Object Class Recognition by Unsupervised Scale-Invariant Learning

R. Fergus[1] P. Perona[2] A. Zisserman[1]

[1] Dept. of Engineering Science
University of Oxford
Parks Road, Oxford
OX1 3PJ, U.K.

{fergus,az}@robots.ox.ac.uk

[2] Dept. of Electrical Engineering
California Institute of Technology
MC 136-93, Pasadena
CA 91125, U.S.A.

perona@vision.caltech.edu

Abstract

We present a method to learn and recognize object class models from unlabeled and unsegmented cluttered scenes in a scale invariant manner. Objects are modeled as flexible constellations of parts. A probabilistic representation is used for all aspects of the object: shape, appearance, occlusion and relative scale. An entropy-based feature detector is used to select regions and their scale within the image. In learning *the parameters of the scale-invariant object model are estimated. This is done using expectation-maximization in a maximum-likelihood setting. In* recognition*, this model is used in a Bayesian manner to classify images. The flexible nature of the model is demonstrated by excellent results over a range of datasets including geometrically constrained classes (e.g. faces, cars) and flexible objects (such as animals).*

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 the training sets should be small and that the operator-assisted steps that are required (e.g. elimination of clutter in the background of the object, scale normalization of the training sample) should be reduced to a minimum or eliminated.

The problem of describing and recognizing categories, as opposed to specific objects (e.g. [6, 9, 11]), has recently gained some attention in the machine vision literature [1, 2, 3, 4, 13, 14, 19] with an emphasis on the detection of faces [12, 15, 16]. There is broad agreement on the issue of representation: object categories are represented as collection of features, or parts, each part has a distinctive appearance and spatial position. Different authors vary widely on the details: the number of parts they envision (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 often proceeds by an exhaustive search over image position and scale.

We focus our attention on the probabilistic approach proposed by Burl *et al.* [4] 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.* [18, 19] proposed a maximum likelihood unsupervised learning algorithm for the constellation model which successfully learns object categories from cluttered data with minimal human intervention. We propose here a number of substantial improvement 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

Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)
1063-6919/03 $17.00 © 2003 IEEE

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 [8, 10]. Fourthly, 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. A final 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 figure 1.

2. Approach

Our approach to modeling object classes follows on from the work of Weber *et al.* [17, 18, 19]. An object model consists of a number of parts. Each part has an appearance, relative scale and can be be occluded or not. 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. 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 in a Bayesian manner, using the model parameters estimated in the learning.

The model, region detector, and representation of appearance are described in detail in the following subsections.

2.1. Model structure

The model is best explained by first considering recognition. We have learnt a generative object class model, with P parts and parameters θ. We are then presented with a new image and we must decide if it contains an instance of our object class or not. In this query image we have identified N interesting features with locations $\mathbf{X}$, scales $\mathbf{S}$, and appearances $\mathbf{A}$. We now make a Bayesian decision, R:

$$\begin{aligned} R &= \frac{p(\text{Object}|\mathbf{X},\mathbf{S},\mathbf{A})}{p(\text{No object}|\mathbf{X},\mathbf{S},\mathbf{A})} \\ &= \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{Object})\;p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\text{No object})\;p(\text{No object})} \\ &\approx \frac{p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta)\;p(\text{Object})}{p(\mathbf{X},\mathbf{S},\mathbf{A}|\theta_{bg})\;p(\text{No object})} \end{aligned}$$

The last line is an approximation since we will only use a single value for θ (the maximum-likelihood value) rather than integrating over $p(\theta)$ as we strictly should. Likewise, we assume that all non-object images can also be modeled by a background with a single set of parameters θ_{bg}. The ratio of the priors may be estimated from the training set or set by hand (usually to 1). Our decision requires the calculation of the ratio of the two likelihood functions. In order to do this, the likelihoods may be factored as follows:

$$p(\mathbf{X},\mathbf{S},\mathbf{A}|\,\theta) = \sum_{\mathbf{h}\in H} p(\mathbf{X},\mathbf{S},\mathbf{A},\mathbf{h}|\,\theta) =$$

$$\sum_{\mathbf{h}\in H} \underbrace{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}_{Appearance}\underbrace{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}_{Shape}\underbrace{p(\mathbf{S}|\mathbf{h},\theta)}_{Rel.\ Scale}\underbrace{p(\mathbf{h}|\theta)}_{Other}$$

Since our model only has P (typically 3-7) parts but there are N (up to 30) features in the image, we introduce an indexing variable $\mathbf{h}$ which we call a *hypothesis*. $\mathbf{h}$ is a vector of length P, where each entry is between 0 and N which allocates a particular feature to a model 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 $O(N^P)$.

