International Journal of Computer Vision 71(3), 273–303, 2007

DOI: 10.1007/s11263-006-8707-x

Weakly Supervised Scale-Invariant Learning of Models for Visual Recognition

R. FERGUS
Department of Engineering Science, University of Oxford, Parks Road, Oxford OX1 3PJ, U.K.
fergus@robots.ox.ac.uk, fergus@csail.mit.edu

P. PERONA
Department of Electrical Engineering, California Institute of Technology, MC 136-93, Pasadena, CA 91125, U.S.A.
perona@caltech.edu

A. ZISSERMAN
Department of Engineering Science, University of Oxford, Parks Road, Oxford, OX1 3PJ, U.K.
az@robots.ox.ac.uk

First online version published in July, 2006

Abstract. We investigate a method for learning object categories in a weakly supervised manner. Given a set of images known to contain the target category from a similar viewpoint, learning is translation and scale-invariant; does not require alignment or correspondence between the training images, and is robust to clutter and occlusion. Category models are probabilistic constellations of parts, and their parameters are estimated by maximizing the likelihood of the training data. The appearance of the parts, as well as their mutual position, relative scale and probability of detection are explicitly described in the model. Recognition takes place in two stages. First, a feature-finder identifies promising locations for the model's parts. Second, the category model is used to compare the likelihood that the observed features are generated by the category model, or are generated by background clutter. The flexible nature of the model is demonstrated by results over six diverse object categories including geometrically constrained categories (e.g. faces, cars) and flexible objects (such as animals).

Keywords: object recognition, parts and structure model, constellation model, semi-supervised learning

1. Introduction

Representation, detection and learning are the main issues that need to be tackled in designing a visual system for recognizing object categories. The first challenge is coming up with models that can capture the 'essence' of a category, i.e. what is common to the objects that belong to it, and yet are flexible enough to accommodate object variability (e.g. presence/absence of distinctive parts such as mustache and glasses, variability in overall shape, changing appearance due to lighting conditions, viewpoint etc.). The challenge of detection is defining metrics and inventing algorithms that are suitable for matching models to images efficiently in the presence of occlusion and clutter. Learning is the ultimate challenge. If we wish to be able to design visual systems that can recognize, say, 10,000 object categories, then effortless learning is a crucial

step. This means that those training steps that require a human operator (e.g. collection of good quality training exemplars of the category; elimination of clutter; correspondence and scale normalization of the training examples) should be reduced to a minimum, or eliminated altogether.

In this paper we develop and discuss a probabilistic model for an object category which can be learnt from a set of training images of instances of that category, requiring only weak supervision. The model represents a single visual aspect of the object category (e.g. the side view of a car) and accommodates intra-category variability. The training images are required to contain a single instance of the category with a common visual aspect and orientation. These requirements are the extent of the weak supervision. The instances do not need to be aligned (e.g. centred) or scale normalized or put in correspondence, and the images may contain clutter (i.e. foreground segmentation is not required).

Attempts at object recognition date back to the origin of the computer vision field. However, for the most part the emphasis has been on efficiently recognizing single 2D or 3D object instances (e.g. a particular stapler) (Lowe, 1985) rather than an object category (all types of staplers) under unrestricted viewpoints. A number of successful approaches—geometric alignment, geometric hashing, appearance manifolds etc.—have been developed with objects being represented by their wireframe outline or internal appearance. Much of the emphasis in these earlier attempts was on viewpoint mappings ranging from 2D transformations (e.g. affinities), through parallel projection (affine cameras) to the full generality of perspective projection. This progress is covered in text books such as Forsyth and Ponce (2002).

However, the emphasis in this paper is not on viewpoint invariance, indeed learning is restricted to scale and translation invariance, but on modelling and learning intra-category variability. A number of recent papers have also tackled this problem. A key issue is how to represent an object category. One popular approach is to model categories as a collection of features, or parts, each part having a distinctive appearance and (in most cases) spatial position (Agarwal and Roth, 2002; Amit and Geman, 1999; Borenstein and Ullman, 2002; Burl et al., 1998; Csurka et al., 2004; Felzenszwalb and Huttenlocher, 2000; Heisele et al., 2002; Jurie and Schmid, 2004; Leibe et al., 2004; Schmid, 2001; Schneiderman and Kanade, 2000; Thureson and Carlsson, 2004; Torralba et al., 2004; Weber et al., 2000c). Different authors vary widely on the details: the number of parts they envisage (from a few to thousands of parts), how these parts are detected and represented, how their position is represented, whether the variability in part appearance and position is represented explicitly or is implicit in the details of the matching algorithm. The issue of learning is perhaps the least well understood. Most authors rely on manual steps to eliminate background clutter and normalize the pose of the training examples. Recognition can also proceed by an exhaustive search over image position and scale (LeCun et al., 2004; Rowley et al., 1998; Schneiderman and Kanade, 2000; Sung and Poggio, 1998; Torralba et al., 2004; Viola and Jones, 2001).

We focus our attention on the probabilistic approach proposed by Burl et al. (1998) which models objects as random constellations of parts. This approach presents several advantages: the model explicitly accounts for shape variations and for the randomness in the presence/absence of features due to occlusion and detector errors. It accounts explicitly for image clutter. It yields principled and efficient detection methods. Weber et al. (2000b,c) proposed a maximum likelihood weakly supervised learning algorithm for the "constellation model" which successfully learns object categories in a translation invariant manner from cluttered data with minimal human intervention. We propose here four substantial improvements to the constellation model and to its maximum likelihood learning algorithm. First, while Burl et al. and Weber et al. model explicitly shape variability, they do not model the variability of appearance. We extend their model to take this aspect into account. Second, appearance here is learnt simultaneously with shape, whereas in their work the appearance of a part is fixed before shape learning. Third, they use correlation to detect their parts. We substitute their front end with an interest operator, which detects regions and their scale in the manner of Lindeberg (1998) and Mikolajczyk and Schmid (2001). Fourth, Weber et al. did not experiment extensively with scale-invariant learning, most of their training sets are collected in such a way that the scale is approximately normalized. We extend their learning algorithm so that new object categories may be learnt efficiently, without supervision, from training sets where the object examples have large variability in scale. An additional contribution is experimenting with a number of new image datasets to validate the overall approach over several object categories. Examples images from these datasets are shown

in Fig. 1.

The aim of this paper is to describe our probabilistic object model and learning algorithm in sufficient detail to make implementation possible, as well as giving an insight into its design. In Section 2 we give the structure of the model and describe our region detector. In Section 3 we show how to estimate the parameters of our model, given a set of training images. Section 4 describes the use of the model in recognition. Our approach is then tested on a wide variety of data in Section 5. Experiments investigating our algorithm's operation are also performed, including the sensitivity of parameter settings and the importance of different components within the model. Finally, conclusions are drawn in Section 6.

2. Model Structure

Our approach to modeling object categories follows on from the work of Burl et al. (1998), Weber (2000), Weber et al. (2000b), Weber et al. (2000c). An object model consists of a number of parts. Each part has an appearance, relative scale and can be occluded or not. Each part has a certain probability of being erroneously detected in the background clutter. Shape is represented by the mutual position of the parts. The entire model is generative and probabilistic, so appearance, scale, shape and occlusion are all modeled by probability density functions, which here are Gaussians or multinomials. The model is scale and translation invariant in both learning and recognition. The process of learning an object category is one of first detecting regions and their scales, and then estimating the parameters of the above densities from these regions, such that the model gives a maximum-likelihood description of the training data. Recognition is performed on a query image by again first detecting regions and their scales, and then evaluating the regions using the model parameters estimated in the learning. Note that parts refer to the model, while features refer to detections in the image.

The model is best explained by first considering recognition. Assume, we have learnt a generative object category model, with P parts and parameters θ_{fg}. We also assume that all non-object images can be modeled by a background with a single, fixed, set of parameters θ_{bg}. We are then presented with a new image and we must decide if it contains an instance of our object category or not. In this query image we have identified N interesting features with locations $\mathbf{X}$, scales $\mathbf{S}$, and appearances $\mathbf{A}$. We now make a decision as to the presence/absence of the object by comparing the ratio of category posterior densities, R, to a threshold T:

$$\begin{aligned} R &= \frac{p(\text{Object} \mid \mathbf{X}, \mathbf{S}, \mathbf{A})}{p(\text{No object} \mid \mathbf{X}, \mathbf{S}, \mathbf{A})} \\ &= \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{A} \mid \text{Object})\ p(\text{Object})}{p(\mathbf{X}, \mathbf{S}, \mathbf{A} \mid \text{No object})\ p(\text{No object})} \\ &\approx \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{A} \mid \theta_{fg})\ p(\text{Object})}{p(\mathbf{X}, \mathbf{S}, \mathbf{A} \mid \theta_{bg})\ p(\text{No object})} \end{aligned} \tag{1}$$

The last expression is an approximation since we represent the category with its (imperfect) model, parameterized by θ. The ratio of the priors may be estimated from the training set or set by hand (usually to 1).

Since our model only has P (typically 3–7) parts but there are N (typically 10 to 30) features in the image, we use an indexing variable $\mathbf{h}$ (as introduced in Burl et al. (1998)) which we call a *hypothesis*. $\mathbf{h}$ is a vector of length P, where each entry is between 0 and N which allocates a particular feature to a model part. No feature is allowed to belong to more than one part. The unallocated features are assumed to be part of the background, with 0 indicating the part is unavailable (e.g. because of occlusion). The set H is all valid allocations of features to the parts; consequently $|H|$ is $O(N^P)$. Computing R in (1) requires the calculation of the ratio of the two likelihood functions. In order to do this, the likelihoods are factored as follows:

$$\begin{aligned} &p(\mathbf{X}, \mathbf{S}, \mathbf{A} \mid \theta_{fg}) \\ &= \sum_{\mathbf{h} \in H} p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h} \mid \theta_{fg}) \\ &= \sum_{\mathbf{h} \in H} \underbrace{p(\mathbf{A} \mid \mathbf{X}, \mathbf{S}, \mathbf{h}, \theta_{fg})}_{\text{Appearance}} \\ &\quad \times \underbrace{p(\mathbf{X} \mid \mathbf{S}, \mathbf{h}, \theta_{fg})}_{\text{Shape}} \underbrace{p(\mathbf{S} \mid \mathbf{h}, \theta_{fg})}_{\text{Rel. Scale}} \underbrace{p(\mathbf{h} \mid \theta_{fg})}_{\text{Other}} \end{aligned} \tag{2}$$

We now look at each of the likelihood terms and derive their actual form. The likelihood terms model not only the properties of the features assigned to the models parts (the foreground) but also the statistics of features in the background of the image (those not picked out by the hypothesis). Therefore it will be helpful to define the following notation: $\mathbf{d} = sign(\mathbf{h})$ (which is a binary vector giving the state of occlusion for each part, i.e. $d_p = 1$ if part p is present and $d_p = 0$ if absent), $n_{fg} = sum(\mathbf{d})$ (the number of foreground features under the current hypothesis) and $n_{bg} = N - n_{fg}$ (the number of background features).

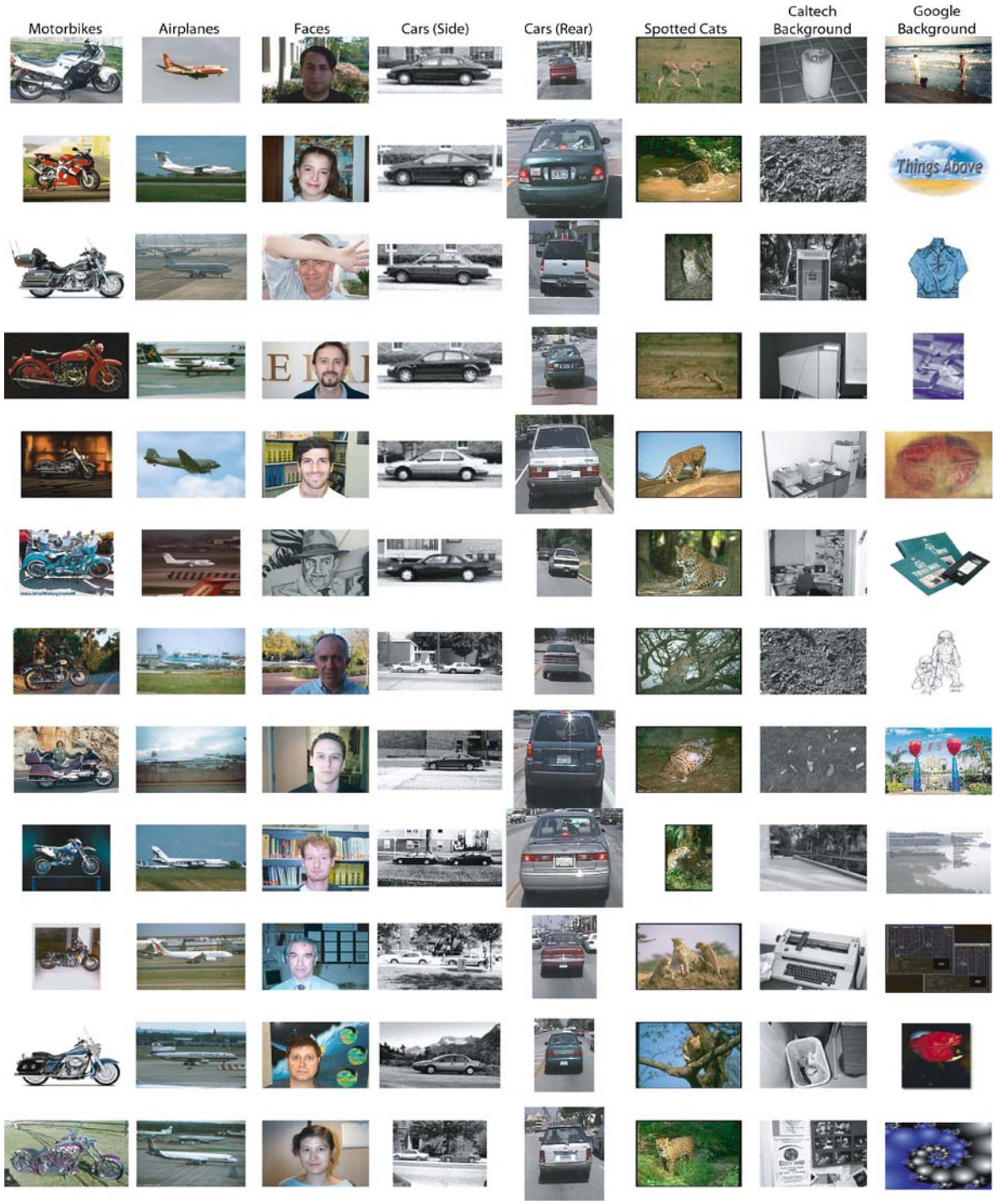


Figure 1. Some sample images from the datasets. Note the large variation in scale in, for example, the cars (rear) database. These datasets are from both http://www.vision. caltech.edu/html-files/archive.html and http://www.robots. ox.ac.uk∼vgg/data/, except for the Cars (Side) from (http://l2r.cs.uiuc.edu/∼cogcomp/ index_research.html) and Spotted Cats from the Corel Image library. A Powerpoint presentation of the figures in this paper can be found at http://www.robots.ox.ac.uk/∼vgg/presentations.html.

If no object is present, then all features in the image belong to the background. Thus we only have one possible hypothesis: $\mathbf{h}_0 = \mathbf{0}$, the null hypothesis. The likelihood in this case is:

$$\begin{aligned} &p(\mathbf{X}, \mathbf{S}, \mathbf{A} \mid \theta_{bg}) \\ &\quad = p(\mathbf{A} \mid \mathbf{X}, \mathbf{S}, \mathbf{h}_0, \theta_{bg}) \\ &\qquad \times p(\mathbf{X} \mid \mathbf{S}, \mathbf{h}_0, \theta_{bg}) p(\mathbf{S} \mid \mathbf{h}_0, \theta_{bg}) p(\mathbf{h}_0 \mid \theta_{bg}) \end{aligned} \quad (3)$$

As we will see below, $p(\mathbf{X}, \mathbf{S}, \mathbf{A} \mid \theta_{bg})$ is a constant for a given image. This simplifies the computation of the likelihood ratio in (1), since $p(\mathbf{X}, \mathbf{S}, \mathbf{A} \mid \theta_{bg})$ can be moved inside the summation over all hypotheses in (2), to cancel with the foreground terms.

