Int J Comput Vis (2012) 98:243–262
DOI 10.1007/s11263-011-0507-2

Object Recognition by Sequential Figure-Ground Ranking

João Carreira · Fuxin Li · Cristian Sminchisescu

Received: 19 February 2011 / Accepted: 8 November 2011 / Published online: 19 November 2011

Abstract We present an approach to visual object-class segmentation and recognition based on a pipeline that combines multiple figure-ground hypotheses with large object spatial support, generated by bottom-up computational processes that do not exploit knowledge of specific categories, and sequential categorization based on continuous estimates of the spatial overlap between the image segment hypotheses and each putative class. We differ from existing approaches not only in our seemingly unreasonable assumption that good *object-level segments* can be obtained in a feed-forward fashion, but also in formulating recognition as a regression problem. Instead of focusing on a one-vs.-all winning margin that may not preserve the ordering of segment qualities inside the non-maximum (non-winning) set, our learning method produces a *globally consistent* ranking with close ties to segment quality, hence to the extent entire object or part hypotheses are likely to spatially overlap the ground truth. We demonstrate results beyond the current state of the art for image classification, object detection and semantic segmentation, in a number of challenging datasets including Caltech-101, ETHZ-Shape as well as PASCAL VOC 2009 and 2010.

Keywords Object recognition · Semantic segmentation · Learning and ranking

The first two authors contributed equally.

J. Carreira · F. Li · C. Sminchisescu (✉)
University of Bonn, INS, Wegelerstrasse 6, Bonn 53115, Germany
e-mail: cristian.sminchisescu@ins.uni-bonn.de

1 Introduction

Recognizing and localizing different categories of objects in images is essential for scene understanding. Approaches to object-category recognition based on sliding windows have recently been demonstrated convincingly in difficult benchmarks (Viola and Jones 2001; Felzenszwalb et al. 2010; Vedaldi et al. 2009). By scanning the image at multiple locations and scales, recognition is phrased as a binary decision problem for which many powerful classifiers exist. Recent developments have shown that scanning hundreds of thousands of windows efficiently can be feasible for certain types of features and classifiers (Vedaldi et al. 2009; Blaschko and Lampert 2008). The bounding box approach to recognition has proven successful for object categories with stable features that can 'fill' the correct window significantly, like faces or motorbikes, it nevertheless tends to be unsatisfactory for objects with more complex appearance and geometry, or for advanced tasks such as pose prediction and action recognition where the knowledge of an object's shape is also important.

This motivates the focus on *semantic segmentation*, where the objective is to both identify the spatial support of objects, and to recognize their category. In semantic segmentation, the brute-force sliding windows approach to generic category recognition may not be feasible. Consider Fig. 1(a). A reliable object detector might locate the person and place a bounding box around her. However, the non-canonical pose may impose a large bounding box, or alternatively a large search space if different rotations of the bounding box are scanned, still leaving a non-trivial contour hypothesis space to be explored, even inside the correct bounding box, *e.g.* Fig. 1(b).

The semantic segmentation problem could be approached top-down (Borenstein and Ullman 2002; Leibe et al. 2008),

by storing exemplars to guide the search in new images. However, since the variability of object shapes is large, only an approximate contour alignment between the training exemplars and new object instances can be expected. Interesting solutions have been proposed recently, although generalization to a large class of shapes remains non-trivial (Kumar et al. 2005; Levin and Weiss 2009). In fact, some of the best performing methods for semantic segmentation currently do not employ shape priors but directly classify individual pixels, based on statistics of patches enclosing them (Shotton et al. 2006; Csurka and Perronnin 2008; Ladicky et al. 2009a).

An open problem for segmentation and recognition is the design of tractable models capable to make more informed decisions using increased spatial support. It appears necessary to be able to work at some intermediate spatial scale, ideally on segments that can model entire objects, or at least sufficiently distinct parts of them. The idea of doing recognition on segments larger than just piecewise uniform regions (superpixels) is not new, but has been barred for a long time by the lack of progress in reliably obtaining such segments.

Fig. 1 (**a**) A girl relaxing on a bench. Both top-down approaches and bottom-up sliding window methods can encounter difficulties segmenting or detecting a person in this non-canonical pose. (**b**) Semantic segmentation results produced by our algorithm

However, recent developments in segmentation algorithms provide a surprisingly effective solution (Carreira and Sminchisescu 2010b). For most images, the Constrained Parametric Min Cuts (CPMC) algorithm can generate a set of 20–200 figure-ground hypotheses, among which segments covering full objects are extracted with high probability (see Fig. 2). This motivates our exploration of visual recognition directly from a pool of holistic segment hypotheses extracted bottom-up. Recognition proceeds similarly with sliding windows methods, but in the drastically reduced search-space of plausible object segments. This enables the use of more powerful learning machinery based on multiple features and nonlinear kernels, trained with a large number of segments with different degree of overlap with the target object.

Besides leveraging recent progress in figure-ground segmentation methods for recognition, we contribute with a formulation that casts recognition as a one-against-all regression problem of predicting the quality of segments. The quality of a segment for a given category is measured as the maximum amount of overlap between the segment and a ground truth object of that category. Therefore, the correct category can be simultaneously determined from the predicted qualities for each of the multiple classes. This makes it possible to use all information available in those segments that only partially overlap with the ground truth and, we show, gives a significant boost in the recognition performance. We further develop a sequential recognition strategy that can identify multiple spatial supports and analyze images containing several objects from different categories.

The paper is organized as follows. Section 2 reviews related work. Section 3 describes the overall framework. Sections 4–6 describe the three main components of the framework: segment ranking (Sect. 4), segment scoring and categorization (Sect. 5), and sequential segment post-processing

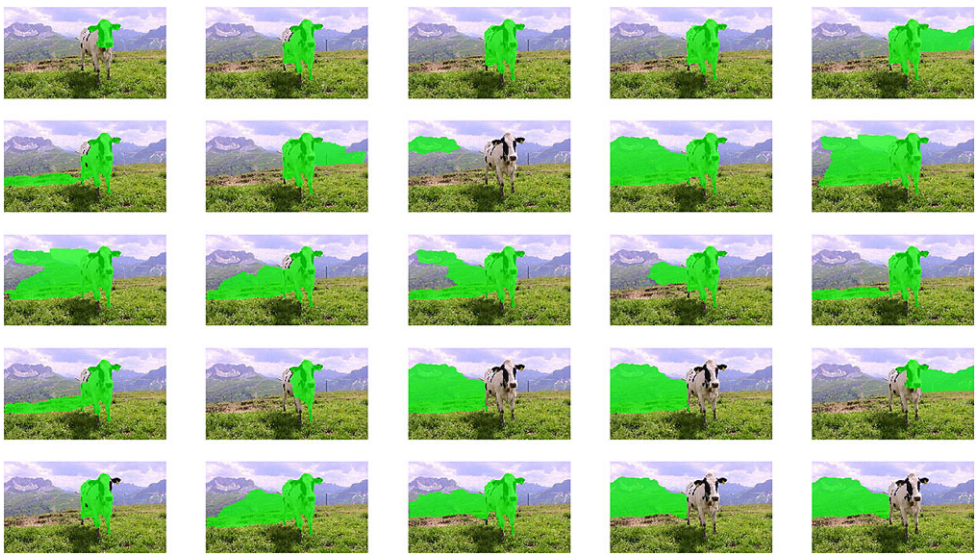

Fig. 2 Examples of segments used in the recognition process. Clearly, among the multiple figure-ground hypotheses generated by CPMC (Carreira and Sminchisescu 2010b) there are good segments that cover the object of interest entirely. The challenge for recognition is to pull them out

(Sect. 6). In Sect. 7, we test the various components of the system and report state-of-the-art results on three object recognition tasks: image classification, object detection and semantic segmentation. Section 8 concludes the paper and discusses ideas for future work.

2 Related Work

We will confine our review of the state of the art to recognition techniques that estimate the spatial layout of objects. These techniques can be broadly classified as bottom-up or data-driven and top-down or model-based, although the separation is to some extent blurred as many methods have both bottom-up and top-down components.

Bottom-up Recognition. Bottom-up recognition techniques use no prior shape knowledge to obtain the object regions. They often either categorize among a set of predefined region hypotheses, like our method, or directly classify pixels.

Rabinovich et al. (2007) use a stability heuristic (Rabinovich et al. 2006) to select a reduced list of segmentations obtained using normalized cuts (Shi and Malik 2000) for different number of segments and different cue combinations. Segments are described by bags of features and those with the highest label confidence given by a k-nearest neighbor classifier are retained. Malisiewicz and Efros (2008) generate a large pool of segments (Malisiewicz and Efros 2007) and recognize them using a nearest-neighbor classifier based on learned distance functions. Todorovic and Ahuja (2008), compute a hierarchical segmentation and find object subtrees similar to those learned during training. Unlike other methods they also model the relationship between objects and their subregions. A difficulty to overcome is the reliance on the structure of the hierarchical segmentation, which may not always be stable.

Another set of bottom-up approaches decides the object category directly at the level of image pixels (He et al. 2004; Shotton et al. 2009), or superpixels (Fulkerson et al. 2009; Gonfaus et al. 2010), based on features extracted over a supporting neighborhood. Textonboost (Shotton et al. 2009) classifies each pixel using a linear predictor on texton-layout features, learned using boosting. These features count the number of occurrences of a particular texton in a rectangular region at locations relative to each pixel. Because the output of local predictors can be noisy, often these approaches impose spatial constraints in a Conditional Random Field (CRF) framework to obtain smoother solutions. Smoothness can be obtained using contrast-sensitive pairwise potentials (Boykov and Jolly 2001), which facilitate label transitions at image discontinuities, or higher-order P^n potentials (Kohli et al. 2008) defined over extended image segments. These aim to bias the results towards solutions with small label variation inside homogeneous segments.

A common property of many approaches is the extraction of features over overlapping spatial supports, in order to increase robustness. One variant combines pixel and global image predictions (Csurka and Perronnin 2010; Gonfaus et al. 2010). Another variant adds predictions over extended regions obtained from low-level image segmentations (Ladicky et al. 2009b). Instead of reconciling predictions over overlapping regions, Gould et al. (2009a, 2009b) minimize an energy function over both the set of image segmentations and their labeling. Pantofaru et al. (2008) notice that pixels grouped together by all segments in different image partitions should have the same label and average category predictions on superpixels obtained by intersecting all segments.

A difficulty for pixel-level methods is segmenting multiple nearby instances of the same object without modeling the objects globally. This limitation has been partially addressed recently by adding rectangular bounding box detection constraints (Gould et al. 2009b; Ladicky et al. 2010) to a global energy formulation. In our method segments and their associated class scores are used instead. Arguably these are closer to the desired ground truth spatial object layout than bounding boxes.

Model-based Recognition. An alternative to bottom-up recognition is the use of shape models to constrain estimates of the spatial support of objects. This does not rule out models with bottom-up components that still use high-level information to obtain the final segmentation.

