One-Shot Learning of Object Categories

Li Fei-Fei, *Member*, *IEEE*, Rob Fergus, *Student Member*, *IEEE*, and Pietro Perona, *Member*, *IEEE*

Abstract—Learning visual models of object categories notoriously requires hundreds or thousands of training examples. We show that it is possible to learn much information about a category from just one, or a handful, of images. The key insight is that, rather than learning from scratch, one can take advantage of knowledge coming from previously learned categories, no matter how different these categories might be. We explore a Bayesian implementation of this idea. Object categories are represented by probabilistic models. Prior knowledge is represented as a probability density function on the parameters of these models. The posterior model for an object category is obtained by updating the prior in the light of one or more observations. We test a simple implementation of our algorithm on a database of 101 diverse object categories. We compare category models learned by an implementation of our Bayesian approach to models learned from by Maximum Likelihood (ML) and Maximum A Posteriori (MAP) methods. We find that on a database of more than 100 categories, the Bayesian approach produces informative models when the number of training examples is too small for other methods to operate successfully.

Index Terms—Recognition, object categories, learning, few images, unsupervised, variational inference, priors.

1 Introduction

Recognition is one of the most useful functions of our visual system. We recognize materials (marble, orange peel), surface properties (rough, cold), objects (my car, a willow tree), and scenes (a thicket of trees, my kitchen) at a glance and without touching them. We recognize both individuals (my mother, my office), as well as categories (a 1960s hairdo, a frog). By the time we are six years old, we recognize more than 10^4 categories of objects [4], and keep learning more throughout our life. As we learn, we organize both objects and categories into useful and informative taxonomies and relate them to language. Replicating these abilities in the machines that surround us would profoundly affect the practical aspects of our lives, mostly for the better. Certainly, this is the most exciting and difficult puzzle that faces computational vision scientists and engineers in this decade.

A rich palette of diverse ideas has been proposed during the past few years, especially on the problem of recognizing objects and object categories (see our brief review of the literature below). There is broad consensus on the fact that models need to capture the great diversity of forms and appearances of the objects that surround us. This means models containing hundreds, sometimes thousands, of parameters. It is common knowledge in statistics that estimating a given number of parameters requires a many-fold larger number of training examples—as a consequence, learning one object category requires a batch process involving thousands or tens of thousands of training examples [13], [34], [39], [36].

- *L. Fei-Fei is with the University of Illinois Urbana-Champaign, 405 N. Mathews Ave., MC 251, Urbana, IL 61801. E-mail: feifeili@uiuc.edu.*
- *R. Fergus is with the University of Oxford, Parks Road, Oxford, OX1 3PJ, UK. E-mail: fergus@robots.ox.ac.uk.*
- *P. Perona is with the California Institute of Technology, Mail Code 136-93, Pasadena, CA 91125. E-mail: perona@vision.caltech.edu.*

Manuscript received 30 Aug. 2004; revised 12 July 2005; accepted 12 July 2005; published online 14 Feb. 2006.
Recommended for acceptance by R. Basri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0460-0804.

Unfortunately, it is often difficult and expensive to acquire large sets of training examples. Compounding this problem, most algorithms for learning categories require that each training exemplar be aligned (typically by hand) with a prototype. This becomes particularly problematic when fiducial points are not readily identifiable (can we find a natural alignment for images of octopuses, of cappuccino machines, of bonsai trees?). This is a large practical obstacle on the way to learning thousands of object categories. It would be far better if we managed to find ways to train new categories with few examples.

Is there any hope? We believe so. A young child learns many categories per day [4]. It seems unlikely that this would require a large set of training images for each category as well as much supervision.

We hypothesize that, once a few categories have been learned the hard way, some information may be abstracted from that process to make learning further categories more efficient. In other words, we should be able to make use of the knowledge that has been gained so far rather than starting from scratch each time we learn a new category. We pursue here this hypothesis in a Bayesian setting: We extract "general knowledge" from previously learned categories and represent it in the form of a prior probability density function in the space of model parameters. Given a training set, no matter how small, we update this knowledge and produce a posterior density, which is then used for detection/recognition. Our experiments show that this is a productive approach and that, indeed, some useful information about categories may be obtained from a few, even one, training example.

We begin with a brief review of the literature in Section 2. A detailed review of the mathematical framework of our recognition system follows in Section 3. Section 4 briefly introduces our methods for learning the model. Detailed derivations are given in [9]. We then proceed to test our ideas experimentally. In Section 5, we give implementational details for each stage of the system, from feature detection (Section 5.1) to the experimental setup for learning and recognition (Section 6.2). In Section 6.3, we demonstrate the algorithm with a walkthough for the motorbike category. We

0162-8828/06/$20.00 © 2006 IEEE Published by the IEEE Computer Society

then contrast our Bayesian learning algorithm with the traditional ML approach in Section 6.4. In Section 6.5, we show a variety of experiments run on the 101 object categories, including: a comparison with MAP algorithm, deliberate degradation of the system, and a confusion table for all 101 categories. Section 7 concludes the paper. For convenience, we will denote our algorithm as the "Bayesian One-Shot" algorithm throughout the text.

2 Literature Review

Researchers in this area face three main challenges. Representation: How should we model objects and categories? Learning: How may we acquire such models? Detection/recognition: Given a new image, how do we detect the presence of a known object/category amongst clutter, and despite occlusion, viewpoint, and lighting changes? The great richness and diversity of methods and ideas in the literature indicates that these issues are far from being settled. However, there is broad consensus on a few significant points. First of all, the shape and appearance of the objects that surround us is complex and diverse; therefore, models should be rich (lots of parameters, heterogeneous descriptors). Second, the appearance of objects within a given category may be highly variable, therefore, models should be flexible enough to handle this. Third, in order to handle intraclass variability and occlusion, models should be composed of features, or parts, which are not required to be detected in all instances; the mutual position of these parts constitutes further model information. Fourth, it is difficult, if not impossible, to model class variability using principled a priori techniques; it is best to learn the models from training examples. Fifth, computational efficiency must be kept in mind.

Work on recognition may be divided into two groups: recognition of individual objects [16], [20], [27], [31] and recognition of categories [2], [5], [13], [24], [26], [32], [33], [34], [35], [39], [36]. Individual objects are easier to handle, therefore, more progress has been made on efficient recognition [27], lighting-invariant [27], [28], and viewpoint-invariant [22], [31] representations and recognition. Categories are more general, requiring more complex representations and are more difficult to learn; most work has therefore focused on modeling and learning. Viewpoint and lighting have not been treated explicitly (exceptions include [38], [40]), but rather treated as an additional source of in-class variability.

We are interested in the problem of learning and recognition of categories (as opposed to individual objects). While the literature proposed learning methods that require batch processing of thousands of training examples, the present work focuses on the previously unexplored problem of efficient learning: How could we estimate models of categories from very few, one in the limit, training examples? Most researchers have focused on special-interest categories: human faces [34], [36], pedestrians [37], handwritten digits [24], and automobiles [34], [13]. Instead, we wish to develop techniques that apply equally well to any category that a human would readily recognize. With this objective in mind, we carried out our experiments on a large number of categories.

Another aspect that we wish to emphasize is the ability to learn with minimal supervision. We prefer to develop methods that do not rely on hand-alignment of the training examples, for the reasons mentioned in the introduction. For this reason, we use statistical models and probabilistic detection techniques developed by [5], [13], [26], [39], which will be reviewed in Section 3.2. A comprehensive treatment of these models may be found in Weber's [41] and Fergus' [42] PhD theses.

3 Theoretical Approach

3.1 Overall Bayesian Framework

Let's say that we are looking for a flamingo bird in a query image that is presented to us. To decide whether there is a flamingo bird or not, we compare the probability of a flamingo being present in the image with the probability of only background clutter being present in the image. The decision is simple: If the probability of a flamingo being present is higher, we decide this image contains an instance of a flamingo. If it is the other way around, we decide there is no flamingo. To compute the probability of a flamingo being present in an image, we need a model of a flamingo, which we learn from a set of training images containing examples of flamingos. Then, we could compare this probability with the background model and, in turn, make our final decision.

We can now translate the above events into a probabilistic framework. Let $\mathcal{I}$ be the query image, which may contain an example of the foreground category $\mathcal{O}_{fg}$. The alternative is that it contains background clutter belonging to a generic background category $\mathcal{O}_{bg}$. $\mathcal{I}_t$ is the set of training images that we have used as the foreground category. Now, the decision of whether this query image $\mathcal{I}$ has the foreground object or not can be written in the following way:

$$R = \frac{p(\mathcal{O}_{fg}|\mathcal{I},\mathcal{I}_t)}{p(\mathcal{O}_{bg}|\mathcal{I},\mathcal{I}_t)} = \frac{p(\mathcal{I}|\mathcal{I}_t,\mathcal{O}_{fg})\ p(\mathcal{O}_{fg})}{p(\mathcal{I}|\mathcal{I}_t,\mathcal{O}_{bg})\ p(\mathcal{O}_{bg})}. \tag{1}$$

If R, the ratio of the class posteriors, is greater than some threshold, T, then we decide the image contains an instance of the object. If it is less than T, then the image does not contain the object. In (1), we use Bayes Rule for the expansion, giving us a ratio of likelihoods and a ratio of priors on the object categories. We can now further expand (1) by introducing a parametric model for the foreground and background category, whose parameters are $\boldsymbol{\theta}$ and $\boldsymbol{\theta}_{bg}$, respectively:

$$\begin{aligned} R &\propto \frac{\int p(\mathcal{I}|\boldsymbol{\theta},\mathcal{O}_{fg})p(\boldsymbol{\theta}|\mathcal{I}_t,\mathcal{O}_{fg})\ d\boldsymbol{\theta}}{\int p(\mathcal{I}|\boldsymbol{\theta}_{bg},\mathcal{O}_{bg})p(\boldsymbol{\theta}_{bg}|\mathcal{I}_t,\mathcal{O}_{bg})\ d\boldsymbol{\theta}_{bg}} \\ &= \frac{\int p(\mathcal{I}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{I}_t,\mathcal{O}_{fg})\ d\boldsymbol{\theta}}{\int p(\mathcal{I}|\boldsymbol{\theta}_{bg})p(\boldsymbol{\theta}_{bg}|\mathcal{I}_t,\mathcal{O}_{bg})\ d\boldsymbol{\theta}_{bg}}. \end{aligned} \tag{2}$$

The ratio of priors, $\frac{p(\mathcal{O}_{fg})}{p(\mathcal{O}_{bg})}$, is a constant, thus it is omitted in (2) since it maybe be incorporated into the decision threshold. In addition, we have simplified $p(\mathcal{I}|\boldsymbol{\theta},\mathcal{O}_{fg})$ and $p(\mathcal{I}_t|\boldsymbol{\theta}_{bg},\mathcal{O}_{bg})$ into $p(\mathcal{I}|\boldsymbol{\theta})$ and $p(\mathcal{I}_t|\boldsymbol{\theta}_{bg})$, respectively. The learning procedure involves estimating $p(\boldsymbol{\theta}|\mathcal{I}_t,\mathcal{O}_{fg})$, the distribution of model parameters given the training images. Once this is known, we can evaluate R by integrating out over $\boldsymbol{\theta}$. We now look at the particular object model used.

3.2 The Object Category Model

Our chosen representation is a Constellation model [6], [39], [13]. Given a query image, $\mathcal{I}$, we find a set of N interesting regions in the image. From these N regions, we obtain two variables: $\mathcal{X}$—the locations of the regions and $\mathcal{A}$—the appearances of the regions. Section 5.1 gives details of how

$\mathcal{X}$ and $\mathcal{A}$ are obtained. It is $\mathcal{X}$ and $\mathcal{A}$ that we now model, $\mathcal{I}$ no longer being used directly. Similarly, in the case of the training images $\mathcal{I}_t$, we obtain $\mathcal{X}_t$ and $\mathcal{A}_t$. Thus, (2) becomes:

$$\begin{aligned} R &\propto \frac{\int p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta},\mathcal{O}_{fg})p(\boldsymbol{\theta}|\mathcal{X}_t,\mathcal{A}_t,\mathcal{O}_{fg})\,d\boldsymbol{\theta}}{\int p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta}_{bg},\mathcal{O}_{bg})p(\boldsymbol{\theta}_{bg}|\mathcal{X}_t,\mathcal{A}_t,\mathcal{O}_{bg})\,d\boldsymbol{\theta}_{bg}} \\ &= \frac{\int p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta})p(\boldsymbol{\theta}|\mathcal{X}_t,\mathcal{A}_t,\mathcal{O}_{fg})\,d\boldsymbol{\theta}}{\int p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta}_{bg})p(\boldsymbol{\theta}_{bg}|\mathcal{X}_t,\mathcal{A}_t,\mathcal{O}_{bg})\,d\boldsymbol{\theta}_{bg}}. \end{aligned} \tag{3}$$

