A Visual Category Filter for Google Images

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

[1] Dept. of Engineering Science,
University of Oxford, Parks Road,
Oxford, OX1 3PJ, UK.
{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 extend the *constellation model* to include heterogeneous parts which may represent either the appearance or the geometry of a region of the object. The parts and their spatial configuration are learnt simultaneously and automatically, without supervision, from cluttered images.
We describe how this model can be employed for ranking the output of an image search engine when searching for object categories. It is shown that visual consistencies in the output images can be identified, and then used to rank the images according to their closeness to the visual object category.
Although the proportion of good images may be small, the algorithm is designed to be robust and is capable of learning in either a totally unsupervised manner, or with a very limited amount of supervision.
We demonstrate the method on image sets returned by Google's image search for a number of object categories including bottles, camels, cars, horses, tigers and zebras.

1 Introduction

Just type a few keywords into the Google image search engine, and hundreds, sometimes thousands of pictures are suddenly available at your fingertips. As any Google user is aware, not all the images returned are related to the search. Rather, typically more than half look completely unrelated; moreover, the useful instances are not returned first – they are evenly mixed with unrelated images. This phenomenon is not difficult to explain: current Internet image search technology is based upon words, rather than image content – the filename of the image and text near the image on a web-page [4]. These criteria are effective at gathering quickly related images from the millions on the web, but the final outcome is far from perfect.

We conjecture that, even without improving the search engine per se, one might improve the situation by measuring 'visual consistency' amongst the images that are returned and re-ranking them on the basis of this consistency, so increasing the fraction of good images presented to the user within the first few web pages. This conjecture stems from the observation that the images that are related to the search typically are

T. Pajdla and J. Matas (Eds.): ECCV 2004, LNCS 3021, pp. 242–256, 2004.

visually similar, while images that are unrelated to the search will typically look different from each other as well.

How might one measure 'visual consistency'? One approach is to regard this problem as one of probabilistic modeling and robust statistics. One might try and fit the data (the mix of images returned by Google) with a parametrized model which can accommodate the within-class variation in the requested category, for example the various shapes and labels of bottles, while rejecting the outliers (the irrelevant images). Learning a model of the category under these circumstances is an extremely challenging task. First of all: even objects within the same category do look quite different from each other. Moreover, there are the usual difficulties in learning from images such as lighting and viewpoint variations (scale, foreshortening) and partial occlusion. Thirdly, and most importantly, in the image search scenario the object is actually only present in a sub-set of the images, and this sub-set (and even its size) is unknown.

While methods exist to model object categories [9,13,15], it is essential that the approach can learn from a contaminated training set with a minimal amount of supervision. We therefore use the method of Fergus *et al.* [10], extending it to allow the parts to be heterogeneous, representing a region's appearance or geometry as appropriate. The model and its extensions are described in section 2. The model was first introduced by Burl *et al.* [5]. Weber *et al.* [23] then developed an EM-based algorithm for training the model on cluttered datasets with minimal supervision. In [10] a probabilistic representation for part appearance was developed; the model made scale invariant; and both appearance and shape learnt simultaneously.

Other approaches to this problem [7,19] use properties of colour or texture histograms. While histogram approaches have been successful in Content Based Image Retrieval [2,12,21], they are unsuitable for our task since the within-class returns vary widely in colour and texture.

We explore two scenarios: in the first the user is willing to spend a limited amount of time (e.g. 20-30 seconds) picking a handful of images of which they want more examples (a simple form of relevance feedback [20]); in the second the user is impatient and there is no human intervention in the learning (i.e. it is completely unsupervised).

Since the model only uses visual information, a homonymous category (one that has multiple meanings, for example "chips" would return images of both "French fries" and "microchips") pose problems due to multiple visual appearances. Consequently we will only consider categories with one dominant meaning in this paper. The algorithm only requires images as its input, so can be used in conjunction with any existing search engine. In this paper we have chosen to use Google's image search.

2 The Model

In this section we give an overview of our previously developed method [10], together with the extension to heterogeneous parts.

An object model consists of a number of parts which are spatially arranged over the object. A part here may be a patch of pixels or a curve segment. In either case, a part is represented by its intrinsic description (appearance or geometry), its scale relative to the model, and its occlusion probability. The overall model shape is represented by the

mutual position of the parts. The entire model is generative and probabilistic, so part description, scale, model shape and occlusion are all modeled by probability density functions, which are Gaussians.

The process of learning an object category is one of first detecting features with characteristic scales, and then estimating the parameters of the above densities from these features, such that the model gives a maximum-likelihood description of the training data. Recognition is performed on a query image by again first detecting features (and their scales), and then evaluating the features in a Bayesian manner, using the model parameters estimated in the learning.

2.1 Model Structure Overview

A model consists of P parts and is specified by parameters θ. Given N detected features with locations $\mathbf{X}$, scales $\mathbf{S}$, and descriptions $\mathbf{D}$, the likelihood that an image contains an object is assumed to have the following form:

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

where the summation is over allocations, $\mathbf{h}$, of parts to features. Typically a model has 5-7 parts and there will be around thirty features of each type in an image.