One class of model-based approaches assumes that object parts correspond to homogeneous image regions and these can be computed reliably. The methods assemble homogeneous image segments into full objects (Mori et al. 2004; Srinivasan and Shi 2007; Cour and Shi 2007) using knowledge of their part decomposition. Mori et al. (2004) first detect key parts among salient segments obtained using the output of the Normalized Cuts algorithm, then solve a constraint satisfaction problem to find probable configurations. Srinivasan and Shi (2007) compute several independent Normalized Cut segmentations by varying the number of clusters, then search for high-scoring interpretations obtained by assembling parts starting from those positioned lower in the image. Partial object segmentations obtained after each merge operation are matched against shape exemplars and used to prune implausible hypotheses. Cour and Shi (2007) show how to efficiently select sets of superpixels that best match an object template under a Hamming distance comparison metric. They first locate a set of parts, then repeat the process to assemble them into complete object hypotheses.

The difficulty of consistently segmenting object parts motivates another class of approaches that does not rely on low-level image segmentation. One possibility is to search

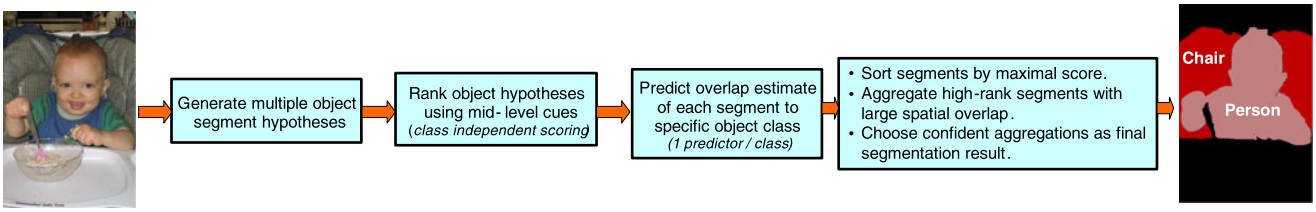

Fig. 3 Our semantic segmentation pipeline. Initially, an image is segmented into multiple figure-ground hypotheses constrained at multiple image locations and spatial scales, these are ranked (using mid-level cues) based on their plausibility to exhibit 'object-like' regularities (CPMC algorithm (Carreira and Sminchisescu 2010b)). Quality functions for different categories are learnt to rank the likelihood of segments to belong to each class. Several top-scoring segments are selected for post-processing. The final spatial support and the category labels are obtained sequentially from these segments, based on a weighted sum of selected segment scores

densely for object parts, then form segmentations by assembling stored partial ground truth responses associated with each part. Borenstein and Ullman (2002) segment objects in new images by combining partial ground truth segmentations associated with object fragments in training images. They identify putative fragments at image locations where the value of a predefined correlation function is maximal, then select those that locally optimize a cost function that combines the relevance of identified fragments, the value of their image correlation and a global consistency criterion. Leibe et al. (2008) employ a related top-down idea, but instead of convolving the image with masked fragments, compute descriptors on scale-invariant interest points and use a voting scheme to select consistent subsets.

As objects appear in a large variety of poses and shapes, dominantly top-down methods produce object segmentations that are often qualitative and can miss image detail. One way to improve such results is to integrate low-level information as image edges (Kumar et al. 2005; Toshev et al. 2010) or bottom-up hierarchical segmentations (Borenstein and Ullman 2008). Yu and Shi (2003) solve a constrained eigenvalue problem to find object segmentations biased by both object patch correlation and low-level edge alignment. Schoenemann and Cremers (2010) solve a minimum ratio cycle problem on a product graph consisting of responses on the boundary of a shape template. The Objcut method (Kumar et al. 2005) computes a segmentation biased both by low-level image cues and the output of a part-based probabilistic object-class model (pictorial structure) by solving a single min-cut problem. Toshev et al. (2010) developed a boundary structure segmentation technique that uses new chordiogram shape descriptors that make possible to match an image to an exemplar and simultaneously compute a binary segmentation as the result of a semi-definite programming relaxation.

Some techniques use more detailed processing only after a bounding box is obtained, being natural extensions to object detection methods. Yang et al. (2010) compute object bounding boxes using a deformable parts detector (Felzenszwalb et al. 2010) and use color cues and simple shape priors on the bounding box and the rectangular parts returned by the detector to obtain a segmentation. Gu et al. (2009) vote for the location and scale of bounding boxes based on matches between regions in the image and regions inside exemplar bounding boxes. They assign confidence scores to foreground and background regions and propagate these decisions to the rest of the image based on low-level similarities, by constraining an initial segmentation obtained using Ultrametric Contour Maps (Arbelaez and Cohen 2008).

3 Method Overview

Our recognition methodology relies on figure-ground segments generated by bottom-up computational processes. Our initial processing step produces a set of figure-ground segmentation hypotheses (out of which only figure segments are retained) for each image using the combinatorial CPMC segmentation algorithm (Carreira and Sminchisescu 2010b; Carreira and Sminchisescu 2012) (Fig. 2). The number of segments in this set depends on the image content: images with more edge structures tend to have more segments. Once segmentation hypotheses are obtained, the recognition framework consists of three stages: (1) segment ranking and filtering, (2) segment categorization and, (3) sequential aggregation and post-processing of multiple categorized segments.

The full recognition pipeline is depicted in Fig. 3. In the first stage, a *class-independent* quality function is learned in order to rank all segment hypotheses. This mid-level step separates segments with object-like regularities from those that do not have them. Based on the ranking produced in this step, a maximum (fixed) number of segments is selected for each image. These will be used for training and testing in later stages. This number depends on the difficulty of the dataset and is usually much smaller than the average number of segments generated by the algorithm (40–100 in our experiments). While our segmentation method is based on CPMC (Carreira and Sminchisescu 2010b), additional processing is implemented in the framework, and this will be described in detail in Sect. 4.

Fig. 4 An illustration of our segment categorization process. Each segment is given as input to regressors specialized for each category, producing estimated qualities. The maximal score across categories is used to sort segments and decide on their category

In the second stage, we learn a continuous scoring function for each object category, to assess the likelihood that a segment hypothesis belongs to that class. We follow a one-against-all methodology: the scoring function for each category is trained with all the input segment hypotheses that correspond or not to that category. In this way, each of the scoring functions is also discriminative and separates well one class from the others.

In the final stage, we sort the segment hypotheses by their scores and sequentially make detection and segmentation decisions based on a weighted combination of responses collected at high-rank segments. Image classification results are generated by taking maximal scores over all classes and among all image segments.

One of the main innovative points of this work, besides using *multiple figure-ground segmentations* from CPMC (rather than, *e.g.*, different multi-region image segmentations at different scales), is that category learning is performed by *regressing* on a quality function measuring the spatial overlap with the ground truth segments. Different segments carry different levels of information. For instance, in Fig. 2, a segment capturing the entire cow carries the most significant amount of information in determining its category. Parts of the animal, like the head, contain a lower, yet significant level of information. Segments that cover the cow and surrounding grassland provide context about where the cows can typically be found. Even background segments carry some information, *e.g.*, persistent mountain-grass segments show that this is a wilderness picture, and some objects like a sofa or a TV are unlikely in the scene.

Our regression-based training scheme is designed to more effectively (and accurately) exploit the various levels of information available in different segments. The quality function measures overlap with ground truth, which is a smooth measure of quality that degrades gracefully: full object segments have the highest overlap, parts of objects and surrounding segments have moderate overlap and dominantly background segments have the lowest (or no) overlap. By regressing on overlap, we more judiciously use partial information in all segments.

Prediction from our regression model generates a natural ranking of all segments based on their importance. This is illustrated in Fig. 4. Our decision stage exploits this ranking to create an accurate object mask. We group together high-confidence segments that cover a similar region and attempt to consolidate a single mask (and its label) by integrating information from all segments. To achieve this, a confidence score is computed for each pixel as the weighted sum of scores of the segments that cover it. If all segments agree that a given pixel should belong to a given category, the likelihood of this assignment will be high. If there are conflicts, for example one segment votes that a pixel is more likely part of a dog whereas the other three vote for a cat, the confidence would decrease (see Fig. 7). A learned threshold on the pixel confidence score determines if the pixel should be included in the final mask.

4 Segment Generation and Filtering

4.1 Basic Approach

The input to our processing pipeline are multiple figure-ground segmentations obtained by CPMC (Carreira and Sminchisescu 2010b). These are obtained by solving a series of constrained min-cut problems, for putative foreground seeds constrained on a regular image grid and for background seeds sampled as various subsets of pixels on image borders. Multiple significant scale breakpoints (solutions) for these problems are computed using parametric max-flow in polynomial time (Gallo et al. 1989).

Ranking segments based on their mid-level properties is the second step in the framework. During this phase, the segments generated by CPMC are filtered based on a quality function learned using regression, with covariates chosen as mid-level segment properties and Gestalt features (see Carreira and Sminchisescu 2010b). We additionally use SIFT and HOG descriptors computed on the foreground to augment the feature set used to predict segment quality. Section 5.1 provides detail on the computation of these histogram features.

The regression function we learn for segmentation is class-independent (there is a single such function in the framework), with input given by segment features and output given by the maximal overlap between a segment and all the ground truth segments. The scale of the problem rapidly runs into millions: for instance, a dataset of 2000 images and 1000 segments for each image gives rise to a problem with 2 million examples. Therefore, at first linear methods appear to be the only practical choice for learning. However, random Fourier approximations can be used to transform the features linearly, to accurately approximate non-linear similarity measures (Rahimi and Recht 2007; Bo and Sminchisescu 2009; Vedaldi and Zisserman 2010). In the Fourier methodology we consider an initial kernel and generate a new set of features based on randomly sampling multiple components from its Fourier transform. A linear regressor working on the transformed representation usually offers performance close to those of nonlinear kernel machines (Rahimi and Recht 2007). In this paper, we use random Fourier approximations for all image features and for all kernels employed for class-independent ranking. The mid-level segment descriptors are transformed using random Fourier projections corresponding to a Gaussian kernel, and the histogram features (SIFT and HOG) are transformed separately using Fourier embeddings derived from the skewed chi-square kernel (Li et al. 2010b). The resulting dimensions are concatenated to generate the final covariate vector.

Beside random Fourier approximations, we employ additional processing for segment ranking. In the next subsection we define a customized overlap measure that is better tailored to the performance metric used on the PASCAL VOC challenge (Everingham et al. 2010). In Sect. 4.3 we show how to learn the class-independent ranking function using linear regression, for problems where it is no longer possible to load the entire training set into memory.

