Available online at www.sciencedirect.com
ScienceDirect
Computer Vision and Image Understanding 106 (2007) 59–70
Computer Vision and Image Understanding
www.elsevier.com/locate/cviu

Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories

Li Fei-Fei [a,*], Rob Fergus [b], Pietro Perona [c]

[a] Princeton University, 35 Olden St., Princeton, NJ 08540, USA
[b] Oxford University, Parks Road, Oxford OX1 3PJ, UK
[c] California Institute of Technology, 136-93 Mail Code, Pasadena, CA 91125, USA

Received 15 August 2005; accepted 19 September 2005
Available online 9 March 2007
Communicated by Arthur E. C. Pece

Abstract

Current computational approaches to learning visual object categories require thousands of training images, are slow, cannot learn in an incremental manner and cannot incorporate prior information into the learning process. In addition, no algorithm presented in the literature has been tested on more than a handful of object categories. We present an method for learning object categories from just a few training images. It is quick and it uses prior information in a principled way. We test it on a dataset composed of images of objects belonging to 101 widely varied categories. Our proposed method is based on making use of prior information, assembled from (unrelated) object categories which were previously learnt. A generative probabilistic model is used, which represents the shape and appearance of a constellation of features belonging to the object. The parameters of the model are learnt incrementally in a Bayesian manner. Our incremental algorithm is compared experimentally to an earlier batch Bayesian algorithm, as well as to one based on maximum likelihood. The incremental and batch versions have comparable classification performance on small training sets, but incremental learning is significantly faster, making real-time learning feasible. Both Bayesian methods outperform maximum likelihood on small training sets.

Keywords: Object recognition; Categorization; Generative model; Incremental learning; Bayesian model

1. Introduction

One of the most exciting and difficult open problems of machine vision is enabling a machine to recognize objects and object categories in images. Significant progress has been made on the issues of representation of objects [17,18] and object categories [2,3,5–8] with a broad agreement for models that are composed of 'parts' (textured patches, features) and 'geometry' (or mutual position of the parts). Much work remains to be done on the issue of category learning, i.e. estimating the model parameters that are to be associated to a given category. Three difficulties face us at the moment. First, a human operator must identify explicitly each category to be learned, while it would be desirable to let a machine identify automatically each category from a broad collection of images. While Weber et al. [5] have demonstrated encouraging results on identifying automatically three categories in a limited image database, in order to reach human performance one would like to see tens of thousands of categories identified automatically from possibly millions of images [1]. Second, most algorithms require the image of each exemplar object to be geometrically normalized and aligned with a prototype (e.g. clicking on eyes, nose and mouse corners on face images). This is expensive and tedious; furthermore, it is problematic when fiducial points are not readily identifiable (can we find a natural alignment for images of octopus, of cappuccino machines, of human bodies in different poses?). Progress on this topic has been recently reported by Weber et al. [5] who

* Corresponding author.
E-mail addresses: feifeili@cs.princeton.edu (L. Fei-Fei), fergus@robots.ox.ac.uk (R. Fergus), perona@vision.caltech.edu (P. Perona).

1077-3142/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2005.09.012

proposed and demonstrated a method for training in clutter without supervision (see also the follow-up paper by Fergus et al. [8] proposing an improved scale-invariant algorithm). The third challenge has to do with the size of the training set that is required: as many as 10^4 training examples for some algorithms. This is not surprising: a well-known rule-of-thumb says that the number of training examples has to be 5–10 times the number of object parameters—hence the large training sets for models containing hundreds of parameters. Yet, humans are often able to learn new categories from a much smaller training set (how many cell-phones did we need to see in order to learn to recognize one?). On this issue Fei-Fei et al. [4] recently proposed a Bayesian framework to use priors derived from previously learned classes in order to speed up learning of a new class. In a limited set of experiments on four categories they showed that 1–3 training examples are sufficient to learn a new category. Their method is batch, in that the training examples need to be considered simultaneously.

In this paper, we explore the third issue further. First, we argue that batch learning is an undesirable limitation. An organism, or a machine, exploring the environment should not be required to store explicitly the set of training images that it has encountered so far. The current best estimate of a given class should be a sufficient memory of past experience. To this end we develop an incremental Bayesian algorithm and we study the performance/memory trade-off. Second, we collected a training set of 101 categories and we assess the new incremental Bayesian algorithm against the batch Bayesian algorithm of Fei-Fei et al. and against the maximum likelihood method of Fergus et al. We wish to emphasize at this point that previous work on object categories has been tested for the most part on 1 or 2 categories [3] with the exception of Weber et al. [5] who tested on four and Fergus et al. [8] who tested on six. We consider the effort to collect and test on dataset that is 15 times larger one of the major contributions of this paper.

In Section 2 we outline the generative framework to represent object classes. Although some of the details can be found in [4], we reproduce them here for the sake of clarity. In Section 3, we show how to apply variational inference to learn the parameters of the constellation model. In [4], a batch learning algorithm was developed for the same framework. Here we extend this into an incremental method in which the algorithm is only given a single training image at a time during the learning stage, a process much more natural for living organisms. As an overview, in Fig. 1 we schematically compare and contrast the structure of the Bayesian algorithm in learning and recognition with the Maximum likelihood algorithm used by [5,7,8]. We then discuss in details the experimental setup, with emphasis on our dataset of 101 object categories in Section 4. Finally in Section 5 we present experimental results on the 101 object categories. We conclude this paper with a discussion in Section 6.

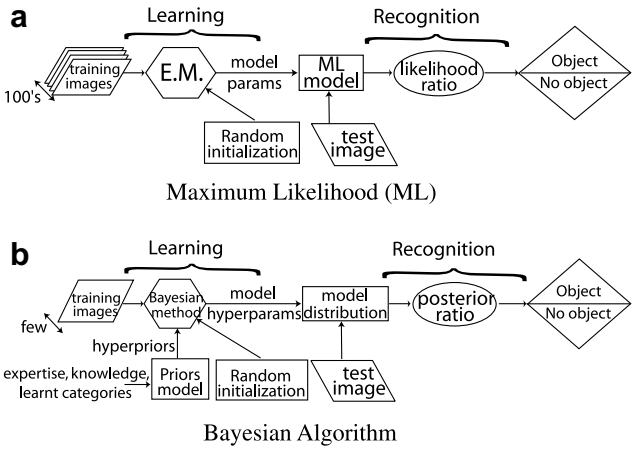


Fig. 1. Schematic comparison between maximum likelihood algorithm [5,7,8] and the Bayesian learning algorithm [4]. Note the different number of training images, the different learning algorithm and recognition criterion.

