Object localisation using the Generative Template of Features

Citation for published version:
Allan, M & Williams, CKI 2009, 'Object localisation using the Generative Template of Features', *Computer Vision and Image Understanding*, vol. 113, no. 7, pp. 824-838. https://doi.org/10.1016/j.cviu.2009.02.002

Digital Object Identifier (DOI):
http://dx.doi.org/10.1016/j.cviu.2009.02.002

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
Computer Vision and Image Understanding

Object Localization using the Generative Template of Features

Moray Allan, Christopher K. I. Williams *

School of Informatics, University of Edinburgh,
5 Forrest Hill, Edinburgh EH1 2QL, UK

Abstract

We introduce the Generative Template of Features (GTF), a parts-based model for visual object category detection. The GTF consists of a number of parts, and for each part there is a corresponding spatial location distribution and a distribution over 'visual words' (clusters of invariant features). The performance of the GTF is evaluated for object localisation, and it is shown that such a relatively simple model can give state-of-the-art performance. We also demonstrate how a Hough-transform-like method for object localisation can be derived from the GTF model.

Key words:
object recognition, object localization, Generative Template of Features, visual words

* To whom correspondence should be addressed.
Email address: c.k.i.williams@ed.ac.uk (Christopher K. I. Williams).

1 Introduction

Over the last few years there has been a surge of interest in the problem of object recognition for object classes (as opposed to specific objects). We can distinguish between two related problems for object class recognition: classification, meaning detecting whether an object of the given class is present in an image, and localisation, meaning determining the position of an object in an image.

We focus on object localisation using the Generative Template of Features (GTF) model. We assume that each input image has been preprocessed by running a region-of-interest detector and matching the results to some set of 'visual words'. The model, described fully in section 3 below, consists of a number of parts, with each part having a spatial location distribution and a distribution over visual words. Our main contributions are explaining how to train the model in a supervised manner, evaluating its performance on cluttered images, and showing how pose-space prediction methods can be derived from the GTF.

Section 2 below gives a summary of related work on object class recognition. Section 3 describes the GTF model and the relationship between the GTF and scanning window and pose-clustering methods. Section 4 describes some experiments on learning and recognising object classes using the GTF. Section 5 discusses the experimental results and some ways to enhance the GTF's performance.

2 Related work

This section discusses approaches to object category localisation, grouping them as scanning-window methods, pose-clustering methods, and correspondence-based methods.

Scanning window methods run an object detector at different possible object scales and locations across an image, considering each object bounding box hypothesis and searching for maxima in the detector output. For example, Le Cun et al. [1] used a scanning window approach with a neural network-based handwritten-digit detector trained using backpropagation. More recently scanning window approaches were used for example with a face detector using local image patches trained by boosting [2], and a car detector using clustered interest regions [3]. Kapoor and Winn [4] used a located hidden random field to learn discriminative object parts to detect cars and horses. The located hidden random field is a conditional random field extended to assign pixels unobserved object part labels, and to model the spatial relationship between these parts.

Pose-clustering methods (see [5], §18.3) allow individual image features to vote on object instantiation parameters, then look for maxima in the summed voting space. For example, straight lines can be recognised by allowing noisy line segment features to vote for a range of line orientations passing through each feature's location, then looking for vote responses above some predetermined threshold [6]. This approach can be generalised to detect arbitrary shapes [7]. Lowe [8] matches images to models for individual objects using a Hough transform approach. Leibe et al. [9] take a similar approach to the case of object category recognition. They extract image patches at Harris interest points, cluster them using normalised greyscale correlation, and then vote based on the offsets within the object bounding boxes at which a given patch cluster was seen in the training data.

Correspondence-based methods [10] match image features to features in a model. For example, the 'constellation model' [11–14] uses a number of object parts which are found in characteristic positions relative to each other, matching each part to a region in each image. For example, for faces we might think of having the eyes, nose, and mouth as parts. The constellation model learns the joint probability density on part locations. The constellation model can be slow to train, and at test time potentially requires a search over all possible correspondences between image features and object parts. As this is a combinatorial search it is exponentially slow unless aggressive pruning of the search tree can be achieved. Star models [15] and other structures from the more general class of k-fans [16] allow larger number of object parts to be used.

3 Generative Template of Features

This section describes the Generative Template of Features model. We first discuss the main points of the model, then give some information on modelling choices we have made in modelling the background features (section 3.1), object scale (section 3.2), and mixing proportion (section 3.3).

We assume that a region-of-interest detector has been run on each image, and that a local image descriptor like Lowe's SIFT descriptor [17] has been computed for each region of interest. Regions could also be sampled from the images at random [18]. We cluster the descriptors obtained from a set of training images to create a dictionary of 'visual words' (as for example in [3,9]; our clustering procedure is described below in section 4.1). Thus for image m with N_m interest points we have pairs $(\mathbf{x}_{mi}, w_{mi})$, $i = 1, \ldots N_m$, where $\mathbf{x}_{mi}$ denotes the position of feature i in image m, and w_{mi} denotes the visual word to which it matches. X_m is a matrix of N_m feature positions for image m, and W_m is a vector of the N_m corresponding visual words for the image.

Consider an object which has pose variables $\boldsymbol{\theta}$. Here $\boldsymbol{\theta}$ could denote for example the (x, y) position of the object in the image, position plus scale and rotation, or it could be more complex and include information on an object's internal degrees of freedom. Under the model defined in [19] we have

$$p(\boldsymbol{\theta}, X_m, W_m) = p(\boldsymbol{\theta}) \prod_{i=1}^{N_m} p(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}), \tag{1}$$

i.e. each feature $(\mathbf{x}_{mi}, w_{mi})$ is generated conditionally independently given $\boldsymbol{\theta}$.

Since image features may be generated either from background clutter or from a foreground object of interest, we propose a mixture model

$$p(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}) = (1 - \alpha)p_b(\mathbf{x}_{mi}, w_{mi}) + \alpha p_f(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}), \tag{2}$$

where p_b denotes the background model, p_f the foreground model for the object, and α is the probability of choosing to generate from the foreground. The background model may, for example, generate features anywhere in the image and with a broad distribution of visual word types, with $p_b(\mathbf{x}, w) = p_b(\mathbf{x})p_b(w)$. We use a more complex background model, which assigns lower background probability to the foreground area (Figure 1(b)), as described in section 3.1 below.

The term $p_f(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta})$ can be further decomposed using the notion of parts. If z_{mi} determines which of the P possible parts of an object a feature is generated from, then under the object foreground model we have

$$p_f(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}) = \sum_{z_{mi}=1}^{P} p_f(\mathbf{x}_{mi}|z_{mi}, \boldsymbol{\theta}) p(w_{mi}|z_{mi}) p(z_{mi}). \tag{3}$$

This equation means that the location of feature mi, $\mathbf{x}_{mi}$, depends on the part from which it is generated and the object's instantiation parameters, while the visual word w_{mi} is generated from a multinomial distribution that depends on only the identity of the part that the visual word is generated from. Note that it is not absolutely necessary to cluster the descriptors into a discrete set of visual words; one could define $p(w|z)$ over real-valued descriptors, using for example a mixture of Gaussians, and possibly make $p(w|z)$ also vary with the object pose $\boldsymbol{\theta}$.

In our GTF implementation we use a spatial grid of Gaussians $p_f(\mathbf{x}_{mi}|z_{mi}, \boldsymbol{\theta})$ as the object parts which generate foreground feature locations, with each part generating visual words from an associated multinomial $p(w|z)$. Figure 1(a) shows a 6 × 4 GTF. The part locations are transformed to fit the object template by translations (t_1, t_2) and x and y scalings s_1 and s_2; the variance of the Gaussians also scales proportionally to changes in s_1 and s_2. A similar generative model was defined in [20] for black (ink) pixels, without the visual words component.

For any set of images of a particular class of objects which has been normalised to a common reference frame (by translation, scaling, etc.), we expect to see regions which have propensities to generate particular visual words. For example, if we normalise a set of side views of cars, there will be regions towards the bottom left and right of the views which tend to generate visual words associated with wheels. While we define a spatial grid of object parts, it would also be possible to adapt part locations during GTF training.

Support for the hypothesis that there is one object of a given type O_j in the scene, plus background features, can be evaluated by computing

$$p(X_m, W_m|O_j) = \int p(\boldsymbol{\theta}, X_m, W_m|O_j)d\boldsymbol{\theta} \tag{4}$$

for each object model, including a pure background model O_0 to account for the case where there is no object of a known type present. In general the integral in equation 4 is not analytically tractable but if $\boldsymbol{\theta}$ is low dimensional it can be approximated, for example by using numerical quadrature based on a grid of points in $\boldsymbol{\theta}$-space, or by using Laplace's approximation at a mode of $p(\boldsymbol{\theta}|X_m, W_m, O_j)$. Localization of an object with respect to the $\boldsymbol{\theta}$ parameters such as position and scale can be carried out with the GTF by scanning the template over the image at a dense grid of $\boldsymbol{\theta}$ settings, and detecting maxima of $p(\boldsymbol{\theta}|X_m, W_m, Oj)$ in this space, or alternatively by using a coarse grid in $\boldsymbol{\theta}$ space with hill-climbing search.

Unlike correspondence-based methods, the GTF model does not enforce generation from each part. The conditional independence assumption in the generative model gives a non-zero probability that a part is not chosen on any of the draws from the foreground. This is useful behaviour when part of an object is occluded, but it can also lead to incorrect detections. This problem was observed by [20] where it was called the 'beads in white space' problem, as ink generators (beads) could occur without penalty in regions where there were no black (ink) pixels. One way to deal with this would be to use the GTF to find promising regions of $\boldsymbol{\theta}$ space, and then evaluate potential detections with these bounding boxes using a separate discriminatively-trained classifier.

This strategy was used by Revow et al. [20] on the digit recognition problem, and more recently has been used, for example, by Fritz et al. [21] for the recognition of cars, motorbikes, cows and horses.

Most scanning window methods use discriminatively-trained classifiers, but we can also use a scanning window approach with the GTF. Unlike discriminative classifiers, the GTF is capable of being trained in an unsupervised manner. Note that even if we scan an object hypothesis across an image, evaluating possible object bounding boxes in turn, the GTF's likelihood term $p(X_m, W_m|\boldsymbol{\theta})$ still considers the probability of the 'background' features outside the enclosing bounding box, while most discriminative methods only learn the equivalent of the GTF's object foreground model.

