Combining Appearance Models and Markov Random Fields for Category Level Object Segmentation

Diane Larlus, Frédéric Jurie

Combining Appearance Models and Markov Random Fields for Category Level Object Segmentation

Diane Larlus
LEAR - INRIA, INPG
diane.larlus@inrialpes.fr

Frédéric Jurie
LEAR - INRIA, Caen University
frederic.jurie@unicaen.fr

Abstract

Object models based on bag-of-words representations can achieve state-of-the-art performance for image classification and object localization tasks. However, as they consider objects as loose collections of local patches they fail to accurately locate object boundaries and are not able to produce accurate object segmentation. On the other hand, Markov Random Field models used for image segmentation focus on object boundaries but can hardly use the global constraints necessary to deal with object categories whose appearance may vary significantly. In this paper we combine the advantages of both approaches. First, a mechanism based on local regions allows object detection using visual word occurrences and produces a rough image segmentation. Then, a MRF component gives clean boundaries and enforces label consistency, guided by local image cues (color, texture and edge cues) and by long-distance dependencies. Gibbs sampling is used to infer the model. The proposed method successfully segments object categories with highly varying appearances in the presence of cluttered backgrounds and large view point changes. We show that it outperforms published results on the Pascal VOC 2007 dataset.

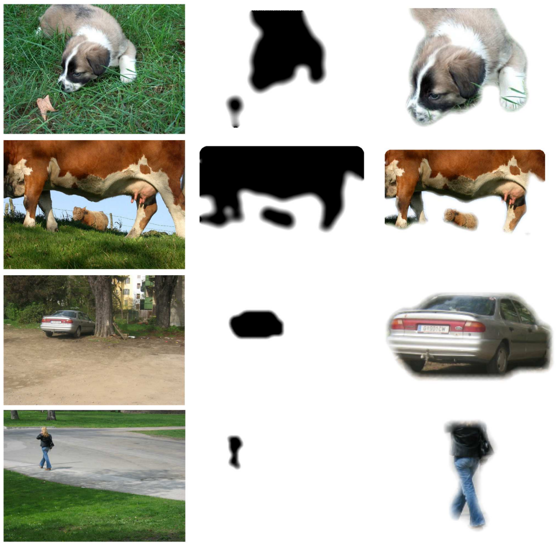

Figure 1. Instances of object categories are localized in images (col 1), producing masks (col 2) that can be used to automatically extract objects (col 3). The proposed method automatically produces these segmentation masks without any user interaction.

1. Introduction

This paper investigates the problem of producing accurate and clean segmentation of object classes in images, without giving any prior information on object identities, orientations, positions and scales.

Image segmentation has been addressed for several decades. Many different approaches have been investigated, trying to combine various image properties such as color, texture, edges, motion, etc., in an unsupervised way. However, segmentation using only bottom-up processes usually fail to capture high level information; image segmentation is indeed deeply related to image understanding.

The problem addressed here is the segmentation of objects belonging to known categories (also called *figure-ground segmentation*), assuming that the categories are defined by sets of training images used to learn object appearance models[1]. These training images play a fundamental role because object models build from these images allow to recognize and segment object.

Figure 1 gives an illustration of the problem we are addressing as well as results of our algorithm. Starting from cluttered images including objects of interest, the method is able to localize objects and to automatically produce seg-

[1] Please note the difference between *image* segmentation and *object* segmentation. Image segmentation corresponds to the situation where everything in the image have to be segmented whereas in object segmentation, only objects of interest are considered.

mentation masks that can be used to extract objects without any human interaction.

In the rest of the paper, we first present related work and an overview of the proposed method. Then we describe our model and its estimation. Finally we give experimental results and conclusions.

2. Related works

It has recently been shown [4] that models that consider images as loose sets of visual words, as well as being very efficient for image classification, can also be successfully applied to the localization of object class instances in images. These models' robustness to orientation and scale change allows them to cope with large variations in object appearance. This kind of model can also be combined with Dirichlet processes, to produce spatially localized clusters [15], [11]. Unfortunately, object shapes are very badly defined by blobs, so the localization provided is very rough. Cao *et al*. [2] tried to overcome this limitation by combining segmented regions and keypoints. Regions give a good initial segmentation of objects; however there is a difficult trade off concerning their size.

