Object Categorization by Learned Universal Visual Dictionary

J. Winn, A. Criminisi and T. Minka

Microsoft Research, Cambridge, UK – http://research.microsoft.com/vision/cambridge/recognition/

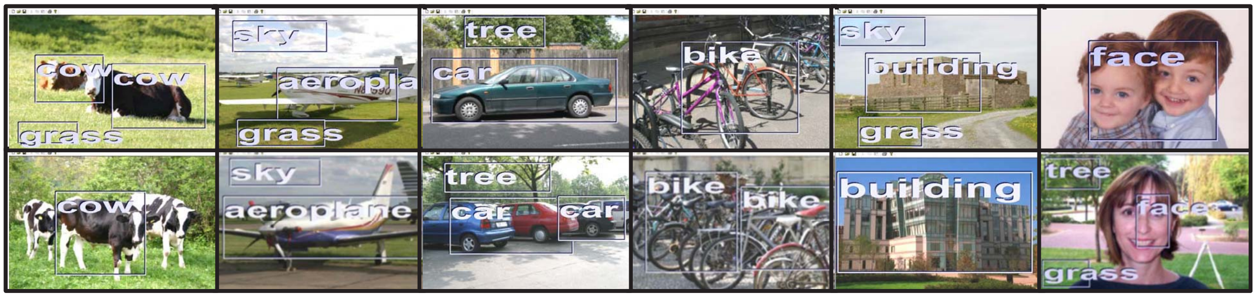

Figure 1: Exemplar snapshots of our interactive object categorization demo application. A user selects (sloppily) a region of interest and our algorithm associates an object class label with it. Despite large differences in pose, size, illumination and visual appearance the correct class label (e.g. cow, building, car...) is automatically associated with each selected object instance. Some of these test images were downloaded from the web and none were part of the training set. A video of the interactive demo may be found at the above web site.

Abstract

This paper presents a new algorithm for the automatic recognition of object classes from images (categorization). Compact and yet discriminative appearance-based object class models are automatically learned from a set of training images.

The method is simple and extremely fast, making it suitable for many applications such as semantic image retrieval, web search, and interactive image editing. It classifies a region according to the proportions of different visual words (clusters in feature space). The specific visual words and the typical proportions in each object are learned from a segmented training set. The main contribution of this paper is two fold: i) an optimally compact visual dictionary is learned by pair-wise merging of visual words from an initially large dictionary. The final visual words are described by GMMs. ii) A novel statistical measure of discrimination is proposed which is optimized by each merge operation.

High classification accuracy is demonstrated for nine object classes on photographs of real objects viewed under general lighting conditions, poses and viewpoints. The set of test images used for validation comprise: i) photographs acquired by us, ii) images from the web and iii) images from the recently released Pascal dataset. The proposed algorithm performs well on both texture-rich objects (e.g. grass, sky, trees) and structure-rich ones (e.g. cars, bikes, planes).

1. Introduction

This paper studies the problem of constructing compact and discriminative models of object classes and presents a novel algorithm for the automatic recognition of objects from images. An example is shown in fig. 1 where the objects in the manually selected test regions (marked as rectangles) have correctly been recognized by the proposed algorithm as instances of the classes cow, aeroplane, car, face etc.

Object categorization is difficult because differing pose, scale, illumination and intrinsic visual differences produce highly different images for objects of the same class. For example fig. 1 shows deformable objects (sitting/standing cows), and extreme partial occlusions (in the car and bike images). Existing shape-based modeling techniques are not designed to deal with these large variations. Thus, we have built our algorithm upon appearance-based models drawn from the material classification literature. Specifically, we have borrowed the *texton*-based models developed in the context of texture recognition [9, 16] and extended by [1].

The challenge in object categorization is to find class models that are *invariant* enough to incorporate naturally-occurring intra-class variations and yet *discriminative* enough to distinguish between different classes. In this paper we propose a supervised learning algorithm which automatically finds such models. Additionally we require the learned models to be compact and light-weight so as to enable *efficient* classification.

The learned models specify the typical proportions of textons in each class, regardless of spatial layout. To our surprise we have found that the learned models perform extremely well with both shape-free objects (sky, grass and trees) and also with highly structured object-classes (faces, cars, aeroplanes and bikes).

2. Previous work

Object class recognition is a well-studied vision problem, with approaches ranging from voting independent patches to full models of spatial layout and deformation. For example, constellation models [3, 4, 5], fragment-based models [14] and pictorial structures [8] try to locate distinctive object parts and determine constraints on their spatial arrangement. While these approaches are potentially very powerful, the spatial models which are typically used cannot handle significant deformations such as large out-of-plane rotations. They also do not consider objects with variable numbers of parts such as buildings and trees. Our approach can be viewed as a simplified parts model in which the parts can be arbitrarily rearranged but tend to occur in particular proportions, such as leaves on trees or windows on buildings. This approach runs the risk of not being able to discriminate shapes, but surprisingly it seems sufficient to recognize a wide range of object classes without explicit shape modeling.