3.1 GTF background model

The background model used in our experiments below generates feature locations from a mixture, with probability β assigned to a uniform distribution across the image (in our experiments $\beta = 0.05$), and probability $1-\beta$ assigned to a distribution that generates approximately from a uniform distribution across locations in the image outside the object bounding box, as illustrated in Figure 2.

If we have indicator functions $I_f(\mathbf{x})$, $I_b(\mathbf{x})$, $I(\mathbf{x})$ which are respectively one inside the object's enclosing bounding box, one in the background (outside the object's bounding box), and one anywhere in the image, and which are zero elsewhere, we can declare a background feature-location distribution, independent of visual word identity:

$$\begin{aligned} p_b(\mathbf{x}) &= \frac{\beta}{A} I(\mathbf{x}) + \frac{1-\beta}{A-A_f} I_b(\mathbf{x}) = \frac{\beta}{A} I(\mathbf{x}) + \frac{1-\beta}{A-A_f} \left(I(\mathbf{x}) - I_f(\mathbf{x})\right) \\ &= \left(\frac{\beta}{A} + \frac{1-\beta}{A-A_f}\right) I(\mathbf{x}) - \frac{1-\beta}{A-A_f} I_f(\mathbf{x}), \end{aligned} \tag{5}$$

where A is the area of the image and $A_f = s_1 s_2$ is the foreground area.

To give a differentiable function we approximate $I_f(\mathbf{x})/A_f$ using the GTF's foreground grid of Gaussians, using the same visual word distribution $p_b(w_{mi})$ across all the object parts, such that

$$p_b(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}) = \left(\frac{\beta}{A} + \frac{1-\beta}{A-A_f}\right) p_b(w_{mi}) - \frac{1-\beta}{A-A_f} A_f p_h(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}), \tag{6}$$

where $p_h(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta})$ gives a 'hole' the same shape as the foreground:

$$p_h(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}) = p_b(w_{mi}) \sum_{z_{mi}=1}^{P} p_f(\mathbf{x}_{mi}|z_{mi}, \boldsymbol{\theta}) p(z_{mi}). \tag{7}$$

Compare equation 7 with 3, where each foreground object part had its own visual word distribution $p(w_{mi}|z_{mi})$.

3.2 Scaling the model

We define the GTF template as 1 × 1 pixels, and scale it by x and y scale factors s_1 and s_2 to fit each object bounding box. The grid of GTF component parts is also scaled by s_1 and s_2, so that the parts retain their positions relative to the template bounding box. The location variance with which the Gaussian for each foreground part generates feature locations is scaled similarly.

To model the probability of seeing an object bounding box of a given width

s_1 and height s_2, we fit a Gaussian in log scale space:

$$p(s_1, s_2) = \frac{\sqrt{|\mathbf{S}|}}{2\pi} \times \tag{8}$$

$$\exp\left(-\frac{1}{2}\left(S_{11}(\log s_1 - \mu_1)^2 + S_{22}(\log s_2 - \mu_2)^2 + 2S_{12}(\log s_1 - \mu_1)(\log s_2 - \mu_2)\right)\right) \tag{9}$$

where $\boldsymbol{\mu}$ and $\mathbf{S}$ are the mean and the inverse of the covariance matrix of the Gaussian. The SIFT descriptors used to create visual words are not invariant across all aspect ratio changes, but as this model expresses we do not expect to see extreme variations in aspect ratio between objects of a single class.

We assume that the object centre is generated uniformly across the image.

3.3 Mixing proportion model

We learn a model for the mixing proportion α (see equation 2) from the training data, parameterised by the proportion of the image area covered by the foreground object. The model, learnt by linear regression, is of the form:

$$\alpha = \gamma \frac{s_1 s_2}{A} \tag{10}$$

where A is the area of the image.

3.4 Relation to scanning window methods

The translation and scale invariant probabilistic Latent Semantic Analysis model used in [22] is similar to our model, except that it uses hard ‘cells’ (or box basis functions) instead of overlapping Gaussians, and is applied in an unsupervised learning context. Fergus et al. [22] concentrate on object categorisation; the average precision scores they report for object localisation (their Table 3) are quite poor.

The model of Sudderth et al. [19] does use a mixture of Gaussians. They learn general spatial offsets for the parts rather than using a grid, though note that a sufficiently fine grid of parts can approximate the effect of any learnt part distribution. Their focus is on learning parts which can be shared across object categories such as various kinds of animal. Our system can learn to generate the same visual words for multiple classes, but Sudderth et al. also use a Dirichlet process to share part visual word distributions across multiple classes. They sample over object location hypotheses to estimate the probability that an image is generated by a given object category, where our GTF implementation uses a grid search followed by hill-climbing then calculates an approximate integral.

Fergus et al. [22] and Sudderth et al. [19] carry out training using unsupervised learning. In section 4.1 below we evaluate the performance the GTF can achieve using supervised learning, where examples of the object classes of interest are annotated with bounding boxes in the training data.

3.5 Relation to making predictions in pose-space

To consider different possible object pose parameters in the localisation task, we have to compute $p(\boldsymbol{\theta}|X_m, W_m)$. Taking logs of equation 1 we obtain

$$\log p(\boldsymbol{\theta}, X_m, W_m) = \log p(\boldsymbol{\theta}) + \sum_{i=1}^{N_m} \log p(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}). \qquad (11)$$

As the data (X_m, W_m) are fixed we have $p(\boldsymbol{\theta}, X_m, W_m) \propto p(\boldsymbol{\theta}|X_m, W_m)$, with $p(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta})$ viewed as a function of $\boldsymbol{\theta}$. Thus the generative model can be used to hypothesize detections in $\boldsymbol{\theta}$-space by finding the maxima of $p(\boldsymbol{\theta}|X_m, W_m)$, for example by hill-climbing. Such an explanation of the probabilistic Hough transform can be found, for example, in [23], although without the use of specific visual word features, which provide more information and thus tighter distributions.

To spell this out further, consider a distinctive visual word which occurs in only one position on an object. This feature will be predictive of the location of the centre of the object, but as it can also be generated from the background part there is also an associated broad outlier distribution as derived from quation 3.

Equation 11 shows how to run the generative model backwards to provide predictions in parameter space. However, given training data with features $\{(\mathbf{x}_{mi}, w_{mi})\}$ it is natural to build predictors for $p(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi})$, for example by creating a Parzen windows estimator for $p(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi})$ [9]. How should we then combine these predictions from each feature in order to obtain $p(\boldsymbol{\theta}|X_m, W_m)$? Fortunately Bayes' rule comes to our aid, as

$$p(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}) = \frac{p(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi})p(\mathbf{x}_{mi}|w_{mi})p(w_{mi})}{p(\boldsymbol{\theta})}. \tag{12}$$

Here $p(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi})$ is obtained from the predictive model, $p(w_{mi})$ is just the marginal probability of visual word w_{mi} over the training set, and $p(\mathbf{x}_{mi}|w_{mi})$ is the probability of seeing a visual word of type w_{mi} in position $\mathbf{x}_i$. This could be estimated using a density estimator for the location of features of a given type in the collection of training data. Alternatively, if $p(\boldsymbol{\theta})$ has a non-informative location component, then we might expect that $p(\mathbf{x}_{mi}|w_{mi})$ should be uniform across locations in the image. This use of Bayes' theorem to replace likelihood terms with predictive distributions has been called the *scaled likelihood* method, see for example [24].

Putting equations 11 and 12 together we obtain

$$\log p(\boldsymbol{\theta}|X_m, W_m) = \sum_{i=1}^{N_m} \log p(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi}) - (N_m - 1)\log p(\boldsymbol{\theta}) + c, \tag{13}$$

where c is a constant independent of $\boldsymbol{\theta}$. Thus we have shown rigorously how to obtain $p(\boldsymbol{\theta}|X_m, W_m)$ from individual predictions $p(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi})$ up to a normalization constant. Note, however, that to compute the marginal likelihood (equation 4) from equation 13 additional terms involving $p(\mathbf{x}_{mi}|w_{mi})$ and $p(w_{mi})$ must be included.

Recently, Leibe et al. [9] have used such ideas to predict an object's location based on the observed position of visual words. However, we note that the equation they use (their equation 6), is, in our notation,

$$\text{score}_m(\boldsymbol{\theta}) = \sum_{i=1}^{N_m} p_f(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi}). \tag{14}$$

Equation 14 does not at first sight agree with equation 1: for a start it sums probabilities rather than multiplying probabilities or summing log probabilities. However, using equation 2 we have

$$\begin{aligned}\prod_{i=1}^{N_m} p(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta}) = \prod_{i=1}^{N_m} p_b(\mathbf{x}_{mi}, w_{mi}) \times& \\ \left[(1-\alpha)^{N_m} + \alpha(1-\alpha)^{N_m-1} \sum_{i=1}^{N_m} \frac{p_f(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta})}{p_b(\mathbf{x}_{mi}, w_{mi})} + O(\alpha^2)\right]&.\end{aligned} \tag{15}$$

If α is small and $p(\boldsymbol{\theta})$ is non-informative w.r.t. location then using equation 12 for $p_f(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta})$ we obtain to first order

$$p(\boldsymbol{\theta}|X_m, W_m) = c_0 + c_1 \sum_{i=1}^{N_m} \frac{p_f(\mathbf{x}_{mi}, w_{mi})}{p_b(\mathbf{x}_{mi}, w_{mi})} p_f(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi}), \tag{16}$$

where c_0 and c_1 depend on the image features but not on $\boldsymbol{\theta}$, and $p_f(\mathbf{x}_{mi}, w_{mi}) =$

$\int p_f(\mathbf{x}_{mi}, w_{mi}|\boldsymbol{\theta})p(\boldsymbol{\theta})d\boldsymbol{\theta}$. Minka [25] has also discussed how a robustified product of probabilities gives rise to a sum of probabilities to first order.

