Multi-class image segmentation using Conditional Random Fields and Global Classification

Nils Plath NILSP@CS.TU-BERLIN.DE
Marc Toussaint MTOUSSAI@CS.TU-BERLIN.DE
TU Berlin, Machine Learning Group, Franklinstrasse 28/29, D-10587 Berlin, Germany

Shinichi Nakajima NAKAJIMA.S@NIKON.CO.JP
Nikon Corporation, Optical Research Laboratory, 1-6-3 Nishi-Ohi, Shinagawa-ku, Tokyo 140-8601, Japan

Abstract

A key aspect of semantic image segmentation is to integrate local and global features for the prediction of local segment labels. We present an approach to multi-class segmentation which combines two methods for this integration: a Conditional Random Field (CRF) which couples to local image features and an image classification method which considers global features. The CRF follows the approach of Reynolds & Murphy (2007) and is based on an unsupervised multi scale pre-segmentation of the image into patches, where patch labels correspond to the random variables of the CRF. The output of the classifier is used to constraint this CRF. We demonstrate and compare the approach on a standard semantic segmentation data set.

1. Introduction

Visual scene understanding, e.g., in a robotics context, means to identify and locate separate objects in a scene. Such a segmentation is a crucial precursor for any object-related behavior, like reaching for an object, moving towards an object or telling a user where an object is located. Even without depth information (disparity, motion) humans are extremely good in segmenting still images whereas this is still a great challenge for Machine Learning (ML) methods (Everingham et al., 2008).

An interesting issue in this context is that most state-of-the-art ML methods for image classification – that is, the task of predicting whether an object from a certain class is contained in the image – rely completely on global image features (Zhang et al., 2007), since at a local level the discriminating appearance of image patches will deteriorate (e.g., wheels can be parts of various kinds of vehicles.) Hence, typical image classification methods do not attempt to locate the object or use information from a special focal area. They only predict the presence of an object, in some sense without knowing where it is. This fact suggests two implications. First, the object context seems equally relevant for the prediction as the object area itself. Second, although image segmentation is the problem of assigning a local object label to every local pixel, we may expect that a purely local mapping will not perform well because the work on classification demonstrates the importance of the context.

In this paper we propose a method for multi-class image segmentation which comprises two aspects for coupling local and global evidences. First, we formulate a Conditional Random Field (Lafferty et al., 2001) that couples local segmentation labels in a scale hierarchy. This idea follows largely previous work by (Reynolds & Murphy, 2007), which we extend to the multi-class case and the use of SVMs to provide the local patch evidences. Second, we use global image classification information to decide prior to the segmentation process which segment labels are considered possible in a given new image. Experiments show that without this use of global classification, the segmentation performs poorly. However, with a hypothetical optimal classification the segmentation outperforms the best state-of-the-art segmentation algorithms, while it is among the top third using our own image classifier.

Previous work on segmentation includes for example (Awasthi et al., 2007). Here, the authors also construct a tree structured CRF from image patches. In

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009.

contrast to our method, though, they split the image into patches laying on a regular grid and connecting them using a quad-tree structure. Also, they do not have a image classification step before segmenting an image. Another approach (Csurka & Perronnin, 2008) transforms low-level patch features into high-level features based on Fisher kernels and defines a score system for patches to determine whether a patch is part of fore- or background. Their framework also includes an image classification step, however, no CRF are used (though it could be extended to use CRF). Finally, in (Reynolds & Murphy, 2007) segmentation is facilitated by constructing a tree-structured conditional random field on image regions on multiple scales. Their method is able to segment objects and divide image into fore- and background. In its original form it is only applicable to two-class scenarios, foreground and background. We extend this to the multi-class case and introduce an image classification step to improve segmentation results.

Our segmentation algorithm follows a number of steps which we briefly summarize here:

1. We first use an unsupervised segmentation to fragment the whole image in a number of patches on multiple scale levels.

2. For each patch on each level we compute color, texture and SIFT features.

3. From all patch features we also compute a global image feature vector.

4. We use trained SVMs to predict a class membership for each patch and for the whole image.

5. Depending on the image structure and the global classification we build a CRF model – a dependency tree between patch labels – and use the SVMs outputs as local evidences in this .

