A Bayesian Approach to Unsupervised One-Shot Learning of Object Categories

Li Fei-Fei[1] Rob Fergus[2] Pietro Perona[1]

[1] Dept. of Electrical Engineering,
California Institute of Technology,
MC 136-93, Pasadena,
CA 91125, U.S.A.
{feifeili, perona}@vision.caltech.edu

[2] Dept. of Engineering Science,
University of Oxford,
Parks Road, Oxford,
OX1 3PJ, U.K.
fergus@robots.ox.ac.uk

Abstract

Learning visual models of object categories notoriously requires thousands of training examples; this is due to the diversity and richness of object appearance which requires models containing hundreds of parameters. We present a method for learning object categories from just a few images ($1 \sim 5$). It is based on incorporating "generic" knowledge which may be obtained from previously learnt models of unrelated categories. We operate in a variational Bayesian framework: object categories are represented by probabilistic models, and "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. Our ideas are demonstrated on four diverse categories (human faces, airplanes, motorcycles, spotted cats). Initially three categories are learnt from hundreds of training examples, and a "prior" is estimated from these. Then the model of the fourth category is learnt from 1 to 5 training examples, and is used for detecting new exemplars a set of test images.

1. Introduction

It is believed that humans can recognize between $5,000$ and $30,000$ object categories [1]. Informal observation tells us that learning a new category is both fast and easy, sometimes requiring very few training examples: given 2 or 3 images of an animal you have never seen before, you can usually recognize it reliably later on. This is to be contrasted with the state of the art in computer vision, where learning a new category typically requires thousands, if not tens of thousands, of training images. These have to be collected, and sometimes manually segmented and aligned [2, 3, 4, 5, 6, 7] – a tedious and expensive task.

Computer vision researchers are neither being lazy nor unreasonable. The appearance of objects is diverse and complex. Models that are able to represent categories as diverse as frogs, skateboards, cell-phones, shoes and mushrooms need to incorporate hundreds, if not thousands of parameters. A well-known rule-of-thumb says that the number of training examples has to be 5 to 10 times the number of object parameters – hence the large training sets. The penalty for using small training sets is over fitting: while in-sample performance may be excellent, generalization to new examples is terrible. As a consequence, current systems are impractical where real-time user interaction is required, e.g. searching an image database. By contrast, such ability is clearly demonstrated in learning in humans. Does the human visual system violate what would appear to be a fundamental limit of learning? Could computer vision algorithms be similarly efficient? One possible explanation of human efficiency is that when learning a new category we take advantage of prior experience. While we may not have seen ocelots before, we have seen cats, dogs, chairs, grand pianos and bicycles. The appearance of the categories we know and, more importantly, the *variability* in their appearance, gives us important information on what to expect in a new category. This may allow us to learn new categories from few(er) training examples.

We explore this hypothesis in a Bayesian framework. Bayesian methods allow us to incorporate prior information about objects into a "prior" probability density function which is updated, when observations become available, into a "posterior" to be used for recognition. Bayesian methods are not new to computer vision [8]; however, they have not been applied to the task of learning models of object categories. We use here "constellation" probabilistic models of object categories, as developed by Burl *et al.* [9] and improved by Weber et al. [6] and Fergus et al. [10]. While they maximized model likelihood to learn new categories, we use variational Bayesian methods by incorporating "general" knowledge of object categories [11, 12, 14]. We show that our algorithm is able to learn a new, unrelated category using one or a few training examples.

In Section 2 we outline the theoretical framework of

recognition and learning. In Section 3 we introduce in detail the method used in our experiments. In Section 4 we discuss experimental results on four real-world categories of different objects (Figure 1).

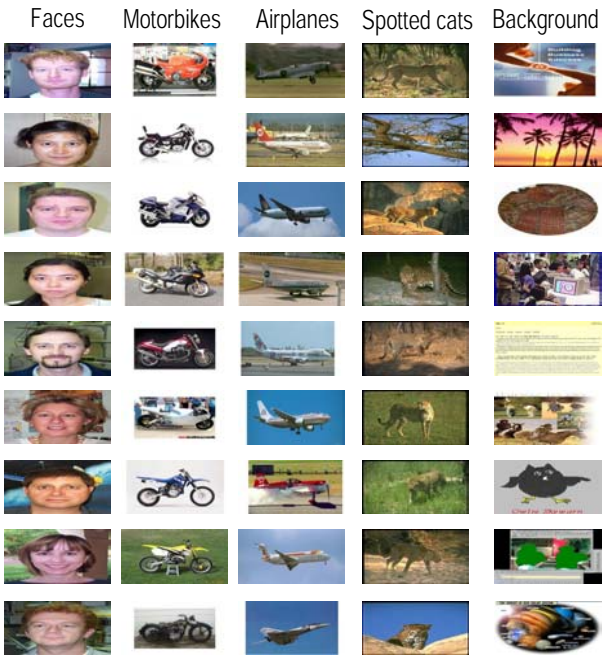


Figure 1: Some sample images from our datasets. The first four columns contain example images from the four object categories used in our experiments. The fifth column shows examples from the background dataset. This dataset is obtained by collecting images through the Google image search engine (www.google.com). The keyword "things" is used to obtain hundreds of random images. Note only grayscale information is used in our system. Complete datasets can be found at http://vision.caltech.edu/datasets.

2. Approach

Our goal is to learn a model for a new object class with very few training examples (in the experiments we use just $1-5$) in an unsupervised manner. We postulate that this may be done if we can exploit information obtained from previously learnt classes. Three questions have to be addressed in order to pursue this idea: How do we represent a category? How do we represent "general" information coming from the categories we have learnt? How do we incorporate a few observations in order to obtain a new representation? In this section we address these questions in a Bayesian setting. Note that we will refer to our current algorithm the Bayesian One-Shot algorithm.

2.1. Recognition

The model is best explained by first considering recognition. We will introduce a generative model of the object category built upon the constellation model for object representation [6, 9, 10].