2.1. Appearance

Here we describe the form of $p(\mathbf{A} \mid \mathbf{X}, \mathbf{S}, \mathbf{h}, \theta)$ which is the appearance term of the object likelihood. We can simplify the expression to $p(\mathbf{A} \mid \mathbf{h}, \theta)$ if, given the detected features, we assume their appearance and location to be independent. Each feature's appearance is represented as a point in an appearance space, defined below. Each part p has a Gaussian density within this space, with mean and covariance parameters $\theta^{app}_{fg,p} = \{\mathbf{c}_p, V_p\}$ which is independent of other parts' densities. The background model has fixed parameters $\theta^{app}_{bg} = \{\mathbf{c}_{bg}, V_{bg}\}$. Both V_p and V_{bg} are assumed to be diagonal. The appearance density is computed over all features: each feature selected by the hypothesis is evaluated under the appropriate part density while all features not selected by the hypothesis are evaluated under the background density:

$$\begin{aligned} p(\mathbf{A} \mid \mathbf{h}, \theta_{fg}) = &\prod_{p=1}^{P} \mathrm{G}(\mathbf{A}(h_p) \mid \mathbf{c}_p, V_p)^{d_p} \\ &\times \prod_{j=1, j \setminus \mathbf{h}}^{N} \mathrm{G}(\mathbf{A}(j) \mid \mathbf{c}_{bg}, V_{bg}) \end{aligned} \quad (4)$$

where G is the Gaussian distribution. $\mathbf{A}(h_p)$ is the appearance of the feature picked by h_p. If no object is present, then all features are evaluated under the background density:

$$p(\mathbf{A} \mid \mathbf{h}_0, \theta_{bg}) = \prod_{j=1}^{N} \mathrm{G}(\mathbf{A}(j) \mid \mathbf{c}_{bg}, V_{bg}) \quad (5)$$

As $p(\mathbf{A} \mid \mathbf{h}_0, \theta_{bg})$ is a constant and so is not dependent on $\mathbf{h}$, so we can cancel terms between (4) and (5) when computing the likelihood ratio in (1):

$$\frac{p(\mathbf{A} \mid \mathbf{h}, \theta_{fg})}{p(\mathbf{A} \mid \mathbf{h}, \theta_{bg})} = \prod_{p=1}^{P} \left(\frac{\mathrm{G}(\mathbf{A}(h_p) \mid \mathbf{c}_p, V_p)}{\mathrm{G}(\mathbf{A}(h_p) \mid \mathbf{c}_{bg}, V_{bg})} \right)^{d_p} \quad (6)$$

So the appearance of each feature in the hypothesis is evaluated under foreground and background densities and the ratio taken. If the part is occluded, the ratio is 1 ($d_p = 0$).

This term is an addition to the previous incarnations of the constellation model in Burl et al. (1998); Weber et al. (2000c).

2.2. Shape

Here we describe the form of $p(\mathbf{X} \mid \mathbf{S}, \mathbf{h}, \theta)$ which is the shape term of the object likelihood. The shape of the object is represented by a joint Gaussian density of the locations of features within a hypothesis, once they have been transformed into a scale and translation-invariant space. This representation allows the modeling of both inter and intra part variability: interactions between the parts (both attractive and repulsive) as well as uncertainty in location of the part itself.

Translation invariance is achieved by using the location of the feature assigned to the first non-occluded part as a landmark. We then model the shape of the remaining features in the hypothesis relative to this landmark feature. Scale invariance is achieved by using the scale of the landmark part to normalize the locations of the other features in the constellation. This approach avoids an exhaustive search over scale that other methods use. If the index of the first non-occluded part is l, then the landmark feature's location is $\mathbf{X}(h_l)$ and its scale is $S(h_l)$.

$\mathbf{X}(\mathbf{h})$ is a $2P$ vector holding the x and y coordinates of each feature in hypothesis h, i.e. $\mathbf{X}(\mathbf{h}) = \{x_{h_1}, \ldots, x_{h_P}, y_{h_1}, \ldots, y_{h_P}\}$. To obtain translation invariance, we subtract the location of the landmark from $\mathbf{X}(\mathbf{h})$: $\mathbf{X}^*(\mathbf{h}) = \{x_{h_1} - x_{h_l}, \ldots, x_{h_P} - x_{h_l}, y_{h_1} - y_{h_l}, \ldots, y_{h_P} - y_{h_l}\}$. A scale invariant representation is obtained by dividing through by $S(h_l)$: $\mathbf{X}^{**}(\mathbf{h}) = \frac{X^*(\mathbf{h})}{S(h_l)}$. Note that * indicates a representation that is translation invariant, while ** denotes a representation that is both scale and translation invariant.

We model $\mathbf{X}^{**}(\mathbf{h})$ with a Gaussian density which has parameters $\theta^{shape}_{fg} = \boldsymbol{\mu}, \Sigma\}$. Since any of the P parts can act as the landmark, $\boldsymbol{\mu}, \Sigma$ consist of a set of P $\boldsymbol{\mu}_l$'s and Σ_l's to evaluate $\mathbf{X}^{**}(\mathbf{h})$ with. However, the set members are approximately equivalent to one another. Changing landmark first involves a translation

of $\boldsymbol{\mu}$ and the equivalent transformation of Σ (a referral of variances between the old and new landmark), due to the properties of Gaussian distributions. Second, the translated constellation is normalised by the local scale of the landmark which, being a noisy measurement, results in perturbations to the scale and translation invariant representation. Due to translation invariance, where we eliminate the trivial term corresponding to the landmark, $\boldsymbol{\mu}_l$ is a $2(P-1)$ vector (x and y coordinates of the non-landmark parts). Correspondingly, Σ_l is a $2(P-1)$ by $2(P-1)$ matrix. Note that, unlike appearance whose covariance matrices V_p, V_{bg} are diagonal, Σ_l is a full matrix.

All features not included in the hypothesis are considered as arising from the background. The model for the background assumes features to be spread uniformly over the image (which has area α), with locations independent of the foreground locations. We also assume that the landmark feature can occur anywhere in the image, so its location is modeled by a uniform density of $1/\alpha$. Therefore:

$$p(\mathbf{X} \mid \mathbf{S}, \mathbf{h}, \theta_{fg}) = \left(\frac{1}{\alpha}\mathrm{G}(\mathbf{X}^{**}(\mathbf{h}) \mid \boldsymbol{\mu}_l, \Sigma_l)\right)\left(\frac{1}{\alpha}\right)^{n_{bg}} \quad (7)$$

If a part is occluded then we marginalize it out, which for a Gaussian entails deleting the corresponding dimensions from the mean and covariance matrix and adjusting the normalization constant. See Fergus (2005); Weber (2000) for more details.

If no object is present, then all detections are in the background and are consequently modeled by a uniform distribution:

$$p(\mathbf{X} \mid \mathbf{S}, \mathbf{h}_0, \theta_{bg}) = \left(\frac{1}{\alpha}\right)^{N} \quad (8)$$

Again, this is a constant, so we can cancel between (7) and (8) for the likelihood ratio in (1) to give:

$$\frac{p(\mathbf{X} \mid \mathbf{S}, \mathbf{h}, \theta_{fg})}{p(\mathbf{X} \mid \mathbf{S}, \mathbf{h}_0, \theta_{bg})} = \mathrm{G}(\mathbf{X}^{**}(\mathbf{h}) \mid \boldsymbol{\mu}_l, \Sigma_l)\, \alpha^{n_{fg}-1} \quad (9)$$

Note that if only one part is visible, then (9) is 1. This term is similar in nature to the shape term in Weber et al. (2000c), with the additional use of scale information from the features to obtain scale-invariance. Leung et al. (1998) proposed an affine invariant shape representation by the use of three model parts as a basis, transforming density of the remaining model parts into generalization of a Dryden-Mardia density Mardia and Dryden (1989). The complex nature of this density restricts its use to recognition, hence learning must be performed manually. A requirement of our weakly supervised learning scheme is that the transformed shape density is Gaussian, thus we are restricted to using one landmark only. In our case, a feature only gives location and scale (i.e. not orientation), therefore we are currently restricted to scale and translation invariance.

2.3. Relative Scale

Here we describe the form of $p(\mathbf{S} \mid \mathbf{h}, \theta)$ which is the relative scale term of the object likelihood. This term has the same structure as the shape term. The scale of parts relative to the scale of the landmark feature is modeled by a Gaussian density in log space which has parameters $\theta_{fg}^{\text{scale}} = \{\mathbf{t}, U\}$. Again, since the landmark feature could belong to any of the P parts, these parameters are really a set of equivalent $\mathbf{t}_l$, U_l's. The parts are assumed to be independent of one another, thus U_l is a diagonal $(P-1)$ by $(P-1)$ matrix, with $\mathbf{t}_l$ being a $(P-1)$ vector. The background model assumes a uniform distribution over scale (within a range r).

$$p(\mathbf{S} \mid \mathbf{h}, \theta_{fg}) = \left(\frac{1}{r}\mathrm{G}(\log \mathbf{S}^{*}(\mathbf{h}) \mid \mathbf{t}_l, U_l)\right)\left(\frac{1}{r}\right)^{n_{bg}} \quad (10)$$

If the object is not present, all detections are modeled by the uniform distribution:

$$p(\mathbf{S} \mid \mathbf{h}_0, \theta_{bg}) = \left(\frac{1}{r}\right)^{N} \quad (11)$$

Thus the ratio of likelihood becomes:

$$\frac{p(\mathbf{S} \mid \mathbf{h}, \theta_{fg})}{p(\mathbf{S} \mid \mathbf{h}_0, \theta_{bg})} = \mathrm{G}(\log \mathbf{S}^{*}(\mathbf{h}) | \mathbf{t}_l, U_l)\, r^{n_{fg}-1} \quad (12)$$

This term is an addition to the previous incarnations of the Constellation model of Burl et al. (1998); Weber et al. (2000c).

2.4. Occlusion and Statistics of the Feature Finder

$$p(\mathbf{h} \mid \theta_{fg}) = p_{Poiss}(n_{bg}|M)\, \frac{1}{{}^{n}C_r(N, n_{fg})}\, p(\mathbf{d} \mid \mathbf{D}) \quad (13)$$

$\mathbf{h}$ contains three types of information thus its distribution is a product of three terms. The first term

models the number of features in the background, using a Poisson distribution, which has a mean M. The second is a book-keeping term for the hypothesis variable: we are picking n_{fg} features from a total of N and since we have no bias toward particular features, all combinations are equally likely thus it is a constant for all $\mathbf{h}$.

$$\frac{p(\mathbf{X},\mathbf{S},\mathbf{A}\mid\theta_{fg})}{p(\mathbf{X},\mathbf{S},\mathbf{A}\mid\theta_{bg})} = \sum_{\mathbf{h}\in H}\frac{p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}\mid\theta_{fg})}{p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}_0\mid\theta_{bg})}$$

$$= \sum_{\mathbf{h}\in H}\prod_{p=1}^{P}\left(\frac{\mathrm{G}(\mathbf{A}(h_p)\mid\mathbf{c}_p,V_p)}{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_{bg},V_{bg})}\right)^{d_p}\frac{\mathrm{G}(\mathbf{X}^{**}(\mathbf{h})\mid\boldsymbol{\mu}_l,\Sigma_l)\ \mathrm{G}(\log\mathbf{S}^{*}(\mathbf{h})|\mathbf{t}_l,U_l)\,(\alpha r)^{n_{fg}-1}\ p_{Poiss}(n_{bg}\mid M)p(\mathbf{d}|\mathbf{D})}{p_{Poiss}(N|M)\ {}^{n}C_r(N,n_{fg})} \tag{16}$$

The last term is a joint distribution on the occlusions of model parts. It is a multinomial density (of size 2^P), modeling all possible occlusion patterns $\mathbf{d}$, having a parameter, $\mathbf{D}$. This joint distribution allows the modeling of correlations in occlusion: nearby parts are more often occluded together than far apart things. In the null case, we only have only possible hypothesis, $\mathbf{h}_0$, so the only term from (13) that remains is the Poisson which now has to account for all features belonging to the background:

$$p(\mathbf{h}_0|\theta_{bg}) = p_{Poiss}(N|M) \tag{14}$$

Thus the ratio becomes:

$$\frac{p(\mathbf{h}\mid\theta_{fg})}{p(\mathbf{h}\mid\theta_{bg})} = \frac{p_{Poiss}(n_{bg}|M)}{p_{Poiss}(N\mid M)}\ \frac{1}{{}^{n}C_r(N,n_{fg})}\ p(\mathbf{d}\mid\mathbf{D}) \tag{15}$$

These terms were introduced by Weber et al. (2000c).

2.5. Model Structure Summary

The model encompasses many of the properties of an object, all in a probabilistic way, so this model can represent both geometrically constrained objects (where the shape density would have a small covariance) and objects with distinctive appearance but lacking geometric form (the appearance densities would be tight, but the shape density would now be looser). Some additional assumptions inherent in our chosen model structure include: given a set of detected features, their appearance and location are independent; the foreground features' appearances are independent to one another; the background features' are independent to the foreground and each other.

An important limitation of the model, as presented, is that we can only model one aspect of the object. While this is a major limitation, many objects often appear in a distinctive pose (e.g. faces from the front) thus single aspect recognition is still a worthwhile problem. More importantly, our approach can be extended to multiple aspects by using a mixture of constellation models, in the manner of Weber et al. (2000).

Using (6), (9), (12) and (15) we can write the likelihood ratio from (1) as:

The intuition is that the majority of the hypotheses will be low scoring as they will be picking up features from background clutter on the image but hopefully a few features will genuinely be part of the object and hypotheses using these will score highly. However, we must be able to locate features over many different instances of the object and over a range of scales in order for this approach to work.

2.6. Feature Detection

Features are found using the detector of Kadir and Brady (2001).[1] This method finds regions that are salient over both location and scale. For each point in the image a histogram $P(I)$ is made of the intensities in a circular region of radius (scale) s. The entropy $H(s)$ of this histogram is then calculated and the local maxima of $H(s)$ are candidate scales for the region. The saliency of each of these candidates is measured by $H\frac{dP}{ds}$ (with appropriate normalization for scale (Kadir and Brady, 2001; Lindeberg, 1998)).

This gives a 3-D saliency map (over x, y and s). Regions of high saliency are clustered over both location and scale, with a bias toward clusters of large scale, since they tend to be more stable between object instances. The centroids of the clusters then provide the features for learning and recognition, their coordinates within the saliency map defining the centre and radius of each feature.

A good example illustrating the saliency principle is that of a bright circle on a dark background. If the scale is too small then only the white circle is seen, and there is no extremum in entropy. There is an entropy extremum when the scale is slightly larger than the radius of the bright circle, and thereafter the entropy decreases as the scale increases.

In practice this method gives stable identification of features over a variety of sizes and copes well with intra-category variability. The saliency measure is designed to be invariant to scaling, although experimental tests show that this is not entirely the case due to aliasing and other effects. Note, only monochrome information is used to detect and represent features. The performance of the algorithm is dependent on finding good features from which to learn a model. The effect of different feature detector settings is investigated in Section 5.

2.7. Feature Representation

The feature detector identifies regions of interest in each image. The coordinates of the centre give us $\mathbf{X}$ and the size of the region gives $\mathbf{S}$. Figure 2 illustrates this on six typical images from the motorbike dataset.

Once the regions are identified, they are cropped from the image and rescaled to the size of a small $k \times k$ patch (typically $11 \leq k \leq 21$ pixels). Thus, each patch exists in a k^2 dimensional space. Since the appearance densities of the model must also exist in this space, we must somehow reduce the dimensionality of each patch whilst retaining its distinctiveness, since a $100 \pm$ dimensional Gaussian is unmanageable from a numerical point of view and also the number of parameters involved ($2k^2$ per model part) are too many to be estimated.

This is done by using principal component analysis (PCA). We utilise three variants:

1. Intensity based PCA. The $k \times k$ patches are normalised to have zero mean and unit variance. This is to remove the effects of lighting variation. They are then projected into a fixed PCA basis in the intensity space of $k \times k$ patches, having l basis vectors. As used in Fergus et al. (2003).
2. Gradient based PCA. Inspired by the performance of PCA-SIFT in region matching Ke and Sukthankar (2004), we take the x and y gradients of the $k \times k$ patch. The derivatives are computed by symmetric finite difference (cropping to avoid edge effects). The magnitude of the gradients within the patch is then normalised to be 1, removing lighting variations. Note that we do not perform any orientation normalization as in Ke and Sukthankar (2004). The outcome is a vector of length $2k^2$, with the first k elements representing the x derivative, and the second k the y derivatives. The normalized gradient-patch is then projected into a fixed PCA basis of l dimensions.
3. Gradient based PCA with energy and residual. As for 2 but with two additional measurements made for each gradient-patch: its unnormalized energy and the residual between the reconstructed gradient-patch using the PCA basis and the original gradient-patch. Each region is thus represented by a vector of length $l + 2$. The last two dimensions act as a crude interest measure of the region, while the remaining dimensions actually represent its appearance.

Thus the appearance of each region is represented by a vector of PCA coefficients of length l or $l + 2$. Combining the vectors from all regions we obtain $\mathbf{A}$ for an image.

The PCA basis is computed from patches extracted using all Kadir and Brady regions found on all the training images of Motorbikes; Faces; Airplanes; Cars (Rear); Leopards and Caltech background. Note that this basis is used for all object categories. We assume that the covariance terms between components will be zero, thus V_p (the covariance of a part's appearance) is diagonal in nature. Alternative representations such as ICA and Fisher's linear discriminant were also tried, but in experiments they were shown to be inferior.

