Learning to Group Objects

Learning to Group Objects

Victoria Yanulevskaya Jasper Uijlings Nicu Sebe
University of Trento, Italy
yanulevskaya|sebe@disi.unitn.it, jrr.uijlings@ed.ac.uk

Abstract

This paper presents a novel method to generate a hypothesis set of class-independent object regions. It has been shown that such object regions can be used to focus computer vision techniques on the parts of an image that matter most leading to significant improvements in both object localisation and semantic segmentation in recent years. Of course, the higher quality of class-independent object regions, the better subsequent computer vision algorithms can perform. In this paper we focus on generating higher quality object hypotheses. We start from an oversegmentation for which we propose to extract a wide variety of region-features. We group regions together in a hierarchical fashion, for which we train a Random Forest which predicts at each stage of the hierarchy the best possible merge. Hence unlike other approaches, we use relatively powerful features and classifiers at an early stage of the generation of likely object regions. Finally, we identify and combine stable regions in order to capture objects which consist of dissimilar parts. We show on the PASCAL 2007 and 2012 datasets that our method yields higher quality regions than competing approaches while it is at the same time more computationally efficient.

1. Introduction

Finding and identifying objects within an image is very challenging not only because of the large variety in appearance of objects, but also due to the staggering number of possible locations an object can occupy. Humans have the remarkable ability to quickly find coherent regions in an image which surely facilitates recognition. Recently, impressive progress has been made in computer vision on finding sets of coherent regions that accurately cover objects [6, 10, 20, 23]. By reducing the set of possible interesting locations and providing better object boundaries, these works facilitate the use of more expensive and advanced computer vision techniques on the regions of an image that matter most, giving rise to substantial improvements in object localisation and semantic segmentation [2, 7, 14, 5, 20]. In this paper our goal is to improve the generation of class-independent object regions.

Our work is illustrated in Figure 1 and rests on four pillars: (1) We use a greedy, bottom-up hierarchical grouping algorithm, as is done in [20]. This contrasts with [6, 10] who use a graph based framework with randomly sampled foreground and background seeds to generate a variety of segments. While the graph-based models produce globally optimal segmentations, the sizes of the segments are dependent on a parameter, which is varied to create segments of all scales. Instead, using all regions in a hierarchical grouping algorithm naturally captures all possible scales without any need for parameter selection and optimization. Furthermore, region features can be propagated, making texture/colour measurements more reliable as regions grow.

(2) We use relatively strong regional features at an early stage while generating object hypotheses. This contrasts with [6, 10] who rely on simple features to generate a first set of candidates, only to use more advanced features in the filtering stage. By using relatively strong regions features from the beginning we can much more accurately and reliably generate segments without needing filtering.

(3) We use Random Forests to learn which segments should be grouped together at each stage of the hierarchy. This contrasts with [20] who foregoes learning altogether. While [6, 10] do use learning in both their segment generation and filtering stages, they combine it with more powerful features only at the filtering stage.

(4) We identify "stable" regions and consider combinations of such regions in order to capture objects which consist of dissimilar parts. Many objects consist of parts which, taken individually, bear no resemblance to each other. For example, the wheels of vehicles are completely different in appearance than the rest of the vehicle. The head of a person is completely dissimilar to the clothes he/she wears. The works of [6, 10] rely on random seeds to combine dissimilar parts into a single region, while [20] cannot find such objects. In this paper we identify possible object parts and merge adjacent ones to find objects consisting of dissimilar parts.

