Object Class Recognition by Boosting a Part-Based Model

Aharon Bar-Hillel, Tomer Hertz and Daphna Weinshall
{email: aharonbh,tomboy,daphna@cs.huji.ac.il}
School of Computer Science and Engineering and the Center for Neural Computation
The Hebrew University of Jerusalem, Jerusalem Israel 91904

Abstract

We propose a new technique for object class recognition, which learns a generative appearance model in a discriminative manner. The technique is based on the intermediate representation of an image as a set of patches, which are extracted using an interest point detector. The learning problem becomes an instance of supervised learning from sets of unordered features. In order to solve this problem, we designed a classifier based on a simple, part based, generative object model. Only the appearance of each part is modeled. When learning the model parameters, we use a discriminative boosting algorithm which minimizes the loss of the training error directly. The models thus learnt have clear probabilistic semantics, and also maintain good classification performance. The performance of the algorithm has been tested using publicly available benchmark data, and shown to be comparable to other state of the art algorithms for this task; our main advantage in these comparisons is speed (order of magnitudes faster) and scalability.

1. Introduction

Object class recognition is a fundamental problem in vision, and has recently drawn considerable interest [2, 3, 11]. The problem seems harder than the recognition of specific objects, mainly because of the large inner class variance that exists in most visual object categories. As in many other computer vision tasks, the key to any progress may lie in our ability to find and manipulate a "good" intermediate representation. This representations should comply with the following basic requirements: (1) It should be efficiently and reliably extracted from images. (2) It needs to be rich enough to capture the invariant aspects of the classes at hand. (3) It should be amenable to learning techniques which expose these invariant aspects.

Different flavors of part based representations, where an object is modeled using a set of representative parts (and possibly some relations among them), have been adopted in the recent work mentioned above. Such a distributed model can naturally cope with the large inner variance among class objects, and may also cope with occlusion. More specifically, parts were often represented by image patches. For example, in [2, 3] the basic image representation is an unordered set of image patches and their locations. Such a representation emerges when an interest point detector is applied to the image, and then local patch descriptors are extracted from the locations highlighted by the detector. This is a rather efficient procedure, which leads to a relatively compact representation of the image. The success of [1, 2, 3] implies that such representations may be rich enough to allow for "good" object class classification. In this work we therefore follow this approach, and represent each image using a small number of image patches that are automatically extracted using an interest point detector.

Given such a representation, object recognition is a problem of classification using sets of unordered features as input. Learning a classifier in this setting is equivalent to learning a function from an unordered set of features to a set of binary labels. Unfortunately, this is a rather non standard learning problem, which is rarely considered in the learning literature. On the other hand, learning from ordered feature vectors is a well understood problem, for which powerful algorithms have been developed. One of the main contributions of this paper is a modification of one such algorithm that can learn from unordered sets of features.

Specifically, we propose a boosting learning scheme which learns a discriminative classifier from object images and background images. The classifier is based on a simple part based probabilistic model. Currently the model contains only appearance models of the various parts, and does not enforce any demands on their absolute or relative locations. Recognition is achieved by comparing the likelihood score of an image, computed using the part based model, with a fixed threshold. The boosting process learns the parts and updates the threshold in a serial manner, using a greedy procedure.

The algorithm has two key characteristics. The first is the simplicity of the generative object model, which is our key to computational efficiency. The second important point is that the generative model's parameters are learnt using a

discriminative technique. This means that the probabilistic model's parameters are not chosen to maximize the likelihood of object images. Instead they are chosen to minimize a loss function over the classification training error. This, we believe, is a key to good classification performance. Thus the main advantages of the approach are speed and efficiency - learning the model is relatively fast, and recognition is almost instantaneous given a set of patch features extracted from an image. The learnt classifiers are inherently translation invariant and also have high degree of scale invariance. The recognition performance is comparable to other state of the art algorithms, and the parts learnt have clear semantics in many cases.