We have now computed $\mathbf{X}$, $\mathbf{S}$, and $\mathbf{A}$ for use in learning or recognition. For a typical image, this takes 10–15 seconds (all timings given are for a 2 Ghz machine), mainly due to the unoptimized feature detector. Optimization should reduce this to a few seconds.

3. Learning

In a weakly supervised learning scenario, one is presented with a collection of images containing examples of objects belonging to a given class amongst clutter. However the position and scale of the object with each image is unknown; no correspondence between exemplars is given; parts of the object may be missing or occluded. The challenge is to make sense of this profusion of data. Weber et al. (2000c) and Weber (2000) approached the problem of weakly supervised learning of object categories in clutter as a maximum likelihood estimation. For this purpose they derived an EM algorithm for the constellation model. We follow their approach in deriving an EM algorithm to estimate the parameters of our improved model.

The task of learning is to estimate the parameters $\theta_{fg} = \{\boldsymbol{\mu}, \Sigma, \mathbf{c}, V, M, \mathbf{D}, \mathbf{t}, U\}$ of the model

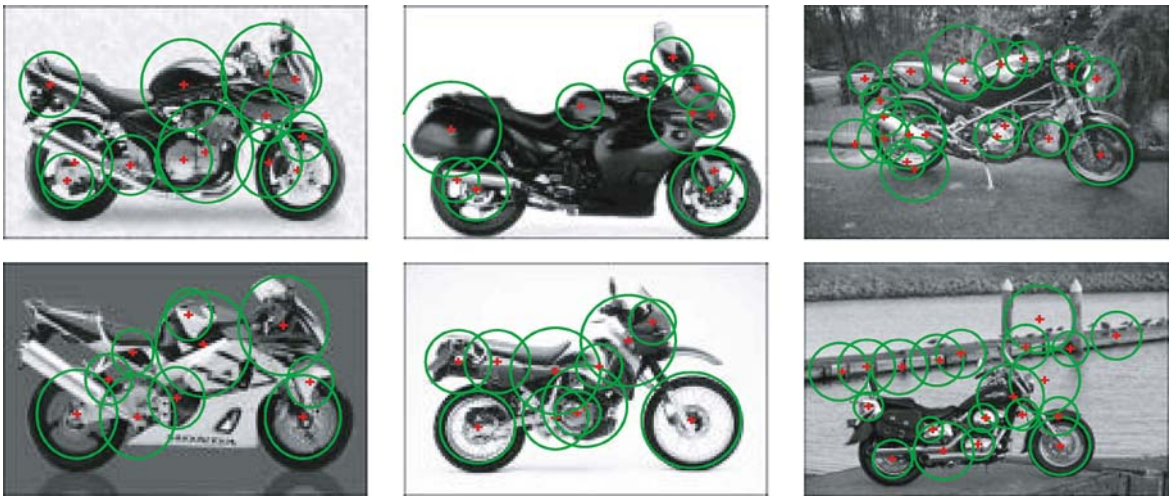


Figure 2. Six typical motorbikes images with the output of the Kadir-Brady operator overlaid. The +'s illustrate the centre of the salient region, while the circles show the scale of the region. Notice how the operator fires more frequently on more salient regions, ignoring the uniform background present in some of the images.

discussed above. The goal is to find the parameters $\hat{\theta}_{ML}$ which best explain the data $\mathbf{X}, \mathbf{S}, \mathbf{A}$ from all the training images, that is maximize the likelihood: $\hat{\theta}_{ML} = \arg\max_{\theta} \; p(\mathbf{X}, \mathbf{S}, \mathbf{A} | \, \theta_{fg})$. Note that the parameters of the background, θ_{bg}, are constant during learning.

Learning is carried out using the expectation-maximization (EM) algorithm Dempster et al. (1976) which iteratively converges, from some random initial value of θ_{fg} to a maximum (which might be a local one).

We now look at each stage in the learning procedure, giving practical details of its implementation and performance, using the motorbike dataset as an example. We assume that $\mathbf{X}, \mathbf{S}, \mathbf{A}$ have already been extracted from the images, examples of which are shown in Fig. 2. In this example, we are using the gradient based PCA representation, with $k = 11$ and $l = 15$.

3.1. Initialization

Initially we have no knowledge about the structure of the object to be learnt so we are forced to initialize the model parameters randomly. However, the model which has a large number of parameters, must be initialized sensibly to ensure that the parameters will converge to a reasonable maximum. For shape, the means are set randomly over the area of the image and the covariances to be large enough so that all hypotheses have a roughly equal weighting, so avoiding a bias toward nearby points. The appearance densities are initialised to zero mean, plus a small random perturbation, while the variances are set to be large. The same initialization settings are used in all experiments. In Fig. 3 we show three typical model initializations of the shape term (the appearance term is hard to visualize due to the large number of dimensions).

3.2. EM Update Equations

The algorithm has two stages: (i) the E-step in which, given the current value of θ_{fg} at iteration k, θ_{fg}^{k}, some sufficient statistics are computed and (ii) the M-step where we compute the parameters for the next iteration, θ_{fg}^{k+1} using these sufficient statistics.

We now give the equations for both the E-step and M-step. The E-step requires us to compute the posterior density of the hidden variables, which in our case are the hypotheses. This is calculated using the joint:

$$p\left(\mathbf{h}|\mathbf{X}, \mathbf{S}, \mathbf{A}, \theta_{fg}^{k}\right) = \frac{p\left(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h} | \, \theta_{fg}^{k}\right)}{\sum_{h \in H} p\left(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h} | \, \theta_{fg}^{k}\right)} = \frac{\frac{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h} | \, \theta_{fg}^{k})}{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}_0 | \, \theta_{bg})}}{\sum_{h \in H} \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h} | \, \theta_{fg}^{k})}{p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}_0 | \, \theta_{bg})}} \tag{17}$$

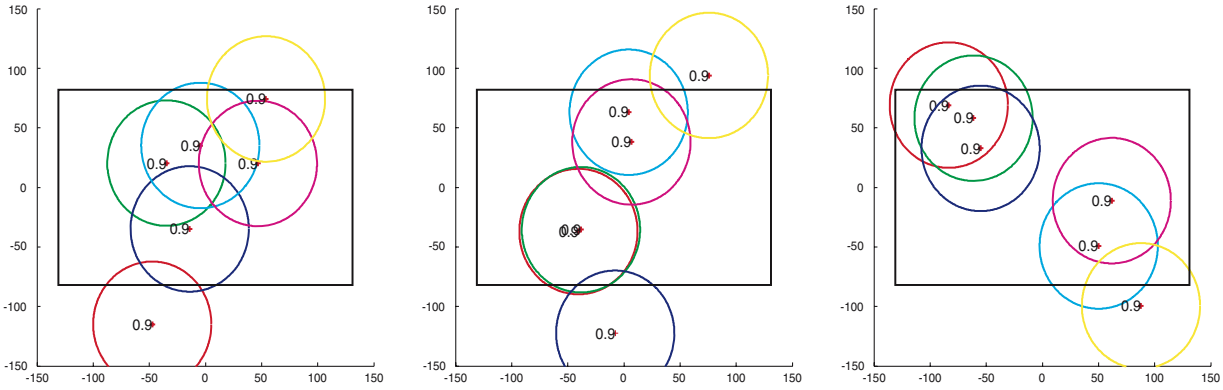


Figure 3. Three typical initializations for a 6 part motorbike shape model. The circles represent the variance of each part at 1 standard deviation (the inter-part covariance terms, which cannot easily be shown, are set to zero) with the mean being the centre of the circles. The probability of each part being present is shown just to the left of the mean. The average image size is indicated by the black box. As the images are resized to a constant width and their aspect ratio is unknown, we ensure that tall images are not penalised by allowing the initialization of some of the parts to lie outside the mean image box. Axis units are pixels. The variances here are referred to the centroid of the model.

We divide through by $p(\mathbf{X}, \mathbf{S}, \mathbf{A}, \mathbf{h}_0 | \theta_{bg})$ as it is easier to compute the joint ratio rather than the joint directly. We then calculate the following sufficient statistics for each image, i from which we have previously extracted $\mathbf{X}^i, \mathbf{A}^i, \mathbf{S}^i$: $E[\mathbf{X}^{**i}]$, $E[\mathbf{X}^{**i}\mathbf{X}^{**i\,T}]$, $E[\mathbf{A}^i_p]$, $E[\mathbf{A}^i_p \mathbf{A}^{i\,T}_p]$, $E[\mathbf{S}^{*i}]$, $E[\mathbf{S}^{*i}\mathbf{S}^{*i\,T}]$, $E[n^i]$, $E[\mathbf{D}^i]$ where the expectation is taken with respect to the posterior, $p(\mathbf{h}|\mathbf{X}, \mathbf{S}, \mathbf{A}, \theta^k_{fg})$, for example:

$$E[\mathbf{X}^{**i}] = \sum_{h \in H} p\left(\mathbf{h} \,|\, \mathbf{X}^i, \mathbf{S}^i, \mathbf{A}^i, \theta^k_{fg}\right) \mathbf{X}^{**i}(h) \quad (18)$$

Note that for simplicity we have not considered the case of missing data. The extensions to the above rules for dealing with this may be found in Weber (2000). The general principle is to condition on the features that are present to work out the expected values of those that are missing.

In the M-step we then compute $\theta^{k+1}_{fg} = \{\boldsymbol{\mu}^{k+1}, \Sigma^{k+1}, \mathbf{c}^{k+1}, V^{k+1}, \mathbf{t}^{k+1}, U^{k+1}, M^{k+1}, \mathbf{D}^{k+1}\}$:

$$\boldsymbol{\mu}^{k+1} = \frac{1}{I} \sum_{i=1}^{I} E[\mathbf{X}^{**i}]$$

$$\Sigma^{k+1} = \frac{1}{I} \sum_{i=1}^{I} E[\mathbf{X}^{**i}\mathbf{X}^{**i\,T}] - \boldsymbol{\mu}^{k+1}\boldsymbol{\mu}^{k+1\,T}$$

$$\mathbf{c}^{k+1}_p = \frac{1}{I} \sum_{i=1}^{I} E\left[\mathbf{A}^i_p\right] \quad \forall p \in P$$

$$V^{k+1}_p = \frac{1}{I} \sum_{i=1}^{I} E\left[\mathbf{A}^i_p \mathbf{A}^{i\,T}_p\right] - \mathbf{c}^{k+1}_p \mathbf{c}^{k+1\,T}_p \quad \forall p \in P$$

$$\mathbf{t}^{k+1} = \frac{1}{I} \sum_{i=1}^{I} E[\mathbf{S}^{*i}]$$

$$U^{k+1} = \frac{1}{I} \sum_{i=1}^{I} E[\mathbf{S}^{*i}\mathbf{S}^{*i\,T}] - \mathbf{t}^{k+1}\mathbf{t}^{k+1\,T}$$

$$M^{k+1} = \frac{1}{I} \sum_{i=1}^{I} E[n^i]$$

$$\mathbf{D}^{k+1} = \frac{1}{I} \sum_{i=1}^{I} E[\mathbf{D}^i]$$

where I is total number of training images. The two stages are then repeated until θ^k_{fg} converges to a stable point. See Section 3.5 below for a discussion on the convergence properties of the algorithm. Details on the operation of both steps for the model can be found in Weber (2000); Fergus (2005).

3.3. *Computational Considerations*

In computing the sufficient statistics in the E-step we need to evaluate the likelihood for every hypothesis. Since there are $O(N^P)$ per image, this is the major computational bottleneck in our approach. Possible ways around this include:

1. Use the mode: Approximate the summation over $\mathbf{h}$ by just taking the mode, i.e. the best hypothesis in each frame. The problem with this is that the initial assignments are totally random and so initially picking the best hypothesis is unlikely to be close to the

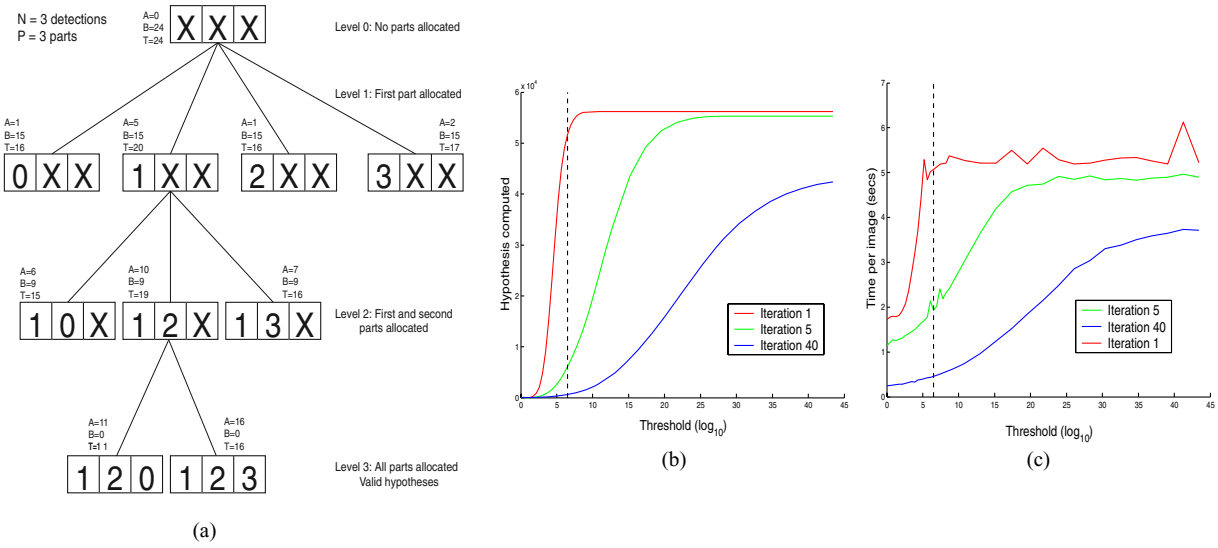


Figure 4. (a) Illustration of A^* search process for a 3 part toy model and an image with 3 regions. As the tree is descended, regions are allocated to parts, with the leaves of the tree constituting complete hypotheses. The score (T) of each node in the tree is a sum of the likelihood of the allocated features (A) and an upper-bound on the likelihood of the remaining, unallocated features (B). The list of open nodes is stored in a binary heap, with the node with the highest overall value (T) being the one opened next. By repeatedly removing complete nodes from the tree, the hypotheses can be extracted in order of likelihood. (b) A graph showing how the number of hypotheses evaluated increases as the likelihood drops below that of the best hypothesis on a typical motorbike model. The vertical line shows the value of the threshold used in experiments. The three curves correspond to different stages of learning: red—at the beginning when the model variances are large; green—in the middle and blue—when the model variances are large (see Fig. 6(c) for evolution of model during learning). Note that as the model variances decrease, the number of hypotheses evaluated decreases. (c) As for (b), but the y-axis is evaluation time per image.

optimal one and it is difficult to escape from such local minima in subsequent iterations. The practical consequences are that the model is more prone to numerical explosions (probabilities go to zero somewhere); or the models converge to bad local maxima. See Fig. 8(b) for an example of the latter.

2. Sample hypotheses: While sampling methods could be used to evaluate the marginalization in the E-step (Jerrum and Sinclair, 1997), the imposition of constraints (e.g. see Section 3.6) on possible hypotheses introduces complications. These constraints mean that it would be more difficult to move through the space of possible hypotheses, since many proposed moves would not be valid, increasing the chances of hitting local minima. For this reason, we prefer more direct computation methods.
3. Reduce the dependencies: The cause of our problems is assuming that the location of all parts is dependent on one another. A simpler dependency structure could be adopted, conditioning on a single landmark part, for example. This would reduce the $O(N^P)$ problem to $O(N^2 P)$. This is investigated in Fergus et al. (2005).
4. Efficient search methods: Only portion of the hypotheses have a high probability thus we can accurately approximate the summation over all hypotheses by just considering this subset. By utilizing various heuristics, specific to our application, we can efficiently compute the few hypotheses contributing much of the probability mass. The details of this are now investigated in Section 3.4.

3.4. *Efficient Search Methods*

Computing the very small portion of the hypotheses that have a high probability enables the learning procedure to run in a reasonable time.