Furthermore, if $p(\boldsymbol{\theta})$ has a non-informative location component then the spatial part of $p_f(\mathbf{x}_{mi}, w_{mi})$ will be non-informative and we can refine equation 16 to obtain

$$p(\boldsymbol{\theta}|X_m, W_m) = c_0 + c_2 \sum_{i=1}^{N_m} \frac{p_f(w_{mi})}{p_b(w_{mi})} p_f(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi}), \tag{17}$$

where $p_f(w) = \sum_{z=1}^{P} p(w|z)p(z)$, the weighted average of the multinomial vectors in the foreground parts. Equation 17 is close to equation 14, though note the weighting of each predictive distribution $p_f(\boldsymbol{\theta}|\mathbf{x}_{mi}, w_{mi})$ by the factor $p_f(w_{mi})/p_b(w_{mi})$. If visual word w_{mi} is more probable under the background model then its prediction will be discounted. We note that Dorko and Schmid [26] have discussed selecting discriminative foreground features for use in equation 14, but that their criterion is based on intuitive arguments rather than on a formal derivation.

4 Experiments

In the experiments below we use the data from the PASCAL 2005 Visual Object Classes challenge[1] [27]. The data set consists of a large set of images, each of which contains at least one labelled object against cluttered backgrounds of many unlabelled objects. The labelled objects belong to four categories: bicycles, cars, motorbikes, and people. In the first set of experiments we use the 'train' and 'val' data sets as training and test sets respectively to see how the GTF's performance varies with different parameter choices, while in the

[1] http://www.pascal-network.org/challenges/VOC/voc2005/

second set of experiments we use ‘train’ and ‘val’ combined as a training set, and the ‘test1’ data set as test data. Note that the PASCAL data set has different properties from many other data sets used in image classification tasks, such as the ‘Caltech 5’ data: there may be multiple objects in each image, and there is a high degree of background clutter.

The task is to detect objects of the four categories in test images: each detection should state the type of the object, as well as its position in the image and the width and height of its bounding box. A detection is accepted as correct if the intersection between the prediction and true object covers at least half the area of a bounding box drawn to enclose both, as in [27]. Each detection must be assigned a confidence value. The PASCAL challenge uses two evaluation measures to compare object detection systems: localisation performance is measured by average precision, while image classification performance is measured by the area under the receiver-operating-characteristic curve. The GTF is primarily an object localisation system, but by assigning confidences to the detections it makes we can also use it as a classifier.

4.1 Implementation details

This section gives some details about the GTF implementation used in the experiments below (additional explanation is given in [28]).

We used a GTF with a grid of 8 by 8 Gaussian components for the GTF parts. x and y scale factors s_1 and s_2 are used to bring the template into registration with objects in training images, and to fit it to object instantiation hypotheses in test images. For any given object centre and scale factors we can translate and scale the template and its component Gaussians to calculate $p(\boldsymbol{\theta}, X_m, W_m)$, where $\boldsymbol{\theta} = (t_1, t_2, s_1, s_2)$.

To search for $\boldsymbol{\theta}$ to optimize $p(X_m, W_m|\boldsymbol{\theta})$, we initially search over a coarse grid of positions at a number of scales. The scales for grid search are chosen based on the range of scales seen in the training data, with a factor of $\sqrt{2}$ between each scale, and the grid step size for the search at each selected scale is given by tiling the image with object hypotheses of that scale. We then use conjugate gradient ascent to refine $\boldsymbol{\theta} = (t_1, t_2, s_1, s_2)$, taking the maximum probability object locations found at the various scales as initialisations for gradient ascent. Expectation maximisation could also be used for this search. After finding local maxima by gradient ascent, we use Laplace's method to estimate the probability mass in each region, fitting a Gaussian to the second partial derivatives at each maximum (see [28] for details). The maximum corresponding to the region with highest mass is chosen as the best detection for the image.

In general learning the GTF requires estimation of the distributions $p(z)$, $p(\mathbf{x}|z)$ and $p(w|z)$ for each part. However, we fix $p(\mathbf{x}|z)$ using a spatial grid of Gaussians, rather than adapting the object part locations. Given training images for each object class annotated with bounding boxes we can use supervised learning to estimate $p(w|z)$. Each bounding box for a given object class is rescaled so as to be centered and have the same area as the template. (We use separate x and y scaling factors, so the rectangular bounding boxes can be brought into perfect alignment.) Given these aligned data it is straightforward to learn the parameters of the template by EM. Since we keep the background model's uniform distribution mixing proportion, β, small (see section 3.1), we can learn the foreground and background visual word distributions separately, using training features from only inside or only outside bounding boxes appropriately, making training much faster. $p(w|z)$ is found by the following update equation:

$$p(w=a|z=j) \leftarrow \frac{\sum_{m=1}^{M}\sum_{i=1}^{N_m} p(z_{mi}=j|\mathbf{x}_{mi},w_{mi})\delta(w_{mi}=a)}{\sum_{m=1}^{M}\sum_{i=1}^{N_m} p(z_{mi}=j|\mathbf{x}_{mi},w_{mi})} \tag{18}$$

where $\delta(w_{mi}=a)$ is a zero/one indicator function.

Figure 6 illustrates a trained 600-cluster 8×8 GTF for each object class, by showing the visual words most strongly associated with each component part (highest $p(z|w)$; sorting by $p(w|z)$ favours frequently-seen uninformative visual words). A representative image is chosen for each visual word from the foreground for the object class in question. The bicycle GTF shows a variety of wheel features all over the template, as bicycles are seen in many different poses in the training images, from diverse viewpoints. For the car GTF various wheel features can be seen at the bottom left and bottom right of the template. Wheel and handlebar features can be seen on the motorbike GTF. The person GTF shows some face features near the top of the template, and foot features at the bottom.

4.2 Learning visual words

We preprocess the data by scaling down larger images to fit within a 640×640 pixel square, preserving their aspect ratios, and then use two interest point detectors to find the Harris affine [29,30] and maximally stable extremal [31] regions of interest. [2] For each image we run the two region detectors, combine

[2] We thank the Oxford Visual Geometry Group for making their feature detector code available at `http://www.robots.ox.ac.uk/~vgg/research/affine/`.

the lists of regions which they find, then calculate a descriptor for each region. We use a 128-dimensional SIFT descriptor [32,17] to represent each region's appearance.

To create 'visual words' we cluster features from training images. We take separately the features found within the bounding boxes of each object type (the PASCAL training data includes manually-drawn bounding boxes for the object classes of interest), and the features found in the background of images outside all object bounding boxes, running k-means clustering on the descriptors for each set of features. With four object classes, we run five separate clusterings, one for each class and one for background features, then finally combine the five sets of cluster centres. For example, if we use k-means clustering to find 120 cluster centres for each class, we then combine these to obtain an overall clustering with 600 cluster centres.

Each object's k-means clustering was run 12 times, with the cluster centres which gave the lowest mean descriptor-to-centre distances chosen to go into the final combined clustering. On each run the cluster centres were initialised to a different randomly-chosen set of k feature descriptors.

Figure 3 shows the foreground feature count per cluster within car training set bounding boxes, the background feature count per cluster in training set images where cars occur, and the foreground count as a proportion of the sum of the two counts. This proportion is the 'purity' of the cluster, describing how strongly cluster membership identifies a feature as foreground rather than background, and thus how useful it is in object localisation. Since the cars are the second object class, clusters 81–160 of the 400 are those derived from k-means clustering of the car foreground features. These clusters have the highest counts in the foreground, and the highest car purity, while clusters 321-400, which were derived from clustering background features, have the highest counts in the background, and lowest purity.

Figure 4 shows a representative image for each cluster of a combined clustering with 80 k-means clusters per class, giving a total of 400 clusters. For each cluster, the image region whose SIFT descriptor is nearest the cluster centre is displayed, rotated and scaled from the original image region to a fixed-size square image. The images are shown sorted by the clusters' purity for the car class, left-to-right, top-to-bottom.

Using the same clustering, Figure 5 shows the 12 highest purity clusters for the car class. The features from the cars in the training set which match to each cluster are shown, with each region represented by a scaled image of the region contents displayed at the region centre's position in a normalised object bounding box. Normalising the object bounding boxes to a unit square brings the objects into approximate correspondence, although the objects vary in shape, their pose is not fully labelled, and the bounding boxes sometimes exclude truncated parts of objects.

The highest-purity clusters largely correspond to wheel-like features, which are rarely found in the background. These features tend to occur towards the bottom left and bottom right of the bounding box. Note that the multi-scale character of the features in use mean that some features describe large regions of objects, such as a wheel in context with the car bodywork, or even an entire car. Features belonging to the same cluster may look dissimilar, since cluster membership is based on distance in SIFT descriptor space rather than on direct comparisons between image patch appearances.

4.3 Evaluation measures

The performance evaluation below uses the measures defined in sections 4.1 and 5.1 of [27], calculating the performance of each object category detector in terms of average precision (AP) and the area under the receiver-operating-characteristic curve (AUC). The average precision evaluates object detection and localisation performance, while the area under the receiver-operating-characteristic curve evaluates image classification performance. The average precision here is the mean precision at a set of 11 equally-spaced recall levels. Both performance measures give values in the range $[0, 1]$, with perfect results giving a score of one. Each object category detector is run on the whole set of test images, including images where there is no object of the given category.

To generate precision-recall curves we need to assign a confidence to each hypothesis. We set this confidence value based on the ratio between the probability of the hypothesis under the fitted GTF model and its probability under a GTF where the foreground and background components share the same 'background' visual word distribution. Making this comparison between the probability under class and non-class models prevents the confidence values being dominated by the probability assigned to the locations of the image features. We find the log of the ratio of the probabilities, then set the confidence to its average per region of interest, to allow a fair comparison between images where different numbers of regions are detected.

In the experiments here we only look for a single object in each image. Higher recall could be achieved by allowing multiple detections per image.

4.4 Results

This section looks at the GTF's performance on the PASCAL Visual Object Classes Challenge 2005 test data. We use a single set of parameters, which were chosen by looking at the performance on the validation data, with the aim of finding a compromise which gives good localisation performance for all the object categories. A GTF with 8×8 component parts was used, with each part generating feature locations from a Gaussian with variance $\left(\frac{1}{8}\right)^2$. The image regions were left at their detected scales, and visual words were created with 120 clusters for each of the four classes and 120 for background features. (These experimental results are examined in more detail in [28].)

Table 1 shows evaluation scores for the performance on the PASCAL test data of four GTF object category detectors trained with the same parameters on the combined training and validation data. The AP and AUC rows give the average precision and area under the ROC curve.