Experiments on the Pascal 2007 and 2012 datasets show

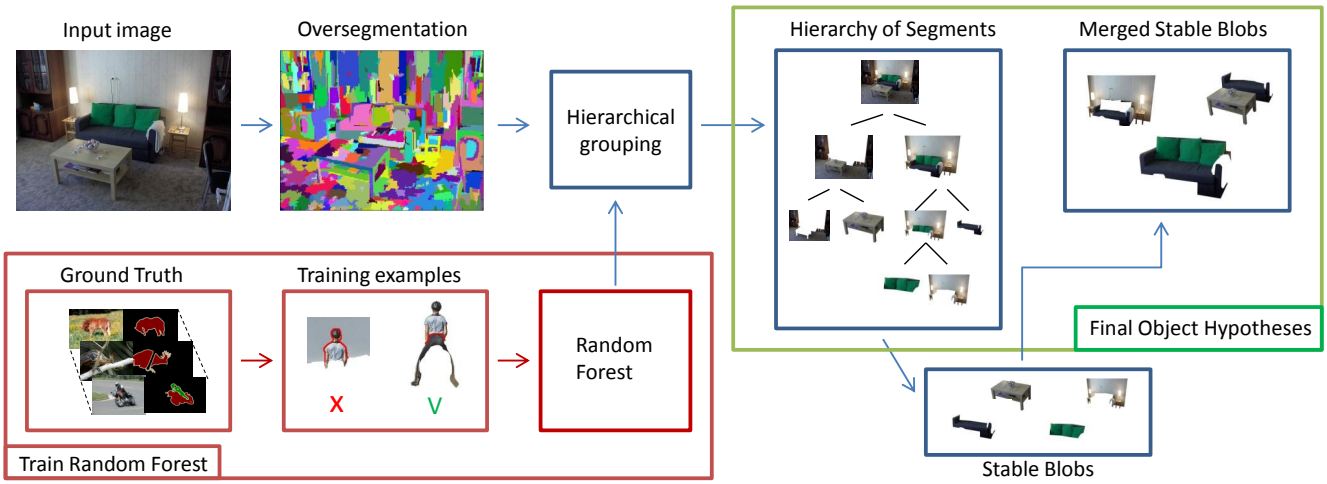

Figure 1. The proposed framework for generating class-independent object hypotheses. We start with an oversegmentation. Then we greedily group segments together in a hierarchical fashion where we use a pre-trained Random Forest classifier to determine at each stage of the hierarchy which regions should be merged. Afterwards we identify stable blobs and merge all adjacent pairs. Merging stable blobs allows for the discovery of objects which consist of visually dissimilar parts.

that our work significantly outperforms [20] in terms of quality and quantity of extracted regions, and significantly outperforms [6, 10] in terms of quality and computational efficiency.

2. Related work

Uijlings *et al.* [20] propose a greedy, bottom-up hierarchical grouping algorithm to generate a small set of class independent object locations. The main idea of their approach is to use a diverse set of strategies in order to find all possible objects. They show that they can generate a small set of high quality bounding boxes, which are successfully used in a powerful localisation system based on Bag-of-Words [9, 19]. In this paper we also start from a greedy hierarchical grouping algorithm but differ from [20] in three important ways: (1) We rely on machine learning to predict which regions should be merged rather than using a simple combination of similarities. (2) We use a much larger and more powerful set of region features and measurements. (3) Our work is able to find objects composed of dissimilar parts by combining stable regions.

Both Carreira and Sminchisescu [6] and Endres and Hoiem [10] have similar pipelines: first, a large pool of figure-ground binary segmentations is generated, which is afterwards ranked to suppress redundant and unlikely object hypotheses. Specifically, Carreira and Sminchisescu [6] use a segmentation algorithm which solves many binary graph-cuts optimization problems for different initializations. As similarity measures colour distribution and contour energy of globalPb [3] are used. Furthermore, the authors train a random regression forest to rank generated segments. Here the authors consider a large set of features which are inherent to an object, such as contrast with the background, convexity, smooth contours, alignment of contours with image edges, and the location in the image. The authors successfully use their regions in a subsequent semantic segmentation process [17]. Endres and Hoiem [10] first generate hierarchical segmentation with agglomerative grouping based on energy of occlusion boundaries and figure-ground likelihood [16]. Afterwards, generated regions are used as seeds for a figure-ground Conditional Random Fields segmentation. The latter uses colour distribution, texture distribution, and an affinity function as similarity measures. At the final stage, Endres and Hoiem use the slack-rescaled method with loss penalty to rank generated object hypothesis. Here they describe the appearance of a region with the following features: alignment of contours with occlusion boundaries, contrast with the background, geometrical location within an image, and the probability of being 'stuff' like grass or sky. In contrast to [6, 10], we do not use a re-ranking phase. Instead, we rely on powerful region features and machine learning at an early stage to generate a small set of object hypotheses.

More recently, Weiss *et al.* [23] proposed a method for generating potential object regions in three steps: First they generate bounding boxes using two class-independent methods [3, 18] and the well-known class-specific part-based model [12]. Then they use class-specific shape priors within the bounding boxes to get a rough shape estimate, after which superpixels from an oversegmentation are used to generate the final regions. While they report improvements over [6, 10], the generated regions are highly class specific, which may make the approach of [23] suitable for better semantic segmentation, but not for generating a class-independent set of object hypotheses which is the aim of [6, 10, 20] and the work we present here.

Several methods aim to generate class-independent object *windows*. Alexe *et al.* [1] introduce a generic objectness measure based on three object characteristics: well de-

fined contours, contrast to the surrounding, and uniqueness within an image. They randomly sample a large amount of windows, where the windows with the highest objectness measure are served as object hypotheses. They successfully use their windows to speed up the part-based localisation method of [12]. Rahtu *et al.* [18] build on [1] and propose better objectness cues as well as an improved initial sampling strategy based on the non-generic Pascal VOC object location distribution. Bounding boxes often contain large parts of the background which may be undesirable when extracting object features. In this paper, we focus on sampling image regions of arbitrary shape to better capture the exact location of an object.

3. Method

Our method starts from an oversegmentation produced by the algorithm by Felzenszwalb and Huttenlocher [13]. This results in an oversegmentation whose segments (ideally) do not cover different objects yet are large enough to extract region-based features. We propose to extract a wide variety of region-based features which are detailed in section 3.1. We greedily and hierarchically group adjacent regions together until there is a single segment covering the complete image. To determine which regions should be merged, we train a Random Forest classifier as detailed in section 3.2. Finally, some objects consist of parts which are visually dissimilar, hence a visual grouping strategy can only find the parts of such objects. We address this by identifying stable regions likely to contain parts and merging these, as explained in section 3.3.