2.1.1 Bayesian framework

We start with a learnt object class model and its corresponding model distribution $p(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a set of model parameters for the distribution. We are then presented with a new image and we must decide if it contains an instance of our object class or not. In this query image we have identified N interesting features with locations $\mathcal{X}$, and appearances $\mathcal{A}$. We now make a Bayesian decision, R. For clarity, we explicitly express training images through the detected feature locations $\mathcal{X}_t$ and appearances $\mathcal{A}_t$.

$$R = \frac{p(\text{Object}|\mathcal{X},\mathcal{A},\mathcal{X}_t,\mathcal{A}_t)}{p(\text{No Object}|\mathcal{X},\mathcal{A},\mathcal{X}_t,\mathcal{A}_t)} \quad (1)$$

$$= \frac{p(\mathcal{X},\mathcal{A}|\mathcal{X}_t,\mathcal{A}_t,\text{Object})\ p(\text{Object})}{p(\mathcal{X},\mathcal{A}|\mathcal{X}_t,\mathcal{A}_t,\text{No object})\ p(\text{No object})} \quad (2)$$

$$\approx \frac{\int p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta},\text{Object})p(\boldsymbol{\theta}|\mathcal{X}_t,\mathcal{A}_t,\text{Object})\ d\boldsymbol{\theta}}{\int p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta}_{bg},\text{No Object})p(\boldsymbol{\theta}_{bg}|\mathcal{X}_t,\mathcal{A}_t,\text{No Object})\ d\boldsymbol{\theta}_{bg}} \quad (3)$$

Note the ratio of $\frac{p(\text{Object})}{p(\text{No Object})}$ in Eq.2 is usually set manually to 1, hence omitted in Eq.3. Evaluating the integrals in Eq.3 analytically is typically impossible. Various approximations can be made to simplify the task. The simplest one, Maximum Likelihood (ML) assumes that $p(\boldsymbol{\theta})$ is a delta-function centered at $\boldsymbol{\theta} = \boldsymbol{\theta}_{ML}$ [6, 10]. This allows the integral to collapse to $p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta}_{ML})$. The ML approach is clearly a crude approximation to the integral. It assumes a well peaked $p(\boldsymbol{\theta}|\mathcal{X}_t,\mathcal{A}_t)$ so that $\boldsymbol{\theta}_{ML}$ is a suitable estimation of the entire distribution. Such an assumption relies heavily on sufficient statistics of data to give rise to such a sensible distribution. But in the limit of just a few training examples, ML is most likely to be ill-fated.

At the other extreme, we can use numerical methods such as Markov-Chain Monte-Carlo (MCMC) to give an accurate estimate, 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.

An alternative is to make approximations to the integrand until the integral becomes tractable. One method of doing this is to use a variational bound [11, 12, 13, 14]. We show in Sec.2.2 how these methods may be applied to our model.

2.1.2 Object representation: The constellation model

Now we need to examine the details of the model $p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta})$. We represent object categories with a constellation model [6, 10]

$$p(\mathcal{X},\mathcal{A}|\boldsymbol{\theta}) = \sum_{\mathbf{h}\in H} p(\mathcal{X},\mathcal{A},\mathbf{h}|\boldsymbol{\theta}) = \sum_{\mathbf{h}\in H} \underbrace{p(\mathcal{A}|\mathcal{X},\mathbf{h},\boldsymbol{\theta})}_{Appearance}\underbrace{p(\mathcal{X}|\mathbf{h},\boldsymbol{\theta})}_{Shape} \quad (4)$$

where $\boldsymbol{\theta}$ is a vector of model parameters. Since our model only has P (typically 3-7) parts but there are N (up to 100) features in the image, we 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. The set of all hypotheses H consists all valid allocations of features to the parts; consequently $|H^n|$, the total number of hypotheses in image n is $O(N^P)$.

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). Note, that in the model the following assumptions are made: shape is independent of appearance; for shape the joint covariance of the parts' position is modeled, whilst for appearance each part is modeled independently. In the experiments reported here we use a slightly simplified version of the model presented in [10] 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.

Appearance. Each feature's appearance is represented as a point in some appearance space, defined below. Each part p has a Gaussian density within this space, with mean and precision parameters $\boldsymbol{\theta}_p^{\mathcal{A}} = \{\boldsymbol{\mu}_p^{\mathcal{A}}, \boldsymbol{\Gamma}_p^{\mathcal{A}}\}$ which is independent of other parts' densities. Each feature selected by the hypothesis is evaluated under the appropriate part density. The distribution becomes $p(\mathcal{A}|\mathcal{X}, \mathbf{h}, \boldsymbol{\theta}) = \prod_{p=1}^{P} \mathcal{G}(\mathcal{A}(\boldsymbol{h}_p)|\boldsymbol{\mu}_p^{\mathcal{A}}, \boldsymbol{\Gamma}_p^{\mathcal{A}})$ where $\mathcal{G}$ is the Gaussian distribution. The background model has the same form of distribution, $\mathcal{G}(\mathcal{A}(\boldsymbol{h}_p)|\boldsymbol{\mu}_{bg}^{\mathcal{A}}, \boldsymbol{\Gamma}_{bg}^{\mathcal{A}})$, with parameters $\boldsymbol{\theta}_{bg}^{\mathcal{A}} = \{\boldsymbol{\mu}_{bg}^{\mathcal{A}}, \boldsymbol{\Gamma}_{bg}^{\mathcal{A}}\}$.

Shape. The shape is represented by a joint Gaussian density of the locations of features within a hypothesis, after they have been transformed into a scale-invariant space. This is done using the scale information from the features in the hypothesis, in order to avoid an exhaustive search over scale. For each hypothesis, the coordinates of the leftmost part are subtracted off the coordinates of all the parts. This enables our model to achieve translational invariance. The density can be written as $\mathcal{G}(\mathcal{X}(\mathbf{h})|\boldsymbol{\mu}^{\mathcal{X}}, \boldsymbol{\Gamma}^{\mathcal{X}})$with parameters $\boldsymbol{\theta}^{\mathcal{X}} = \{\boldsymbol{\mu}^{\mathcal{X}}, \boldsymbol{\Gamma}^{\mathcal{X}}\}$. The background shape density is uniform because we assume the features are spread uniformly over the image (which has area α) and are independent of the foreground locations.

