# Cascaded Sparse Spatial Bins for Efficient and Effective Generic Object Detection

David Novotny[1,2]
[1]Visual Geometry Group
University of Oxford
david@robots.ox.ac.uk

Jiri Matas[2]
[2]Center for Machine Perception
Czech Technical University in Prague
matas@cmp.felk.cvut.cz

## Abstract

*A novel efficient method for extraction of object proposals is introduced. Its "objectness" function exploits deep spatial pyramid features, a novel fast-to-compute HoG-based edge statistic and the EdgeBoxes score [42]. The efficiency is achieved by the use of spatial bins in a novel combination with sparsity-inducing group normalized SVM.*

*State-of-the-art recall performance is achieved on Pascal VOC07, significantly outperforming methods with comparable speed. Interestingly, when only 100 proposals per image are considered the method attains 78% recall on VOC07. The method improves mAP of the RCNN state-of-the-art class-specific detector, increasing it by 10 points when only 50 proposals are used in each image. The system trained on twenty classes performs well on the two hundred class ILSVRC2013 set confirming generalization capability.*

## 1. Introduction

Object detectors have often been applied in the sliding window fashion scoring bounding boxes in all considered positions, scales and aspect ratios using either an inexpensive classifier [13, 7] or cascades [36, 35]. The development of sophisticated and computationally demanding deep learning based object detectors [15, 16] stressed the need to decrease the number of fully scored bounding boxes while retaining high recall levels.

Similar to the first stages of the cascades, object proposals [3, 1, 34] are class-agnostic high-recall-low-precision object detectors that tackle computational efficiency by rejecting likely background regions while retaining bounding boxes covering instances of the semantic object classes which are later classified by the final class-specific object detector.

The authors were supported by the Czech Science Foundation project GACR P103/12/G084 and by the Technology Agency of the Czech Republic TE01020415 V3C – Visual Computing Competence Center.

State-of-the-art proposal methods either generate candidate boxes from image segments, e.g. groups of superpixels or randomly initialized binary segmentation outputs [34, 24, 3, 10, 2], or select proposals from a large pool of densely sampled image regions according to a predefined "objectness" score [1, 27, 5, 40]. The latter approaches, also known as "window scoring" methods [17], utilize diverse types of inexpensive features that most commonly capture edge statistics along the scored region boundaries [27, 42, 5].

In this paper we introduce a method for extraction of object proposals using the window scoring approach. The key novelty is the use of spatial bins [23] in combination with group normalized SVM which enables to carry out the superficially complex proposal score computation surprisingly fast. The proposed objectness function exploits the following sources of information: the deep spatial pyramid features introduced in [16], a novel fast-to-compute HoG-based edge statistic which also takes advantage of the spatial bins and the EdgeBoxes score [42]. Optionally, recall of the method can be boosted by selective search [34] but this slows down the detection slightly.

We experimentally verified that: (1) The introduced method gives state-of-the-art results when comparing the overlap-recall curves. (2) The performance of the state-of-the-art class-specific RCNN detector [15] on our object proposals improves and the performance is less sensitive to the number of used proposals in comparison with other state-of-the-art proposal methods. (3) Despite being trained on a dataset that contains a small set of distinct object classes, it generalizes to previously unseen classes. These factors result in a proposal method that is as fast as standardly used Selective Search in "fast mode" [15, 16] while achieving better recalls.

The rest of the paper is organized as follows. Sect. 2 gives brief information about modern proposal approaches. A concise explanation of our method is provided in Sect. 3. The details about the features we use are in sections 4, 5, 6. An explanation of the utilized feature selection approach

1550-5499/15 $31.00 © 2015 IEEE
DOI 10.1109/ICCV.2015.137

resides in Sect. 7. Sect. 9 explains the special type of non-maximum suppression we employ and Sect. 10 provides results and discussions of concluded experiments. Sect. 11 presents conclusions of our work.

## 2. Related work

Noting that an exhaustive description and evaluation of recent state-of-the-art is presented in Hosang *et al*. [18, 17] a brief explanation of key proposal methods is given in this section.