2. The generative model

To illustrate the generative model, 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 $\mathscr{X}$, and appearances $\mathscr{A}$. We now make a Bayesian decision, R. For clarity, we explicitly express training images through the detected feature locations $\mathscr{X}_t$ and appearances $\mathscr{A}_t$.

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

$$= \frac{p(\mathscr{X},\mathscr{A}|\mathscr{X}_t,\mathscr{A}_t,\text{Object})p(\text{Object})}{p(\mathscr{X},\mathscr{A}|\mathscr{X}_t,\mathscr{A}_t,\text{Noobject})p(\text{No Object})} \tag{2}$$

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

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).

2.1. The constellation model

Our chosen representation for object categories is based on the *constellation model* introduced by Burl [7] and developed further by Weber et al. [5] and Fergus et al. [8]. A constellation model consists of a number of parts, each encoding information on both the shape and appearance. The appearance of each part is modeled and the shape of the object is represented by the mutual position of the parts [8]. The entire model is generative and probabilistic, so appearance and shape are all modeled by probability density functions, which are Gaussians. The model is best explained by first considering recognition. We have learned

a generative object model, with P parts and a posterior distribution on the parameters $\boldsymbol{\theta}$: $p(\boldsymbol{\theta}|\mathscr{X}_t, \mathscr{A}_t)$ where $\mathscr{X}_t$ and $\mathscr{A}_t$ are the location and appearances of interesting features found in the training data. We assume that all non-object images can also be modeled by a background with a single set of parameters $\boldsymbol{\theta}_{\text{bg}}$ which are fixed. The ratio of the priors may be estimated from the training set or set manually (usually to 1). Our decision then requires the calculation of the ratio of the two likelihood functions. In order to do this, the likelihoods may be factored as follows:

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

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* which allocates each image feature either to an object or to the background.

2.1.1. Appearance

Each feature's appearance is represented as a point in some appearance space. Each part p has a Gaussian density (denoted by $\mathscr{G}$) within this space, with mean and precision parameters $\boldsymbol{\theta}_p^{\mathscr{A}} = \{\mu_p^{\mathscr{A}}, \Gamma_p^{\mathscr{A}}\}$ which is independent of other parts' densities.

2.1.2. Shape

The shape is represented by a joint Gaussian density of the locations of features within a hypothesis. For each hypothesis, the coordinates of all parts are subtracted off from the left most part coordinates. Additionally, it is scale is used to normalize the constellation. This enables our model to achieve scale and translational invariance. The density has parameters $\boldsymbol{\theta}^{\mathscr{X}} = \{\mu^{\mathscr{X}}, \Gamma^{\mathscr{X}}\}$.

2.2. 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 $\boldsymbol{\mu}_\omega^{\mathscr{X}}, \boldsymbol{\mu}_\omega^{\mathscr{A}}$; a precision matrix of shape and appearance $\boldsymbol{\Gamma}_\omega^{\mathscr{X}}, \boldsymbol{\Gamma}_\omega^{\mathscr{A}}$. The $\mathscr{X}$ and $\mathscr{A}$ superscripts denote shape and appearance terms respectively. Collecting all mixture components and their corresponding parameters together, we obtain an overall parameter vector $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{\mu}^{\mathscr{X}}, \boldsymbol{\mu}^{\mathscr{A}}, \boldsymbol{\Gamma}^{\mathscr{X}}, \boldsymbol{\Gamma}^{\mathscr{A}}\}$. Assuming we have now learnt the model distribution $p(\boldsymbol{\theta}|\mathscr{X}_t, \mathscr{A}_t)$ from a set of training data $\mathscr{X}_t$ and $\mathscr{A}_t$, we define the model distribution in the following way

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

where the mixing component is a symmetric Dirichlet: $p(\boldsymbol{\pi}) = \mathscr{D}ir(\lambda_\omega \mathbf{I}_\Omega)$, the distribution over the shape precisions is a Wishart $p(\boldsymbol{\Gamma}_\omega^{\mathscr{X}}) = \mathscr{W}(\boldsymbol{\Gamma}_\omega^{\mathscr{X}}|a_\omega^{\mathscr{X}}, \mathbf{B}_\omega^{\mathscr{X}})$ and the distribution over the shape mean conditioned on the precision matrix is Normal: $p(\boldsymbol{\mu}_\omega^{\mathscr{X}}|\boldsymbol{\Gamma}_\omega^{\mathscr{X}}) = \mathscr{G}(\boldsymbol{\mu}_\omega^{\mathscr{X}}|\mathbf{m}_\omega^{\mathscr{X}}, \beta_\omega^{\mathscr{X}}\boldsymbol{\Gamma}_\omega^{\mathscr{X}})$. Together the shape distribution $p(\boldsymbol{\mu}_\omega^{\mathscr{X}}, \boldsymbol{\Gamma}_\omega^{\mathscr{X}})$ is a Normal-Wishart density [9,13]. Note that the set of variables $\{\lambda_\omega, a_\omega, \boldsymbol{B}_\omega, \boldsymbol{m}_\omega, \beta_\omega\}$ are hyper-parameters for defining their corresponding distributions of model parameters. Identical expressions apply to the appearance component in Eq. (5). A graphical model representation is shown in Fig. 2, where only one mixture component is used.

2.3. Bayesian decision

Recall that for the query image we wish to calculate the ratio of $p(\text{Object}|\mathscr{X}, \mathscr{A}, \mathscr{X}_t, \mathscr{A}_t)$ and $p(\text{No Object}|\mathscr{X}, \mathscr{A}, \mathscr{X}_t, \mathscr{A}_t)$. 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: $p(\mathscr{X}, \mathscr{A}|\mathscr{X}_t, \mathscr{A}_t, \text{Object})$ which expands to $\int p(\mathscr{X}, \mathscr{A}|\boldsymbol{\theta}) p(\boldsymbol{\theta}|\mathscr{X}_t, \mathscr{A}_t)\, \mathrm{d}\boldsymbol{\theta}$. Since the likelihood $p(\mathscr{X}, \mathscr{A}|\boldsymbol{\theta})$ contains Gaussian densities and the parameter posterior, $p(\boldsymbol{\theta}|\mathscr{X}_t, \mathscr{A}_t)$ is its conjugate density (a Normal-Wishart) the integral has a closed form solution of a multivariate Student's T distribution (denoted by $\mathscr{S}$):

$$p(\mathscr{X}, \mathscr{A}|\mathscr{X}_t, \mathscr{A}_t, \text{Object}) = \sum_{\omega=1}^{\Omega} \sum_{h=1}^{|H^n|} \tilde{\pi}_\omega \mathscr{S}\left(\mathscr{X}_h|g_\omega^{\mathscr{X}}, \mathbf{m}_\omega^{\mathscr{X}}, \boldsymbol{\Lambda}_\omega^{\mathscr{X}}\right) \times \mathscr{S}\left(\mathscr{A}_h|g_\omega^{\mathscr{A}}, \mathbf{m}_\omega^{\mathscr{A}}, \boldsymbol{\Lambda}_\omega^{\mathscr{A}}\right) \quad (6)$$

$$g_\omega = a_\omega + 1 - d$$

$$\boldsymbol{\Lambda}_\omega = \frac{\beta_\omega + 1}{\beta_\omega g_\omega} \mathbf{B}_\omega$$

$$\tilde{\pi}_\omega = \frac{\lambda_\omega}{\sum_{\omega'} \lambda_{\omega'}}$$

Note d is the dimensionality of the parameter vector. 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.