6. Thresholding the posterior labels for the finest patches produces the final segmentation.

The next section describes the unsupervised image segmentation and the feature computation. Section 3 will explain the CRF model, how we train it and how we use the global classification. Section 4 demonstrates the approach on the challenging PASCAL VOC 2008 image segmentation data set (Everingham et al., 2008).

2. Pre-segmentation and Features

2.1. Unsupervised pre-segmentation into image patches

The aim of unsupervised image segmentation is to segment the image into *patches* of high textural and color homogeneity. This kind of segmentation is a priori independent from any given labels or object classes, but due to the textural and color homogeneity, the resulting patches are likely to belong to only one object and not overlap with different objects. Felzenszwalb and Huttenlocher (2004) proposes an algorithm based on a graph cut minimization formulation: Every pixel of an image is considered a node of a graph. Starting with an initially unconnected graph, a greedy algorithm decides on whether to connect nodes/pixels (and thereby subgraphs/patches) depending on their similarity and depending on evidence that there actually is a visual boundary between those patches.

The algorithm provided by the authors has three parameters which allow us to tune the typical size and scale of the resulting image patches: a smoothing factor σ, a similarity threshold k, and a minimum patch size m. We can exploit this in our algorithm by computing pre-segmentations of an image on L different scale levels, from fine to coarse patches. Figure 1 displays an example. Formally, we represent the output of such a multi-level pre-segmentation as an integer label $p_i^l \in \{1, .., P\}$ of the ith image pixel on level l. The integer label in $1, .., P$ enumerates all patches found on all levels, i.e., P varies from image to image. In our experiments we use $L = 3$.

The unsupervised pre-segmentation of an image into patches plays an important role in our algorithm. For each patch on each level we will estimate its object class label using local patch features, as described in the next section.

2.2. From Image to Patch Features

Image classification typically means to predict whether a certain object is contained in the image. Current state-of-the-art methods for image classification (Zhang et al., 2007; Nowak et al., 2006) put much emphasis on defining good global image features which can then be mapped, using large-scale statistical Machine Learning methods such as SVMs, to the desired classification. Our approach tries to draw as much as possible on these existing methods. Any global image classification method can usually be translated to a local patch classification method by computing the features only on the patch. The features we use are based on local color, texture, and SIFT descriptors,

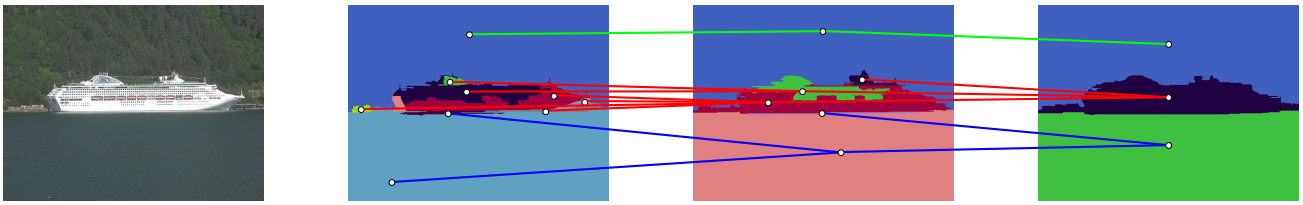

Figure 1. Images are segmented with a greedy algorithm at multiple scales. Each patch q at level l is connected to the one node p in the next coarser level $l+1$ which has the largest overlap with itself. In this example three tree are constructed for this image.

which we explain in the following.

The color features are computed from the HSV color space model of the input image. As in (Reynolds & Murphy, 2007) we compute an HSV histogram over an image patch where hue and saturation are binned together into a 10×10-bin histogram and value is binned separately into a 10-bin histogram. For each patch $p \in \{1, .., P\}$ this gives a feature vector $f_p^{\text{color}} \in \mathbb{R}^{110}$.

The texture features are computed using Gabor filters. Also as in (Reynolds & Murphy, 2007) we applied these filters to the images with $d = 6$ directions and at $s = 4$ scales. The resulting energies are binned on a per patch-basis in a 10-bin histogram for each direction and scale, which gives a feature vector $f_p^{\text{texture}} \in \mathbb{R}^{240}$.