MRFs and their variants (CRF [14] [18], DRF[13]) have a long history in image segmentation. One of the major advantages of MRFs is *regularization*. Class labels (object or background) of two neighboring pixels are correlated, and when local evidence for a label is weak, labels from the neighborhood can provide a valuable help. Shotton *et al*. [14] propose using a CRF to learn a model of object classes for semantic image segmentation. Their model combines appearance, shape and context information. More recently, Win and Shotton [20] used an enhanced CRF based on a spatial ordering of object parts to handle occlusions and Verbeek and Triggs [17] proposed to combine a MRF and aspect models.

Simultaneously, remarkable object segmentation algorithms based on MRFs have been recently proposed by several authors (e.g. [12]), assuming that the object position is roughly provided by a user in the image to be segmented. The key idea is to model image foreground and background color distributions; these distributions are iteratively estimated by graph-cut. These interactive algorithms obtain very good results and the next step is now to get rid of user interactions. We would like to segment objects in a large amount of images *only* by specifying the object category (e.g. " segment the cows ").

However segmentations obtained by MRFs without *shape models* rarely produce accurate segmentation, thus several authors tried to merge these two concepts. One of the most noticeable work is the one of Kumar *et al*. [5] who propose a methodology for combining CRFs and pictorial structure models. Leibe and Schiele [6] use hand segmented images to learn segmentation masks corresponding to visual codebook entries; then the Implicit Shape Model allows to localize objects and segment images. Several other very interesting papers [1, 13, 19] or more recently [7] propose different ways to combine shape models with segmentation. However, the simple geometric assumptions made by models using shapes do not allow to deal with complex appearances of weakly structured object classes (see Figure 2).

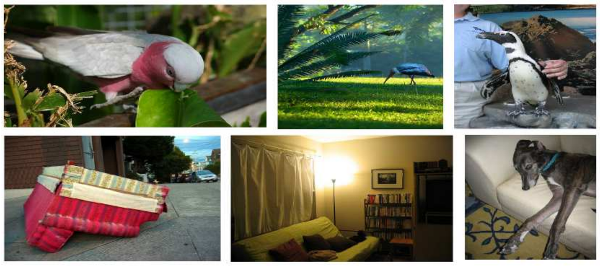

Figure 2. Images from the Pascal VOC 2007 dataset for two different categories: birds (first line) and sofa (second line).

At the end, only a very few of these methods can produce accurate segmentation of objects having wide appearance ranges. As can be seen on the last Pascal VOC challenge results (detailed section 4), there remains much room for improvement, especially when backgrounds are too rich and cluttered to be explicitly modeled.

Overview of our approach. The main contribution of this paper is a tractable model suitable for object segmentation, that takes advantage of two complementary components: (a) a model with *MRF* properties for its ability to produce fields of locally coherent labels and to produce segmentation fitting with low level image boundaries, (b) a *bag-of-words* based object model, allowing the recognition and the localization of objects despite strong view point variations, and ensuring long range consistency of visual information.

3. Model description

Each image is seen as a regular grid of patches. Patches are described by a generative model made of several regions (large sets of patches) which can take any label of interest (one of the object classes or background) and are estimated by the algorithm for each image. The segmentation consists in assigning patches to regions. The number of regions and their positions are unknown. At the same time, this grid of patches is seen as a field of labels where we apply regularization between neighboring patches.

3.1. Visual Features

Two different types of information are extracted from any image to be segmented: a set of n overlapping patches

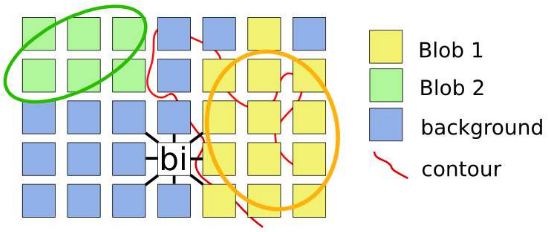

Figure 3. The model computes the best assignment of patches to object blobs and align cuts with natural boundaries of images.

and a gradient map.