In the following we sketch the form for likelihood ratios of each of the above factored terms. Space prevents a full derivation being included, but the full expressions follow from the methods of [17]. 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), $n = N - sum(\mathbf{d})$ (the number of background features under the current hypothesis), and $f = sum(\mathbf{d})$ which is the number of foreground features.

Appearance. Each feature's appearance is represented as a point in some appearance space, defined below. Each part p has a Gaussian density within this space, with mean and covariance parameters $\theta_p^{app} = \{\mathbf{c_p}, V_p\}$ which is independent of other parts' densities. The background model has parameters $\theta_{bg}^{app} = \{\mathbf{c_{bg}}, V_{bg}\}$. Both V_p and V_{bg} are assumed to be diagonal. Each feature selected by the hypothesis is evaluated under the appropriate part density. All features not selected by the hypothesis are evaluated under the background density. The ratio reduces to:

$$\frac{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta)}{p(\mathbf{A}|\mathbf{X},\mathbf{S},\mathbf{h},\theta_{bg})} = \prod_{p=1}^{P}\left(\frac{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_p, V_p)}{\mathrm{G}(\mathbf{A}(h_p)|\mathbf{c}_{bg}, V_{bg})}\right)^{d_p}$$

where G is the Gaussian distribution, and d_p is the pth entry of the vector $\mathbf{d}$, i.e. $d_p = \mathbf{d}(p)$. 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 ($d_p = 0$).