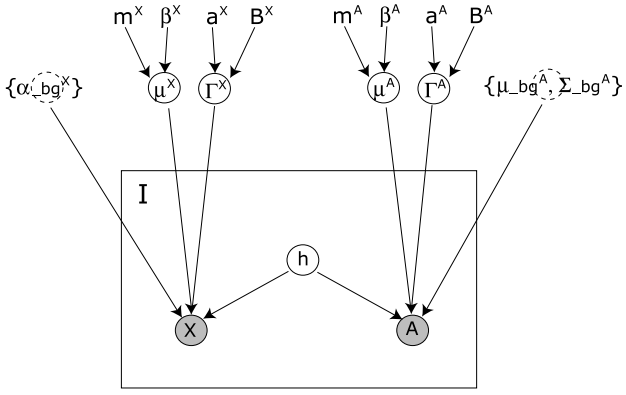


Fig. 2. A graphical model representation of the constellation model. Note that we assume only one mixture component in this representation.

3. Learning the generative model: batch vs. incremental

The task in learning is to estimate the density $p(\boldsymbol{\theta}|\mathscr{X}_t, \mathscr{A}_t)$. This is done using the Variational Bayes procedure [9–11]. It approximates the posterior distribution $p(\boldsymbol{\theta}|\mathscr{X}_t, \mathscr{A}_t)$ by $q(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{h})$. ω is the mixture component label and $\boldsymbol{h}$ is the hypothesis. Using Bayes' rule: $q(\boldsymbol{\theta}, \boldsymbol{\omega}, \mathbf{h}) \approx p(\boldsymbol{\theta}|\mathscr{X}_t, \mathscr{A}_t) \propto p(\mathscr{X}_t, \mathscr{A}_t|\boldsymbol{\theta})p(\boldsymbol{\theta})$. The likelihood terms use Gaussian densities and by assuming priors of a conjugate form, in this case a Normal-Wishart, our posterior q-function is also a Normal-Wishart density. The variational Bayes procedure is a variant of EM which iteratively updates the hyper-parameters and latent variables to monotonically reduces the Kullback-Liebler distance between $p(\boldsymbol{\theta}|\mathscr{X}_t, \mathscr{A}_t)$ and $q(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{h})$. Using this approach allows us to incorporate prior information in a systematic way and is far more powerful that a maximum likelihood approach used in [8]. We first briefly give an overview of the algorithm [4], based on [9], which is a batch learning algorithm. Then we introduce the new incremental version of the algorithm.

3.1. Batch learning

There are two stages to learning: an E-step where the responsibilities of the hidden variables are calculated and an M-step where we update the hyper-parameters of $q(\boldsymbol{\theta}, \boldsymbol{\omega}, \boldsymbol{h})$, $\boldsymbol{\Theta} = \{\boldsymbol{\lambda}, \boldsymbol{m}, \boldsymbol{\beta}, \boldsymbol{a}, \boldsymbol{B}\}$. For each image, n we calculate the responsibilities:

$$\tilde{\gamma}^n_{\omega,h} = \tilde{\pi}_\omega \tilde{\gamma}_\omega(\mathscr{X}^n_h)\tilde{\gamma}_\omega(\mathscr{A}^n_h) \tag{7}$$

using the update rules given in [9]. The hyper-parameters are updated from these responsibilities. This is done by computing the sufficient statistics. While the update rules for the shape components are shown, they are of the same form for the appearance terms. The sufficient statistics, for mixture component ω are calculated as follows:

$$\bar{\pi}_\omega = \frac{1}{N}\sum_{n=1}^{N}\sum_{h=1}^{|H^n|}\gamma^n_{\omega,h} \quad \text{and } \bar{N}_\omega = N\bar{\pi}_\omega \tag{8}$$

$$\bar{\boldsymbol{\mu}}^{\mathscr{X}}_\omega = \frac{1}{\bar{N}_\omega}\sum_{n=1}^{N}\sum_{h=1}^{|H^n|}\gamma^n_{\omega,h}\mathbf{X}^n_h \quad \text{and} \tag{9}$$

$$\bar{\boldsymbol{\Sigma}}^{\mathscr{X}}_\omega = \frac{1}{\bar{N}_\omega}\sum_{n=1}^{N}\sum_{h=1}^{|H^n|}\gamma^n_{\omega,h}(\mathbf{X}^n_h - \bar{\boldsymbol{\mu}}^{\mathscr{X}}_\omega)(\mathbf{X}^n_h - \bar{\boldsymbol{\mu}}^{\mathscr{X}}_\omega)^{\mathrm{T}} \tag{10}$$

Note that to compute these, we need the responsibilities from across all images. From these we can update the hyper-parameters (update rules are in [9]).

3.2. Incremental learning

We now give an incremental version of the update rules, based on Neal and Hinton's adaptation of conventional EM [16]. Let us assume that we have a model with hyper-parameters $\boldsymbol{\Theta} = \{\boldsymbol{\lambda}, \boldsymbol{m}, \boldsymbol{\beta}, \boldsymbol{a}, \boldsymbol{B}\}$, estimated using M previous images ($M \geqslant 0$) and we have N new images ($N \geqslant 1$) with which we wish to update the model. From the M previous images, we have retained sufficient statistics $\pi^e_\omega, \boldsymbol{\mu}^e_\omega, \boldsymbol{\Sigma}^e_\omega$ for each mixture component ω. We then compute the responsibilities for the new images, i.e. $\gamma^n_{\omega,h}$ for $n = 1, \ldots, N$ and from them, the sufficient statistics, $\bar{\pi}_\omega, \bar{\boldsymbol{\mu}}_\omega, \bar{\boldsymbol{\Sigma}}_\omega$ using Eqs. (8) and (10). In the Incremental M-step we then combine the sufficient statistics from these new images with the existing set of sufficient statistics from the previous M images. Then the overall sufficient statistics, $\hat{\pi}_\omega, \hat{\boldsymbol{\mu}}_\omega, \hat{\boldsymbol{\Sigma}}_\omega$ are computed:

$$\hat{\pi}_\omega = \frac{M\pi^e_\omega + N\bar{\pi}_\omega}{M+N} \tag{11}$$

$$\hat{\boldsymbol{\mu}}_\omega = \frac{M\boldsymbol{\mu}^e_\omega + N\bar{\boldsymbol{\mu}}_\omega}{M+N} \tag{12}$$

$$\hat{\boldsymbol{\Sigma}}_\omega = \frac{M\boldsymbol{\Sigma}^e_\omega + N\bar{\boldsymbol{\Sigma}}_\omega}{M+N} \tag{13}$$