The AP2 and AUC2 rows show results with the additional assumption that there is only one class of object present in the image. Each model's output is compared with the maximum output from the other class models on the same image. Since in the PASCAL 2005 Visual Object Classes data there are relatively few images with more than one object, this increases performance in all cases except when the model for the class in question is much better than the other classes' models.

The recall row shows the number of images where the chosen bounding box prediction corresponded to a true object of the class in question. Since we only make one detection per image, the maximum number of objects we could detect would be the number of images, while the maximum AP score we could achieve would be the number of images divided by the number of objects, if we detected this proportion of the objects with precision one.

Figures 7 and 8 show the precision-recall and ROC curves corresponding to the AP2 and AUC2 numbers in the Table. The ROC curves make clear that there is a large variation in classification performance across the classes, with the motorbike classifier by far the best, then the bicycle classifier, then the car classifier, with the person classifier significantly worse again. The precision-recall curves for the bicycle and car detectors show a similar fall-off to different recall levels. As well as reaching a much higher recall, the precision curve for the motorbike classifier remains much flatter, at a high precision level, until its final rapid fall. The precision-recall curve for the person detector shows that only a few people are found, with a low level of precision.

The images where the highest-confidence correct detections are made are unoccluded views of objects from typical viewpoints, such as side-views of cars and motorbikes. The highest-confidence motorbike images also have plain backgrounds. The highest-confidence bicycles are all fairly large in their images, with both wheels clearly visible. The highest-confidence people are also comparatively large in their images, and are dark against light-coloured backgrounds.

A few of the highest-confidence images with incorrect localisations are spurious detections. Other high-confidence incorrect localisations are seen in unusual poses or from unusual viewpoints, or show only part of the object in question. Some of the highest-confidence incorrect detections have multiple objects of the class at similar scales, a case not dealt with by our current implementation of the model. In the highest-confidence person images there are several insufficiently-accurate localisations where a head- or torso-sized bounding box prediction has been made although more of the person is in fact visible in the image.

Table 2 shows evaluation scores for object category detectors trained using the same parameters but tested on the PASCAL 2005 Visual Object Classes ‘test2’ data. This data set contains images collected from Google Image Search, with different properties from the training set or main test set. The degradation in performance seen here also occurs with other methods – as well as perhaps including objects that are harder to detect than the manually compiled images in the main data sets, the ‘test2’ set presents a transfer learning problem.

Table 3 shows the evaluation scores obtained when the true object locations are used instead of predicted object locations. This isolates classification from localisation. Since the overall system is not changed, we still only make one object detection per image. The AP and AP2 numbers improve significantly due to the higher recall level from using the true object locations. The AUC and AUC2 numbers are slightly improved. The classes where our detection performance is worse improve more: the scores improve least on motorbikes, and most on people.

Figure 9 shows $p(\boldsymbol{\theta}|X_m, W_m)$ for some example images, scanning the object hypothesis centre across the image while keeping the object hypothesis scale fixed at the true object size. The pixel intensities represent the log probability for each location, normalised to use the whole range of intensities from full black to full white. The motorbike example shows a clear distinction between class and non-class features, so that the probability density is a roughly Gaussian blob around the true object location. The bicycle and car examples show significant noise in textured regions of the background, though the true object position can still be clearly seen. The person example has the highest relative probability level in the background, as some of the background texture such as the window shutters on the building match quite well with the person GTF.

Figure 10 compares $p(\boldsymbol{\theta}|X_m, W_m)$ under each of the four class GTFs for the bicycle image from Figure 9. The four plots are shown using the same intensity scale. The probability mass for the correct object class's GTF is more concentrated than the other classes', as well as this class distribution's peak being higher than the others'.

Figure 11 compares $p(\boldsymbol{\theta}|X_m, W_m)$ as the scale of the object hypothesis changes, for the bicycle image as in Figures 9 and 10. Each plot shows the probability distribution across the image for an object hypothesis based on the true object bounding box's proportions, but scaled by some factor in x and y. The four plots are shown using the same intensity scale. There is a higher probability across the image at the correct object scale than for the smaller or larger object hypotheses. A maximum is visible at the true object location in each of the plots, but in the plots for smaller object hypotheses there are an increasing number of local maxima, as it is easier for the image background to match the learnt GTF when it is examined at smaller scales.

Figure 12 shows $p(\boldsymbol{\theta}|X_m, W_m)$ for two example images with multiple objects. The two cars in the first image are in fact clearly visible in the plot of the probability distribution, and multiple maxima are visible for the image with people. This suggests that even without extending the model search to deal with multiple objects, a 'greedy' approach that removed image features responsible for a detection and searched again could find some additional objects and increase performance. Local probability maxima can also be seen for each of the people in the second image.

Figure 13 compares $p(\boldsymbol{\theta}|X_m, W_m)$ for a number of object hypothesis scales, for the highest-confidence car image with an incorrect detection. The top left plot shows the location probability distribution for the scale of the left-hand car, 132×53. As in the example in Figure 12, both cars are clearly visible; a lower peak can also be seen midway between the two cars, for a hypothesis which uses the back wheel of the first car and the front wheel of the second. Unlike the example in Figure 12, where a correct localisation was made (so plots for different object hypothesis scales would show lower probabilities), the global maximum for this image corresponds to a stretched bounding box of 365×35, much wider than, and less tall than the true objects. The plot for this scale is shown at the bottom right. Here the maximum probability location is between the two cars, with the bounding box now including both cars.

Table 4 compares the AP and AUC performance of the GTF as in Table 1 with the performance achieved by other methods. The Table includes AP and AUC scores for the Darmstadt Implicit Shape Model entry in the Visual Object Classes challenge, and for the best entry for each class in each category, taken from [27]. The ISM result is included as it is the method in the challenge most similar to the GTF.

For object detection and localisation, as measured by average precision, the GTF beats all the methods from the challenge on the bicycle, motorbike and person classes. On the cars, the class where we lose most from not dealing with multiple objects per image, we have a performance level similar to the

ISM.

The GTF is primarily a detection method, and its classification performance, as measured by the area under the ROC curve, is less competitive. It performs better than the ISM, but is beaten by the best support vector machine-based bag-of-features methods from the PASCAL challenge. The GTF parameters we have been using here were chosen primarily to give good localisation performance; further exploration of the GTF parameter space would give improved classification results.

Both the AP and AUC scores here could probably be improved by optimising the parameters for each class GTF separately, rather than using the same compromise GTF parameters for all four classes, and by optimising the parameters for AUC separately from AP.

5 Discussion

This paper described the Generative Template of Features (GTF), a parts-based model for visual object category detection. We showed how to use the model in a supervised manner, evaluated its localisation and classification performance on cluttered images, and examined its relation to pose-clustering methods.

The GTF's performance could be improved by learning multiple aspects for each class (see for example [33]), rather than combining all views of a class into a single GTF as we do here. For example, we could learn separate visual word distributions for front, side, and rear views of cars. It would also be possible to alter the GTF to use a 3D geometric model. In either case additional annotation data could be provided to label the object aspect, or unsupervised learning could be used.

It is straightforward to extend the Generative Template of Features model to allow multiple objects. One way to handle multiple objects in a scene is to follow the treatment of Sudderth et al. [19]. They extend $\boldsymbol{\theta}$ to hold the instantiation parameters for each object, and define mixing proportions for each object and the background. This approach ignores occlusion, but it would be quite straightforward to use a layered model and to reason about occlusion so as to generate only from visible parts. Alternatively, we might expect that individual models could be run to find good regions of $\boldsymbol{\theta}$-space for the given model, and that the robust background model would explain features from other objects. This parallels the work of Williams and Titsias [34] where such an approach was used to propose good locations for sprite models individually, and a layer ordering was determined in a second pass. Extending the model to allow multiple objects directly makes the search space much larger, but object detection can be sped up by using a greedy approximation to this model: we can start by searching for a single object, then discount image features which have been used in the foreground of the first detection and search again.

6 Acknowledgements

MA gratefully acknowledges support through a research studentship from Microsoft Research Ltd. This work was supported in part by the IST Programme of the European Community, under the PASCAL Network of Excellence, IST-2002-506778. This publication only reflects the authors' views. We thank the anonymous reviewers and area editor for their comments which helped improve

the paper.

References

[1] Y. Le Cun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Handwritten digit recognition with a back-propagation network, in: Advances in Neural Information Processing Systems 2 (NIPS 1989), 1990, pp. 396–404.

[2] P. A. Viola, M. J. Jones, Robust Real-time Face Detection, International Journal of Computer Vision 57 (2) (2004) 137–154.

[3] S. Agarwal, A. Awan, D. Roth, Learning to detect objects in images via a sparse, part-based representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (11) (2004) 1475–1490.

[4] A. Kapoor, J. Winn, Located hidden random fields: Learning discriminative parts for object detection, in: Proceedings of the European Conference on Computer Vision, Vol. 3, 2006, pp. 302–315.

[5] D. A. Forsyth, J. Ponce, Computer Vision: A Modern Approach, Prentice Hall, Upper Saddle River, New Jersey, 2003.

[6] P. V. C. Hough, Methods and means for recognizing complex patterns, u.S. patent 3069654 (December 1962).

[7] D. H. Ballard, Generalizing the Hough transform to detect arbitrary shapes, Pattern Recognition 13 (2) (1981) 111–122.

[8] D. G. Lowe, Local feature view clustering for 3D object recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2001, pp. 682–688.

[9] B. Leibe, A. Leonardis, B. Schiele, Combined object categorization and segmentation with an Implicit Shape Model, in: Workshop on Statistical Learning in Computer Vision, at ECCV 2004, 2004, pp. 17–32.

[10] W. E. L. Grimson, Object Recognition by Computer, MIT Press, Cambridge, MA, 1990.

[11] M. C. Burl, M. Weber, P. Perona, A probabilistic approach to object recognition using local photometry and global geometry, in: Proceedings of the European Conference on Computer Vision, Vol. 2, 1998, pp. 628–641.

[12] M. Weber, M. Welling, P. Perona, Unsupervised learning of models for recognition, in: Proceedings of the European Conference on Computer Vision, 2000, pp. 18–32.

[13] M. Weber, M. Welling, P. Perona, Towards automatic discovery of object categories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2000, pp. 101–108.

[14] R. Fergus, P. Perona, A. Zisserman, Object class recognition by unsupervised scale-invariant learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2003, pp. 264–271.