A similar image labeling task was considered in [2] and [10]. These systems used machine learning techniques to classify regions found by automatic segmentation. However such segmentations often do not correlate with semantic objects, for example an object in shadow may be divided into a shadowed versus non-shadowed part. Our solution to this problem is to: i) take the region as input from the user or ii) test a variety of regions and pick the one that is most likely from the point of view of the classifier as opposed to a separate segmentation algorithm.

The approach proposed here can be considered an extension of the method of [1]. In that work, images were described by histograms over a dictionary (of selected size) of visual words[1]. The visual words were chosen by K-means clustering, and features computed only on a sparse set of interest points. We build upon [1] by automatically learning the optimal visual words and dictionary size. Unlike [1], our approach is dense; i.e. we process every pixel, avoiding early removal of potentially useful regions, such as textureless blue/grey regions which can be distinctive of sky.

3. Training set and visual features

The training image set. Our class models are learned from a set of 240 manually segmented and annotated photographs (fig. 2). Those photographs depict different objects in completely general positions, lighting conditions and viewpoints. The objects belong to the *nine* classes: building, grass, tree, cow, sky, aeroplane, face, car and bicycle.

The training images were manually segmented (quickly and sloppily) into object-defined regions by means of a "paint"-like interface (fig. 2d), with the assigned colours acting as indices into the list of object classes.

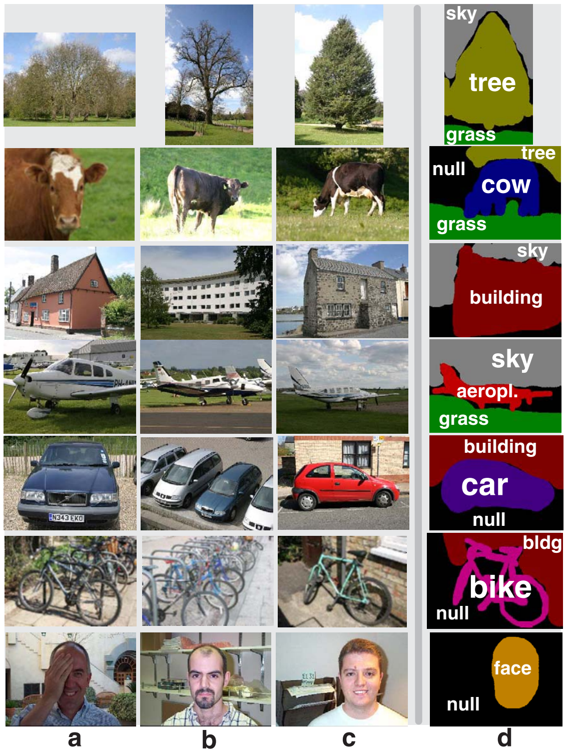

Figure 2: The labeled training set. (Columns **a-c**) A selection of images in the 240-image training set (image size is 320 × 213). Notice the large within-class variability. (Column **d**) Ground-truth annotation for column c. Labeling has been achieved for all training images by a simple, interactive "paint" interface. Same colours correspond to same object class.

The face images were downloaded from the Caltech dataset[2] while the other photographs were taken by us. The entire annotated database is available on our web site.

Textons and texton histograms. Each training image is convolved with a filter-bank to generate a set of filter responses [9, 16]. These filter responses are aggregated over *all* the images in the entire training set (independently from class labels) and clustered using a K-means approach. Mahalanobis distance between features is used during clustering. Then, the set of estimated cluster centres (textons/visual words) and their associated covariances define a universal visual dictionary (UVD). In this initial step large values of K are employed (in the order of thousands), but the next sections will show how to reduce the size of the UVD without loss of class discrimination. Given a UVD, any image can be filtered and each pixel associated with the closest texton in the dictionary, thus generating a map of indices into the UVD. At this point normalized histograms of textons can readily be computed on a region or image basis.

[1] In the literature the terms "textons", "keypoints" or "visual words" have been used with approximately the same meaning, i.e. clusters of filter responses/feature vectors in a high-dimensional space.

[2] http://www.vision.caltech.edu/html-files/archive.html

Filter-banks. In this paper we have tested a number of different filter-banks made of combinations of Gaussians, first and second order derivatives of Gaussians and Gabor kernels. Many filter-banks produced comparable results with the best one made of 3 Gaussians, 4 Laplacian of Gaussians (LoG) and 4 first order derivatives of Gaussians. The three Gaussian kernels (with $\sigma = 1, 2, 4$) are applied to each CIE L,a,b channel [7], thus producing 9 filter responses. The four LoGs (with $\sigma = 1, 2, 4, 8$) were applied to the L channel only, thus producing 4 filter responses. The four derivatives of Gaussians were divided into the two $x-$ and $y-$aligned sets, each with two different values of σ ($\sigma = 2, 4$). Derivatives of Gaussians were also applied to the L channel only, thus producing 4 final filter responses. Therefore, each pixel in each image has associated a $17-$dimensional feature vector. Note that first order derivatives of Gaussian kernels are not rotational invariant. However, rather than deciding a-priori whether to remove rotational dependency or not, we let our supervised learning algorithm decide for us. In addition to this filter-bank, we also investigated the performance of raw 5×5 colour patches ($5 \times 5 \times 3 = 75-$dimensional feature vectors). In our experiments using colour and intensity alone (only the 9 Gaussian filter responses) performed poorly.

