Segmentation as selective search for object recognition

Conference Paper *in* Proceedings / IEEE International Conference on Computer Vision. IEEE International Conference on Computer Vision · November 2011

DOI: 10.1109/ICCV.2011.6126456 · Source: DBLP

4 authors, including:

Jasper R. R. Uijlings
The University of Edinburgh
77 PUBLICATIONS **8,760** CITATIONS
SEE PROFILE

T. Gevers
University of Amsterdam
328 PUBLICATIONS **19,912** CITATIONS
SEE PROFILE

Segmentation as Selective Search for Object Recognition

Koen E. A. van de Sande* Jasper R. R. Uijlings† Theo Gevers* Arnold W. M. Smeulders*
*University of Amsterdam †University of Trento
Amsterdam, The Netherlands Trento, Italy
ksande@uva.nl, jrr@disi.unitn.it, th.gevers@uva.nl, a.w.m.smeulders@uva.nl

Abstract

For object recognition, the current state-of-the-art is based on exhaustive search. However, to enable the use of more expensive features and classifiers and thereby progress beyond the state-of-the-art, a selective search strategy is needed. Therefore, we adapt segmentation as a selective search by reconsidering segmentation: We propose to generate many approximate locations over few and precise object delineations because (1) an object whose location is never generated can not be recognised and (2) appearance and immediate nearby context are most effective for object recognition. Our method is class-independent and is shown to cover 96.7% of all objects in the Pascal VOC 2007 test set using only 1,536 locations per image. Our selective search enables the use of the more expensive bag-of-words method which we use to substantially improve the state-of-the-art by up to 8.5% for 8 out of 20 classes on the Pascal VOC 2010 detection challenge.

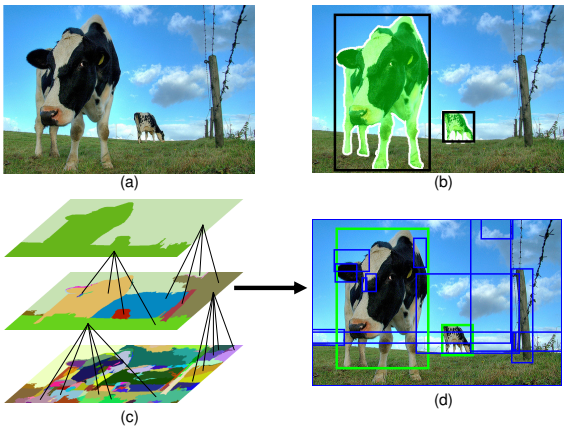

Figure 1. Given an image (a) our aim is to find its objects for which the ground truth is shown in (b). To achieve this, we adapt segmentation as a selective search strategy: We aim for high recall by generating locations at all scales and account for many different scene conditions by employing multiple invariant colour spaces. Example object hypotheses are visualised in (d).

1. Introduction

Object recognition, *i.e.* determining the position and the class of an object within an image, has made impressive progress over the past few years, see the Pascal VOC challenge [8]. The state-of-the-art is based on exhaustive search over the image to find the best object positions [6, 9, 13, 28, 29]. However, as the total number of images and windows to evaluate in an exhaustive search is huge and growing, it is necessary to constrain the computation per location and the number of locations considered. The computation is currently reduced by using a weak classifier with simple-to-compute features [6, 9, 13, 28, 29], and by reducing the number of locations on a coarse grid and with fixed window sizes [6, 9, 27]. This comes at the expense of overlooking some object locations and misclassifying others. Therefore, we propose *selective search*, greatly reducing the number of locations to consider. Specifically, we propose to use segmentation to generate a limited set of locations, permitting the more powerful yet expensive bag-of-words features [5, 23, 26].

Selective search has been exploited successfully by [3, 7] for object delineation, *i.e.* creating a pixel-wise classification of the image. Both concentrate on 10-100 possibly overlapping segments per image, which best correspond to an object. They focus on finding accurate object contours, which is why both references use a powerful, specialized contour detector [2]. In this paper, we reconsider segmentation to use as an instrument to select the best locations for object recognition. Rather than aiming for 10-100 accurate locations, we aim to generate 1,000-10,000 approximate locations. For boosting object recognition, (1) generating several thousand locations per image guarantees the inclusion of virtually all objects, and (2) rough segmentation includes the local context known to be beneficial for object classification [6, 25]. Hence we place our computational attention precisely on these parts of the image which bear the most information for object classification.