Overlapping visual patches. Patches, denoted $\mathcal{P}_i, i \in \{1, \dots, n\}$, are square image regions. Four different characteristics are computed from each patch. First of all, a visual codebook is obtained by clustering SIFT [8] representations of the patches. Then, each patch $\mathcal{P}_i$ is associated to the closest codeword. The assigned codeword is denoted w_i^{sift}; this is the first characteristic. We also produce visual words based on color information by clustering color descriptors [16]. The patch $\mathcal{P}_i$ is also characterized by its closest color codebook word w_i^{color}. A RGB value is computed by averaging over pixels extracted in the center of the patch. This 3D-vector is denoted rgb_i. Finally we also consider the coordinates of the patch center $X_i = (x_i, y_i)$ in the image.

Gradient Map. In addition to computing patch-based characteristics, we also extract a gradient map $\mathcal{G}$, that gives the strength of the gradient at each pixel location (x, y). The map is computed by the algorithm of [9], that responds to characteristic changes in several *local cues* associated with natural boundaries.

Thus, the information carried by an image is entirely summarized by the gradient map $\mathcal{G}$ and the characteristics of the n overlapping patches $\mathcal{P}_i$, i.e. $\{w_i^{sift}, w_i^{color}, rgb_i, X_i, 1 \leq i \leq n\}$.

3.2. The blob-based generative model

This section specifies a generative model suitable for *rough* object/background segmentation. We use a model inspired by [15] with explicit spatial structure information: we consider that an image is made of regions of elliptic shape that we call *blobs*, and that each blob generates some patches with its own model. Intuitively, if an image contains three objects (a car, a pedestrian and a bike), we may have three blobs, one over each object region. Each blob is then responsible for generating the image pixels in its region, by generating a set of patches which appearance corresponds to the object category (car patches for the car blob, and so on). We also have a background region generating the remaining patches. This enforces the spatial coherence of the generated patches over the blob region.

Probability of the set of patches $\mathcal{P}$ given the Dirichlet model is obtained by multiplying probabilities of each patch $\mathcal{P}_i$ given the model. The generation of a patch requires to a) select a region (object blob or background) and b) generate a patch using the patch model *specific* to that region. The remaining of this section details the probabilities of selecting a region, and of generating a patch given a region.

The blob generation is assumed to follow a Dirichlet process. The Dirichlet process exhibits a self-reinforcing property; the more often a given value has been sampled in the past, the more likely it is to be sampled again. Dirichlet processes can be seen as the limit as K goes to infinity of a finite mixture model using K components[2] [10]. It means that for each new generated patch, it can either belong to an already generated image blob B_k, with probability $\frac{N_k}{n-1+\alpha}$ where N_k is its population, or either start a new region with a probability $\frac{\alpha}{n-1+\alpha}$, α being the concentration parameter of the Dirichlet process. These probabilities will be called p_{dir} in the next section.

We characterize each blob $B_k, 1 \leq k \leq K$, with a set of random variables: $\Theta_k = \{\mu_k, \Sigma_k, C_k, l_k, N_k\}$. μ_k, Σ_k are the mean and the covariance matrix describing the elliptic shape of the blob, l_k is the blob label (object category), C_k is a Gaussian mixture model representing the colors of the blob, N_k is the number of patches generated by the blob. The background is defined by a color distribution $C_b g$.

We characterize each patch $\mathcal{P}_i$ by its features $(w_i^{sift}, w_i^{color}, rgb_i, X_i)$ and also by two other random variables b_i and c_i. b_i is the index of the region (object blob or background) that generates the patch $(1 \leq b_i \leq K)$ and c_i is the color mixture component the patch is assigned to (detailed later).