[15] R. Fergus, P. Perona, A. Zisserman, A sparse object category model for efficient learning and exhaustive recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 380–387.

[16] D. Crandall, P. Felzenszwalb, D. Huttenlocher, Spatial priors for part-based recognition using statistical models, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2005, pp. 10–17.

[17] D. G. Lowe, Distinctive image features from scale-invariant keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110.

[18] E. Nowak, F. Jurie, B. Triggs, Sampling strategies for bag-of-features image classification, in: Proceedings of the European Conference on Computer Vision, Vol. 4, 2006, pp. 490–503.

[19] E. B. Sudderth, A. Torralba, W. T. Freeman, A. S. Willsky, Learning hierarchical models of scenes, objects, and parts, in: Proceedings of the International Conference on Computer Vision, Vol. 2, 2005, pp. 1331–1338.

[20] M. Revow, C. K. I. Williams, G. E. Hinton, Using generative models for handwritten digit recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 18(6) (1996) 592–606.

[21] M. Fritz, B. Leibe, B. Caputo, B. Schiele, Integrating representative and discriminant models for object category detection, in: Proceedings of the International Conference on Computer Vision, Vol. 2, 2005, pp. 1363–1370.

[22] R. Fergus, L. Fei-Fei, P. Perona, A. Zisserman, Learning object categories from Google's image search, in: Proceedings of the International Conference on Computer Vision, Vol. 2, 2005, pp. 1816–1823.

[23] R. S. Stephens, Probabilistic approach to the Hough transform, Image and Vision Computing 9 (1) (1991) 66–71.

[24] N. Morgan, H. A. Bourlard, Neural networks for statistical recognition of continuous speech, Proceedings of the IEEE 83 (5) (1995) 742–770.

[25] T. P. Minka, The 'summation hack' as an outlier model, technical note (August 2003).
URL
http://research.microsoft.com/~minka/papers/minka-summa%tion.pdf

[26] G. Dorko, C. Schmid, Object class recognition using discriminative local features, Tech. Rep. RR-5497, INRIA Rhône Alpes (2005).

[27] M. Everingham, A. Zisserman, C. Williams, L. V. Gool, M. Allan, C. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorko, S. Duffner, J. Eichhorn, J. Farquhar, M. Fritz, C. Garcia, T. Griffiths, F. Jurie, D. Keysers, M. Koskela, J. Laaksonen, D. Larlus, B. Leibe, H. Meng, H. Ney, B. Schiele, C. Schmid, E. Seemann, J. Shawe-Taylor, A. Storkey, S. Szedmak, B. Triggs, I. Ulusoy, V. Viitaniemi, J. Zhang., Selected Proceedings of the First PASCAL Challenges Workshop, LNAI, Springer-Verlag, 2005, Ch. on the Pascal Visual Object Classes Challenge.

[28] M. Allan, Sprite learning and object category recognition using invariant features, Ph.D. thesis, School of Informatics, University of Edinburgh (2007).

[29] K. Mikolajczyk, C. Schmid, An affine invariant interest point detector, in: Proceedings of the European Conference on Computer Vision, 2002, pp. 128–142.

[30] F. Schaffalitzky, A. Zisserman, Automated scene matching in movies, in: Proceedings of the Challenge of Image and Video Retrieval, 2002, pp. 186–197.

[31] J. Matas, O. Chum, M. Urban, T. Pajdla, Robust wide baseline stereo from maximally stable extremal regions, Computer Vision and Image Understanding 22 (10) (2004) 761–767.

[32] D. G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the International Conference on Computer Vision, Vol. 2, 1999, pp. 1150–1157.

[33] E. Seemann, B. Leibe, B. Schiele, Multi-aspect detection of articulated objects, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, 2006, pp. 1582–1588.

[34] C. K. I. Williams, M. K. Titsias, Greedy learning of multiple objects in images using robust statisics and factorial learning, Neural Computation 16 (5) (2004)

1039–1062.

	bicycle	car	motorbike	person
AP	0.265	0.411	0.760	0.014
AP2	0.467	0.422	0.888	0.030
AUC	0.855	0.936	0.908	0.822
AUC2	0.966	0.943	0.995	0.890

Table 1

Test data performance evaluation of 8×8 GTF, variance $= \left(\frac{1}{8}\right)^2$, region scale factor $= 1$, 120 clusters per class.

	bicycle	car	motorbike	person
AP	0.084	0.118	0.092	0.008
AP2	0.147	0.114	0.360	0.009
AUC	0.618	0.704	0.558	0.628
AUC2	0.713	0.740	0.852	0.693

Table 2

Performance evaluation on ‘test2’ data of 8×8 GTF, variance $= \left(\frac{1}{8}\right)^2$, region scale factor $= 1$, 120 clusters per class.

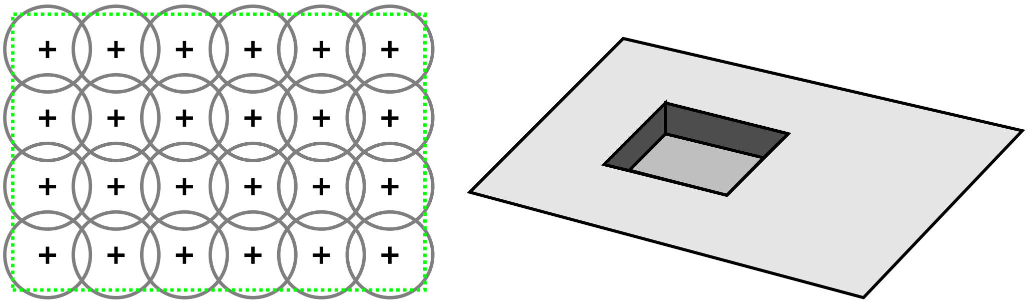


Fig. 1. (a) GTF foreground model with grid of Gaussians; (b) background model.

	bicycle	car	motorbike	person
	bicycle	car	motorbike	person
AP	0.547	0.775	0.823	0.345
AP2	0.766	0.781	0.897	0.455
AUC	0.859	0.970	0.929	0.877
AUC2	0.962	0.975	0.992	0.938

Table 3

Classification-only test data performance evaluation of 8×8 GTF, variance $= \left(\frac{1}{8}\right)^2$, region scale factor $= 1$, 120 clusters per class.

	bicycle	car	motorbike	person
GTF AP2	0.467	0.422	0.888	0.030
ISM AP	—	0.468	0.865	—
best PASCAL AP	0.119	0.613	0.886	0.013
GTF AUC2	0.966	0.943	0.995	0.890
ISM AUC	—	0.578	0.919	—
best PASCAL AUC	0.982	0.992	0.998	0.979

Table 4

Comparison of performance of GTF as in Table 1 with other methods' performance.

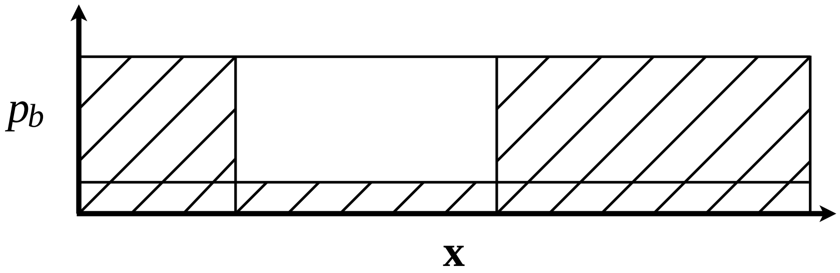


Fig. 2. Background feature-location model: uniform across image plus uniform outside the object bounding box.

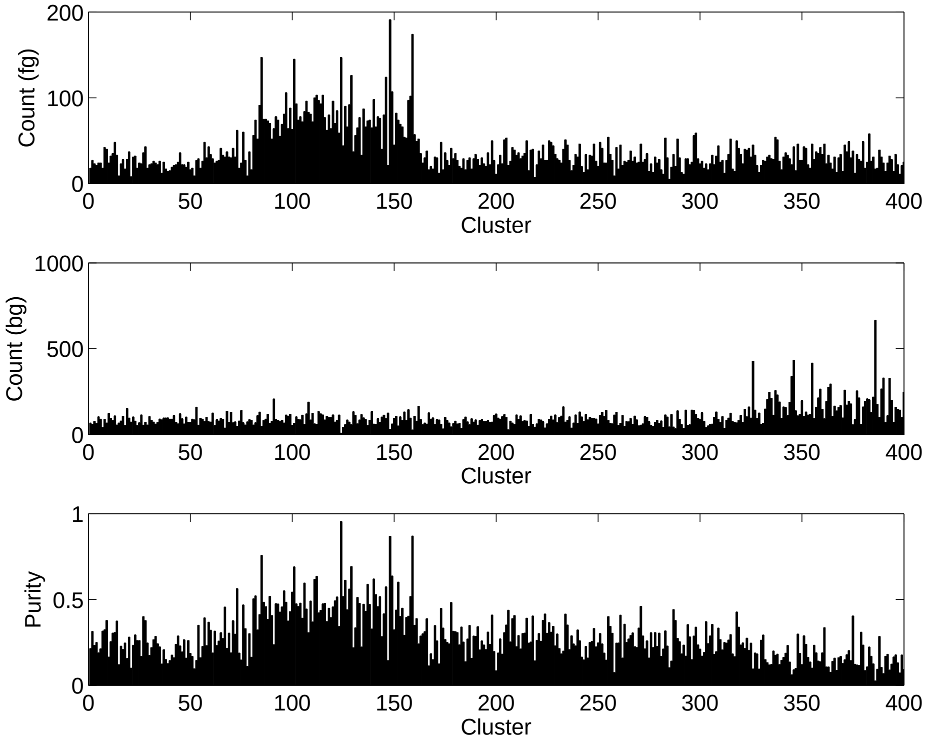


Fig. 3. Cluster purity for car features in example clustering.