1.1. Learning a part based model from an unordered feature set

While other approaches to learning from unordered sets exist in the literature (e.g. [2]), our work is closely related to the approach suggested by [3]. In this work object class recognition is handled using a generative model, which models the part's appearance, relative location and scale. The model parameters are learnt in an unsupervised manner (i.e. from object images alone), with the aim of increasing the likelihood of object images. The problem of learning from unordered sets is tackled by considering all the possible ordered vectors of parts that can be formed using the feature set. The problem with this approach is its computational efficiency. The location and scale models form complex probabilistic dependencies between all of the model's parts (in the language of graphical models, we say that the entire part model graph becomes one large clique). In these circumstances, the assessment of the likelihood for all the possible vectors cannot be simplified, and it is exponential in the number of model's parts. Hence only a small number of parts can be considered.

Like [3], our approach is based on a generative model, and therefore we must also compute the score of all possible feature vectors. However, in the tradition of graphical models, we circumvent the exponential explosion by reducing the dependencies between the different object parts. In this paper we take this idea to the extreme and assume that the model's parts are independent of one another (see Section 2.2). This reduces the computational effort to linear in the number of parts. If we wish to be invariant to translation and scale transformations, we cannot include any scale or location information in the part's model. Therefore, in this paper we only use an appearance model. As we will show in our experimental results, this highly simplified model can in many cases achieve excellent recognition performance, which is comparable to the results presented in [3].

Our simple part appearance model bears some resemblance to the approach described in [11, 9], which advocates the use of real image patches of intermediate size for classi-

fication and for other computer vision tasks. Another work closely related to our work from the discriminative point of view is the work of [1]. In this work a part-based model is trained to discriminate object images from background images, using the Adaboost algorithm. However, parts are not probabilistically modeled, and are instead represented using SIFT descriptors. In Section 3 we compare our performance to the results presented in [1].

1.2. Discriminative learning of a probabilistic model

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

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

where θ is a constant threshold.

Discriminative classifiers 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 $sign(f(x))$, 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 $p(x|y)$ [10]. 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 [7].

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. It is plausible to expect that a carefully designed classifier, whose functional characteristics are determined by generative modeling, will give better performance than a classifier from an arbitrary parametric family. In accordance with the arguments above, in this paper we follow a hybrid path: We choose the functional form of the classifier based on a simple generative model of the data, and then learn the model's parameters in a discriminative setting.

Thus, while sharing with [3] the idea of learning a part-based probabilistic model from an unordered feature set, our choice of discriminative learning is related to the boosting technique used in a number of recent detection systems

[12, 6, 1]. However, this latter resemblance is more superficial; unlike our proposed approach in which an object is modeled as a collection of parts with flexible locations, these papers design detection systems in which an object is modeled using rigid templates ('Patch based' in the terminology of [6]). Each weak hypothesis depends on the response of specific localized filters in the template. The resulting classifier is 'opaque' is the sense that it does not have a probabilistic interpretation.

2. The recognition model

In this section we describe our proposed method. In Section 2.1 we briefly describe how an image is transformed into a set of local patch descriptors. Section 2.2 presents our simple generative object model and the functional form of our suggested classifier. In Section 2.3 we show how such a model can be learnt in a discriminative setting, using an adaptation of Adaboost with confidence intervals [8]. In Section 2.4 we describe the weak learners we use.

2.1. Feature extraction and representation

Our initial feature extraction and representation scheme follows the scheme presented by [3]. First, images are rescaled to have a uniform horizontal axis length (200 pixels). Features are detected using the Kadir and Brady detector [4]. The detector searches for image regions with high entropy. It finds a set of circular region candidates of various scales, where each region corresponds to some local maximum of an entropy based score in scale space. The initial candidate set includes thousands of candidates for a typical image. Following [3], we multiply the entropy based score by the candidate's scale, thus creating a preference for large image patches, which are usually more informative.

A set of N high scoring features with limited overlap is then chosen using an iterative greedy procedure. While [3] uses a small set of features ($N = 30$), the computational efficiency of our method allows us to use a larger set ($N = 60, 100, 200$), with higher overlap between features. This overlap increases the likelihood that an object part in an image will be captured by a patch with the correct scale and alignment.