4. Modeling object classes

This section describes the main contribution of this paper: a statistical algorithm for learning a compact and yet discriminative representation of object classes.

4.1. Objects as texture conglomerates

In texture classification [9, 16] classes are modeled by histograms of textons (visual words). The assumption being that similar distributions of textons (from a unique dictionary) apply to similar textures. In this paper we represent objects as conglomerates of different texture regions and thus we apply the same histogram-of-texton modeling technique. Note that we never need to explicitly recognize each component texture (each "part"), as much as the overall distribution of the "words" from the dictionary.

Interestingly, the size and nature of the dictionary affects the class models and thus the discrimination power. In [16] it was noticed that there is an optimal dictionary size K for which classification accuracy is maximum. Both larger or smaller visual dictionaries do not perform as well.

Unlike previous techniques which manually fix the dictionary size and then estimate the textons by unsupervised clustering, here we infer both the best visual words and dictionary size from the training data in a supervised fashion. In fact, we propose a new statistical generative technique that, by merging textons from a large initial dictionary, estimates a new, considerably smaller target dictionary without loss of class discriminability. The two driving forces of our supervised clustering technique are high class discriminability and compactness of dictionary. Not only do we estimate the appropriately small size of the UVD, but we also make sure that we maintain high classification accuracy by only merging visual words which do not need to be kept separate.

The next section will describe inference of the optimal UVD and object class models.

4.2. Learning an optimal visual dictionary and modeling compact object classes

Each image in the training database is convolved with each of the P filters in the selected filter-bank. Then, each pixel position is associated with a $P-$dimensional feature vector $\mathbf{p}$. Note that here all available image data is processed, rather than only some specified interest locations.

The whole set of feature vectors are then clustered using K-means with a large value K (in the order of thousands). The set of resulting $P-$dimensional clusters (textons) and the associated covariances constitutes the *initial* dictionary $\mathcal{F}$. The goal here is to "manipulate" $\mathcal{F}$ and come up with a new discriminative dictionary $\mathcal{T}$ with size $T \ll K$.

We have given a set of N annotated training regions, with ground-truth class labels $c \in \{1 \cdots C\}$. Each training region has a texton distribution $\mathbf{h}$ (histogram over the initial dictionary $\mathcal{F}$) associated with it; and also a corresponding histogram of "target" textons $\mathbf{H}$. All histograms are normalized to sum to one. The aim is to find the best mapping $\mathbf{H} = \phi(\mathbf{h})$. The strategy used here is to define ϕ as a pairwise merging operation acting on textons. The intuition is that by merging textons which do not help distinguish between classes, one can produce a much more compact and yet discriminative visual dictionary $\mathcal{T}$.

4.2.1 The generative model for texton histograms

We wish our model to prefer that histograms over the final dictionary are similar for regions of the same class (reducing intra-class variation). Hence, we model the set of histograms for each class using a Gaussian distribution with mean $\bar{\mathbf{H}}_c$ and diagonal covariance whose diagonal entries form the vector $\boldsymbol{\beta}_c$. Thus, a key assumption is that whenever an object of class c appears in an image, the corresponding region histogram $\mathbf{H}$ is close to the mean class histogram $\bar{\mathbf{H}}_c$, in terms of a Mahalanobis distance given by $\boldsymbol{\beta}_c$.

We define this relationship probabilistically for a class with parameters $\boldsymbol{\theta} = (\bar{\mathbf{H}}, \boldsymbol{\beta})$ by

$$P(\mathbf{H}|\boldsymbol{\theta}) = \prod_{i=1}^{T} \mathcal{N}\left(H_i^{\frac{1}{2}} | \bar{H}_i^{\frac{1}{2}}, \beta_i^{-1}\right) \qquad (1)$$

where H_i denotes the i^{th} bin of a histogram. Note that the value in each bin is raised to a power of a half. This is the *variance stabilizing transformation* [6] of a multinomial (or equivalently a Poisson distribution) which has the effect of making the variance constant, rather than linearly dependent on the mean $\bar{\mathbf{H}}$. Hence, it makes the assumption of a Gaussian model with constant variance more accurate for this multinomial data.

Applying Bayesian methodology, we define a common prior over the parameters for each class $\boldsymbol{\theta}$

$$P(\boldsymbol{\theta}) = \prod_{i=1}^{T} \mathcal{N}\left(\bar{H}_i^{\frac{1}{2}} | \mu, (\lambda\beta_i)^{-1}\right) \mathcal{G}(\beta_i | a, b) \qquad (2)$$

where the hyper-parameters are fixed to $\{\mu = 0, \lambda = 0.1, a = 0.01, b = 0.01\}$; with $\mathcal{G}$ denoting the gamma distribution.