Emphasizing recall (encouraging to include all image fragments of potential relevance) was earlier proposed by

Hoiem *et al*. [14] for surface layout classification and adopted by Russell *et al*. [22] for latent object discovery. In the references its use is limited to changing the scale of the segmentation, while its potential for finding objects has yet to be investigated. Malisiewicz and Efros [21] investigated how well segments capture objects as opposed to the bounding boxes of an exhaustive search. They also mainly change the scale of the segmentation. In contrast, this paper uses a full segmentation hierarchy and accounts for as many different scene conditions as possible, such as shadows, shading, and highlights, by using a variety of invariant colour spaces. Furthermore, we demonstrate the power of segmentation as selective search on the challenging Pascal VOC dataset in terms of both recall and recognition accuracy.

To summarize, we make the following contributions: (1) We reconsider segmentation by adapting it as an instrument to select the best locations for object recognition. We put most emphasis on recall and prefer good object approximations over exact object boundaries. (2) We demonstrate that accounting for scene conditions through invariant colour spaces results in a powerful selective search strategy with high recall. (3) We show that our selective search enables the use of more expensive features such as bag-of-words and substantially improves the state-of-the-art on the Pascal VOC 2010 detection challenge for 8 out of 20 classes.

2. Related Work

In Figure 2, the relation of this paper with other work is visualized. Research within localisation can generally be divided into two categories. 1) Work with emphasis on *recognition* (Section 2.1). Here determining the object class is more important than finding the exact contours and an exhaustive search is the norm. 2) Work with emphasis on *object delineation* (Section 2.2). Here object contours are most important and the use of segmentation is the norm.

There are two exceptions to these categories. Vedaldi *et al*. [27] use jumping windows [4], in which the relation between individual visual words and the object location is learned to predict the object location in new images. Maji and Malik [20] combine multiple of these relations to predict the object location using a Hough-transform, after which they randomly sample windows close to the Hough maximum. Both methods can be seen as a selective search. In contrast to learning, we adopt segmentation as selective search to generate class independent object hypotheses.

2.1. Exhaustive Search for Recognition

As an object can be located at any position and scale in the image, it is natural to search everywhere [6, 13, 28]. However, the visual search space is huge, making an exhaustive search computationally expensive. This imposes constraints on the evaluation cost per location and/or the number of locations considered. Hence most of these sliding window techniques use a coarse search grid and fixed aspect ratios, using weak classifiers and economic image features such as HOG [6, 13, 28]. This method is often used as a preselection step in a cascade of classifiers [13, 28].

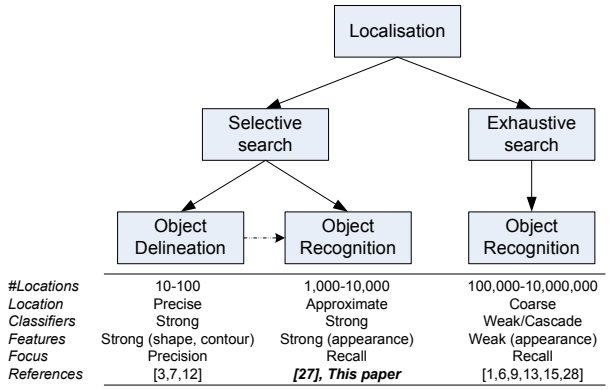

Figure 2. Positioning of this paper with respect to related work.

Related to the sliding window technique is the highly successful part-based object localisation method of Felzenszwalb *et al*. [9]. Their method also performs an exhaustive search using a linear SVM and HOG features. However, they search for objects *and* object parts, whose combination results in an impressive object detection performance.

Lampert *et al*. [15] developed a branch and bound technique to directly search for the optimal window within an image. While they obtain impressive results for linear classifiers, [1] found that for non-linear classifiers the method in practice still visits over a 100,000 windows per image.

While the previous methods are all class-specific, Alexe *et al*. [1] propose to search for *any* object, independent of its class. They train a classifier on the object windows of those objects which have a well-defined shape (as opposed to *e.g*. grass). Then instead of a full exhaustive search they randomly sample boxes to which they apply their classifier. The boxes with the highest "objectness" measure serve as a set of object hypotheses. This set is then used to greatly reduce the number of windows evaluated by class-specific object detectors.

Instead of an exhaustive search, in this paper, we propose to do segmentation as a selective search enabling the immediate use of expensive and potentially more powerful recognition techniques. In contrast to all exhaustive methods except [1], our method yields an object hypotheses set which is completely class independent.

2.2. Selective Search for Object Delineation

In the domain of object delineation, both Carreira *et al*. [3] and Endres and Hoiem [7] propose to generate a set of class independent object hypotheses using segmentation. Both methods generate multiple foreground/background segmentations, learn to predict the likelihood that a fore-

Figure 3. Two examples of our hierarchical grouping algorithm showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

ground segment is a complete object, and use this to rank the segments. Both algorithms show a promising ability to accurately delineate objects within images, confirmed by [17] who achieve state-of-the-art results on pixel-wise image classification using [3]. This paper uses selective search for object *recognition*, hence we put more emphasis on recall and welcome rough object locations instead of precise object delineations. We can omit the excellent yet expensive contour detector of [2] included in [3, 7], making our algorithm computationally feasible on large datasets. In contrast to [3, 7], we use a hierarchical grouping algorithm instead of multiple foreground/background segmentations.