Let us define the probability of generating a patch $\mathcal{P}_i$, given that it is generated by the region B_k of parameters Θ_k (which means that $b_i = k$). It is made of 4 distinct parts, as the model assumes that patch position, color and appearance are independent given the blob they belong to.

$$\begin{aligned} p(\mathcal{P}_i|\Theta_k) &= p(w_i^{sift}, w_i^{color}, rgb_i, X_i|\Theta_k) \\ &= p(w_i^{sift}|\Theta_k) p(w_i^{color}|\Theta_k) p(rgb_i|\Theta_k) p(X_i|\Theta_k) \end{aligned} \tag{1}$$

The position X_i of a patch follows a uniform distribution for the background and a normal distribution of parameters μ_k and Σ_k for object blobs : $p(X_i|\Theta_k, l_k) = \mathcal{N}(X_i, \mu_k, \Sigma_k)$.

We assume that object blobs and background have a color model made of a Gaussian Mixture (with 3 components in our experiments), as suggested by [12]. This allows to capture object-instance and image-background *specific color* appearances and build regions of coherent appearance even if some parts are less informative. This is

[2] K could go to the infinity while in practice the finite number of patches makes K finite and bounded

different from the color word w_i^{color} which encodes *class-specific* color appearance. For simplicity, we assume that each patch is generated by a unique GMM component, and this is encoded in the variable c_i introduced earlier.

Finally, the probabilities of the SIFT and color codewords only depend on the class label, i.e. $p(w_i^{sift}|\Theta_k)$=$p(w_i^{sift}|l_k)$ and $p(w_i^{color}|\Theta_k)$=$p(w_i^{color}|l_k)$. These distributions encode object appearance information and are responsible for the recognition ability of our model. This is the only information shared between images. They are learned via annotated training images from which visual words are extracted. The distributions are estimated by a counting process; indeed we count how often each visual word appears in each class and how often it appears in the background.

3.3. A MRF structured field of blob assignment.

The assignment of patches to object blobs or background $b = \{b_i, 1 \leq i \leq n\}$ determines the segmentation of the image. That segmentation is enhanced with our second component, the MRF of blob assignment, which regularizes the assignment of neighbor patches and also aligns cuts with natural image contrast. This field is defined on a grid (8-connectivity) that corresponds to patch centers. It is defined over labels which are the b_i values previously introduced *i.e.* the regions assignment.

In the previous section, the generative model fully defines the probability $p(\mathcal{P}|b, \Theta)$ of generating all patches given the assignment b and the blob parameters Θ. Then prior probability of the label field does not depend on the Θ parameters and is assumed to combine two independent models $p(b) \propto p_{dir}(b)p_{mrf}(b)$. The first part p_{dir} comes from the Dirichlet process description of the blobs distribution. and the second part p_{mrf} encodes neighbor dependencies imposed by the MRF.

Our model considers the joint probability of patch observations and blob assignment, which is decomposed in

$$\begin{array}{rcl} p(\mathcal{P}, b|\Theta) & \propto & p(\mathcal{P}|b,\Theta)p(b|\Theta) \\ & \propto & p(\mathcal{P}|b,\Theta)p_{dir}(b)p_{mrf}(b) \end{array} \quad (2)$$

where $\mathcal{P}$ represents observations associated to patches, b the label field and Θ all parameters related to the generative model.

This joint probability can be rewritten using an energy function E, $p(b, \mathcal{P}) \propto \exp(-E)$, which makes the formulation of the MRF easier:

$$E \;=\; U + \gamma \textstyle\sum_{i,j \in \mathcal{N}} V_{i,j} \quad (3)$$

where $\mathcal{N}$ represents couples of graph neighbors in the patch grid , γ is a constant parameter which weights the proportion of the two terms and

$$U = -\log(p(\mathcal{P}|b,\Theta)p_{dir}(b)) \quad (4)$$

The sum over $V_{i,j}$ represents the model p_{mrf}. It is defined as

$$V_{i,j} = [l_{b_i} \neq l_{b_j}] \exp(-\beta \Phi_{i,j}), \quad (5)$$

where [.] is the indicator function. $V_{i,j}$ is a potential that enforces local coherence of the object/background labels, via constraints on the similarity of neighbor patch labels, and also encourages cut along the image gradient via the function Φ. $\Phi_{i,j}$ is the maximum gradient $\mathcal{G}$ value between the position of the center of the patches $\mathcal{P}_i$ and $\mathcal{P}_j$, β a constant computed as in [12] (see Fig.3 for an illustration).