Similarly it is assumed that non-object background images can be modeled by a likelihood of the same form with parameters θ_{bg}. The decision as to whether a particular image contains an object or not is determined by the likelihood ratio:

$$R = \frac{p(\mathbf{X}, \mathbf{S}, \mathbf{D}|\, \theta)}{p(\mathbf{X}, \mathbf{S}, \mathbf{D}|\theta_{bg})} \tag{1}$$

The model, at both the fitting and recognition stages, is scale invariant. Full details of the model and its fitting to training data using the EM algorithm are given in [10], and essentially the same representations and estimation methods are used.

2.2 Heterogeneous Parts

Existing approaches to recognition learn a model based on a single type of feature (e.g. image patches [3,16], texture regions [18] or Haar wavelets [22]). However, the different visual nature of objects means that this is limiting. For some objects, like wine bottles, the essence of the object is captured far better with geometric information (the outline) rather than by patches of pixels. Of course, the reverse is true for many objects, like humans faces. Consequently, a flexible visual recognition system must have multiple feature types. The flexible nature of the constellation model makes this possible. As the description densities of each part are independent, each can use a different type of feature.

In this paper, only two types of features are included, although more can easily be added. The first consists of regions of pixels, this being the feature type used previously; the second consists of curve segments. Figure 1 illustrates these features on two typical images. These feature are complementary: one represents the *appearance* of object patches, the other represents the object *geometry*.

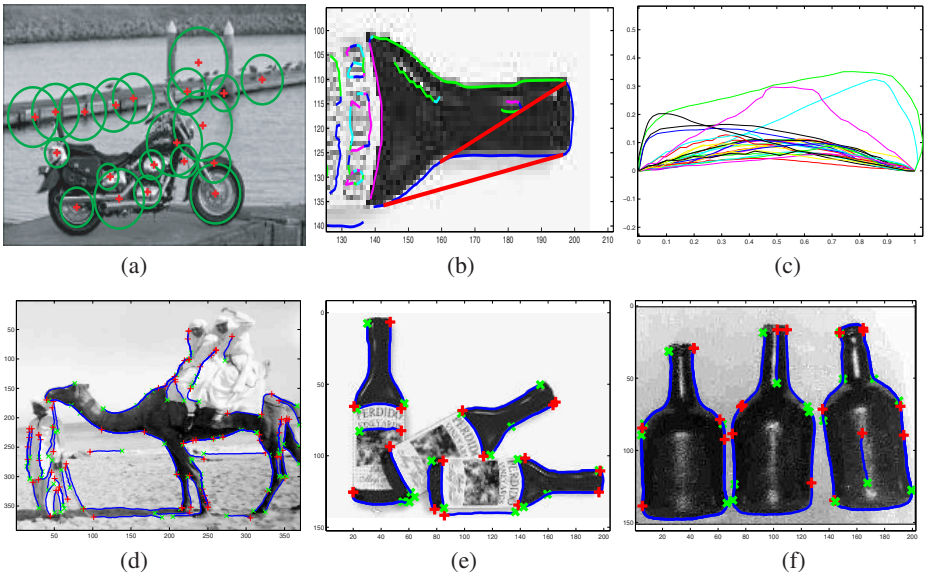


Fig. 1. (a) Sample output from the region detector. The circles indicate the scale of the region. **(b)** A long curve segment being decomposed at its bitangent points. **(c)** Curves within the similarity-invariant space - note the clustering. **(d)**, **(e)** & **(f)** show the curve segments identified in three images. The green and red markers indicate the start and end of the curve respectively

2.3 Feature Detection

Pixel patches. Kadir and Brady's interest operator [14] finds regions that are salient over both location and scale. It is based on measurements of the grey level histogram and entropy over the region. The operator detects a set of circular regions so that both position (the circle centre) and scale (the circle radius) are determined, along with a saliency score. The operator is largely invariant to scale changes and rotation of the image. For example, if the image is doubled in size then a corresponding set of regions will be detected (at twice the scale). Figure 1(a) shows the output of the operator on a sample image.

Curve segments. Rather than only consider very local spatial arrangements of edge points (as in [1]), extended edge chains are used, detected by the Canny edge operator [6]. The chains are then segmented into segments between bitangent points, i.e. points at which a line has two points of tangency with the curve. Figure 1(b) shows an example.

This decomposition is used for two reasons: first, bitangency is covariant with projective transformations. This means that for near planar curves the segmentation is invariant to viewpoint, an important requirement if the same, or similar, objects are imaged at different scales and orientations. Second, by segmenting curves using a bi-local property interesting segments can be found consistently despite imperfect edgel data.

Bitangent points are found on each chain using the method described in [17]. Since each pair of bitangent points defines a curve which is a sub-section of the chain, there

may be multiple decompositions of the chain into curved sections as shown in figure 1(b). In practice, many curve segments are straight lines (within a threshold for noise) and these are discarded as they are intrinsically less informative than curves. In addition, the entire chain is also used, so retaining convex curve portions.

2.4 Feature Representation

The feature detectors gives patches and curves of interest within each image. In order to use them in our model their properties are parametrized to form $\mathbf{D} = [\mathbf{A}, \mathbf{G}]$ where $\mathbf{A}$ is the appearance of the regions within the image, and $\mathbf{G}$ is the shape of the curves within each image.