3.1. Region-based Features

We propose to extract a wide variety of region-features. We include the features and measurements proposed in [20] to facilitate evaluating the impact of our region-based feature set. These are defined as [20]:

Colour similarity. The histogram intersection between colour histograms, which use 25 bins per colour channel for a 75-dimensional histogram. Whereas [20] uses only the colour space of the initial oversegmentation, we use always four different colour spaces: Lab, HSV, Opp (Opponent colour space [15]), and rgI (normalized red, normalized green, and intensity).

Texture similarity. The histogram intersection between texture histograms of the regions. For texture histograms for each colour channel its gradient responses in 8 orientations are calculated using a Gaussian derivative filter with $\sigma = 1$. Per channel per orientation a 10-dimensional histogram is created, for a 240 dimensional histogram. We use Lab, HSV, Opp, and rgI texture histograms.

Fill. The area of the combination of two regions divided by the tightly fitting bounding box around this area.

Size. The area of the combination of two regions divided by the size of an image.

Additionally, we include the Image Patch Exemplars of Varma and Zisserman [21] and a range of features based on contours and borders, where we define a contour as being the boundary of a complete region, and a border as the boundary *between two* regions. Specifically, we define the similarity measures between region a and b which would combine into region c as follows:

Image Patch Exemplar Similarity [21]. We sample 3x3 raw image patches at every pixel which can be represented as 27-dimensional vectors. We train a visual vocabulary on the Pascal VOC 2007 training set using the VLFeat hierarchical k-means [22] with depth 2 and 64 splits per level, resulting in 4096 visual words. Similarity is measured by the histogram intersection between normalised frequency histograms of the visual words. We always use the colour spaces Lab, HSV, Opp, and rgI.

Insideness. measures how well one region fits into another region. If two regions touch at a single point, together they can hardly form an object. On the other hand, if their border is at the same time a contour of one region, combining them fills up a hole in the other region. Generally, high insideness tends to yield convex regions.

$$\text{insideness}(a,b) = \frac{\text{borderSize}(a,b)}{\min(\text{contourSize}(a), \text{contourSize}(b))} \quad (1)$$

ContourEdgeR measures edge response along contours, as objects are expected to have clear object boundaries:

$$\text{contourEdgeR}(a,b) = \frac{\text{edgeResponseContour}(c)}{\text{contourSize}(c)}. \quad (2)$$

$\text{edgeResponseContour}(c)$ is the sum of edge responses over all pixels of the contour, where an edge response is the maximum edge response over all colour channels of all four colour spaces (HSV, Lab, Opp, rgI).

BorderEdgeR is one minus the average edge response of the border of the segments:

$$\text{borderEdgeR}(a,b) = 1 - \frac{\text{edgeResponseBorder}(a,b)}{\text{borderSize}(a,b)}. \quad (3)$$

$\text{edgeResponseBorder}(a,b)$ is sum of edge responses over all pixels of the border between segment a and b. This measure reflects that two segments which have a strong edge between them are unlikely to belong to the same object.

Border smoothness measures the total amount of change in direction of the border between segment a and b. If this border is smooth, it likely coincides with a real object boundary. In contrast, adjacent regions in high textured areas such as grass often have irregular borders.

We first obtain the pixel coordinates of the border which we smooth using a Gaussian derivative filter ($\sigma = 2$) to remove artefacts caused by the pixel grid. On each inflection

point (i.e. the point where the border changes direction) we measure the amount of direction change with respect to the previous inflection point, where we ignore direction changes smaller than 20 degrees. We then convert this measure with a sigmoid function to a value between 0 and 1. If the border is discontinuous, we set the similarity to zero.

Contour compactness is the ratio between the length of a circle having the same area as the combined region c and the size of the contour of the combined region:

$$\text{contourCompactness}(a,b) = \frac{(2\sqrt{\pi \times \text{area}(c)})}{\text{contourSize}(c)}. \quad (4)$$

This measure reflects that most objects are compact. For a circle the value is 1, while for highly elongated regions and regions with rough contours this ratio is close to zero.

Border compactness is a ratio of the distance between end points of the border between regions to the length of the border. When the border is discontinuous we define the similarity to be zero. This measure gives high similarity to borders which are incompact and therefore unlikely object borders. Notice that smooth, highly curved borders may be highly incompact. Therefore, compactness and smoothness are complimentary measures, although their both punish rough curves.