The selected regions are cropped from the image and scaled down to 11×11 pixel patches. They are then normalized to have zero mean and variance of 1. Finally they are transformed into 15 dimensional vectors using PCA, computed using all of the background features (patches). The transformation of a single image into a representative feature set, including feature computation and dimensionality reduction, takes 2-3 seconds using matlab on a 1.3Ghz machine.

2.2. An appearance model based classifier

For recognition we use a simple classifier based on a generative object model. Our guiding principle in building the classifier is to make it as simple as possible, using (hopefully reasonable) assumptions about the problem domain. Let us denote the feature set representing image i as $F(I_i)$. We propose a part-based model, where each part is implemented in a specific image I_i by one of the patch features in $F(I_i)$. The appearance of each part is modeled using a Gaussian distribution. In our current model we *do not* assume any spatial relations between the different parts (i.e we assume that they are independent).

Specifically, a model of part k is a Gaussian $G(\cdot | M_k)$ where $M_k = (\mu_k, \Sigma_k)$ denote the mean and covariance matrix of the part's appearance. Assuming independence of appearance between parts, the log likelihood for a vector of part candidates $(x_1, .., x_P)$ is

$$\sum_{k=1}^{P} \log G(x_k | M_k) \qquad (2)$$

Since our input is not an ordered vector of parts, we should (in principle) sum over all the possible ordered vectors that can be generated from the feature set $F(I_i)$. We choose to approximate this average using the likelihood obtained by the best vector. While the framework can accommodate average computation without additional computational cost, we prefer to work with the best vector since it uniquely identifies a part and its localization in each image. and hence improves the semantics of a part.

$$
\begin{aligned}
p(I_i | M) &\approx \max_{(x_1,..,x_P) \in F(I_i)^P} \sum_{k=1}^{P} \log G(x_k | M_k) \\
&= \sum_{k=1}^{P} \max_{x \in F(I_i)} \log G(x | M_k)
\end{aligned}
$$

Notice that due to the independence assumption, finding the maximal vector (in a set of $|F(I_i)|^P$ possible vectors) simplifies to P independent problems of finding the maximal component (in $O(P \cdot |F(I_i)|)$ time).

Modeling the background hypothesis $p(x | y = -1)$ is tricky, as there is no reason to assume that a simple parametric model (like a Gaussian) will adequately describe such a fragmented set. We therefore approximate the background hypothesis using a constant. Under these assumptions, the functional form of the LRT statistic from (1) becomes

$$f(I) = \sum_{k=1}^{P} \max_{x \in F(I)} \log G(x | M_k) - \Theta \qquad (3)$$

and the models parameters are $\{\mu_k, \Sigma_k\}_{k=1}^{P}$ and Θ.

2.3. Discriminative learning using boosting

We now present a discriminative learning algorithm that learns a model of the form (3) via the following equivalent form

$$f(I) = \sum_{k=1}^{P} \alpha_k \max_{x \in F(I)} \log G(x|\mu_k, \Sigma_k) - \theta_k \qquad (4)$$

The two parametric families are equivalent since any function of the form (4) can be written as a function of the form (3) by substituting

$$\mu_k \leftarrow \mu_k \qquad \Sigma_k \leftarrow \alpha_k \Sigma_k \qquad (5)$$

$$\Theta \leftarrow \sum_{k=1}^{P} [\frac{(\alpha_k - 1)}{2} log 2\pi |\Sigma_k| + \frac{d}{2} log \alpha_k + \theta_k]$$

Given a set of labeled images $\{I_i, y_i\}_{i=1}^{N}$, the algorithm tries to minimize the exponential loss of the margin, summed over all training points:

$$C(f) = \sum_{i=1}^{N} \exp(-y_i f(I_i)) \qquad (6)$$

(6) is the loss minimized by the Adaboost algorithm [8]. In [5], Adaboost with confidence intervals [8] is shown to be a greedy gradient descent of this loss functional in L^2 function space. Our algorithm is derived using the same techniques. The algorithm uses a weak learner to find the parameters μ_k and σ_k in each round. It then determines the parameters α_k and θ_k using a 1-dimensional line search. The algorithm's pseudo-code is given in Alg. 1.