Region representation. As in [10], once the regions are identified, they are cropped from the image and rescaled to a smaller, 11×11 pixel patch. The dimensionality is then reduced using principal component analysis (PCA). In the learning stage, patches from all images are collected and PCA performed on them. Each patch's appearance is then a vector of the coordinates within the first 15 principal components, so giving $\mathbf{A}$.

Curve representation. Each curve is transformed to a canonical position using a similarity transformation such that it starts at the origin and ends at the point $(1, 0)$. If the curve's centroid is below the x-axis then it is flipped both in the x-axis and the line $y = 0.5$, so that the same curve is obtained independent of the edgel ordering. The y value of the curve in this canonical position is sampled at 13 equally spaced x intervals between $(0, 0)$ and $(1, 0)$. Figure 1(c) shows curve segments within this canonical space. Since the model is not orientation-invariant, the original orientation of the curve is concatenated to the 13-vector for each curve, giving a 15-vector (for robustness, orientation is represented as a normalized 2-vector). Combining the 15-vectors from all curves within the image gives $\mathbf{G}$.

2.5 Model Structure and Representation

The descriptors are modelled by the $p(\mathbf{D}|\mathbf{X}, \mathbf{S}, \mathbf{h}, \theta)$ likelihood term. Each part models either curves or patches and this allocation is made beforehand. $\mathbf{h}$ picks a feature for each part from $\mathbf{A}$ or $\mathbf{G}$ (as appropriate) and is then modelled by a 15 dimensional Gaussian (note that both curves and patches are represented by a 15-vector). This Gaussian will hopefully find a cluster of curves/patches close together in the space, corresponding to similar looking curves or patches across images. The relative locations of the model parts are modelled by $p(\mathbf{X}|\mathbf{S}, \mathbf{h}, \theta)$ – which is a joint Gaussian density over all parts. Again, $\mathbf{h}$ allocates a feature to each part. The location of curve is taken as its centroid. The location of a patch is its region centre. For the relative scale term, $p(\mathbf{S}|\mathbf{h}, \theta)$ – again a Gaussian, the length of the curve and the radius of a patch region is taken as being the scale for a curve/patch.

3 Method

In this section the experimental implementation is described: the gathering of images, feature detection, model learning and ranking. The process will be demonstrated on the "bottles" category .

3.1 Image Collection

For a given keyword, Google's image search[1] was used to download a set of images. Images outside a reasonable size range (between 100 and 600 pixels on the major axis) were discarded. A typical image search returned in the region of 450-700 usable images. A script was used to automate the procedure. For assessment purposes, the images returned were divided into 3 distinct groups (see fig. 2):

1. **Good images:** these are good examples of the keyword category, lacking major occlusion, although there may be a variety of viewpoints, scalings and orientations.
2. **Intermediate images:** these are in some way related to the keyword category, but are of lower quality than the good images. They may have extensive occlusion; substantial image noise; be a caricature or cartoon of the category; or the category is rather insignificant in the image, or some other fault.
3. **Junk images:** these are totally unrelated to the keyword category.

Additionally, a dataset consisting entirely of junk images was collected, by using the keyword "things". This background dataset is used in the unsupervised learning procedure.

The algorithm was evaluated on ten datasets gathered from Google: bottles, camel, cars, coca cola, horses, leopards, motorbike, mugs, tiger and zebra. It is worth noting that the inclusion or exclusion of an "s" to the keyword can make a big difference to the images returned. The datasets are detailed in Table 1.

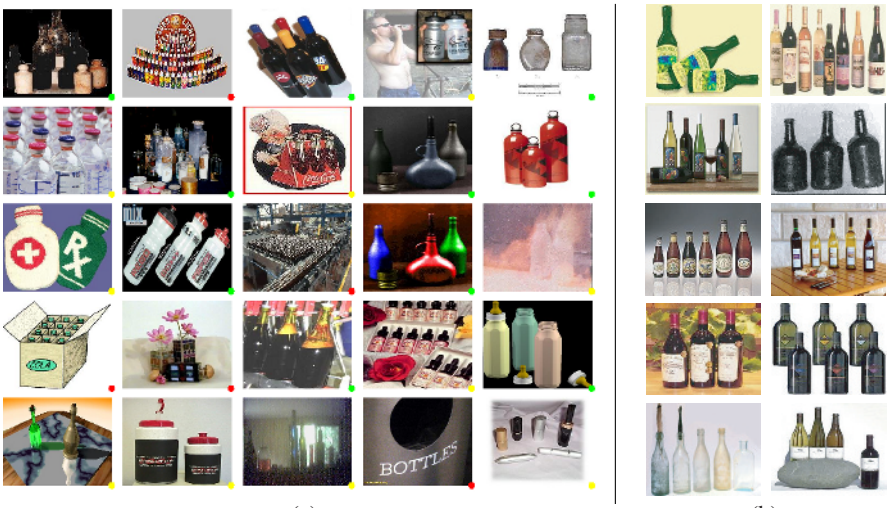
Table 1. Statistics of the datasets as returned by Google.