Gu *et al*. [12] address the problem of carefully segmenting and recognizing objects based on their parts. They first generate a set of part hypotheses using a grouping method based on [2]. Each part hypothesis is described by both appearance and shape features. Then an object is recognized and carefully delineated by using its parts, achieving good results for shape recognition. In their work, the segmentation is limited to a single hierarchy while its power of discovering parts or objects is not evaluated. In this paper, we use multiple hierarchical segmentations diversified through employing a variety of colour spaces, and evaluate their potential to find complete objects.

3. Segmentation as Selective Search

In this section, we adapt segmentation as selective search for object recognition. This adaptation leads to the following considerations:

High recall. Objects whose locations are not generated can never be recognized. Recall is therefore the most important criterion. To obtain a high recall we observe the following: (1) Objects can occur at any scale within an image. Moreover, some objects are contained within other objects. Hence it is necessary to generate locations at all scales, as illustrated in Figure 3. (2) There is no single best strategy to group regions together: An edge may represent an object boundary in one image, while the same edge in another image may be the result of shading. Hence rather than aiming for the single best segmentation, it is important to combine multiple complementary segmentations, *i.e*. we want to diversify the set of segmentations used.

Coarse locations are sufficient. As the state-of-the-art in object recognition uses appearance features, the exact object contours of the object hypotheses are less important. Hence instead of a strong focus on object boundaries (*e.g*. [2]), the evaluation should focus on finding reasonable approximations of the object locations, such as is measured by the Pascal overlap criterion [8].

Fast to compute. The generation of the object hypotheses should not become a bottleneck when performing object localisation on a large dataset.

3.1. Our Segmentation Algorithm

The most natural way to generate locations at all scales is to use all locations from a hierarchical segmentation algorithm (illustrated in Figure 1). Our algorithm uses size and appearance features which are efficiently propagated throughout the hierarchy, making it reasonably fast. Note that we keep the algorithm basic to ensure repeatability and make clear that our results do not stem from parameter tuning but from rethinking the goal of segmentation.

As regions can yield richer information than pixels, we start with an oversegmentation, *i.e*. a set of small regions which do not spread over multiple objects. We use the fast method of [10] as our starting point, which [2] found well-suited for generating an oversegmentation.

Starting from the initial regions, we use a greedy algorithm which iteratively groups the two most similar regions together and calculates the similarities between this new region and its neighbours. We continue until the whole image becomes a single region. As potential object locations, we consider either all segments throughout the hierarchy (including initial segments), or we consider the tight bounding boxes around these segments.

We define the similarity S between region a and b as $S(a,b) = S_{size}(a,b) + S_{texture}(a,b)$. Both components result in a number in range [0,1] and are weighed equally.

$S_{size(a,b)}$ is defined as the fraction of the image that the segment a and b jointly occupy. This measure encourages small regions to merge early and prevents a single region from gobbling up all others one by one.

$S_{texture}(a,b)$ is defined as the histogram intersection between SIFT-like texture measurements [18]. For these measurements, we aggregate the gradient magnitude in 8 directions over a region, just like in a single subregion of SIFT with no Gaussian weighting. As we use colour, we follow [26] and do texture measurements in each colour channel separately and concatenate the results.

3.2. Shadow, Shading and Highlight Edges

To obtain multiple segmentations which are complementary, we perform our segmentation in a variety of colour channels with different invariance properties. Specifically, we consider multiple colour spaces with different degrees of sensitivity to shadow, shading and highlight edges [11]. Standard RGB is the most sensitive. The opponent colour space is insensitive to highlight edges, but sensitive to shadows and shading edges. The normalized RGB space is insensitive to shadow and shading edges but still sensitive to highlights. The hue H is the most invariant and is insensitive to shadows, shading and highlights. Note that we always perform each segmentation in a single colour space, including the initial segmentation of [10].

An alternative approach to multiple colour spaces would be the use of different thresholds for the starting segmentation. We evaluate this approach as well.

3.3. Discussion

Our adaptation of segmentation as selective search for object recognition is designed to obtain high recall by considering all levels of a hierarchical grouping of image segments. Furthermore, by considering multiple colour spaces with increasing levels of invariance to imaging conditions, we are robust to the additional edges introduced into an image by shadows, shading and highlights. Finally, our approach is fast which makes it applicable to large datasets.