From these we can then update the model hyper-parameters. Note the existing sufficient statistics are not updated within the update loop. When the model converges, the final value of the sufficient statistics from the new images are combined with the existing set, ready for the next update: $\pi^e_\omega = \hat{\pi}_\omega, \boldsymbol{\mu}^e_\omega = \hat{\boldsymbol{\mu}}_\omega, \boldsymbol{\Sigma}^e_\omega = \hat{\boldsymbol{\Sigma}}_\omega$. Initially $M = 0$, so $\pi^e_\omega, \boldsymbol{\mu}^e_\omega, \boldsymbol{\Sigma}^e_\omega$ drop from our equations and our model hyper-parameters are set randomly (within some sensible range).

4. Methods

4.1. 101 object categories

We test our Bayesian algorithms (Incremental and Batch) using 101 assorted object categories. The names of 101 categories were generated by flipping through the pages of the Webster Collegiate Dictionary [12], picking a subset of categories that were associated with a drawing. After we generated the list of category names, we used Google Image Search engine to collect as many images as possible for each category. Two graduate students not associated with the experiment then sorted through each category, mostly getting rid of irrelevant images (e.g. a zebra-patterned shirt for the "zebra" category). Fig. 9 shows examples of both the 101 foreground object categories as well as the background clutter category. 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, as the model parts are ordered by their x-coordinate, so have trouble with vertical structures. One could also avoid rotating the image by choosing the y-coordinate as ordering reference. This rotation is used for the sake of programming simplicity. At last, images were scaled roughly to around 300 pixels wide.

4.2. Feature detection

Feature points are found using the detector of Kadir and Brady [14]. 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 $\mathscr{X}$. 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 [8]. 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 $\mathscr{A}$. Note that the same set of parameters were used for feature detection for all 101 object categories. Figs. 4a and 5a show examples of feature detection on some training images.

4.3. Prior and initialization

One critical issue is the choice of priors for the Normal-Wishart distributions. In this paper, learning is performed using a single mixture component. The choice of prior is itself a topic worth full investigation. In order to keep the consistency and prove the concept, we use here a single prior distribution for all 101 object categories. Since prior knowledge should reflect some information of the real-world object categories, we estimated this prior distribution using well-learned maximum likelihood (ML) models of faces, spotted cats and airplanes in [8]. It is important to point out that the idea of using prior information from unrelated categories was proposed independently by Miller et al. [15]. The ML models are simply averaged together for each parameter to obtain a model distribution for the prior. Fig. 3a and b shows the prior shape and appearance model for each category.

Initial conditions are chosen in exactly the same way as [4]. Again, the same initialization is used for all 101 object categories.

4.4. Experimental setup for each category

Each experiment was carried out under identical conditions. For each category dataset, N training images are drawn randomly first. Then 50 testing images are randomly drawn from the remaining images in the dataset. For some dataset, less than 50 images are left after training images are drawn. In this case we use all the remaining ones as testing images. 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 at $0, 1, 3, 6, 15$, repeating the experiments 10 times for each value (using a randomly chosen N training images each time) to obtain a more robust estimate of performance. When $N = 0$, we use the prior model alone to perform object categorization without any training data. Only the Bayesian algorithm is used in this case. In addition, when $N = 1$, ML fails to converge, so we only show results for the Bayesian methods in this case.

When evaluating the models, the decision is a simple object present/absent one. Under these conditions, an algorithm performing at random has a 50% performance rate. All performance values are quoted as area under the receiver-operating characteristic (ROC). ROC curve is obtained by testing the model on 50 foreground test images and 50 background images. 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