Dataset	Bottles	Camel	Cars	Coca-cola	Horses	Leopards	Motorbike	Mugs	Tiger	Zebra	Things
Total size of dataset	700	700	448	500	600	700	500	600	642	640	724
% Good images	41	24	30	17	21	49	25	50	35	44	n.a.
% Intermediate images	26	27	18	12	25	33	16	9	24	33	n.a.
% Junk images	33	49	52	71	54	18	59	41	41	24	n.a.

3.2 Image Re-ranking

Feature detection. Each image is converted to greyscale, since colour information is not used in the model. Curves and regions of interest are then found within the image, using exactly the same settings for all datasets. This produces $\mathbf{X}$, $\mathbf{D}$ and $\mathbf{S}$ for use in learning or recognition. The 25 regions with the highest saliency, and 30 curves with the longest length are used from each image.

[1] `http://www.google.com/imghp`. Date of collection: Jan. 2003. As we write (Feb. 2004) we notice that Google's precision-recall curves have improved during the last 12 months.



(a) (b)

Fig. 2. Images of bottles. (a) the first 25 images returned by Google. The coloured dot in the bottom right hand corner indicates the ground truth category of the image: good (green); intermediate (yellow) or junk (red). **(b)** the 10 hand selected images used in the supervised experiments.

Model Learning. The learning process takes one of two distinct forms: unsupervised learning and limited supervision:

- **Unsupervised learning:** In this scenario, a model is learnt using all images in the dataset. No human intervention is required in the process.
- **Learning with limited supervision:** An alternative approach is to use relevance-feedback. The user picks 10 or so images that are close to the image he/she wants, see figure 2(b) for examples for the bottles category. A model is learnt using these images.

In both approaches, the learning task takes the form of estimating the parameters θ of the model discussed above. The goal is to find the parameters $\hat{\theta}_{ML}$ which best explain the data $\mathbf{X}, \mathbf{D}, \mathbf{S}$ from the chosen training images (be it 10 or the whole dataset), i.e. maximise the likelihood: $\hat{\theta}_{ML} = arg\,max_{\theta}\ p(\mathbf{X}, \mathbf{D}, \mathbf{S}|\,\theta)$. For the 5 part model used in the experiments, there are 243 parameters. In the supervised learning case, the use of only 10 training images is a compromise between the number the user can be expected to pick and the generalisation ability of the model. The model is learnt using the EM algorithm as described in [10]. Figure 3 shows a curve model and a patch model trained from the 10 manually selected images of bottles.

Re-ranking. Given the learnt model, the likelihood ratio (eqn. 1) for each image is computed. This likelihood ratio is then used to rank all the images in the dataset. Note that in the supervised case, the 10 images manually selected are excluded from the ranking.

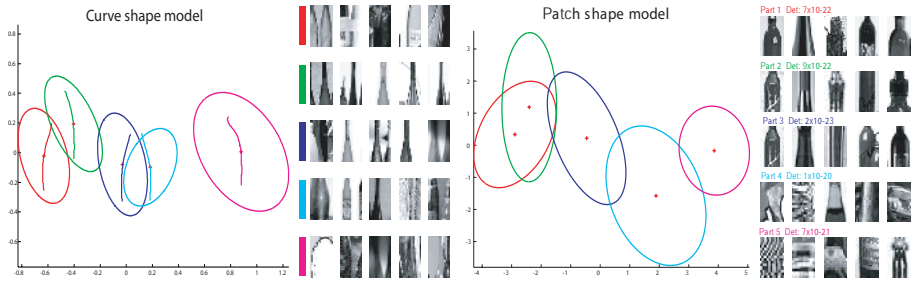


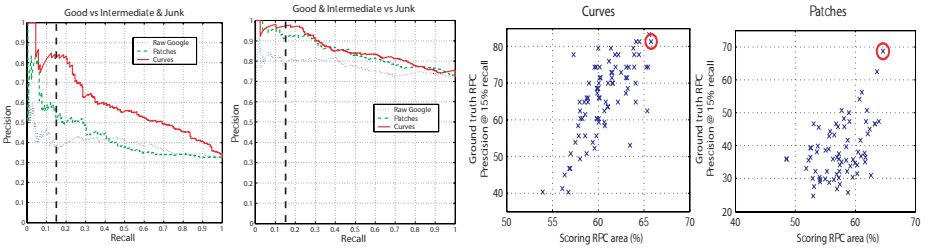
Fig. 3. Models of bottles. (a) & **(b)**: Curve model. **(c)** & **(d)**: Patch model. **(a)** The spatial layout of the curve model with mean curves overlaid. The X and Y axes are in arbitrary units since the model is scale-invariant. The ellipses indicate variance in relative location. **(b)** Patch of images selected by curve features from high scoring hypotheses. **(c)** Spatial layout for patch model. **(d)** Sample patches closest to mean of appearance density. Both models pick out bottle necks and bodies with the shape model capturing the side-by-side arrangement of the bottles.

Speed considerations. If this algorithm is to be of practical value, it must be fast. Once images have been preprocessed, which can be done off-line, a model can be learnt from 10 images in around 45 seconds and the images in the dataset re-ranked in $4-5$ seconds on a 2 Ghz processor.