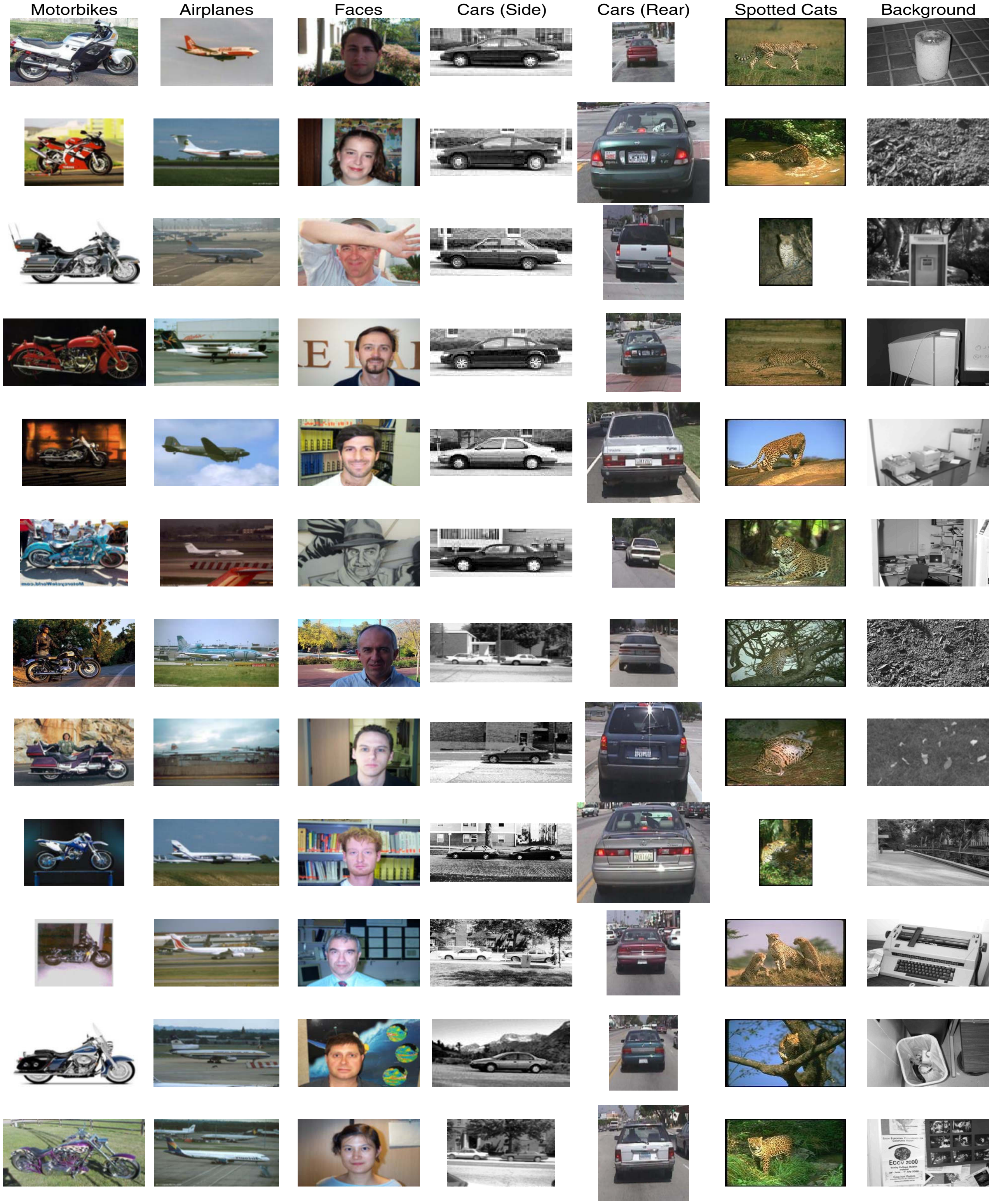

Figure 1: Some sample images from the datasets. Note the large variation in scale in, for example, the cars (rear) database. These datasets are from http://www.robots.ox.ac.uk/∼vgg/data/, except for the Cars (Side) from (http://l2r.cs.uiuc.edu/∼cogcomp/index_research.html) and Spotted Cats from the Corel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

Shape. The shape is represented by a joint Gaussian density of the locations of features within a hypothesis, once they have been transformed into a scale-invariant space. This is done using the scale information from the features in the hypothesis, so avoiding an exhaustive search over scale that other methods use. The density has parameters $\theta^{shape} = \{\boldsymbol{\mu}, \Sigma\}$. Note that, unlike appearance whose covariance matrices V_p, V_{bg} are diagonal, Σ 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. If a part is occluded it is integrated out of the joint foreground density.

$$\frac{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta)}{p(\mathbf{X}|\mathbf{S},\mathbf{h},\theta_{bg})} = \mathrm{G}(\mathbf{X}(\mathbf{h})|\boldsymbol{\mu},\Sigma)\,\alpha^f$$

Relative scale. The scale of each part p relative to a reference frame is modeled by a Gaussian density which has parameters $\theta^{scale} = \{t_p, U_p\}$. The parts are assumed to be independent to one another. The background model assumes a uniform distribution over scale (within a range r).

$$\frac{p(\mathbf{S}|\mathbf{h},\theta)}{p(\mathbf{S}|\mathbf{h},\theta_{bg})} = \prod_{p=1}^{P} \mathrm{G}(\mathbf{S}(h_p)|t_p,U_p)^{d_p}\, r^f$$

Occlusion and Statistics of the feature finder.

$$\frac{p(\mathbf{h}|\theta)}{p(\mathbf{h}|\theta_{bg})} = \frac{p_{Poiss}(n|M)}{p_{Poiss}(N|M)}\,\frac{1}{^nC_r(N,f)}\,p(\mathbf{d}|\theta)$$

The first term models the number of features detected using a Poisson distribution, which has a mean M. The second is a book-keeping term for the hypothesis variable and the last is a probability table (of size 2^P) for all possible occlusion patterns and is a parameter of the model.