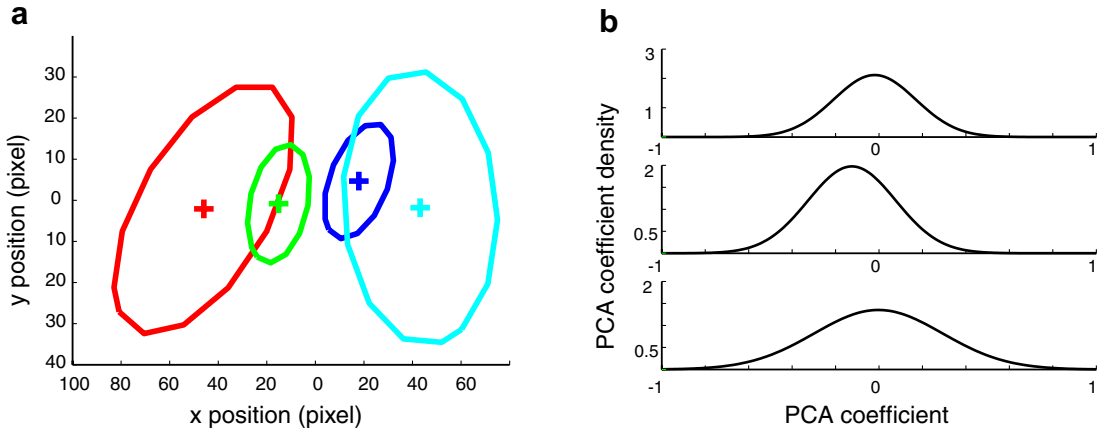
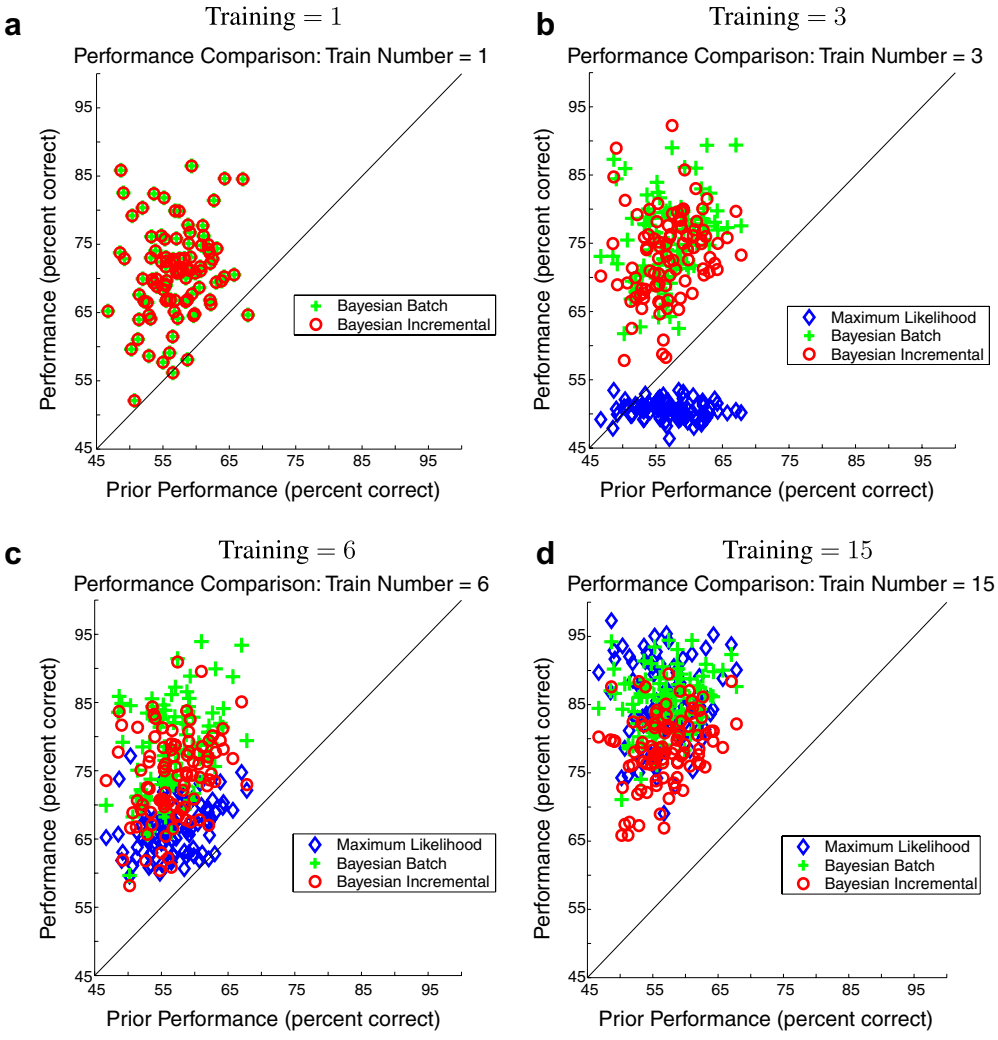


Fig. 3. (a and b) Prior distribution for shape mean ($\boldsymbol{\mu}^{\mathscr{X}}$) and appearance mean ($\boldsymbol{\mu}^{\mathscr{A}}$) for all the categories to be learned. Each prior's hyper-parameters are estimated from models learned with maximum likelihood methods, using "the other" datasets [8]. 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.



e Category list sorted by incremental perf. at Training Number = 15.

1. trilobite	18. bass	35. snoopy	52. gerenuk	69. cellphone	86. barrel
2. face	19. ketch	36. okapi	53. sunflower	70. gramophone	87. joshua-tree
3. pagoda	20. lobster	37. schooner	54. rhino	71. bonsai	88. pigeon
4. tick	21. pyramid	38. binocular	55. cougar-body	72. lotus	89. watch
5. inlineskate	22. rooster	39. motorbike	56. crab	73. cannon	90. dragonfly
6. metronome	23. laptop	40. hedgehog	57. ibis	74. wheel-chair	91. mayfly
7. accordion	24. waterlilly	41. garfield	58. helicopter	75. dolphin	92. cup
8. yinyang	25. wrench	42. airplane	59. dalmatian	76. stegosaurus	93. ewer
9. soccerball	26. strawberry	43. umbrella	60. scorpion	77. brain	94. octopus
10. spotted cat	27. starfish	44. panda	61. revolver	78. menorah	95. platypus
11. nautilus	28. ceilingfan	45. crocodile-head	62. beaver	79. chandelier	96. buddha
12. grand-piano	29. seahorse	46. llama	63. saxophone	80. camera	97. chair
13. crayfish	30. stapler	47. windsor-chair	64. kangaroo	81. ant	98. anchor
14. headphone	31. stop-sign	48. car-side	65. euphonium	82. scissors	99. mandolin
15. hawksbill	32. zebra	49. pizza	66. flamingo	83. butterfly	100. electric-guitar
16. ferry	33. brontosaurus	50. minaret	67. flamingo-head	84. wildcat	101. lamp
17. cougar-face	34. emu	51. dollarbill	68. elephant	85. crocodile	

Fig. 4. (a–d) Performance comparison between ML, Bayesian batch and Bayesian incremental methods for all 101 object categories at Training Number = (1, 3, 6, 15). In each plot, a category has three markers: red-circle represents Bayesian incremental method, green-plus Bayesian batch method and blue-diamond maximum likelihood method. For each panel, the x-axis indicates Bayesian method categorization performance with only the prior model. Note with one training image, the Bayesian incremental method and Batch method are exactly the same. The y-axis indicates categorization performance for each of the three methods. There is no maximum likelihood performance in the degenerate case of Training Number = 1. Moreover, Bayesian Incremental method and Batch method are also the same for Training Number = 1. (e) Category name list is sorted according to the Bayesian Incremental method performance at Training Number = 15.

image = 20. All parameters remain the same for learning different categories of objects.

5. Experimental results

Fig. 4 illustrates the recognition performance at Training Number = 1, 3, 6, 15 for the different algorithms: Bayesian incremental, Bayesian batch and maximum likelihood. The performance of each method is compared with the performance of the prior-model on the given category. Despite the rudimentary prior, a strong performance gain is observed for the majority of the 101 categories, even when only a few training examples are used. With one training example, the Bayesian method achieves an average performance of 71%, with the best results being over 80%. At Training Number = 3, maximum likelihood has a performance at chance while both Bayesian methods achieve a performance near 75%. At Training Number = 15 that the maximum likelihood catches up the recognition performance with the Bayesian methods. At Training Number = 15, the average Bayesian incremental method is not as reliable as the Bayesian batch method. In general the incremental method is much more sensitive to the quality of the training images and to the order in with they are presented to the algorithm. Therefore it is more likely to form suboptimal models.

While performance is a key measurement for recognition algorithms, efficiency in training and learning are also important. One big advantage that we have gained from the Bayesian Incremental method is its fast speed in training. Fig. 5 shows the comparison of average learning time across all 101 categories between these three methods. All methods show approximately linear increase in learning time as the number of training images increases. The incremental method, particularly, shows a very small slope. In our Matlab implementation, it takes approximately 6 s per image to train in Bayesian incremental method, while the batch and maximum likelihood methods take 6 times as long.