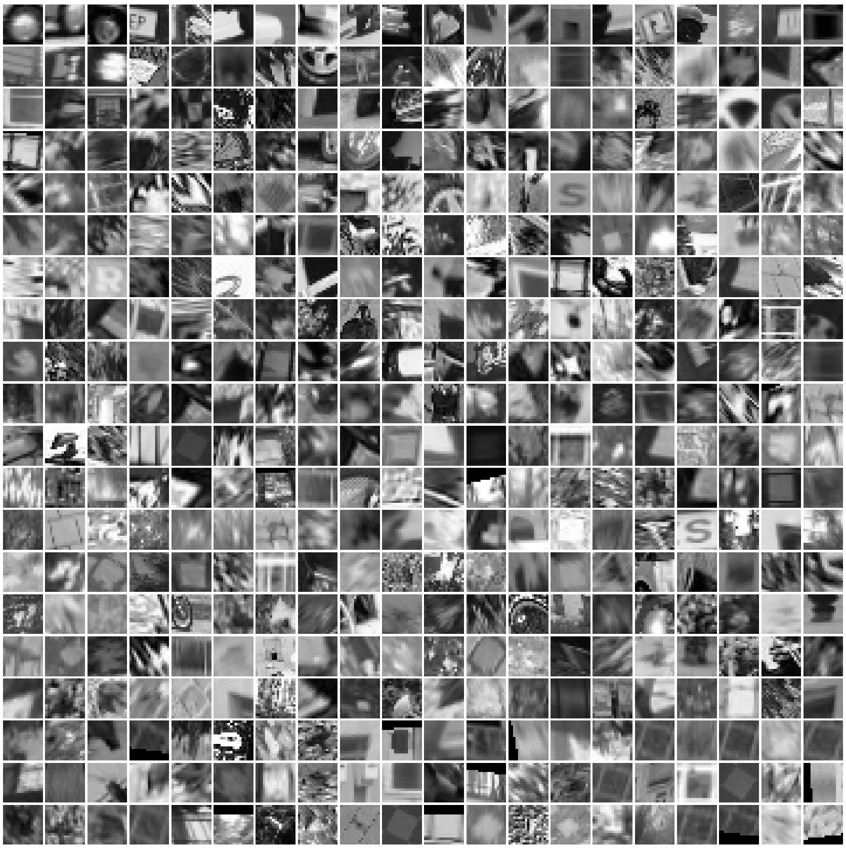


Fig. 4. Representative image for each cluster in an example clustering, with clusters sorted left-to-right, top-to-bottom by purity for cars.

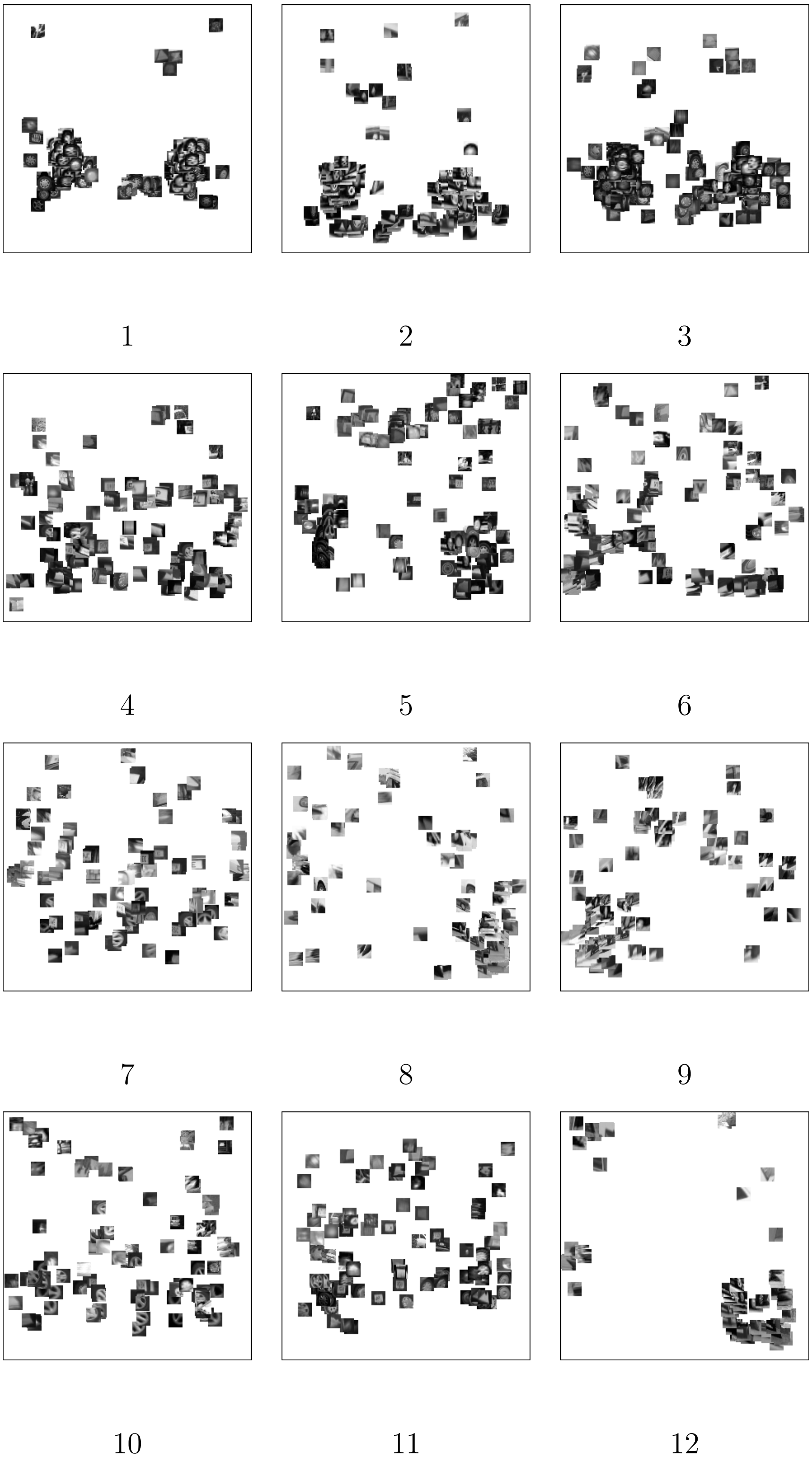
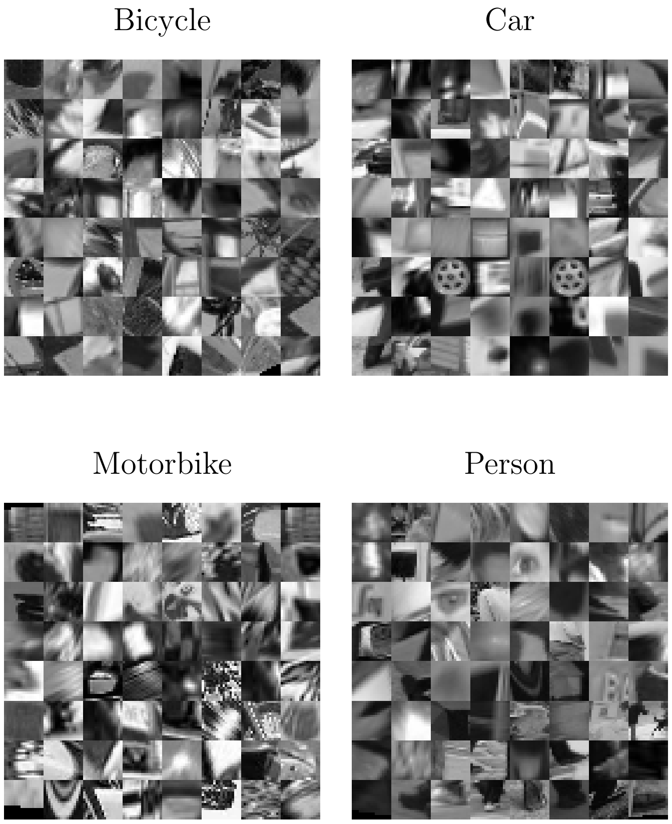


Fig. 5. Highest-purity car clusters for an example clustering, showing matching training set features, each represented by a scaled image of the local region displayed at the feature's location in a normalised object bounding box.



Bicycle

Car

Motorbike

Person

Fig. 6. Visual words most strongly associated with the component parts of example 600-cluster GTFs. Each GTF has an 8×8 grid of component parts.

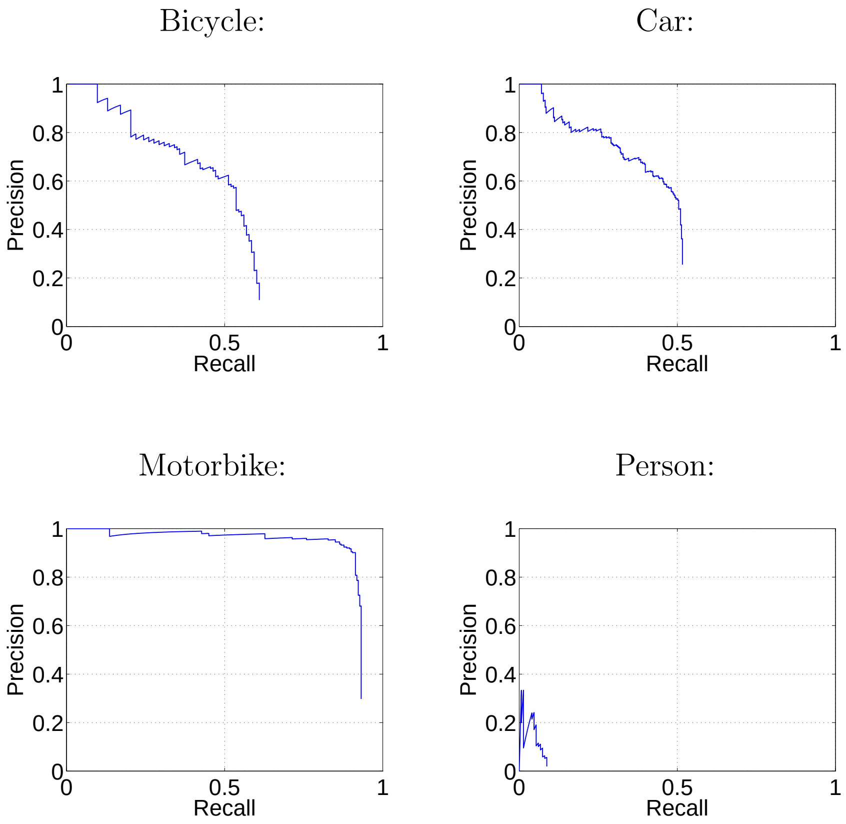
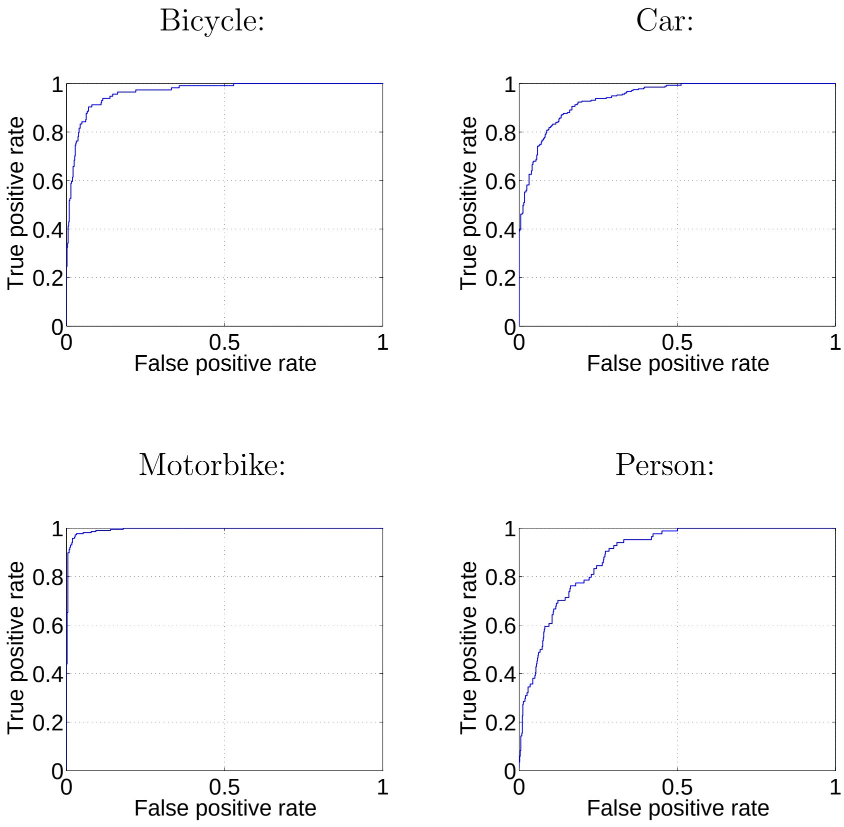