4. Object Recognition System

In this section, we detail how to use the selective search strategy from Section 3 for a complete object recognition system. As feature representation, two types of features are dominant: histograms of oriented gradients (HOG) [6] and bag-of-words [5, 23]. HOG has been shown to be successful in combination with the part-based model by Felzenszwalb *et al*. [9]. However, as they use an exhaustive search, HOG features in combination with a linear classifier is the only feasible choice. To show that our selective search strategy enables the use of more expensive and potentially more powerful features, we use Bag-of-Words for object recognition [13, 15, 27]. We use a more powerful (and expensive) implementation than [13, 15, 27] by employing multiple colour spaces and a finer spatial pyramid division [16].

Specifically we sample descriptors at each pixel on a single scale. We extract SIFT [18] and two recommended colour SIFTs from [26], OpponentSIFT and RGB-SIFT. Software from [26] is used. We use a visual codebook of size 4,096 and a spatial pyramid with 4 levels. Because a spatial pyramid results in a coarser spatial subdivision than the cells which make up a HOG descriptor, our features contain less information about the specific spatial layout of the object. Therefore, HOG is better suited for rigid objects and our features are better suited for deformable object types.

As classifier we employ a Support Vector Machine with a histogram intersection kernel using [24]. We use the fast, approximate classification strategy of [19].

Our training procedure is illustrated in Figure 4. The initial positive examples consist of all ground truth object windows. As initial negative examples we use all object locations generated by our selective search that have an overlap of 20% to 50% with a positive example, unless they have more than 70% overlap with another negative, *i.e*. we avoid near duplicates. This selection of training examples gives reasonably good initial classification models.

Then we enter a retraining phase to iteratively add hard negative examples (*e.g*. [9]): We apply the learned models to the training set using the locations generated by our selective search. For each negative image we add the highest scoring location. As our initial training set already yields good models, our models converge in only two iterations.

For the test set, the final model is applied to all locations generated by our selective search. The windows are sorted by classifier score while windows which have more than 30% overlap with a higher scoring window are considered near-duplicates and are removed.

5. Evaluation

To evaluate the quality of our selective search strategy, we perform the following four experiments:

- **Experiment 1** evaluates how to adapt segmentation for selective search. Specifically we compare multiple flat segmentations against a hierarchy and evaluate the use of increasingly invariant colour spaces.
- **Experiment 2** compares segmentation as selective search on the task of generating good object locations for recognition with [1, 13, 27].
- **Experiment 3** compares segmentation as selective search on the task of generating good object delineations for segmentation with [3, 7].
- **Experiment 4** evaluates the use of our object hypotheses in the object recognition system of Section 4, on the widely accepted object localisation method of [9] and compares it to the state-of-the-art [8, 9, 29].

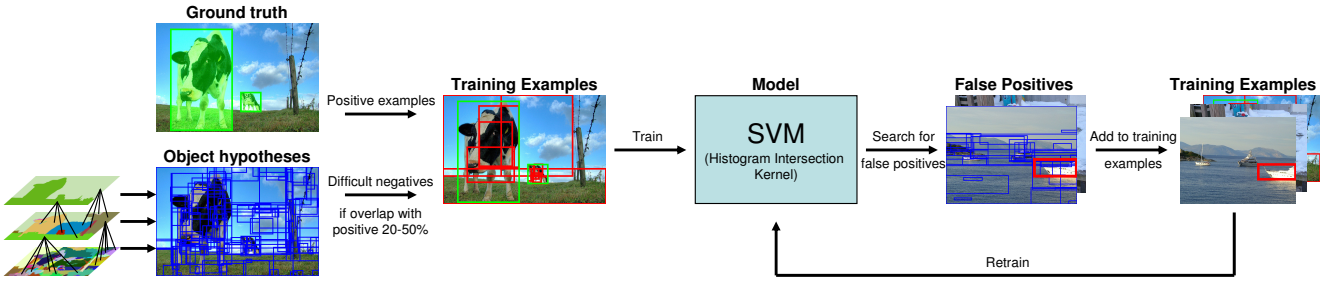

Figure 4. The training procedure of our object recognition pipeline. As positive learning examples we use the ground truth. As negatives we use examples that have a 20-50% overlap with the positive examples. We iteratively add hard negatives using a retraining phase.

In all experiments, we report results on the challenging Pascal VOC 2007 or 2010 datasets [8]. These datasets contain images of twenty object categories and the ground truth in terms of object labels, the location in terms of bounding boxes, and for a subset of the data the object location in terms of a pixel-wise segmentation.