We now examine likelihoods $p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta})$ and $p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta}_{bg})$, where, in the general case, we have a mixture of constellation models, with Ω components:

$$\begin{aligned} p(\mathcal{X},\mathcal{A}|\,\boldsymbol{\theta}) &= \sum_{w=1}^{\Omega}\sum_{\mathbf{h}\in H} p(\mathcal{X},\mathcal{A},\mathbf{h},w|\,\boldsymbol{\theta}) \\ &= \sum_{w=1}^{\Omega} p(w|\boldsymbol{\pi}) \sum_{\mathbf{h}\in H} \underbrace{p\big(\mathcal{A}|\mathbf{h},\boldsymbol{\theta}^{\mathcal{A}}_w\big)}_{Appearance}\underbrace{p\big(\mathcal{X}|\mathbf{h},\boldsymbol{\theta}^{\mathcal{X}}_w\big)}_{Shape} p(\mathbf{h}|\boldsymbol{\theta}_w), \end{aligned} \tag{4}$$

where $\boldsymbol{\theta}=\{\boldsymbol{\pi},\boldsymbol{\theta}^{\mathcal{A}},\boldsymbol{\theta}^{\mathcal{X}}\}$ and $p(\mathbf{h}|\boldsymbol{\theta}_w)$ is a constant. The shape, $\mathcal{X}$, and appearance, $\mathcal{A}$, are assumed to be independent. Typically, a constellation model would have P $(3\sim 7)$ diagnostic features, or parts. But, there are N (up to 100) interest points, or candidate features in the image. We therefore introduce an indexing variable $\mathbf{h}$, which we call a *hypothesis*. $\mathbf{h}$ is a vector of length P, where each entry is between 1 and N, which allocates a particular feature to a model part. Any unallocated features are assumed to belong to the background of the image. The set of all hypotheses H consists of all valid allocations of features to the parts; consequently, $|H|$, the total number of hypotheses is $O(N^P)$. For simplicity, we assume the background model is fixed and has a single parameter value, $\boldsymbol{\theta}_{bg}$, thus the integral in the denominator of (3) collapses to $p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta}_{bg})$. If we believe no object to be present (the $\mathcal{O}_{bg}$ case), then only one hypothesis exists, $\mathbf{h}_0$, the null hypothesis, where all detections are assigned to the background. Hence, the denominator becomes:

$$\begin{aligned} p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta}_{bg}) &= p(\mathcal{X},\mathcal{A},\mathbf{h}_0|\boldsymbol{\theta}_{bg}) \\ &= p\Big(\mathcal{A}|\mathbf{h}_0,\boldsymbol{\theta}^{\mathcal{A}}_{bg}\Big)p\Big(\mathcal{X}|\mathbf{h}_0,\boldsymbol{\theta}^{\mathcal{X}}_{bg}\Big)p\big(\mathbf{h}_0|\boldsymbol{\theta}_{bg}\big). \end{aligned} \tag{5}$$

Since this expression is constant for given $\mathcal{X}$ and $\mathcal{A}$, we can use it to cancel terms in the numerator of (3).

The model encompasses the important properties of an object: shape and appearance, both in a probabilistic way. This allows the model to represent both geometrically constrained objects (where the shape density would have a small covariance, e.g., a face) and objects with distinctive appearance but lacking geometric form (the appearance densities would be tight, but the shape density would now be looser, e.g., an animal principally defined by its texture such as a zebra). The following assumptions are made in the model: Shape is independent of appearance; for shape, the joint covariance of the parts' position is modeled, while for appearance, each part is modeled independently. In the experiments reported here, we use a slightly simplified version of the model presented in [13] by removing the terms involving occlusion and statistics of the feature finder since these are relatively unimportant when we only have a few images to train from.

3.2.1 Appearance

Each feature's appearance is represented as a point in some appearance space, defined in Section 5.1. For now, we could think of each feature being represented by a vector whose values are related to the gray-value pixel intensities of the small neighborhood of the feature. For a given mixture component, each part p has a Gaussian density within this space, with mean and precision parameters $\boldsymbol{\theta}^{\mathcal{A}}_{p,w}=\{\boldsymbol{\mu}^{\mathcal{A}}_{p,w},\boldsymbol{\Gamma}^{\mathcal{A}}_{p,w}\}$, which is independent of other parts' densities. The background model has the same form, with fixed parameters $\boldsymbol{\theta}^{\mathcal{A}}_{bg}=\{\boldsymbol{\mu}^{\mathcal{A}}_{bg},\boldsymbol{\Gamma}^{\mathcal{A}}_{bg}\}$. Note that $\boldsymbol{\Gamma}^{\mathcal{A}}_{p,\omega}$ and $\boldsymbol{\Gamma}^{\mathcal{A}}_{bg}$ are diagonal matrices. Each feature selected by the hypothesis is evaluated under the appropriate part density with features not selected being evaluated under the background model:

$$p\big(\mathcal{A}|\mathbf{h},\boldsymbol{\theta}^{\mathcal{A}}_w\big)=\prod_{p=1}^{P}\mathcal{G}\Big(\mathcal{A}(\boldsymbol{h}_p)|\boldsymbol{\mu}^{\mathcal{A}}_{p,w},\boldsymbol{\Gamma}^{\mathcal{A}}_{p,w}\Big)\prod_{j=1,\,j\setminus\boldsymbol{h}}^{N}\mathcal{G}\Big(\mathcal{A}(j)|\boldsymbol{\mu}^{\mathcal{A}}_{bg},\boldsymbol{\Gamma}^{\mathcal{A}}_{bg}\Big), \tag{6}$$

where $\mathcal{G}$ is the Gaussian distribution and j represents features not assigned to a part in hypothesis $\mathbf{h}$. In addition, we adopt the notation $\mathbf{h}_p$ in (6) to indicate the feature belonging to the pth part of $\mathbf{h}$. If no object is present, then all features are modeled by the background:

$$p(\mathcal{A}|\mathbf{h}_0,\boldsymbol{\theta}^{\mathcal{A}}_{bg})=\prod_{j=1}^{N}\mathcal{G}\Big(\mathcal{A}(j)|\boldsymbol{\mu}^{\mathcal{A}}_{bg},\boldsymbol{\Gamma}^{\mathcal{A}}_{bg}\Big). \tag{7}$$

Note that $p(\mathcal{A}|\mathbf{h}_0,\boldsymbol{\theta}^{\mathcal{A}}_{bg})$ is a constant for a given image; therefore, it can be brought inside the integral and summation over all hypotheses in (3) and (4). This cancels with all other background hypotheses except the true foreground hypothesis $\mathbf{h}$ in (6):

$$\frac{p\big(\mathcal{A}|\mathbf{h},\boldsymbol{\theta}^{\mathcal{A}}_w\big)}{p\Big(\mathcal{A}|\mathbf{h}_0,\boldsymbol{\theta}^{\mathcal{A}}_{bg}\Big)}=\prod_{p=1}^{P}\frac{\mathcal{G}\Big(\mathcal{A}(\boldsymbol{h}_p)|\boldsymbol{\mu}^{\mathcal{A}}_{p,w},\boldsymbol{\Gamma}^{\mathcal{A}}_{p,w}\Big)}{\mathcal{G}\Big(\mathcal{A}(\boldsymbol{h}_p)|\boldsymbol{\mu}^{\mathcal{A}}_{bg},\boldsymbol{\Gamma}^{\mathcal{A}}_{bg}\Big)}. \tag{8}$$

3.2.2 Shape

The shape of each constellation model component may be represented by a joint Gaussian density of the locations of features for a hypothesis, after they have been transformed into a scale and translation-invariant space. Translation invariance is achieved by using the left-most feature in $\mathbf{h}$ as a landmark and translating all feature locations relative to it. Scale invariance is obtained by taking the scale of the landmark feature and using it to normalize the relative locations of the other features [13]. We assume a uniform density α^{-1} for the position of the object, where α is the image area. The relative location of the parts is modeled by a $2(P-1)$-dimensional Gaussian, with a uniform background model for unallocated features:

$$p\big(\mathcal{X}|\mathbf{h},\boldsymbol{\theta}^{\mathcal{X}}_w\big)=\alpha^{-1}\;\mathcal{G}\big(\mathcal{X}(\mathbf{h})|\boldsymbol{\mu}^{\mathcal{X}}_w,\boldsymbol{\Gamma}^{\mathcal{X}}_w\big)\;\;\alpha^{-(N-P)}, \tag{9}$$

where $\boldsymbol{\theta}^{\mathcal{X}}_w=\{\alpha,\boldsymbol{\mu}^{\mathcal{X}}_w,\boldsymbol{\Gamma}^{\mathcal{X}}_w\}$. For the null hypothesis, $p(\mathcal{X}|\mathbf{h}_0,\boldsymbol{\theta}^{\mathcal{X}}_{bg})=\alpha^{-N}$, which is a constant, so we cancel with all other background hypotheses except the true foreground hypothesis $\mathbf{h}$ in (9):

$$\frac{p\big(\mathcal{X}|\mathbf{h},\boldsymbol{\theta}^{\mathcal{X}}_w\big)}{p\Big(\mathcal{X}|\mathbf{h}_0,\boldsymbol{\theta}^{\mathcal{X}}_{bg}\Big)}=\alpha^{P-1}\;\mathcal{G}\big(\mathcal{X}(\mathbf{h})|\boldsymbol{\mu}^{\mathcal{X}}_w,\boldsymbol{\Gamma}^{\mathcal{X}}_w\big). \tag{10}$$

Additionally, to reduce the number of hypotheses that must be considered in each frame, we impose an ordering constraint on each hypothesis's shape, such that the x-coordinate of each part must be monotonically increasing. This reduces the number of hypotheses that must be considered by $P!$ and provides a useful constraint in the learning process.

3.3 Discussion of the Model

We make some comments concerning the model:

1. $\mathcal{X}(\mathbf{h}) \in \mathbb{R}^{2P-2}$ and $\mathcal{A}(\mathbf{h}) \in \mathbb{R}^{kP}$. Thus, for $k = 10$ (dimension of the appearance descriptor), $P = 4$ (number of parts), the shape term has $6 + 21 = 27$ (mean + full covariance matrix) parameters. The appearance term has $40 + 40 = 80$ (mean + diagonal covariance matrix) parameters, thus the model has $27 + 80 = 107$ parameters in total.
2. The total number of hyperparameters for $k = 10$, $P = 4$ is 109 since $\boldsymbol{m}$ and $\boldsymbol{B}$ have the same dimensionality as $\boldsymbol{\mu}$, $\boldsymbol{\Gamma}$. Additionally, β and a (both real numbers) exist for both shape and appearance terms: $107 + 2 = 109$.
3. The constellation model is a generative model of the output of an interest region detector, not the image pixels. Hence, the performance of the model is dependent on the performance of the detectors themselves. See Section 6.5.3 for an investigation into this dependency.
4. In our representation, there is nothing to prevent patches from overlapping, which could lead to overcounting of the evidence for the model. However, given a relatively low number of features per image, this should not be a major problem.
5. The shape model presented above uses a joint density over all parts, thus, the data association problem has complexity $O(N^P)$. While this is the most thorough approach to modeling the location of parts, it presents a major computational bottleneck. Imposing conditional independence by the use of a tree-structured model would reduce the complexity to $O(N^2P)$ in learning and $O(NP)$ in recognition [11], [15]. However, in doing so, other issues arise, such as how the optimal graph structure should be chosen. Since these issues are in themselves complex and are outside the focus of this paper, for the sake of simplicity, we stick with the complete representation, despite its drawbacks.
6. Our model and representation of shape is suited to compact objects which do not have large amounts of articulation (e.g., human bodies). For such categories, different graph structures and coordinate frames (i.e., the angles between parts) may be more appropriate.
7. Our feature representation is currently confined to textured image patches. Alternative representations such as curve contours, which model the outline of the object, could also be used with little modification to the underlying model [14], [12]. This would allow the model to handle categories where the outline of the object is more important than its interior (e.g., bottles).
8. Currently, the background model is very simple: A uniform shape distribution and a single Gaussian distribution for appearance. Their crude nature is a consequence of the requirement, for efficiency, that the denominator in (3) must be able to cancel with the numerator, making evaluation of the likelihood ratio simple. The parametric assumptions of the background model were tested by examining the distribution of thousands of detections from an assorted collection of images. Our observation was that these assumptions were reasonably accurate.
9. The framework describes object detection (i.e., object present or absent), however, it can easily be extended to localization by using the best hypothesis in each image (e.g., by taking a bounding box around it). Multiple instances per image can also be found by a greedy approach: finding the best hypothesis; summing over all hypotheses around its neighborhood to give a value of R for a subwindow of the image; removing all features within the subwindow and repeating until no subwindows with R greater than a given threshold can be found.
10. Our model is formulated as a mixture of Gaussians (4). In practice, we use a single mixture component in this paper for all of the experiments. Weber et al. have demonstrated that, by increasing the number of mixture components, the model is capable of representing different aspects of the object due to pose variations [38].

3.4 Form of the Parameter Posterior

In computing R, we must evaluate the integral $\int p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O})\, d\boldsymbol{\theta}$. In Section 3.2, the form of $p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta})$ was considered. We now look at the posterior of $\boldsymbol{\theta}$, $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O})$. Before we consider how this density might be estimated, its form must be decided upon. Since the integral above is typically impossible to solve analytically, we look at various forms of $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O})$ that approximate the true density while making the integral tractable.

3.4.1 Maximum Likelihood (ML) and Maximum A Posteriori (MAP)

If we assume that the model distribution $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O})$ is highly peaked, we could approximate it with a δ function at $\boldsymbol{\theta}^*$: $\delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$. This allows the integral in (3) to collapse to $p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta}^*)$, whose functional form is given by (4).