2.1.3 Model distribution

Let us consider a *mixture model* of constellation models with Ω components. Each component ω has a mixing coefficient π_ω; a mean of shape and appearance $\boldsymbol{\mu}_\omega^{\mathcal{X}}, \boldsymbol{\mu}_\omega^{\mathcal{A}}$; a precision matrix of shape and appearance $\boldsymbol{\Gamma}_\omega^{\mathcal{X}}, \boldsymbol{\Gamma}_\omega^{\mathcal{A}}$. The $\mathcal{X}$ and $\mathcal{A}$ superscripts denote shape and appearance terms respectively. 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 learnt 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

$$p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t) = p(\boldsymbol{\pi}) \prod_{\omega} p(\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) p(\boldsymbol{\mu}_\omega^{\mathcal{X}}|\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) p(\boldsymbol{\Gamma}_\omega^{\mathcal{A}}) p(\boldsymbol{\mu}_\omega^{\mathcal{A}}|\boldsymbol{\Gamma}_\omega^{\mathcal{A}}) \tag{5}$$

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 [14, 15]. Note $\{\lambda_\omega, a_\omega, \boldsymbol{B}_\omega, \boldsymbol{m}_\omega, \beta_\omega\}$ are hyperparameters for defining their corresponding distributions of model parameters. Identical expressions apply to the appearance component in Eq. 5.

2.1.4 Bayesian decision

Recall that for the query image we wish to calculate the ratio of $p(\text{Object}|\mathcal{X}, \mathcal{A}, \mathcal{X}_t, \mathcal{A}_t)$ and $p(\text{No object}|\mathcal{X}, \mathcal{A}, \mathcal{X}_t, \mathcal{A}_t)$. It is reasonable to assume a fixed value for all model parameters when the object is not present, hence the latter term may be calculated once for all. For the former term, we use Bayes's rule to obtain the likelihood expression: $p(\mathcal{X}, \mathcal{A}|\mathcal{X}_t, \mathcal{A}_t, \text{Object})$ which expands to $\int p(\mathcal{X}, \mathcal{A}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t)\; d\boldsymbol{\theta}$. Since the likelihood $p(\mathcal{X}, \mathcal{A}|\,\boldsymbol{\theta})$ contains Gaussian densities and the parameter posterior, $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t)$ is its conjugate density (a Normal-Wishart) the integral has a closed form solution of a multivariate Student's T distribution (denoted by $\mathcal{S}$):

$$p(\mathcal{X}, \mathcal{A}|\mathcal{X}_t, \mathcal{A}_t, \text{Object}) =$$
$$\sum_{\omega=1}^{\Omega} \sum_{h=1}^{|H^n|} \tilde{\pi}_\omega\, \mathcal{S}(\mathcal{X}_h|\, g_\omega^{\mathcal{X}}, \mathbf{m}_\omega^{\mathcal{X}}, \boldsymbol{\Lambda}_\omega^{\mathcal{X}})\, \mathcal{S}(\mathcal{A}_h|\, g_\omega^{\mathcal{A}}, \mathbf{m}_\omega^{\mathcal{A}}, \boldsymbol{\Lambda}_\omega^{\mathcal{A}})$$
$$\text{where} \quad g_\omega = a_\omega + 1 - d \quad \text{and} \quad \boldsymbol{\Lambda}_\omega = \frac{\beta_\omega + 1}{\beta_\omega g_\omega} \boldsymbol{B}_\omega$$
$$\text{and} \quad \tilde{\pi}_\omega = \frac{\lambda_\omega}{\sum_{\omega'} \lambda_{\omega'}}$$

Note d is the dimensionality defined in Eq. 17. If the ratio of posteriors, R in Eq. 3, calculated using the likelihood expression above exceeds a pre-defined threshold, then the image is assumed to contain an occurrence of the learnt object category.

2.2. Learning

The process of learning an object category is unsupervised [6, 10]. The algorithm is presented with a number of train-

ing images labeled as "foreground images". It assumes there is an instance of the object category to be learnt in each image. But no other information, e.g. location, size, shape, appearance, etc., is provided.

In order to estimate a posterior distribution $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t)$ 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 as variational Bayesian expectation maximization ("VBEM"), applied to a multi-dimensional Gaussian mixture model. We first introduce the basic concept and assumptions of the variational method. Then we illustrate in more detail how such learning is applied to our model.

2.2.1 Variational methods

We have some integral we wish to evaluate: $F = \int_\theta f(\theta) d\theta$. We write $f(\theta)$ as a function of its parameters and some hidden variables, S: $f(\theta) = \int_S g(\theta, S)\, dS$. Applying Jensen's inequality to give us a lower bound on the integral, we get:

$$F = \int_{\theta,S} g(\theta, S)\, dS\, d\theta \quad (6)$$

$$\geq \exp\left(\int_{\theta,S} q(\theta, S) \log \frac{g(\theta, S)}{q(\theta, S)}\, dS\, d\theta \right) \quad (7)$$

$$\text{providing} \int_{\theta,S} q(\theta, S)\, dS\, d\theta = 1 \quad (8)$$

Variational Bayes makes the assumption that $q(\theta, S)$ is a probability density function that can be factored into $q_\theta(\theta) q_S(S)$. We then iteratively optimize q_θ and q_S using expectation maximization (EM) to maximize the value of the lower bound to the integral (see [15, 16]). If we consider $g(\theta, S)$ the "true" p.d.f., by using the above method, we are effectively decreasing the Kullback-Leibler distance between $g(\theta, S)$ and $q(\theta, S)$, hence obtaining a $q(\theta, S)$ that approximates the true p.d.f.