As in [13, 27], the quality of the hypotheses is defined in terms of the average recall over all classes versus the number of locations retrieved. We use the standard Pascal overlap criterion [8] where an object is considered found if the area of the intersection of a candidate location and the ground truth location, divided by the area of their union is larger than 0.5. Note that in the first two experiments the location is a bounding box, and in the third it is a segment.

Any parameter selection was done on the training set only, while results in this paper are reported on the test set.

5.1. Exp. 1: Segmentation for Selective Search

In this experiment, we evaluate how to adapt segmentation for selective search. First, we compare multiple flat segmentations against a hierarchical segmentation. Second, we evaluate the use of a variety of colour spaces.

Flat versus Hierarchy. As our segmentation algorithm starts with the initial oversegmentation of [10], we compare our hierarchical version with multiple flat segmentations by [10]. We do this in RGB colour space. We vary the scale of [10] by setting the threshold k from 100 to 1000 both in steps of 10 and in steps of 50. For our hierarchical algorithm we use the smallest threshold 100. Varying the threshold k results in many more segments than a single hierarchical grouping, because in [10] the segment boundaries resulting from a high threshold are not a subset of those from a small threshold. Therefore we additionally consider two hierarchical segmentations using a threshold of 100 and 200.

Experiment 1: Multiple Flat segmentations versus Hierarchy

	Max. recall (%)	# windows
[10] $k = 100, 150 \ldots 1000$	84.8	665
[10] $k = 100, 110 \ldots 1000$	87.7	1159
Hierarchical $k = 100$	80.6	362
Hierarchical $k = 100, 200$	89.4	511

Table 1. Comparison of multiple flat segmentations versus a hierarchy in terms of recall and the number of windows per image.

As can be seen from Table 1, multiple flat segmentations yield a higher recall than a single hierarchical grouping but using many more locations. However, if we choose two initial thresholds and combine results, our algorithm yields recall of 89.4 instead of 87.7, while using only 511 locations instead of 1159. Hence a hierarchical approach is preferable over multiple flat segmentations as it yields better results, fewer parameters, and selects all scales naturally. Additionally, we found it to be much faster.

Multiple Colour Spaces. We now test two diversification strategies to obtain higher recall. As seen in the previous experiment it is beneficial to use multiple starting segmentations. Furthermore we test how combining different colour spaces with different invariance properties can increase the number of objects found. Specifically, we take a segmentation in RGB colour space, and subsequently add the segmentation in Opponent colour space, normalized rgb colour space, and the Hue channel. We do this for a single initial segmentation with $k = 100$, two initial segmentations with $k = 100, 200$, and four initial segmentations with $k = 100, 150, 200, 250$. Results are shown in Figure 5.

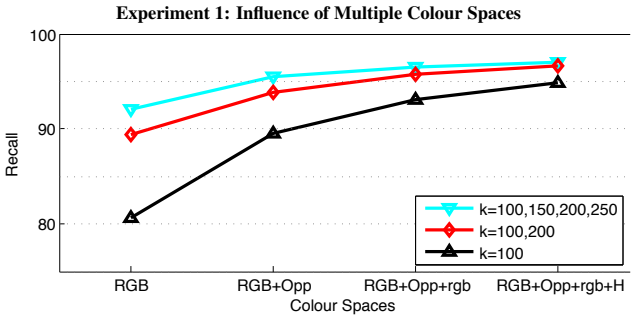

Figure 5. Using multiple colour spaces clearly improves recall; along the horizontal axis increasingly invariant colour spaces are added.

As can be seen, both changing the initial segmentation and using a variety of different colour channels yield complementary object locations. Note that using four different colour spaces works better than using four different initial segmentations. Furthermore, when using all four colour spaces the difference between two and four initial segmentations is negligible. We conclude that varying the colour spaces with increasing invariances is better than varying the

threshold of the initial segmentation. In subsequent experiments we always use these two initial segmentations.

On the sensitivity of parameters. In preliminary experiments on the training set we used other colour spaces such as HSV, HS, normalized rg plus intensity, intensity only, etc. However, we found that as long as one selects colour spaces with a range of invariance properties, the outcome is very similar. For illustration purposes we used in this paper the colour spaces with the most clear invariance properties. Furthermore, we found that as long as a good oversegmentation is generated, the exact choice for k is unimportant. Finally, different implementations of the texture histogram yielded little changes overall. We conclude that the recall obtained in this paper is not caused by parameter tuning but rather by having a good diversification of segmentation strategies through different colour invariance properties.

5.2. Exp. 2: Selective Search for Recognition

We now compare our selective search method to the sliding windows of [13], the jumping windows of [27], and the 'objectness' measure of [1]. Table 2 shows the maximum recall obtained for each method together with the average number of locations generated per image. Our method achieves the best results with a recall of 96.7% with on average 1,536 windows per image. The jumping windows of [27] come second with 94% recall but uses 10,000 windows instead. Moreover, their method is specifically trained for each class whereas our method is completely class-independent. Hence, with only a limited number of object locations our method yields the highest recall.