The model of Weber *et al.* contains the shape and occlusion terms to which we have added the appearance and relative scale terms. Since the model encompasses many of the properties of an object, all in a probabilistic way, 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). From the equations above we can now calculate the overall likelihood ratio from a given set of $\mathbf{X}, \mathbf{S}, \mathbf{A}$. The intuition is that the majority of the hypotheses will be low scoring as they will be picking up features from background junk 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.

2.2. Feature detection

Features are found using the detector of Kadir and Brady [7]. This method finds regions that are salient over both location and scale. For each point on 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 [7, 8]). The N regions with highest saliency over the image provide the features for learning and recognition. Each feature is defined by its centre and radius (the scale).

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 extrema in entropy. There is an entropy extrema 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-class 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.

2.3. Feature representation

The feature detector identifies regions of interest on 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 two typical images from the motorbike dataset.

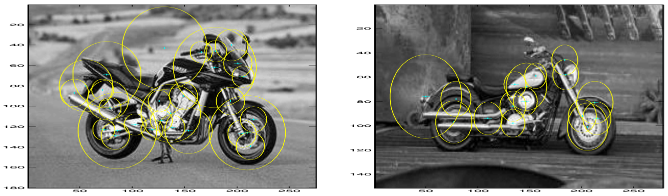

Figure 2: Output of the feature detector

Once the regions are identified, they are cropped from the image and rescaled to the size of a small (typically 11×11) pixel patch. Thus, each patch exists in a 121 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 121-dimensional Gaussian is unmanageable from a numerical point of view and also the number of parameters involved (242 per model part) are too many to be estimated.

This is done by using principal component analysis (PCA). In the learning stage, we collect the patches from all images and perform PCA on them. Each patch's appearance is then a vector of the coordinates within the first k (typically 10-15) principal components, so giving us $\mathbf{A}$. This gives a good reconstruction of the original patch whilst using a moderate number of parameters per part (20-30). 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.

2.4. Learning

The task of learning is to estimate the parameters $\theta = \{\boldsymbol{\mu}, \Sigma, \mathbf{c}, V, M, p(\mathbf{d}|\theta), t, U\}$ of the model 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: $\hat{\theta}_{ML} = arg \underset{\theta}{max}\ p(\mathbf{X}, \mathbf{S}, \mathbf{A}|\, \theta)$.

Learning is carried out using the expectation-maximization (EM) algorithm [5] which iteratively converges, from some random initial value of θ to a maximum (which might be a local one). It works in two stages; the E-step in which, given the current value of θ, some statistics are computed and the M-step in which the current value of θ is updated using the statistics from the E-step. The process is then repeated until convergence. The scale information from each feature allows us to learn the model shape in a scale-invariant space. Since the E-step involves evaluating the likelihood for each hypothesis and there being $O(N^P)$ of them per image, efficient search methods are needed. A^* and space-search methods are used, giving a considerable performance improvement. Despite these methods, a P = 6-7 part model with N = 20-30 features per image (a practical maximum), using 400 training images, takes around 24-36 hours to run. Learning complex models such as these has certain difficulties. Table 1 illustrates how the number of parameters in the model grows with the number of parts, (assuming k = 15). To avoid over-fitting data, large datasets are used (up to 400 images in size). Surprisingly, given the complexity of the search space, the algorithm is remarkable consistent in it's convergence, so much so that validation sets were not necessary. Initial conditions were chosen randomly within a sensible range and convergence usually occurred within 50-100 EM iterations. Using a typical 6 part model on a typical dataset (Motorbikes), 10 runs from different initial conditions gave the same classification performance for 9 of them, with the other showing a difference of under 1%. 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.

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

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

2.5. Recognition

Recognition proceeds by first detecting features, and then evaluating these features using the learnt model, as described in section 2.1. By calculating the likelihood ratio, R, and comparing it to a threshold; the presence or absence of the object within the image may be determined. In recognition, as in learning, efficient search techniques are used since large N and P mean around 2-3 seconds are taken per image. It is also possible to search reliably for more than one instance in an image, as needed for the Cars (Side) dataset.