2.2.2 Variational Bayesian EM ("VBEM")

Recall that we have a mixture model with Ω components. Collecting all mixture components and their corresponding parameters together, we have an overall parameter vector $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}^{\mathcal{X}}, \boldsymbol{\mu}^{\mathcal{A}}, \boldsymbol{\Gamma}^{\mathcal{X}}, \boldsymbol{\Gamma}^{\mathcal{A}}\}$. For n training images, we have $\{\mathcal{X}_t^n, \mathcal{A}_t^n\}$ with $n = 1 \ldots N$. In the constellation model, each image n has $|H^n|$ hypotheses, each one of which picks out P features from $\{\mathcal{X}^n, \mathcal{A}^n\}$ to give $\{\mathcal{X}_h^n, \mathcal{A}_h^n\}$. We have two latent variables, the hypothesis $\boldsymbol{h}$ and the mixture component $\boldsymbol{\omega}$. We assume that the prior on any hypothesis always remains uniform, namely $1/|H^n|$, so it is omitted from the update equations since it is constant. We can now express the likelihood of an image n as:

$$p(\mathcal{X}^n, \mathcal{A}^n|\boldsymbol{\theta}) = \sum_{\omega=1}^{\Omega} \sum_{h=1}^{|H^n|} p(\omega_n = \omega|\boldsymbol{\pi}) p(\mathcal{X}_h^n|\boldsymbol{\mu}_\omega^{\mathcal{X}}, \boldsymbol{\Gamma}_\omega^{\mathcal{X}}) p(\mathcal{A}_h^n|\boldsymbol{\mu}_\omega^{\mathcal{A}}, \boldsymbol{\Gamma}_\omega^{\mathcal{A}}) \quad (9)$$

where $p(\boldsymbol{\omega} = \omega|\boldsymbol{\pi}) = \pi_\omega$. Both the terms involving $\mathcal{X}, \mathcal{A}$ above have a Normal form. The prior on the model parameters has the same form as the model distribution in Eq. 5

$$p(\boldsymbol{\theta}) = p(\boldsymbol{\pi}) \prod_\omega p(\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) p(\boldsymbol{\mu}_\omega^{\mathcal{X}}|\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) p(\boldsymbol{\Gamma}_\omega^{\mathcal{A}}) p(\boldsymbol{\mu}_\omega^{\mathcal{A}}|\boldsymbol{\Gamma}_\omega^{\mathcal{A}}) \quad (10)$$

where the mixing prior is $p(\boldsymbol{\pi}) = \mathcal{D}ir(\lambda_0 \boldsymbol{I}_\Omega)$, and the shape prior is a Normal-Wishart distribution $p(\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) p(\boldsymbol{\mu}_\omega^{\mathcal{X}}|\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) = \mathcal{G}(\boldsymbol{\mu}_\omega^{\mathcal{X}}|\boldsymbol{m}_0^{\mathcal{X}}, \beta_0^{\mathcal{X}} \boldsymbol{\Gamma}_\omega^{\mathcal{X}}) \mathcal{W}(\boldsymbol{\Gamma}_\omega^{\mathcal{X}}|a_0^{\mathcal{X}}, \boldsymbol{B}_0^{\mathcal{X}})$. Identical expressions apply to the appearance component of Eq. 10.

The E-Step. The central idea of VBEM is to approximate the posterior distribution $p(\boldsymbol{\theta}|\mathcal{X}, \mathcal{A})$ by an optimal approximation $q(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{h})$ that is factorizable $q(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{h}) = q(\boldsymbol{\theta}) q(\boldsymbol{\omega}, \boldsymbol{h})$, where $\boldsymbol{\omega}$ and $\boldsymbol{h}$ are hidden variables while $\boldsymbol{\theta}$ is the actual model parameter. In the E-step of VBEM, $q(\boldsymbol{\omega}, \boldsymbol{h})$ is updated according to

$$q(\boldsymbol{\omega}, \boldsymbol{h}) \propto \exp\left[I(\boldsymbol{\omega}, \boldsymbol{h})\right] \quad (11)$$

$$\text{where} \qquad I(\boldsymbol{\omega}, \boldsymbol{h}) = < \log p(\mathcal{X}, \mathcal{A}, \boldsymbol{\omega}, h|\boldsymbol{\theta}) >_{\boldsymbol{\theta}} \quad (12)$$

and the expectation is taken w.r.t. $q(\boldsymbol{\theta})$ [15]. The above equation can be further written as

$$I(\boldsymbol{\omega}, \boldsymbol{h}) = < \log p(\mathcal{X}, \boldsymbol{\omega}, h|\boldsymbol{\theta}) p(\mathcal{A}, \boldsymbol{\omega}|\mathcal{X}, \boldsymbol{h}, \boldsymbol{\theta}) >_{\boldsymbol{\theta}} \quad (13)$$

The rule for updating the indicator posterior is

$$\tilde{\gamma}_{\omega,h}^n = \tilde{\pi}_\omega \tilde{\gamma}_\omega(\mathcal{X}_h^n) \cdot \tilde{\gamma}_\omega(\mathcal{A}_h^n) \quad (14)$$

$$\text{where} \qquad \log(\tilde{\pi}_\omega) = \Psi(\lambda_\omega) - \Psi(\sum_{\omega'} \lambda_{\omega'}) \quad (15)$$

$$\tilde{\gamma}_\omega(\mathcal{X}_h^n) = \exp\left[-\frac{1}{2} (\mathcal{X}_h^n - \boldsymbol{m}_\omega^{\mathcal{X}})^T \bar{\boldsymbol{\Gamma}}_\omega^{\mathcal{X}} (\mathcal{X}_h^n - \boldsymbol{m}_\omega^{\mathcal{X}}) \right] \cdot (\tilde{\Gamma}_\omega^{\mathcal{X}})^{1/2} \exp\left[\frac{-d^{\mathcal{X}}}{2\beta_\omega^{\mathcal{X}}} \right] \quad (16)$$

$$\log \tilde{\Gamma}_\omega^{\mathcal{X}} = \sum_{i=1}^{d^{\mathcal{X}}} \Psi((a_\omega^{\mathcal{X}} + 1 - i)/2) - \log |\boldsymbol{B}_{\boldsymbol{\omega}}^{\boldsymbol{\mathcal{X}}}| + d^{\mathcal{X}} \log 2 \quad (17)$$

$$\bar{\boldsymbol{\Gamma}}_\omega^{\mathcal{X}} = a_\omega^{\mathcal{X}} (\boldsymbol{B}_{\boldsymbol{\omega}}^{\boldsymbol{\mathcal{X}}})^{-1} \quad (18)$$