To be computationally efficient, we defined all the contour/border similarities above in such a way that we can propagate the features they use through the hierarchy: the exact border paths, the size of the borders and contours, and the sum of the edge responses of the borders and contours. Therefore, we only have to determine these features once for the segments of the initial oversegmentation. As before, let segments a and b merge into segment c. If there is a border between a and d $\text{border}(a,d)$ but not between b and d, then $\text{border}(c,d) = \text{border}(a,d)$. However, when there is also a border between b and d, then $\text{border}(c,d) = \text{border}(a,d) \cap \text{border}(b,d)$, where one has find where the borders touch and if the new border is continuous or not. The border size resulting from a merge is simply the sum of the old borders: $\text{borderSize}(c,d) = \text{borderSize}(a,d) + \text{borderSize}(b,d)$. This is the same for edgeResponseborder. As for the contours, both the size and the edgeResponses can be calculated in the same way:

$$\begin{aligned}\text{contourSize}(c) =& \text{contourSize}(a) + \text{contourSize}(b) \\ & - 2 * \text{borderSize}(a,b). \quad (5)\\ \text{edgeResponseContour}(c) =& \text{edgeResponseContour}(a) + \\ & \text{edgeResponseContour}(b) \\ & - 2 * \text{edgeResponseBorder}(a,b). \quad (6)\end{aligned}$$

3.2. Learning with random forests

We want to learn the best combination of the similarity measures defined above. As the similarity measures are all of a different nature, we choose to use Random Forests [4, 8] as they are insensitive to this aspect.

To obtain learning examples, we first apply for all training images a grouping strategy without doing learning but by simply averaging similarities. For all resulting segments, we measure their precision: i.e. the fraction of the area of the segment which is part of a single ground truth object. Ideally, we want to merge segments which belong to the same object. Therefore, as positive examples we consider pairs of adjacent segments who both had a precision (within the same object) higher than a threshold t. As negative examples, we want to penalize pairs where one segment is inside an object and the other segment is not. Hence we consider pairs where one segment has a precision higher than t, and the other a precision lower than t. We use the same treshold $t = 0.8$ for both positive and negative examples. Note that in an initial test we also tried regression rather than classification, but this gave much lower results.

In our experiments we obtained around 450,000 positive and 140,000 negative training examples. We train a random forest consisting of 50 trees, which we found to perform well. For each tree, we use 66% of the training data to construct the tree (bagging), where in each node we perform 10 random splits and take the one with the highest information gain. We stop when either (1) a maximum depth of 15 is reached or (2) when there are less than 5 examples in a leaf. All the training data is used to set the prediction probabilities of the leaf nodes.

3.3. Merging stable regions

As we use a greedy, hierarchical grouping method, the algorithm always tries to merge the most similar regions. However, sometimes objects consists of completely dissimilar parts such as a human head and its body covered by clothes, or a sail and a boat together forming a sail-boat. In order to be able to deal with such objects, we propose to find "stable" regions as a proxy for object parts and merge adjacent stable regions.

Intuitively, the most obvious way to find stable regions is to use a centre-surround measurement on each region. While this works, we found a more computationally efficient way specific to our method which works as good or better. Quite often, segments of a homogeneous region slowly grow by merging with relatively small regions of the same appearance. Once the segment covers a complete homogeneous region, it tends to merge with relatively big regions. Hence, we can identify stable blobs by relatively big jumps in size. In our experiments blobs are considered "stable" when their parent in the hierarchy is more than 25% bigger. In practice this means we consider fewer than 50% of the blobs to be stable. Once we identified the stable blobs, we take all combinations of adjacent stable blobs and add these to our set of object hypotheses.

Table 1. Evaluating our novelties with respect to Selective Search [20] on the segmentation part of PASCAL VOC 2007 test set.

#	Method	Number of Hierarchies	k	Colour Spaces	Features	Random Forest	Merge Stable	MABO	Regions
1	Fast SS [20]	8	50,100	Lab, HSV	C+T+S+F, T+S+F	0	0	0.739	4,753
2	Quality SS [20]	80	50,100, 150,300	Lab, HSV, rgI, H, I	C+T+S+F, T+S+F, F, S	0	0	0.818	23,418
3	Single Hierarchy SS [20]	1	50	Lab	C+T+S+F	0	0	0.598	920
5	Single Hierarchy SS + RF	1	50	Lab	C+T+S+F	1	0	0.602	920
4	All Features this paper	1	50	Lab	4C+4T+4PE+F+S+I+2Co+2E+Sm	0	0	0.622	920
6	All Features + RF	1	50	Lab	4C+4T+4PE+F+S+I+2Co+2E+Sm	1	0	0.675	920
7	All Features + Stable	1	50	Lab	4C+4T+4PE+F+S+I+2Co+2E+Sm	0	1	0.718	4,416
8	All Features + RF + Stable	1	50	Lab	4C+4T+4PE+F+S+I+2Co+2E+Sm	1	1	0.749	2,868

4. Results

We consider two datasets: (1) the segmentation part of PASCAL VOC 2007 test set, and (2) the segmentation part of PASCAL VOC 2012 validation set [11]. We train random forests using the training and validation parts of 2007 only. As the performance measure we use the Mean Average Best Overlap (MABO) and the number of generated regions. Overlap is defined as the intersection of two regions divided by their union, for which we use the code as provided by the the Pascal VOC challenge [11]. Best Overlap is the score of the best overlapping segment with the ground truth of a single object instance. Average Best Overlap is the average over all BO scores of the instances of a single class. The final Mean Average Best Overlap is the average ABO over all 20 Pascal classes. MABO [20] is also used as evaluation measure under different names in [6, 10].