Thus, $V_{i,j}$ is null if patches have similar labels, else it particularly penalizes patches that have different labels and no boundary in between, as we want to allow our model to separate objects and background mainly at image boundaries.

From this energy-based formulation we can go back to the exponential form and derive the posterior probability for the blob assignment labels, that we will need later for the model estimation. To do so, we consider only the observation $\mathcal{P}_i$ matching site b_i and the cliques of the graph containing b_i.

$$p(b_i|b_{-i}, \Theta, \mathcal{P}) \sim p(\mathcal{P}_i|\Theta_{b_i}) \frac{N_{b_i}}{n{-}1{+}\alpha} \exp(-\gamma \sum_{i,j \in \mathcal{N}} V_{i,j}) \quad (6)$$

where b_{-i} denotes $b \setminus \{b_i\}$.

3.4. Model Estimation

The model being defined by the blob component and the MRF structure, its parameters have to be estimated for each image to produce object blobs labels $\{l_i, 1 \leq i \leq K\}$ and patches assignments to blobs $\{b_i, 1 \leq i \leq N\}$. A Gibbs sampler generates an instance of parameter values from the distribution of each variable in turn, conditional on the current values of the other variables. This section defines the conditional distributions on each variable and the way to sample them. The set of parameters to be estimated is $\{\mu_{1:K}, \Sigma_{1:K}, C_{1:K}, l_{1:K}, b_{1:n}, c_{1:n}\}$.

Sampling blob parameters. In the following, observations from $\{P_i, 1 \leq i \leq n\}$ belonging the blob B_k are renamed as $P'_i = (w_i'^{sift}, w_i'^{color}, rgb'_i, X'_i), 1 \leq i \leq N_k$.

The two first blob parameters are μ_k and Σ_k. If W_2 denotes a Wishart distribution,

$$\begin{array}{rcl} \mu_k & \sim & \mathcal{N}(\mu, Mean(X'_{1:N_K}), \frac{1}{N_k} Cov(X'_{1:N_K})) \\ \Sigma_k & \sim & W_2(Cov(X'_{1:N_K}), N_k - 1) \end{array} \quad (7)$$

The third blob parameter is the color mixture of Gaussians, which is simply estimated by a stochastic EM, with each mixture made of nc (3 in our experiments) components. We also estimate the color mixture for the background.

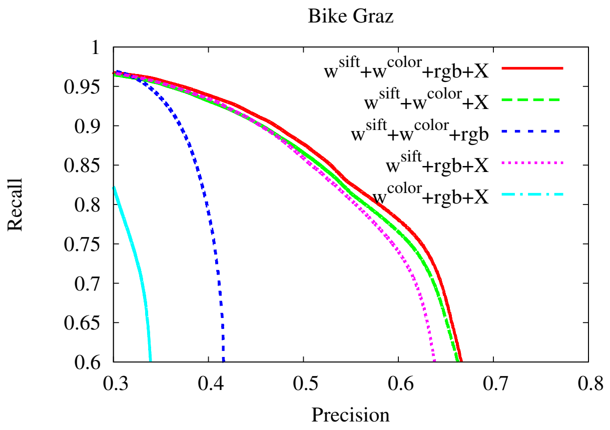

Figure 4. The relative importance of the different features : SIFT (w^{sift}) and color codebooks (w^{color}), color components (rgb) and positions (X) differently combined. Note that the MRF component comes with the X variable.

The last blob parameter is the class label l_k, sampled from:

$$l_k \quad \sim \quad \prod_{i=1}^{N_k} p(w_i'^{sift}|l_k)p(w_i'^{color}|l_k) \qquad (8)$$

due to the assumption of patch independence given the blob that has generated them.

Sampling patch parameters. c_i is the color mixture component affected to a patch. It is computed by sampling from the associated region color Gaussian mixture.
Last but not least, the conditional estimation of the blob membership variables b_i. The probability of generating b_i conditionally on the other variables and the observations $p(b_i|b_{-i}, \Theta, \{b_i\}, \mathcal{P})$ was introduced equation (6).