Experiment 2: Maximum Recall of Selective Search for Recognition

	Max. recall (%)	# windows
Sliding Windows [13]	83.0	200 per class
Jumping Windows [27]	94.0	10,000 per class
'Objectness' [1]	82.4	10,000
Our hypotheses	96.7	1,536

Table 2. Comparison of maximum recall between our method and [1, 13, 27]. We achieve the highest recall of 96.7%. Second comes [27] with 94.0% but using an order of magnitude more locations.

We also compare the trade-off between recall and the number of windows in Figure 6. As can be seen, our method gives a higher recall using fewer windows than [1, 27]. The method of [13] seems to need only few windows to obtain their maximum recall of 83%. However, they use 200 windows per image *per class*, which means they generate 4,000 windows per image. Moreover, the ordering of their hypotheses is based on a class specific recognition score while the ordering of our hypotheses is imposed by the inclusion of segmentations in increasingly invariant colour spaces.

In conclusion, our selective search outperforms other methods in terms of maximum recall while using fewer locations. Additionally, our method is completely class-independent. This shows that segmentation, when adapted for high recall by using all scales and a variety of colour spaces with different invariance properties, is a highly effective selective search strategy for object recognition.

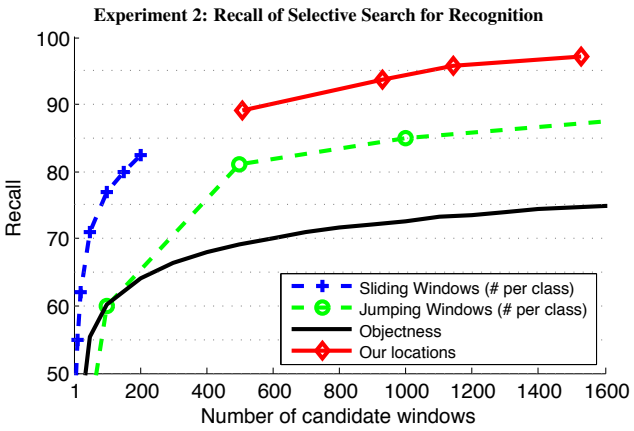

Figure 6. The trade-off between the number of retrieved windows and recall on the Pascal VOC 2007 object detection dataset. Note that for [13, 27] the reported number of locations is *per class*; the total number of windows per image is a factor 20 higher.

5.3. Exp. 3: Selective Search for Object Delineation

The methods of [3, 7] are designed for object delineation and computationally too expensive to apply them to the VOC 2007 detection dataset. Instead we compare to them on the much smaller *segmentation* dataset using not boxes but the segments instead. We generated candidate segments for [3, 7] by using their publicly available code. Note that we excluded the background category in the evaluation.

Results are shown in Table 3. The method of [7] achieves the best recall of 82.2% using 1,989 windows. Our method comes second with a recall of 79.8% using 1973 segments. The method of [3] results in a recall of 78.2% using only 697 windows. However, our method is 28 times faster than [3] and 54 times faster than [7]. We conclude that our method is competitive in terms of recall while still computationally feasible on large datasets.

Experiment 3: Recall of Selective Search for Segmentation

	Max. recall (%)	# windows	Time (s)
Carreira [3]	78.2	697	432
Endres [7]	82.2	1,989	226
Our hypotheses	79.8	1,973	8
Combination	90.1	4,659	666

Table 3. Comparison of our paper with [3, 7] in terms of recall on the Pascal VOC 2007 segmentation task. Our method has competitive recall while being more than an order of magnitude faster.

Interestingly, we tried to diversify the selective search by combining all three methods. The resulting recall is 90.1%(!), much higher than any single method. We conclude that for the purpose of recognition, instead of aiming for the *best* segmentation, it is prudent to investigate how segmentations can *complement* each other.