Each training image region is assumed to contain a single instance of an object class and so $\hat{\mathbf{c}} = [\hat{c}_1 \cdots \hat{c}_N]$ is the vector of training labels associated with all of the N training regions. A particular mapping ϕ defines new texton histograms $\mathbf{H}_1, \cdots, \mathbf{H}_N$. From (1) and (2), the distribution over these histograms conditioned on the ground truth labels is

$$P(\{\mathbf{H}_n\}|\hat{\mathbf{c}}) = \prod_{c=1}^{C} \int \prod_{n \in R_c} P(\mathbf{H}_n | \boldsymbol{\theta}_c) P(\boldsymbol{\theta}_c) d\boldsymbol{\theta}_c \qquad (3)$$

where R_c is the set of regions with object label c and we have marginalised out the class parameters $\boldsymbol{\theta}_c$.

4.2.2 Compactness v discrimination trade-off

If we attempted to find the mapping that maximises the probability of the histograms (3), we would end up merging all the bins together into a single bin, since all histograms for any class would then be identical. Of course, we would be completely unable to discriminate between classes.

Instead, we wish to set up a trade-off between making the histograms more similar within each class and making them more discriminative between classes. Consider finding the conditional probability of the class labels; using Bayes' rule

$$P(\hat{\mathbf{c}}|\{\mathbf{H}_n \equiv \phi(\mathbf{h}_n)\}) = \frac{P(\{\mathbf{H}_n\}|\hat{\mathbf{c}})P(\hat{\mathbf{c}})}{\sum_{\mathbf{c}'} P(\{\mathbf{H}_n\}|\mathbf{c}')P(\mathbf{c}')} \qquad (4)$$

where the sum in the denominator is over all of the C^N possible object labelings and $P(\mathbf{c})$ is the prior over labelings, which we set to be uniform.

We now aim to find the mapping ϕ which maximizes this conditional probability (4). The term in the denominator acts to penalise mappings which reduce discriminability (i.e. which make the observed data likely under class labelings other than the true one). The numerator still favours mappings which lead to small intra-class variances, ensuring that the texton histograms are similar for regions of the same object class. This double pressure enables learning of the correct level of intra-class compactness in relation to inter-class discrimination power and represents the main contribution of this paper. As we are compressing the histogram whilst preserving meaningful information about the class labels, our approach can be considered as an application of the information bottleneck method [13] to the problem of object categorization.

We consider a mapping ϕ which merges bins rather than dropping them. However, if some bins are uninformative about the class label then they will be merged together and large variances will be learned for the merged bin.

4.2.3 Learning the mapping ϕ

The goal of our learning algorithm is to find the mapping ϕ which maximises the conditional probability of the ground truth labels, given the texton histograms of all training regions. To achieve this, we first, need to compute (3), which can be re-written as

$$\begin{aligned} P(\{\mathbf{H}_n\}|\mathbf{c}) &= \prod_{c=1}^{C}\prod_{i=1}^{T} \int \prod_{n \in R_c} P(H_{ni}|\theta_{ci}) P(\theta_{ci})\, d\theta_{ci} \\ &\equiv \prod_{c=1}^{C}\prod_{i=1}^{T} E_{ci} \qquad (5) \end{aligned}$$

where $\theta_{ci} = (\bar{H}_i, \beta_i)$ and E_{ci} is an evidence term for a particular class c and histogram bin. The integral for E_{ci} can be found analytically to be

$$E_{ci} = (2\pi)^{-\frac{|R|}{2}} \left(\frac{\lambda}{\lambda'}\right)^{\frac{1}{2}} \frac{b^a}{b'^{a'}} \frac{\Gamma(a')}{\Gamma(a)} \qquad (6)$$

where $\lambda' = \lambda + |R|$, $\mu' = \left(\mu\lambda + \sum_{n \in R} H_n^{\frac{1}{2}}\right)/\lambda'$, $a' = a + |R|/2$ and $b' = b + \left(\lambda\mu^2 - \lambda'\mu'^2 + \sum_{n \in R} H_n\right)/2$. It follows that we can evaluate the conditional probability of the histograms $\{\mathbf{H}_n\}$ given a labeling $\mathbf{c}$ by finding the product of (6) over both bins and classes.

To evaluate the conditional probability of the labels (4) exactly would require computing $P(\{\mathbf{H}_n\}|\mathbf{c})$ for each of the C^N labelings – clearly an intractable proposition. However, we are only interested in the relative values of $P(\{\mathbf{H}_n\}|\mathbf{c})$ as we change the $\mathbf{c}$. We wish it to have a high value for the ground truth $\hat{\mathbf{c}}$ and low values for other 'competitive' labelings. To achieve one-vs-all discrimination, we need only consider the alternate labeling where all regions are given the same label $\mathbf{c}^{\text{same}}$. Hence, we make the approximation that we can maximise (4) by instead maximising the quantity $\widetilde{P}$ given by

$$\widetilde{P}(\hat{\mathbf{c}}|\{\mathbf{H}_n\}) = \frac{P(\{\mathbf{H}_n\}|\hat{\mathbf{c}})}{P(\{\mathbf{H}_n\}|\hat{\mathbf{c}}) + P(\{\mathbf{H}_n\}|\mathbf{c}^{\text{same}})}$$

where the prior terms have canceled because of the choice of a uniform prior distribution. Now we can rewrite $\widetilde{P}$ in terms of the mapping ϕ to give

$$\widetilde{P}(\phi) = \frac{P(\{\phi(\mathbf{h}_n)\}|\hat{\mathbf{c}})}{P(\{\phi(\mathbf{h}_n)\}|\hat{\mathbf{c}}) + P(\{\phi(\mathbf{h}_n)\}|\mathbf{c}^{\text{same}})}. \qquad (7)$$