4.2 Quality Function

A common measure used to assess segmentation quality is the 'intersection-over-union' overlap, or IOU-overlap. Let S_p and S_q be two generic segments and G_q be a ground truth segment. IOU-overlap is defined as:

Fig. 5 (Best viewed in color) Segments with different overlaps with the ground truth. The two numbers shown are the proposed FB-overlap on the left and the standard IOU-overlap on the right. It can be seen that FB-overlap favors segments that do not contain a lot of background, whereas IOU-overlap is indifferent to such effects

$$O_{iou}(S_p, S_q) = \frac{|S_p \cap S_q|}{|S_p \cup S_q|}. \quad (1)$$

Sample segments from an image and their IOU-overlap to the ground truth are shown in Fig. 5. To show how different these can be, the best 4 segments (w.r.t. the ground truth segment) and the worst 4 segments are shown on the top and bottom rows. On the second and third row, selected segments that partially overlap the object are shown.

The choice of quality function *for training* is not confined to the original IOU-overlap used in Carreira and Sminchisescu (2010b). Depending on the task, different quality functions can be used. For example, in the PASCAL VOC segmentation challenge, the performance measure places more importance on larger objects. Moreover, the accuracy of the background class is also measured, therefore segmentations that handle the background correctly are also preferred. These two constraints are not entirely accounted for by the standard IOU-overlap measure (1). It can be seen from Fig. 5 that some of the very large segments have significant IOU-overlap with the ground truth object, although this is not desirable, in order to accurately classify the background.

To palliate some of these effects, we propose a new overlap measure for training that we refer to as the Foreground-Background Overlap, or FB-overlap. It accounts for both overlap with the foreground and overlap with the background, and compensates against large segments. The measure is computed as:

$$O(S_p, G_q) = \frac{C\sqrt{|S_p|}}{\log|S_p|\sqrt{N_c^{fg}}}\frac{|S_p \cap G_q|}{|S_p \cup G_q|} + \frac{C\sqrt{|\overline{S_p}|}}{\log|\overline{S_p}|\sqrt{N^{bg}}}\frac{|\overline{S_p} \cap \overline{G_q}|}{|\overline{S_p} \cup \overline{G_q}|} \quad (2)$$

where N_c^{fg} and N^{bg} are the number of foreground and background pixels in the entire training set, with c the class of the ground truth segment G_q, and $\overline{S}$ is the image complement of a segment hypothesis. $C = 90$ is a normalization constant that scales the range of the measure so as to match the range of IOU-overlap on the VOC dataset. The class-independent quality function of the segment is computed as

$$O(S_p, I) = \max_{G_q \in I} O(S_p, G_q) \quad (3)$$

where I is the image where the segment resides in.

FB-overlap emphasizes large segments mildly, while still not penalizing significantly small to moderately sized segments—because the background is also considered, oversized segments are not preferred. From Fig. 5, it can be seen that under the new measure, the segments that correspond to objects and parts tend to have higher rankings under FB-overlap than under IOU-overlap. Segments that overlap significantly with the background are given comparatively lower FB-overlap scores. Besides, FB-overlap provides a mechanism to balance the training set sizes among different classes. For example the class `person` in VOC has around 8 million training pixels, whereas `bicycle` has only around 300,000. The overlap in the class of `bicycles` are made mildly higher under the FB-overlap measure in order to equalize the prediction accuracy among different classes.

The formula is derived using ideas from residual analysis (Tukey 1977) on the maximal predicted scores of the regression model (Sect. 5). Our principle in designing the scoring function is that although larger segments are to be favored in general, random segments (that do not correspond to any ground truth) of different size should have roughly the same predicted scores. During the design phase of the measure, the entire framework has been tested several times and changes to the measure were made. The end result is formula (2). In Fig. 6 it can be seen that after tuning, the lower bound scores on all the segment sizes are roughly similar. Overall, the use of FB-overlap improves the VOC result by around 1%. We will use the notation O for either overlap measure in the sequel. Notice however that FB-overlap will only be used in PASCAL VOC training, whereas IOU-overlap is used for all the other datasets.

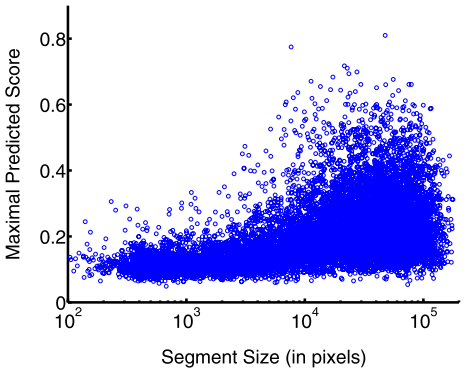

Fig. 6 Predicted FB-overlap on VOC 2010 validation dataset against size of the segment (in pixels). It can be seen that the lowest predicted score on segments of different size is roughly the same under the new FB-overlap measure

4.3 Linear Regression with Partial-Storage

As the number of images and segments increases, they no longer fit into memory. Since SIFT and HOG features are not very sparse, a dense representation needs to be used. For instance, in the VOC 2010 dataset, there are around 10,000 images. We use 800 segments for each image and 3,600 Fourier feature dimensions as training data for segment ranking. This sums to 8 million examples, each having 3,600 dimensions. Storing the features using single precision (4 bytes) requires 107 gigabytes, which is beyond the current memory capacity of many personal computers. Some progress has been made in designing large-scale SVM classifiers (Yu et al. 2010), but those generally require loading the data into memory multiple times and are extremely time-consuming. Previous work on large-scale learning mostly focused on text categorization, but because those features are considerably sparser than in computer vision, the storage problem is less stringent.

In this work we take a simple approach. It is well-known that for least-squares and related methods, the problem can always be transformed into an optimization problem on the mean and the covariance matrix—the sufficient statistics of the Gaussian distribution (Bishop 2007). These can be built from the data in chunks. Formally, in regression, our goal is to solve the quadratic optimization:

$$\min_w \sum_i \left(w^{\mathrm{T}} x_i - y_i\right)^2 + C\Omega(w) \quad (4)$$

where x_i represents segment features, y_i the overlap of a segment, e.g. (2), and $\Omega(w)$ can be any regularizer applied on w, e.g., $\|w\|_2^2$, $\|w\|_1$. This is equivalent to

$$\min_w w^{\mathrm{T}} X^{\mathrm{T}} X w - 2w^{\mathrm{T}} X^{\mathrm{T}} y + C\Omega(w) \quad (5)$$

where $X^{\mathrm{T}}X = \sum_{i=1}^n x_i x_i^{\mathrm{T}}$ and $X^{\mathrm{T}}y = \sum_i x_i^{\mathrm{T}} y_i$ can be computed by loading a single or a chunk of x_i into memory at a time. Therefore, all methods that use a quadratic loss function can work without loading all training data into memory. This includes ordinary least squares, ridge regression, lasso and group lasso methods. We work with ridge regression, under a quadratic regularization term $\Omega(w) = \|w\|_2^2$.

One common pitfall in applying the approach is normalization. For instance, if a standard normalization is to be performed ($x = \frac{x-\bar{x}}{\mathrm{std}(x)}$, where $\bar{x}$ is the mean and $\mathrm{std}(x)$ is the standard deviation), it is tempting to compute the mean and variance for each chunk of data separately because not all data can be loaded into memory simultaneously. However this shortcut does not work well—in our experiments we observed a performance drop of up to 2%. The correct mean and variance still need to be computed, although this means tediously loading the data chunk by chunk, computing $\sum_i x_i$ and $\sum_i x_i^2$ for each chunk, summing it up to obtain the mean and variance and loading the data again, in chunks, to normalize.

5 Segment Categorization

For categorization, we compute multiple figure-ground segmentations and extract multiple sets of features for them. A weighted sum of kernels on different types of features is used, with hyperparameters learned on the validation set. Based on the features and the coefficients of the kernel combination, support-vector regression on the overlap measure generates a scoring function for each object category.

5.1 Multiple Features

Features are extracted for each segment. We use 7 feature types. In order to model the object appearance we extract four bags of words of gray-level SIFT (Lowe 2004) and color SIFT (van de Sande et al. 2010), on a regular grid, two on the foreground and two on the background of each segment. Computing bags of words on the background of a segment models a coarse scene context.

To encode shape information we extract three pyramid HOGs (pHOG) (Bosch et al. 2007), which are concatenations of histograms of gradients extracted at different resolutions. Each level of the pyramid divides each cell from the previous level into four higher resolution cells. The first level has a single cell. The first of our three pHOGs is defined directly on the contour of the foreground, whereas the other two operate on edges detected by globalPB (Maire et al. 2008) inside the foreground. The first two pHOGs adapt the cell dimensions in order to tightly fit the bounding box of the foreground segments, whereas the third uses square cells. The pHOG with square cells always covers a square region of the image, so we pad the image with zero, whenever this square region is partially outside the image. We use these different pHOGs so they can complement each other. The gradient orientation is discretized into 16 bins with values restricted between 0 and 180 degrees, as we chose to ignore the contrast direction.

A chi-square kernel $K(x, y) = \exp(-\gamma \chi^2(x, y))$ is used for each type of histogram features and we use a weighted sum of such kernels for regression. The coefficient and the width hyperparameters of each chi-square kernel are learned using an optimization scheme detailed in Sect. 5.3.

5.2 Learning Scoring Functions with Regression

Let us consider an image I with ground truth segments $\{G_q^I\}$. The segmentation algorithm provides a set of segments $\{S_p^I\}$ for image I. Denote also the K object categories $\{c_1, c_2, \ldots, c_K\}$. Let $\mathbf{1}(x)$ be the indicator function.

As discussed in the previous section, we learn K functions $f_1(S_p^I), \ldots, f_K(S_p^I)$ by regression on a quality measure for segments. For each putative segment S_p^I, we compute its overlap, given by (2), against all ground truth segments $\{G_q^I\}$ in the image. The target value y_{kp}^I for a segment S_p^I and a category c_k is the maximal overlap with ground truth segments that belong to c_k:

$$y_{kp}^I = \max_{G_q^I \in c_k} O\left(S_p^I, G_q^I\right). \tag{6}$$

Usually a segment S_p^I overlaps with at most a few ground truth segments. For categories that do not appear in an image I, $y_{kp}^I = 0$. After training, the estimated qualities for S_p^I on improbable categories tend to be close to 0. Therefore, this regression scheme is able to both estimate the quality of segments and classify them into categories.

To learn the function $f_k(S_p^I)$ for each c_k, we use a nonlinear support vector model (SVR) to regress on y_{kp}^I against x_p^I, the features extracted from segments S_p^I. The SVR optimization problem can be derived as:

$$\begin{aligned} \min_{w,\xi,\eta} \ & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\xi_i + C\sum_{i=1}^{n}\eta_i \\ \text{s.t.} \ & \xi_i \geq 0; \eta_i \geq 0, \forall i \\ & \left\langle w, \phi(x_i)\right\rangle \geq O(y_i, y) - \epsilon - \eta_i \\ & \left\langle w, \phi(x_i)\right\rangle \leq O(y_i, y) + \epsilon + \xi_i \end{aligned} \tag{7}$$

where $\phi(x_i)$ is a nonlinear feature transform of the input x_i, defined implicitly by the kernel $K(x_i, x_j) = \langle \phi(x_i), \phi(x_j)\rangle$ detailed in the next section; ϵ is a small constant, usually 0.05 or 0.1. Using the kernel trick, it is possible to represent $f(S_p^I)$ in dual form as $f(S_p^I) = \sum_i \alpha_i K(x_i, x_p^I)$, where x_i are support vectors from the training set, and the α are coefficients obtained by the SVR optimizer.