Fig. 7. Precision-recall curves for GTF as in Table 1.



Bicycle:

Car:

Motorbike:

Person:

Fig. 8. ROC curves for GTF as in Table 1.

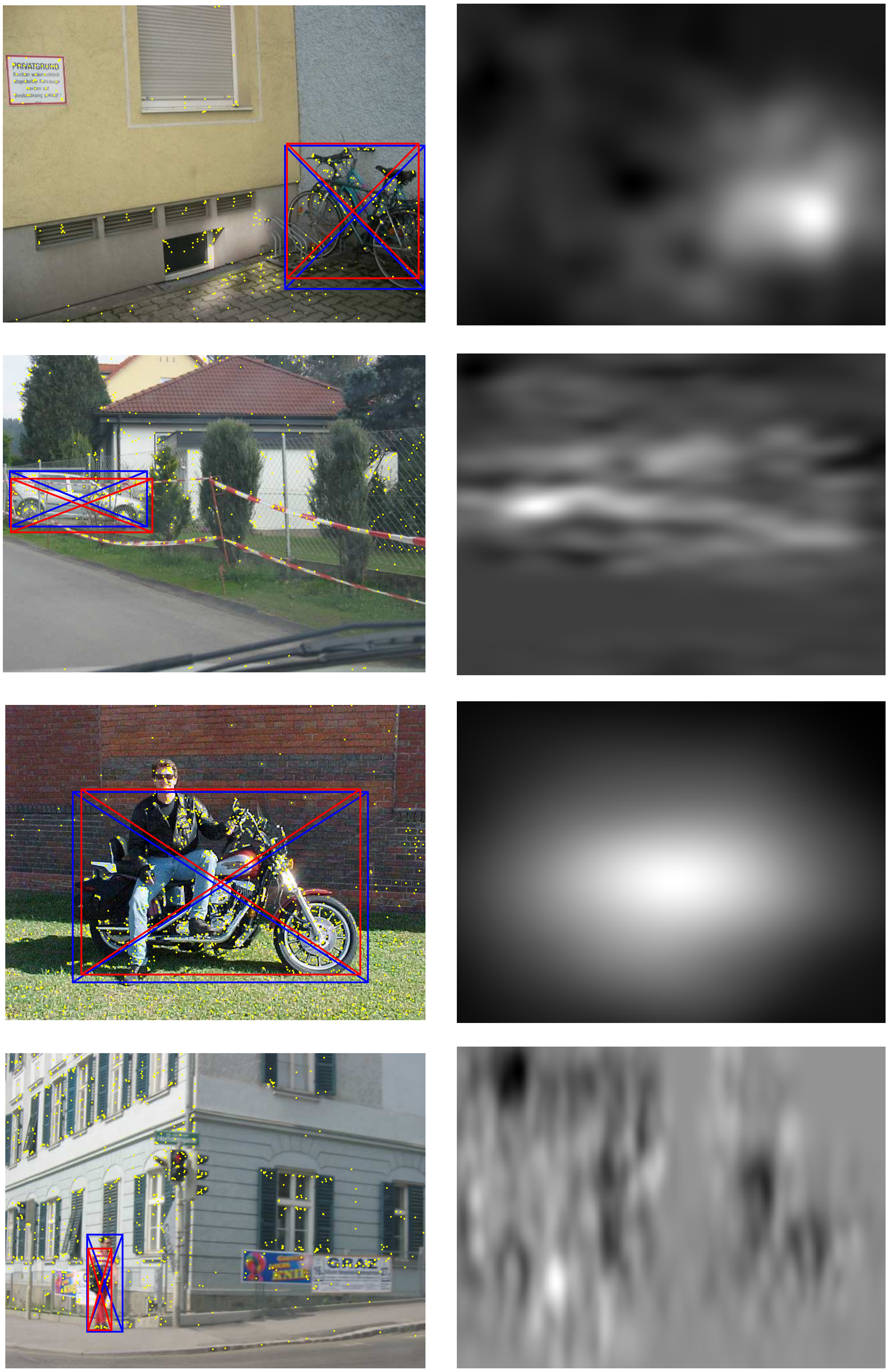


Fig. 9. Example probability surface for each class GTF as in Table 1. Top to bottom: bicycle, car, motorbike, person.

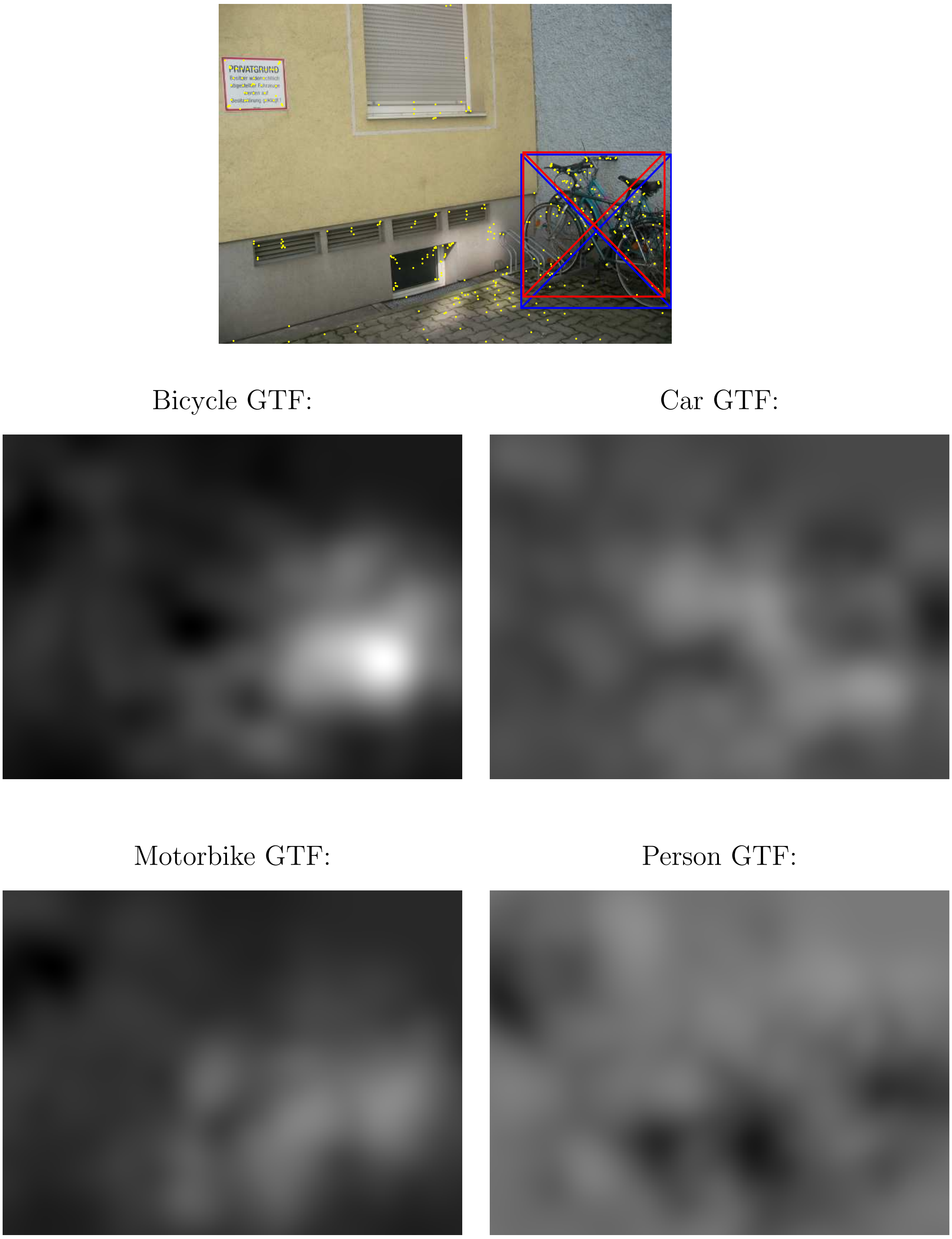


Fig. 10. Example probability surfaces for bicycle image from Figure 9, for each class GTF as in Table 1.