Training data. The probabilities $p(w_k^{sift}|l_k)$ and $p(w_k^{color}|l_k)$ are essential to the object model. We directly learned them from training images. If segmentation masks are available they can be used advantageously. We will see in the experiments that only having the bounding boxes is enough to obtain accurate segmentation.

3.5. From labeled patches to pixels

Our model provides a probability of blob assignment for each patch, and a probability of class label (one of the object class) for each blob. From those, we can compute the class label probability for a patch. The probability for pixel p to belong to an object or to the background is computed by accumulating the knowledge about all patches containing this pixel. This is modeled by a mixture model where weights are functions of the distance between the pixel and the center of the patch. Segmentation masks are obtained by assigning the most probable class to each pixel.

4. Experiments

We consider mainly three challenging datasets for object/background segmentation: TU Graz-02[3], Pascal VOC 2006 and Pascal VOC 2007 [3]. They contain object classes with highly varying appearance together with a generic and cluttered background. Furthermore, the objects present scale and illumination variations, viewpoint changes and occlusions. The TU Graz-02 images contain three object categories : bicycles, cars, persons. Available ground-truth for segmentation makes this database interesting to evaluate the performances of a segmentation method and to study parameters. The Pascal VOC 2006 images include strongly varying views of 10 different categories: bicycles, buses, cats, cars, cows, dogs, horses, motorbikes, people and sheep. The Pascal VOC 2007 possesses 10 additional classes: birds, boats, bottles, chairs, planes, potted plants, sofa, tables, trains and tv/monitors. Segmenting images containing simultaneously many objects of such a large number of categories with the same algorithm and the same parameter settings is a particularly difficult task. Figure 2 gives a good illustration of these difficulties by showing several representative images for 2 different categories (birds, sofa).

This section covers three different aspects of the experiments that validate our method. First we make a quantitative study on Graz-02 dataset emphasizing the importance of each feature that appears in the model. Second, we show qualitative results (i.e. segmentation masks) obtained on the Pascal VOC-2006 dataset. Third, we evaluate our method on the Pascal VOC-2007 dataset, and demonstrate that we largely outperform the best competitive methods on this benchmark dataset. We also present additional experiments on a related problem using the Microsoft dataset.

4.1. Parametric study

Several features are computed from local patches: SIFT codebook indexes w^{sift}, color codebook indexes w^{color}, RGB colors rgb and positions X (see section 3.1 for details). This section evaluates the relative importance of these features in the segmentation results. We compared the full model (denoted $w^{sift} + w^{color} + rgb + X$), with different subsets of these features. We consider the Graz-02 dataset, because it comes with a ground truth.

We use half of the 300 Graz02 images for learning, and the remaining for testing. Visual vocabularies of 5000 elements are created for the SIFT [8] descriptors, and 100 elements for the color [16] descriptors. They are obtained by quantizing descriptors extracted from the training images.

Graz02 images contain only one object category per image so the segmentation task can be seen as a binary classification problem. Thus the accuracy is measured by precision

[3] http://www.emt.tugraz.at/ pinz/data/

recall curves (see Fig. 4) that show how many pixels from the object categories (all images merged) are correctly classified. The different features w^{sift}, w^{color}, rgb and X can be present or not in the models. We observe that the two visual vocabularies w^{sift}, w^{color} are essential. If one of them is missing the performance decreases much, but texture (i.e. w^{sift}) is more critical than color. The MRF, by regularizing the segmentation, improves the results very much, as the comparison of the red (all features) and blue (without X) curves shows. That regularization has strong visual effects on the precision of the segmentation. And last, the color component rgb gives an improvement for two categories out of three. However, when objects are not localized correctly, the color component deteriorates the results.

4.2. Qualitative results

In this section, we propose a visual inspection of the segmentation masks computed on Graz02, MSRC and Pascal VOC-2006 database. For each class, images are segmented into object of interest and background regions. For the Graz (bike, car and person) and MSRC, the object model is trained using provided segmentation masks. On the Pascal dataset (the other categories), object models are trained with bounding boxes only, and not with pixel level segmentation masks, as the ground truth of the pascal challenge only provides bounding boxes. This is why we cannot provide quantitative results on this dataset.