The maximal score and the final segment category are given by $\max_k f_k(S_p^I)$ and $\arg\max_k f_k(S_p^I)$, respectively. However, scores on all categories will be used in the post-processing stage. One can avoid this type of post-processing and directly choose the segment with maximum responses, $\arg\max_{k,p} f_k(S_p^I)$, as output. We call this a *simple decision rule*. In experiments we test this rule against more complex post-processing rules.

A main challenge is, once again, the training set size. Since each segment is used as an example, the number of training examples could be large. We mine hard negatives, an approach that has become popular recently (Felzenszwalb et al. 2010). First, regressors are trained only on ground truth segments and putative segments that best overlap the ground truth for each training object. Then, we classify all training segments, find misclassifications, and re-estimate the model parameters with these segments added to the training set. Given a memory budget, we often add only a subset of the misclassified segments and repeat the process multiple times. Using this procedure, we are able to train on the Caltech-101 and the VOC 2009/2010 datasets in only a few hours.

5.3 Learning the Kernel Hyperparameters

Fundamental to (7) is the form of the kernel function (Kumar and Sminchisescu 2007). Existing multiple kernel learning methods that optimize performance measures on the training set suffer from overfitting in many cases (Kumar and Sminchisescu 2007; Gehler and Nowozin 2009). Therefore, we optimize the kernel hyperparameters on the validation set. Since we employ a weighted addition of multiple kernels, it is infeasible to estimate all kernel hyperparameters by means of grid search. Instead, we use gradient descent on an objective function defined on the validation set. To speed-up the process, we apply the algorithm only on a subsample of the data, consisting of segments that best overlap the ground truth. The idea is that kernels need to at least model well the similarity between the clean segments in different classes. Given two exemplars x_i and x_j the additive kernel model is

$$K(x_i, x_j) = \sum_k \beta_k K_k(x_i, x_j; \gamma_k), \tag{8}$$

where γ_k is the width of the chi-square kernel. We learn β and γ jointly by directly minimizing the misclassification rate over all images in a (hold-out) validation set:

$$\min_{\beta,\gamma} \sum_{S_p^I \in c_k} \mathbf{1}\Big(f_k(S_p^I) < \max_i f_i(S_p^I)\Big), \tag{9}$$

where $f_k(S_p^I) = \sum_{j,k} \alpha_j \beta_k K_k(x_i, x_j; \gamma_k)$ is trained with SVR using the kernel (8) on the current β and γ.

To be able to employ gradient-based optimization algorithms, we use the sigmoid function as a continuous approximation to the indicator:

$$\sum_{S_p^I \in c_k} u\Big(f_k(S_p^I) < \max_i f_i(S_p^I)\Big), \tag{10}$$

where $u(x) = \frac{1}{1+e^{-\sigma_0 x}}$. A quasi-Newton method is used to find a local optimum for the parameters. Since both the number of kernel parameters and the number of examples are small, this process is fast.

We found that hyperparameters obtained by this procedure are very stable. We learned them on the VOC 2009 train and validation sets and used them throughout all our experiments, both in the VOC 2009 and 2010 (validation and test sets) and for the ETHZ Shape, with consistently good performance.

5.4 Connections with Structural SVM

There are interesting connections between our learning approach and the method of Blaschko and Lampert (2008), which uses a structural SVM (Tsochantaridis et al. 2004) to learn a model for detection. For a bounding box y_i and a ground truth bounding box y, let x_i be the feature vector for y_i and x the feature vector for y. The structural SVM formulation for sliding window prediction is:

$$\begin{aligned} \min_{w,\xi} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & \xi_i \geq 0, \forall i \\ & \big\langle w, \phi(x, y) - \phi(x_i, y_i)\big\rangle \geq 1 - O(y_i, y) - \xi_i . \end{aligned} \tag{11}$$

Structural SVMs have a larger feature space than standard SVMs because the output is kernelized and y appears jointly in the embedding function $\phi(x, y)$. However, the output vector of (Blaschko and Lampert 2008) is 5-dimensional: the class label and the locations of the bounding box. This makes the difference between the input and the joint feature space dimensionality unimportant.

Another difference to Blaschko and Lampert (2008) is that all possible rectangular regions are considered. This is feasible within a branch-and-bound procedure (Lampert et al. 2008) that can rapidly prune out irrelevant regions of the search space, for the restricted class of features and linear models used in Blaschko and Lampert (2008). However, it is difficult to adapt both the structural SVM and the branch-and-bound methodology for the much more powerful non-linear SVM predictors and image features we want to be able to use. Our task is easier, however, because our use of a compact pool of image segments eliminates the need to process a large number of bounding boxes.

Ignoring these two differences, the structural SVM (11) looks superficially similar to our SVR formulation (7). It could be seen that if we assume $\langle w, \phi(x, y)\rangle = 1 - \epsilon$, then the last constraint in (11) would be the same as the last constraint in (7). The difference is clear, however: (11) scores the ground truth bounding box and ensures its quality is better than other tentative bounding boxes, with margin determined by the overlap. Meanwhile, (7) simply scores all the segments and measures an absolute quality of the segments. We argue that our approach has important advantages. It does not only guarantee the highest rank for the ground truth, but also the correct ranking for all remaining (putative) segments: those with higher overlap will simply have

higher scores. For structural SVM, only the *smallest* margin between the best segment and other segments is imposed based on the overlap. Since each segment may have an arbitrarily low score without violating margin constraints, the segment ordering is not preserved inside the non-maximum subset.

6 Sequential Segment Post-processing

6.1 Generating Segmentation Results

The challenge of this stage is to form a consistent segmentation and labeling for images containing multiple objects, given a set of plausible, reasonably high ranked segments with initial category labels. The simple decision rule of only using the highest scoring segment cannot handle multiple objects in an image. The non-maximum suppression method that removes all regions overlapping the highest scoring one is standard in bounding box detection, and can be used similarly for segmentation, but we argue that a better approach can be constructed by exploiting the redundancy of class predictions from multiple overlapping segments. Our methodology employs a weighted consolidation of segments and a sequential interpretation strategy, in order to analyze images with multiple objects.

Figure 7 shows an example. After classification, the highest-ranked segment was assigned the correct category, `cat`, but this segment also contains background around the object. The next two segments located the cat exactly, but were classified as `dog`. One can see that predictions for these two segments are not very decisive, since `cat` and `dog` have very similar scores. By taking into account the class predictions of such multiple overlapping segments, it is possible to achieve more robust decisions.

Since the higher-ranked segments should have higher probability of representing full objects, we proceed iteratively. First, we consider the highest-scoring segment as a seed and group segments that intersect it. To decide which segments to group, we compute a segment intersection measure:

$$\mathrm{Int}(S_p, S_q) = \frac{|S_p \cap S_q|}{\min(|S_p|, |S_q|)}. \tag{12}$$

Under this criterion, parts have 100% intersection with full objects, therefore they are always grouped together. We consider segments with intersection $> \tau_1$ ($\tau_1 = 75\%$ chosen based on the validation sets) as candidates for combination. In the end, a list L_1^I (1 is used as index because this is the first candidate mask in the image) of segments is generated, in which partially overlapping segments are sorted according to their descending scores.

We then generate the scores for each pixel and each class in the image by weighted voting based on the segments in the list

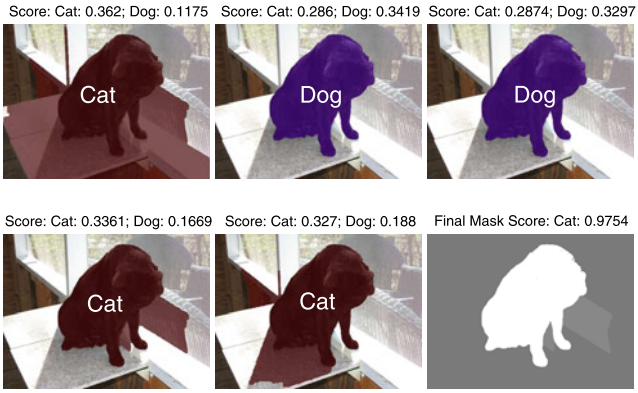

Fig. 7 (Best viewed in color) An image of a cat from the VOC2009 dataset. We show the cat/dog scores of the 5 top scoring segments from the image. It is relatively difficult to distinguish if this instance is a cat or a dog, from the foreground/object information only (e.g., *top-middle* and *top-right segments*). However, our algorithm takes advantage of multiple slightly different overlapping segments to produce a robust decision, that consistently improves upon the simple decision rule. In the Final Mask, the cat itself has the strongest score (indicated by high intensity values)

$$g_k(p_j) = \sum_{S_i \in L_1^I} w_i \mathbf{1}(p_j \in S_i) f_k(S_i), \tag{13}$$

where S_i represents the i-th ranked segment in the list L_1^I, k is a certain class, p_j is a pixel, $f_k(S_i)$ is the predicted score for S_i on class k. Through this equation, scores on segments are transferred to scores on pixels inside segments. Then a weighted combination is taken, with segments with higher prediction having higher weights. For a pixel, its scores are only counted on the segments that overlap it, as given by the term $\mathbf{1}(p_j \in S_i)$. Therefore, pixels that appear in all segments get higher scores, whereas pixels that only rarely appear get lower scores. Besides, because scores are computed for each class separately, if all overlapping segments agree on the label, that class is supported strongly. Finally, each pixel is assigned to the class that has the highest score: $g(p_j) = \max_k g_k(p_j)$.

We define the term *mask* as a figure-ground segmentation with each pixel on the foreground classified to some category, in order to differentiate it from *segments*. To separate foreground and background, only pixels with final scores $> \tau_2$ are displayed in the aggregated mask M_1^I ($\tau_2 = 0.55$ is selected, based on validation data). The score of the mask is given by

$$g\left(M_1^I\right) = \max_{p_j \in I} g(p_j). \tag{14}$$

The last image in Fig. 7 shows an example of the final mask, where it can be seen that the classification is now correct and the scores are highest in the cat region and much lower in other regions.

The weights w_i in (13) are associated with the rank (in the list m) of the segment only, uniformly across different

Algorithm 1 Postprocessing pipeline for image I. Sequential aggregation of multiple categorized segments.

input Segments $\mathbf{S} = \{S_1^I, \ldots, S_m^I\}$, with predicted scores $f^k(S_i^I)$ for each class k.
output Final masks $\{M_i\}$ on the image I.