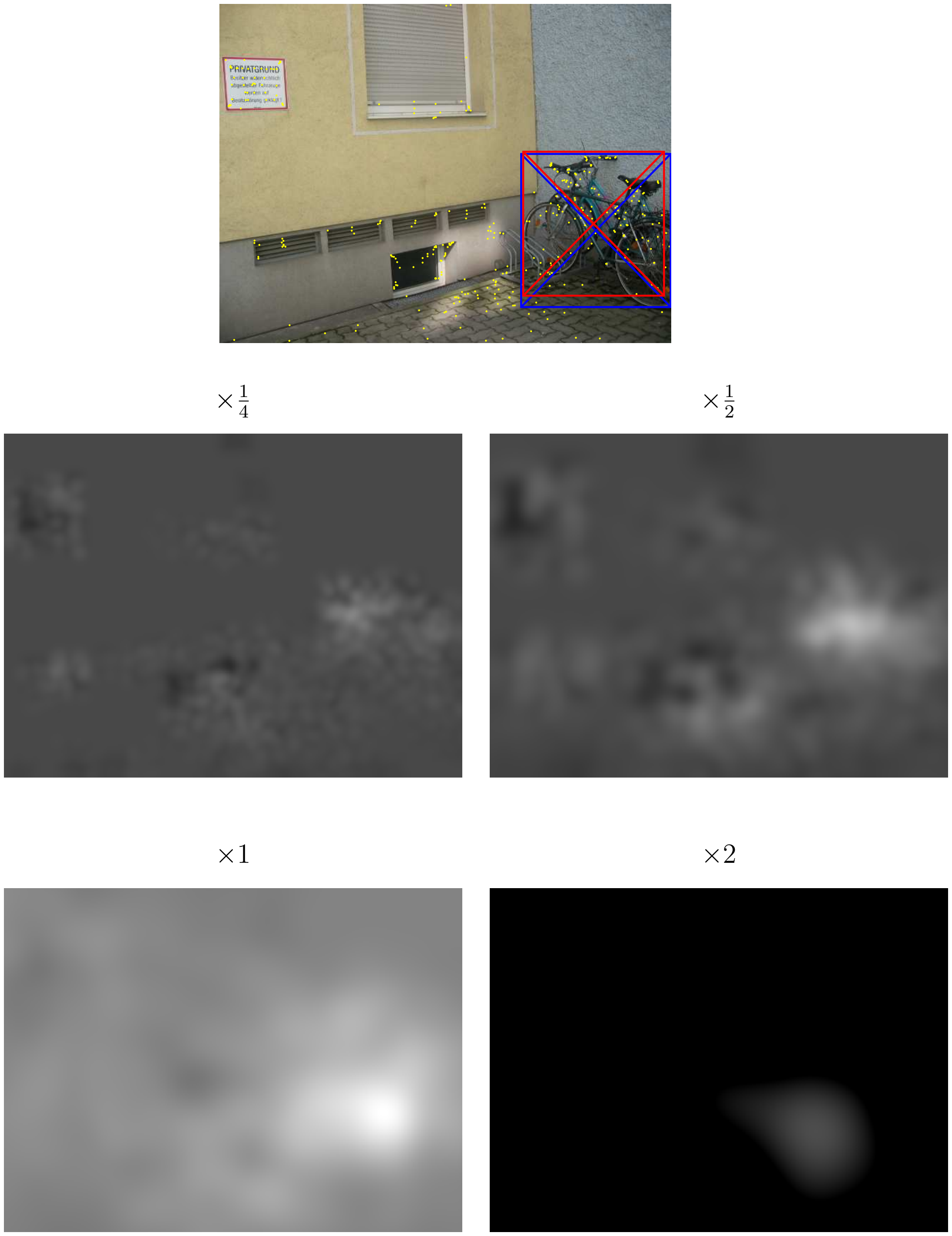


Fig. 11. Example probability surfaces at various hypothesis scales for bicycle image from Figure 9, for GTF as in Table 1.

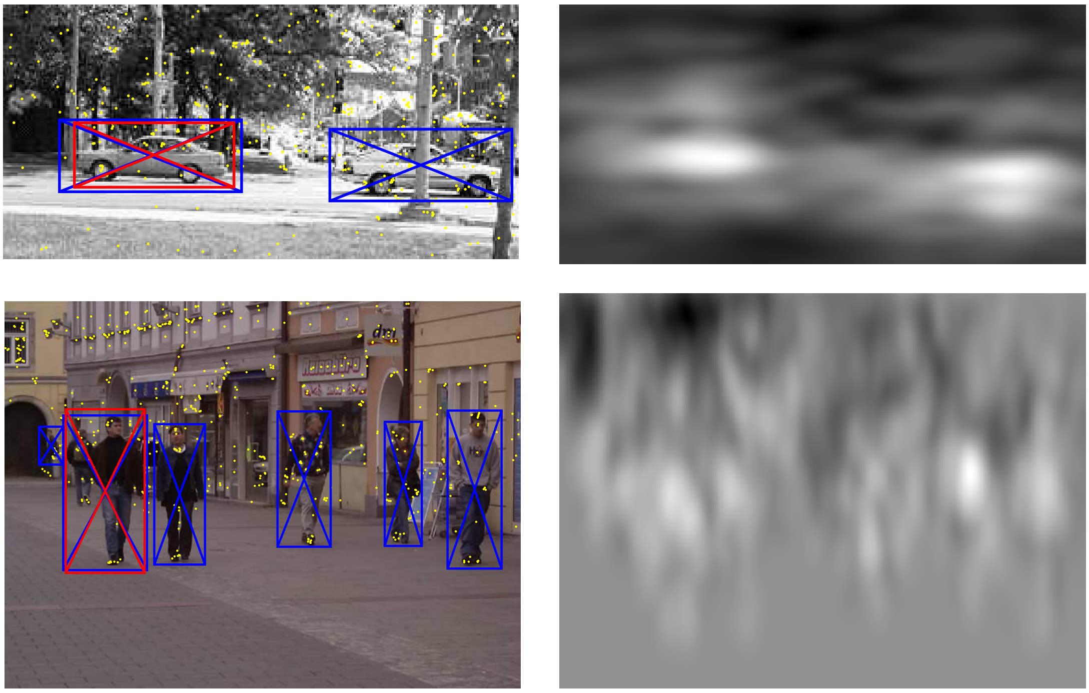
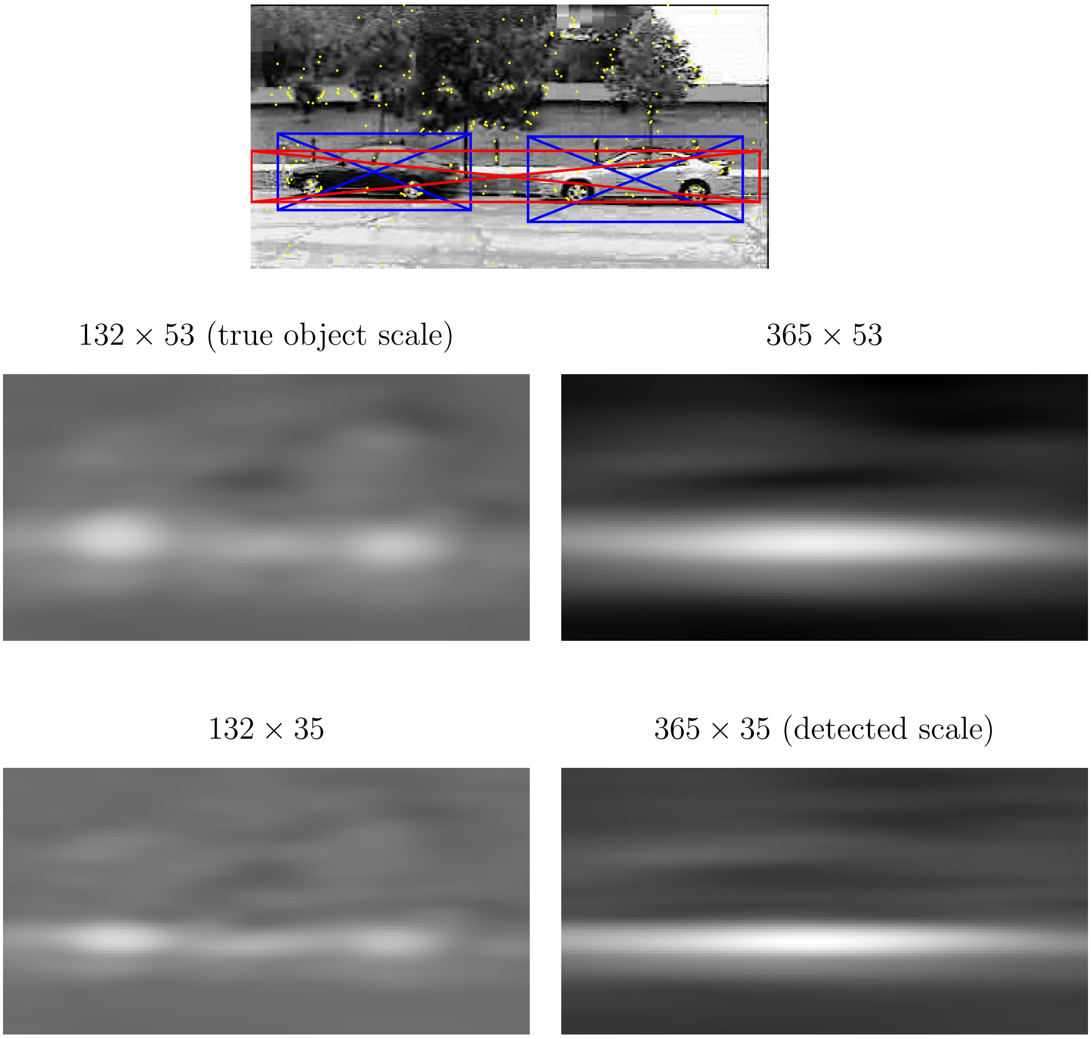


Fig. 12. Example probability surfaces for car and person GTFs as in Table 1.



132 × 53 (true object scale)

365 × 53

132 × 35

365 × 35 (detected scale)

Fig. 13. Example probability surfaces at various hypothesis scales for highest-probability car image with an incorrect detection, for GTF as in Table 1.