The algorithm we use to learn the mapping which maximises $\widetilde{P}$ consists of the following stages,

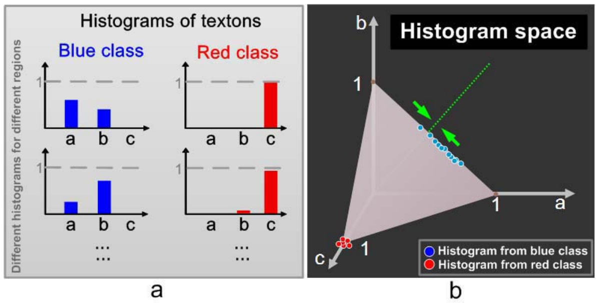

Figure 3: Reducing the dictionary by merging visual words (texton bins). Textons pairs which do not contribute to class discriminability are merged together by our learning algorithm (the pair $\mathbf{ab}$ in the figure).

1. Initialise ϕ to the identity mapping (where no bins are merged).
2. Let ϕ_{ij} be the mapping that merges the pair of bins i and j in ϕ. Compute $\widetilde{P}(\phi_{ij})$ for each pair i and j.
3. Find the mapping $\phi' = \arg\max_{i,j} \widetilde{P}(\phi_{ij})$.
4. If $\widetilde{P}(\phi') > \widetilde{P}(\phi)$, set $\phi = \phi'$ and go to step 2. Otherwise return ϕ as the learned mapping.

The mapping ϕ given by this algorithm defines a grouping of the words in the original dictionary $\mathcal{F}$. This grouping defines a more compact visual dictionary $\mathcal{T}$ which remains discriminative between object classes, with the optimal dictionary size T determined automatically.

One step of the merging algorithm is illustrated in the toy example in fig. 3. We have only the two classes blue and red, and the original dictionary $\mathcal{F}$ consists of the three words $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$. Pixels in regions of the blue class tend to lie only in clusters $\mathbf{a}$ and $\mathbf{b}$, whilst pixels of the red class lie predominantly in cluster $\mathbf{c}$, leading to class texton-histograms of the form shown in fig. 3a. Our learning algorithm has three possible merges to consider $\mathbf{ab}$, $\mathbf{ac}$ and $\mathbf{bc}$. Merging either $\mathbf{ac}$ or $\mathbf{bc}$ will lower $\widetilde{P}$ because it will reduce the discriminability between the two classes. However, merging $\mathbf{ab}$ (projection along the green arrows in fig. 3b) will increase $\widetilde{P}$ because it makes the points in the blue class closer together in histogram space without affecting discrimination between the two classes (fig. 3b). Iterating this basic step produces dictionary size reduction with no loss in class discriminability.

Algorithm efficiency. The computation of $\widetilde{P}(\phi_{ij})$ for each pair of bins can be carried out efficiently since only the terms in (5) relating to bins i and j differ. The number of single bin evidence computations required for the entire learning process is $O(CK^2)$. The efficiency could be further improved by considering only a subset of the possible merges at early stages in the algorithm. We did not find this necessary since we were able to apply the full algorithm to initial dictionary sizes up to $K = 5000$.

4.3. Classification

Once a compact UVD has been obtained we can choose to model object classes in a number of ways. For instance, we could think of describing a class as a set of histograms, each associated with the training regions labeled with the same class label and use nearest neighbour classification. This is clearly a highly multi-modal, non-parametric representation. Given an input test region the closest (e.g. in terms of Euclidean or Mahalanobis distance) training histogram is found and the corresponding object class label returned.

Alternatively, we could use the Gaussian class models with the posterior over the parameters $\boldsymbol{\theta}_c$ that we have learned from the training data. We classify each new (test) histogram $\mathbf{H}'$ by finding the setting of c which maximises $\int P(\mathbf{H}' \,|\, c, \boldsymbol{\theta}_c) P(\boldsymbol{\theta}_c | \{\mathbf{H}_n\}, \hat{\mathbf{c}})\, d\boldsymbol{\theta}_c$. The unimodality of Gaussian distributions may be seen as a disadvantage, however the results section will show how the multi-modal nature of the data is captured by the supervised word merging process. Advantages of the Gaussian class models over nearest neighbours ones are: i) their compactness (storing all training examples can be avoided), and ii) the fact that Gaussian models provide proper *parametric* densities. In the following sections we evaluate and compare the behaviour of parametric and non-parametric class models for object class recognition.

5. Results

In this section we assess the effectiveness of the proposed class-modeling technique by: i) measuring object class recognition accuracy with respect to different image databases, ii) comparing compactness of class models, iii) measuring the effect of our learning algorithm on the class discrimination ratio, iv) showing results from our interactive categorization demo application.