Finally, for standard image classification bag-of-words features based on SIFT descriptors have been found critical for high performances. We first compute a standard SIFT discriptor at regular grid points over the whole image. We choose a grid width of 6 pixels so that the local SIFT circular regions, with a radius of 16 pixels, overlap. We assume that during training we have created a visual codebook of SIFT descriptors, i.e., we have clustered the SIFT data to yield 1000 visual prototypes using regular K-Means. Hence, at each grid point we can associate the computed SIFT discriptor with an integer $i \in \{1, .., 1000\}$ indicating the prototype it belongs to. To construct the patch feature, we consider all the grind points in a given patch and compute the prototype histogram. Since we choose a small enough grid size and sufficiently high minimum patch size k we never observed that a patch without any grid points in practice. For each patch p this gives a feature vector $f_p^{\text{SIFT}} \in \mathbb{R}^{1000}$.

The full patch feature $f_p = (f_p^{\text{color}}, f_p^{\text{texture}}, f_p^{\text{SIFT}})$ is the concatenation of the described features.

As mentioned in the introduction, the local classification inherent in image segmentation can profit from the global classification of the whole image because the latter is a compact representation of the context. We will make use of global image classification and hence also need to define global image features. As before we define a regular grid over the images, here with a 10 pixel width, and compute SIFT descriptors over multiple scales, i.e. the radii are changed on each layer (4, 6, 8, and 16 pixels). Then visual prototypes are constructed with K-Means clustering. For computational reasons we randomly selected 10 images from each object class and computed 1200 cluster centers followed by a quantization of the image features. Additionally we computed a 2-level spatial pyramid representation of the images, which has been shown to perform well in (Lazebnik et al., 2006). The final image-level feature vector is the concatenation of all histograms over all regions in the pyramid and over all scales. The training of the image classifier makes use of a SVM with a χ^2 kernel. The details of training are the same as for the segmentation, as will be explained later.

3. Conditional Random Fields

The patch features allow us to learn a patch classifier that would predict the object class membership of each patch independently. However, given the spatial relation between patches – and in particular the relation (overlap) between patches on the different scale levels – the patch labels should not be considered independent. A standard framework for the prediction of dependent labels of multiple output variables is structured output regression (Lafferty et al., 2001; Altun et al., 2003; Tsochantaridis et al., 2005). Generally, a discriminative function $F(y, x)$ over all output variables $y = (y_1, .., y_P)$ is defined depending on the input x such that the prediction for all output variables is given as $f(x) = \text{argmin}_y F(y, x)$. The factorization of $F(y, x)$ w.r.t. y typically corresponds to a graphical model for which the argmin can efficiently be computed using inference methods.

Let us first focus on only one object class, $N = 1$;

we will discuss the general multi-class segmentation below. In this case we need to segment the image into foreground (label 1) and background (label 0). We will predict these labels not on the level of pixels but on the level of patches. Hence, in our case the output variables $y = (y_1, .., y_P)$ are the binary labels $y_p \in \{0, 1\}$ we want to predict for each patch.

We assume a tree as the coupling structure between these variables. Let $\pi(p)$ denote the set of children (descending nodes) of a patch p in this tree, then the discriminative function is of the form

$$F(y, x) = \prod_{p=1}^{P} \phi(y_p, x) \prod_{q \in \pi(p)} \psi(y_p, y_q) \ , \qquad (1)$$

where $\phi(y_p, x)$ represents input dependent evidences for each patch, and $\psi(y_p, y_q)$ represents a pair-wise coupling between the labels of a parent and child patch. We first explain how we construct the tree for a specific image before giving details on these potentials.

The tree (actually forest) is fully defined by specifying the parent of each node. A node corresponds to a patch from one of the multiple scale levels. Every patch connects to the one parent patch in the next coarser level which has the largest overlap with itself: if q is a patch of level l, then its parent p in level $l+1$ is

$$p = \underset{p}{\operatorname{argmax}} \frac{|I_p \cap I_q|}{|I_q|} \ , \qquad (2)$$

where $I_p = \{i \in I : p_i^{l+1} = p\}$ is the image region that corresponds to p. As a result, every patch in the coarsest level does not have a parent and is a root of a tree. Figure 1 illustrates such a tree.

The potentials $\phi(y_p, x)$ represent the local evidence for labelling the patch p as forground depending on the input. We will assume that only the local patch features f_p are used, $\phi(y_p, x) = \phi(y_p, f_p)$.