Sort the segments descending by maximal score $f(S_i^I) = \max_k f^k(S_i^I)$ on all classes.
n = 1
while S is not empty **do**
  Select $S_n^I = \arg\max_i f(S_i^I)$, the segment with the highest maximal score.
  Find all segments that have at least τ_1 intersection with S_n^I, let them be L_n^I, still sorted by maximal score.
  For each pixel p_j in the image, compute pixel score $g_k(p_j)$ for each class k by

$$g_k(p_j) = \sum_{S_i \in L_n^I} w_i \mathbf{1}(p_j \in S_i) f_k(S_i). \tag{15}$$

  for each pixel p_j **do**
    if $\max_k g_k(p_j) < \tau_2$ **then**
      $M_n(p_j) =$ background
      {Classify p_j as background.}
    else
      $M_n(p_j) = \arg\max_k g_k(p_j)$
      {Classify p_j as class k.}
    end if
  end for
  if $\max_{k,j} g_k(p_j) > \tau_3$ **then**
    Output M_n
    {The score of the mask is given by the highest pixel score in the mask. It must exceed a threshold to be retained in the final semantic segmentation.}
  end if
  Delete all segments in L_n^I from **S**.
  n = n + 1
end while

images and classes. These are learned using linear regression on targets that measure the overlap of the generated masks with ground truth, in the validation set.

After we have generated a final mask M_1^I from segments in L_1^I, we remove the segment set L_1^I and the foreground region in M_1^I from the image and consider it consolidated. Then we proceed with the next highest-ranked segment. Based on the same procedure we generate L_2^I and M_2^I, *etc*. Altogether in the VOC dataset usually 6–7 final masks are sufficient. In the end, the final masks are filtered, and only those with mask score $g(M_j^I) \geq \tau_3$ ($\tau_3 = 0.66$ chosen based on validation data) are retained in the final result. It can be seen that the false positive rate is high, therefore so many stages are needed to reduce variance. With more training data and improved regression accuracy, we can probably remove some of the filtering steps. The post-processing method is detailed as Algorithm 1.

We also implement a simple filter based on the class co-occurrence frequencies in the VOC training set (Gonfaus et al. 2010). A co-occurrence frequency matrix is computed, whose ij-th entry counts the number of times two objects of class i and j co-occur in the same image. During testing, we filter object pairs that never co-occur. This only improves performance slightly in our experiments (see Table 1). Further discussion on alternative decision rules appears in Sect. 7.1.

6.2 Generating Detection and Classification Results

To generate detection results, the method changes slightly. We use overlap (1) to replace the intersection measure (12) used for grouping segments. This is because when using an intersection measure, small objects are combined within a larger segment containing them. For instance, sometimes we combine two bottles placed next to each other into one large segment enclosing both. This may not affect the segmentation performance measure, but for detection, a single bounding box would enclose both bottles and would count as one false positive and two false negatives. Adapting the criterion from intersection to overlap makes the method work well for detection. Also, we do not use a threshold to determine whether to output a segment as in Algorithm 1. Instead, we simply output all the generated final masks. For classification, in each image we simply find the mask with the highest score and output its label.

7 Experiments

The experiments are divided in two parts. The first section shows proof-of-concept studies, where various important aspects of the algorithm are tested. In the second section, we show results of our recognition framework (denoted SvrSegm, abbreviated from SVR on SEGMentations) applied to three key tasks in image understanding: image classification, object localization and object segmentation. We also compare with previously reported results.

The segments used in all experiments except those on PASCAL VOC 2010 were generated by CPMC based on the same 5×5 grid of seeds and the same parameters detailed in the original CPMC paper (Carreira and Sminchisescu 2010b). The experiments on PASCAL VOC 2010 used CPMC with a different set of parameters tuned for producing a larger initial pools of segments. Additionally, these experiments used an expanded set of seeds. Further detail on the PASCAL VOC 2010 segments can be found in the documentation provided with the publicly available CPMC segmentation implementation (Carreira and Sminchisescu 2010a).

Table 1 Study of the effects of post-processing on the VOC2010 validation set. The `Simple` scheme uses no post-processing and outputs only the best segment. `NMS` is the result obtained using non-maximum suppression. `1-Seg` outputs at most 1 best segment from post-processing, but allows to combine multiple segments. `No new segment` allows an arbitrary number of segments, but selects the segment from the original pool that is closest to the post-processing result. In `No co-occur`, the result is not filtered by the frequency matrix of segment co-occurrence. `Full` uses the full post-processing pipeline described in the paper

Class name	Simple	NMS	1-Seg	No new segment	No co-occur	Full
Mean	30.47	31.84	33.28	33.76	33.91	**34.30**
Background	79.01	80.74	81.60	81.71	**82.03**	**82.03**
Aeroplane	35.65	41.66	**44.47**	42.13	43.80	43.97
Bicycle	16.66	16.03	**16.92**	16.03	16.14	16.29
Bird	30.99	31.22	**34.76**	33.24	32.38	32.55
Boat	29.65	32.21	**34.42**	33.59	33.61	33.81
Bottle	40.72	41.94	40.81	42.26	**43.07**	**43.07**
Bus	44.88	48.25	47.72	47.64	49.55	**49.70**
Car	**56.92**	53.63	55.64	55.58	53.94	56.19
Cat	34.35	36.20	37.10	35.86	**37.26**	36.28
Chair	4.94	**7.35**	4.24	6.26	6.79	6.79
Cow	8.51	8.80	11.57	13.08	**13.48**	13.13
Dining Table	12.53	14.43	19.84	**24.12**	23.56	23.31
Dog	13.94	14.98	16.57	17.43	17.35	**17.52**
Horse	**32.53**	29.03	31.14	29.44	30.30	30.33
Motorbike	42.04	41.36	**47.61**	46.42	45.47	46.80
Person	26.26	30.85	27.67	33.35	**33.73**	33.71
Potted Plant	**20.54**	20.15	18.74	18.70	19.01	19.01
Sheep	30.36	35.62	33.20	36.74	36.31	**38.67**
Sofa	14.90	15.79	15.94	**20.19**	17.47	19.93
Train	35.28	37.20	41.93	41.86	41.94	**42.39**
TV/Monitor	29.25	31.16	**36.94**	33.33	35.00	34.75

The initial pools of segments have, averaged over all images, 95 segments for the ETHZ shape dataset, 64 for Caltech 101, 145 for VOC2009 and 736 (with the new parameters) for PASCAL VOC 2010. One possible way to measure the CPMC performance on a dataset is to compute the maximum IOU-overlap between each ground truth object and any generated segment, then average over all objects. This score also illustrates how difficult the low-level segmentation is for each dataset. Our pools of CPMC segments obtain 0.83 on Caltech 101, 0.85 on ETHZ Shapes and 0.66 on PASCAL VOC 2009. With the new CPMC configuration, on PASCAL VOC 2010 we obtain a maximum IOU-overlap of 0.74. Note that the PASCAL VOC datasets are considerably more challenging for low-level object segmentation. More detail about these datasets will be given in the next subsections.

7.1 Proof-of-Concept Experiments

In this subsection we test two concepts presented in the paper: (1) Regression against overlap and (2) Post-processing. We use the PASCAL VOC 2010 dataset to perform these tests.

The PASCAL VOC 2010 segmentation dataset contains 1928 images (with 4203 objects) for training, which are divided into 964 images (2075 objects) in the `train` set and 964 images (2028 objects) in the `val` set. Objects are selected from 20 classes. A hold-out `test` set of 964 images is used to evaluate the performance of the algorithm. For this data, annotations are not available and one must submit results to an external evaluation server.[1] The performance is measured using per-class overlap, defined as:

$$\text{segmentation accuracy} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{16}$$

where TP is the number of true positive pixels of the class, FP is the number of false positive pixels and FN is the number of false negative pixels. The TP, FP, and FN values are summed across all the images of the test set. In the end, the 21 per-class overlaps (all the 20 classes plus the background class) are shown, and the mean performance is an average over the 21 individual accuracies. Naturally, this performance measure favors big segments, which may often be

[1] available at http://host.robots.ox.ac.uk:8080/.

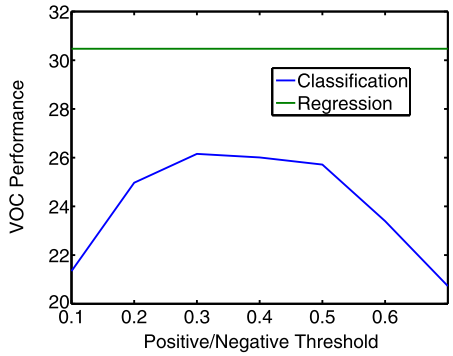

Fig. 8 Comparison of classification and regression approaches. Even the best threshold for classification gives results vastly inferior to regression

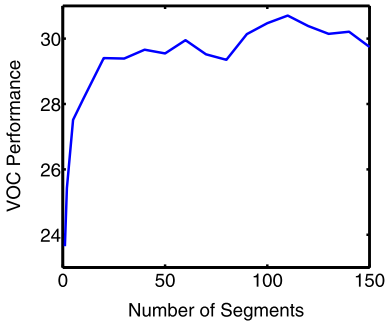

Fig. 9 Performance as a function of the number of segments. Performance improves very quickly initially, as more segments are added and reaches its peak for 110 segments. Beyond that value, it deteriorates slightly

more important in understanding the image, although this is perhaps arguable. Our FB-overlap measure (2) is designed to reflect the evaluation objective in a principled manner, and shows the flexibility of our approach in adapting to different objectives.

In this subsection we perform experiments by training on the `train` set and testing on the `val` set. This is consistent with the recommended usage of the two sets: to test the model and identify parameters. We use the VOC mean performance to evaluate the models.

First, we test our one-vs-all regression scheme against the more commonly used classification approach. We set an acceptance threshold on the overlap so that segments with overlap higher than a threshold are considered positives for the class and the remaining ones are considered negative; we varied this parameter from 0.1 to 0.7. All the other parameters are the same except that we use SVM classification instead of regression. To avoid interference from external factors, post-processing is disabled in this experiment, and only the best segment for each image is reported. The result is shown in Fig. 8. The regression scheme obtained 30.47% as VOC mean score. Among the threshold values tested for classification, the best threshold (0.3) achieved 26.15%. Therefore, the one-against-all regression approach brings at least a 4% performance improvement, and has one less parameter to tune compared to classification (the acceptance threshold).

Another relevant aspect of study is the number of segments required by the algorithm in order to obtain good results. This can also be seen as a test on the performance of the class-independent segment ranking method (Sect. 4). For this study we again disabled post-processing operations and output only the best segment for each image. The results in Fig. 9, perhaps surprisingly, show that even by using only a few segments, the results are not much lower than the best ones that we achieved. Moreover, when using more than 110 segments, the accuracy does not saturate but deteriorates slightly. Since the classifier has limited inductive power, it seems that when there are too many low quality segments in both the training and testing sets, spending too much capacity on predicting those well negatively impacts the ability to correctly generalize on good segments. This justifies our need of a multi-stage segment filtering approach.