A tree structure is used to search the space of all possible hypotheses. The leaves of the tree are complete hypotheses with each level representing a part: moving down the tree, features are allocated to parts until a complete hypothesis is formed. At a node within the tree, an upper-bound on the probability of remaining, unallocated parts can be computed enabling us to employ the A^* algorithm (Grimson and Lozano-Pérez,

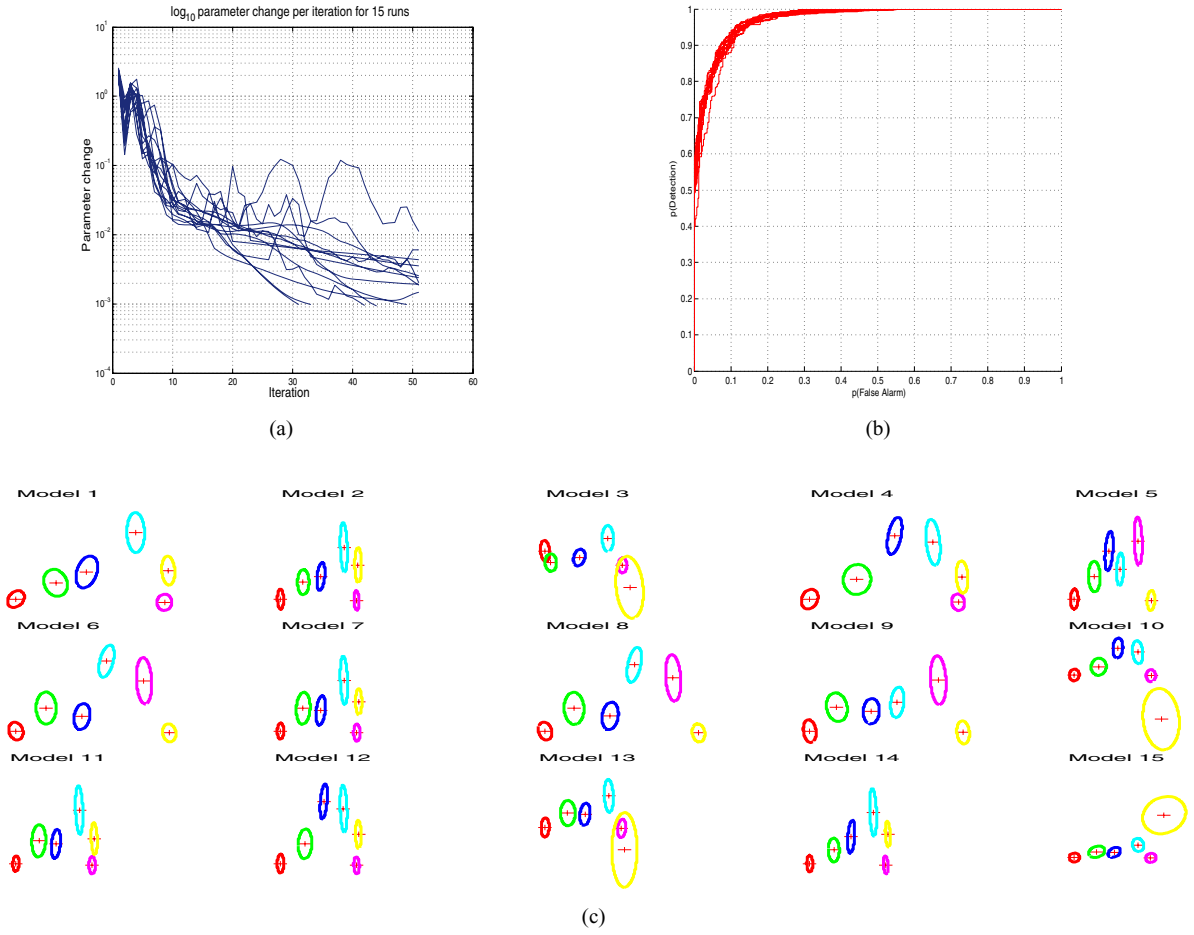


Figure 5. 15 learning runs for the motorbike category, each with a different random initialisation. (a) The maximal parameter change per iteration for 15 runs, started at different initial conditions. (b) The 15 ROC curves (evaluating test data) corresponding to each of the runs, the mean error rate is 9.3%, with a standard deviation of 0.6%. (c) The shape models for each of the 15 runs. Note the different maxima found.

1987; Hart et al., 1968). This allows the efficient exploration of the tree, resulting in the guaranteed discovery of the best hypothesis. This can be removed from the tree and the search continued until the next best hypothesis is found. In this manner, we can extract the hypotheses ordered by likelihood. Figure 4(a) shows a toy example of this process.

If q parts have been allocated, the upper bound on the probability for the remaining $P - q$ parts is easily computed thanks to the form of the densities in the model. Given the occlusion states of the unallocated parts, the upper bound is a constant, thus it becomes a case of finding the maximum upper bound for each of the 2^{P-q} possible states.

A binary heap stores the list of open branches on the tree, having $\log n$ access time. Conditional densities for each part (i.e. conditioning on previously allocated parts) are pre-computed to minimize the computation necessary at each branch. Details of the A^* search can be found in Fergus et al. (2001).

For each image, we compute all hypotheses until they become smaller than some threshold (e^{-15}) smaller than the best hypothesis. Fig. 4(b) shows how the number of hypotheses varies for a given likelihood. This threshold was chosen to ensure that the learning progressed within a reasonable time while evaluating as many hypotheses as possible. Additionally, space search methods are used to prune the branches explored at each new node in the tree. At a given level of the tree, the joint density of the shape term allows the density of location of the current part to be computed by conditioning on the previously allocated parts. Only a subset

of the N detections need be evaluated by this density: we assume that we can neglect detections if their probability is worse than having all remaining parts be missing. Since the occlusion probabilities are constant for a given learning iteration, this gives a threshold which truncates the density. If the covariance of the density is small, only the best few detections need to be evaluated, enabling significant numbers of hypotheses to be ignored.

Despite using these efficient methods, learning a $P = 6$–7 part model with $N = 20$–30 features per image (a practical maximum), using 400 training images, takes around 24 hours to run. This equates to spending 3–4 seconds per image, on average, at each iteration (given a total running time of 24 hours, with 400 training images and 50 EM iterations). It should be noted that learning only needs to be performed once per category, due to the good convergence properties as discussed in Section 3.5.

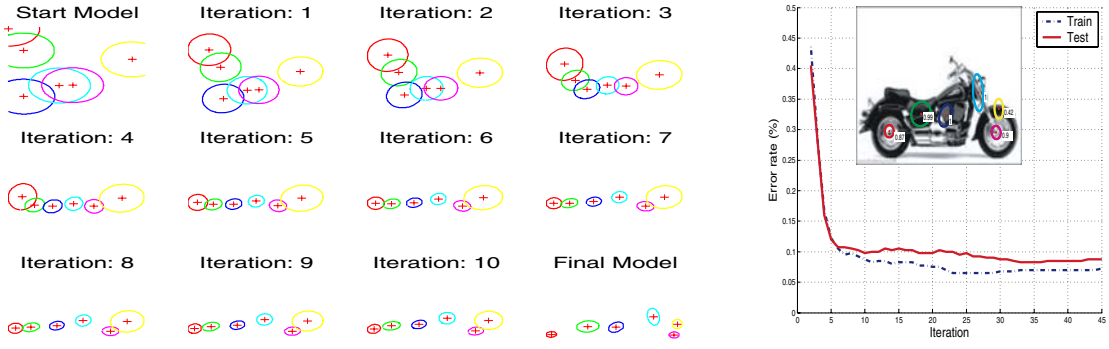
It is worth noting that just finding the best hypothesis (the mode), is not that much quicker than taking the small subset of high scoring hypotheses (see Fig. 4(c)), since a reasonable portion of the tree structure must be explored before a complete hypothesis is found. This provides another justification for summing over multiple hypotheses rather than just taking the best.

3.5. *Convergence*

Table 1 illustrates how the number of parameters in the model grows with the number of parts (assuming $l = 15$). Despite the large number of parameters in the model, its convergence properties are respectable. Figure 5 shows the convergence properties and classification performance of 15 models started from different initial conditions but with identical data. Note that while the models converge at different rates and to different points in parameter space (see the different shape models for each run in Fig. 5(c)),the ROC curves of test set performance are very similar, the standard deviation at equal-error rate being 0.6%. Figure 6(a) shows the shape model evolving throughout the learning process for a typical learning run. Figure 6(b) shows the classification performance of the model improving as the model converges. Both figures demonstrate that the majority of the performance is obtained within the early stages of learning. However, occasionally a superior maximum can be found after a large number of iterations, therefore we continue until we are sure that the model has reached a stable point. Two criteria were used to stop the EM iteration: (i) Number of iterations exceeds some limit (50) and (ii) The absolute value of norm of parameter change per iteration drops below some limit (10^{-3}—for the shape term this equates to around 1/10th of a pixel). In practice the former criterion is used more often.Figure 7gives an insight into the

Table 1. Relationship between number of parameters and number of parts in model

Parts	2	3	4	5	6	7
# parameters	77	123	177	243	329	451



(a) (b)

Figure 6. (a) The evolution of the motorbike shape model throughout learning. (b) Classification performance versus learning iteration. The inset shows the final shape model superimposed on a motorbike image.

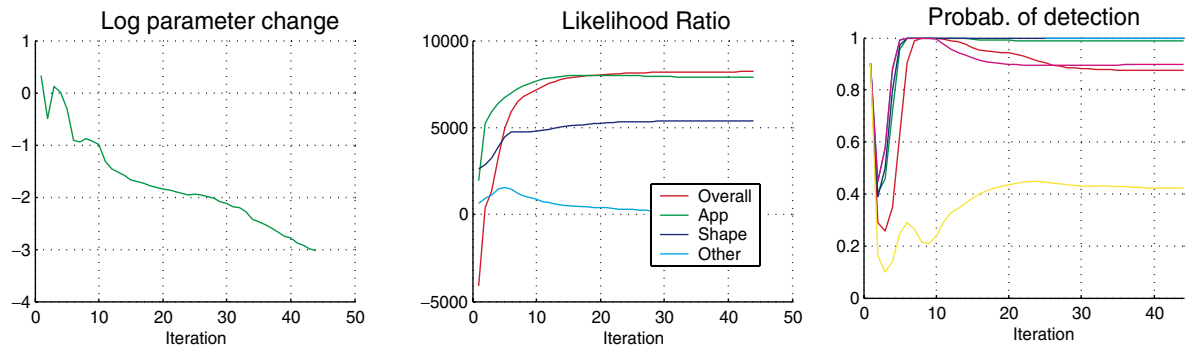


Figure 7. A typical learning run on the motorbikes dataset. (a) The magnitude of largest parameter change versus learning iteration (log-scale) (b) Overall log-likelihood ratio versus iteration. Likelihoods for each of the components within the model are also shown. (c) Evolution of the probability of each part being present. Each colour corresponds to a different part. Note that the probability drops down to a low value for the first couple of iterations before the model finds some structure in the data and the probability picks up again.

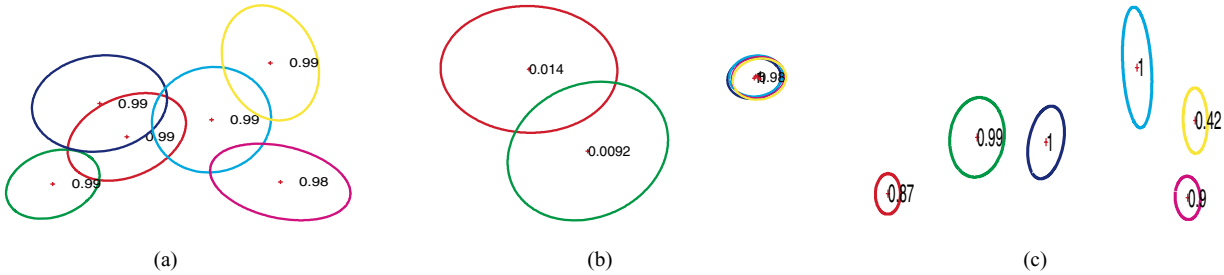


Figure 8. (a) A motorbike shape model learnt using no ordering constraint on regions with hypotheses. Note the large overlapping distributions due to permutations of nearby features. The model is also $P!$ slower to learn that those with ordering constraints imposed. (b) A motorbike shape model learnt just using the best hypothesis in each frame. A poor local maximum has been found with two parts being redundant, having a low probability of occurrence. (c) The shape model obtained if the an ordering constraint on the x-coordinates of the hypothesis is imposed.

convergence of the model during a typical learning run. Figure 7(b) shows how the log-likelihood ratio over all images increases monotonically as it should. The plot also gives a breakdown of the terms within the model. Figure 7(a) shows the parameter change per iteration steadily reducing until it hits a 10^{-3} limit. In Fig. 7(c) the probability of each part being present is shown. Initially, the probability starts out low, but within 10 iterations or so each one has locked onto a sensible signal thus has a high probability. Between 20 and 50 iterations, some fine settling of the probabilities can be seen as the parts converge down on features of the object. Convergence to a successful model is dependent on a variety of factors, but two important ones are: (a) a consistent set of regions from which to learn and (b) the introduction of an ordering constraint on the x-coordinates of regions within a hypothesis. While the former is discussed in more detail in Section 5.3.1, we now elaborate on the latter. To aid both convergence and speed, an ordering constraint is placed on the x-coordinates of features allocated to parts: the features selected must have a monotonically-increasing x-coordinate. This reduces the total number of hypotheses by $P!$ but unfortunately imposes an artificial constraint upon the shape of the object. If the object happens to be orientated vertically then this constraint can exclude the best hypothesis. Clearly in this scenario, imposing a constraint on the y-coordinate ordering would resolve the problem but it is not clear how to choose such an appropriate constraint automatically, other than learning models with different ordering constraints and picking the one with the highest likelihood. SeeFig. 8(a) for an example of a model learnt without this constraint.

3.6. *Background Model*

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 were estimated from a set of background images and not updated within the EM iteration.

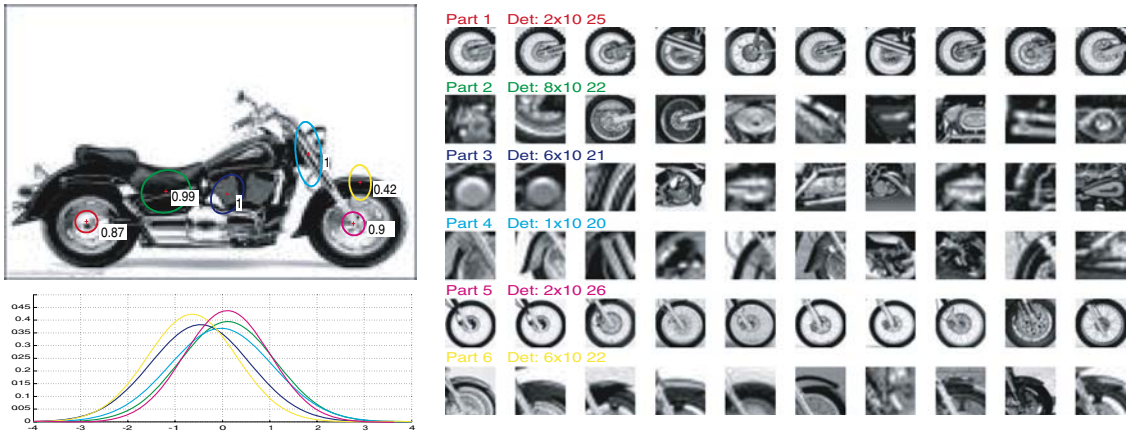
3.7. *Final Model*

In Fig. 9 we show a complete model, from one of the 15 training runs on the motorbike dataset. It is pleasing to see that a sensible spatial structure has been picked out and that the appearance samples correspond to distinctive parts of the motorbike. In Fig. 10 the appearance density of the model is analyzed. In Fig. 10(a) the 15 principal components of appearance are shown. For ease of viewing, the basis is shown in intensity space, having integrated the original gradient space basis. Figure 10(b) shows the distribution in appearance space of patches assigned to each model part by the best hypothesis in an image and for the remaining background patches. For example, consider the parts of the model corresponding to the wheels of the motorbike (the 1st (red) and 5th (magenta) parts in Fig. 9). Looking at the histograms in 10(b), we can see that the 8th principal component is a doughnut shape and that the red and magenta histograms are considerably skewed from the background histogram, unlike those for other model parts of this descriptor. The assumption that the appearance data is Gaussian in nature can also be examined in Fig. 10. For the most part the foreground data is well approximated by a single Gaussian component. The background data seems to follow a Gaussian distribution as well. In Fig. 10(c) histograms show how discriminative each part is. Although many of the descriptors are weak classifiers, they combine to give a strong classifier. Notice that the parts corresponding to wheels have a foreground histogram which is quite distinct from the background histogram. Other parts, such as the third one, are not so discriminative in appearance, having foreground and background distributions that overlap considerably. Instead, this part may contribute in the shape term to the model.

4. Recognition

The recognition process is very similar in nature to learning. For query image, t, recognition proceeds by first detecting features, giving $\mathbf{X}^t$ and $\mathbf{S}^t$. These features are processed in same way as in learning, giving $\mathbf{A}$.

Once $\mathbf{X}^t$, $\mathbf{A}^t$ and $\mathbf{S}^t$ have been obtained we then compute the likelihood ratio using (1). To determine if an object is in the image, we should, according to (2) sumover all hypotheses. However, in practice the



(a) (b)

Figure 9. A complete model: (a) Top: Shape density superimposed on an image of a motorbike. The ellipses represent the covariance of each part (the inter-part covariance terms cannot easily be shown) and the probability of each part being present is shown just to the right of the mean. Bottom: Relative scale densities. (b) Samples belonging to each part (i.e. from the best hypothesis in a training image) which are closest to the mean of the appearance density. The colours correspond to the colours in (a). The determinant of each appearance density is given to provide an idea of the relative tightness of each parts' density. A high exponent corresponds to a tight density with a consistent appearance and vice versa.