where $\Psi()$ is the Digamma function and $d^{\mathcal{X}}$ is the dimensionality of $\mathcal{X}_h^n$. Superscript $\mathcal{X}$ indicates the parameters are related to the shape component of the model. The RHS of the above equations consist of hyperparameters for the parameter posteriors (i.e. λ, $\boldsymbol{m}$, $\boldsymbol{B}$, β and a). $\tilde{\gamma}_\omega(\mathcal{A}_h^n)$ is computed exactly the same way as $\tilde{\gamma}_\omega(\mathcal{X}_h^n)$, using the corresponding parameters of the appearance component. We then normalize to give

$$\gamma_{\omega,h}^n = \frac{\tilde{\gamma}_{\omega,h}^n}{\sum_{\omega',h'} \tilde{\gamma}_{\omega',h'}^n} \quad (19)$$

which is the probability that component ω is responsible for hypothesis $\boldsymbol{h}$ of the n^{th} training image.

The M-Step. In the M-step, $q(\boldsymbol{\theta})$ is updated according to

$$q(\boldsymbol{\theta}) \propto \exp\left[I(\boldsymbol{\theta})\right] p(\boldsymbol{\theta}) \tag{20}$$

$$\text{where} \qquad I(\boldsymbol{\theta}) = < \log p(\mathcal{X}, \mathcal{A}, \boldsymbol{\omega}, h|\theta) >_{\boldsymbol{\omega},\boldsymbol{h}} \tag{21}$$

Again, the above equation can be written as

$$I(\boldsymbol{\theta}) = < \log p(\mathcal{X}, \boldsymbol{\omega}, h|\boldsymbol{\theta}) p(\mathcal{A}, \boldsymbol{\omega}|\mathcal{X}, \boldsymbol{h}, \boldsymbol{\theta}) >_{\boldsymbol{\omega},\boldsymbol{h}} \tag{22}$$

and the expectation is taken w.r.t. $q(\boldsymbol{\omega}, \boldsymbol{h})$.

We show here the update rules for the shape components. The equations are exactly the same for the appearance components. We define the following variables

$$\bar{\pi}_\omega = \frac{1}{N} \sum_{n=1}^{N} \sum_{h=1}^{|H^n|} \gamma_{\omega,h}^n \tag{23}$$

$$\bar{N}_\omega = N \bar{\pi}_\omega \tag{24}$$

$$\bar{\boldsymbol{\mu}}_\omega^{\mathcal{X}} = \frac{1}{\bar{N}_\omega} \sum_{n=1}^{N} \sum_{h=1}^{|H^n|} \gamma_{\omega,h}^n \mathcal{X}_h^n \tag{25}$$

$$\bar{\boldsymbol{\Sigma}}_\omega^{\mathcal{X}} = \frac{1}{\bar{N}_\omega} \sum_{n=1}^{N} \sum_{h=1}^{|H^n|} \gamma_{\omega,h}^n (\mathcal{X}_h^n - \bar{\boldsymbol{\mu}}_\omega^{\mathcal{X}})(\mathcal{X}_h^n - \bar{\boldsymbol{\mu}}_\omega^{\mathcal{X}})^T \tag{26}$$

We then update the hyperparameters as follows. For the mixing coefficients we have a Dirichlet distribution $q(\boldsymbol{\pi}) = \mathcal{D}ir(\boldsymbol{\lambda})$ where the hyperparameters are updated by: $\lambda_\omega = \bar{N}_\omega + \lambda_0$. For the means, we have $q(\boldsymbol{\mu}_\omega^{\mathcal{X}}|\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) = \mathcal{G}(\boldsymbol{m}_\omega^{\mathcal{X}}, \beta_\omega^{\mathcal{X}} \boldsymbol{\Gamma}_\omega^{\mathcal{X}})$ where

$$\boldsymbol{m}_\omega^{\mathcal{X}} = \frac{\bar{N}_\omega \bar{\boldsymbol{\mu}}_\omega^{\mathcal{X}} + \beta_0^{\mathcal{X}} \boldsymbol{m}_0^{\mathcal{X}}}{\bar{N}_\omega + \beta_0^{\mathcal{X}}} \tag{27}$$

$$\beta_\omega^{\mathcal{X}} = \bar{N}_\omega + \beta_0^{\mathcal{X}} \tag{28}$$

For the noise precision matrix we have a Wishart density $q(\boldsymbol{\Gamma}_\omega^{\mathcal{X}}) = \mathcal{W}(a_\omega^{\mathcal{X}}, \boldsymbol{B}_\omega^{\mathcal{X}})$ where

$$\boldsymbol{B}_\omega^{\mathcal{X}} = \frac{\bar{N}_\omega \beta_0^{\mathcal{X}} (\bar{\boldsymbol{\mu}}_\omega^{\mathcal{X}} - \boldsymbol{m}_0^{\mathcal{X}})(\bar{\boldsymbol{\mu}}_\omega^{\mathcal{X}} - \boldsymbol{m}_0^{\mathcal{X}})^T}{\bar{N}_0 + \beta_0^{\mathcal{X}}} + \bar{N}_\omega \bar{\boldsymbol{\Sigma}}_\omega^{\mathcal{X}} + \boldsymbol{B}_0^{\mathcal{X}}$$

$$a_\omega^{\mathcal{X}} = \bar{N}_\omega + a_0^{\mathcal{X}} \tag{29}$$

3. Implementation

3.1. Feature detection and representation

We use the same features as in [10]. They are found using the detector of Kadir and Brady [17]. 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 $\mathcal{X}$. Figure 2 illustrates this on two images from the motorbike and airplane datasets. 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 PCA. A fixed PCA basis, pre-calculated from the background datasets, is used for this task, which gives us the first 10 principal components from each patch. The principal components from all patches and images form $\mathcal{A}$.