3.3 Robust Learning in the Unsupervised Case

We are attempting to learn a model from a dataset which contains valid data (the good images) but also *outliers* (the intermediate and junk images), a situation faced in the area of robust statistics. One approach would be to use all images for training and rely on the models' occlusion term to account for the small portion of valid data. However, this requires an accurate modelling of image clutter properties and reliable convergence during learning. An an alternative approach, we adapt a robust fitting algorithm, RANSAC [11], to our needs. A large number of models are trained (~ 100), each one using a set of randomly drawn images sufficient to train a model (10 in this case). The intuition is that at least one of these will be trained on a higher than average proportion of good images, so will be a good classifier. The challenge is to find a robust unsupervised scoring function that is highly correlated to the underlying classification performance. The model with the highest score is then picked as model to perform the re-ranking of the dataset.

Our novel scoring approach uses a second set of images, consisting entirely of irrelevant images, the aforementioned background dataset. Thus there are now two datasets: (a) the one to be ranked (consisting of a mixture of junk and good images) and (b) the background dataset. Each model evaluates the likelihood of images from both datasets and a differential ranking measure is computed between them. In this instance, we compute the area under a recall-precision curve (RPC) between the two datasets. In our experiments we found a good correlation between this measure and the ground truth RPC precision: the final model picked was consistently in the top 15% of models, as demonstrated in figs. 4(c) & (d).



(a) (b) (c) (d)

Fig. 4. (a) & **(b)** Recall-Precision curves computed using ground truth for the supervised models in figure 3. In **(a)**, the good images form the positive set and the intermediate and junk images form the negative one. In **(b)**, good and intermediate images form the positive set and junk images, the negative one. The dotted blue line is the curve of the raw Google images (i.e. taken in the order they are presented to the user). The solid red line shows the performance of the curve model and the dashed green line shows the performance of the patch model. As most users will only look at the first few pages of returned results, the interesting area of the plots is the left-hand side of the graph, particularly around a recall of 0.15 (as indicated by the vertical line). In this region, the curve model clearly gives an improvement over both the raw images and the patch model (as predicted by the variance measure). **(c)** & **(d)**: Scatter plots showing the scoring RPC area versus ground truth RPC area for curve and patch models respectively in the unsupervised learning procedure. Each point is a model learnt using the RANSAC-style unsupervised learning algorithm. The model selected for each feature type is indicated by the red circle. Note that in both plots it is amongst the best few models.

3.4 Selection of Feature Type

For each dataset in both the supervised and unsupervised case, two different models are learnt: one using only patches and another using only curves. A decision must be made as to which model should give the final ranking that will be presented to the user. This is a challenging problem since the models exist in different spaces, so their likelihoods cannot be directly compared. Our solution is to compare the variance of the unsupervised models' scoring function. If a feature type is effective then a large variance is expected since a good model will score much better than a mediocre one. However, an inappropriate feature type will be unable to separate the data effectively, no matter which training images were used, meaning all scores will be similar.

Using this approach, the ratio of the variance of the RANSAC curve and patch models is compared to a threshold (fixed for all datasets) and a selection of feature type is made. This selection is then used for both the unsupervised and supervised learning cases. Figure 5 shows the first few re-ranked images of the bottles dataset, using the model chosen - in this case, curves.

4 Results

Two series of experiments were performed: the first used the supervised learning method while the second was completely unsupervised. In both sets, the choice between curves and patches was made automatically. The results of the experiments are summarised in table 2.

Fig. 5. Re-ranked bottle images. The dot in the bottom right corner shows the label of the image. The thin magenta curves on each image show the curve segments detected. The best hypothesis is also highlighted with thick coloured lines. The duplicate images present in the dataset are the reason that some of the 10 training images appear in the figure. Notice that the model seems to pick up the neck of the bottles, with its distinctive curvature. These images clearly contain more bottles than those of figure 2.

Table 2. Summary of results: Precision at 15% recall - equivalent to around two web-pages worth of images. Good images vs. intermediate & junk. The second row gives raw Google output precision. Rows 3 & 4 give results of supervised learning, using 10 handpicked images. Rows 5 & 6 give results of unsupervised RANSAC-style learning. Rows 7 & 8 are included to show the comparison of the RANSAC approach to unsupervised learning using all images in the dataset. Bold indicates the automatically selected model. For the forms of learning used (supervised and RANSAC-style unsupervised), this model selection is correct 90% of the time. The final column gives the average precision across all datasets, for the automatically chosen feature type.