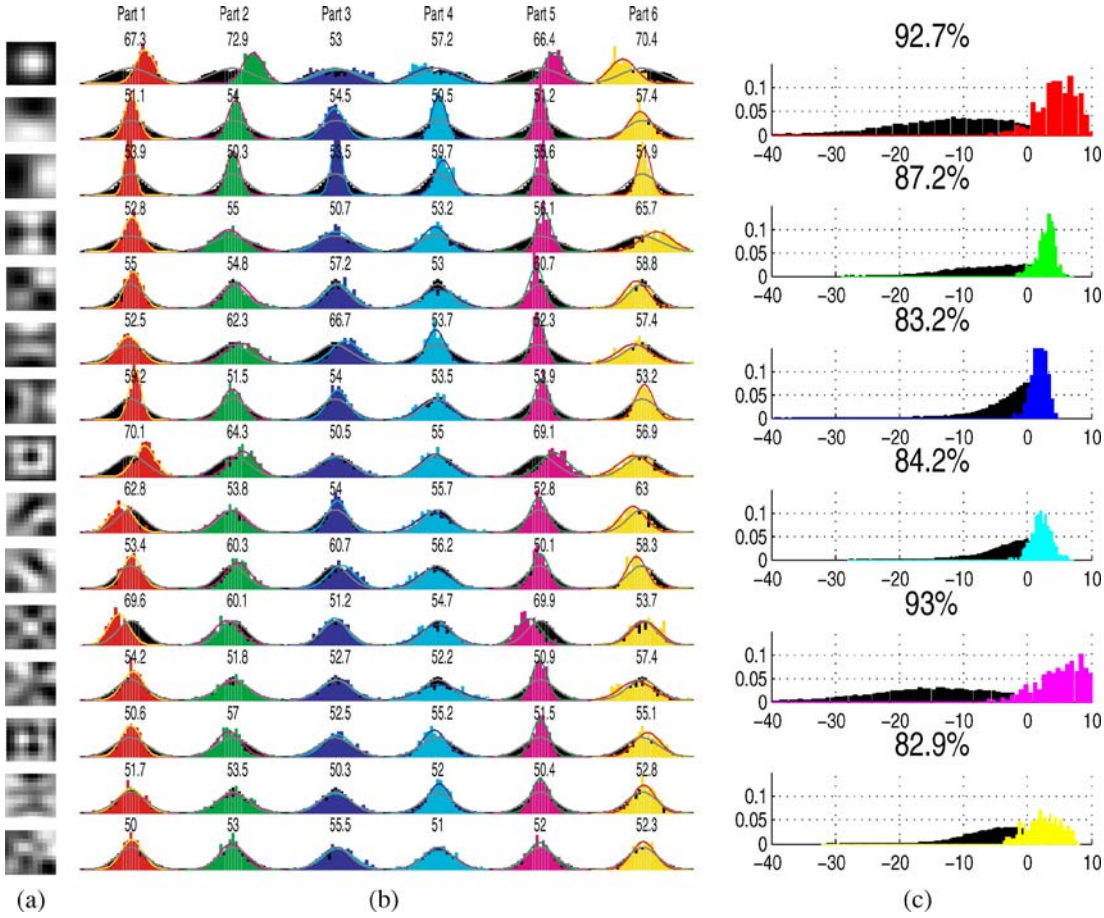


Figure 10. A breakdown of the appearance term in the motorbike model of Fig. 9. (a) The 15 principal components of the PCA basis. (b) Histograms of the background (in black) and foreground coordinate (coloured) on each of the 15 principal direction (rows) for each of the 6 model parts (columns). Superimposed on the histograms is the Gaussian fitted in the learning procedure. The background histogram is produced by considering all regions found by the feature detector from the background dataset. The foreground histogram is produced by considering the regions assigned to the best hypothesis of each image in the training set. The number above each histogram gives the true positive detection rate (in %) at the point of equal false positive and false negative errors on ROC curve between the foreground and background histograms for the particular part/descriptor, so giving a measure of how discriminative it is. (c) The likelihood histograms of both the background data and the foreground data (as used in (b)) under the density for each model part. The title of each plot gives the true detection rate (computed from the ROC curve at the point of equal error), giving a measure of discrimination for each part overall.

best results are obtained by just selecting the best one, since the background images typically contain many low-scoring hypotheses, which if combined can give a false alarm. The likelihood ratio, assuming we take the ratio of the priors in (1) to be 1, is the same as the ratio of posteriors, R. This is then compared to a threshold to make the object present/absent decision. This threshold is determined from the training set to give the desired balance between false positives and false negatives.

If we wish to localize each instance of the object within the image, the best hypothesis is taken and a bounding box around it formed at its location. We then sum over all hypotheses which are within this box. If the total is greater than the threshold then an instance of the object is placed at the centroid of the box and all features within the box are deleted. The next best hypothesis is then found and the process repeated until the sum of hypotheses within the box falls below the threshold.

The same efficient search methods described in Section 3.4 are used in the recognition process to find the single best hypothesis. However, recognition is faster

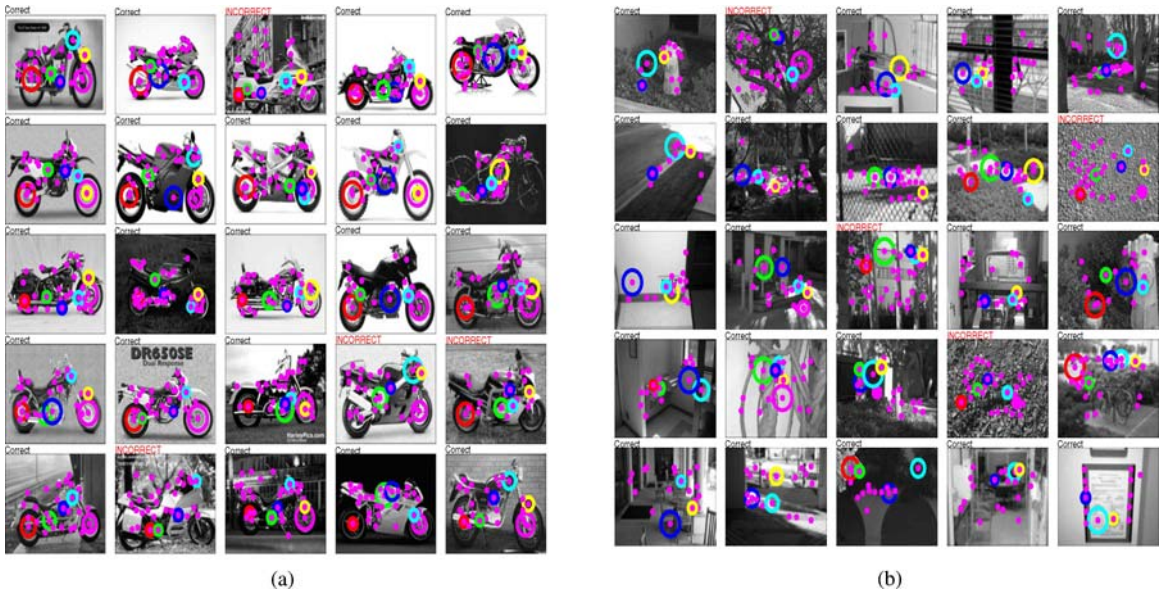


Figure 11. (a) The motorbike model from Fig. 9 evaluating a set of query images. The pink dots are features found on each image and the coloured circles indicate the features of the best hypothesis in the image. The size of the circles indicates the scale of feature. The outcome of the classification is marked above each image, incorrect classifications being highlighted in red. (b) The model evaluating query images consisting of scenes around Caltech—the negative test set.

as the covariances are tight (as compared with the initial values in the learning process) so the vast majority of hypotheses may safely be ignored. However the large N and P mean the process still takes around 2–3 seconds per image to perform the search, in addition to the 10 seconds/image needed to extract the features.

5. Results

A variety of experiments are carried out on a number of different datasets, each one containing images representing a different category. Since the model only handles a single viewpoint, the datasets consist of images that are mirror-reversed, if necessary, so that all instances face in a similar direction, although there is still variability in location and scale within the images. Additional experiments are performed to elucidate the properties of our model and include baseline methods.

For each experiment, the dataset are split randomly into two separate sets of equal size. The model is then trained on the first and tested on the second. In recognition, the models are tested in both classification and localisation roles.

In classification, where the task is to determine the presence or absence of the object within the image, the performance is evaluated by plotting receiver-operating characteristic (ROC) curves. To ease comparisons we use a single point on the curve to summarize its performance, namely the point of equal error (i.e. p(True positive) $= 1 -$ p(False positive)) when testing against one of two background datasets. For example a fig. of 9% means that 91% of the foreground images are correctly classified and 9% of the background images were incorrectly classified (i.e. thought to be foreground). While the number of foreground test images varied between experiments, depending on the number of images available for each category, the foreground and background sets are always of equal size.

In localisation, where the task is to place a bounding box around the object within the image, the performance is evaluated using recall-precision curves (RPC),[2] since the concept of a true negative is less clear in this application. To be considered a correct detection, the area of overlap, a_o between the estimated bounding box B and the ground truth bounding box, B_{gt} must exceed 0.5 according to the criterion:

$$a_o = \frac{\text{area}(B \cap B_{gt})}{\text{area}(B \cup B_{gt})} \tag{19}$$

When evaluating the UIUC dataset, we adopt the same criterion as used in Agarwal and Roth (2002). The ex-

periments use identical software and settings for all categories (with the exception of the UIUC dataset, due to the small image sizes). Images were resampled to 300 pixels in width by bilinear interpolation, regions extracted at scales between 10 and 30 pixels in radius. The gradient-based PCA with additional energy and residual terms is used, with parameters: $k = 21, l = 15$. In learning, we use $P = 6$ and $N = 20$. N was increased to 30 in recognition.

5.1. Datasets

Six diverse datasets were used in the experiments: motorbikes, airplanes, spotted cats, faces, cars (rear) and cars (side). Examples from these datasets can be seen in Fig. 1. The datasets of motorbikes, airplanes, cars (rear), faces and cluttered scenes around Caltech (used as the negative test set) are available from our websites (Fergus and Perona, 2003). Two additional background datasets were used. The first is collected from Google's image search using the keyword "things", resulting in a highly diverse collection of images. The second is a set of empty road scenes for use as a realistic background test set for the cars (rear) dataset. The cars (side) dataset is the UIUC dataset (Agarwal et al., 2002). The spotted cat dataset, obtained from the Corel database, is only 100 images originally, so another 100 were added by reflecting the original images, making 200 in total. Amongst the datasets, only the motorbikes, airplanes and cars (rear) contained any meaningful scale variation. All images from the datasets are converted to grayscale as colour is not used in our experiments. Table 2 gives the size of training set used for each dataset in the experiments.

5.2. Experiments

Figures 12–14 show models and test images with a mix of correct and incorrect classifications for four of the datasets. Notice how each model captures a simple description, be it in appearance or shape or both, of the object. The face and motorbike datasets have tight shape models, but some of the parts have a highly variable appearance. For these parts any feature in that location will do regardless of what it looks like (hence the probability of detection is (1). Conversely, the spotted cat dataset has a loose shape model, but a highly distinctive appearance for each patch. In this instance, the model is just looking for patches of spotty fur, regardless of their location. The differing nature of these examples illustrate the flexible nature of the model.

Table 2. Classification results on five datasets. (a) is the error rate(%) at the point of equal-error on an ROC curve for a scale-variant model, testing against the Caltech background dataset (with the exception of Cars (rear) which uses empty road scenes as the background test set). (b) is the same as (a) except that a scale-invariant model is used. (c) is the same as (b), except that the Google background dataset was used in testing.

Dataset	Total size of dataset	(a)	(b)	(c)
Motorbikes	800	3.3	3.3	6.0
Faces	435	10.6	8.3	10.1
Airplanes	800	6.7	6.3	6.5
Spotted cats	200	12.0	11.0	11.0
Cars (rear)	800	12.3	9.2	9.3

Table 3. Confusion table between the four categories. Each row gives a breakdown of how a query image of a given category is classified (in %). No clutter dataset was used, rather images belonging to each category acted as negative examples for models trained for the other categories. The optimum would be 100% on the diagonal with zero elsewhere.

	Recognised category				
Query image	A	C	F	S	M
(A)irplane	88.8	6.0	0.3	0.7	4.2
(C)ars (rear)	19.7	67.0	0.8	3.3	9.2
(F)ace	2.8	1.4	86.2	2.3	7.3
(S)potted cats	3.0	1.0	3.0	76.0	17.0
(M)otorbike	1.3	0.0	0.0	1.0	97.7

The majority of errors are a result of the object receiving insufficient coverage from the feature detector. This happens for a number of reasons. One possibility is that, when a threshold is imposed on N (for the sake of speed), many features on the object are removed. Alternatively, the feature detector seems to perform badly when the object is much darker than the background (see examples in Fig. 12). Finally, the clustering of salient points into features, within the feature detector, is somewhat temperamental and can result in parts of the object being missed. Table 3 shows a confusion table between the different categories, using the models evaluated in Table 2. Despite being inherently generative, the models can distinguish between the categories well. The cars rear model seems to be somewhat weak, with many car images being claimed by the airplane model.

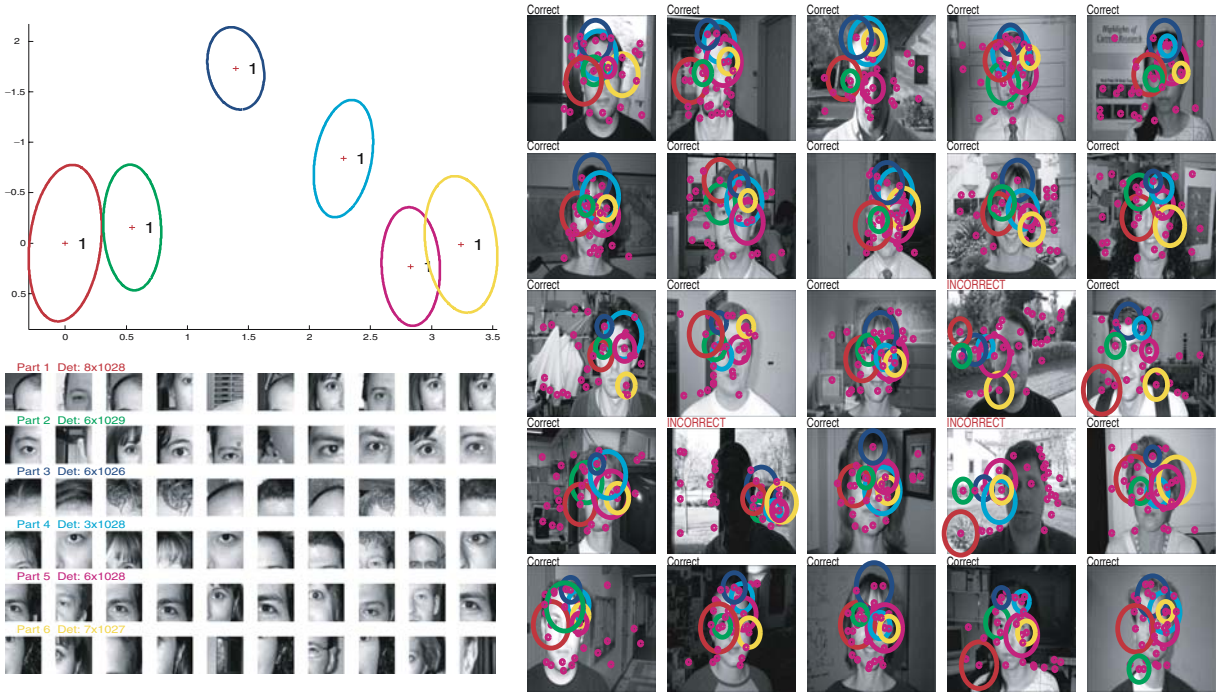


Figure 12. A typical face model with 6 parts with a mix of correct and incorrect detections.