3. Results

Experiments were carried out as follows: each dataset was split randomly into two separate sets of equal size. The model was then trained on the first and tested on the second. In recognition, the decision was (as described above) a simple object present/absent one, except for the cars (side) dataset where multiple instances of the object were to be found. The performance figures quoted are the receiver-operating characteristic (ROC) equal error rates (i.e. p(True positive)=1-p(False positive)) testing against the background dataset. For example a figure of 91% means that 91% of the foreground images were correctly classified but 9% of the background images were incorrectly classified (i.e. thought to be foreground). A limited amount of preprocessing was performed on some of the datasets. For the motorbikes and airplanes the images were flipped to ensure the object was facing the same way. The spotted cat dataset was 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.

There were two phases of experiments. In the first those datasets with scale variability were normalized so that the objects were of uniform size. The algorithm was then evaluated on the datasets and compared to other approaches. In the second phase the algorithm was run on the datasets containing scale variation and the performance compared to the scale-normalized case.

In all the experiments, the following parameters were used: $k = 15$, $P = 6$ and on average $N = 20$. The only parameter that was adjusted at all in all the following experiments was the scale over which features were found. The standard setting was $4 - 60$ pixels but for the scale-invariant experiments, this was changed to account for the wider scale variation in features.

Figures 5-8 show models and test images for four of the datasets. Notice how each model captures the essence, 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. 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.

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 figure 5). 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.

Figure 3 shows a recall-precision curve[1] (RPC) and a table comparing the algorithm to previous approaches to object class recognition [1, 17, 19]. In all cases the performance of the algorithm is superior to the earlier methods, despite not being tuned for a particular dataset.

Figure 4(a) illustrates the algorithm performs well even when the signal-to-noise ratio is degraded by introducing background images into the training set and Fig. 4(b) shows how variation in the number of parts affects performance.

Dataset	Ours	Others	Ref.
Motorbikes	92.5	84	[17]
Faces	96.4	94	[19]
Airplanes	90.2	68	[17]
Cars(Side)	88.5	79	[1]

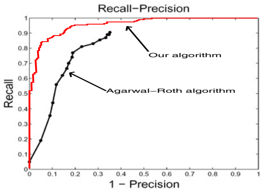

Figure 3: Comparison to other methods [1, 17, 19]. The diagram on the right shows the RPC for [1] and our algorithm on the cars (side) dataset. On the left the table gives ROC equal error rates (except for the car (side) dataset where it is a recall-precision equal error) on a number of datasets. The errors for our algorithm are at least half those of the other methods, except for the face dataset.

[1] 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.

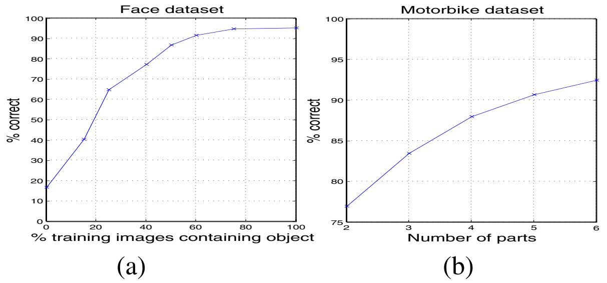

Figure 4: (a) shows the effect of mixing background images into the training data (in this case, the face dataset). Even with a 50-50 mix of images with/without objects, the resulting model error is a tolerable 13%. In (b), we see how the performance drops off as the number of parts in the model is reduced.