4.1. Comparison with Selective Search

The basis of our algorithm is similar to the Selective Search approach [20]: Selective Search starts with the same oversegmentation method [13]. Then it greedily groups regions with the highest similarity together until the complete image is a single region. All regions are considered potential object locations and the final set of object locations is created by combining multiple hierarchies with different starting segmentations and merging criteria. In this section we first evaluate the impact of the three novelties introduced in this paper: (1) we use a much wider variety of region-based similarity measures, including the similarity measures from Selective Search. (2) we use a Random Forest classifier to optimally combine different similarity measures, while in the Selective Search all similarities are added together using equal weights. (3) we identify stable blobs and enrich a set of segments generated by the hierarchical grouping by considering combinations of stable blobs.

For ease of presentation, our novelties are evaluated starting from a single hierarchy of [20]. In particular, we choose the threshold of [13] to be $k = 50$ (the minimum size of a segment in the oversegmentation [13] is set to k pixels in all experiments), we use Lab colour space, and we use all four similarity measures of [20]: colour (C), texture(T), size (S), and fill (F). For reference we also include the two main strategies of [20]: the 'Fast' and the 'Quality' method, combining 8 and 80 hierarchies respectively.

Results are shown in Table 1. The third entry is a single hierarchy from [20] and generates 920 regions per image on average yielding a MABO of 0.598. If we train a Random Forest using the same features, we get an insignificant improvement to 0.602 MABO, which means that for this limited set of features a simple combination is optimal. Note that the final number of regions for a single hierarchy is determined by the initial oversegmentation. As we keep this the same, the total number of regions does not change (until we introduce stable blobs).

Next we use all features as defined in Section 3.1: size (S), colour similarity in 4 colour spaces (4C), texture similarity in 4 colour spaces (4T), image patch exemplars in 4 colour spaces (4PE), fill (F), size (S), insideness (I), contour and border compactness (2Co), contour and border edge responses (2E), and border smoothness (Sm). If we simply add all similarity measures, we obtain 0.622 MABO, an increase of 0.02. However, if we use Random Forest to learn how to combine similarity measures, we obtain 0.675 MABO. Hence learning adds a significant 0.05 MABO and is necessary to obtain good results with our large set of features that are diverse in nature.

To get an idea of the importance of similarity measures as learned by the random forest, we calculate its performance when the values of a single feature are randomized while all others are left unchanged. Then the accuracy of the random forest on the corrupted data is compared with the accuracy on the original data and we measure importance as the drop in accuracy. Results are shown in Figure 2. We observe that all features contribute. The measures Fill, Size, Patch Exemplars and Colour Similarity in Lab colour space are particularly important. The significance of Lab colour space may be because in our experiment the initial segmentation has been done in Lab colour space.

Finally, we examine the impact of adding combinations of stable blobs (SB) to the object hypotheses. Entry 8 in Table 1 demonstrates that with 2,868 regions we reach MABO of 0.749, which is higher than the Fast Selective Search result of 0.739 with 4,753 regions (first entry). Notice, that the quality of the combinations of stable blobs is highly dependent on the quality of the regions generated by the hierarchical grouping: Method 8 (with RF) outperforms method 7 (no RF) in both MABO and the number of regions.

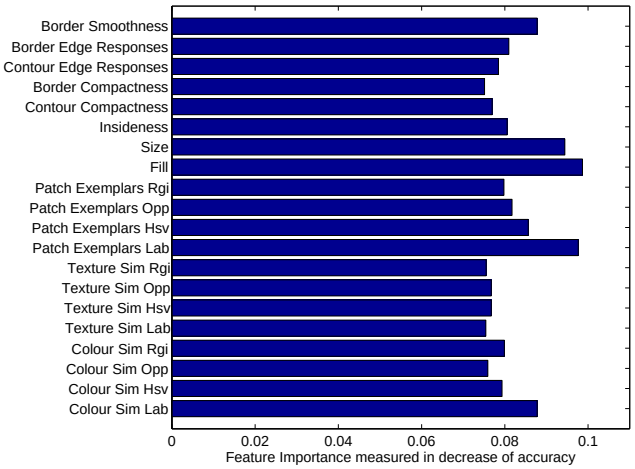

Figure 2. The importance of similarity measures for the random forest classifier.

To conclude, the new similarities, random forest learning, and combination of stable blobs, all contribute to a significant increase in performance, to the point where we obtain a higher MABO with fewer regions than the Fast Selective Search method [20].