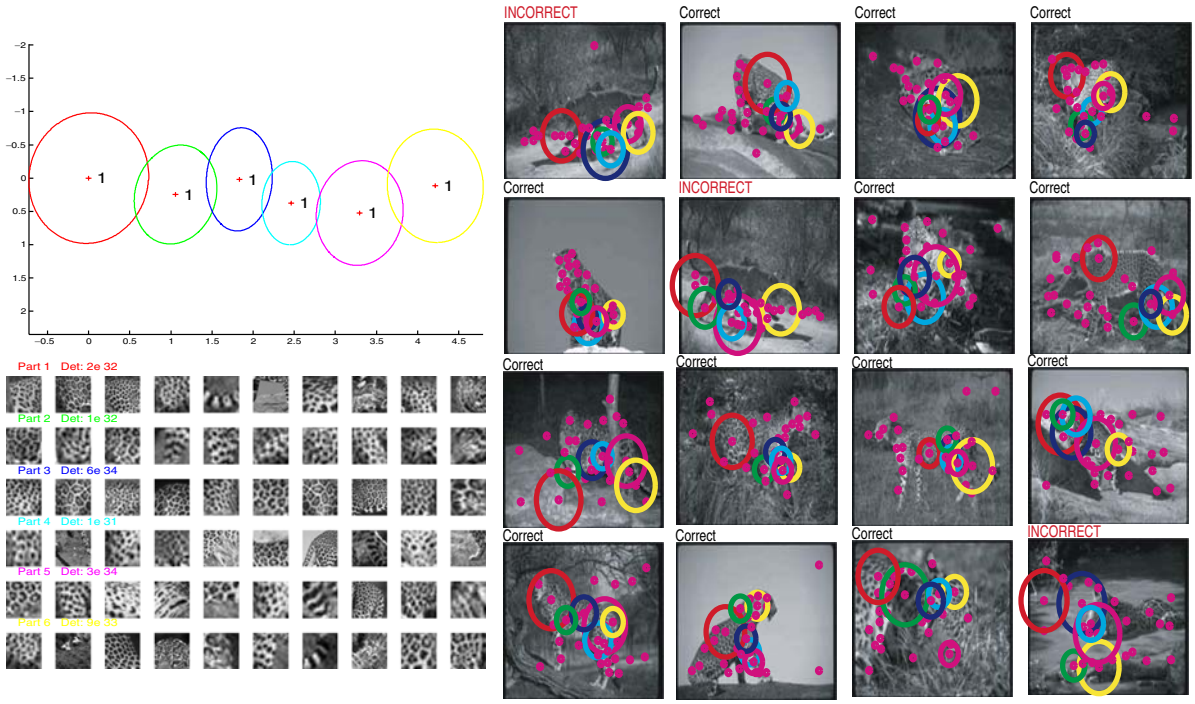


Figure 13. A typical spotted cat model with 6 parts. Note the loose shape model but distinctive "spotted fur" appearance.

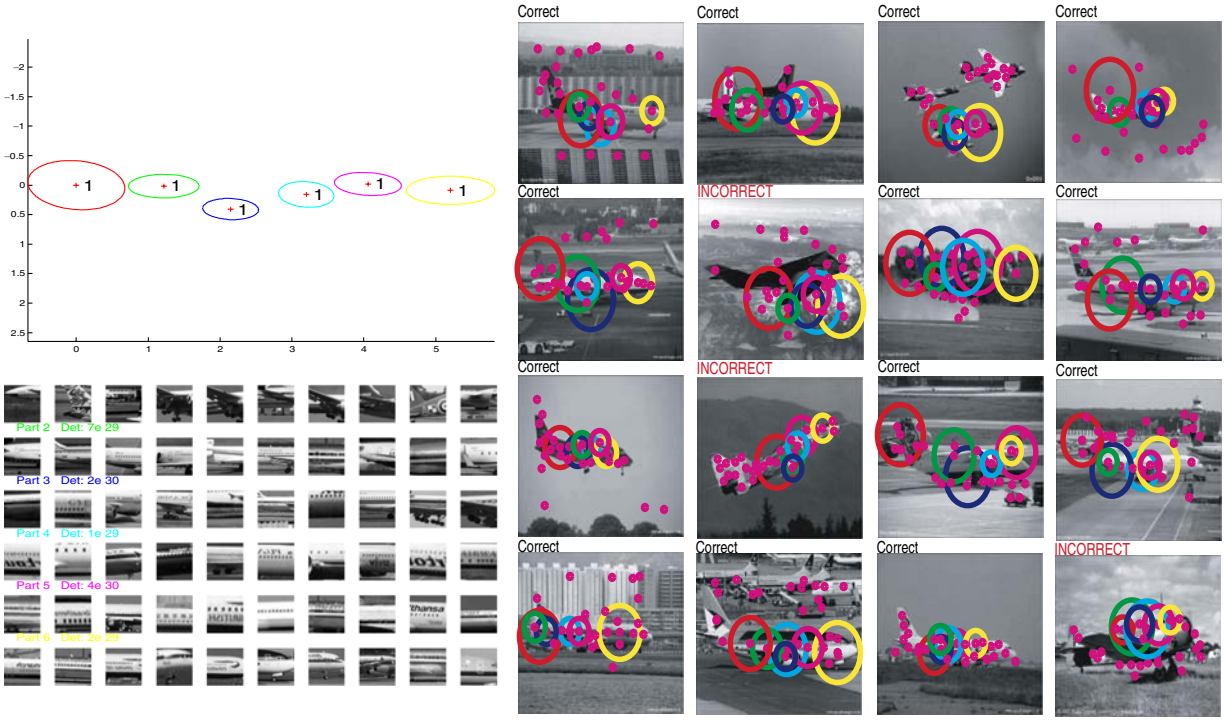


Figure 14. A typical airplane model with 6 parts. The long horizontal structure of the fuselage is captured by the shape model.

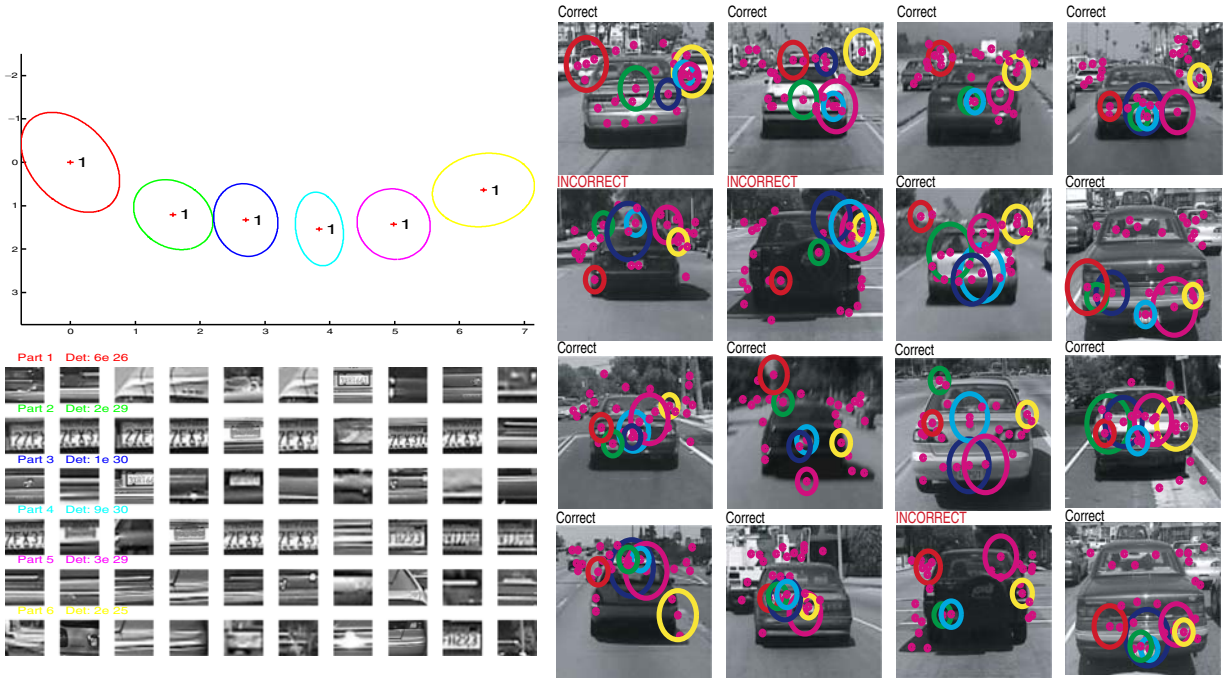
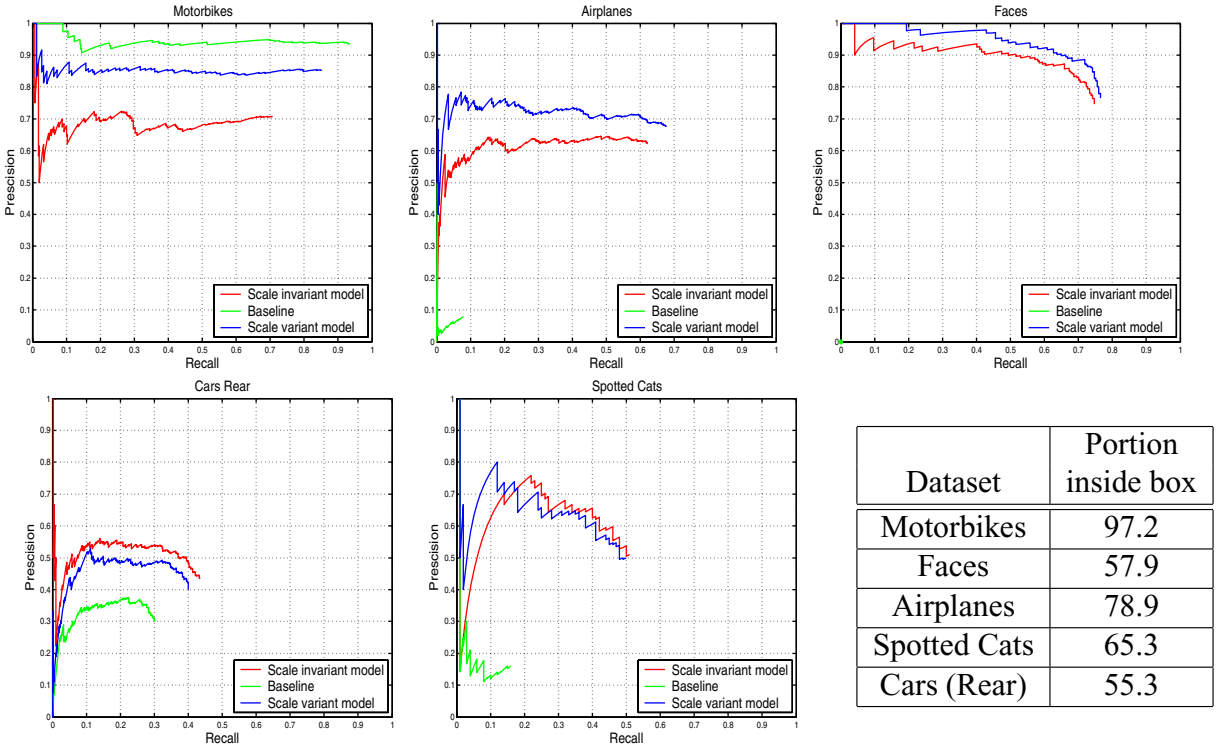


Figure 15. A 6 part Cars (Rear) model. The model consists of low-level horizontal line type structures on the back of the car, along with the license plate.



Dataset	Portion inside box
Motorbikes	97.2
Faces	57.9
Airplanes	78.9
Spotted Cats	65.3
Cars (Rear)	55.3

Figure 16. Recall-precision curves evaluating the localisation ability of the models for each of the five datasets. The fixed scale model is shown in blue; the scale-invariant model in red and a crude baseline in green. With the exception of cars rear, the fixed scale model preforms better than the scale-invariant probably because the scale of the objects in the training and testing set is fairly constant for most categories. The table lists the portion of regions lying within the ground truth bounding box of the object for each dataset. This is a crude measure for the difficulty of each category.

Detection performance on the five datasets are shown in Fig. 16. The predicted bounding box was produced by taking the bounding box around the regions of the best hypothesis and then enlarging it by 40% since the model representation tends to miss some parts of the object (e.g. the tail of the airplanes). It was assumed that a single instance was present in each frame. Recall precision curves are plotted for the fixed scale model; the scale-invariant model and a crude baseline, using the criterion in (19). The baseline consists of assuming the object occupies the whole image and using the likelihood ratios for each image from the scale-invariant model. Hence the baseline gives an indication of the total area the objects occupy within the dataset. For example, the motorbikes tend to fill most of the frame so the baseline beats two model variants but performs badly on the other categories. Another baseline measure is given in Fig. 16(f) where we specify the portion of regions inside the ground truth bounding box of the object. It is interesting to note that the fixed-scale models beat the scale-invariant models since in many of the datasets the scale variation is not that large and the scale-invariant model is making predictions within a larger space. The exception to this is the cars (rear) dataset, where the scale variation is larger and the scale-invariant model outperforms the scale variant one.

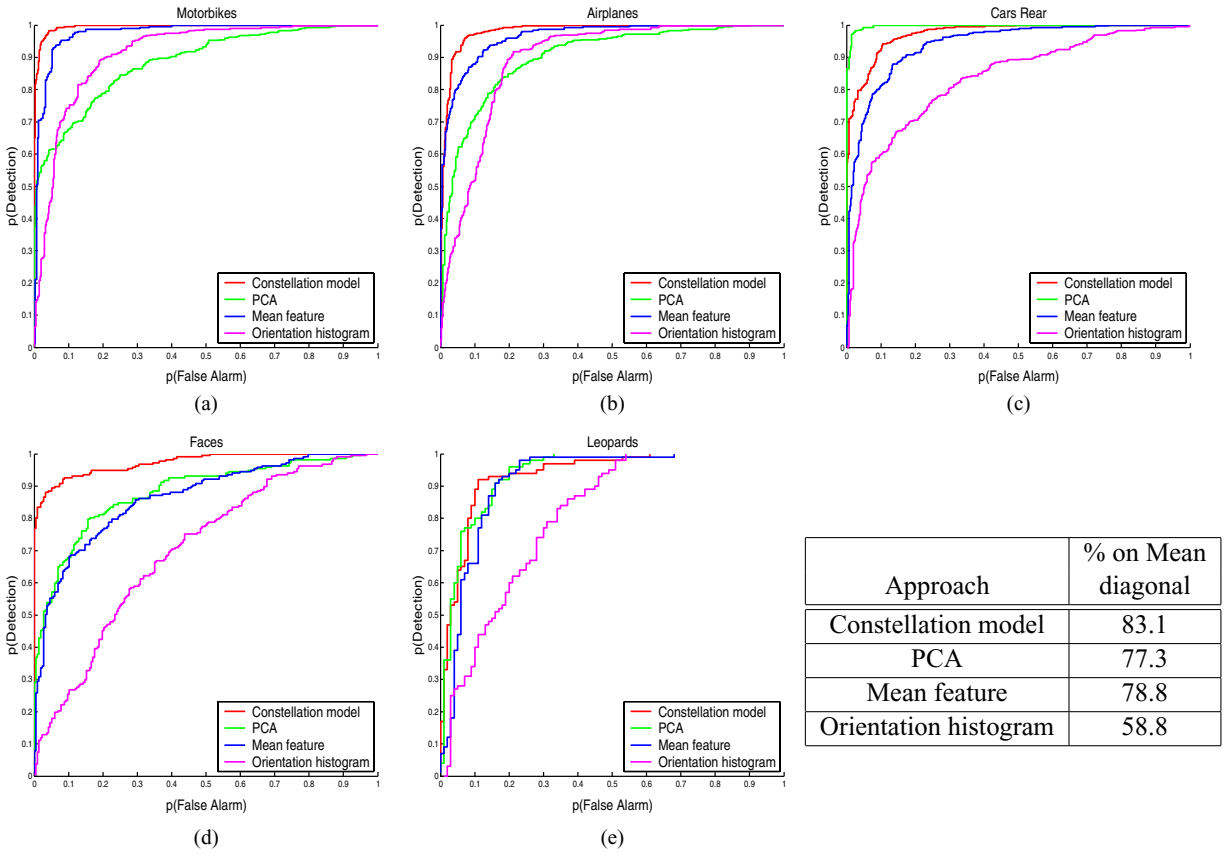
Using the cars (side) dataset, the ability of the model to perform multiple-instance localization is tested. The training images were mirror-reversed, if necessary so that all cars would face in the same direction, but the test examples contained cars of both orientations. Given the approximate symmetry of the object, the right-facing model had little problem picking out left-facing test examples. The model and test examples are shown in Fig. 17, while the recall precision curve comparing to Agarwal and Roth (2002) is shown in Fig. 25(a).

5.2.1. Baseline Experiments. To put the performance of the model in context, we apply a variety of baseline methods to the datasets:

Figure 17. The UIUC Cars (Side) dataset. The task here is to localize the object instance(s) within the image. On the left, we show the 7 part model. On the right, examples are shown from the test set, with correct localizations highlighted in green, while false alarms are shown in red.

- *Orientation histograms:* The gradients of the whole image are computed and thresholded to remove areas of very low gradient magnitude. An 8 bin histogram is computed of the orientations within the gradient image, with weighting according to the magnitude of the gradient. Each image is thus represented as by an 8-vector. The classifier consists of a single Gaussian density for both the foreground and background class, modeling the mean and variance of the 8 histogram bins. The parameters of these 8 dimensional densities are estimated from the training data. A query image is evaluated by computing the likelihood ratio of the images' 8-vector under the foreground and background Gaussian models.
- *Mean feature:* The arithmetic mean of **A** (the appearance of all features in each image) is computed, giving a 17-vector for each image. The classifier consists of a single Gaussian density both for the foreground and background class. The parameters of these 17 dimensional densities are estimated from the training data. A query image is evaluated by computing the likelihood ratio of the images' 17-vector under the foreground and background Gaussian models. This baseline method is designed to reveal the discriminative power of feature detector itself.
- PCA: Each image is resized to a 21 $\times$ 21 patch. After appropriate normalisation, it is then projected into a 25 dimensional PCA basis (precomputed and the same for all categories). The classifier consists of a single Gaussian density for both the foreground and background class, modeling the coefficients of each basis vector. The parameters of these 25 dimensional densities are estimated from the training data. A query image is evaluated by computing the likelihood ratio of the images' 25-vector under the foreground and background Gaussian models.