There are two ways of obtaining $\boldsymbol{\theta}^*$. The simplest one is Maximum Likelihood (ML) estimation [39], [13]. Here, $\boldsymbol{\theta}^* = \boldsymbol{\theta}^{\mathrm{ML}}$ is computed by picking the $\boldsymbol{\theta}$ that gives rise to the highest likelihood value of the training data:

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^{\mathrm{ML}} = \underset{\theta}{argmax}\; p(\mathcal{X}_t, \mathcal{A}_t|\boldsymbol{\theta}). \tag{11}$$

If we had some prior knowledge about $\boldsymbol{\theta}$, we could also use this information to help estimate $\boldsymbol{\theta}^*$. The idea is to weigh the likelihood of training examples at $\boldsymbol{\theta}$ by the prior probability of $\boldsymbol{\theta}$ at that point. This is called the Maximum A Posteriori (MAP) estimation.

$$\boldsymbol{\theta}^* = \boldsymbol{\theta}^{\mathrm{MAP}} = \underset{\theta}{argmax}\; p(\mathcal{X}_t, \mathcal{A}_t|\boldsymbol{\theta}) p(\boldsymbol{\theta}). \tag{12}$$

The form of $p(\boldsymbol{\theta})$ needs to be chosen carefully to ensure that the estimation procedure is efficient. In [10], we shall give a more detailed account of $p(\boldsymbol{\theta})$ and methods for estimating $\boldsymbol{\theta}^{\mathrm{MAP}}$.

Both ML and MAP assume a well peaked $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O})$ so that $\delta(\boldsymbol{\theta} - \boldsymbol{\theta}^*)$ is a suitable estimate of the entire distribution.

But, when there is a limited number of training examples, the distribution may not be well peaked, in which case both ML and MAP are likely to yield poor models.

3.4.2 Other Inference Methods

Sampling methods. At the other extreme, we can use numerical methods such as Gibbs Sampling [18] or Markov-Chain Monte-Carlo (MCMC) [19] to give an accurate estimate of the integral in (3), but these can be computationally very expensive. In the constellation model, the dimensionality of $\boldsymbol{\theta}$ is large (~ 100) for a reasonable number of parts, making MCMC methods impractical for our problem. Additionally, the use of sampling-based methods is something of an art: Issues such as what sampling regime to use have no simple answer. Hence, they are less attractive as compared with methods giving a distinct solution.

Recursive Approximations. A variety of variational approximations exist that are recursive or incremental in nature [21], [29]. In such schemes, the data points are processed sequentially with the (approximate) marginal posterior $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O})$ being updated after each new data point. We explore one of such methods in [9].

3.4.3 Conjugate Densities

The final approach is to assume that $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg})$ has a specific parametric form, such that the integral in (3) has a closed-form solution. Recalling the numerator of (3):

$$\int p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg})\, d\boldsymbol{\theta}. \tag{13}$$

Our goal is to find a parametric form of $p(\boldsymbol{\theta})$ such that the learning of $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg})$ is feasible and the evaluation of (13) is tractable. This could be achieved by taking advantage of a class of prior distributions that are conjugate to their posterior distributions. In other words, a conjugate prior for a given probabilistic model is one for which the resulting posterior has the same functional form as the prior [17]. In the case of $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg})$, we use a Normal-Wishart distribution as its conjugate prior. Given that $p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta})$ was chosen to be a product of Gaussians (in Section 3.2), the entire integral of (13) becomes a multivariate Student's T distribution. Efficient learning schemes exist for estimating the hyper-parameters of the Normal-Wishart distribution [3], having the same computational complexity as standard ML methods. These are introduced in Section 4.

3.5 Recognition Using a Conjugate Density Parameter Posterior

Having specified a functional form for the parameter posterior, we now give the actual equations for use in recognition.

3.5.1 Parameter Distribution

Recall the mixture of constellation models from (4):

$$p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta}) = \sum_{\omega=1}^{\Omega} p(\omega|\boldsymbol{\pi}) \sum_{h=1}^{|H|} p\big(\mathcal{X}(\mathbf{h})|\boldsymbol{\mu}_\omega^{\mathcal{X}}, \boldsymbol{\Gamma}_\omega^{\mathcal{X}}\big) p\big(\mathcal{A}(\mathbf{h})|\boldsymbol{\mu}_\omega^{\mathcal{A}}, \boldsymbol{\Gamma}_\omega^{\mathcal{A}}\big). \tag{14}$$

Each component ω has a mixing coefficient π_ω, a mean of shape and appearance $\boldsymbol{\mu}_\omega^{\mathcal{X}}, \boldsymbol{\mu}_\omega^{\mathcal{A}}$, and a precision matrix of shape and appearance $\boldsymbol{\Gamma}_\omega^{\mathcal{X}}, \boldsymbol{\Gamma}_\omega^{\mathcal{A}}$. Collecting all mixture components and their corresponding parameters together, we obtain an overall parameter vector $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}^{\mathcal{X}}, \boldsymbol{\mu}^{\mathcal{A}}, \boldsymbol{\Gamma}^{\mathcal{X}}, \boldsymbol{\Gamma}^{\mathcal{A}}\}$. Assuming we have now learned the model distribution $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t)$ from a set of training data $\mathcal{X}_t$ and $\mathcal{A}_t$, we define the model distribution in the following way:

$$p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t) = p(\boldsymbol{\pi}) \prod_\omega p\big(\boldsymbol{\mu}_\omega^{\mathcal{X}}|\boldsymbol{\Gamma}_\omega^{\mathcal{X}}\big) p\big(\boldsymbol{\Gamma}_\omega^{\mathcal{X}}\big) p\big(\boldsymbol{\mu}_\omega^{\mathcal{A}}|\boldsymbol{\Gamma}_\omega^{\mathcal{A}}\big) p\big(\boldsymbol{\Gamma}_\omega^{\mathcal{A}}\big), \tag{15}$$

where the mixing component is a symmetric Dirichlet: $p(\boldsymbol{\pi}) = \mathcal{D}ir(\lambda_\omega \boldsymbol{I}_\Omega)$, the distribution over the shape precisions is a Wishart: $p(\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) = \mathcal{W}(\boldsymbol{\Gamma}_\omega^{\mathcal{X}}|a_\omega^{\mathcal{X}}, \boldsymbol{B}_\omega^{\mathcal{X}})$, and the distribution over the shape mean conditioned on the precision matrix is Normal: $p(\boldsymbol{\mu}_\omega^{\mathcal{X}}|\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) = \mathcal{G}(\boldsymbol{\mu}_\omega^{\mathcal{X}}|\boldsymbol{m}_\omega^{\mathcal{X}}, \beta_\omega^{\mathcal{X}}\boldsymbol{\Gamma}_\omega^{\mathcal{X}})$. Together, the shape distribution $p(\boldsymbol{\mu}_\omega^{\mathcal{X}}, \boldsymbol{\Gamma}_\omega^{\mathcal{X}})$ is a Normal-Wishart density [3], [30]. Note that $\{\lambda_\omega, a_\omega, \boldsymbol{B}_\omega, \boldsymbol{m}_\omega, \beta_\omega\}$ are hyper-parameters for defining distributions of model parameters. Identical expressions apply to the appearance component in (15). We will show an empirical way of obtaining these hyper-parameters in Section 6.3.

3.5.2 Closed-Form Calculation of R

Recall that:

$$\begin{aligned} R &= \frac{p(\mathcal{X}, \mathcal{A}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg})}{p(\mathcal{X}, \mathcal{A}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{bg})} \\ &= \frac{\int p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg})\, d\boldsymbol{\theta}}{\int p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta}_{bg}) p(\boldsymbol{\theta}_{bg}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{bg})\, d\boldsymbol{\theta}_{bg}}. \end{aligned} \tag{16}$$

Due to the use of conjugate densities, the integral in the numerator becomes a multimodal multivariate Student's T distribution (denoted by $\mathcal{S}$):

$$\begin{aligned} &p(\mathcal{X}, \mathcal{A}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg}) = \\ &\sum_{\omega=1}^{\Omega} \sum_{h=1}^{|H|} \tilde{\pi}_\omega\, \mathcal{S}\big(\mathcal{X}_h|\, g_\omega^{\mathcal{X}}, \mathbf{m}_\omega^{\mathcal{X}}, \boldsymbol{\Lambda}_\omega^{\mathcal{X}}\big)\, \mathcal{S}\Big(\mathcal{A}_h|\, g_\omega^{\mathcal{A}}, \mathbf{m}_\omega^{\mathcal{A}}, \boldsymbol{\Lambda}_\omega^{\mathcal{A}}\Big), \\ &\text{where } g_\omega = a_\omega + 1 - d \text{ and } \boldsymbol{\Lambda}_\omega = \frac{\beta_\omega + 1}{\beta_\omega g_\omega} \boldsymbol{B}_\omega \text{ and } \tilde{\pi}_\omega = \frac{\lambda_\omega}{\sum_{\omega'} \lambda_{\omega'}}. \end{aligned} \tag{17}$$

Note that d is the dimensionality of the parameter vector $\boldsymbol{\theta}$. The denominator of (16) is a constant, since we only consider a single value of $\boldsymbol{\theta}_{bg}$: $\boldsymbol{\theta}_{bg}^{ML}$, i.e., $p(\boldsymbol{\theta}_{bg}|\mathcal{X}t, \mathcal{A}_t, \mathcal{O}_{bg}) = \delta(\boldsymbol{\theta}_{bg} - \boldsymbol{\theta}_{bg}^{ML})$.

4 Learning Using a Conjugate Density Parameter Posterior

The process of learning an object category is weakly supervised [39], [13]. The algorithm is presented with a number of training images labeled as "foreground images." It assumes that there is an instance of the object category to be learned in each image. But, no other information, e.g., location, size, shape, appearance, etc., is provided apart from minimal preprocessing (see Section 6.1 for details). The algorithm first detects interesting features in these training images and then estimates the parameters of the model densities from these regions. Since the model is linear and Gaussian with conjugate priors, it should have a closed-form solution. However, the discrete indexing variable $\mathbf{h}$, representing the assignment of features to parts, prevents such a solution. Instead, an iterative variational method that resembles the Expectation-Maximization (EM) algorithm [7] is used to estimate the variational posterior. Afterward, recognition is performed on a query image by repeating the process of detecting regions and then evaluating the regions using the model parameters estimated in the learning process.

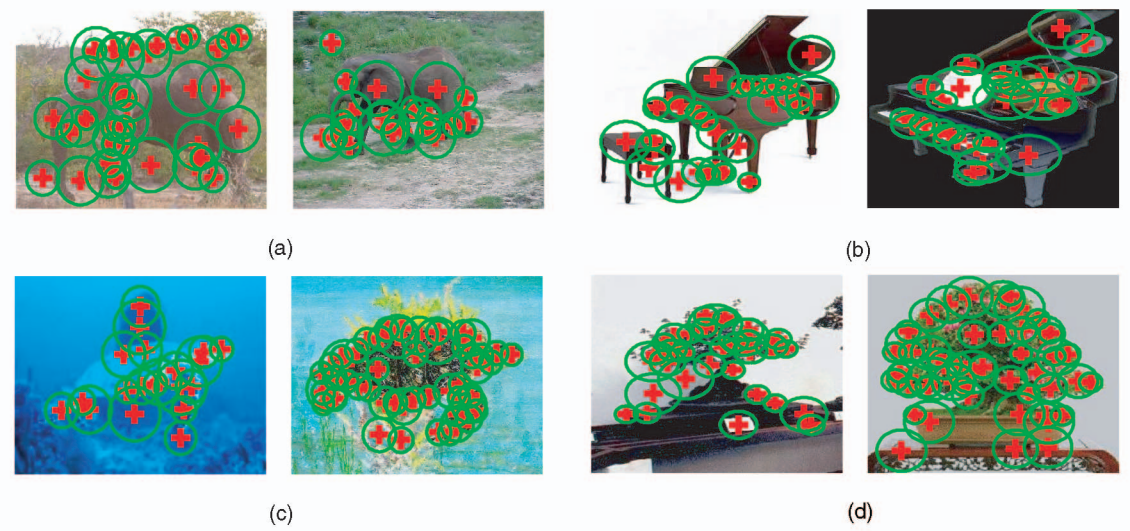


Fig. 1. Output of the feature detector on sample images from four categories. (a) Elephant. (b) Grand piano. (c) Hawksbill. (d) Bonsai tree.

The goal of learning is to obtain a posterior distribution $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg})$ of the model parameters given a set of training data $\{\mathcal{X}_t, \mathcal{A}_t\}$ as well as some prior information. We formulate this learning problem using Variational Bayesian Expectation Maximization (VBEM), applied to a multi-dimensional Gaussian mixture model as introduced by Attias [3]. Detailed derivations of VBEM are given in [10]. In addition, we also give a detailed derivation of the MAP parameter estimation in [10].

5 Implementation

5.1 Feature Detection and Representation

We use the same features as in [13]. They are found using the detector of Kadir and Brady [23]. This method finds regions that are salient over both location and scale. Gray-scale images are used as the input. The most salient regions are clustered over location and scale to give a reasonable number of features per image, each with an associated scale. The coordinates of the center of each feature give us $\mathcal{X}$. This particular feature detector was chosen as it tends to give a small number of informative features per image as compared to other detectors, such as multiscale Harris which give hundreds or thousands of less distinctive features. Fig. 1 illustrates this on images from four data sets. Once the regions are identified, they are cropped from the image and rescaled to the size of a small (11×11) pixel patch. Each patch exists in a 121-dimensional space. We then reduce this dimensionality by using principle component analysis (PCA). A fixed PCA basis, precalculated from the background data sets, is used for this task. We then collect the coefficients of the first 10 principal components from each patch to form $\mathcal{A}$.

5.2 Learning

We now discuss the practical aspects of the Bayesian One-Shot learning procedure—the choice of the prior density, $p(\boldsymbol{\theta})$, and details of the Bayesian One-Shot implementation.