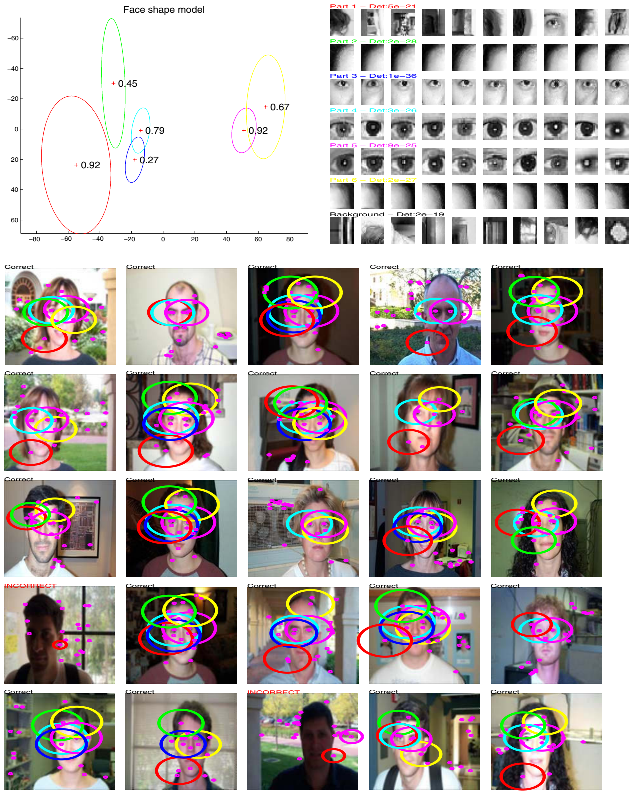

Figure 5: A typical face model with 6 parts. The top left figure shows the shape model. The ellipses represent the variance of each part (the covariance terms cannot easily be shown) and the probability of each part being present is shown just to the right of the mean. The top right figure shows 10 patches closest to the mean of the appearance density for each part and the background density, along with the determinant of the variance matrix, so as to give an idea as to the relative tightness of each distribution. Below these two are some sample test images, with a mix of correct and incorrect classifications. 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 score of the hypothesis (the bigger the better).

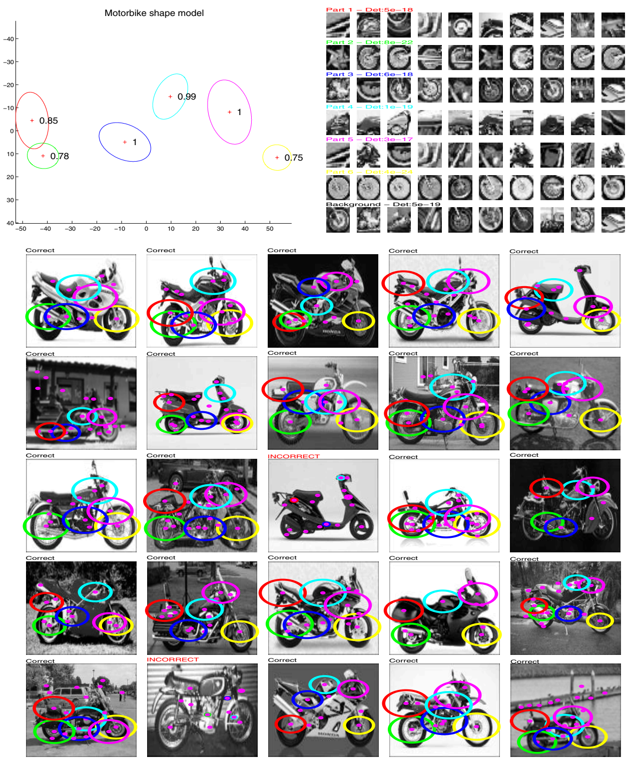

Figure 6: A typical motorbike model with 6 parts. Note the clear identification of the front and rear wheels, along with other parts such as the fuel tank.