4.2. Comparison with the state-of-the-art

In this section we compare the proposed method with the object hypotheses of [6], [10], and [20]. In all experiments we used the publicly available code of the authors.

We test two configurations of the proposed method: 2H and 4H. The first (2H) is based on 2 initial segmentations in Lab and HSV colour spaces with the scale parameter $k = 100$. To obtained it, we build 2 hierarchical groupings with 20 similarity measures using random forest, then we identify stable blobs and add their combinations. The second configuration of our method (4H) is the same, but based on 4 initial segmentations: Lab and HSV colour spaces with the scale parameter $k = 50$ and 100. Notice, that in all cases we use only one random forest trained on 'Lab $k = 50$' initial segmentation, which generalizes well on other initial segmentations. This demonstrates that the considered similarity measures are not tuned to a particular initial image partition.

Table 2 shows that the proposed method outperforms the state-of-the-art in terms of MABO, reaching 0.771 with 2 initial segmentations and 0.812 with 4 initial segmentations. In terms of the number of regions, we generate 1.5 and 4 times more segments that [10] and [6] respectively, and 0.6 less than [20]. However, it is important to notice that both [6] and [10] are quite computationally demanding, whereas our method with 4 initial segmentations takes only 39 seconds per image, and the version with 2 initial segmentation is even faster, it generates a high quality set of object hypotheses in 17 seconds.

Table 2. Comparison with the state-of-the-art.

	VOC 2007		VOC 2012		Time
Method	MABO	Regions	MABO	Regions	(s)
[6]	0.732	695	0.759	642	432
[10]	0.752	1,827	0.760	1,512	226
[20]	0.739	4,753	0.759	4,624	3.8
2H	0.770	2,815	0.781	2,659	17
4H	0.812	10,231	0.820	9,784	39

4.3. Example Segmentations

To illustrate what it means to generate object hypotheses with MABO of 0.77, we show representative best regions for all 20 object classes on Pascal 2012. Therefore we show instances that have an overlap score close to the average best overlap for that class in Figure 3. The class 'bicycle' has the lowest Average Best Overlap of 0.473. This is because a bicycle does not correspond to a coherent region but rather to a wireframe, whereas our method aims to identify coherent regions. However, the second lowest Average Best Overlap is much higher at 0.704 and corresponds to the 'chair' class. Such a jump in performance illustrates that the focus on coherent regions is beneficial for most of the object classes. Nine over 20 considered classes have Average Best Overlap higher than 0.8, including bird which is a very heterogeneous class. As can been seen in Figure 3, the proposed method even finds a parrot occluded by a wire-mesh, which is possible since the wires are too thin to become regions by themselves.

Figure 4 contains several examples where our method cannot find proper segments for the target object. An important reason of failure is when the target object is small, such as in the bottle image (and many other instances which we do not show). In this case, the initial oversegmentation already makes mistakes which are irreversible in our algorithm. Another problem arises when the object is occluded by similar objects, such as the first two images in the second row (car and cat). Without recognizing objects or recognizing the 3D environmental layout and its constraints on physical objects, there is little hope of accurately separating similar objects that occlude each other. Such information would also help when objects blend very well into the background, such as the table and the baby. Another problem is occlusion, such as in the airplane and car images on the first row, the second cat image on the second row, the chair, and the horse. As our algorithm only merges adjacent regions, disjunct objects will not be found. While it is possible to generate disjunct regions, it significantly increases the search space and, if not done carefully, may introduce many extra non-object regions. Finally, our algorithm is not designed to deal with wire-frames, hence it often has problems with bicycles (first row Figure 4), and sometimes chairs (see second row) and plants (third row).