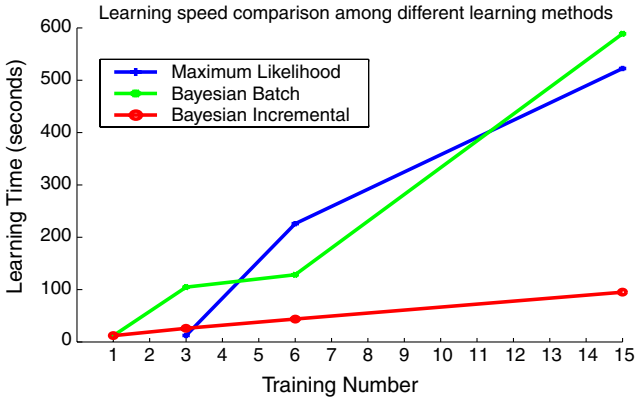


Fig. 5. Average learning time for ML, Bayesian batch and Bayesian incremental methods over all 101 categories.

In Figs. 6 and 7 we show in details the results from the grand-piano and cougar-face categories, both of which performed well (ROC Areas 16% and 15%, respectively, for Training Number = 15). As the number of training examples increases, we observe that the shape model is more defined and structured with reducing variance. This is expected since the algorithm should be more and more confident of what is to be learned. Figs. 6c and 7c shows examples of the part appearance that are closest to the mean distribution of the appearance. Notice that critical features such as keyboards for the piano and eyes or whiskers for the cougar-face are successfully learned by the algorithm. Three learning methods' performances are compared in Figs. 6d and 7d. The Bayesian methods clearly show a big advantage over the ML method when training number is small. Bayesian Incremental, however, shows more greater performance fluctuations as compared to the Bayesian Batch method. Finally, we show some classified test images, using an incremental model trained from a single image.

It is also useful to look at the other end of the performance spectrum—those categories that have low recognition performance. We give some insights 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. 8, 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. 8a shows that a variety of viewpoints is 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 simplified version Burl, Weber and Fergus' constellation model [5,7,8] as it ignores the possibility of occluded parts.

A great source of improvement could potentially come from prior model information. The prior model is currently rather weak and improvements in this area would undoubtedly improve the models performance. However, the performance could be degraded if the model was to incorporate misleading information—as illustrated in Fig. 8b. Our choice of prior for this paper is kept as simple as possible to facilitate the experiments. We expect further exploration into this topic can help improving recognition performances greatly.

6. Summary and discussion

We presented a Bayesian incremental algorithm for learning generative models of object categories from a very limited training set. Our work is an extension of Fei-Fei et al.'s Bayesian batch method [4]. We have tested both methods using a prior derived from three unrelated categories, alongside a simplified version of Fergus et al.'s maximum likelihood method, on a large dataset of 101 object categories.

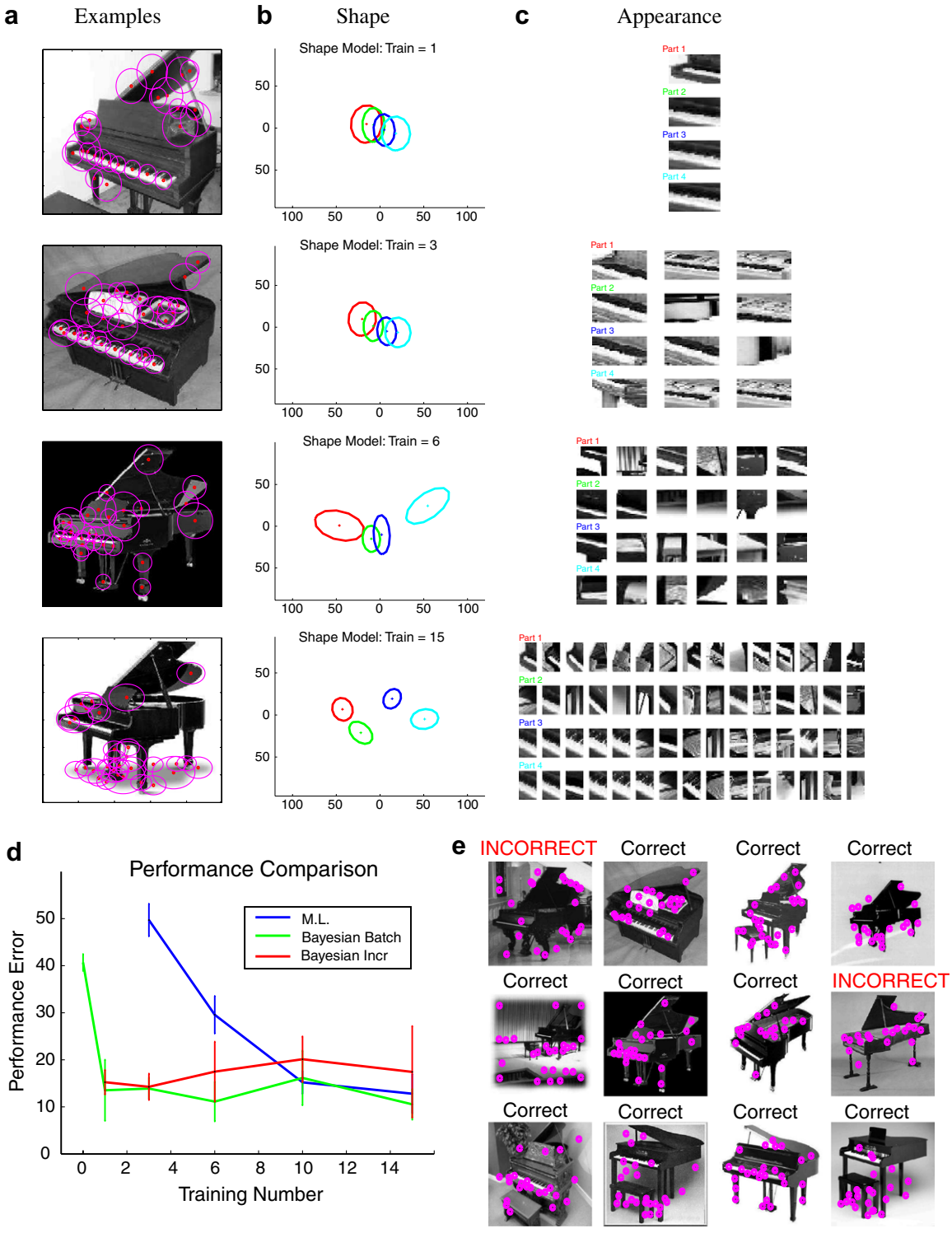


Fig. 6. Results for the "grand-piano" category. (a) Examples of feature detection. (b) The shape models learned at Training Number = (1, 3, 6, 15). Similarly to Fig. 3a, the x-axis represents the x position, measured by pixels, and the y-axis represents the y position, measured by pixels. (c) The appearance patches for the model learned at Training Number = (1, 3, 6, 15). (d) The comparative results between ML, Bayesian batch and Bayesian incremental methods (the error bars show the variation over the 10 runs). (e) Recognition result for the incremental method at Training Number = 1. Pink dots indicate the center of detected interest points.