5.2.1 Choice of Prior

One critical issue is the choice of priors for the Dirichlet and Norm-Wishart distributions. In this paper, learning is performed using a single mixture component, i.e., $\Omega = 1$. So, λ is set to 1, since π_ω will always be 1. Ideally, the values for the shape and appearance priors should reflect object models in the real world. In other words, if we have already learned a sufficient number of categories of objects (e.g., hundreds or thousands), we would have a pretty good idea of the average shape/appearance mean and variances given a new object category. In reality, we do not have the luxury of such a number of object categories. We use three categories of object models learned in a ML manner from [13] to form our priors. They are: spotted cats, faces, and airplanes. The hyper-parameters of the prior are then estimated from the parameters of the existing category models. An example of this process is given in Section 6.3.

5.2.2 Details of the Bayesian One-Shot Algorithm

1. Initial conditions are chosen in the following way: Shape and appearance means are set to the means of the training data itself. Covariances are chosen randomly within a sensible range. Namely, they are initialized to be roughly in the order of the average dimensions of the training images.
2. Learning is halted when the largest parameter change per iteration (across all parameters) falls below a certain threshold (10^{-4}) or the maximum number of iterations is exceeded (typically 500). In general, convergence occurs within 100 iterations.
3. Since the model is a generative one, the background images are not used in learning except for one instance: The appearance model has a distribution in appearance space modeling background features. Estimating this from foreground data proven inaccurate, so the parameters are estimated from a set of background images and not updated within the Bayesian One-Shot iteration.
4. Learning a category takes roughly less than a minute on a 2.8 GHz machine when the number of training images is less than 10 and the model is composed of four parts. The algorithm is implemented in Matlab. It is also worth mentioning that the current algorithm does not utilize any efficient search methods, unlike [13]. It has been shown that increasing the number of parts in a constellation model results in greater

recognition power provided enough training examples are given [13]. Were efficient search techniques used, 6-7 parts could be learned, since the Bayesian One-Shot update equations require the same amount of computation as the traditional ML ones. However, all our experiments currently use four part models for both the current algorithm and ML.

6 Experimental Results

6.1 Data Sets

In the first set of experiments, the same four object categories as in [13], [8] were used,[1] namely, human faces, motorbikes, airplanes, and spotted cats. These data sets contain a fair amount of background clutter and scale variation, although each category is presented from a consistent viewpoint.

In addition, two naive subjects collected another data set of 97 object categories for the second set of experiments. The 97 categories were combined with the motorbikes, airplanes, faces, and spotted cats to give a data set of 101 object categories. The names of the 97 new categories were generated by flipping through the pages of the Webster Collegiate Dictionary [1], picking a subset of categories that were associated with a drawing. Using a script, all images returned by the Google Image Search engine for each category name were downloaded. The two subjects then sorted through the images for each category, getting rid of irrelevant images (e.g., a zebra-patterned shirt for the "zebra" category). Fig. 2 shows examples from 101 foreground object categories as well as the background clutter category (obtained by typing "things" into Google).

Minimal preprocessing was performed on the categories. Categories such as motorbike, airplane, cannon, etc., where two mirror image views were present, were manually flipped, so all instances faced in the same direction. Additionally, categories with a predominantly vertical structure were rotated to an arbitrary angle. This is due to the convention that the left-most part of each hypothesis is used as a reference point to translate the rest of the parts (see Section 3.2.2). With vertically orientated structures, the horizontal ordering of the features will be somewhat arbitrary, so artificially giving a large vertical variability.

6.2 Experimental Setup

Each experiment is carried out as follows: Each data set is randomly split into two disjoint sets of equal size. N training images are drawn randomly from the first. A fixed set of 50 are selected from the second, forming the test set. We then learn models using Variational Bayesian, ML, and MAP approaches and evaluate their performance on the test set. For evaluation purposes, we also use 50 images from a background data set. For each category, we vary N from 1 to 6, repeating the experiments 10 times for each value (using a different set of N training images each time) to obtain a more robust estimate of performance. When $N = 1$, ML, and MAP fail to converge, so we only show results for the Bayesian One-Shot algorithm in this case.

When evaluating the models, the task is a binary decision—object present or absent. All performance values are quoted as equal error rates from the receiver-operating characteristic curve (ROC) (i.e., p (True positive) $= 1 - p$ (False alarm)). The ROC curve is obtained by testing the model on 50 foreground test images and 50 background images. For example, a value of 85 percent means that 85 percent of the foreground images are correctly classified but 15 percent of the background images are incorrectly classified (i.e., false alarms).

In all the experiments, the following parameters are used: number of parts in model = 4, number of PCA dimensions for each part appearance = 10, and average number of detections of interest point for each image = 20. It is also important to point out that all parameters remain the same for learning all different categories. In other words, exactly the same piece of software was used in all experiments.

6.3 Walkthrough for the Motorbike Category

We now go through the experimental procedure step-by-step for the motorbike category. Six training images are selected (examples of which are shown in Fig. 3a). The Kadir and Brady interest operator is applied to them, giving $\mathcal{X}_t$. Each of these regions is then transformed into the fixed PCA basis, to give $\mathcal{A}_t$.

Next, we consider the prior we will use in learning. This has been constructed from models trained using ML from the three other data sets: spotted cats, faces, and airplanes. Ten ML models were trained for each category, giving a total of 30 models, each being a point in $\boldsymbol{\theta}$-space. The parameters of the prior, $\{\boldsymbol{m}_0, \beta_0, a_0, \boldsymbol{B}_0\}$ for both the shape and appearance components of the model are then directly computed from these points in the following manner:

- $\boldsymbol{m}_0$ is estimated by computing the mean of $\boldsymbol{\mu}^{ML}$ over the $M = 30$ ML models: $\boldsymbol{m}_0 = \frac{1}{M}\sum_m \boldsymbol{\mu}_m^{ML}$.
- a_0 is fixed to be number of degress of freedom in the precision matrix $\boldsymbol{\Gamma}^{ML}$, which differs between the shape and appearance terms. For shape, $a_0^{\mathcal{X}} = 2(P-1)(P-2)$, while $a_0^{\mathcal{A}} = kP$.
- $\boldsymbol{B}_0$ is estimated by letting $a_0\boldsymbol{B}_0^{-1}$, the mean of the precision, be $\frac{1}{M}\sum_m \boldsymbol{\Gamma}^{ML}$ and using the previously calculated value of a_0 to give $\boldsymbol{B}_0$.
- β_0 is estimated as the ratio between the precision of the mean and the mean of the precision: $\beta_0 = \frac{\|1/M\sum_m (\boldsymbol{\mu}_m^{ML} - \boldsymbol{m}_0)^2\|}{\|a_0\boldsymbol{B}_0^{-1}\|}$.

Fig. 4 illustrates both the ML models (as points colored by category) and the prior density fitted to them. Since the parameter space is high-dimensional, it is difficult to visualize. But, by considering each appearance descriptor separately, the mean and variance of the part from each model can be plotted in 2D. Note that all parts use the same prior density for appearance. For shape, the mean and variance of location of each part relative to the landmark part is shown. To understand how the prior assists in learning, models were trained on background data alone and their parameters also plotted in Fig. 4 (as magenta $*$'s). The prior density was estimated only from the ML category models, not these background models. However, they serve to illustrate the point that models lacking visual consistency occupy a different part of the parameter space to coherent models. The prior captures this knowledge, then, in the learning process, it biases $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t, \mathcal{O}_{fg})$ to areas of $\boldsymbol{\theta}$-space corresponding to visually consistent models.

Now that the prior and training data, $\mathcal{X}_t$ and $\mathcal{A}_t$, have been obtained, we commence the learning process described in Section 4. We only use one mixture component, so $\Omega = 1$. The initial values of the hyper-parameters $\{\lambda_\omega, a_\omega, \boldsymbol{B}_\omega, \boldsymbol{m}_\omega, \beta_\omega\}$ are initialized as in Fig. 3b. Note that,

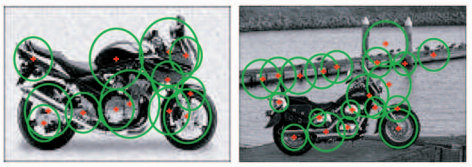
1. Available from www.vision.caltech.edu.

Fig. 2. The 101 object categories and the background clutter category. Each category contains between 45 and 400 images. Two randomly chosen samples are shown for each category. The categories were selected prior to the experiments and the images collected by operators not associated with the experiment. The last row shows examples from the background data set. This data set is obtained by collecting images through the Google image search engine (www.google.com). The keyword "things" is used to obtain the background data set. Note that only gray-scale information is used in our system. Complete data sets can be found at http://vision.caltech.edu/feifeili/101_ObjectCategories.

since we only have one component, we do not need to worry about setting λ.

The initial posterior densities are illustrated in green in Fig. 5. Then, we run the Bayesian One-Shot algorithm until convergence is reached. Fig. 5 shows the learned parameter densities in red. They can be seen to be much tighter then the initial density, often lying close to the prior density (in black), which is likely to exert a large influence with so few training images. The model corresponding to the mean of the parameter density is shown in Fig. 6.

In the recognition phase, the learned model is applied to 50 images containing motorbikes and 50 images of scenes not containing motorbikes. Fig. 6 shows the ROC curve for the model, along with sample images when the threshold, T, is set so as to give equal numbers of false alarms and missed detections.



(a)

Hyperparameter	Shape	Appearance
$\boldsymbol{M}$	$\frac{1}{I}\sum_i \frac{1}{\lvert H\rvert}\sum_h \mathcal{X}(h)$	$\frac{1}{I}\sum_i \frac{1}{\lvert H\rvert}\sum_h \mathcal{A}(h)$
β	β_0	β_0
a	$2(P-1)(P-2)$	kP
$\boldsymbol{B}$	$0.1\boldsymbol{I}_{2(P-1)}$	$0.1\boldsymbol{I}_{kP}$

(b)

Fig. 3. (a) Sample training images for the motorbike category, with the output of the feature detector overlaid. (b) Initial values of the hyperparameters of the parameter posterior for shape and appearance.

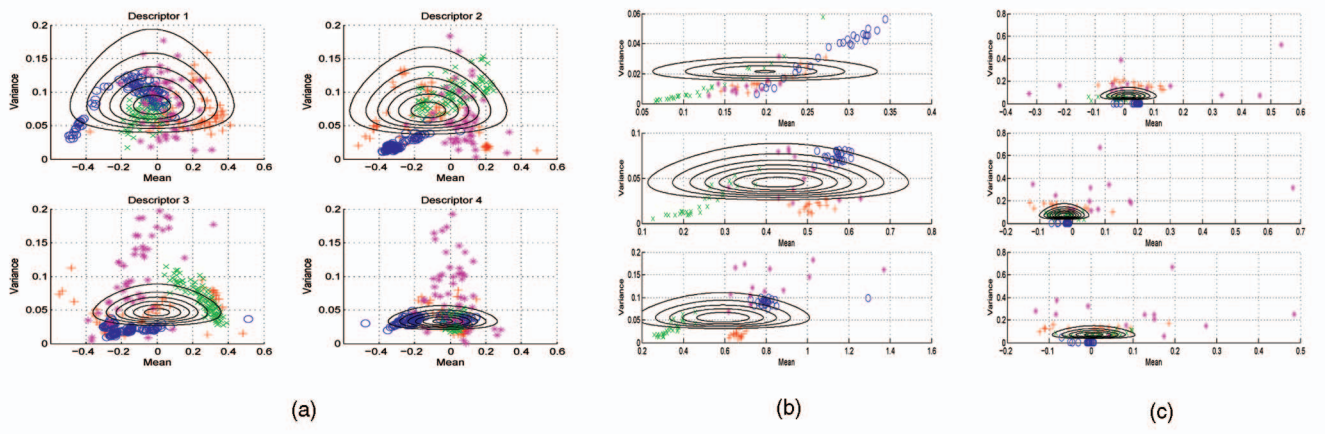


Fig. 4. A visualization of the prior parameter density, estimated from ML models of spotted cats (green $\times$s), faces (red $+$s), and airplanes (blue $\circ$s). Models trained on background data are shown as magenta $*$s, but are not used in estimating the prior density. In all figures, the mean is plotted on the x-axis and the variance on the y-axis. (a) Appearance parameter space for the first four descriptors. (b) X component of the shape term for each of the nonlandmark model parts. (c) Y component of shape. This figure is best viewed in color with magnification.

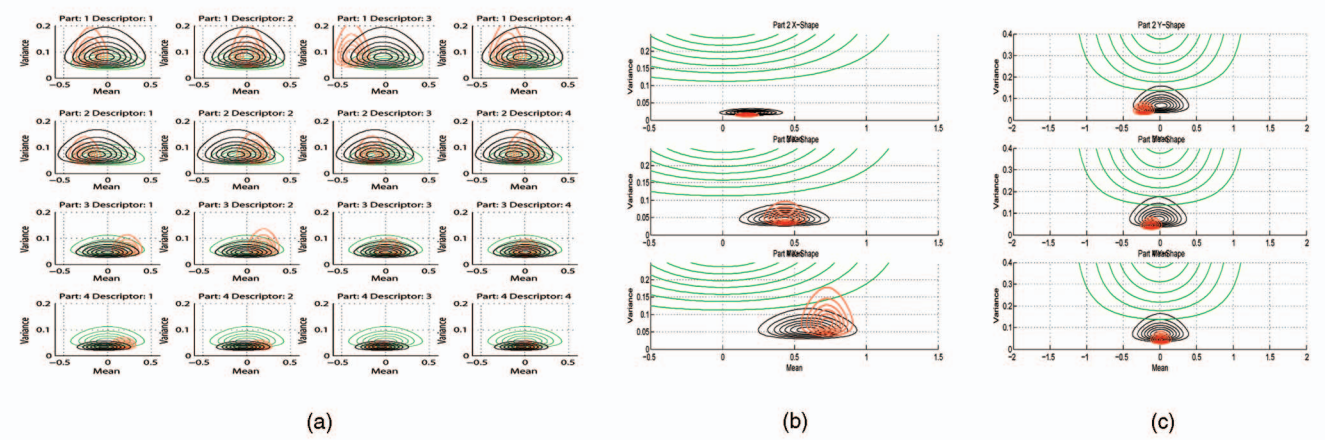


Fig. 5. The learning process. (a) Appearance parameter space, showing the mean and variance distributions for each of the models' four parts for the first four descriptors. The parameter densities are colored as follows: black for the prior, green for the initial posterior density, and red for the density after 30 iterations of Bayesian One-Shot, when convergence is reached. (b) X component of the shape term for each of the model parts. (c) Y component of shape. Note that, in both (b) and (c), only the variance terms along the diagonal are visualized—not the covariance terms. This figure is best viewed in color with magnification.