Typical segmentation results are shown Figure 1 and 4. Our algorithm automatically detects and segments objects accurately despite large intra-class variations and scale and orientation changes, even in the case of weak supervision (training with bounding boxes only).

4.3. Quantitative results

Due to its popularity we compared our method with results recently publish on the MSRC2 dataset[4]. The task is significantly different because the background is divided into several classes (grass, building, trees ..) so the goal is not really to do figure/ground segmentation but to segment images. Table 1 gives the performance of our algorithm on the 13 object classes of the dataset. We compared with Textonboost results [14] and with Markov Field Aspect Models (MFAM) [17]. Our method gives comparable results, although it is not designed explicitly for this kind of task.

	Cow	Sheep	Aeroplane	Face	Car	Bicycle	Sign	Bird	Chair	Cat	Dog	Body	Boat
Textonboost	58	50	60	74	63	**75**	35	19	15	**54**	19	**62**	7
MFAM	73	**84**	**88**	70	**68**	74	33	19	**34**	46	49	54	**31**
Our Method	**84**	81	66	**78**	50	62	**36**	**22**	16	43	**52**	30	9

Table 1. Results on the MSRC2 dataset

Pascal challenge VOC 2007 The Pascal VOC 2007 challenge [3] is an international benchmark that involves tens among the best computer vision groups. Thus we use this dataset to compare the performances of our object category segmentation algorithm to state-of-the-art algorithms. The competition consists in generating pixel-wise segmentation giving the class of the object visible at each pixel, or ”background” otherwise, which is exactly the task we are tackling with our approach. The dataset is made of 20 object classes and one background class. The dataset includes more than 5000 images for training, 422 of them are precisely annotated with segmentation masks. For the other images, only the bounding boxes are given.

We evaluate our accuracy with the Pascal VOC 2007 protocol. We compute the average segmentation accuracy across the twenty classes and the background class. The segmentation accuracy for a class is the number of correctly labeled pixels of that class, divided by the total number of pixels of that class in the ground truth labeling [3].

For training category appearance models, we use all the annotations (both the segmentation masks and the bounding boxes). Then the model is estimated for each image using the detector INRIA_PlusClass [3] to initialize the blob positions and labels.

The results obtained on the 20 classes are presented in Table 2. We also report in this table the best results obtained during the competition. Our average performance is almost 10% higher than this best known performance.

5. Conclusions

We have presented a novel framework for object category image segmentation. The key element that distinguished this method from existing approaches is the combination, within the same model, of two complementary components. First, a blob-based component detects objects using occurrences of visual words. It produces an approximate segmentation, roughly splitting the different components of the image. Second, a MRF-based component produces clean cuts, guided by image intensity, contour and texture edges. A Gibbs sampling algorithm allows the efficient estimation of the parameters of the model.

We have shown that our model achieves very accurate segmentation masks on the Pascal VOC 2006 and Graz02 datasets. On most classes our method improves on the best scores obtained on the benchmark dataset Pascal VOC 2007.

Acknowledgments The authors thank Eric Nowak for his help and Hedi Harzallah for providing the detection results.

[4] available at http://research.microsoft.com/vision/cambridge/recognition

	backgrd	plane	bicycle	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motorbike	person	plant	sheep	sofa	train	monitor	**mean**
TKK	22.9	18.8	20.7	5.2	16.1	3.1	1.2	78.3	1.1	2.5	0.8	23.4	69.4	44.4	42.1	0	64.7	30.2	34.6	89.3	70.6	**30.4**
Our Method	41.0	20.2	72.3	25.3	17.0	27.8	23.1	66.6	77.8	31.1	11.1	0.8	3.6	67.6	53.7	66.9	34.6	23.9	33.6	65.9	73.8	**39.9**

Table 2. Results on the pascal VOC 2007 dataset. First row gives the best results known on this dataset (details can be found in [3]). The second row is the result we obtain.