The potentials $\psi(y_p, y_q)$ describe the pairwise coupling between overlapping patched from layers $l+1$ and l, respectively. Reynolds and Murphy (2007) made such edge potentials dependent on the visual similarity between the patches. Although this seems a reasonable heuristic, we found that this local similarity measure is too brittle to yield robust results. We decided to use a set of parameters γ_l to tune the coupling between pairs of levels separately. This has the advantage that the coupling can be increased for layers for which the patch-level classifiers exhibit a high amount of certainty. For parent p and child q in layers $l+1$ and l we define the pairwise potentials as the following 2×2-matrix

$$\psi(y_p, y_q) \quad = \quad \begin{pmatrix} e^{\gamma_l} & e^{-\gamma_l} \\ e^{-\gamma_l} & e^{\gamma_l} \end{pmatrix} \qquad (3)$$

Inference in the tree (or forest) is done using standard exact inference (we use an implementation of the Junction Tree Algorithm).

Concerning the multi-class case we tried some alternatives. A straight-forward extension to the binary tree is to extend the patchwise labels to be an integer $y_p \in \{0, 1, .., N\}$ where N is the number of object classes. The local evidences $\phi(y_p, f_p)$ can be represented by $N+1$ separate models and the pairwise couplings $\psi(y_p, y_q)$ become a $(N+1)\times(N+1)$-matrix with diagonal entries e^{γ_l} and constant off-diagonal entries $e^{-\gamma_l}$. This approach seems to work fine if in every image we have always all N object classes present. However, in realistic data, some images contain only one object class out of $N = 20$, other images three or more. Generally, the uncertainties in object classes that are not present in an image cause, particularly for large N, a significant perturbation of the segmentation of the present object class. We decided to aim for a more robust approach where the segmentation within an object class should be independent of N and uncertainties in other object classes. We settled for having N separate binary trees and compute the posterior label distribution for each tree. Since we compute patches on multiple scale levels, we will use the predicted labels of the smallest scale patches as the predicted segmentation. The final label we assign to a patch p is the object class for which the posterior is maximal if this maximum is larger than a threshold, and background otherwise. The threshold is determined using cross-validation.

3.1. Training

The conditional random field framework we introduced above would naturally lead to a structured output regression method for training the ϕ's and ψ's. However, we simplified the training problem by decomposing the training of local evidences $\phi(x_p, f_p)$ from the inference mechanisms on top of that (Reynolds & Murphy, 2007).

We assume we are given a dataset of segmented images, i.e., images for which each pixel i is labeled with an integer $S_i \in \{0, .., N\}$ indicating either that the pixel belongs to one of N object classes or to the background ($S_i = 0$). An example is depicted in figure 2.

We can apply the multi-scale pre-segmentation on each image of this dataset and compute the patch features

for all patches. When more than 75% of the pixels of a patch are labelled with the same object class then we consider this patch as positive training data for the classifier for this object class, otherwise it is considered negative training data (Reynolds & Murphy, 2007). That is, for each patch $p = 1, .., P$ and each object class $n = 1, .., N$ we have a label $y_{pn} \in \{+1, -1\}$ indicating whether p has more than 75% overlab with n. We train N separate SVMs on all patches using these labels y_{pn}.

An appropriate kernel for histogram features is a χ^2-kernel, which was shown to be a suitable similarity measure histograms features (Hayman et al., 2004; Zhang et al., 2007). The parameter b determines the width of the kernel, which is set to the mean of the χ^2 distances over all pairs of training samples (Lampert & Blaschko, 2008).

The regularization parameter c was tuned using cross-validation. Since the data is highly biased towards negative examples, it is a good idea to duplicate positive training examples so that the number of training samples in both groups is approximately the same. We used the Shogun SVM implementation (Sonnenburg et al., 1999), which balances the data set automatically.

Applying a re-normalization using logistic regression (Platt, 1999) allows us to treat the output of the classifiers as probabilities

$$\phi(y_p = 1|f_p) \quad = \quad \frac{1}{1 + e^{-(a_l \mathrm{svm}(f_p) + b_l)}} \qquad (4)$$

where l is the scale level of the patch p and the parameters a_l and b_l control the slope and the bias.

3.2. Coupling global image classification with local segmentation

What we found one of the most important keys to good image segmentation is that global image information is taken into account. As discussed in the introduction, the patch classifiers we described in the previous section rely only on local image information. However, as is evident from the success of image classification techniques that neglect completely the location of an object and use the whole image, the image context is an essential ingredient for good classification. This is also true for the segmentation task.