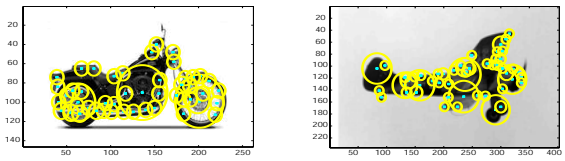
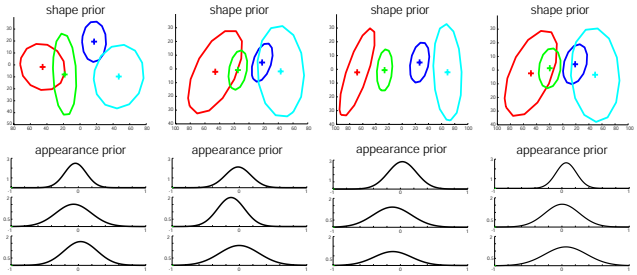


Figure 2: Output of the feature detector

3.2. Learning

The task of learning is to estimate the distribution of $p(\boldsymbol{\theta}|\mathcal{X}_t, \mathcal{A}_t)$. VBEM estimates the hyperparameters $\{\boldsymbol{m}^{\mathcal{X}}, \beta^{\mathcal{X}}, a^{\mathcal{X}}, \boldsymbol{B}^{\mathcal{X}}, \boldsymbol{m}^{\mathcal{A}}, \beta^{\mathcal{A}}, a^{\mathcal{A}}, \boldsymbol{B}^{\mathcal{A}}\}$ that define the distributions of the parameters $\{\boldsymbol{\pi}, \boldsymbol{\mu}^{\mathcal{X}}, \boldsymbol{\mu}^{\mathcal{A}}, \boldsymbol{\Gamma}^{\mathcal{X}}, \boldsymbol{\Gamma}^{\mathcal{A}}\}$. The goal is to find the distribution of $p(\boldsymbol{\theta})$ that best explains the data $\{\mathcal{X}_t, \mathcal{A}_t\}$ from all the training images.



(a) Face prior (b) Motorbike prior (c) Spotted cat prior (d) Airplane prior

Figure 3: Prior distribution for shape mean ($\boldsymbol{\mu}^{\mathcal{X}}$) and appearance mean ($\boldsymbol{\mu}^{\mathcal{A}}$) for faces, motorbikes, spotted cats and airplanes. Each prior's hyperparameters are estimated from models learnt with maximum likelihood methods, using "the other" datasets [10]. For example, face prior is obtained from models of motorbikes, spotted cats and airplanes. Only the first three PCA dimensions of the appearance priors are displayed. All four parts of the appearance begin with the same prior distribution for each PCA dimension.

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. 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 learnt a sufficient number of classes 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 classes. We use four classes of object

models learnt in a ML manner from [10] to form our priors. They are: spotted cats, motorbikes, faces and airplanes. Since we wish to learn the same four datasets with our algorithm, we use a "leave one out" strategy. For example, when learning motorbikes we obtain priors by averaging the learnt model parameters from the other three categories(i.e. spotted cats, faces and airplanes), hence avoiding the incorporation of an existing motorbike model. The hyperparameters of the prior are then estimated from the parameters of the existing category models. Figure 3.2 shows the prior shape model for each category.

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. Learning is halted when the parameters change per iteration falls below a certain threshold (10^{-4}) or exceeds a maximum number of iterations (typically 500). In general convergence occurs within less than 100 iterations. Repeated runs with the same data but different random initializations consistently give virtually indistinguishable classification results. 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 proved inaccurate so the parameters are estimated from a set of background images and not updated within the VBEM iteration. Learning a class takes roughly about $3-5$ seconds on a 2.8 GHz machine when number of training images is less than 10 and the model is composed of 4 parts. The algorithm is implemented in Matlab. It is also worth mentioning that the current algorithm does not utilize any efficient search method unlike [10]. 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 [10]. Were efficient search techniques used, 6-7 parts could be learnt, since the VBEM update equations are require the same amount of computation as the traditional ML ones. However, all our experiments currently use 4 part models for both the current algorithm and ML.

3.3. Experimental Setup

Each experiment is carried out in the following way. Each dataset 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, which form the test set. We then learn models using both Bayesian and ML approaches and evaluate their performance on the test set. For evaluation purposes, we also use 50 images from a background dataset of assorted junk images from the Internet. 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 fails to converge, so we only show results for the Bayesian One-Shot algorithm in this case.

When evaluating the models, the decision is a simple object present/absent one. All performance values are quoted as equal error rates from the receiver-operating characteristic (ROC) (i.e. p(True positive) = 1 - p(False alarm)). ROC curve is obtained by testing the model on 50 foreground test images and 50 background images . For example, a value of 85% means that 85% of the foreground images are correctly classified but 15% of the background images are incorrectly classified (i.e. false alarms). A limited amount of preprocessing is performed on some of the datasets. For the motorbikes and airplanes some images are flipped to ensure all objects are facing the same way. 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 except for the different priors obtained as described above, all parameters remain the same for learning different categories of objects.

4. Results

Our experiments demonstrate the benefit of using prior information in learning new object categories. Fig. 4-7 show models learnt by the Bayesian One-Shot algorithm on the four datasets. It is important to notice that the "priors" alone are not sufficient for object categorization (Fig. 4-7 (a)). But by incorporating this general knowledge into the training data, the algorithm is capable of learning a sensible model with even 1 training example. For instance, in Fig. 4(c), we see that the 4-part model has captured the essence of a face (e.g. eyes and nose). In this case it achieves a recognition rate as high as 82%, given only 1 training example. Table 1 compares our algorithm with some published object categorization methods. Note that our algorithm has significantly faster learning speed due to much smaller number of training examples.

Algorithm	Training number	Learning speed	Categories	Error Rate (%)	Remarks
Bayesian One-Shot	1 to 5	<1min	faces, motorbikes, spotted cats, airplanes	$8-22$	4-part model, unsupervised
[6] [10]	200 to 400	hours	faces, motorbikes, spotted cats, airplanes, cars	$5.6-10$	6-part model, unsupervised
[7]	$10,000$	weeks	faces	$7.3-21.7$	aligned manually
[5]	~ 2000	days	faces, cars	$5.6-17$	aligned manually
[2]	~ 500	days	faces	$7.5-24.1$	aligned manually

Table 1: A comparison between the Bayesian One-Shot learning algorithm and alternative approaches to object category recognition. The error rate quoted for the Bayesian One-Shot model is for 5 training images.