In Figs. 18(a)-(e), we show ROC curves for the three baseline methods and the constellation model on the five datasets in a classification task. Two of the baseline methods perform reasonably well. However, the constellation model is only beaten in one case—cars (rear)—by the PCA baseline. In Fig. 18(f) we compare the performance in a multi-class setting by giving the mean diagonal of a confusion table over the five categories for each of the approaches, with the constellation model showing superior performance. It should be emphasized that the constellation model allows localization of each detected object and of its parts, while the baseline methods do not.



Approach	% on Mean diagonal
Constellation model	83.1
PCA	77.3
Mean feature	78.8
Orientation histogram	58.8

Figure 18. ROC curves of the baseline experiments on the five datasets (a)–(e). The red curve is the constellation model while the green, blue and magenta curves are PCA; the mean feature and orientation histogram baselines respectively. The constellation model beats the baselines in all cases except for the cars rear dataset. (f) The table gives the mean diagonal of a confusion table across all five datasets for the 4 different approaches.

5.3. Analysis of Performance

We look at the model working under various different conditions such as altering the number of model parts; contaminating the training data, and removing terms from the model.

5.3.1. Changing Scale of Features If the scale range of the saliency operator is changed, the set of regions extracted also changes, resulting in a different model being learnt. Figure 19 shows the effects of altering the feature scale on the face dataset. Figures 19(a) and (b) show a model using regions extracted using a hand selected scale range of 5–12 pixels in radius. The model picks out the eyes as well as the hairline. The standard scale range of 10-30 pixels in radius results in a model shown in Fig.19(c) and (d). The eyes are no longer picked up by the model, which relies entirely on the hairline. The models have similar error rates: 8.3% for the generic scale setting and 10.1% for the hand-tuned one.

This illustrates that while the output of the feature detector varies depending on the settings used, the ability of the learning algorithm to find consistency within whatever feature set it is given makes the algorithm as a whole less sensitive to the performance of the feature detector.

5.3.2. Feature Representation We now investigate different methods of representing the appearance of the regions within an image. In Section 5.2 we used a 15 dimensional gradient-based PCA approach with two additional dimensions capturing (i) the energy of the gradients in the regions and (ii) the residual between the original region and the point in PCA space. The 2nd column of Table 4 shows the classification

error rates for the five datasets. If the two extra dimensions are removed from the representation, then the error rates increase slightly, as seen in the 3rd column of Table 4. Fergus et al. (2003) used a 15 dimensional intensity based PCA representation. Its performance is compared to the two gradient based PCA approaches in the 4th column of Table 4. Here a generic PCA basis used for all categories, as opposed to the per-category PCA basis of Fergus et al. (2003).

Table 4. Classification results on five datasets for three different representations.

Dataset	Gradient PCA 15+2	Gradient PCA 15	Intensity PCA 15
Motorbikes	3.3	7.5	9.5
Faces	8.3	13.3	10.1
Airplanes	6.3	8.0	6.8
Spotted cats	11.0	11.0	13.5
Cars (rear)	8.8	11.5	7.8

5.3.3. Number of Parts in Model The number of parts within the model must be chosen beforehand and has an exponential effect on the learning time, despite the use of efficient search methods. Clearly, more parts gives more coverage of the object, but it makes the model slower to learn and introduces over-fitting problems, due to the increased numbers of parameters. Figure 20 shows that there is little improvement between $P = 6$ and $P = 7$. If it were possibleto investigate beyond

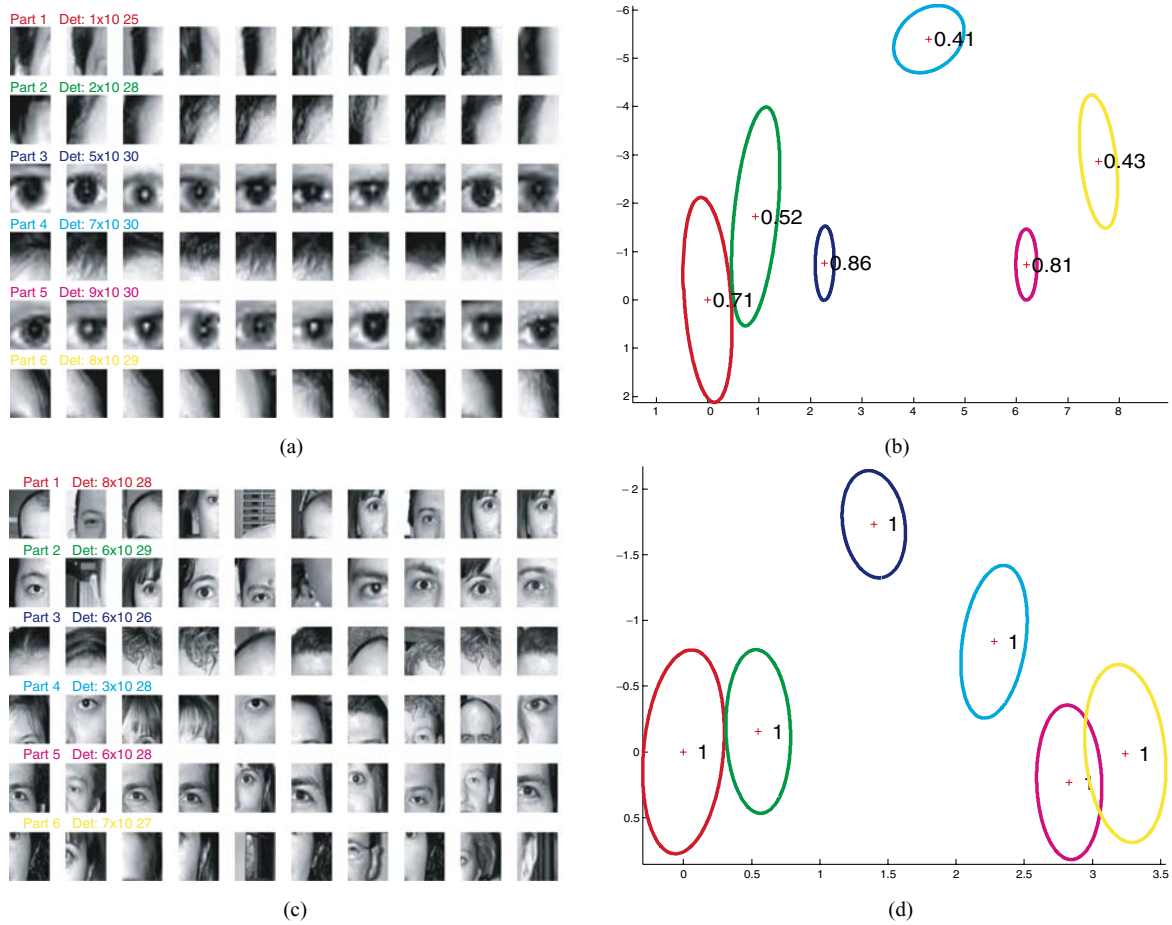


Figure 19. (a) & (b): A face model trained on images where the feature scale has been hand-tuned. The model captures both the hairline and eyes of the face. (c) & (d): A face model trained on regions extracted using a generic feature scale (as used in the experiments of Section 5.2. The de-tuned feature detector no longer picks out distinctive features such as the eyes. Comparing the determinants of the parts in each model reveal that the tuned model has tighter appearance densities.

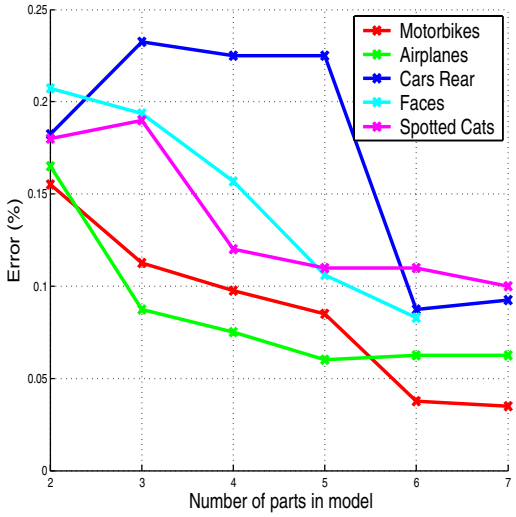


Figure 20. The effect of the number of parts, P, in the model versus error rate for the five datasets.

$P = 7$, no doubt the error rate would start to increase due to overfitting.

5.3.4. Contribution of the Different Model Terms. Figure 21 shows contribution of the shape and appearance terms to the performance of the model. The figure shows ROC curves evaluating classification performance and RPC evaluating localisation on the six datasets. A single model was learnt for each category, using a complete model with both shape and appearance, while in recognition, three different forms of the model were used: (i) the complete model, (ii) the complete model, but ignoring the appearance term and (iii) the complete model, but ignoring the shape term. Across all six datasets, it is noticeable that the appearance term is more important than the shape for classification. Only in the case of airplanes did the shape-only model perform well. Indeed, for cars (rear) the removal of the shape term actually improves the performance slightly. This is due to the ordering constraint imposed on the shape which in this case may be removing the best hypothesis from each frame. However, when the task is localisation, the shape-only model outperforms the appearance only one. In some cases the difference is dramatic, e.g. airplanes and cars (side).

When learning was performed with degraded models, the use of the appearance component alone produced a model with a performance close to that of the full model. However, when using only the shape component, the models frequently did not converge to a meaningful model, resulting in chance level classification performance.

5.3.5. Over-Fitting. The large number of parameters in the model, as shown in Table 1, means that a large number of images are needed in training to avoid overfitting the training data. Figure 22 shows that around 250 training images are needed for the model to generalize well. The use of priors in learning can dramatically reduce the training requirement on the number of images, down to just a handful or even one. See Fei-Fei et al. (2003) for a Bayesian extension to the constellation model, incorporating such priors in learning.

5.3.6. Contamination of the Training Set. Collecting hundreds of images of a given category of sufficient quality is a laborious time-consuming job. Being able to learn from datasets where some of the images consist only of clutter or are of insufficient quality is a useful property since it simplifies the task of building a training set. In Fig. 23 we show the results of deliberately introducing background images (e.g. images not containing an object belonging to the class that is being learnt) into a training set. For most of the datasets the drop in performance is small even with 50% background images in the training set. The model handles this level of contamination by interpreting the background images as ones where all the model parts are occluded. Notice that the graph bottoms out at around 20%–30% correct when the training set is entirely background images, while one would have expected 50% correct (i.e. chance performance). This implies that the background images contain some structure that is being learnt by the model, implying that our assumptions about the nature of the background are not strictly true (see Section 2). In Fergus et al. (2004), the problem of learning from contaminated data is investigated in more depth.

5.3.7. Sampling from the Model. Since our models are generative in nature we can sample from them. However, we are not able to directly sample back into pixel-space since the use of PCA for appearance introduces a non-probabilistic stumbling block. While we are able to draw samples within the PCA-space, there exists a many-to-one mapping from patch-space into the PCA-space. Additionally, the normalization of the

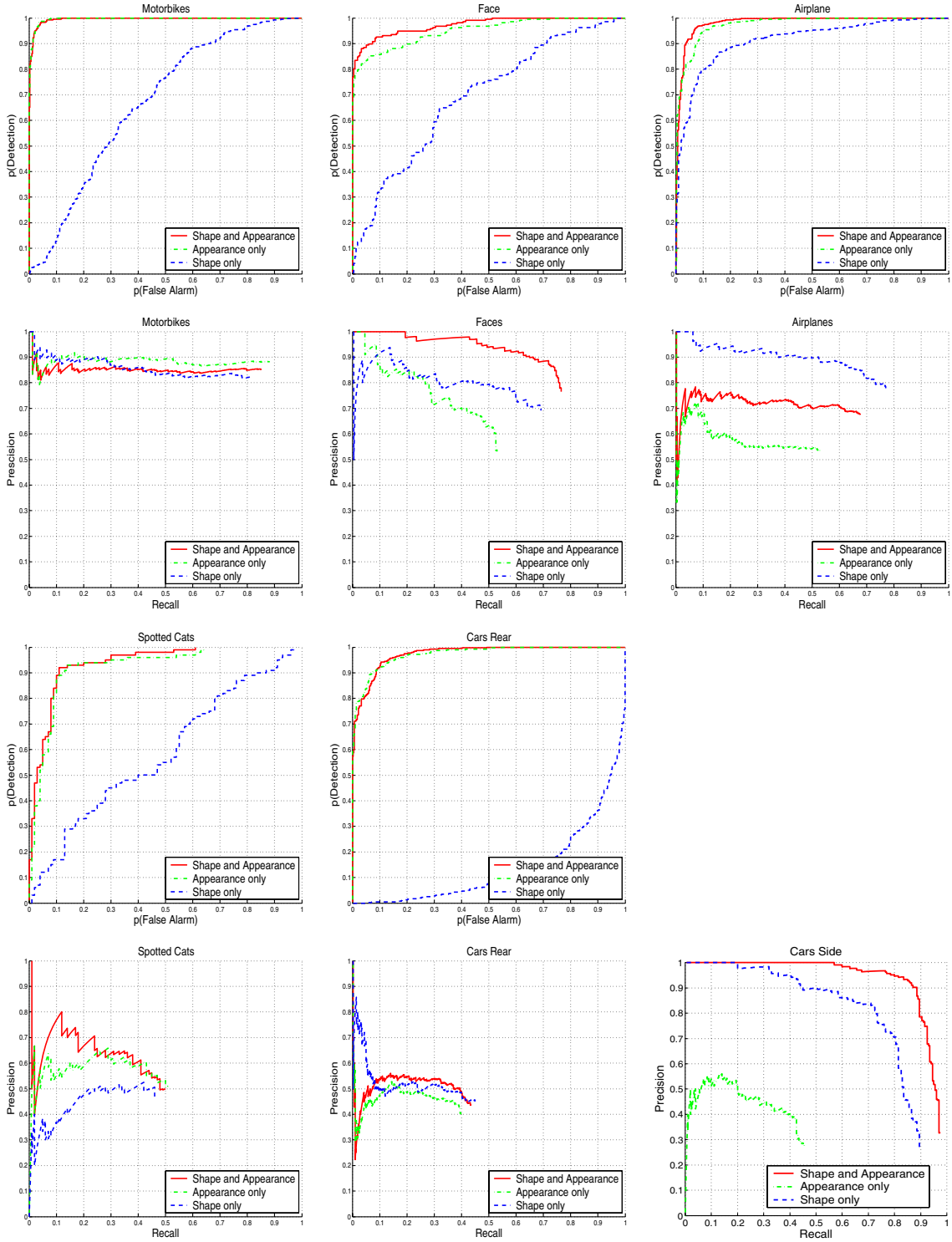


Figure 21. ROC and RPC curves for 6 different datasets. The effects of removing the shape or appearance terms are shown. The models rely heavily on appearance in a classification task (1st and 3rd rows). In detection, by contrast, the shape is more important than appearance (2nd and 4th rows).