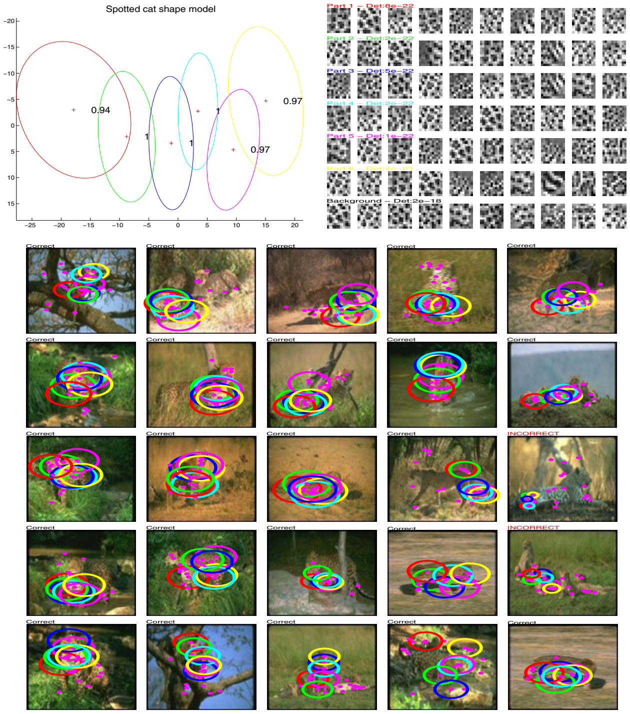

Figure 7: A typical spotted cat model with 6 parts. Note the loose shape model but distinctive "spotted fur" appearance.

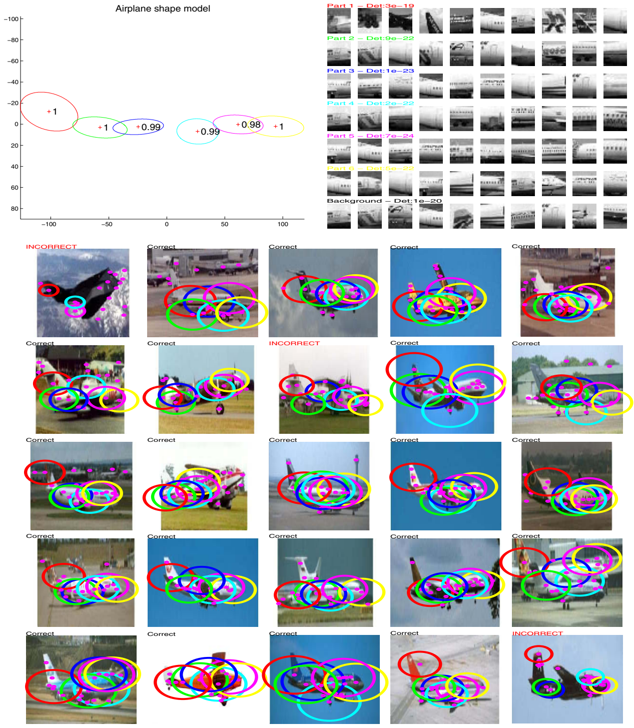

Figure 8: A typical airplane model with 6 parts.

Dataset	Total size of dataset	Object width (pixels)	Motorbike model	Face model	Airplane model	Cat model
Motorbikes	800	200	92.5	50	51	56
Faces	435	300	33	96.4	32	32
Airplanes	800	300	64	63	90.2	53
Spotted Cats	200	80	48	44	51	90 .0

Table 2: A confusion table for a number of datasets. The diagonal shows the ROC equal error rates on test data across four categories, where the algorithm's parameters were kept *exactly* the same, despite a range of image sizes and object types. The performance above can be improved dramatically (motorbikes increase to 95.0%, airplanes to 94.0% and faces to 96.8%) if feature scale is adjusted on a per-dataset basis. The off-diagonal elements demonstrate how good, for example, a motorbike model is at distinguishing between spotted cats and background images: 48% - at the level of chance. So despite the models being inherently generative, they perform well in a distinctive setting.

Table 2 shows the performance of the algorithm across the four datasets, with the learnt models illustrated in figures 5-8. Exactly the same algorithm settings are used for all models. Note that the performance is above 90% for all four datasets. In addition, the table shows the confusion between models which is usually at the level of chance.

Table 3 compares the performance of the scale-invariant models on unscaled images to that of the scale-variant models on the pre-scaled data. It can be seen that the drop in performance is marginal despite a wide range of object scales. In the case of the cars (rear) dataset, there is a significant improvement in performance with the scale-invariant model. This is due to the feature detector performing badly on small images ($<$ 150 pixels) and in the pre-scaled case, all were scaled down to 100 pixels. Figure 9 shows the scale-invariant model for this dataset. This dataset was tested against background road scenes (rather than the background images, examples of which are in Fig. 1) to make a more realistic experiment.