In the following, we explain the algorithm in three stages: First (in Section 2.3.1) we describe the sample re-weighting policy and the weak learner's task as one of performing gradient descent in functional space. Note that while the derivation of the sample weights is done in accordance with [5], we consider some variants of the weak learner's task. In Section 2.3.2 we derive the update rules for the parameters θ_k and α_k (steps 3,4 and the algorithm's initialization), and note an important property of the sampling weights. Finally (in Section 2.3.3), we derive the score to be maximized by the weak learner in step 1.

2.3.1 Greedy gradient descent in function space

In [5] the choice of weak hypotheses in boosting is derived from considerations of gradient descent in a functional space with the inner product $< f, g > = \sum_{i=1}^{N} f(I_i)g(I_i)$. At boosting round k, the algorithm greedily tries to add a component to the current function $f_{k-1}(I)$. Thus it considers extensions of the form $f_k(I) = f_{k-1}(I) + \alpha_k \cdot h_k(I)$, where $h_k(I) = \max_{x \in F(I)} \log G(x|\mu_k, \Sigma_k) - \theta_k$. We look for a

---

Algorithm 1 Discriminative learning algorithm

Given $\{(I_i, y_i)\}_{i=1}^{N}$ $y_i \in \{-1, 1\}$, initialize:

$\theta_0 = \frac{1}{2} log \frac{\#\{y_i = -1\}}{\#\{y_i = 1\}}$
$w_i^1 = exp(y_i \cdot \theta_0)$ $i = 1, .., N$
$w_i^1 = w_i^1 / ||w^1||_2$

For $k = 1, .., P$

1. Use a weak learner to find a part hypothesis of the form $h_k(I) = \max_{x \in F(I_i)} G(x|\mu, \Sigma)$ which maximizes

$$\frac{\sum_{i=1}^{N} w_i^k y_i h(I_i)}{||h(I) - \frac{1}{N} \sum_{i=1}^{N} h(I_i)||_2}$$

2. If no hypothesis is found where the expression above is positive, terminate the loop and set $P = k - 1$.

3. Use line search to find α_k which maximizes

$$\arg \max_{\alpha} \frac{\sum_{i:y_i=1} w_i^k exp(-\alpha h_k(I_i))}{\sum_{i:y_i=-1} w_i^k exp(\alpha h_k(I_i))}$$

4. Set $\theta_k = \frac{1}{2} log \frac{\sum_{i:y_i=-1} w_i^k exp(\alpha h_k(I_i))}{\sum_{i:y_i=1} w_i^k exp(-\alpha h_k(I_i))}$

5. Update weights

$$w_i^{k+1} = w_i^k exp(-y_i(\alpha_k h_k(I_i) - \theta_k))$$
$$w_i^{k+1} = w_i^{k+1} / ||w^{k+1}||_2$$

Output the final hypothesis $f(I) = \sum_{k=1}^{P} \alpha_k h_k(I) - \Theta$ where $\Theta = \sum_{k=0}^{P} \theta_k$.

'direction' $g(I)$ in function space for which $C(f(I) + \epsilon g(I))$ most rapidly decreases. This 'direction' is found by differentiating $C(f)$ from (6) w.r.t. f.

$$\nabla C(f)(x_i) = \frac{d}{df(x_i)} \sum_{j=1}^{N} \exp(-y_j f(x_j)) \qquad (7)$$
$$= -y_i \exp(-y_i f(x_i))$$

Note that in [5] it is recommended that the weak learner will minimize the inner product of the new $h_k(I)$ and the gradient. Denoting the weights $w_i^k = \exp(-y_i f_k(x_i))$, the weak learner's task is to find

$$\arg \min_{h(I)} < \nabla c(f), h(I) > = \arg \max_{h(I)} \sum_{i=1}^{N} y_i w_i^k h(I_i) \qquad (8)$$