6.4 Caltech 4 Data Set

We first tested our algorithm on the four object categories used by Weber et al. [39] and Fergus et al. [13]. They are faces, motorbikes, airplanes, and spotted cats. Our experiments demonstrate the benefit of using prior information as well as using a full Bayesian algorithm in learning new object categories (Figs. 7 and 8). In Figs. 7 and 8, given zero training images, the detection rate for each category is at chance level 50 percent. This tells us that, given only the prior model, it is not sufficient to capture characteristic information of the particular categories we are interested in. Only by incorporating this prior knowledge into the training data is the algorithm capable of learning a sensible model with only one training example. For instance, in Fig. 7c, we see that the 4-part model has captured the essence of a face (e.g., eyes and nose). In this case, it achieves an average detection rate of 82 percent, given only one training example.

6.5 Caltech 101 Data Set

We have tested our algorithm on a large data set of 101 object categories (Fig. 2). We summarize different aspects of our experiments in the following sections.

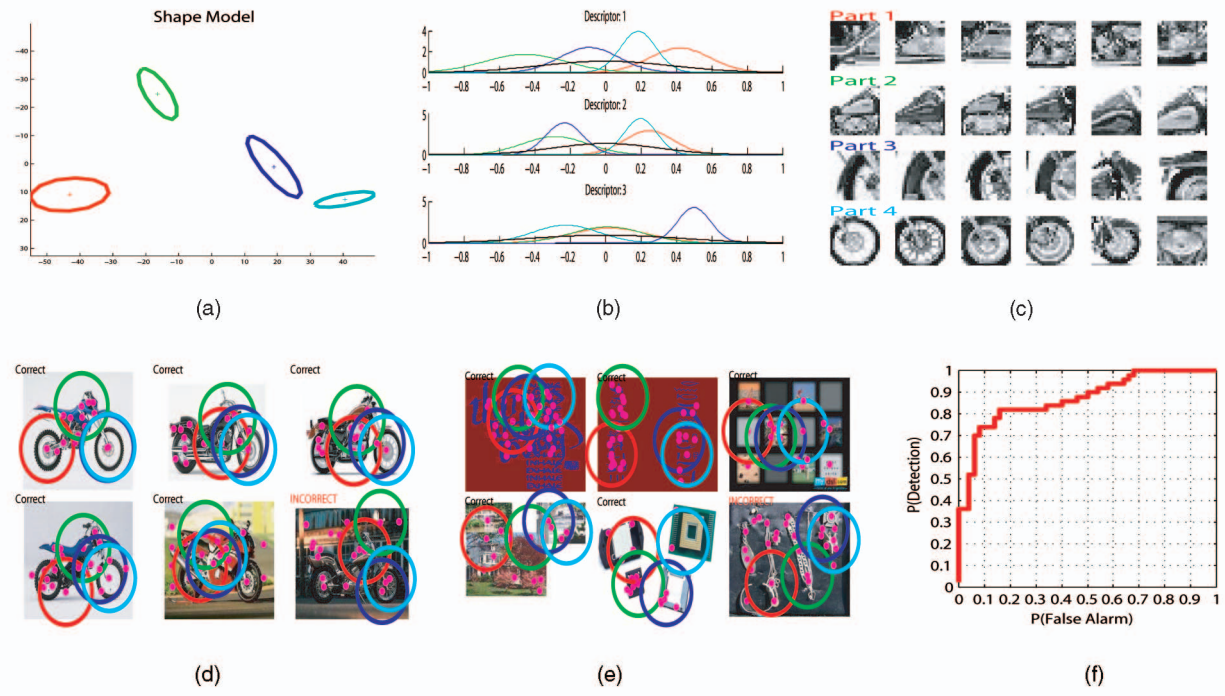


Fig. 6. The mean posterior model. (a) The shape component of the model. The four +s and ellipses indicate the mean and variance in position of each part. The interpart covariance terms are not shown. (b) The mean appearance distributions for the first three PCA dimensions. Each color indicates one of the four parts. The background density is shown in black. (c) The detected feature patches in the training image closest to the mean of the appearance densities for each of the four parts. (d) Some examples of foreground test images for the model, with a mix of correct and incorrect classifications. The pink dots are features found on each image and the colored circles indicate the best hypothesis in the image. The size of the circles indicates the score of the hypothesis (the bigger the better). (e) The model running on some background query images. (f) The ROC curve for the model on the test set. The equal error rate is around 18 percent.

6.5.1 Overall Results: ML versus MAP versus Bayesian

Using the Bayesian formulation, we are able to incorporate prior knowledge of the object world into the learning scheme. In addition, we are also capable of averaging over the uncertainties of models by integrating over the model distributions. Do both of these two factors contribute to the efficient learning of our algorithm? Or is it only the prior that truly matters?

We are able to answer this question by comparing the detection result of the Bayesian One-Shot algorithm not only to the ML method, but also to the MAP algorithm (as derived in [10]). Both the Bayesian One-Shot and the MAP algorithms are given exactly the same prior distributions for learning for each of the 101 categories. While Fig. 9 illustrates that prior knowledge helps in learning new object categories, the introduction of priors alone cannot account for all the advantages of our Bayesian formulation. The Bayesian algorithm consistently performed better than both the ML and MAP methods given a few training examples. While MAP learning takes advantage of the prior density, it is fundamentally the same as maximum likelihood in that a single parameter set is estimated for the object category. Given few training examples, such an assumption is likely to overfit the data points. The Bayesian algorithm reduces the overfit by averaging over model uncertainties.

6.5.2 Good Models and Bad Models

Figs. 10 and 11 show in detail the results from the grand-piano and cougar-face categories, both of which have achieved reasonable performances given few training examples (equal error rates of 84 percent and 85 percent, respectively, for 15 training examples). In the left-most columns, four examples of feature detection results are presented. The center of each detection circle indicates the location of the feature detected while the size of the circle indicates its scale. The second column shows the resulting shape model for the Bayesian One-Shot method for $\{1, 3, 6, 15\}$ training images. As the number of training examples increases, we observe that the shape model is more defined and structured with a reduction in variance. This is expected since the algorithm should be more and more confident of what is to be learned. The third column shows examples of the part appearance that are closest to the mean distribution of the appearance. Notice that distinctive features such as keyboards for the piano and eyes or whiskers for the cougar-face are successfully learned by the algorithm. Two learning methods' performances are compared in the top panel of the last column. The Bayesian methods clearly show a big advantage over the ML method when the training number is small.

It is also useful to look at the other end of the performance spectrum—those categories that have low recognition performance. We give some informal observations into the cause of the poor performance. Feature detection is a crucial step for both learning and recognition. On both the crocodile and mayfly figures in Fig. 12, notice that some testing images marked "INCORRECT" have few detection points on the target object itself. When feature detection fails either in learning or recognition, it affects the performance results greatly. Furthermore, Fig. 10a shows that a variety of viewpoints are present in each category. In this set of experiments, we have only used one mixture component, hence, only a single viewpoint can be accommodated. Our model is also a

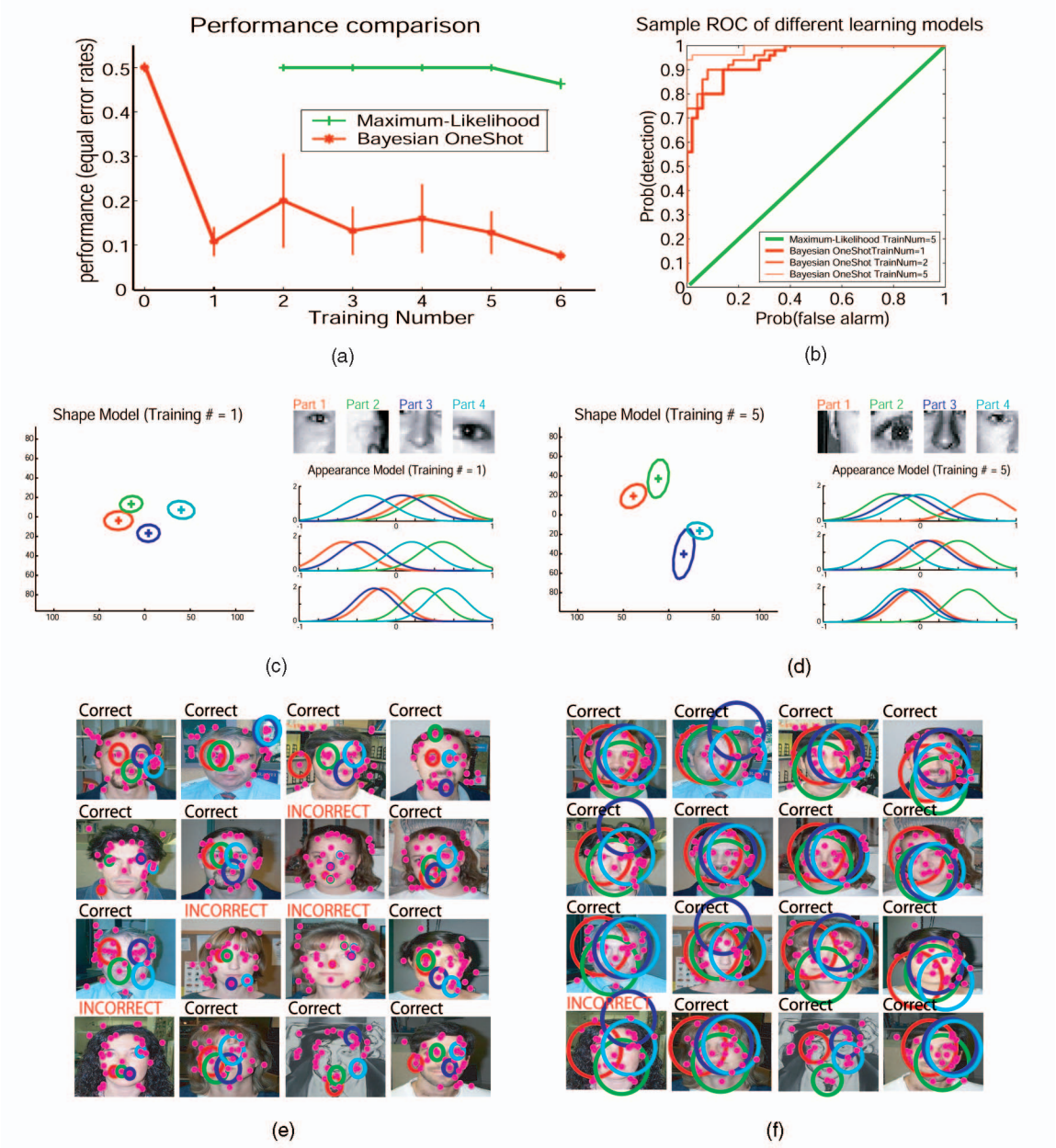


Fig. 7. Summary of face model. (a) Test performances of the algorithm given $0-6$ number of training image(s) (red line). Zero number of training images is when only the prior model is used. Note the prior alone is not sufficient for categorization. Each data point is obtained by 10 repeated runs with different randomly drawn training and testing images. Error bars show one standard deviation from the mean performance. This result is compared with the maximum-likelihood (ML) method (green). Note that ML cannot learn the degenerate case of a single training image. (b) Sample ROC curves for the Bayesian One-Shot algorithm (red) compared with the ML algorithm (green line). The curves shown here use typical models drawn from the repeated runs summarized in (a). (c), (d), (e), and (f) show typical models learned with one and five training images. (c) Shape model, appearance samples, and appearance densities (of the first three descriptors) for a model trained on one image. (e) Sample foreground test images for the model shown in (c). (d) and (f) correspond to a model trained on five images. Note that the size of the open circles on (e) and (f) indicates the strength of the hypothesis (scaled according to the log likelihood score), not the variance of the part locations. In other words, the bigger the circles, the stronger the algorithm believes in the recognition decision. Only the mean locations of the parts are shown here in pink dots. (c) Model from one training example. (d) Model from five training example. (e) Testing examples from model in (c). (f) Testing examples from model in (d).

simplified version Burl et al.'s constellation model [6], [39], [13] as it ignores the possibility of occluded parts.