First of all, it is possible to train complex models of object categories with a handful of images. It is clear that Bayesian methods allow category learning from small training sets. On one, three and six training examples the maximum likelihood method is unable to discriminate any category from images containing random clutter. By

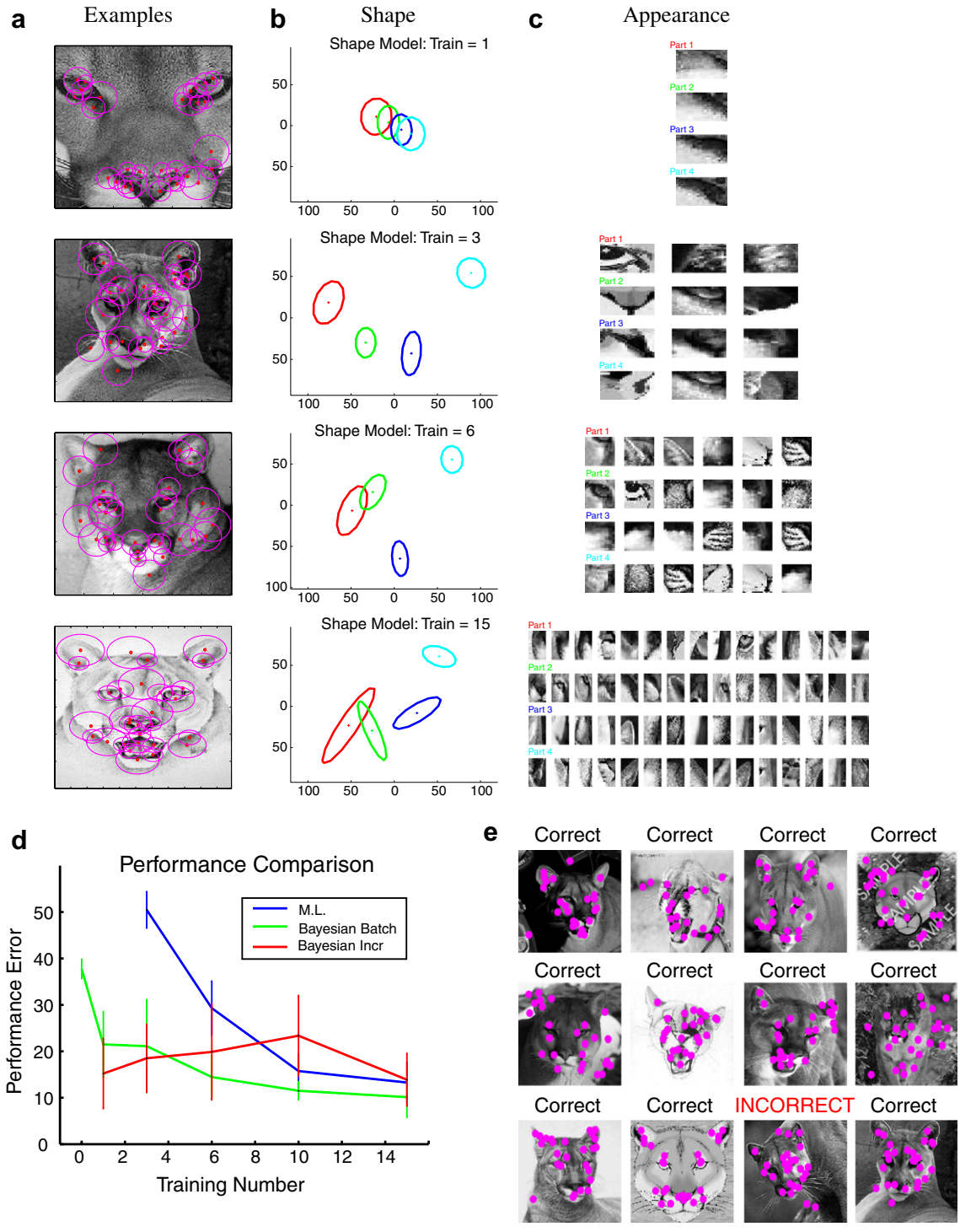


Fig. 7. Results for the "cougar face" category.

contrast the Bayesian methods achieve better than chance performance on 90 higher than 80.

Second, the desirable incremental learning feature leads to much faster learning speed (Fig. 3), but is paid with worse recognition performance for larger training set sizes. We conjecture that this is due to the fact that less information is carried along by the incremental algorithm from one learning epoch to the next, while the batch algorithm has all training images available at the same time thus allowing, for example, to test a larger number of hypotheses on how foreground and clutter features should be assigned in each training image.

Third, the maximum likelihood method matches the performance of the Bayesian methods when the training set reaches size 15. This is surprising, given that the number of parameters in each model is 50, and therefore a few