While this is a reasonable choice, it is suboptimal in the following sense. Recall that the boosting process independently chooses the optimal hypothesis weight α and adds $\alpha h(I)$ to the complex hypothesis $f(I)$. Thus $\lambda h(I)$ and $h(I)$ should receive the same score. It seems therefore preferable to normalize $h(I)$ in the inner product, which is equivalent to considering the cosine of the angle between the gradient and $h(I)$:

$$\arg\min_{h(I)} \frac{<\nabla c(f), h(I)>}{||\nabla c(f)|| \cdot ||h(I)||} = \arg\max_{h(I)} \frac{\sum_{i=1}^{N} y_i w_i^k h(I_i)}{\sqrt{\sum_{i=1}^{N} h^2(I_i)}} \quad (9)$$

In our algorithm we therefore use a variant on the traditional Adaboost sampling weights, slightly altering the score that has to be maximized by the weak learner. In Section 2.3.3 we derive the score based on the optimization problem (9) and our specific model.

2.3.2 Optimization of θ and α

Our update of θ and α, as well as the specific score maximized by the weak learner, are based on the following lemma:

Lemma 1. *Assume we are given a function $f : I \rightarrow R$, and we wish to minimize the loss (6) of the function $\tilde{f} = f - \theta$ where θ is a constant. Assume also that there are both $+1$ and -1 labels in the dataset*

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

$$\theta^* = \frac{1}{2} \log \left[\frac{\sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i))}{\sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i))} \right] \quad (10)$$

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

$$\sum_{\{i:y_i=1\}}^{N} \exp(-\tilde{f}^*(I_i)) = \sum_{\{i:y_i=-1\}}^{N} \exp(\tilde{f}^*(I_i)) \quad (11)$$

3. The loss of $\tilde{f}^*$ is

$$2 \left[\sum_{\{i:y_i=1\}}^{N} \exp(-\tilde{f}^*(I_i)) \cdot \sum_{\{i:y_i=-1\}}^{N} \exp(\tilde{f}^*(I_i)) \right]^{\frac{1}{2}} \quad (12)$$

Proof. We first differentiate the loss w.r.t. θ

$$0 = \frac{d}{d\theta} \sum_{i=1}^{N} \exp(-y_i[f(I_i) - \theta]) \quad (13)$$

$$= -\sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i) + \theta) + \sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i) - \theta)$$

For $\tilde{f} = f - \theta$, (13) gives property (11). Solving for θ gives

$$\exp(\theta) \sum_{\{i:y_i=1\}}^{N} \exp(-f(I_i)) = \quad (14)$$

$$\exp(-\theta) \sum_{\{i:y_i=-1\}}^{N} \exp(f(I_i))$$

from which (10) follows. Finally, we can compute the loss using the optimal θ^*

$$\sum_{i=1}^{N} \exp(-y_i[f(I_i) - \theta^*]) =$$

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

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

from which (12) follows. □

Corollary 1. *1. It follows from (10)) that we can choose θ_k optimally once $h_k(I)$, α_k have been chosen. This is done in step 4 of the algorithm using $f(I) = \sum_{j=1}^{k-1} [\alpha_j h_j(I) - \theta_j] + \alpha_k h_k(I)$, and in the intialization using $f(I) = 0$.*

2. (12) allows us to find the optimal α, as done in step 3 of the algorithm. For each value of α the optimal loss value obtained by adding $\alpha h_k(I) - \theta_k^*(\alpha)$ is computed using (12), and line search is conducted to find the optimal α.

3. Since the sampling weights at each round satisfy $w_i \propto exp(-y_i f(x_i))$, (11) implies that after updating the threshold, the sum of weights of positive data examples and the sum of weights of negative data examples is equal. This simplifies the task which is posed to the weak learner, as discussed in Section 2.3.3.