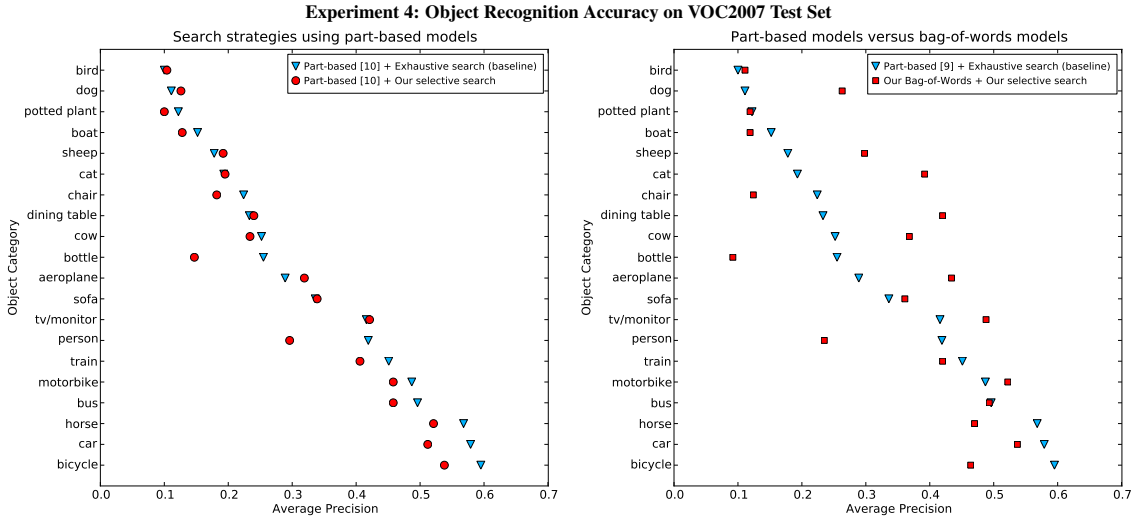

Figure 7. Object recognition results on the PASCAL VOC 2007 test set. For the left plot, object models are trained using the part-based Felzenszwalb system [9], which uses exhaustive search by default. For the right plot, object models are trained using more expensive bag-of-words features and classifiers; exhaustive search is not feasible with these models.

5.4. Exp. 4: Object Recognition Accuracy

In this experiment, we evaluate our object hypotheses on a widely accepted part-based object recognition method [9] and inside the object recognition system described in Section 4. The latter is compared to the state-of-the-art on the challenging Pascal VOC 2010 detection task.

Search strategies using part-based models. We compare various search strategies on the method of Felzenszwalb [9]. We consider the exhaustive search of [9] to be our baseline. We use our selective search boxes as a filter on the output of [9], as facilitated by their code, where we discard all locations whose Pascal Overlap is smaller than 0.8. In practice this reduces the number of considered windows from around 100,000 per image per class to around 5,000. Results are shown on the left in Figure 7. Overall using our boxes as a filter reduces Mean Average Precision from 0.323 MAP to 0.296 MAP, 0.03 MAP less while evaluating 20 times fewer boxes. Note that for some concepts like *aeroplane*, *dog*, *dining table*, and *sheep* there is even a slight improvement, suggesting a trade-off between high recall and precision for object detection accuracy.

If we use all 10,000 boxes of [1] in the same manner on [9], the MAP reduces to 0.215. But in [1] they have an additional hill-climbing step which enables them to consider only 2,000 windows at the expense of 0.04 MAP. This suggest that a hill-climbing step as suggested by [1] could improve results further when using our boxes.

Part-based HOG versus bag-of-words. A major advantage of selective search is that it enables the use of more expensive features and classifiers. To evaluate the potential of better features and classifiers, we compare the bag-of-words recognition pipeline described in Section 4 with the baseline of [9] which uses HOG and linear classifiers. Results on the right in Figure 7 show improvements for 10 out of 20 object categories. Especially significant are the improvements the object categories *cat, cow, dog, sheep, diningtable*, and *aeroplane*, which we improve with 11% to 20%. Except aeroplane, these object categories all have flexible shape on which bag-of-words is expected to work well (Section 4). The baseline achieves a higher accuracy for object categories with rigid shape characteristics such as *bicycle*, *car*, *bottle*, *person* and *chair*. If we select the best method for each class, instead of a MAP of 0.323 of the baseline, we get a MAP of 0.378, a significant, absolute improvement of 5% MAP.

To check whether the differences on the right in Figure 7 originate mainly from the different features, we combined bag-of-words features with the exhaustive search of [9] for the concepts *cat* and *car*. With *cat*, bag-of-words gives 0.392 AP for selective and 0.375 AP for exhaustive search, compared to 0.193 AP for part-based HOG features. With *car*, bag-of-words gives 0.547 for selective and 0.535 for exhaustive search, and 0.579 for part-based HOG features.

Comparison to the state-of-the-art. To compare our results to the current state-of-the-art in object recognition, we have submitted our bag-of-words models for the Pascal VOC 2010 detection task to the official evaluation server. Results are shown in Table 4, together with the top-4 from the competition. In this independent evaluation, our system improves the state-of-the-art by up to 8.5% for 8 out of 20 object categories compared to all other competition entries.