We also test the importance of various factors in post-processing. Compared with the straightforward approach of selecting the best segment for each image, there are two improvements from post-processing: (1) Improving the quality of the segment; (2) Obtaining multiple segments per image instead of just one. In order to separate these factors, we compare the full post-processing results with strategies that only extract one segment per image.

We show detailed results of this experiment in Table 1, where the improvement provided by each step is recorded. From the results, we note that post-processing improves the quality of the segmentation by about 3% (improvements are observed in 17 out of 21 classes) when moving from `Simple` to `1-Seg`. Besides, our approach significantly outperforms non-maximum suppression (`NMS`). However, allowing for multiple segments leads to mixed results: the performance deteriorates in 8 out of 21 classes and only improves in 12. The co-occurrence criterion is not entirely satisfactory either: from the simpler `No co-occur` to `Full`, only 4 classes show significant performance improvement.

7.2 Performance Experiments

7.2.1 Image Classification: Caltech-101

We also test the image classification performance of our algorithm in the Caltech-101 benchmark (Fei-Fei et al. 2007). As in standard approaches, we report results averaged on all the 101 classes, over 3 different random splits. For each class, we use 5, 15 or 30 images for training and up to 15 images for testing, following the common setting in the literature. We train the model using ground truth segmentation masks provided with the dataset. In Fig. 10, we compare our

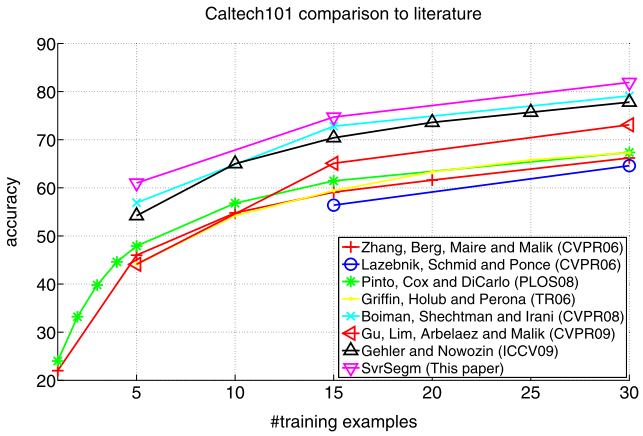

Fig. 10 Comparisons on Caltech-101 (Boiman et al. 2008; Grauman and Darrell 2005; Griffin et al. 2007; Lazebnik et al. 2006; Pinto et al. 2008; Zhang et al. 2006). SvrSegm outperforms the current state of the art for all training regimes

results against existing approaches. Our scores consistently improve the current state-of-the-art in all training regimes. In particular, our approach outperforms other multiple kernel frameworks such as Gehler and Nowozin (2009) and the segmentation-based framework of Gu et al. (2009).

We have also run some of the proof-of-concept experiments on this dataset, in order to compare our regression scheme with SVC (support vector classification). We also evaluate the impact of post-processing. Since the outputs of our SVR are different from those of SVC, we do not employ the post-processing algorithm in this comparison, but use only the simple decision rule. It turns out that in Caltech-101, the simple decision rule works well. Table 2 confirms that regression works significantly better than classification. More sophisticated post-processing does not outperform the simple decision rule in this case, except for the small training regimes (5 training images). Two experiments were pursued further. The first uses only the best segment in our hypothesis pool for both training and testing; the second uses only the ground truth segment for the same purpose. The experiments show that we are very close to saturation: the results generated by training and testing only on our best segment for each image are not significantly better than results based on multiple segments. Arguably, in this dataset, improvements are more likely to emerge from better features and better segments, than the decision rule itself.

7.2.2 Detection: ETHZ Shape Classes

We compare our detection results with the ones reported in Gu et al. (2009), a competitive segmentation-based recognition approach. We use the ETH Zurich database (Ferrari et al. 2007) which contains 5 shape categories and 255 images. We follow the experimental settings in Ferrari et al. (2007),

Table 2 Comparisons of different settings of SvrSegm for learning in Caltech-101. Our regression on overlap framework significantly outperforms classifier-based implementations. Post-processing helps somewhat for small training sets. We also show the result produced by using only the best ranked segments and ground truth segments (in both training and testing), to give an idea of the best performance the current recognition framework could obtain by improving the segmentation

Method	5 Train	15 Train	30 Train
Classification	58.6	72.6	79.2
Regression	59.6	74.7	82.3
Reg. w/ Post-Processing	60.9	74.7	81.9
Best Segment	62.4	75.8	82.5
Ground Truth Segment	71.7	83.7	89.3

and use the PASCAL criterion to decide if a detection is correct. The image set is evenly split into training and testing sets and performance is averaged over 5 random splits. For training with just bounding box data, we automatically extracted an object mask inside each bounding box and set it as the ground truth segmentation mask. This mask is obtained by first generating multiple segments inside the bounding box, then selecting the one that maximizes a mid-level segment quality score—the output of the predictor in Carreira and Sminchisescu (2010b), from which we subtract the sum of Euclidean distance of the segment to each edge of the ground truth bounding box, as a penalty for deviation from the frame constraint.

ETHZ results are given in Fig. 11. Our method outperforms the state of the art by nearly an order of magnitude—at 0.02 FPPI (false positives per image) our detection rate is comparable with the detection rate at 0.2 FPPI in Gu et al. (2009). Comparisons between algorithms at 0.02 FPPI are shown in Table 4. We achieve 98.3%, a nearly perfect detection rate for the Swans category, at less than 0.02 FPPI.

We also evaluate the quality of our object segmentations using the ground truth segmentation masks made available by Gu et al. (2009). Following Gu et al. (2009), we report pixel average precision (AP) on each class. For each, a ROC curve is computed by varying the detection threshold on the mask scores of segments. AP is computed as the area under the curve. Comparisons with (Gu et al. 2009) in Table 3 show improvement in most classes.

Results of SvrSegm for various training conditions are shown in Fig. 12. We use three variants for the scoring function: overlap with the bounding box (named Bounding Box in the figure); overlap with automatic object mask generated from the bounding box (Automatic Overlap) and overlap with the ground truth object mask (Ground Truth). The algorithm appears to be robust to noise in the overlap measure. We also trained and tested our recognition framework using segments from Arbelaez et al. (2009) (denoted

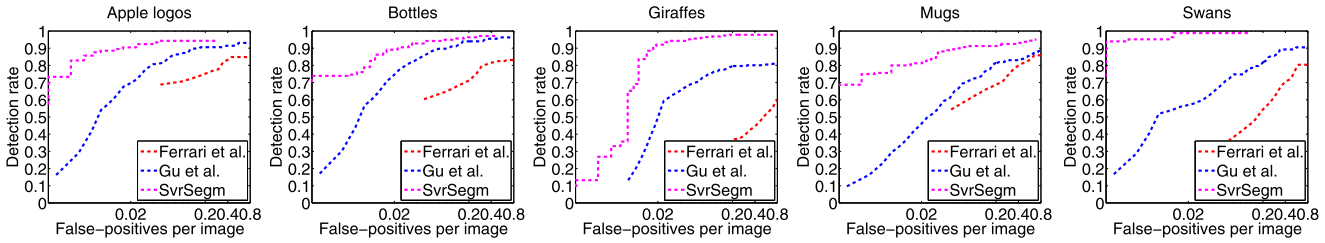

Fig. 11 Comparisons on ETHZ-Shape classes. SvrSegm is trained using only bounding box data

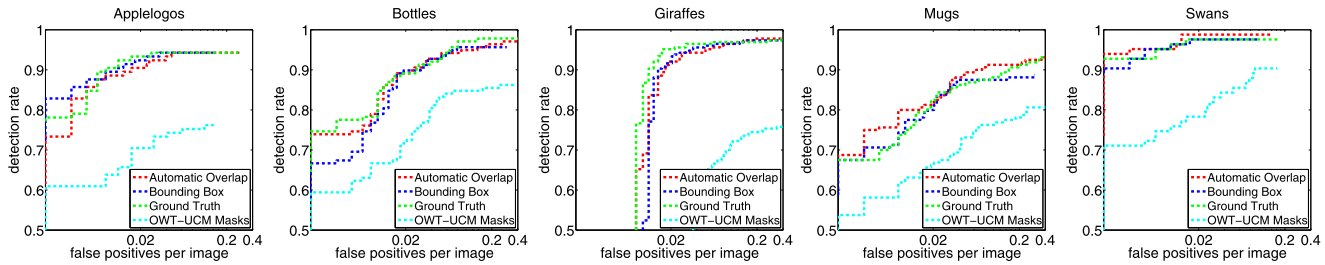

Fig. 12 Comparisons on ETHZ-Shape classes for different training conditions. SvrSegm is trained to predict overlap with object masks generated from the bounding box (Automatic Overlap), overlap with the bounding box (Bounding Box) and ground truth object masks (Ground Truth). We also both trained and tested with segments from Arbelaez et al. (2009) (OWT-UCM Masks)

Table 3 Segmentation results for ETHZ-Shape. Performance (%) is measured as pixel-wise mean AP over 5 trials, following (Gu et al. 2009)

Categories	Gu et al.	SvrSegm
Applelogos	77.2 ± 11.1	89.0 ± 1.9
Bottles	90.6 ± 1.5	90.0 ± 2.1
Giraffes	74.2 ± 2.5	75.4 ± 1.9
Mugs	76.0 ± 4.4	77.7 ± 5.9
Swans	60.6 ± 1.3	80.5 ± 2.8
Average	75.7 ± 3.2	82.5 ± 1.2

Table 4 Detection rate at 0.02 FPPI in ETHZ-Shape. SvrSegm noticeably improves on the state-of-the art in this regime

Categories	Ferrari et al.	Gu et al.	SvrSegm
Applelogos	68.83	69.75	90.48
Bottles	60.32	74.59	89.13
Mugs	46.06	54.33	81.25
Giraffes	23.75	49.63	92.07
Swans	31.60	56.98	98.31
Average	47.76	59.40	90.25

OWT-UCM Segments). We observe that this setting produces lower scores than the one obtained using CPMC segments. A possible explanation is that the OWT-UCM segments usually do not correspond to full objects but to parts and other image regions. This type of input does not appear to be effective in conjunction with our recognition framework.

7.2.3 Segmentation and Labeling: VOC 2009 and 2010

The SvrSegm algorithm was used in BONN_SVM-SEGM entry for the PASCAL VOC 2009 Challenge and the BONN_SVR_SEGM entry for the PASCAL VOC 2010 Challenge. The system was declared a winner in the VOC 2009 challenge and a joint winner of the 2010 challenge. This section describes the results obtained in these challenges, and our subsequent efforts on the VOC 2010 dataset, after the challenge, which results in the best performance reported so far for this dataset on the test set: 43.8% accuracy.