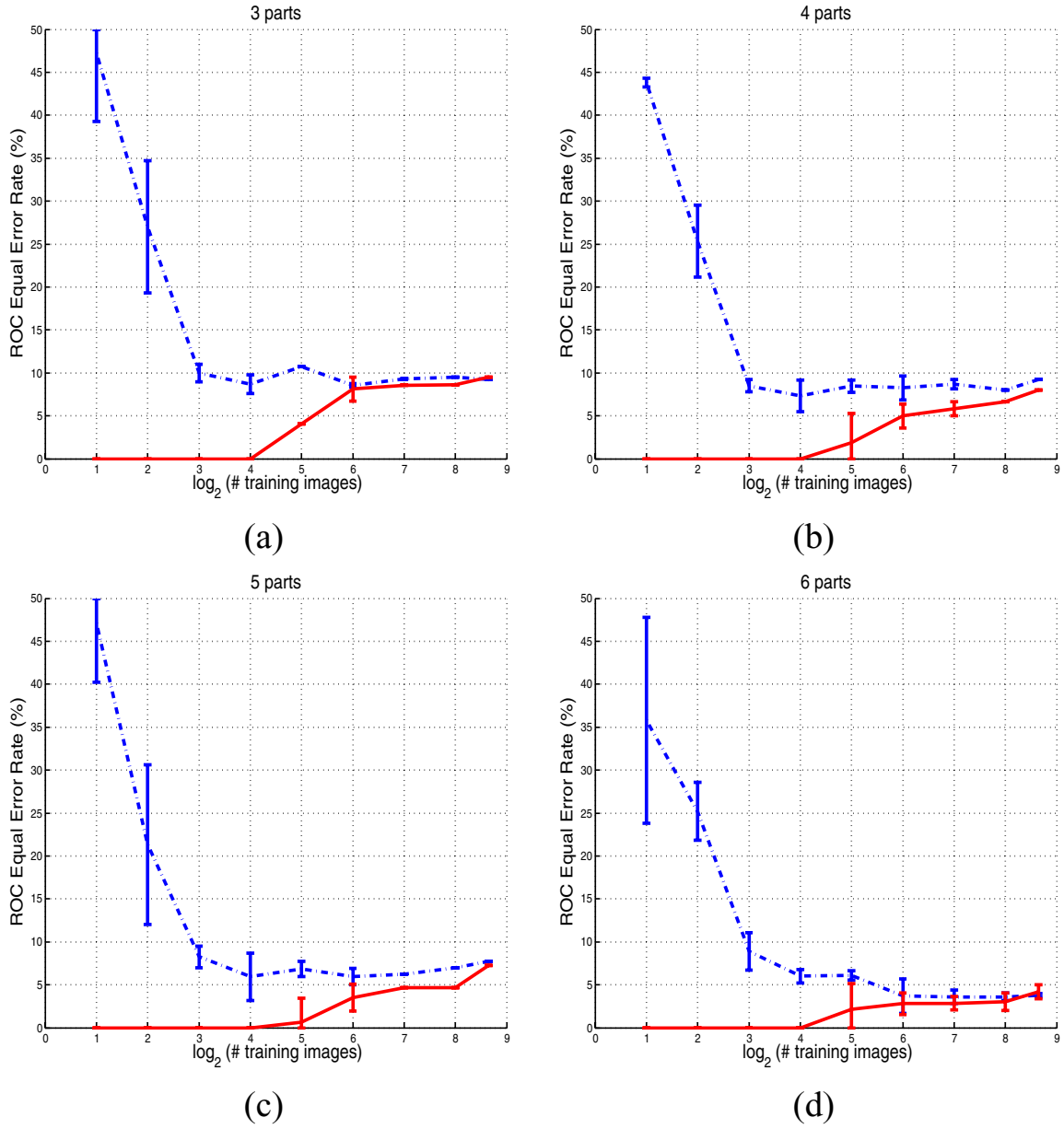


Figure 22. (a)–(d) The training (red) and test (blue dot-dashed) errors versus the number of training images for the motorbike dataset with models having (a) 3 parts, (b) 4 parts, (c) 5 parts, (d) 6 parts. Note the log scale on the x-axis. The curves converge to the Bayes' error once the number of training images is 256. The error bars show two standard deviations on performance, computed from 10 runs for 2–64 training images and 5 runs for 128–400 images.

patch introduces a similar many-to-one mapping problem. However, we can use one of two approximations: (i) Draw a sample from the appearance density and find the closest data point from all images and use its corresponding pixel patch or (ii) Form the patch directly from it's coordinates in the k dimensional PCA-space, assuming that all $121 - k$ coefficients are zero. Note that this will give a patch that is still normalized. In Fig. 24 we show samples drawn using method (i) above.

5.4. *Comparison with other methods*

We now compare our algorithm to a variety of different approaches in both classification and detection tasks. In the table in Fig. 25 we show results on the same datasets (with identical training and testing images) for the following algorithms: an earlier incarnation of the constellation model by Weber et al. (2000c) and Weber (2000); the region-based discriminative SVM-based method of Opelt et al. (2004); the bag-of-words pLSA approach of Sivic et al. (2005) and the Hough space voting scheme of Leibe et al. (2004). The table also lists the supervision required in training. The performance of the model can be seen to be comparable to the other approaches, beating the other methods in the majority of cases. The method of Leibe et al. (2004) achieves better performance; however, it requires the manual segmentation ofthe objects within the training

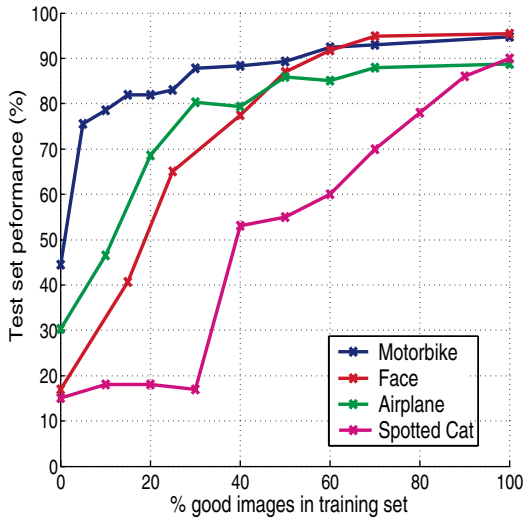


Figure 23. The effect of mixing background images into the training data, for 4 of the different datasets. With the exception of the leopard dataset, even with a 50-50 mix of images with/without objects, the resulting model error has only increased by a small margin, compared to training on uncontaminated data.

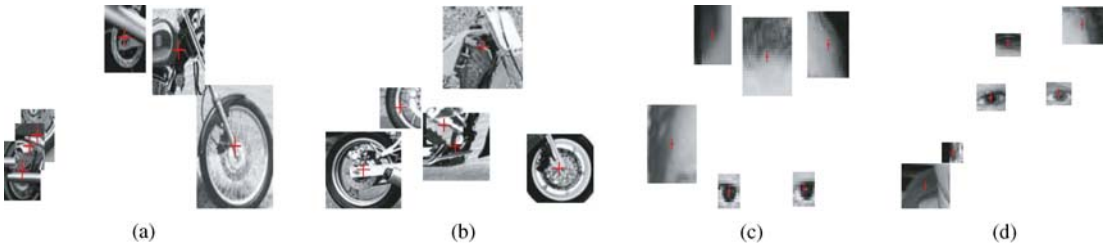
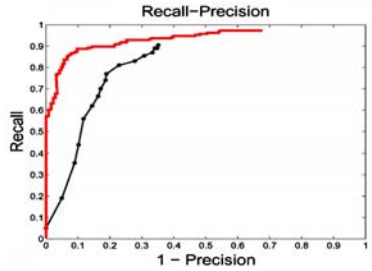


Figure 24. (a) & (b) Two samples drawn from a motorbike model. (c) & (d) Two samples drawn from face models.



Method	Ours	Weber [44, 41]	Opelt [32]	Sivic [36]	Leibe [26]
Supervision	I	I	I	N	I,S
Motorbikes	3.3	16.0	7.8	15.4	6.0
Faces	8.3	6.0	6.5	5.3	-
Airplanes	6.3	32.0	11.1	3.4	-
Spotted Cats	11.0	-	-	-	-
Cars Rear	8.8	-	8.9	21.4	6.1
Cars Side	11.5	-	17.0	-	3.0

Figure 25. Comparison to other methods Agarwal and Roth (2002); Leibe et al. (2004); Opelt et al. (2004); Sivic et al. (2005); Weber (2000); Weber et al. (2000c). The diagram on the left shows the RPC for Agarwal and Roth (2002) and our algorithm on the cars (side) dataset. On the right the table gives ROC equal error rates (except for the car (side) dataset where it is a recall-precision equal error) on a number of datasets. The second row shows the supervision required in training for each method: N-None; I-Image labels and flipping to give consistent viewpoint; S-Object segmentation.

images and has only been applied to a subset of the categories.

Figure 25 shows a recall-precision curve comparing the algorithm to Agarwal and Roth (2002), with our algorithm showing a distinct improvement. The superior performance of Leibe et al. (2004) on this dataset may be explained by the manual segmentation of the training images and the use of a validation scheme in their approach, which helps to eliminate false positives.

6. Conclusions and Further Work

We have proposed an improved learning scheme for the 'constellation model' which is scale-invariant and where shape and appearance are learnt simultaneously. We tested an implementation of such a scheme on six diverse datasets. We find that learning is robust with respect to clutter, scale variation and inter-object variability. Given exemplars of a similar pose, no futher human intervention is required to segment, normalize or otherwise preprocess the training data.

We have compared directly to the previous work on the constellation model of Weber et al. and shown an improvement in performance (see Fig. 25). We feel that the key improvement is that appearance and shape are learnt simultaneously, giving a cleaner learning scheme that avoids the reliance of first obtaining a good appearance model before the shape can be learnt. In the scheme of Weber et al., the clustering procedure to form a codebook of possibly useful features is a vital step. Examination of cases where the approach fails reveals that the codebook consists of a large number of generic features such as orientated edges which are not discriminative.

We have also compared to a wide variety of other approaches and baseline methods and shown the our approach compares favorably with them. The baseline methods reveal that our datasets are not overly challenging and therefore harder datasets should be collected.

We find that when our algorithm fails to detect the presence of an object, the most frequent cause of this false-reject error is that an insufficient number of features were detected by the front-end feature finder. It is clear that better feature finders as well as a more diverse array of features (e.g. sections of contours) should be employed. This is investigated in Fergus et al. (2005).

Other observations: (a) the system works well both on artificial (cars, planes, motorcycles) and natural (faces,cats) data; (b) while this paper uses many categories, it is clear that now we need to push forward with experiments involving hundreds of categories; (c) colour was not exploited and for some categories this may be important; (d) for classification, the appearance of parts seems more useful than their relative location, therefore some simplification of the shape term might reduce the computational complexity of both learning and recognition, as well as reducing the number of parameters in the model (see (Crandall et al., 2005; Fergus et al., 2005) for an investigation of this point); (e) we have made a number of design choices (size of patches, Gaussian distributions, parameterizations etc.) which were guided by common sense and need to be validated experimentally; (f) we find that our method is parsimonious in the number of training examples that are needed to train a model of a given complexity. However, it would be clearly advantageous to further reduce this number Fei-Fei et al. (2003).

A serious drawback of the current approach is that it is limited to a single viewpoint. Extensions to multiple views would be an important addition to the model. One approach is to use a mixture of constellation models in the manner of Weber et al. (2000), each mixture component handling a separate view.

There are two other areas where refinements of the model might be beneficial. The first is allowing a multimodal appearance density for each part. This will allow more complex appearances to be represented, for example faces with and without sunglasses. Second, we have built in scale-invariance, but greater invariance should also be possible. This would enable learning and recognition from images with much larger viewpoint variation. For example, similarity invariance can be achieved in recognition by searching over rotations.

Acknowledgments

We are indebted to Li Fei-Fei, David Lowe and Andrew Blake for their insights and suggestions. We also thank Timor Kadir for advice on the feature detector. D. Roth for providing the Cars (Side) dataset. Funding was provided by National Science Foundation Engineering Research Center for Neuromorphic Systems Engineering, the UK EPSRC, EC Project CogViSys and PASCAL Network of Excellence.

Notes

1. An implementation of this feature detector is available at http://www.robots.ox.ac.uk/∼timork/salscale.html
2. Recall is defined as the number of true positives over total positives in the data set, and precision is the number of true positives over the sum of false positives and true positives.

References

Agarwal, S., Awan, A., and Roth, D. 2002. Uiuc car dataset. http://l2r.cs.uiuc.edu/ cogcomp/Data/Car/.

Agarwal, S. and Roth, D. 2002. Learning a sparse representation for object detection. In *Proc. of the European Conference on Computer Vision*, pp. 113–130.

Amit, Y. and Geman, D. 1999. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715.

Borenstein, E., and Ullman, S. 2002. Class-specific, top-down segmentation. In *Proc. of the European Conference on Computer Vision*, pp. 109–124.

Burl, M., Weber, M., and Perona, P. 1998. A probabilistic approach to object recognition using local photometry and global geometry. In *Proc. of the European Conference on Computer Vision*, pp. 628–641.

Crandall, D., Felzenszwalb, P., and Huttenlocher, D. 2005. Spatial priors for part-based recognition using statistical models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, Vol. 1, pp. 10–17.

Csurka, G., Bray, C., Dance, C., and Fan, L. 2004. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, pp. 1–22.

Dempster, A., Laird, N., and Rubin, D. 1976. Maximum likelihood from incomplete data via the EM algorithm. *JRSS* **B**, 39:1–38.

Fei-Fei, L., Fergus, R. and Perona, P. 2003. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proc. of the 9th International Conference on Computer Vision, Nice, France*, pp. 1134–1141.

Felzenszwalb, P., and Huttenlocher, D. 2000. Pictorial structures for object recognition. In *Proc. of the IEEE Conference on Omputer Vision and Pattern Recogniion*, pp. 2066–2073.

Fergus., R. 2005. *Visual Object Category Recognition*. PhD thesis, University of Oxford, UK.

Fergus, R., and Perona, P. 2003. Caltech object category datasets. http://www.vision.caltech.edu/html-files/archive.html.

Fergus, R., Perona, P., and Zisserman, A. 2004. A visual category filter for google images. In *Proceedings of the 8th European Conference on Computer Vision, Prague, Czech Republic*, Springer-Verlag, pp. 242–256.

Fergus, R., Perona, P., and Zisserman, A. 2005. A sparse object category model for efficient learning and exhaustive recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, San Diego*, vol. 1, pp. 380–387.

Fergus, R., Perona, P., and Zisserman, P. 2003. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*.

Fergus, R., Weber, M. and Perona, P. 2001. Efficient methods for object recognition using the constellation model. Technical report, California Institute of Technology.

Forsyth, D.A. and Ponce, J. 2002. *Computer Vision: A Modern Approach*. Prentice Hall.

Grimson, W.E.L., and Lozano-Pérez, T. 1987. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(4):469–482.

Hart, P., Nilsson, N., and Raphael, B. 1968. A formal basis for the determination of minimum cost paths. *IEEE Transactions on SSC*, 4:100–107.

Heisele, B., Serre, T., Pontil, M., Vetter, T., and Poggio, T. 2002. Categorization by learning and combining object parts. In *Advances in Neural Information Processing Systems 14,* Vancouver, Canada, vol. 2, pp. 1239–1245.

Jerrum, M. and Sinclair, A. 1997. The Markov chain Monte Carlo method. In D. S. Hochbaum, (ed.), *Approximation Algorithms for NP-hard Problems*. PWS Publishing, Boston.

Jurie, F. and Schmid, C. 2004. Scale-invariant shape features for recognition of object categories. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, Washington, DC, pp. 90–96.

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

Ke, Y. and Sukthankar, R. 2004. PCA-SIFT: A more distinctive representation for local image descriptors. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition,* Wasington, DC.

LeCun, Y., Huang, F. and Bottou, L. 2004. Learning methods for generic object recognition with invariance to pose and lighting. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition,* Wasington, DC, IEEE Press.

Leibe, B., Leonardis, A. and Schiele, B. 2004. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*.

Leung, T., Burl, M. and Perona, P. 1998. Probabilistic affine invariants for recognition. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 678–684.

Lindeberg., T. 1998. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 30(2):77–116.

Lowe, D.G. 1985. *Perceptual Organization and Visual Recognition*. Kluwer Academic Publishers.

Mardia, K.V. and Dryden, I.L. 1989. Shape distributions for landmark data. *Adv. Appl. Prob.*, 21:742–755.

Mikolajczyk, K. and Schmid, C. 2001. Indexing based on scale invariant interest points. In *Proc. of the 8th International Conference on Computer Vision,* Vancouver, Canada, pp. 525–531.

Opelt, A., Fussenegger, A., and Auer, P. 2004. Weak hypotheses and boosting for generic object detection and recognition. In *Proc. of the 8th International Conference on Computer Vision,* Prague, Czech Republic, 2004.

Rowley, H., Baluja, S., and Kanade, T. 1998. Neural network-based face detection. *IEEE PAMI*, 20(1):23–38.

Schmid, C. 2001. Constructing models for content-based image retrieval. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 39–45.

Schneiderman, H. and Kanade, T. 2000. A statistical approach to 3D object detection applied to faces and cars. In *Proc. Computer Vision and Pattern Recognition*, pp. 746–751.

Sivic, J., Russell, B., Efros, A., Zisserman, A. and Freeman, W. 2005. Discovering object categories in image collections. Technical Report A. I. Memo 2005-005, Massachusetts Institute of Technology.

Sung, K. and Poggio, T. 1998. Example-based learning for view-based human face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):39–51.

Thureson, J. and Carlsson, S. 2004. Appearance based qualitative image description for object class recognition. In *Proc. of the 8th European Conference on Computer Vision,* Prague, Czech Republic, pp. 518–529.

Torralba, A., Murphy, K.P., and Freeman, W.T. 2004. Sharing features: efficient boosting procedures for multiclass object detection. In *Proc. of the 8th European Conference on Computer Vision,* Prague, Czech Republic, pp. 762–769.

Viola, P. and Jones, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 511–518.

Weber, M. 2000. Unsupervised learning of models for object recognition. PhD thesis, California Institute of Technology, Pasadena, CA.

Weber, M. Einhauser, W. Welling, M., and Perona, P. 2000. Viewpoint-invariant learning and detection of human heads. In *Proc. 4th IEEE Int. Conf. Autom. Face and Gesture Recog., FG2000*, pp. 20–27.

Weber, M., Welling, M. and Perona, P. 2000. Towards automatic discovery of object categories. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 101–109.

Weber, M., Welling, M., and Perona, P. 2000. Unsupervised learning of models for recognition. In *Proc. of the European Conference on Computer Vision*, pp. 18–32.