2.3.3 The weak leaner's task

The task of the weak learner in step 1 of the algorithm, is to find a hypothesis of the form

$$\max_{x \in F(I)} \log G(x|\mu, \Sigma) - \theta \quad (15)$$

which maximizes the criterion (9) or its simpler version in (8). Note that although the threshold θ_k is not determined

by the weak learner, we still consider a weak learner with a threshold parameter. The parameter θ as chosen by the weak learner is not used to determine θ_k, since choosing θ_k using (10) is provably optimal (in terms of loss minimization), and hence preferable to the one found by the weak learner using 'local' gradient considerations. However, when choosing $h_k(I)$ the weak learner can choose better hypotheses when given the additional flexibility supplied by the threshold.

Denote by $V(\mu, \Sigma)$ the N dimensional vector whose components are $V_i = \max_{x \in F(I_i)} \log G(x|\mu, \Sigma)$. For the maximization of (8), substitute (15) into (8) to get

$$\sum_{i=1}^{N} w_i y_i (V_i - \theta) = \sum_{i=1}^{N} w_i y_i V_i \qquad (16)$$

In 16, the additional parameter θ is multiplied by the sum of signed weights, which is 0 according to property (11). In Section 2.4 we briefly describe a weak learner which optimizes (16) using gradient ascent on μ.

Optimizing criterion (9) is more difficult, as its denominator $||h||_2 = ||V - \theta \cdot \vec{1}||$ does depend on θ. However, the minimization of this denominator w.r.t θ is easy given V, and the optimal value for θ is the mean of V. The weak learner's task in this case is hence to choose μ, Σ such that $V(\mu, \Sigma)$ maximizes

$$\frac{\sum_{i=1}^{N} w_i y_i V_i}{\sqrt{\sum_{i=1}^{N} (V_i - \frac{1}{N} \sum_{i=1}^{N} V_i)^2}} \qquad (17)$$

2.4. Weak learners

In order to maximize the score (17) we first use a simple selection learner. This learner only considers models in which μ is set to one of the patches in $\bigcup_{i=1}^{N} F(I_i)$. Given a set of weights the algorithm repeats the following selection procedure K times:

- Sample an image j according to the distribution $\{w_i\}$.

- Randomly select patch c from $F(I_j)$ according to the uniform distribution.

- Compute the score (17) for $V_i = \max_{x \in F(I_i)} \log G(x|c, I)$

The patch c with the highest score is chosen and the model $G(\cdot|c, I)$ is returned.

The weak learners we currently use set $\Sigma = I$ and do not try to maximize the model's score by altering Σ.[1] It can be

---

[1] Choosing a covariance matrix is more delicate than choosing the mean because of positive-defi niteness considerations, and is left for future work.

shown that using this covariance matrix for our score (where the score is computed over two PCA coefficients vectors p, q), the result approximates the normalized correlation between the image patches P, Q (represented respectively by p, q). Formally

$$\log G(Q|P, I) \approx C + \frac{Q \cdot P}{||Q||||P||} \qquad (18)$$

The quality of the approximation depends only on the variance preservation quality of the PCA transformation.

The second weak learner we consider is not limited to real image patches. Instead, this weak learner searches for the model's mean μ using stochastic gradient ascent dynamics. Since this process involves differentiation, we note that the derivatives of the score (16) are much simpler than those of (17) and can therefore be computed more efficiently. Hence we choose to optimize (16) for the second weak learner. The derivative of (16) w.r.t μ is

$$\frac{d}{d\mu} \sum_{i=1}^{N} w_i^k y_i \max_{x \in F(I_i)} G(x|\mu, I) = \sum_{i=1}^{N} w_i^k y_i (x_i^* - \mu) \quad (19)$$

where $x_i^* = \max_{x \in F(I_i)} G(x|\mu, I)$.

Our gradient based weak learner chooses its initial model using the selection based weak-learner (using $K = 100$). It then performs stochastic gradient ascent as follows :

- Sample an image i according to the sample weights w_i.

- Update the model's mean $\mu = \mu + \eta y_i (x_i^* - \mu)$ for small $\eta > 0$

- Recompute the maximal scoring patches
$x_i^* = \max_{x \in F(I_i)} G(x|\mu, I) \quad i = 1, .., N$.