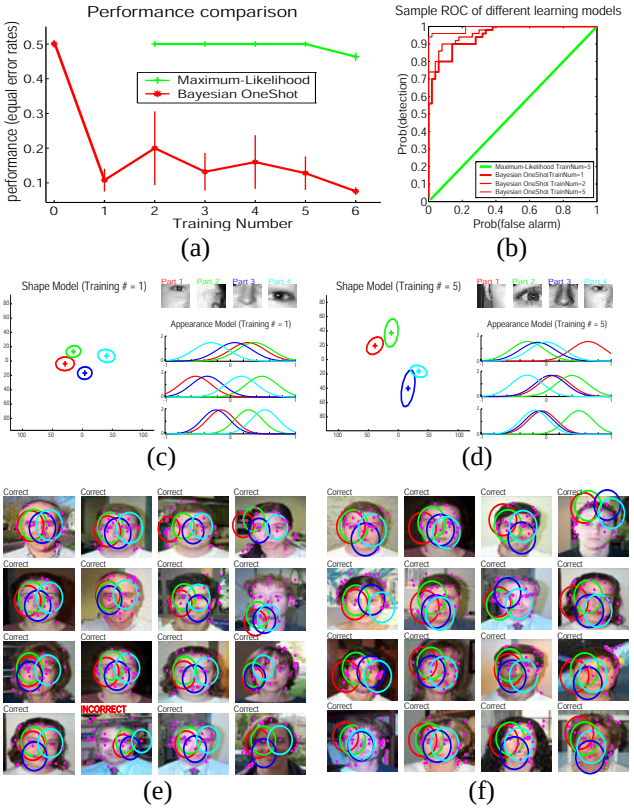


Figure 4: Summary of face model. **(a)** Test performances of the algorithm given $0-6$ number of training image(s) (red). 0 number of training image is the case of using the prior model only. Note this "general" information itself is not sufficient for categorization. Each performance is obtained by 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 ML cannot learn the degenerate case of a single training image. **(b)** Sample ROC curves for the algorithm (red) compared with the ML algorithm (green). The curves shown here use typical models drawn from the repeated runs summarized in **(a)**. Details are shown in **(c)-(f)**. **(c)** A typical model learnt with 1 training example. The left panel shows the shape component of the model. The four $+$'s and ellipses indicate the mean and variance in position of each part. The covariance terms are not shown. The top right panel shows the detected feature patches in the training image closest to the mean of the appearance densities for each of the four parts. The bottom right panel shows the mean appearance distributions for the first 3 PCA dimensions. Each color indicates one of the four parts. Note the shape and appearance distributions are much more "model specific" compare to the "general" prior model in Fig.3.2. **(e)** Some sample foreground test images for the model learnt in **(c)**, 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). **(d)** and **(f)** are similar to **(c)** and **(e)**. But the model is learnt from 5 training images.

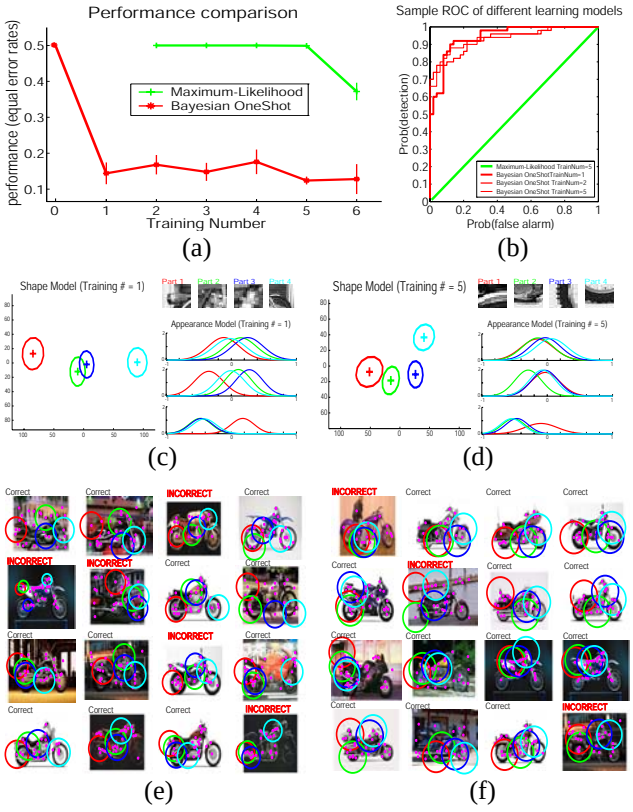


Figure 5: Summary of motorbike model.

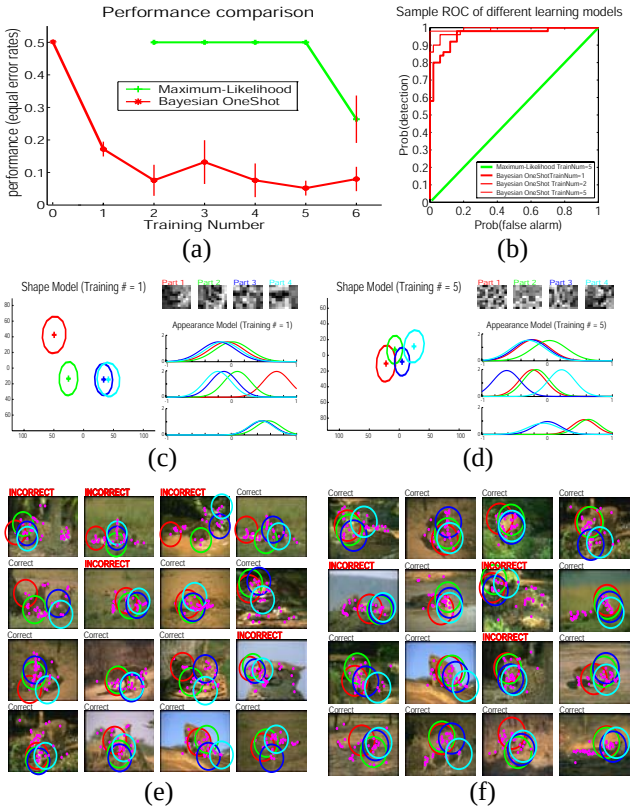


Figure 6: Summary of spotted cat model.

