence, for any $\vec{C}$, the optimal part parameters θ^k are identical, as maxima of the same function. Clearly the maximum over $\vec{C}$ also posses this property.

The proof can be repeated in a similar way for the star model presented in Fergus et al. (2005), in which the center node is an additional 'landmark' part, as long as the sum over all model interpretations in an image is replaced by the single maximal likelihood interpretation.

Appendix 2: Part Weights Introduction

Here we establish the functional equivalence between classifiers with and without part weights for weak learners of the form (14). We use the identity

$$\log G(x|\mu, \Sigma) - \nu = \alpha[\log G(x|\mu', \Sigma') - \nu'] \qquad (35)$$

where

$$\mu' = \mu, \quad \Sigma' = \alpha\Sigma,$$
$$\nu' = \frac{1}{\alpha}\left[\nu + \frac{d(1-\alpha)}{2}\log 2\pi + \frac{1-\alpha}{2}\log|\Sigma| - \frac{\alpha d}{2}\log\alpha\right]$$

to introduce part weights into the classifier. This identity is true for all $\alpha > 0$. We apply this identity to each part k in the classifier (14), with $\alpha^k = |\Sigma_a^k|^{-1/d}$, to obtain

$$\begin{aligned}f(I) &= \sum_{k=1}^P \max_{x\in F(I)} \log G(x|\mu_a^k, \Sigma_a^k) - \nu^k \\ &= \sum_{k=1}^P \alpha^k[\max_{x\in F(I)} \log G(x|\mu_a^k, \Sigma_a^{k'}) - \nu^{k'}] \qquad (36)\end{aligned}$$

where $\Sigma_a^{k'} = \alpha^k \Sigma_a^k$ is has a fixed determinant of 1 for all parts. The weights α^k therefore (inversely) reflect covariance scale.

Appendix 3: Proof of Lemma 1

We differentiate the loss w.r.t. ν

$$\begin{aligned}0 &= \frac{d}{d\nu}\sum_{i=1}^N \exp(-y_i[f(I_i) - \nu]) \\ &= -\sum_{\{i:y_i=1\}} \exp(-f(I_i) + \nu) \\ &\quad + \sum_{\{i:y_i=-1\}} \exp(f(I_i) - \nu). \qquad (37)\end{aligned}$$

For $\tilde{f} = f - \nu$, (37) gives property (21). Solving for ν gives

$$\begin{aligned}\exp(\nu) \sum_{\{i:y_i=1\}} \exp(-f(I_i)) &= \exp(-\nu) \sum_{\{i:y_i=-1\}} \exp(f(I_i)) \qquad (38)\end{aligned}$$

from which (20) follows. Finally, we can compute the loss using the optimal ν^*

$$\begin{aligned}\sum_{i=1}^N \exp(-y_i[f(I_i) - \nu^*]) &= \left[\frac{\sum_{\{i:y_i=-1\}} \exp(f(I_i))}{\sum_{\{i:y_i=1\}} \exp(-f(I_i))}\right]^{\frac{1}{2}} \sum_{\{i:y_i=1\}} \exp(-f(I_i)) \\ &\quad + \left[\frac{\sum_{\{i:y_i=-1\}} \exp(f(I_i))}{\sum_{\{i:y_i=1\}} \exp(-f(I_i))}\right]^{-\frac{1}{2}} \sum_{\{i:y_i=-1\}} \exp(f(I_i))\end{aligned}$$

from which (22) follows.

References

Agarwal, S., Awan, A., & Roth, D. (2004). Learning to detect objects in images via a sparse, part based representation. *Pattern Analysis and Machine Intelligence*, *20*(11), 1475–1490.

Agarwal, S., & Roth, D. (2002). Learning a sparse representation for object detection. In *ECCV* (pp. 113–130).

Bar-Hillel, A., Hertz, T., & Weinshall, D. (2005a). Efficient learning of relational object class models. In *ICCV*.

Bar-Hillel, A., Hertz, T., & Weinshall, D. (2005b). Object class recognition by boosting a part based model. In *CVPR*. Los Alamitos: IEEE Computer Society

Borenstein, E., Sharon, E., & Ullman, S. (2004). Combining top-down and bottom-up segmentation. In *IEEE workshop on perceptual organization in computer vision (CVPR)*.

Chan, A. B., Vasconcelos, N., & Moreno, P. J. (2004). A family of probabilistic kernels based on information divergence.

Csurka, G., Bray, C., Dance, C., & Fan, L. (2004). Visual categorization with bags of keypoints. In *ECCV*.