The 2009 segmentation challenge provides 1,499 images (containing 3211 objects) in the `trainval` dataset and 750 images in the hold-out `test` set to evaluate the performance of submitted algorithms. Additionally there are 5,555 images (with 14,007 objects) where only bounding box annotations are available. We did not use images with bounding box annotations at the time of the challenge, where our entry was declared as winner with an accuracy of 36.3% (in evaluating different methodologies, notice that some of the participants used these additional images to train their system (Gonfaus et al. 2010)). The results of the challenge are reproduced in Table 5. Some systems from the detection challenge have automatic entries in the segmentation challenge, since a trivial segment from the bounding box can be generated. However this often gives relatively uncompetitive results that we omit in the table.

Table 5 VOC 2009 segmentation results on the test set, for various research teams participating in the challenge. SvrSegm is the method presented in this paper

Name	SvrSegm	BROOKES MSRC	CVC	LEAR	MPI	NEC UIUC	UC3M	UCI	UCLA	UoC TTI
Mean	**36.3**	24.8	34.5	25.7	15.0	29.7	14.5	24.7	13.8	29.0
background	**83.9**	79.6	80.2	79.1	70.9	81.8	69.8	80.7	51.2	78.9
aeroplane	64.3	48.3	**67.1**	44.6	16.4	41.9	20.8	38.3	13.9	35.3
bicycle	21.8	6.7	26.6	15.5	8.7	23.1	9.7	**30.9**	7.0	22.5
bird	21.7	19.1	**30.3**	20.5	8.6	22.4	6.3	3.4	3.9	19.1
boat	**32.0**	10.0	31.6	13.3	8.3	22.0	4.3	4.4	6.4	23.5
bottle	**40.2**	16.6	30.0	28.8	20.8	27.8	7.9	31.7	8.1	36.2
bus	**57.3**	32.7	44.5	29.3	21.6	43.2	19.7	45.5	14.4	41.2
car	49.4	38.1	41.6	35.8	14.4	**51.8**	21.8	47.3	24.3	50.1
cat	**38.8**	25.3	25.2	25.4	10.5	25.9	7.7	10.4	12.1	11.7
chair	5.2	5.5	5.9	4.4	0.0	4.5	3.8	4.8	6.4	**8.9**
cow	**28.5**	9.4	27.8	20.3	14.2	18.5	7.5	14.3	10.3	**28.5**
diningtable	22.0	**25.1**	11.0	1.3	17.2	18.0	9.6	8.8	14.5	1.4
dog	19.6	13.3	23.1	16.4	7.3	**23.5**	9.5	6.1	6.7	5.9
horse	33.6	12.3	**40.5**	28.2	9.3	26.9	12.3	21.5	9.7	24.0
motorbike	45.5	35.5	**53.2**	30.0	20.3	36.6	16.5	25.0	23.6	35.3
person	33.6	20.7	32.0	24.5	18.2	34.8	16.4	**38.9**	20.0	33.4
pottedplant	27.3	13.4	22.2	12.2	6.9	8.8	1.5	14.8	2.3	**35.1**
sheep	**40.4**	17.1	37.4	31.5	14.1	28.3	14.2	14.4	12.6	27.7
sofa	18.1	18.4	**23.6**	18.3	0.0	14.0	11.0	3.0	12.3	14.2
train	33.6	37.5	**40.3**	28.8	13.2	35.5	14.1	29.1	17.0	34.1
tv/monitor	**46.1**	36.4	30.2	31.9	13.2	34.7	20.3	45.5	13.2	41.8

After the challenge, we have also exploited bounding box annotations crudely (only one segment which best overlaps the bounding box is used, with overlap value always set to 0.8) to produce the slightly improved 37.24% accuracy reported in Li et al. (2010a). This result is not included in this paper because the methodology is slightly different, but see our work in Li et al. (2010a) for details.

As described in Sect. 7.1, in the 2010 segmentation challenge, the `trainval` set is augmented to 1,928 images (with 4,203 objects) and the `test` set is augmented to 964 images. An additional 8,175 images (containing 19,171 objects) have only bounding box annotations. This approach was one of the joint winners with an accuracy of 39.7%. The version we submitted to the challenges was trained only based on segmentation annotation and without taking advantage of the information in the additional images that contain only bounding box annotations. After the challenge we included those additional images in the training set. For each ground truth object, we selected the 10 segments whose bounding-box had the best IOU-overlap with the object bounding box, and set those overlap values as desired outputs. With this additional training data, we obtain a further 4% performance improvement on VOC 2010, resulting in 43.8%. To our knowledge this is the best result reported on this dataset so far. Table 6 provides details.

Figure 13 illustrates some successfully segmented images from the VOC test set. It can be seen that our method handles background clutter, partially occluded objects, objects with low contrast with the background, as well as multiple objects in the same image. The first two images shown in the last row have particularly low contrast—the sheep in the first image or the black suit of the child in the second one are almost the same color as the background. Our approach nevertheless succeeds in identifying the correct spatial support of those objects and also predicts their category correctly.

However, despite our moderate success, the performance on the VOC dataset remains at around 44%, which means there is still substantial room for improvement. In order to gain intuition on directions for future development, we also show images where the method fails. Figure 14 shows images that illustrate various types of failure. We partition the errors into 4 groups. In group 1, errors come from the inability to correctly select segments. Usually, the segments selected by the algorithm are to some degree intuitive. For instance, in the last image where we classified a segment as boat, a background segment shaped as a boat was selected

Table 6 VOC 2010 segmentation results on the test set. For our method, SvrSegm, models trained both *with* and *without* additional bounding box data and images from the training set for object detection are shown (WITH DET and W/O DET, respectively)

Name	SvrSegm WITH DET	SvrSegm W/O DET	BROOKES	CVC	STANFORD	UC3M	UOCTTI
Mean	**43.8**	39.7	30.3	40.1	29.1	27.8	31.8
background	**84.6**	84.2	70.1	81.1	80.0	73.4	80.0
aeroplane	**59.0**	52.5	31.0	58.3	38.8	45.9	36.7
bicycle	**28.0**	27.4	18.8	23.1	21.5	12.3	23.9
bird	**44.0**	32.3	19.5	39.0	13.6	14.5	20.9
boat	35.5	34.5	23.9	**37.8**	9.2	22.3	18.8
bottle	**50.9**	47.4	31.3	36.4	31.1	9.3	41.0
bus	**68.0**	60.6	53.5	63.2	51.8	46.8	62.7
car	53.5	54.8	45.3	**62.4**	44.4	38.3	49.0
cat	**45.6**	42.6	24.4	31.9	25.7	41.7	21.5
chair	**15.3**	9.0	8.2	9.1	6.7	0.0	8.3
cow	**40.0**	32.9	31.0	36.8	26.0	35.9	21.1
diningtable	**28.9**	25.2	16.4	24.6	12.5	20.7	7.0
dog	33.5	27.1	15.8	29.4	12.8	**34.1**	16.4
horse	**53.1**	32.4	27.3	37.5	31.0	34.8	28.2
motorbike	53.2	47.1	48.1	**60.6**	41.9	33.5	42.5
person	37.6	38.3	31.1	**44.9**	44.4	24.6	40.5
pottedplant	35.8	**36.8**	31.0	30.1	5.7	4.7	19.6
sheep	48.5	**50.3**	27.5	36.8	37.5	25.6	33.6
sofa	**23.6**	21.9	19.8	19.4	10.0	13.0	13.3
train	39.3	35.2	34.8	**44.1**	33.2	26.8	34.1
tv/monitor	42.1	40.9	26.4	35.9	32.3	26.1	**48.5**

and wrongly labeled as boat. In turn, the aircraft is also quite hard to detect since it is small and almost entirely occluded.

In the second failure group, the algorithm does not successfully handle multiple interacting objects, such as men on motorbike. These types of images are difficult to segment purely bottom-up because of the complicated patterns of mutual occlusion between objects. It might also be, in part, a problem of the current post-processing method, whose sequential nature (fix a mask before considering the next one) does not always allow for a joint analysis of multiple segments and categories. We have recently developed alternative formulations to address some of these issues (Carreira et al. 2010; Ion et al. 2011).

The third failure group illustrates errors in classification. Currently, confusions mostly arise between a few relatively similar category pairs: cow–horse, dog–cat, dog–horse, dog–sheep, other tables (which are labeled as background in the challenge)–dining table, sofa–chair, and TV/Monitor–other similar shaped objects (e.g., windows, glasses on doors). Otherwise, if a segment is correctly recovered, it is usually correctly classified. Considering the relatively small training set, we believe that such errors are not very problematic in the long run, as more training data becomes available.

The fourth failure group shows that it is sometimes difficult for the method to determine the proper spatial extent of objects. This can happen when parts of objects are recovered (the table and the bottle in the group), an overly large segment contains the object (the sofa and the bird in the group) or reflections occur (the boat in the group).

It is also worth mentioning that because normal tables are not classified as dining tables in the VOC dataset, the trained dining table classifier mainly looks for dishes, plates, glasses and other stuff on the table, instead of the table itself. This annotation may just be too fine-grained considering the dataset size and distribution. At the same time it is to some degree ambiguous as in principle almost any table can be used as a dining table.

8 Conclusion

We have described a semantic image interpretation framework based on a novel front end algorithm, CPMC (Carreira and Sminchisescu 2010b), that generates multiple figure-ground segmentations, followed by sequential object labeling. Unlike previous methods that rely on classification, we frame recognition as a regression problem of estimating the

Fig. 13 Successful semantic segmentations produced by our method on the VOC test set. Notice that the object boundaries are relatively accurate and that our method can handle partial views and background clutter

Fig. 14 Failure modes of our semantic segmentation on the VOC testset, split into four groups. See text for discussion

spatial overlap of generated segments with the target object of the desired category. Instead of selecting only one segment, we produce a ranking in the space of all putative segments based on spatial overlap. This makes it possible to bet-

ter exploit segments that partially overlap the ground truth in order to consolidate recognition. We demonstrate state-of-the-art results in image classification, object detection and semantic segmentation in Caltech-101, ETHZ-Shapes and PASCAL VOC 2009 and VOC 2010. Our approach is dominantly bottom-up: object class knowledge is used only after plausible object segmentations have been obtained. In the long run, a closer integration of top-down information could improve performance. In this work, however, we make a case that bottom-up modules that extract object-level segments beyond superpixels can achieve good performance. They are a plausible front-end for both segmentation and recognition tasks.

Acknowledgements This work was supported, in part, by the European Commission, under MCEXT-025481, and by CNCSIS-UEFISCU, under project number PN II-RU-RC-2 / 2009.

References

Arbelaez, P., & Cohen, L. (2008). Constrained image segmentation from hierarchical boundaries. In *Computer vision and pattern recognition, IEEE computer society conference on* (pp. 1–8).