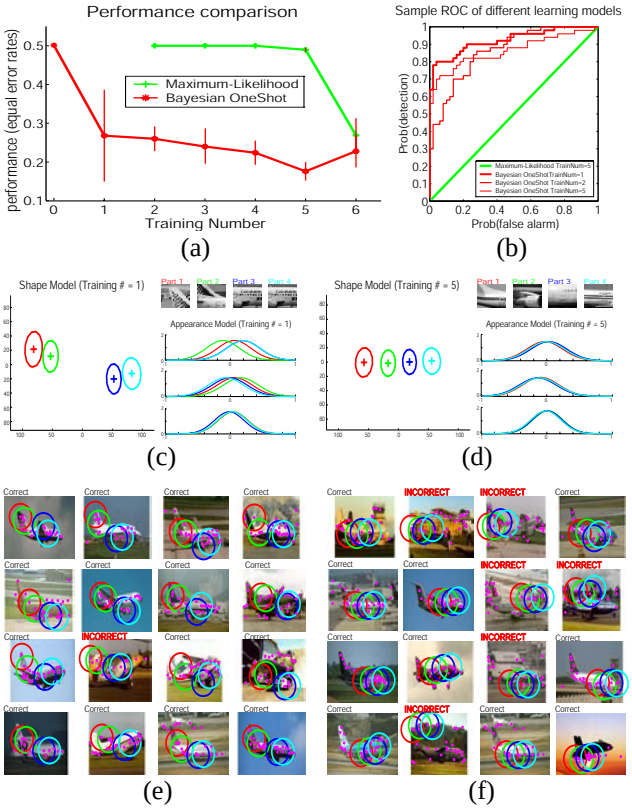


Figure 7: Summary of airplane model.

5. Conclusions and future work

We have demonstrated that given a single example (or just a few), we can learn a new object category. As Table 1 shows, this is beyond the capability of existing algorithms. In order to explore this idea we have developed a Bayesian learning framework based on representing object categories with probabilistic models. "General" information coming from previously learnt categories is represented with a suitable prior probability density function on the parameters of such models. Our experiments, conducted on realistic images of four categories, are encouraging in that they show that very few (1 to 5) training examples produce models that are already able to discriminate images containing the desired objects from images not containing them with error rates around $8 - 22\%$.

A number of issues are still unexplored. First and foremost, more comprehensive experiments need to be carried out on a larger number of categories, in order to understand how prior knowledge improves with the number of known categories, and how categorical similarity affects the process. Second, in order to make our experiments practical we have simplified the probabilistic models that are used for representing objects. For example a probabilistic model for occlusion is not implemented in our experiments [6, 9, 10]. Third, it would be highly valuable for practical applications (e.g. a vehicle roving in an unknown environment) to develop an incremental version of our algorithm, where each training example will incrementally update the probability density function defined on the parameters of each object category [18]. In addition, the minimal training set and learning time that appear to be required by our algorithm makes it possible to conceive of visual learning applications where real-time training and user interaction are important.

Acknowledgments

We would like to thank David MacKay, Brian Ripley and Yaser Abu-Mostafa for their most useful discussions.

References

[1] I. Biederman, "Recognition-by-Components: A Theory of Human Image Understanding." *Psychological Review*, vol. 94, pp.115-147, 1987.

[2] H.A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 23-38, Jan, 1998.

[3] K. Sung and T. Poggio, "Example-based Learning for View-based Human Face Detection", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 1, pp. 39-51, Jan, 1998.

[4] Y. Amit and D. Geman, "A computational model for visual selection", *Neural Computation*, vol. 11, no. 7, pp1691-1715, 1999.

[5] H. Schneiderman, and T. Kanade, "A Statistical Approach to 3D Object Detection Applied to Faces and Cars", *Proc. CVPR*, pp. 746-751, 2000.

[6] M. Weber, M. Welling and P. Perona, "Unsupervised learning of models for recognition", *Proc. 6th ECCV*, vol. 2, pp. 101-108, 2000.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features", *Proc. CVPR*, vol. 1, pp. 511-518, 2001.

[8] D.C. Knill and W. Richards, (ed.) *Perception as Bayesian Inference*, Cambridge University Press, 1996.

[9] M.C. Burl, M. Weber, and P. Perona, "A probabilistic approach to object recognition using local photometry and global geometry", *Proc. ECCV*, pp.628-641, 1998.

[10] R. Fergus, P. Perona and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning", *Proc. CVPR*, vol. 2, pp. 264-271, 2003.

[11] S. Waterhouse, D. MacKay, and T. Robinson, "Bayesian methods for mixtures of experts", *Proc. NIPS*, pp. 351-357, 1995.

[12] D. MacKay, "Ensemble learning and evidence maximization", *Proc. NIPS*, 1995.

[13] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola and L.K. Saul, "An introduction to variational methods for graphical models", *Machine Learning*, vol. 37, pp. 183-233, 1999.

[14] H. Attias, "Inferring parameters and structure of latent variable models by variational bayes", *15th conference on Uncertainty in Artificial Intelligence*, pp. 21-30, 1999.

[15] W.D. Penny, "Variational Bayes for d-dimensional Gaussian mixture models", Tech. Rep., University College London, 2001.

[16] T.P. Minka, "Using lower bounds to approximate integrals", Tech. Rep., Carnegie Mellon University, 2001.

[17] T. Kadir and M. Brady, "Scale, saliency and image description", *International Journal of Computer Vision*, vol. 45, no. 2, pp. 83-105, 2001.

[18] R.M. Neal and G.E. Hinton, "A view of the EM algorithm that justifies incremental, sparse and other variants." in *Learning in Graphical Models* edt. M.I. Jordan, pp. 355-368, Kluwer academic press, Norwell, 1998.