Accuracy of classification in in-house dataset. In order to measure classification accuracy we have split the 240-image in-house database (fig. 2) into 50% training set and 50% test set. The training images are used to estimate both the visual dictionary and the nine class models.

Accuracy of classification for different class models are shown in table 1. If the test image region boundaries were determined from our ground-truth segmentation (and ignoring ground-truth class labels) then a nearest neighbour classification approach (with or without dictionary compression) and Gaussian class models reached pretty much the same accuracy of about 93%. However, the combination of learned Gaussian models and learned UVD achieves the highest compactness of modeling and thus the highest classification efficiency. For this training set classification via multi-class SVM techniques or Gaussian mixture models produced inferior results.

	Recognition accuracy		
	Dict. size	Accuracy	Accuracy (bbox)
N. Neigh.	K=2000	**93.4%**	76.3%
N. Neigh.	T=216	**92.7%**	78.5%
Gaussian	T=216	**93.4%**	77.4%

Table 1: Accuracy of classification for in-house dataset. The Gaussian method is over 140 times faster than nearest neighbours with $K = 2000$.

True	**Inferred label**								
label	Build.	Grass	Tree	Cow	Sky	Aerop.	Face	Car	Bicyc.
Building	**38**			2	1		1	2	1
Grass		**66**	1						
Tree	1	1	**30**						1
Cow				**21**			2		
Sky					**46**				
Aeroplane	4					**11**			
Face							**15**		
Car								**15**	
Bicycle	1								**14**

Table 2: Confusion matrix for in-house data with learned Gaussian class models. Final dictionary size $T = 216$.

As the last column of table 1 shows, using the regions' bounding boxes rather than the more accurate delineation reduced the classification performances. However, we expect more clutter-robust histogram distances [12] to improve the performance even in the case of bounding-box region selection. Furthermore, automatic region segmentation [11] may also bring large improvements in this case.

The confusion matrix for classification using the learned Gaussian class models is reported in table 2. It can be seen that most image regions have been classified correctly into one of the nine object classes (numbers on the main diagonal). However, a few mistakes were made, e.g. four aeroplanes were incorrectly classified as buildings.

Accuracy of classification in the Pascal dataset. Similar experiments were run on the Pascal Visual Object Classes challenge training dataset[3] (587 images[4]). In order to test our algorithm we have once again split the dataset into two equally large training and test sets. The measured classification accuracy is reported in table 3. Notice that in the Pascal dataset only bounding boxes of image regions are provided. By comparing the results in the two tables 3 and 1 we would expect classification accuracy to increase substantially if better region delineation was provided, e.g. through GrabCut automatic segmentation [11].

	Recognition accuracy	
	Dict. size	Accuracy (bbox)
Nearest Neighbour	K=1200	76.9%
Nearest Neighbour	T=134	74%
Gaussian	T=134	73.3%

Table 3: Accuracy of classification for Pascal dataset. The Gaussian method is over 310 times faster than nearest neighbours with $K = 1200$.

True	**Inferred label**			
label	Car	Bicyc.	Motor.	Person
Car	**65**	4	4	2
Bicycle	9	**36**	4	10
Motorbike	11	12	**81**	4
Person	1	10	4	**24**

Table 4: Confusion matrix for Pascal data with learned Gaussian class models. Final dictionary size $T = 134$ textons, with bounding-box only region selection.

The corresponding confusion matrix for Gaussian class models is shown in table 4. Unsurprisingly motorbikes and bicycles are confused with one another. Furthermore, high level of confusion is detected for the class "person" due to the high variability of people's clothing. However, the large majority of object instances have been classified correctly.

Comparing performance of Gaussian class models before and after learning. Figure 4 shows the improvement in classification accuracy when using Gaussian class models before and after learning, for different sizes of the initial visual dictionary. It can be observed that our supervised learning algorithm improves the classification accuracy dramatically, especially for larger numbers of visual words; where the highest accuracy is reached. Notice that, without our learned dictionary, it would be impossible to achieve above 90% accuracy with the Gaussian model (red curve). With the learned dictionary, performance comparable to nearest neighbour classification is achieved.

Figure 5 compares the accuracy of Gaussian class models and nearest neighbour classification for different initial dictionary sizes K. For small initial dictionaries, nearest neighbours is slightly superior to Gaussian models, though the difference has similar magnitude to the differences from different runs of K-means. Then for $K \geq 2000$, their performance becomes the same (if not inverted), which suggests that the merged textons are absorbing the multi-modal nature of the visual object classes.

Model selection. Observing fig. 5 we can notice that for large values of K the performance of nearest neighbour classification starts to degrade. However, ignoring K-means noise the blue performance curve is monotonically non-decreasing. This effect is advantageous since now we need not to worry about choosing the 'optimal' dictionary size

[3] http://www.pascal-network.org/challenges/VOC/voc/index.html

[4] We have ignored the grey-level car-only UIUC images since the simple classification rule "grey image → car" would have artificially boosted our classification results.