Dataset	Bottles	Camel	Cars	Coca-cola	Horses	Leopards	Motorbike	Mugs	Tiger	Zebra	Average
Raw Google	39.3	36.1	31.7	41.9	31.1	46.8	48.7	84.9	30.5	51.9	44.3
10 images (Curves)	**82.9**	**80.0**	**78.3**	35.3	28.3	39.5	**48.6**	**75.0**	43.8	**74.1**	65.9
10 images (Patches)	52.3	68.6	47.4	**54.5**	**23.6**	**69.0**	42.5	55.7	**72.7**	74.1	
RANSAC unsupervised-Curves	**81.4**	**78.8**	**69.0**	29.5	25.0	41.5	**61.3**	**68.2**	43.4	**71.2**	58.9
RANSAC unsupervised-Patches	68.6	48.7	42.6	**26.0**	**25.0**	**50.0**	20.4	66.7	**58.9**	54.5	
All images unsupervised-Curves	**76.1**	**81.2**	**41.7**	43.3	23.2	51.0	**34.5**	**76.3**	44.0	**64.6**	52.9
All images unsupervised-Patches	35.0	27.4	44.4	**23.6**	**22.4**	**55.4**	17.9	62.5	**53.2**	50.0	

4.1 Supervised Learning

The results in table 2 show that the algorithm gives a marked improvement over the raw Google output in 7 of the 10 datasets. The evaluation is a stringent one, since the model must separate the good images from the intermediate and junk, rather than just separating the good from the junk. The curve features were used in 6 instances, as compared to 4 for patches. While curves would be expected to be preferable for categories such as bottles, their marked superiority on the cars category, for example, is surprising. It can be explained by the large variation in viewpoint present in the images. No patch features could be found that were stable across all views, whereas long horizontal curves in close proximity were present, regardless of the viewpoint and these were used by the model, giving a good performance. Another example of curves being unexpectedly effective, is on the camel dataset, as shown in figure 6. Here, the knobbly knees and legs of the camel

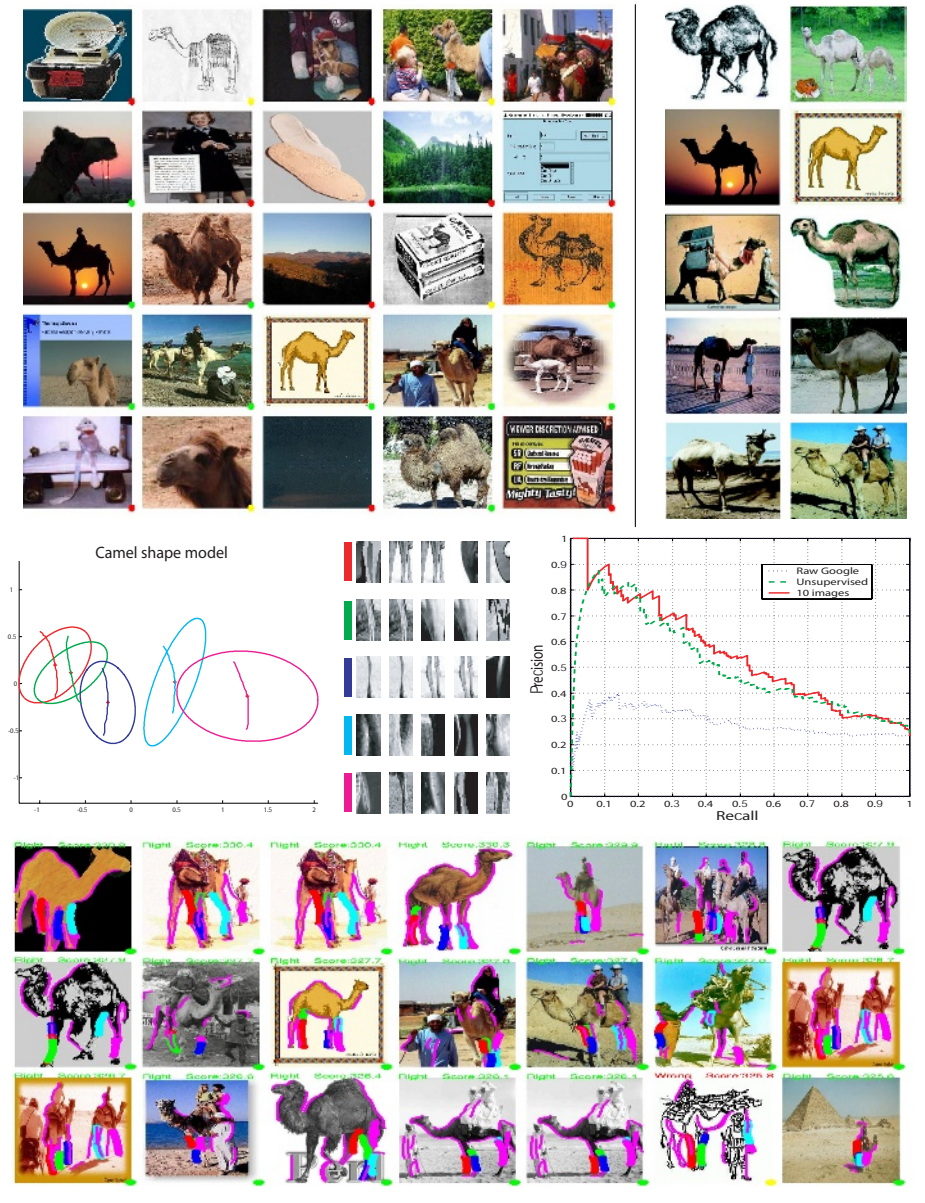


Fig. 6. Camel. The algorithm performs well, even in the unsupervised scenario. The curve model, somewhat surprisingly, locks onto the long, gangly legs of the camel. From the RPC (good vs intermediate & junk), we see that for low recall (the first few web-pages returned), both the models have around double the precision of the raw Google images.

are found consistently, regardless of viewpoint and clutter, so are used by the model to give a precision (at 15% recall) over twice that of the raw Google images. The failure to improve Google's output on 3 of the categories (horses, motorbikes and mugs), can be mainly attributed to an inability to obtain informative features on the object. It is worth noting that in these cases, either the raw Google performance was very good (mugs) or the portion of good images was very small ($\leq$25%).