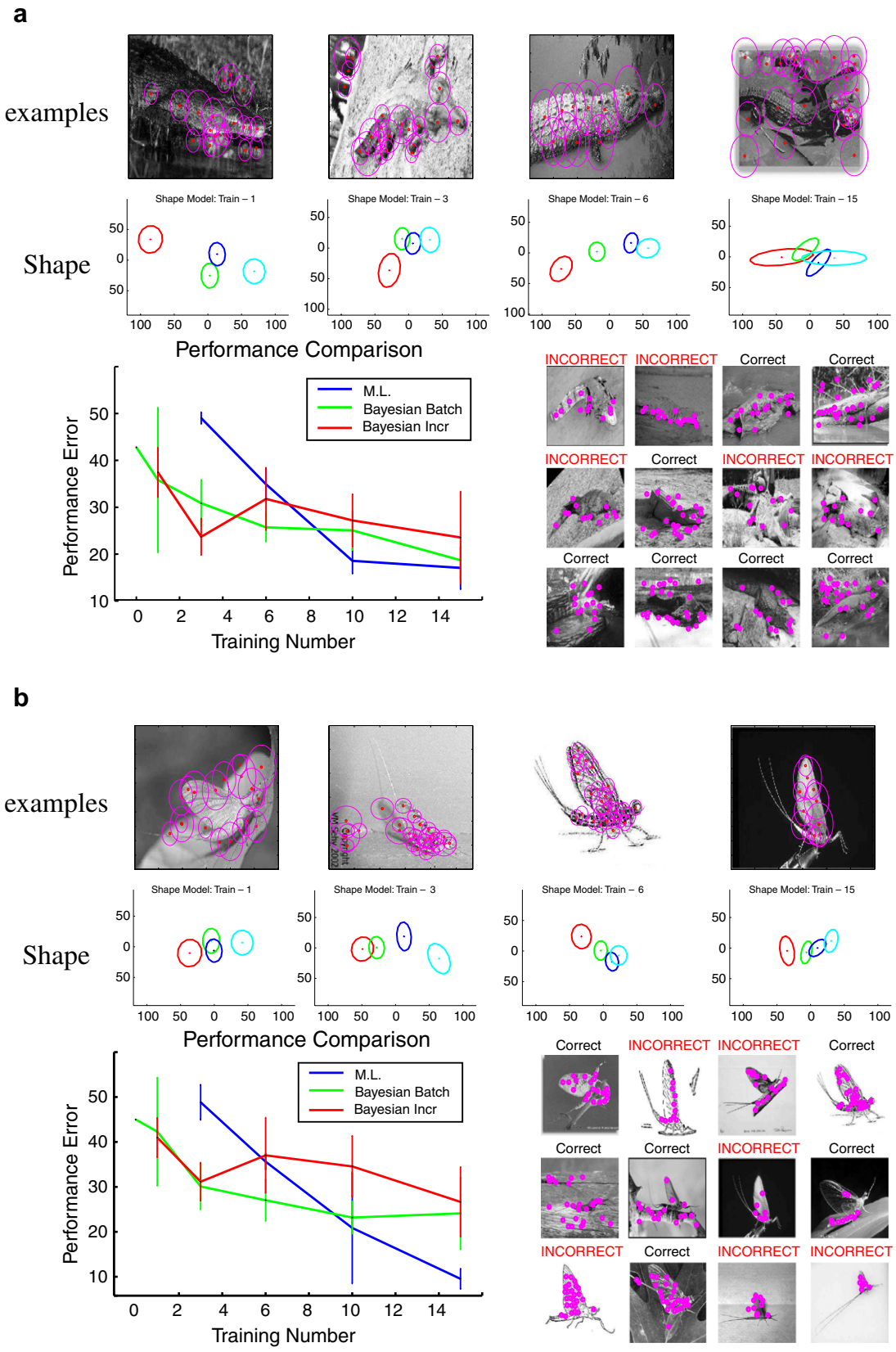


Fig. 8. Two categories with unsatisfactory performance. (a) Crocodile (ROC area = 26%). (b) Mayfly. (ROC area = 27%).

Fig. 9. 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 experimentation, collected by operators not associated with the experiment, and no category was excluded from the experiments. The last row 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 gray-scale information is used in our system. Complete datasets can be found at http://vision.caltech.edu.

hundred training examples are in principle required by a maximum likelihood method—one might have expected the Bayesian methods to be bettered by ML only around 100 training examples. The most likely reason for this result is that the prior that we employ is too simple. Bayesian methods live and die by the quality of 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 we would need many more categories to train a reasonable prior.

Fourth, the good news is that the problem of recognizing automatically hundreds, perhaps thousands, of object categories does not belong to a hopelessly far future. We hope that the success of our method 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 datasets.

Acknowledgment

The authors thank Marc'Aurelio Ranzato and Marco Andreetto for their help in image collection.

References

[1] I. Biederman, Recognition-by-components: a theory of human image understanding, Psychol. Rev. 94 (1987) 115–147.
[2] Y. Amit, D. Geman, A computational model for visual selection, Neural Comput. 11 (7) (1999) 1691–1715.
[3] H. Schneiderman, T. Kanade, A statistical approach to 3D object detection applied to faces and cars, in: Proc. Conf. on Computer Vision Pattern and Recognition, 2000, pp. 746–751.
[4] L. Fei-Fei, R. Fergus, P. Perona, A Bayesian approach to unsupervised learning of object categories, in: Proc. Int. Conf. on Computer Vision, 2003, pp. 1134–1141.
[5] M. Weber, M. Welling, P. Perona, Unsupervised learning of models for recognition, in: Proc. 6th Europ. Conf. on Computer Vision, vol. 2, 2000, pp. 101–108.
[6] P. Viola, M. Jones, Rapid object detection using a boosted cascade of simple features, in: Proc. Conf. on Computer Vision Pattern and Recognition, vol. 1, 2001, pp. 511–518.
[7] M.C. Burl, M. Weber, P. Perona, A probabilistic approach to object recognition using local photometry and global geometry, in: Proc. Europ. Conf. on Computer Vision, pp. 628–641.
[8] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: Proc. Conf. on Computer Vision Pattern and Recognition, vol. 2, 2003, pp. 264–271.
[9] H. Attias, Inferring parameters and structure of latent variable models by variational bayes, in: 15th Conf. on Uncertainty in Artificial Intelligence, 1999, pp. 21–30.
[10] M.I. Jordan, Z. Ghahramani, T.S. Jaakkola, L.K. Saul, An introduction to variational methods for graphical models, Mach. Learn. 37 (1999) 183–233.
[11] S.R. Waterhouse, D.J.C. MacKay, A.J. Robinson, Bayesian Methods for Mixtures of Experts, NIPS, 1996.
[12] Merriam-Webster's collegiate dictionary, 10th ed. Springfield, Massachusetts, USA, 1994.
[13] W.D. Penny, Variational Bayes for d-dimensional Gaussian mixture models, Tech. Rep., University College London, 2001.
[14] T. Kadir, M. Brady, Scale, saliency and image description, Int. J. Comp. Vis. 45 (2) (2001) 83–105.
[15] E. Miller, N. Matsakis, P. Viola, Learning from one example through shared densities on transforms, in: Proc. Conf. on Computer Vision Pattern and Recognition, 2000, pp. 464–471.
[16] R.M. Neal, G.E. Hinton, A view of the EM algorithm that justifies incremental, sparse and other variants, in: M.I. Jordan (Ed.), Learning in Graphical Models, Kluwer academic press, Norwell, 1998, pp. 355–368.
[17] D. Lowe, Object recognition from local scale-invariant features, in: Proc. Int. Conf. on Computer Vision, 1999, pp. 1150–1157.
[18] C. Schmid, R. Mohr, Local Greyvalue Invariants for Image Retrieval, IEEE Trans. Pattern Anal. Machine Intell. 19 (5) (1997) 530–534.