Dataset	Total size of dataset	Object size range (pixels)	Pre-scaled performance	Unscaled performance
Motorbikes	800	200-480	95.0	93.3
Airplanes	800	200-500	94.0	93.0
Cars (Rear)	800	100-550	84.8	90.3

Table 3: Results for scale-invariant learning/recognition.

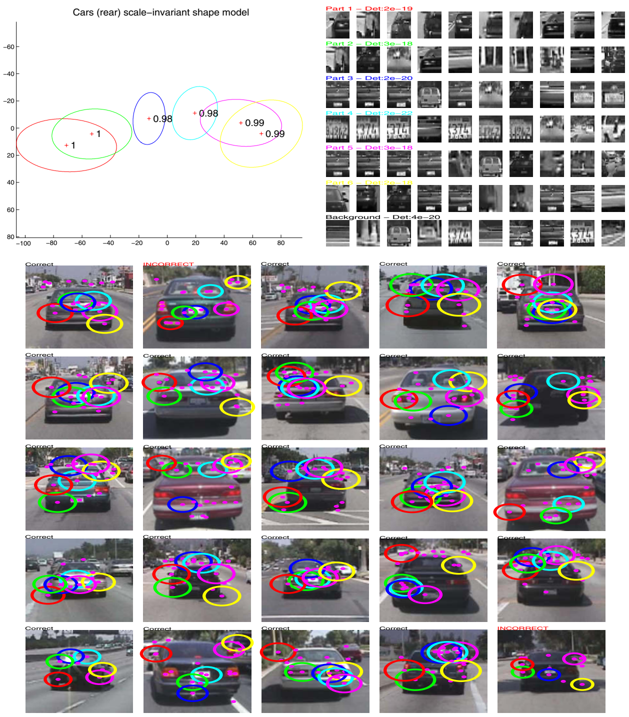

Figure 9: A scale-invariant car model with 6 parts.

4. Conclusions and Further work

The recognition results presented here convincingly demonstrate the power of the constellation model and the associated learning algorithm: the same piece of code performs well (less than 10% error rate) on six diverse object categories presenting a challenging mixture of visual characteristics. Learning is achieved without any supervision on datasets that contain a wide range of scales as well as clutter.

Currently, the framework is heavily dependent on the feature detector picking up useful features on the object. We are addressing this by extending the model to incorporate several classes of feature, e.g. edgels. Other than this there are two areas where improvements will be very beneficial. The first is in a further generalization of the model structure to have a multi-modal appearance density with a single shape distribution. This will allow more complex appearances to be represented, for example faces with and without sunglasses. Second, we have built in scale-invariance, but full affine-invariance should also be possible. This would enable learning and recognition from images with much larger viewpoint variation.

Acknowledgements

Timor Kadir for advice on the feature detector. D. Roth for providing the Cars (Side) dataset. Funding was provided by CNSE, the UK EPSRC, and EC Project CogViSys.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. In *Proc. ECCV*, pages 113–130, 2002.

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

[3] E. Borenstein. and S. Ullman. Class-specific, top-down segmentation. In *Proc. ECCV*, pages 109–124, 2002.

[4] M. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proc. ECCV*, pages 628–641, 1998.

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

[6] W. E. L. Grimson. *Object Recognition by Computer, The Role of Geometric Constraints*. MIT Press, 1990.

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

[8] T. Lindeberg. Feature detection with automatic scale selection. *IJCV*, 30(2):77–116, 1998.

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

[10] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, 2001.

[11] C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. *IJCV*, 16(2), 1995.

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

[13] C. Schmid. Constructing models for content-based image retrieval. In *Proc. CVPR*, volume 2, pages 39–45, 2001.

[14] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. CVPR*, 2000.

[15] K. Sung and T. Poggio. Example-based learning for view-based human face detection. *IEEE PAMI*, 20(1):39–51, Jan 1998.

[16] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages 511–518, 2001.

[17] M. Weber. *Unsupervised Learning of Models for Object Recognition*. PhD thesis, California Institute of Technology, Pasadena, CA, 2000.

[18] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. In *Proc. CVPR*, June 2000.

[19] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. ECCV*, pages 18–32, 2000.