4.2 Unsupervised Learning

In this approach, 6 of the 10 cases were significantly better than the raw Google output. Many of them were only slightly worse than the supervised case, with the motorbike category actually superior. This category is shown in figure 7.

In table 2, RANSAC-style learning is compared to learning directly from all images in the dataset. The proportion of junk images in the dataset determines which of the two approaches is superior: using all images is marginally better when the proportion is small, while the RANSAC approach is decisively better with a large proportion of junk.

5 Discussion and Future Work

Reranking Google images based on their similarity is a problem that is similar to classical visual object recognition. However, it is worth noting the significant differences. In the classical setting of visual recognition we are handed a clean training set consisting of carefully labelled 'positive' and 'negative' examples; we are then asked to test our algorithm on fresh data that was collected independently. In the present scenario the training set is not labelled, it contains a minority (20-50%) of 'good' examples, and a majority of either 'intermediate' or 'junk' examples. Moreover, after learning, our task is to sort the 'training' set, rather than work on fresh data.

Selecting amongst models composed of heterogeneous features is a difficult challenge in our setting. If we had the luxury of a clean labelled training set, then part of this could have been selected as a validation set and then used to select between all-curve and all-patch models. Indeed we could then have trained heterogeneous models where parts could be either curves or patches. However, the non-parametric RPC scoring methods developed here are not up to this task.

It is clear that the current features used are somewhat limited in that they capture only a small fraction of the information from each image. In some of the datasets (e.g. horses) the features did not pick out the distinctive information of the category at all, so the model had no signal to deal with and the algorithm failed as a consequence. By introducing a wider range of feature types (e.g. corners, texture) a wider range of datasets should be accessible to the algorithm.

Overall, we have shown that in the cases where the model's features (patches and curves) are suitable for the object class, then there is a marked improvement in the ranking. Thus we can conclude that the conjecture of the introduction is valid – visual consistency ranking is a viable visual category filter for these datasets.

There are a number of interesting issues in machine learning and machine vision that emerge from our experience: (a) Priors were not used in either of the learning