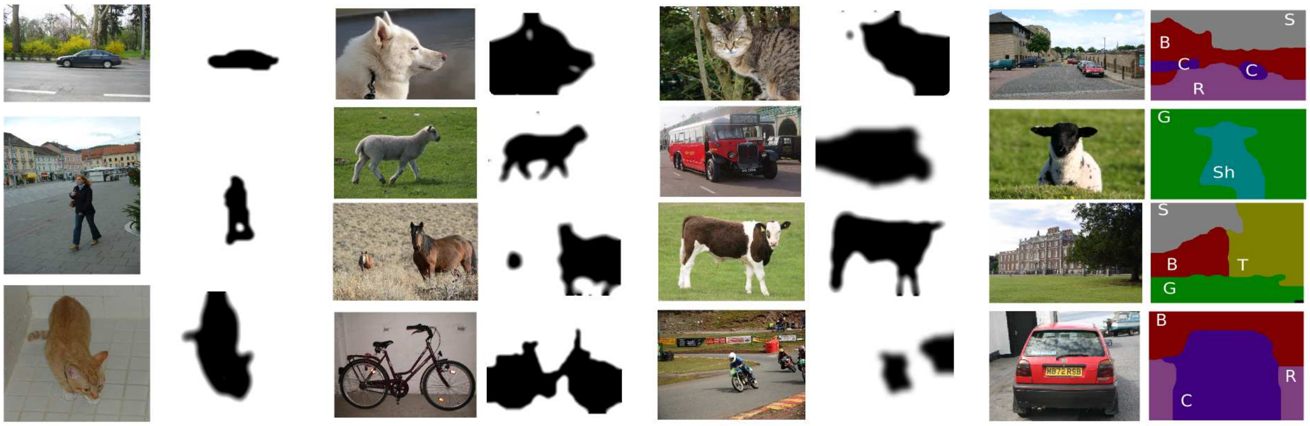

Figure 5. Examples of segmentation obtained by our method for the Graz-02, Pascal VOC 2006 and Microsoft (best viewed in color) datasets. For the latest the following coding is used: G for grass, Sh for sheep, S for sky, B for building, T for tree, C for car.

References

[1] E. Borenstein and J. Malik. Shape guided object segmentation. In *CVPR*, pages 969–976, 2006.

[2] L. Cao and L. Fei-Fei. Spatially coherent latent topic model for concurrent object segmentation and classification. In *ICCV*, 2007.

[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007. http://www.pascal-network.org/challenges/VOC/voc2007/workshop.

[4] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning object categories from google"s image search. In *ICCV*, 2005.

[5] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. In *CVPR*, 2005.

[6] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, 2003.

[7] Z. Lin, L. S. Davis, D. Doermann, and D. DeMenthon. Hierarchical part-template matching for human detection and segmentation. In *ICCV*, 2007.

[8] D. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2), 2004.

[9] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE PAMI*, 26(5):530–549, 2004.

[10] R. M. Neal. Markov chain sampling methods for dirichlet process mixture models. Technical Report 9815, Dept. of Statistics, University of Toronto, Sep 1998.

[11] P. Orbanz and J. M. Buhmann. Smooth image segmentation by nonparametric bayesian inference. In *ECCV*, 2006.

[12] C. Rother, V. Kolmogorov, and A. Blake. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[13] J. Shotton, A. Blake, and R. Cipolla. Contour-based learning for object detection. In *ICCV*, pages I:503–510, 2005.

[14] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, pages I: 1–15, 2006.

[15] E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Describing visual scenes using transformed dirichlet processes. In *NIPS*, 2005.

[16] J. van de Weijer and C. Cordelia Schmid. Coloring local feature extraction. In *ECCV*, pages 334–348, 2006.

[17] J. Verbeek and B. Triggs. Region classification with markov field aspect models. In *CVPR*, 2007.

[18] J. Verbeek and B. Triggs. Scene segmentation with crfs learned from partially labeled images. In *NIPS*, 2008.

[19] J. Winn and N. Jojic. Locus: Learning object classes with unsupervised segmentation. In *ICCV*, 2005.

[20] J. Winn and J. Shotton. The layout consistent random field for recognizing and segmenting partially occluded objects. In *CVPR*, pages 37–44, 2006.