Dorkó, G., & Schmid, C. (2005, submitted). Object class recognition using discriminative local features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Everingham, M. R., Zisserman, A., Williams, C. K. I., & Van Gool, L. et al. (2006). The 2005 pascal visual object classes challenge. In J. Quinonero-Candela, I. Dagan, B. Magnini, & F. d'Alche-Buc (Eds.), *LNAI: Vol. 3944. Machine learning challenges. Evaluating predictive uncertainty, visual object classification, and recognising textual entailment* (pp. 117–176).

Fei-Fei, L., Fergus, R., & Perona, P. (2003). A bayesian approach to unsupervised one shot learning of object catgories. In *ICCV*.

Feltzenswalb, P., & Huttenlocher, D. (2005). Pictorial structures for object recognition. *International Journal of Computer Vision*, *61*, 55–79.

Fergus, R., Perona, P., & Zisserman, A. (2003). Object class recognition by unsupervised scale invariant learning. In *CVPR*. Los Alamitos: IEEE Computer Society

Fergus, R., Perona, P., & Zisserman, A. (2005). A sparse object category model for efficient learning and exhaustive recognition. In *CVPR*.

Freund, Y., & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *ICML* (pp. 148–156).

Friedman, J. H., Hastie, T., & Tibshirani, R. (2000). Additive logistic regression: a statistical view ofboosting. *Annals of Statistics*, *28*, 337–407.

Fritz, M., Leibe, B., Caputo, B., & Schiele, B. (2005). Integrating representative and discriminant models for object category detection. In *ICCV*.

Gao, D., & Vasconcelos, N. (2004). Discriminant saliency for visual recognition from cluttered scenes. In *NIPS*.

Holub, A. D., & Perona, P. (2005). A discriminative framework for modeling object classes. In *CVPR*.

Holub, A. D., Welling, M., & Perona, P. (2005). Combining generative models and fisher kernels for object class recognition. In *ICCV*.

Kadir, T., & Brady, M. (2001). Scale, saliency and image description. *International Journal of Computer Vision*, *45*(2), 83–105.

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *ECCV workshop on statistical learning in computer vision*.

Li, Y., Shapiro, L., & Bilmes, J. (2005). A generative /discriminative learning algorithm for image classification. In *ICCV* (Vol. 2, pp. 1605–1612).

Loeff, N., Arora, H., Sorokin, A., & Forsyth, D. (2005). Efficient unsupervised learning for localization and detection in object categories. In *NIPS*.

Lowe, D. (2001). Local feature view clustering for 3D object recognition. In *CVPR*, (pp. 682–688).

Mason, L., Baxter, J., Bartlett, P., & Frean, M. (2000). Boosting algorithms as gradient descent in function space. In *NIPS* (pp. 512–518).

Murphy, K. P., Torralba, A., & Freeman, W. T. (2003). Using the forest to see the trees: a graphical model relating features, objects and scenes. In *NIPS*.

NG, A. Y., & Jordan, M. I. (2001). On discriminative vs. generative classifiers: A comparison of logistic regression and naive bayes. In *NIPS*.

Opelt, A., Fussenegger, M., Pinz, A., & Auer, P. (2004a). *Object recognition with boosting* (Technical report tr-emt-2004-01). Submitted to PAMI.

Opelt, A., Fussenegger, M., Pinz, A., & Auer, P. (2004b). Weak hypotheses and boosting for generic object detection and recognition. In *ECCV*.

Schapire, R. E., & Singer, Y. (1999). Improved boosting using confidence-rated predictions. *Machine Learning*, *37*(3), 297–336.

Serre, T., Wolf, L., & Poggio, T. (2005). A new biologically motivated framework for robust object recognition. In *CVPR*.

Thureson, J., & Carlsson, S. (2004). Appearance based qualitative image description for object class recognition. In *ECCV* (pp. 518–529).

Torralba, A., Murphy, K., & Freeman, W. T. (2004). Contextual models for object detection using boosted random fields. In *NIPS*.

Ullman, S., Vidal-Naquet, M., & Sali, E. (2002). Visual features of intermediate complexity and their use in classification. *Nature Neuroscience*, *5*, 682–687.

Ulusoy, I., & Bishop, C. M. (2005). Generative versus discriminative methods for object recognition. In *CVPR* (Vol. 2, pp. 258–265).

Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.

Vidal-Naquet, M., & Ullman, S. (2003). Object recognition with informative features and linear classification. In *ICCV*.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *CVPR*.