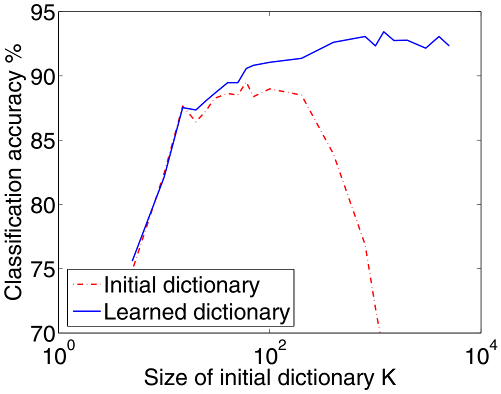

Figure 4: Classification performance for Gaussian class models. Before (red) and after learning (blue), for different sizes of the initial UVD.

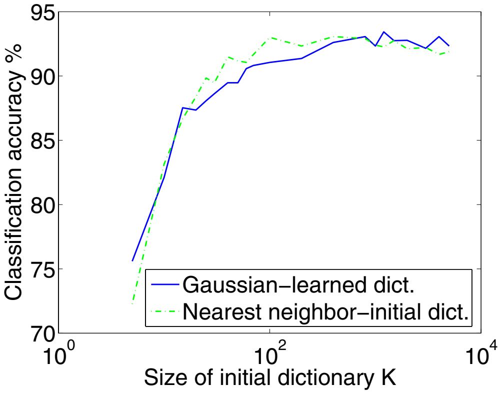

Figure 5: Comparing classification performance for Gaussian class models vs nearest neighbours classification.

since selecting a sufficiently large value (e.g. $K > 3000$) suffices.

Learning features. As an alternative to the hand crafted 17-dimensional filter responses we have also tested our algorithm using 75-dimensional, 5x5 colour patches in the hope of learning automatically discriminative features from raw input pixel data. Interestingly, we found that comparable classification accuracy was achieved if the initial dictionary size was large enough ($K > 1000$) (*cf.* [15]). However, the larger dimensionality of the feature vectors affected both training and testing efficiency.

Information summarization. Reducing the dictionary size to increase classification efficiency without compromising accuracy is fundamental when dealing with large numbers of object classes (e.g. in the order of hundreds or thousands). Moreover, even for a limited number of classes speed may be important i) when scanning entire photographs to detect and classify all the objects contained within, ii) for content-based clustering of web images, or iii) for the analysis of videos. Figure 6 illustrates the compression effect by plotting the automatically computed final UVD size T in relation to the original size K. The relationship is highly non linear, with the ratio T/K shrinking as K grows (the line $T = K$ is drawn for comparison). As K becomes very large, T asymptotically approaches a maximum UVD size $T^\star \approx 230$.

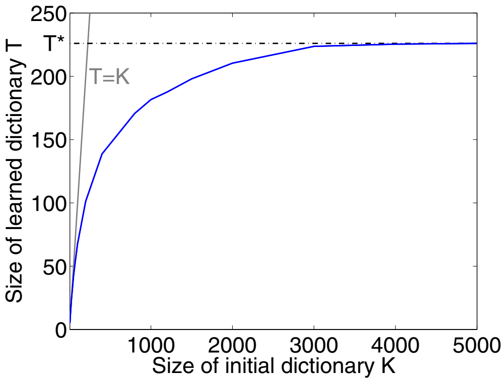

Figure 6: Dictionary size compression. The relationship between initial (K) anc

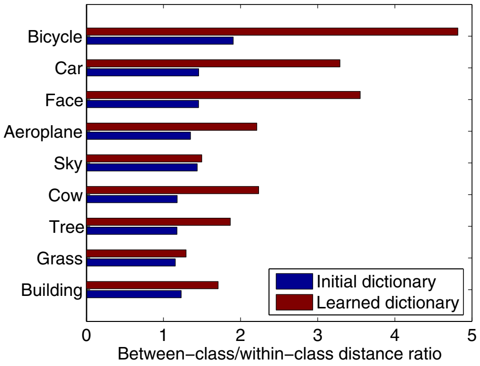

Figure 7: Ratio between inter- and intra- class distances on test set with initial and learned dictionaries. In this experiment the initial dictionary size was fixed at $K = 2000$.

Discrimination ratio. It is informative to look at how our algorithm affects the ratio between average inter-class and intra-class distances. In general, greater classification accuracy is achieved for a greater distance ratio. Figure 7 compares the distance ratio before and after learning. The intra-class distance is the log-probability of a region known to be in the class. The inter-class distance is the log-probability of a region known not to be in the class. We take the average of both and then the ratio. As it can be seen, learning increases the discriminability of *all* nine classes, in many cases quite considerably. The least effect is on the grass and sky classes probably due to the fact that these homogeneous 'objects' are already modeled well by only a small number of textons in the initial dictionary. The most positive effect is on structured objects such as bicycles, faces and cars.