6.5.3 A Further Investigation on Prior Models and Feature Detectors

One useful question to ask is whether learning is improved by constructing the prior model from more categories. To investigate this, we randomly select 20 object categories that will incrementally contribute to the prior model. We learn a model for each of the 20 categories, forming a set of models C. We also randomly select 30 object categories from the rest of the data set, calling this set S. We train a model for each category in S using a prior constructed from N models drawn from C. We vary N from 0 to 20. For $N = 0$, the prior model is a broad, noninformative distribution over the shape and appearance space. For $N > 0$, we pick a model from C and update the prior as a weighted average between the old prior model and the new category model, the weighting being $N - 1$

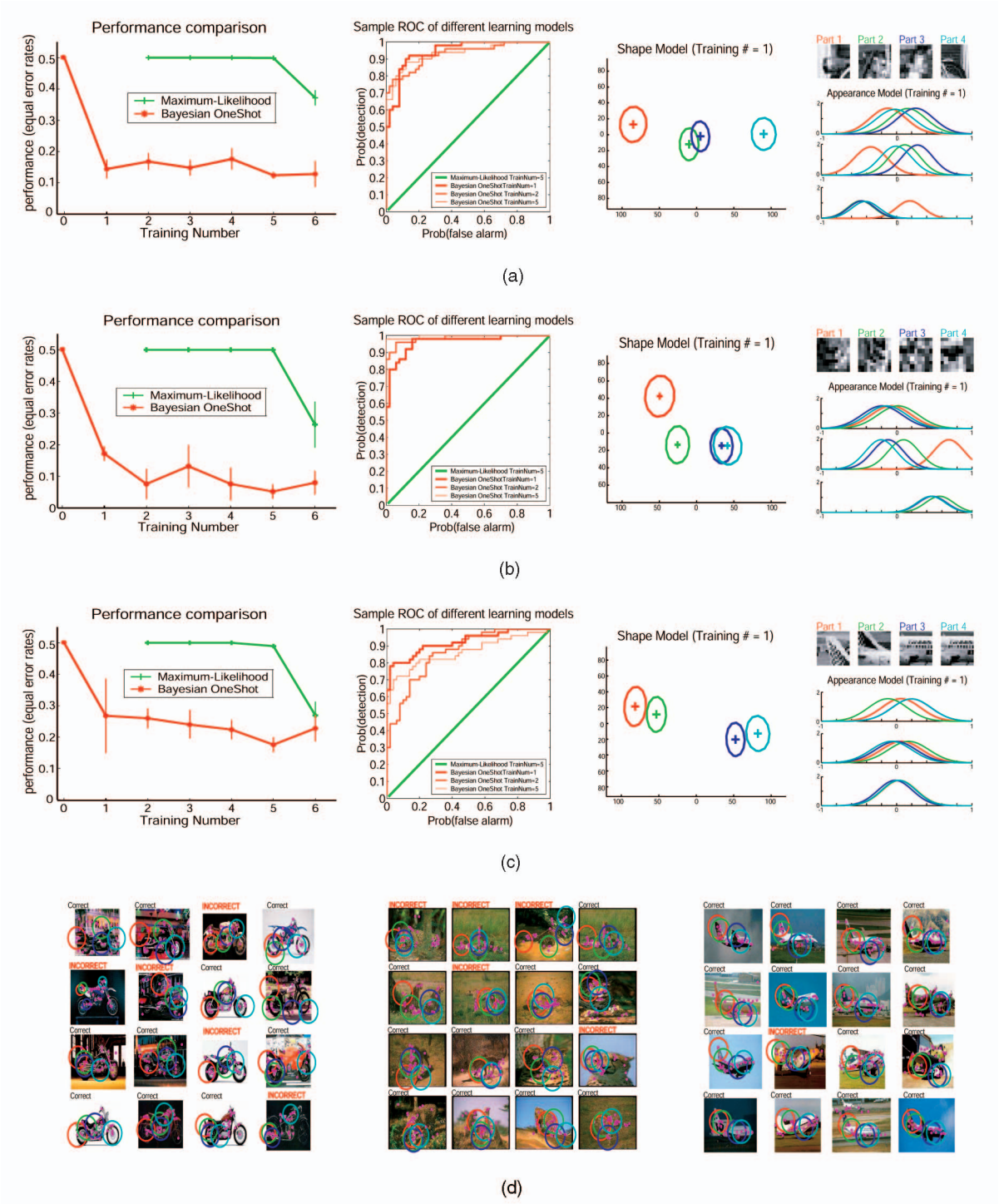


Fig. 8. Summary of the three other categories from the Caltech 4 data sets: motorbike, spotted cat, and airplane for one training example. Note that, in (b), the sample patches are of relatively low resolution. This is due to the lower resolution of the original images of the spotted cat category. (a) Summary of motorbike model. (b) Summary of spotted cat model. (c) Summary of airplane model. (d) Testing examples from models in (a), (b), and (c). respectively.

and 1, respectively. Fig. 13a shows the relationship between the number of categories contributing to the prior model and the performances averaged over all categories in S. We see a trend of decreasing error when the number of categories in the prior model is between one and eight, although this trend becomes less clear beyond eight.

We also explored the effect of feature detections on the overall object detection performances. Two human subjects annotated the whole data set, giving ground truth information of the location and the contours of the objects within each image. Given this information, we are able to compute the proportion of features detected within the object boundary as a fraction of the total number in the image. In Fig. 13b, we show the relationship between the quality of the feature detections and the performances for each training number. In general, a very weak positive correlation is observed between

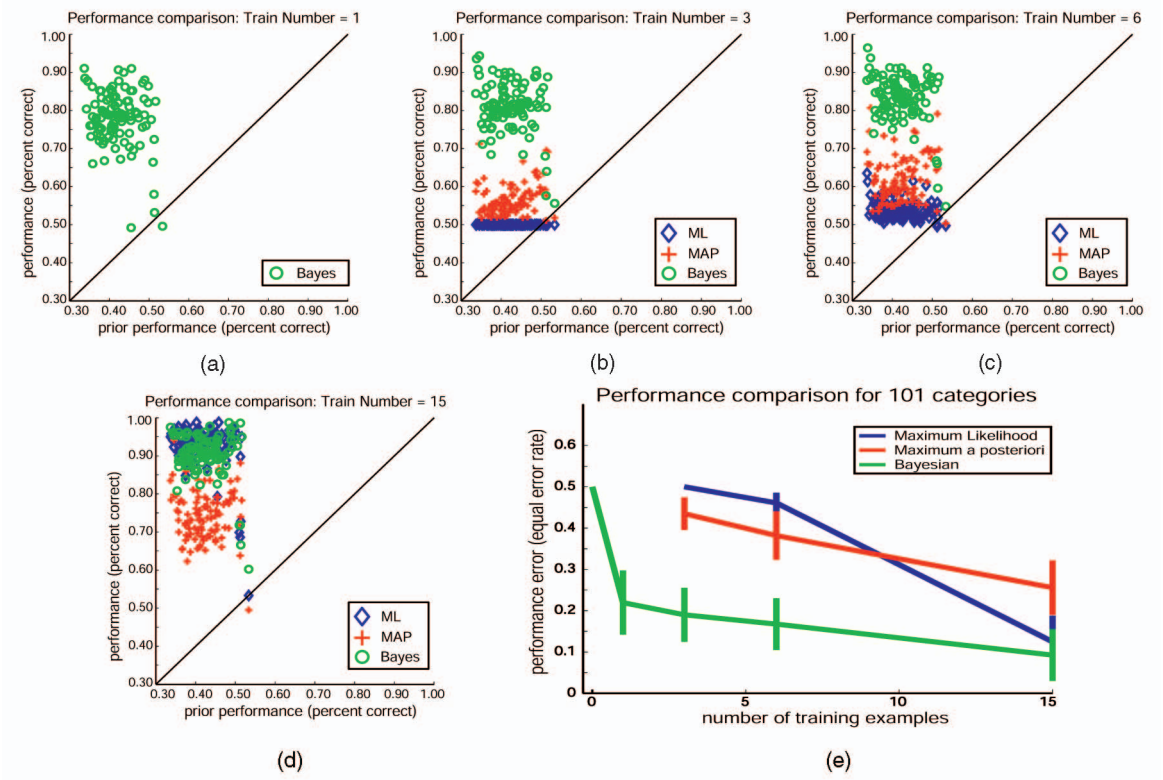


Fig. 9. Performance on 101 categories using three different learning methods: Maximum Likelihood (ML), Maximum A Posteriori (MAP), and the Bayesian One-Shot algorithm. (a), (b), (c), and (d) show the performance given training number(s) 1, 3, 6, and 15 and compare them with performance of the prior alone. "Percent correct" is measured as $1 - \text{Eq. Error Rate}$. (e) summarizes the four panels above, showing the mean performance (Eq. Error Rate). The error bars indicate one standard deviation.

feature detection quality and performance. This correlation seems to increase slightly as the training number increases.

6.5.4 Bayesian One-Shot Algorithm: Shape-Only versus App-Only versus Shape-App models

In Section 3.2, we detailed the formulation of object category models. Each model of an object category carries two sources of information: shape and appearance. We show in Fig. 14 that the contributions of shape and appearance components of the model vary when the object category to be learned differs. While some categories depend more on the shape component (e.g., faces, electrical guitars, side view of cars, etc.), others rely more on the appearance (leopards, octopi, ketch, etc.). Overall, a full model has a significant advantage over a shape-only or appearance-only model in terms of categorization performances.

6.5.5 Bayesian One-Shot Algorithm: Discrimination among 101 Categories

So far, we have tested our algorithm in a *detection* scenario: For a particular object category, we are only deciding if it is present or not. We now test the algorithm in a *discrimination* scenario: one where we have multiple categories (i.e., more than two) and must correctly classify the query images from each. In our experiment, we first learn a model for each of the 101 object categories. Query images are then drawn from the test set of each category in turn and evaluated by all 101 models. For a given image, the assignment of the category it belongs to is in the "winner-take-all" fashion. In other words, the category model that achieved the highest likelihood score is assigned to the image. For each category of images, we repeat the experiment 50 times with different randomly chosen training and test images. This gives a vector of 101 entries, each being the average of the "winner-take-all" assignment over the 50 repetitions. We do this for each of the 101 categories, so obtaining the confusion table in Fig. 15. By averaging the correct discrimination rates, i.e., the entries along the diagonal of Fig. 15a, we obtain the average correct discrimination rates for 3, 6, and 15 training examples of, respectively, 10.4, 13.9, and 17.7 percent. These rates would be approximately 1 percent if the classifiers were making random decisions.

6.5.6 Discussions

Our results highlight a number of issues that we continue to investigate. The most important one is the choice of priors. We have used a very general prior constructed from three categories and would like to further explore the effects of different priors. Notice that, in Fig. 9, the Maximum Likelihood method, on average, gives a similar level of performance to the Bayesian One-Shot algorithm for 15 training images. This is surprising, given the large number of parameters in each model and, therefore, a few hundred training examples are, in principle, required by a maximum likelihood method—one might have expected that the ML method to converge with the Bayesian One-Shot method at only around 100 training examples. The most likely reason for this result is that the prior that we employ is very simple. Similarly, this overly simple prior (along with other weaknesses of the model) might also be responsible for a lack of more dramatic improvement of the performances observed in Figs. 7 and 8. Bayesian methods live and die by the quality of

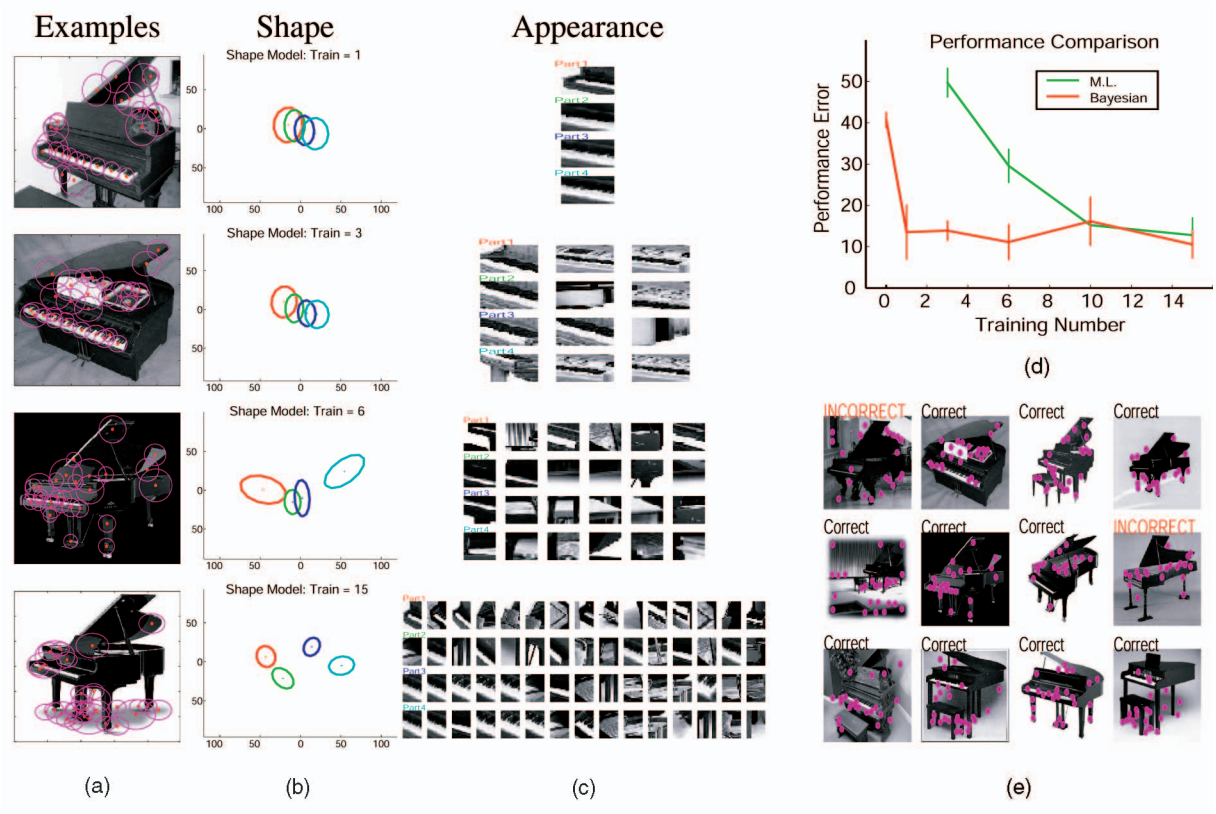


Fig. 10. Results for the "grand-piano" category. Column 1 shows examples of feature detection. Column 2 shows the shape models learned from $\{1, 3, 6, 15\}$ training images. Column 3 shows the appearance patches for the model learned from $\{1, 3, 6, 15\}$ training images. The top panel of Column 4 shows the comparative results between ML and Bayesian methods (the error bars show the variation over the 10 runs). The bottom panel of Column 4 shows the recognition result for the Bayesian One-Shot algorithm for one training image. Pink dots indicate the center of detected interest points.