Based on the global image feature vector f_I we train N separate image classifiers (SVMs) to predict whether a certain object class is present in a new image. Only when a class is predicted we build the binary random tree and consider this class label in the segmentation procedure described in the previous section.

4. Experiments

In this section we report results on the data set provided by the Pascal VOC2008 challenge (Everingham et al., 2008). The data set consists of three predefined subsets for training, validation, and testing. The groundtruths, as shown in figure 2, are given only for the training and the validation set. However, the organizers of the challenge offer to evaluate the results submitted to them.

The images, for which the challenge defines 20 object categories, are randomly downloaded samples from the famous Flickr photo portal and the groundtruths were produced in meticulous hand-work. On average the training and validation set comprise around 40 color images per class with varying sizes. No official statistics about the test data are published by the organizers of the challenge.

Performance is measured by the average segmentation accuracy across all object classes and the background. The accuracy a is assessed using the intersection/union metric

$$a \quad = \quad \frac{\mathrm{tp}}{\mathrm{tp} + \mathrm{fp} + \mathrm{fn}} \qquad (5)$$

where tp, fp, and fn are true positives, false positives, and false negative, respectively.

Our classification algorithm reached an average precision of 42% on the test data.

The segmentation framework has been set up according to the parameters in table 1. The initial stage of patch generation involved careful hand-tuning as to satisfy the balance between the number of patches created per level and general quality of the patches in terms of how they overlap with the actual objects in the images. Settings for color and texture features remain the same as in (Reynolds & Murphy, 2007). The configuration for the SIFT components is a trade-off between computational efficiency and discriminative features. For the creation of the visual codebook we limited the number of descriptors used and randomly sampled $10,000$ vectors per object class and applied K-Means clustering.

The entire process chain is shown in figure 4. The sample images are pre-segmented as described above, then for each patch p in each of the three layers l the confidence is computed. In figure 4 (b)-(d) the confidences for each patch are shown as gray values, where black corresponds to a confidence of zero and white to one. The resulting posteriors at the leaf level are illustrated in figure 4 (e). The last column of this figure depicts the final segmentation after thresholding the

Table 1. Settings used in experiments. Settings for the classifiers are not shown, since they are determined automatically and may vary depending on the data source.We tested 3 different segmentation thresholds t, for which all other settings are identical. For t_{ind} we computed one threshold for each object class individually, t_{glo} uses mean of t_{ind}, and $t_{\max}$ is set to the maximum of t_{ind}.

step	parameters	values
patches	σ_l	$\{0.5, 0.75, 1.25\}$
	k_l	$\{500, 500, 700\}$
	m_l	$\{50, 200, 1200\}$
color feat.	bins_{HS}	10×10
	bins_{S}	10
texture feat.	$\text{bins}_{\text{Gabor}}$	10
	$\text{directions}_{\text{Gabor}}$	6
	$\text{scales}_{\text{Gabor}}$	4
SIFT features	grid size	6×6
	visual prototypes	1000
trees	a_l	$\{2, 1.5, 1.5\}$
	b_l	$\{0, 0, 0\}$
	γ_l	$\{2, 1\}$
BP	t_{ind}	n thresholds
	t_{glo}	mean of t_{ind}
	$t_{\max}$	$\max(t_{\text{ind}})$

posteriors.

Table 2 shows detailed results of our segmentation results and compares them to the competitors of this year's challenge. In average our results settle in the middle. However, it is important to keep in mind that our algorithm can only be as good as the prior image classification. In a different setting we assumed to have a perfect image classifier. In this case the segmentation significantly outperforms the best reported algorithms. Surprisingly applying individual thresholds (tind) does not perform as well as a globally applied threshold (tglo), as more false positives will be reported. The third configuration (tmax) uses the maximal value of the individual threshold. This setting reduces false positives but at the same time increases the number of false negatives.

Figure 3 depicts examples of successful and unsuccessful segmentations. A correct segmentation becomes very probable if the object classes in the image have been determined correctly. In the case where the image classification contains a lot of false positives the results deteriorate. One reason is that the additional patch classifiers may be very confident that certain patches resemble their training data, e.g. a cow could be mistaken for any other animal with similar fur color and texture or the sky could be labelled as aeroplane since it is in a contextual relation to that class.

Figure 2. Randomly selected images of the VOC2008 challenge. Rows with odd indices depict the provided image, rows with even indices are the according hand-made segmentation groundtruths.