Interactive classification application. In order to further test the findings of this paper we have built an interactive

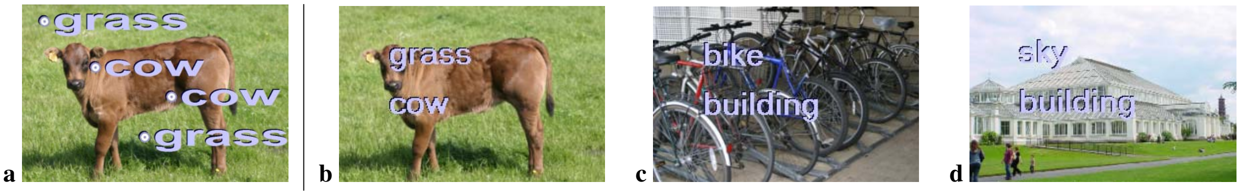

Figure 8: Applications and extensions. a) Single click object categorization: the user "touches" an object and the algorithm associates a category label. b-d) Multi-class object detection: our algorithm automatically lists the object classes contained in the input image. No user interaction is required here.

object recognition demo application where a user selects a rectangular region in an image and the system instantly estimates the associated class label. Twelve exemplar snapshots are shown in fig. 1. Thanks to our appearance-based models, selecting just a portion of the object of interest suffices, thus demonstrating high robustness with respect to occlusions and missing parts.

Single click categorization. High algorithmic efficiency allows us to: i) select a single image point $\mathbf{x}$, ii) run a whole range of classification tests for different sizes and shapes of the regions of interest centred in $\mathbf{x}$, and iii) determine the MAP object class at the selected location in real time. An example of single click class recognition is shown in fig. 8a.

Applications in image understanding. Useful applications of our class modeling algorithm include: i) multi-class object detection, ii) object localization, iii) content-based image segmentation and iv) content-based clustering. For instance, applying single click classification to a regular grid of image positions enables automatic detection of all objects within an image, as shown in fig. 8b-d. Space restrictions negate a more detailed explanation. The reader is kindly invited to browse our web pages for more examples and a video of our interactive recognition demo.

6. Conclusion

This paper has studied the problem of defining and estimating descriptive and compact visual models of object classes for efficient object class recognition. A new supervised learning algorithm has been proposed for estimating appearance-based models from training images. The algorithm is designed to produce highly compact class descriptions with large discrimination power; accuracy and efficiency of classification being essential prerequisites for semantic image retrieval, clustering and editing.

In contrast to previous work here we have avoided focusing only on sparse sets of interest-points or parts. Instead, all pixels are taken into account; with the discriminative features learned automatically. This enables treating both texture-rich and texture-less objects in a unified way.

Surprisingly, our learned Gaussian class models have performed comparably to multi-modal nearest neighbour classification. Advantages of the Gaussian models are their compactness and the fact that they are parametric densities.

Finally, our appearance-based models have turned out to be surprisingly powerful for categorizing both "texture-rich" and "structure-rich" objects.

Currently, we are investigating integration of appearance with local shape information to maintain high class discriminability while increasing the number of object classes. The statistical framework developed in this paper readily allows such integration.

References

[1] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual categorization with bags of keypoints. In *Proc. of the 8th ECCV, Prague*, May 2004.

[2] P. Duygulu, N. de Freitas, K. Barnard, and D.A. Forsyth. Object recognition as machine translation. In *Proc. ECCV*, Copenhagen, 2002.

[3] L. Fei-Fei, R. Fergus, and P. Perona. A Bayesian approach to unsupervised one-shot learning of object categories. In *Proc. of the 9th IEEE ICCV, Nice, France*, pages 1134–1141, October 2003.

[4] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. of IEEE CVPR*, Madison, WI, June 2003.

[5] R. Fergus, P. Perona, and A. Zisserman. A visual category filter for Google images. In *Proc. of the 8th ECCV, Prague*, May 2004.

[6] P. Fryzlewicz and G. P. Nason. A Haar-Fisz algorithm for Poisson intensity estimation. *Journal of Computational and Graphical Statistics*, 13:621–638, 2004.

[7] J. M. Kasson and W. Plouffe. An analysis of selected computer interchange color spaces. In *ACM Transactions on Graphics*, volume 11, pages 373–405, October 1992.

[8] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Extending pictorial structures for object recognition. In *Proc. of BMVC*, London, 2004.

[9] T. Leung and J. Malik. Recognizing surfaces using three-dimensional textons. In *Proc. IEEE ICCV*, Kerkyra, Greece, 1999.

[10] R. W. Picard and T. P. Minka. Vision texture for annotation. *Multimedia Syst.*, 3(1):3–14, 1995.

[11] C. Rother, V. Kolmogorov, and A. Blake. GrabCut -interactive foreground extraction using iterated graph cuts. In *ACM Trans. on Graphics (SIGGRAPH)*, August 2004.

[12] Y. Rubner and C. Tomasi. Texture-based image retrieval without segmentation. In *Proc. IEEE ICCV*, Kerkyra, Greece, 1999.

[13] N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *37th Allerton Conf. on Communications and Computation*, 1999.

[14] S. Ullman, E. Sali, and M. Vidal-Naquet. A fragment-based approach to object representation and classification. In *Proc. 4th Intl. Workshop on Visual Form, IWVF4*, Capri, Italy, 2001.

[15] M. Varma and A. Zisserman. Texture classification: Are filter banks necessary? In *Proc. of IEEE CVPR*, Madison, WI, June 2003.

[16] M. Varma and A. Zisserman. A statistical approach to texture classification from single images. *IJCV*, 62(1–2):61–81, April 2005.