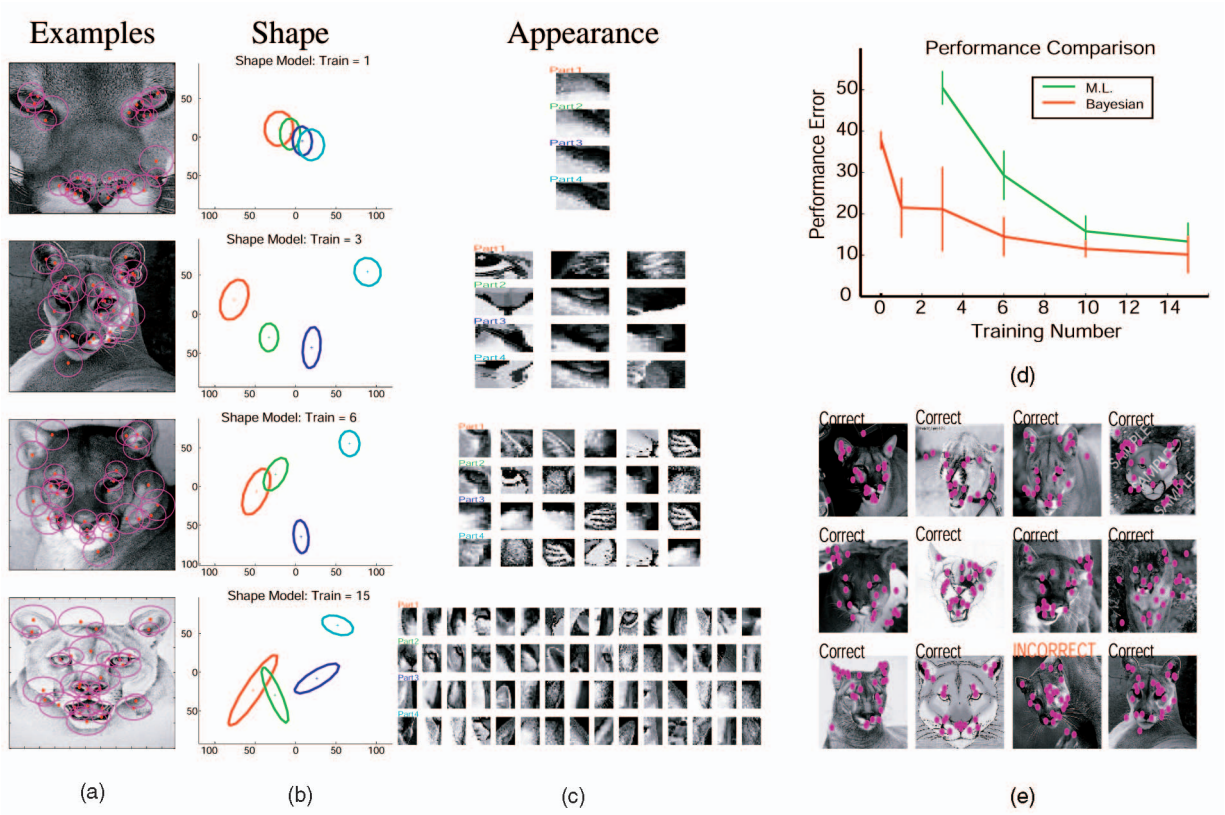


Fig. 11. Results for the "cougar face" category.

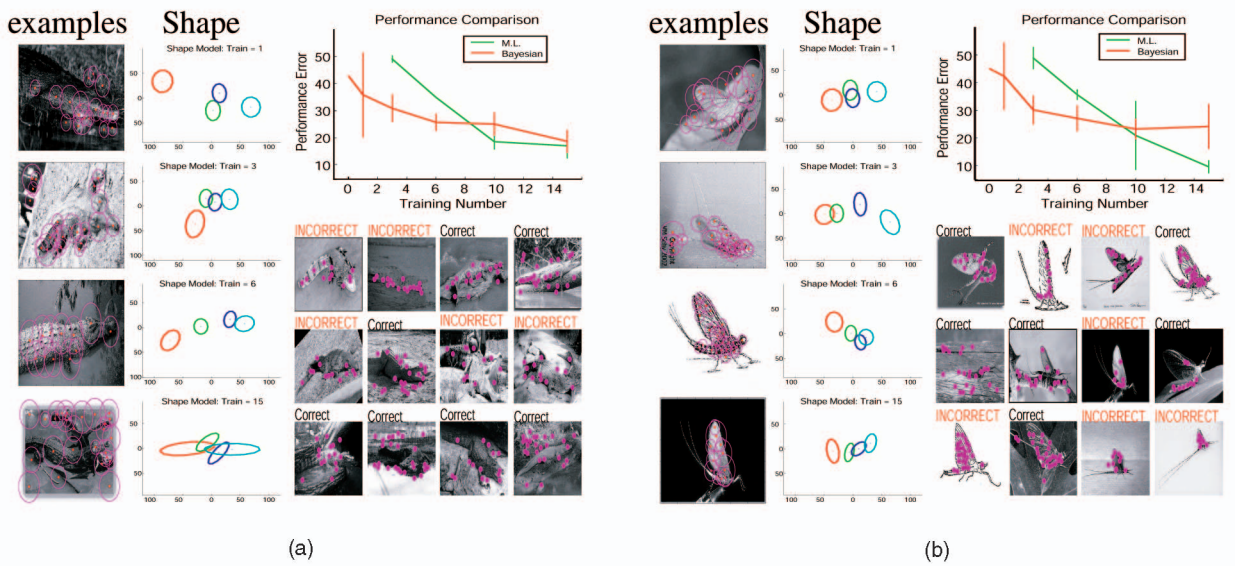


Fig. 12. Two categories with poor performance. (a) Crocodile (equal error rate = 35 percent for one training example). (b) Mayfly (equal error rate = 42 percent for one training example).

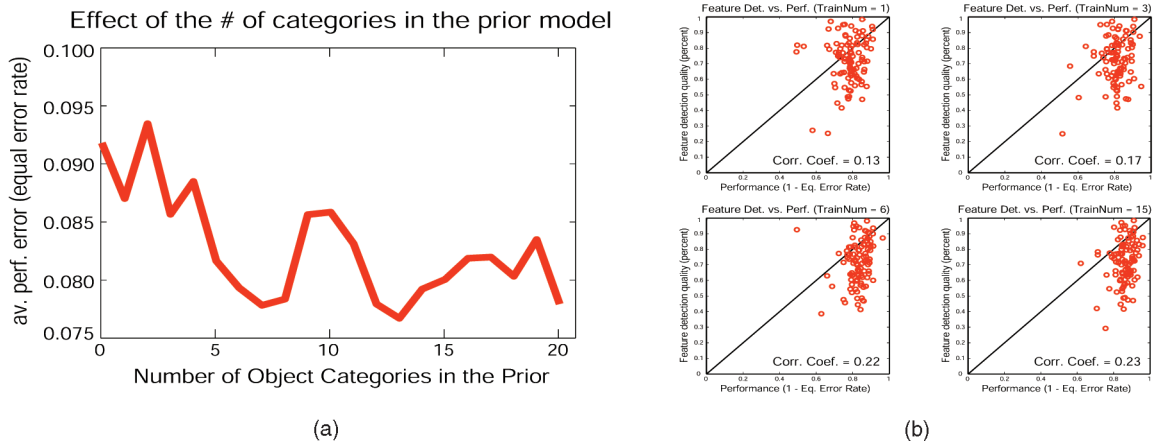


Fig. 13. (a) Effect of the number of object categories in the prior model on the performance of testing categories. There are 20 randomly drawn object categories for training the prior model. There are 30 other randomly drawn object categories in the testing category set. The x-axis indicates the number of object categories in the prior model. The y-axis indicates the average performance error of the 30 test categories given the prior model. (b) Quality of feature detection compared with object detection performances of the 101 categories given $\{1, 3, 6, 15\}$ training images. The x-axis of each plot is the detection performance of the model. The y-axis is the quality of feature detection, defined by the percentage of detection points landing within the outline of the object over the total number of detections. For each category, we average the percentage over all images within it.

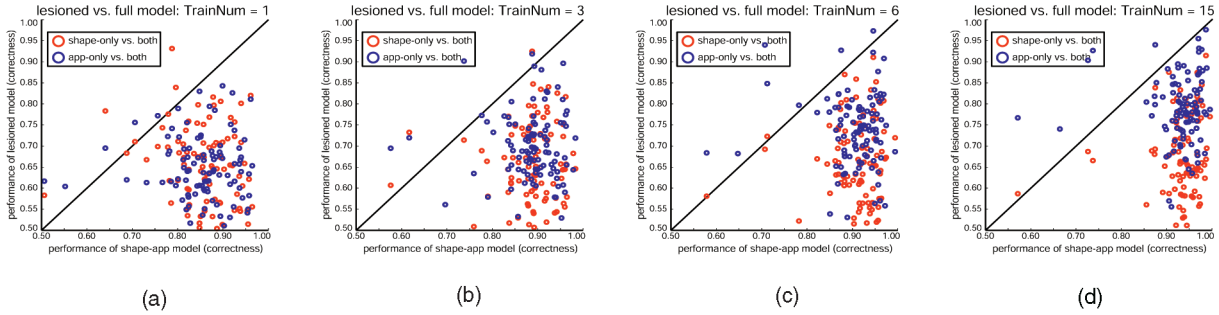


Fig. 14. Shape only models and appearance only models compared with models using shape and appearance for each of the 101 categories given $\{1, 3, 6, 15\}$ training images ((a), (b), (c), and (d)). The x-axis of each plot is the detection performance of models using both shape and appearance. The y-axis is the detection performance of shape-only models and appearance-only models for each category. (a) TrainNum = 1. (b) TrainNum = 3. (c) TrainNum = 6. (d) TrainNum = 15.

the prior that is used. Our prior density is derived from only three object categories. Given the variability of our training set, it is realistic that a prior based on many more categories would yield a better performance. We have tested this hypothesis using a simple, synthetic example in Figs. 15b and 15c. Our goal is to learn a simple triangular shape model

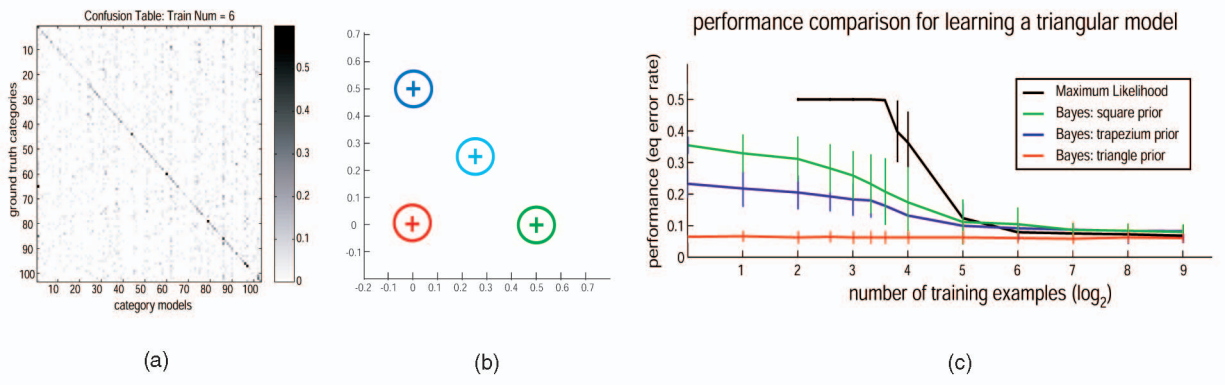


Fig. 15. (a) A confusion table for six training examples. The x-axis enumerates the category models, one for each category, giving 101 in total. The y-axis is the ground truth category for the query image. The intensity of an entry in the table corresponds to the probability of a given query image being classified as a given category. Since the categories are consistently ordered on both axes, the ideal case would consist of a completely black diagonal line, showing perfect discrimination power of all category models over all categories of objects. (b) The synthetic triangle model used in (c). Note the triangle is characterized by a 4-part model. (c) Effect of different priors for learning a triangle object category. Note that the point of convergence between the ML method and the Bayesian One-Shot method depends on the choice of prior distribution. When a prior is very effective (e.g., a triangular prior for learning a triangular model), it takes more than 100 training examples to converge. But, when the prior is not very effective (e.g., square or trapezium priors for learning a triangular model), it takes less than 30 training examples for the two methods to converge.