Arbelaez, P., Maire, M., Fowlkes, C., & Malik, J. (2009). From contours to regions: an empirical evaluation. In *IEEE conference on computer vision and pattern recognition*.

Bishop, C. M. (2007) Pattern recognition and machine learning *Information science and statistics*, 1st edn, 2006. Springer, Berlin corr. 2nd printing edn.

Blaschko, M. B., & Lampert, C. H. (2008). Learning to localize objects with structured output regression. In *European conference on computer vision* (pp. 2–15).

Bo, L., & Sminchisescu, C. (2009). Efficient match kernels between sets of features for visual recognition. In *Advances in neural information processing systems*.

Boiman, O., Shechtman, E., & Irani, M. (2008). In defense of nearest-neighbor based image classification. In *Computer vision and pattern recognition, IEEE conference on CVPR* 2008 (pp. 1–8).

Borenstein, E., & Ullman, S. (2002). Class-specific, top-down segmentation. In *European conference on computer vision*.

Borenstein, E., & Ullman, S. (2008). Combined top-down/bottom-up segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *30*(12), 2109–2125.

Bosch, A., Zisserman, A., & Munoz, X. (2007). Representing shape with a spatial pyramid kernel. In *CIVR'07*.

Boykov, Y., & Jolly, M. P. (2001). Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV* (pp. 105–112).

Carreira, J., & Sminchisescu, C. (2010a). Constrained parametric min-cuts for automatic object segmentation, release 1. http://sminchisescu.ins.uni-bonn.de/code/cpmc/.

Carreira, J., & Sminchisescu, C. (2010b). Constrained parametric min cuts for automatic object segmentation. In *IEEE conference on computer vision and pattern recognition*.

Carreira, J., & Sminchisescu, C. (2012). Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transaction on Pattern Analysis and Machine Intelligence* (accepted).

Carreira, J., Ion, A., & Sminchisescu, C. (2010). *Image segmentation by discounted cumulative ranking on maximal cliques* (Tech. Rep.). 06-2010 (arXiv:1009.4823), Computer Vision and Machine Learning Group, Institute for Numerical Simulation, University of Bonn. Available at http://arxiv.org/abs/1009.4823.

Cour, T., & Shi, J. (2007). Recognizing objects by piecing together the segmentation puzzle. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Csurka, G., & Perronnin, F. (2008). A simple high performance approach to semantic segmentation. In *BMVC*.

Csurka, G., & Perronnin, F. (2010). An efficient approach to semantic segmentation. *International Journal of Computer Vision* 1–15.

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, *88*(2), 303–338.

Fei-Fei, L., Fergus, R., & Perona, P. (2007). Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. *Computer Vision and Image Understanding*, *106*(1), 59–70.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*, 1627–1645.

Ferrari, V., Jurie, F., & Schmid, C. (2007). Accurate object detection with deformable shape models learnt from images. In *IEEE conference on computer vision and pattern recognition*.

Fulkerson, B., Vedaldi, A., & Soatto, S. (2009). Class segmentation and object localization with superpixel neighborhoods. In *International conference on computer vision* (pp. 670–677).

Gallo, G., Grigoriadis, M. D., & Tarjan, R. E. (1989). A fast parametric maximum flow algorithm and applications. *SIAM Journal on Computing*, *18*(1), 30–55. doi:10.1137/0218003.

Gehler, P. V., & Nowozin, S. (2009). On feature combination for multiclass object classification. In *International conference on computer vision*.

Gonfaus, J., Boix, X., de Weijer, J. V., Bagdanov, A., Serrat, J., & Gonzàlez, J. (2010). Harmony potentials for joint classification and segmentation. In *IEEE conference on computer vision and pattern recognition*.

Gould, S., Fulton, R., & Koller, D. (2009a). Decomposing a scene into geometric and semantically consistent regions. In *International conference on computer vision*.

Gould, S., Gao, T., & Koller, D. (2009b). Region-based segmentation and object detection. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams & A. Culotta (Eds.), *Advances in neural information processing systems* (pp. 655–663).

Grauman, K., & Darrell, T. (2005). The pyramid match kernel: discriminative classification with sets of image features. In *International conference on computer vision* (Vol. 2, pp. 1458–1465).

Griffin, G., Holub, A., & Perona, P. (2007). *Caltech-256 object category dataset* (Tech. Rep. 7694). California Institute of Technology.

Gu, C., Lim, J. J., Arbeláez, P., & Malik, J. (2009). Recognition using regions. In *IEEE conference on computer vision and pattern recognition*.

He, X., Zemel, R. S., & Carreira-Perpiñán, M. (2004). Multiscale conditional random fields for image labeling. *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 695–702).

Ion, A., Carreira, J., & Sminchisescu, C. (2011). Image segmentation by figure-ground composition into maximal cliques. In *International conference on computer vision*.

Kohli, P., Ladicky, L., & Torr, P. (2008). Robust higher order potentials for enforcing label consistency. In *IEEE conference on computer vision and pattern recognition* (pp. 1–8).

Kumar, A., & Sminchisescu, C. (2007). Support kernel machines for object recognition. In *International conference on computer vision*.

Kumar, M. P., Torr, P. H. S., & Zisserman, A. (2005). Obj cut. In *IEEE conference on computer vision and pattern recognition*.

Ladicky, L., Russell, C., Kohli, P., & Torr, P. H. S. (2009a). Associative hierarchical crfs for object class image segmentation. In *International conference on computer vision*.

Ladicky, L., Russell, C., Kohli, P., & Torr, P. H. S. (2009b). Associative hierarchical crfs for object class image segmentation. In *International conference on computer vision*.

Ladicky, L., Sturgess, P., Alaharia, K., Russel, C., & Torr, P. H. (2010). What, where & how many ? combining object detectors and crfs. In *European conference on computer vision*.

Lampert, C., Blaschko, M., & Hofmann, T. (2008). Beyond sliding windows: object localization by efficient subwindow search. In *Computer vision and pattern recognition. IEEE conference on CVPR* 2008 (pp. 1–8).

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, pp. 2169–2178).

Leibe, B., Leonardis, A., & Schiele, B. (2008). Robust object detection with interleaved categorization and segmentation. *International Journal of Computer Vision*, *77*(1–3), 259–289.

Levin, A., & Weiss, Y. (2009). Learning to combine bottom-up and top-down segmentation. *International Journal of Computer Vision*, *81*(1), 105–118.

Li, F., Carreira, J., & Sminchisescu, C. (2010a). Object recognition as ranking holistic figure-ground hypotheses. In *IEEE conference on computer vision and pattern recognition*.

Li, F., Ionescu, C., & Sminchisescu, C. (2010b). Random Fourier approximations for skewed multiplicative histogram kernels. In *Annual symposium of the German association for pattern recognition (DAGM)*.

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

Maire, M., Arbelaez, P., Fowlkes, C., & Malik, J. (2008). Using contours to detect and localize junctions in natural images. In *IEEE conference on computer vision and pattern recognition*.

Malisiewicz, T., & Efros, A. (2007). Improving spatial support for objects via multiple segmentations. In *British machine vision conference*.

Malisiewicz, T., & Efros, A. A. (2008). Recognition by association via learning per-exemplar distances. In *IEEE conference on computer vision and pattern recognition*.

Mori, G., Ren, X., Efros, A., & Malik, J. (2004). Recovering human body configurations: combining segmentation and recognition. In *Computer vision and pattern recognition. Proceedings of the 2004 IEEE computer society conference on CVPR 2004* (Vol. 2, pp. II-326–II-333).

Pantofaru, C., Schmid, C., & Hebert, M. (2008). Object recognition by integrating multiple image segmentations. In *European conference on computer vision*.

Pinto, N., Cox, D. D., & DiCarlo, J. J. (2008). Why is real-world visual object recognition hard? *PLoS Computational Biology 4*(1), e27.

Rabinovich, A., Belongie, S., Lange, T., & Buhmann, J. M. (2006). Model order selection and cue combination for image segmentation. In *IEEE conference on computer vision and pattern recognition* (Vol. 1, pp. 1130–1137).

Rabinovich, A., Vedaldi, A., & Belongie, S. (2007). *Does image segmentation improve object categorization*? (Tech. Rep.). CS2007-090.

Rahimi, A., & Recht, B. (2007). Random features for large-scale kernel machines. In *Advances in neural information processing systems*.

Schoenemann, T., & Cremers, D. (2010). A combinatorial solution for model-based image segmentation and real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*, 1153–1164.

Shi, J., & Malik, J. (2000) Normalized cuts and image segmentation. *IEEE Transaction on Pattern Analysis and Machine Intelligence*. doi:10.1109/34.868688.

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European conference on computer vision* (pp. 1–15).

Shotton, J., Winn, J., Rother, C., & Criminisi, A. (2009). Textonboost for image understanding: multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, *81*, 2–23.

Srinivasan, P., & Shi, J. (2007). Botom-up recognition and parsing of the human body. In *IEEE conference on computer vision and pattern recognition*.

Todorovic, S., & Ahuja, N. (2008). Learning subcategory relevances for category recognition. In *IEEE conference on computer vision and pattern recognition*.

Toshev, A., Taskar, B., & Daniilidis, K. (2010). Object detection via boundary structure segmentation. In *IEEE conference on computer vision and pattern recognition* (pp. 950–957).

Tsochantaridis, I., Hofmann, T., Joachims, T., & Altun, Y. (2004). Support vector machine learning for interdependent and structured output spaces. In *Proceedings of the international conference of machine learning*.

Tukey, J. W. (1977). *Exploratory data analysis*. Addison-Wesley: Reading.

van de Sande, K. E. A., Gevers, T., & Snoek, C. G. M. (2010). Evaluating color descriptors for object and scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *9*, 1582–1596.

Vedaldi, A., & Zisserman, A. (2010). Efficient additive kernels via explicit feature maps. In *IEEE conference on computer vision and pattern recognition*.

Vedaldi, A., Gulshan, V., Varma, M., & Zisserman, A. (2009). Multiple kernels for object detection. In *International conference on computer vision*.

Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In *IEEE conference on computer vision and pattern recognition*.

Yang, Y., Hallman, S., Ramanan, D., & Fowlkes, C. (2010). Layered object detection for multi-class segmentation. In *IEEE conference on computer vision and pattern recognition*.

Yu, H. F., Hsieh, C. J., Chang, K. W., & Lin, C. J. (2010). Large linear classification when data cannot fit in memory. In *ACM SIGKDD conference on knowledge discovery and data mining*.

Yu, S. X., & Shi, J. (2003). Object-specific figure-ground segregation. In *IEEE conference on computer vision and pattern recognition* (Vol. 2, p. 39).

Zhang, H., Berg, A., Maire, M., & Malik, J. (2006). Svm-knn: discriminative nearest neighbor classification for visual category recognition. In *Computer vision and pattern recognition. IEEE computer society conference on* (Vol. *2*, pp. 2126–2136).