In conclusion, our selective search yields good object locations for part-based models, as even without the hill-climbing step of [1] we need to evaluate 20 times fewer windows at the expense of 0.03 MAP in average precision. More importantly, our selective search enables the use of

Experiment 4: Object Recogntion Accuracy on VOC2010 Test Set

System	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv
NLPR	.533	**.553**	**.192**	**.210**	.300	.544	.467	.412	**.200**	**.315**	.207	.303	.486	.553	.465	.102	.344	.265	**.503**	.403
MIT UCLA [29]	.542	.485	.157	.192	.292	**.555**	.435	.417	.169	.285	.267	.309	.483	.550	.417	.097	.358	.308	.472	.408
NUS	.491	.524	.178	.120	.306	.535	.328	.373	.177	.306	.277	.295	**.519**	**.563**	.442	.096	.148	.279	.495	.384
UoCTTI [9]	.524	.543	.130	.156	**.351**	.542	**.491**	.318	.155	.262	.135	.215	.454	.516	**.475**	.091	.351	.194	.466	.380
This paper	**.582**	.419	**.192**	.140	.143	.448	.367	**.488**	.129	.281	**.287**	**.394**	.441	.525	.258	**.141**	**.388**	**.342**	.431	**.426**

Table 4. Results from the Pascal VOC 2010 detection task test set, comparing the approach from this paper to the current state-of-the-art. We improve the state-of-the-art up to 0.085 AP for 8 categories and equal the state-of-the-art for one more category.

expensive features and classifiers which allow us to substantially improve the state-of-the-art for 8 out of 20 classes on the VOC2010 detection challenge.

6. Conclusions

In this paper, we have adopted segmentation as a selective search strategy for object recognition. For this purpose we prefer to generate many approximate locations over few and precise object delineations, as objects whose locations are not generated can never be recognised and appearance and immediate nearby context are effective for object recognition. Therefore our selective search uses locations at all scales. Furthermore, rather than using a single best segmentation algorithm, we have shown that for recognition it is prudent to use a set of complementary segmentations. In particular this paper accounts for different scene conditions such as shadows, shading, and highlights by employing a variety of invariant colour spaces. This results in a powerful selective search strategy that generates only 1,536 class-independent locations per image to capture 96.7% of all the objects in the Pascal VOC 2007 test set. This is the highest recall reported to date.

We show that segmentation as a selective search strategy is highly effective for object recognition: For the part-based system of [9] the number of considered windows can be reduced by 20 times at a loss of 3% MAP overall. More importantly, by capitalizing on the reduced number of locations we can do object recognition using a powerful yet expensive bag-of-words implementation and improve the state-of-the-art for 8 out of 20 classes for up to 8.5% in terms of Average Precision.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. 2, 4, 6, 7

[2] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *TPAMI*, 2011. 1, 3

[3] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. 1, 2, 3, 4, 6

[4] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007. 2

[5] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Statistical Learning in Computer Vision*, 2004. 1, 4

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 1, 2, 4

[7] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, 2010. 1, 2, 3, 4, 6

[8] M. Everingham, L. van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010. 1, 3, 4, 5

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *TPAMI*, 32:1627–1645, 2010. 1, 2, 4, 7, 8

[10] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient Graph-Based Image Segmentation. *IJCV*, 59:167–181, 2004. 3, 4, 5

[11] T. Gevers and H. M. G. Stokman. Classification of color edges in video into shadow-geometry, highlight, or material transitions. *TMM*, 5(2):237–243, 2003. 4

[12] C. Gu, J. J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 3

[13] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009. 1, 2, 4, 5, 6

[14] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 2007. 2

[15] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Efficient subwindow search: A branch and bound framework for object localization. *TPAMI*, 31:2129–2142, 2009. 2, 4

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 4

[17] F. Li, J. Carreira, and C. Sminchisescu. Object recognition as ranking holistic figure-ground hypotheses. In *CVPR*, 2010. 3

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60:91–110, 2004. 4

[19] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008. 4

[20] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, 2009. 2

[21] T. Malisiewicz and A. A. Efros. Improving spatial support for objects via multiple segmentations. In *BMVC*, 2007. 2

[22] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 2

[23] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 1, 4

[24] S. Sonnenburg, G. Raetsch, S. Henschel, C. Widmer, J. Behr, A. Zien, F. de Bona, A. Binder, C. Gehl, and V. Franc. The shogun machine learning toolbox. *JMLR*, 11:1799–1802, 2010. 4

[25] J. R. R. Uijlings, A. W. M. Smeulders, and R. J. H. Scha. What is the spatial extent of an object? In *CVPR*, 2009. 1

[26] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek. Evaluating color descriptors for object and scene recognition. *TPAMI*, 32:1582–1596, 2010. 1, 4

[27] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 1, 2, 4, 5, 6

[28] P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57:137–154, 2004. 1, 2

[29] L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 1, 4, 8