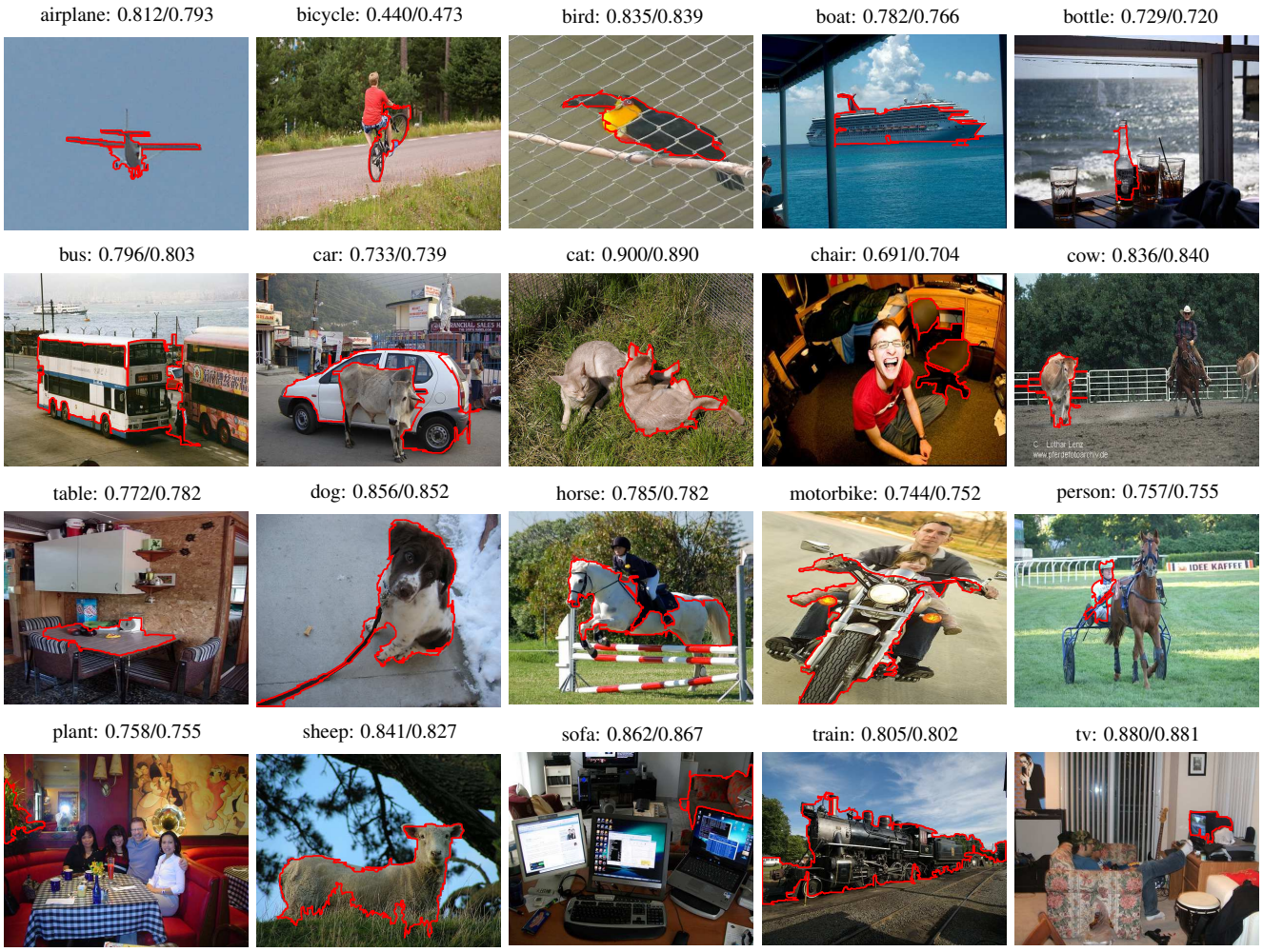

Figure 3. Examples of generated object proposals for 20 classes of Pascal VOC 2012 whose overlap is close to the average best overlap of that class. The first number is the region overlap with the ground truth, and the second number is the average best overlap per class. The red line corresponds to the contour of regions created using the proposed method based on 2 hierarchical groupings.

4.4. Discussion

The main goal of generating class-independent object hypotheses is to improve semantic segmentation and object recognition by focusing computational resources on promising parts of the image. MABO provides an upper bound for semantic segmentation. It is less clear how MABO influences object recognition, since this depends on the recognition method of choice. For example, part-based recognition [12] is more sensitive to the exact (box-based) location [1] than Bag-of-Words [7, 20]. Furthermore, not all object parts are equally informative: a cat's head is smaller than its body, yielding a smaller MABO score but providing more discriminative information. Nevertheless, best overlap scores give a reasonable indication of the usefulness of object proposals. In future work we want to refine the analysis into small/large objects and use the occluded/difficult flags present in the ground truth. The desired number of regions is mostly bounded by computational resources: good recognition systems do not suffer from additional but bad regions, they only require a single high-quality region.

5. Conclusions

In this paper we presented a novel algorithm for generating class-independent object regions. We start from an oversegmentation after which we perform a greedy, hierarchical grouping algorithm. Regions are represented through a wide variety of relatively powerful region-features. We train a Random Forest to determine at each step of the hierarchy which regions should be merged. We deal with objects consisting of dissimilar parts by identifying stable regions and merging adjacent ones. Our results show a significant performance improvement in terms of both quality and quantity of regions with respect to [20]. We report significant improvements over [6, 10] in terms of both computational efficiency and quality of regions.

Acknowledgements

This research has been partially supported by the EC STREP project xLiMe.