This process is repeated and η is gradually decreased until no further score improvement is achieved.

3. Experimental results

Datasets We used the Caltech datasets compiled by [3], which are publicly available.[2] The database consists of 4 sets of object classes: Faces (450 images), Motorbikes (800), Airplanes (800) and Cars (rear view, 800 images). Two background datasets are also provided: General background dataset, and a Road background dataset. Each of the images was represented using $S = 200$ features as described in Section 2.1.[3] All the datasets except the face dataset include significant scale variation. For fair comparison we used exactly the same train and test object images as in [3]. We also used half of the background images as training data, and the remaining half as test data.

---

[2] http://www.robots.ox.ac.uk/ vgg/data
[3] We thank Rob Fergus for his help with the parameter settings of the Kadir and Brady detector.
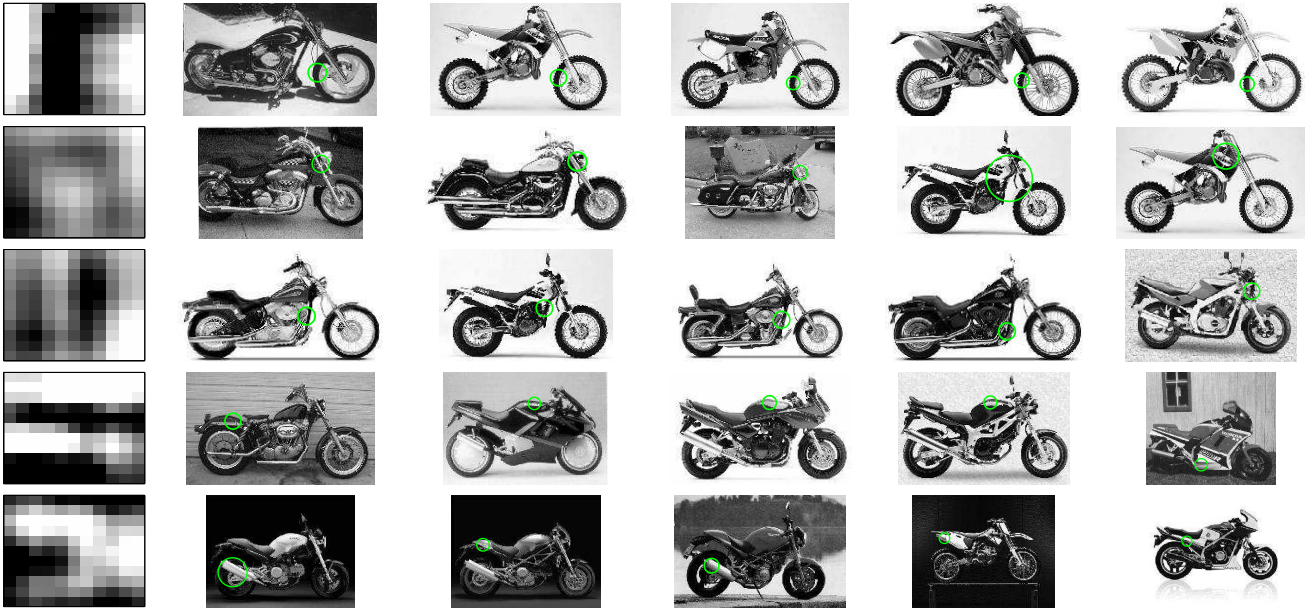
Figure 1. Results of our gradient based algorithm on the Motorbikes dataset. In each row the first image presents a part model. The remaining images are the model's highest scoring images. A circle is drawn to mark the maximal scoring patch in each image. Results are best seen in color.

Experimental setup In each of our experiments we trained a classifier using one of the object datasets and one of the background datasets, resulting in 4 different tests. All the parameters of the algorithm were kept constant over all the experiments. On each of the tests, we compared the performance of the following: (1) boosting with the selection based weak learner and $K = 2000$, (2) boosting with the gradient based learner, (3) the results reported in [3] and (4) the results reported in [1]. Our boosting algorithm was run for 60 iterations.