5. Conclusions

We presented an extension of the work done in (Reynolds & Murphy, 2007) from simple figure-ground segmentation to multi-class segmentation. Our framework consists of a simple image classifier to detect the presence of categories of objects in an image and applies a semantical segmentation according to the class information. The quality of our segmentation relies strongly on the performance of the various patch-level and image-level classifiers. Thus, using more elaborate features may lead to better results. These could include shape features (Bosch et al., 2007; Belongie et al., 2002). Changing the method of the object categorization step to state-of-the-art classification algorithms, such as (Zhang et al., 2007; Nowak et al., 2006), would bring us closer to the segmentation performance we observed assuming the correct class prediction.

6. Acknowledgements

This work was supported by the German Research Foundation (DFG), Emmy Noether fellowship TO 409/1-3.

References

Altun, Y., Tsochantaridis, I., & Hofmann, T. (2003). Hidden markov support vector machines. *Proceedings of the 20th International Conference on Machine Learning (ICML 2003)* (pp. 3–10).

Awasthi, P., Gagrani, A., & Ravindran, B. (2007). Image modelling using tree structured conditional random fields. *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI 2007)* (pp. 2060–2065).

Belongie, S., Malik, J., & Puzicha, J. (2002). Shape

Figure 3. (a) Examples of successful segmentation. (b) Unsuccessful segmentations.

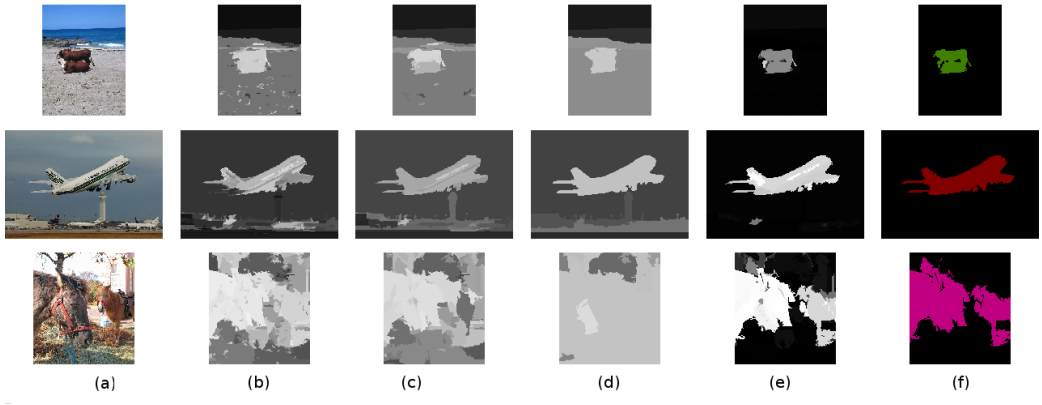

Figure 4. Demonstration of spatial smoothing with Conditional Random Fields (CRF) and segmentation (a) Sample input images. (b)-(d) Confidences of local classifiers for patches p for the layers $l = 1 \ldots 3$, i.e. from fine to coarse layers. Only the output of the classifier for which the posteriors after belief propagation are the highest are shown. (e) Posteriors on leaf layer (finest) after belief propagation (f) Applying a global threshold on the posteriors and assigning each segment with the maximum marginal probability yields the final semantic segmentation.

matching and object recognition using shape context. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*, 509–522.

Bosch, A., Zisserman, A., & Munoz, X. (2007). Representing shape with a spatial pyramid kernel. *Proceedings of the 6th ACM International Conference on Image and Video Retrieval (CIVR 2007)* (pp. 401–408).

Csurka, G., & Perronnin, F. (2008). A simple high performance approach to semantic segmentation. *Proceedings of the British Machine Vision Conference (BMVC 2008)*.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2008). The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results. http://www.pascal-network.org/challenges/VOC/voc2008/workshop/.

Felzenszwalb, P., & Huttenlocher, D. (2004). Efficient graph-based image segmentation. *International Journal of Computer Vision*, *59*, 167–181.

Hayman, E., Caputo, B., Fritz, M., & Eklundh, J. (2004). On the significance of real-world conditions for material classification. *Proceedings of the European Conference on Computer Vision (ECCV 2004)* (pp. 253–266).