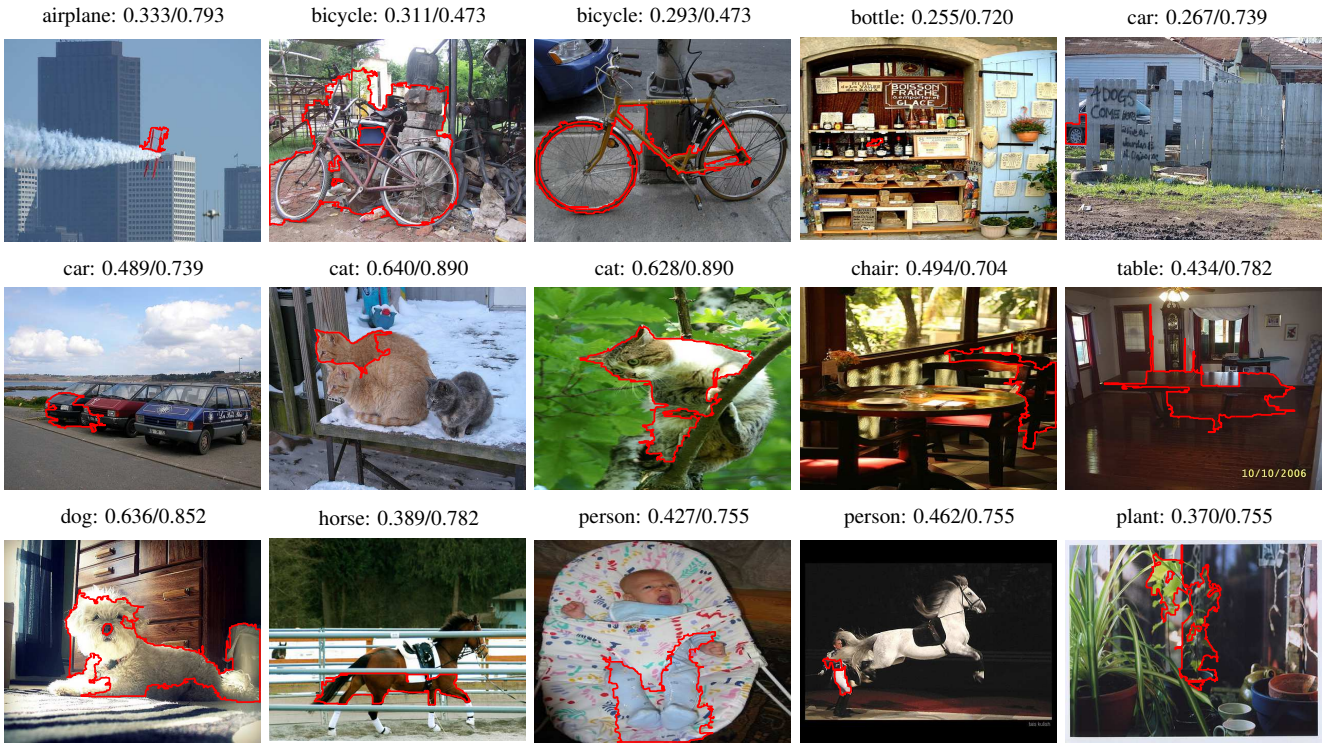

Figure 4. Selection of examples of objects for which the best segment has a low Best Overlap score. The first number is the BO score for this image while the second number is the ABO for that class.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *PAMI*, 2012. 2, 3, 7

[2] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *CVPR*, 2013. 1

[3] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *PAMI*, 2011. 2

[4] L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001. 4

[5] J. Carreira, F. Li, and C. Sminchisescu. Object recognition by sequential figure-ground ranking. *IJCV*, 2012. 1

[6] J. Carreira and C. Sminchisescu. Cpmc: Automatic object segmentation using constrained parametric min-cuts. *PAMI*, 2012. 1, 2, 5, 6, 7

[7] R. Cinbis, J. Verbeek, and C. Schmid. Segmentation driven object detection with fisher vectors. In *ICCV*, 2013. 1, 7

[8] A. Criminisi, J. Shotton, and E. Konukoglu. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. *Foundations and Trends in Computer Graphics and Vision*, 2012. 4

[9] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual Categorization with Bags of Keypoints. In *ECCV International Workshop on Statistical Learning in Computer Vision*, Prague, 2004. 2

[10] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *PAMI*, 2013. 1, 2, 5, 6, 7

[11] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *IJCV*, 2010. 5

[12] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2010. 2, 3, 7

[13] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, (2), 2004. 3, 5

[14] S. Fidler, R. Mottaghi, A. Yuille, and R. Urtasun. Bottom-up segmentation for top-down detection. In *CVPR*, 2013. 1

[15] J. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color Invariance. *PAMI*, 2001. 3

[16] D. Hoiem, A. Stein, A. Efros, and M. Hebert. Recovering occlusion boundaries from a single image. In *ICCV*, 2007. 2

[17] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, 2010. 2

[18] E. Rahtu, J. Kannala, and M. Blaschko. Learning a category independent object detection cascade. In *ICCV*, 2011. 2, 3

[19] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *ICCV*, 2003. 2

[20] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders. Selective search for object recognition. *IJCV*, 2013. 1, 2, 3, 5, 6, 7

[21] M. Varma and A. Zisserman. A statistical approach to material classification using image patch exemplars. *PAMI*, 2009. 3

[22] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. http://www.vlfeat.org/, 2008. 3

[23] D. Weiss and B. Taskar. SCALPEL: Segmentation cascades with localized priors and efficient learning. In *CVPR*, 2013. 1, 2