TABLE 1
A Comparison between a Variety of Object Recognition Approaches

Authors	*Categories*	*# Categories*	*# Training images*	*Framework*	*Hand alignment*	*Segmented*
Fei-Fei *et al.*	Assorted	101	1-5	Gen.	N	N
Fergus *et al.* [14]	Assorted	6	> 100	Gen.	N	N
Weber *et al.* [40]	Cars, Faces	2	> 100	Gen. + Disc.	N	N
Viola & Jones [37]	Faces	1	$\sim$ 10,000	Disc.	Y	Y
Schneiderman & Kanade [35]	Cars	1	2,000	Disc.	Y	N
Rowley *et al.* [33]	Cars	1	500	Disc.	Y	N
Amit *et al.* [2]	Faces, Characters	3	300	Gen.	Y	Y
LeCun *et al.* [25]	Digits	10	60,000	Disc.	N	Y
LeCun *et al.* [26]	Assorted	5	$\sim$300,000	Disc.	Y	N

The framework column specifies if the approach is generative (Gen.) or discriminative (Disc.) or both. The hand alignment and segmented columns indicate if the training data needs to be hand-aligned or hand-segmented for a given approach.

(Fig. 15b). We test the effect of priors on the Bayesian One-Shot algorithm by giving the system three different priors: a triangular shape prior (similar to the synthetic model in Fig. 15b used to generate the data. Note that a fourth part of the triangle model is located off the vertex), a trapezium shape prior, and a square shape prior. The Bayesian One-Shot algorithm with three different priors is compared to the maximum likelihood method. We observe that it takes more than 100 training examples for the ML method to "catch up" with the Bayesian One-Shot learning method given the triangular shape prior. On the contrary, it takes much smaller number of training examples for the ML method to converge with the other two Bayesian One-Shot learning method with noneffective priors.

7 Conclusions and Future Work

We have demonstrated that, contrary to intuition, useful aspects of a new object category may be learned from a single training example (or just a few). As Table 1 shows, this is beyond the capability of existing algorithms.

The key insight we have exploited is that categories we have already learned give us information that helps us to learn new categories with fewer training examples. To pursue this idea, we developed a Bayesian learning framework based on representing object categories with probabilistic models. Prior information from previously learned categories is represented with a suitable prior probability density function on the parameters of their models. These prior models are updated with the few training examples available to produce posteriors which, in turn, may be used for both detection and discrimination.

Our experiments, conducted on images from 101 categories, are encouraging in that they show that very few (1 to 5) training examples produce models that are already able to achieve a detection performance of around 70-95 percent. Furthermore, that the categories from which the prior knowledge is learned do not need to be visually similar to the categories that one wishes to learn.

While our experiments are very encouraging, they are by no means satisfactory from a practical standpoint. Much can be done toward the goal of obtaining better error rates, as our current implementation is, at the moment, just a toy. In order to curtail the complexity of our experiments, we have simplified the probabilistic models that are used for representing objects. For example, a probabilistic model for occlusion ([39], [6], [13]) was not implemented, and we only used four parts in our models, definitely not enough to represent the full complexity of object appearance. Furthermore, we only used three known categories to derive a prior. This is clearly a very small set which ought to be substantially broadened in a real-world situation.

However, at this point, it is probably more important to make progress at the conceptual level and much still needs to be done. For example, would a more sophisticated, multimodal prior be beneficial in learning? Is it easier to learn new categories which are similar to some of the "prior" categories? How should one best represent prior knowledge? Is there any other productive point of view, besides the Bayesian one which we have adopted here, that allows one to incorporate prior knowledge? In addition, it would be highly valuable to learn incrementally [29] where each training example will update the probability density function defined on the parameters of each object category; we presented a few ideas toward this in [9].

One last note of optimism: We feel that the problem of recognizing automatically hundreds, perhaps thousands, of object categories does not belong to a hopelessly far future. We hope that the positive outcome of our experiments on the large majority of 101 very diverse and challenging categories, despite the simplicity of our implementation and the rudimentary prior we employ, will encourage other vision researchers to test their algorithms on larger and more diverse data sets [43].

ACKNOWLEDGMENTS

The authors would like to thank Andrew Zisserman, David Mackay, Brian Ripley, and Joel Lindop. This work was supported by the Caltech CNSE, the UK EPSRC, and EC Project CogViSys.

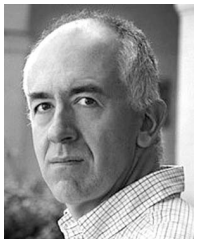
REFERENCES

[1] *Merriam-Webster's Collegiate Dictionary,* 10th ed., Springfield, Mass.: Merriam-Webster, Inc., 1994.
[2] Y. Amit and D. Geman, "A Computational Model for Visual Selection," *Neural Computation,* vol. 11, no. 7, pp. 1691-1715, 1999.
[3] H. Attias, "Inferring Parameters and Structure of Latent Variable Models by Variational Bayes," *Proc. 15th Conf. Uncertainty in Artificial Intelligence,* pp. 21-30, 1999.
[4] I. Biederman, "Recognition-by-Components: A Theory of Human Image Understanding," *Psychological Rev.,* vol. 94, pp. 115-147, 1987.
[5] M. Burl and P. Perona, "Recognition of Planar Object Classes," *Proc. Conf. Computer Vision and Pattern Recognition,* pp. 223-230, 1996.
[6] M. Burl, M. Weber, and P. Perona, "A Probabilistic Approach to Object Recognition Using Local Photometry and Global Geometry," *Proc. European Conf. Computer Vision,* pp. 628-641, 1996.
[7] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," *J. Royal Statistical Soc.,* vol. 29, pp. 1-38, 1976.
[8] L. Fei-Fei, R. Fergus, and P. Perona, "A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories," *Proc. Ninth Int'l Conf. Computer Vision,* pp. 1134-1141, Oct. 2003.
[9] L. Fei-Fei, R. Fergus, and P. Perona, "Learning Generative Visual Models from Few Training Examples: An Incremental Bayesian Approach Tested on 101 Object Categories," *Proc. Workshop Generative-Model Based Vision,* 2004.
[10] L. Fei-Fei, R. Fergus, and P. Perona, supplemental material, http://computer.org/tpami/archives.htm, 2006.
[11] P. Felzenszwalb and D. Huttenlocher, "Pictorial Structures for Object Recognition," *Int'l J. Computer Vision,* vol. 1, pp. 55-79, 2005.
[12] P. Felzenszwalb and D. Huttenlocher, "Representation and Detection of Deformable Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 2, pp. 208-220, Feb. 2005.
[13] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *Proc. Computer Vision and Pattern Recognition,* pp. 264-271, 2003.
[14] R. Fergus, P. Perona, and A. Zisserman, "A Visual Category Filter for Google Images," *Proc. Eighth European Conf. Computer Vision,* 2004.
[15] R. Fergus, P. Perona, and A. Zisserman, "A Sparse Object Category Model for Efficient Learning and Exhaustive Recognition," *Proc. Computer Vision and Pattern Recognition,* 2005.
[16] D. Forsyth and A. Zisserman, "Shape from Shading in the Light of Mutual Illumination," *Image and Vision Computing,* pp. 42-29, 1990.
[17] A. Gelman, J.B. Carlin, H.S. Stern, and D.B. Rubin, *Bayesian Data Analysis.* Chapman Hall/CRC, 1995.
[18] R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice.* Chapman Hall, 1992.
[19] R. Gilks and P. Wild, "Adaptive Rejection Sampling for Gibbs Sampling," *Applied Statistics,* vol. 41, pp. 337-348, 1992.
[20] W. Grimson and D. Huttenlocher, "On the Sensitivity of the Hough Transform for Object Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 12, no. 3, pp. 255-274, Mar. 1990.
[21] K. Humphreys and M. Titterington, "Some Examples of Recursive Variational Approximations for Bayesian Inference," *Advanced Mean Field Methods.* M. Opper and D. Saad, eds., MIT Press, 2001.
[22] D. Huttenlocher, G. Klanderman, and W. Rucklidge, "Comparing Images Using the Hausdorff Distance," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 850-863, Sept. 1993.
[23] T. Kadir and M. Brady, "Scale, Saliency and Image Description," *Int'l J. Computer Vision,* vol. 45, no. 2, pp. 83-105, 2001.
[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-Based Learning Applied to Document Recognition," *Proc. IEEE,* vol. 86, no. 11, pp. 2278-2324, Nov. 1998.
[25] Y. LeCun, F. Huang, and L. Bottou, "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting," *Proc. Conf. Computer Vision and Pattern Recognition,* 2004.
[26] T. Leung, M. Burl, and P. Perona, "Finding Faces in Cluttered Scenes Using Labeled Random Graph Matching," *Proc. Int'l Conf. Computer Vision,* pp. 637-644, 1995.
[27] D. Lowe, "Object Recognition from Local Scale-Invariant Features," *Proc. Int'l Conf. Computer Vision,* pp. 1150-1157, 1999.
[28] K. Mikolajczyk and C. Schmid, "An Affine Invariant Interest Point Detector," *Proc. European Conf. Computer Vision,* vol. 1, pp. 128-142, 2002.
[29] R. Neal and G. Hinton, "A View of the EM Algorithm that Justifies Incremental, Sparse and Other Variants," *Learning in Graphical Models,* M.I. Jordan, ed., pp. 355-368, Norwell, Mass.: Kluwer Academic Press, 1998.
[30] W. Penny, "Variational Bayes for d-Dimensional Gaussian Mixture Models," technical report, Univ. College London, 2001.
[31] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce, "3D Object Modeling and Recognition Using Affine-Invariant Patches and Multi-View Spatial Constraints," *Proc. Computer Vision and Pattern Recognition,* pp. 272-280, 2003.
[32] H. Rowley, S. Baluja, and T. Kanade, "Neural Network-Based Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, pp. 23-38, Jan. 1998.
[33] E. Sali and S. Ullman, "Combining Class-Specific Fragments for Object Classification," *Proc. British Machine Vision Conf.,* vol. 1, pp. 203-213, 1999.
[34] H. Schneiderman and T. Kanade, "A Statistical Approach to 3D Object Detection Applied to Faces and Cars," *Proc. Computer Vision and Pattern Recognition,* pp. 746-751, 2000.
[35] K. Sung and T. Poggio, "Example-Based Learning for View-Based Human Face Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, pp. 39-51, Jan. 1998.
[36] P. Viola and M. Jones, "Rapid Object Detection Using a Boosted Cascade of Simple Features," *Proc. Computer Vision and Pattern Recognition,* vol. 1, pp. 511-518, 2001.
[37] P. Viola, M. Jones, and D. Snow, "Detecting Pedestrians Using Patterns of Motion and Appearance," *Proc. Int'l Conf. Computer Vision,* pp. 734-741, 2003.

[38] M. Weber, W. Einhaeuser, M. Welling, and P. Perona, "Viewpoint-Invariant Learning and Detection of Human Heads," *Proc. Fourth Int'l Conf. Automated Face and Gesture Recognition,* pp. 20-27, 2000.
[39] M. Weber, M. Welling, and P. Perona, "Unsupervised Learning of Models for Recognition," *Proc. European Conf. Computer Vision,* vol. 2, pp. 101-108, 2000.
[40] A. Torralba, K.P. Murphy, and W.T. Freeman, Sharing Features: Efficient Boosting Procedures for Multiclass Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 762-769, 2004.
[41] M. Weber, "Unsupervised Learning of Models for Object Recognition," PhD thesis, Calif. Inst. of Technology, Pasadena, 2000.
[42] R. Fergus, "Visual Object Category Recognition," PhD thesis, Univ. of Oxford, U.K., 2005.
[43] A. Berg, T. Berg, and J. Malik, "Shape Matching and Object Recognition Using Low Distortion Correspondence," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 26-33, June 2005.

Li Fei-Fei graduated from Princeton University in 1999 with a physics degree. She received the PhD degree in electrical engineering from the California Institute of Technology in 2005. In the spring of 2005, she was a visiting scientist at the Microsoft Research Center in Cambridge, United Kingdom. She became an assistant professor of electrical and computer engineering at the University of Illinois Urbana-Champaign (UIUC) in July 2005. She is now a full-time member of the faculty at the Beckman Institute of UIUC. Her main research interest is vision, particularly high-level visual recognition. In computer vision, she has worked on both object and natural scene recognition. In human vision, she has studied the interaction of attention and natural scene and object recognition. She is a member of the IEEE.

Rob Fergus received the MEng degree from the University of Cambridge in 2000 and the MSc degree from the California Institute of Technology in 2002. He is currently a PhD candidate in the Visual Geometry Group at the University of Oxford. His research interests include object recognition, efficient algorithms for computer vision, and machine learning. He was the recipient of the best paper prize at the IEEE CVPR 2003 conference. He is a student member of the IEEE.



Pietro Perona graduated in electrical engineering from the Universita di Padova in 1985. He received the PhD degree from the University of California at Berkeley in 1990. He was a postdoctoral fellow at the Laboratory for Information and Decision Systems at MIT in 1990-1991 and became an assistant professor of electrical engineering at the California Institute of Technology in 1991. In 1996, he became professor of electrical engineering and of computation and neural systems. Since 1999, he has been the director of the National Science Foundation Engineering Research Center in Neuromorphic Systems Engineering at Caltech. He has served on the editorial board of the *International Journal of Machine Vision*, the *Journal of Machine Learning Research*, *Vision Research*, and as co-general chair of the IEEE Conference of Computer Vision and Pattern Recognition (CVPR 2003). He is interested in the computational aspects of vision; his current research focus is visual recognition. He has worked on PDEs for image analysis and segmentation (anisotropic diffusion), multiresolution-multi-orientation filtering for early vision, human texture perception and segmentation, dynamic vision, grouping, detection and analysis of human motion, human perception of 3D shape from shading, learning and recognition of object categories, human categorization of scenes, interaction of attention, and recognition. He is a member of the IEEE.