Results Figs. 1-2 show some example part models and test images for two of the datasets: Motorbikes and Faces. The models were learned using the gradient based learner. As can be seen the part models have interesting and identifiable semantics. For the face data set about $40-50$ out of the 60 models used by the algorithm are of this type. Analysis of the part models shows that in many cases, a distinguished object part (e.g a wheel, or an eye) is modeled using a number of model parts (12 for the wheel, 10 for the eye). In this sense our model seems to describe each object part using a mixture model. Fig. 3 shows test Error as a function of the number of parts in the learnt model.

For comparison and performance evaluation, Table 3 presents the test errors of the various methods under consideration. All the object datasets were trained and tested against a 'General background' (except for the Cars Rear dataset which was trained and tested against 'Road background'). It is important to note here that the results of [3]
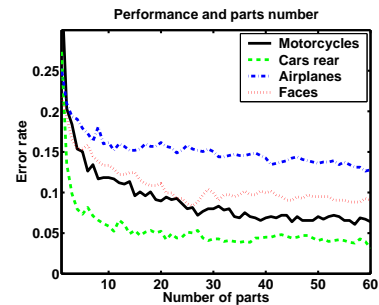
Figure 3. Test Error as a function of the number of parts in the learnt model.

quoted below (when using the general background) were obtained using scale-normalized images, i.e. each object image was manually rescaled so the objects will be of the same size. In the experimental evaluation of our own algorithm, images were not individually rescaled.

Test Errors using General background				
Data Name	**Selection learner**	**Gradient learner**	**Fergus et. al**	**Opelt et. al**
Motorbikes	7.2%	6.9%	7.5%	7.8%
Cars Rear	6.8%	2.3%	9.7%	8.9%
Airplanes	14.2%	10.3%	9.8%	11.1%
Faces	7.9%	8.35%	3.6%	6.5%

The parts chosen by our two selection methods show interesting differences. The selection-based learner uses patches from the object images as part models. The models trained using the gradient-based learner, although ini-

Figure 2. Results of our gradient based algorithm on the Faces dataset. See Fig. 1 for details. Results are best seen in color.

tialized with a patch selected by the selection-based learner, are modified by the algorithm's dynamics in order to maximize the weak learner's discriminative performance. When comparing these models to real image patches, the gradient based models seem accentuated, or "cartoon" like. Interestingly, these models seem reminiscent of the features used by [12, 6]. However, in our case they are learned from the training data instead of being pre-defined.

4. Discussion

We proposed a novel algorithm for object class recognition using part-based representations, where parts are modeled generatively. The proposed classifier was trained using a discriminative boosting algorithm. In our current work, we suggested a simple model in which parts are independent and did not model any relations between the parts, such as their relative location and scale. In our future work we plan to extend the algorithm to incorporate relations between parts. In this respect, the major advantage of our approach is that our generative model allows us to introduce such extensions in a relatively straightforward manner, while keeping the computational cost of the algorithm linear in the number of parts and the number of features.

References

[1] Opelt A., Fussenegger M., Pinz A., and Auer P. Weak hypotheses and boosting for generic object detection and recognition. In *ECCV*, 2004.

[2] Shivani A. and Roth D. Learning a sparse representation for object detection. In *ECCV*. Springer, 2002.

[3] R. Fergus, P.Perona, and A. Zisserman. Object class recognition by unsupervised scale invariant learning. In *CVPR*. IEEE Computer Society, 2003.

[4] T. Kadir and M. Brady. Scale, saliency and image description. *IJCV*, 45(2):83–105, November 2001.

[5] L. Mason, J. Baxter, P. Bartlett, and M. Frean. Boosting algorithms as gradient descent in function space. In *NIPS*, pages 512–518, 2000.

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

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

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

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

[10] V.N. Vapnik. *Statistical learning theory*. John wiley and sons, 1998.

[11] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *ICCV*, 2003.

[12] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.