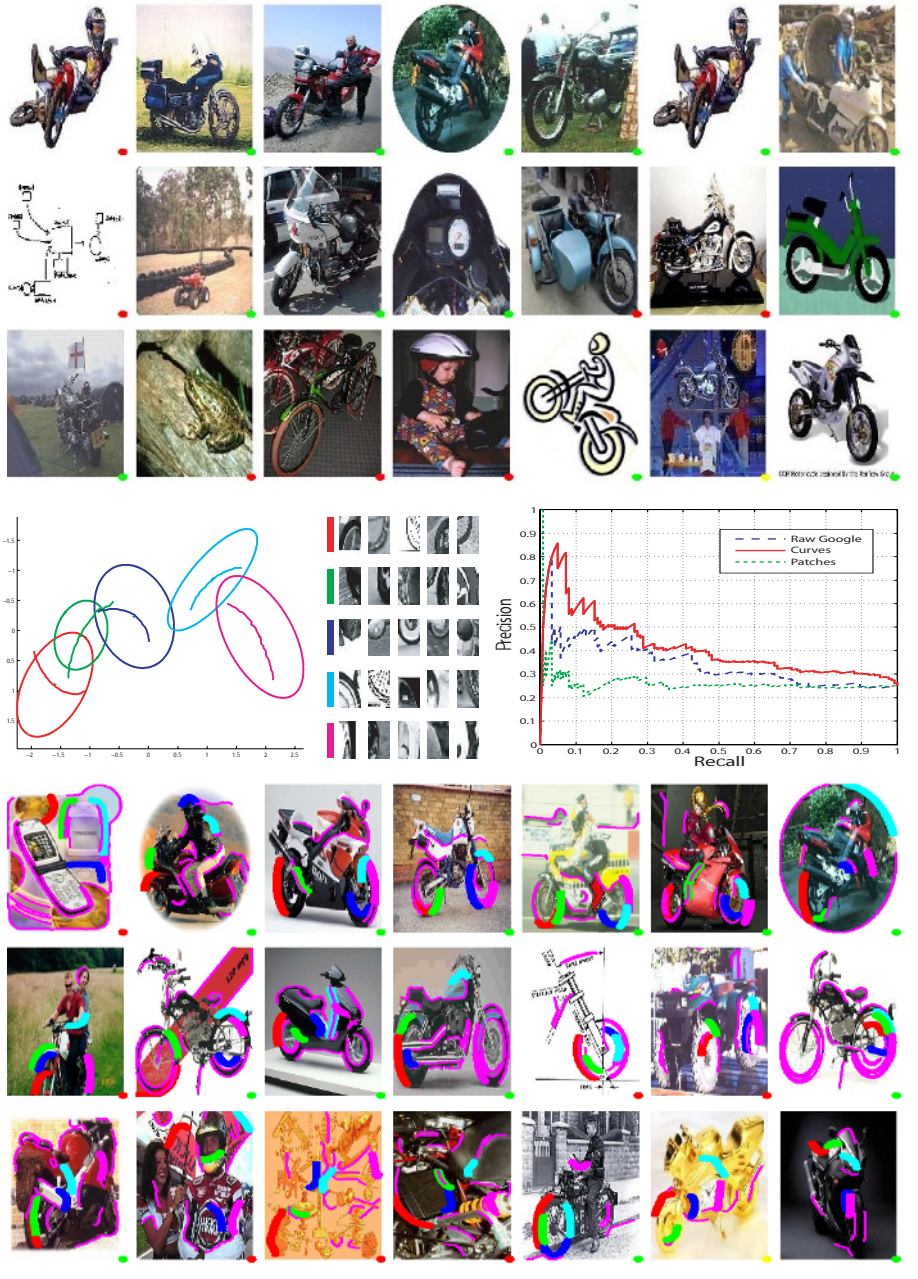


Fig. 7. Motorbike. The top scoring unsupervised motorbike model, selected automatically. The model picks up on the wheels of the bike, despite a wide range of viewpoints and clutter. The RPC (good vs intermediate & junk) shows the curve model performing better than Google's raw output and the model based on patches (which is actually worse than the raw output).

scenarios. In Fei-Fei *et al.* [8] priors were incorporated into the learning process of the constellation model, enabling effective models to be trained from a few images. Applying these techniques should enhance the performance of our algorithm. (b) The 'supervised' case could be improved by using simultaneously the small labelled training data provided by the user, as well as the large unlabelled original dataset. Machine learning researchers are making progress on the problem of learning from 'partially labeled' data. We ought to benefit from that effort.

Acknowledgements. Financial support was provided by: EC Project CogViSys; UK EPSRC; Caltech CNSE and the NSF.

References

1. Y. Amit and D. Geman. A computational model for visual selection. *Neural Computation*, 11(7):1691–1715, 1999.
2. J. Bach, C. Fuller, R. Humphrey, and R. Jain. The virage image search engine: An open framework for image management. In *SPIE Conf. on Storage and Retrieval for Image and Video Databases*, volume 2670, pages 76–87, 1996.
3. E. Borenstein and S. Ullman. Class-specific, top-down segmentation. In *Proc. ECCV*, pages 109–124, 2002.
4. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *7th Int. WWW Conference*, 1998.
5. M. Burl, T. Leung, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. In *Proc. ECCV*, 1998.
6. J. F. Canny. A computational approach to edge detection. *IEEE PAMI*, 8(6):679–698, 1986.
7. T. Deselaers, D. Keysers, and H. Ney. Clustering visually similar images to improve image search engines. In *Informatiktage 2003 der Gesellschaft für Informatik, Bad Schussenried, Germany.*, 2003.
8. L. Fei-Fei, R. Fergus, and P. Perona. A bayesian approach to unsupervised one-shot learning of object categories. In *Proceedings of the 9th International Conference on Computer Vision, Nice, France*, pages 1134–1141, 2003.
9. P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. In *Proc. CVPR*, pages 2066–2073, 2000.
10. R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003.
11. M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.
12. T. Gevers and A. W. M. Smeulders. Content-based image retrieval by viewpoint-invariant color indexing. *Image and vision computing*, 17:475–488, 1999.
13. B. Heisele, T. Serre, M. Pontil, T. Vetter, and T. Poggio. Categorization by learning and combining object parts. In *Advances in Neural Information Processing Systems 14, Vancouver, Canada.*, volume 2, pages 1239–1245, 2002.
14. T. Kadir and M. Brady. Scale, saliency and image description. *IJCV*, 45(2):83–105, 2001.
15. B. Leibe and B. Schiele. Analyzing appearance and contour based methods for object categorization. In *Proc. CVPR*, 2003.
16. M. Riesenhuber and T. Poggio. Hierarchical models of object recognition in cortex. *Nature Neuroscience*, 2(11):1019–1025, 1999.

17. C. Rothwell, A. Zisserman, D. Forsyth, and J. Mundy. Planar object recognition using projective shape representation. *IJCV*, 16(2), 1995.
18. C. Schmid. Constructing models for content-based image retrieval. In *Proc. CVPR*, volume 2, pages 39–45, 2001.
19. S. Tong and E. Chang. Support vector machine active learning for image retrieval. *ACM Multimedia*, 2001.
20. N. Vasconcelos and A. Lippman. Learning from user feedback in image retrieval systems. In *Proc. ECCV*, 2000.
21. R. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. Technical Report UU-CS-2000-34, Department of Computing Science, Utrecht University, 2000.
22. P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *Proc. CVPR*, pages 511–518, 2001.
23. M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. In *Proc. ECCV*, pages 18–32, 2000.