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *Proceedings*

Table 2. Segmentation results for Pascal VOC2008 challenge. The results for the method described in this paper are the two rows on the bottom. (tglo) indicates that a global threshold for image classification has been used, (tind) indicates the use of one separate threshold per object class, for (tmax) we choose the maximal value of (tind) as global threshold. To demonstrate the capacities of our segmentation algorithm (true class), we tested it on the validation set using the classification information provided with the data. With this additional information our method outperforms all other reported methods.

	mean	background	aeroplane	bicycle	bird	boat	bottle	bus	car	cat	chair
BrookesMSRC	20.1	75.0	36.9	4.8	22.2	11.2	13.7	13.8	20.4	10.0	8.7
Jena	8.0	47.8	7.2	3.1	4.6	5.6	2.2	0.6	13.4	0.0	0.7
MPI_norank	7.0	66.3	6.7	1.2	2.1	3.1	2.5	5.8	2.6	2.9	1.1
MPI_single	12.9	75.4	19.1	7.7	6.1	9.4	3.8	11.0	12.1	5.6	0.7
XRCE_Seg	25.4	75.9	25.8	15.7	19.2	21.6	17.2	27.3	25.5	24.2	7.9
our method (tind)	12.7	58.1	12.5	4.1	5.2	5.8	6.1	11.9	9.7	6.8	2.2
our method (tglo)	13.1	62.1	9.9	4.7	4.9	11.0	6.1	17.0	9.9	7.8	1.1
our method (tmax)	11.2	75.0	21.1	0.0	2.6	8.9	11.3	5.8	9.1	3.0	5.3
our method (true classes)	35.6	73.8	48.2	20.9	40.9	28.6	21.8	46.3	32.7	39.4	9.6

	cow	dining table	dog	horse	motorbike	person	potted plant	sheep	sofa	train	tv/monitor
BrookesMSRC	3.6	28.3	6.6	17.1	22.6	30.6	13.5	26.8	12.1	20.1	24.8
Jena	7.5	0.7	5.7	4.4	8.9	8.7	5.0	9.2	3.4	12.2	17.8
MPI_norank	1.7	4.0	2.6	3.7	5.1	10.5	0.8	5.8	2.1	8.4	8.0
MPI_single	3.7	15.9	3.6	12.2	16.1	15.9	0.6	19.7	5.9	14.7	12.5
XRCE_Seg	25.4	9.9	17.8	23.3	34.0	28.8	23.2	32.1	14.9	25.9	37.3
our method (tind)	6.8	15.6	8.1	10.1	26.6	18.9	6.2	10.5	2.3	12.8	26.9
our method (tglo)	11.2	11.2	7.4	10.5	23.2	16.5	5.3	12.1	3.4	17.6	21.8
our method (tmax)	0.0	0.0	7.1	7.4	18.3	18.5	7.2	0.1	1.8	3.3	30.1
our method (true classes)	49.6	41.5	21.0	30.3	46.4	35.3	19.6	30.8	22.4	57.2	30.3

of the 18th International Conference on Machine Learning (ICML 2001) (pp. 282–289).

Lampert, C., & Blaschko, M. (2008). A multiple kernel learning approach to joint multi-class object detection. *Pattern Recognition: Proceedings of the 30th DAGM Symposium (DAGM 2008)* (pp. 31–40).

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR 2006)* (pp. 2169–2178).

Nowak, E., Jurie, F., & Triggs, B. (2006). Sampling strategies for bag-of-features image classification. *Proceedings of the European Conference on Computer Vision (ECCV 2006)* (pp. 490–503).

Platt, J. (1999). Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. advances in large margin classifiers. *Advances in Large Margin Classifiers*, *1*, 61–74.

Reynolds, J., & Murphy, K. (2007). Figure-ground segmentation using a hierarchical conditional random field. *Proceedings of the Fourth Canadian Conference on Computer and Robot Vision (CVR 2007)* (pp. 175–182).

Sonnenburg, S., Raetsch, G., Schaefer, C., & Schoelkopf, B. (1999). Large scale multiple kernel learning. *Journal of Machine Learning Research*, *7*, 1531–1565.

Tsochantaridis, I., Joachims, T., T.Hofmann, & Y.Altun (2005). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, *6*, 1453–1484.

Zhang, J., Marszalek, M., Lazebnik, S., & Schmid, C. (2007). Local features and kernels for classification of texture and object categories: A comprehensive study. *International Journal of Computer Vision*, *73*, 213–238.