Many proposal methods build on the seminal Selective Search (SS) of Van de Sande *et al*. [34] which progressively aggregates superpixels obtained by the Felzenszwalb and Huttenlocher method [14] into larger groups based on their similarity. The SS approach still is one of the best in terms of recall and quality of the proposal localization when a large number of candidate windows is requested (more than 1000 per image). Its disadvantage is the inability to select a smaller convenient subset of candidates since it lacks a suitable way of evaluating proposal importance. The relatively slow extraction speed of 10 seconds per image is improved in the "fast mode", accelerating to $\sim$2.5sec/image. However, the accelerated mode looses the high recalls when larger proposal pools are requested. Modifications of Selective Search include Randomized Prim's [24] which learns superpixel similarity measures and employs an order of magnitude faster grouping algorithm. However this comes at the cost of lower attained recalls.

In Constrained parametric min-cuts [3] (CPMC), every proposal is a solution of a min-cut segmentation problem initialized with a random seed. The proposals are ranked on the basis of various types of features. While this approach is able to deliver state-of-the-art recall and localization performance, its speed of a few minutes per image is a significant disadvantage. The approach of Endres and Hoiem [10] bears resemblance to CPMC in the sense that a foreground / background regressor initialized by different seeds is learned for obtaining a set of proposals that are subsequently ranked. The method is slow, about two times faster than CPMC. Multiscale Combinatorial Grouping [2] (MCG) introduced a fast hierarchical segmentation algorithm. On top of that, an efficient exploration of the large combinatorial space of the produced segments is employed in the grouping stage. While the method achieves state-of-the-art performance in terms of recalls it is slow at approx. 30 sec per image.

Rigor [19] address the speed problem of CPMC by reusing max-flow computations. Similarly, Geodesic object proposals [21] replace the min-cut algorithm with a much faster geodesic distance transform seeded by a learned foreground/background regressor. While Rigor has the same speed as Selective Search it has slightly lower recalls. Geodesic proposals run at 1 image/sec and their recall is comparable to Selective Search. However, due to its inability to assign scores to proposals, it is it is not obvious how to limit the number of output candidates.

Rantalankila *et al*. [28] combine the superpixel merging approach [34] with CPMC [3]. The results in [17] indicate that the method is inferior to state-of-the-art both in terms of speed and attained recalls.

Methods based on the sliding window paradigm extract features lying inside predefined bounding boxes and score them using a learned classifier. The work of Alexe *et al*. [1] was the first of this kind. Later Rahtu *et al*. [27] improved [1] by adding more powerful features and by learning a more convenient cascade of structured output SVM classifiers [33]. Additionally, Zhang *et al*. [40] proposed cascade of ranking SVMs that score inexpensive edge-based features. Despite the high speed of these approaches their recall performance is inferior to state-of-the-art [17].

EdgeBoxes (EB) is a fast proposal algorithm, running at 0.3 sec per image, with compelling performance [42]. EB scores proposals using a single feature - the number of contours that are fully enclosed by a bounding box minus those that overlap its boundary. After scoring each region, non-maximum suppression (NMS) takes place. Different overlap thresholds of NMS provide a compromise between accuracy and recall.

BING ([5]) is also based on edge features and provides fairly high recall at low IoU[1] thresholds at the speed of 300 frames per second. However, its performance is significantly inferior to other methods at higher IoU thresholds. This leads to poor performance when used in combination with class-specific object detectors [18]. Moreover, its high recall is more a result of the careful placement of initial bounding boxes than of the discriminative power of the used features and classifier [41].

Deep learning methods have recently entered the field of generic object detectors. DeepMultiBox [11] directly regresses the locations of proposals from an image using a deep convolutional network. Szegedy *et al*. [32] builds on top of [11] and achieves state-of-the-art detection performance on ILSVRC2012 [29]. Although both [11] and [32] evaluate the performance of a class-specific detector that uses their proposals, neither paper presents overlap-recall curves of their generic object detectors preventing comparison with other state-of-the-art proposal methods.

A very recent work of Karianakis *et al*. [20] uses integral channel features detector [8]. The individual channels are filters from the convolutional layers of a deep neural network. This work is perhaps the most similar to our approach. The differences include: (1) the way our deep features are extracted and how the feature selection is carried out. (2) Besides deep features, we use a novel edge-based statistic. (3) We use SVM classifier instead of Ad-

[1]IoU: Intersection Over Union Pascal bounding box overlap metric.

aBoost. (4) Our results are superior in terms of overlap-recall curves.

## 3. Method overview

We selected a window scoring approach since the segmentation based ones are, apart from Selective Search in "fast mode", very slow due to their reliance on superpixel generation or min-cut segmentation algorithms. Since most of the large pool of tested image windows contains background, we employ a well-known paradigm consisting of a cascade of progressively more complex classifiers [36, 35] to introduce an early rejection mechanism. While there are many possible choices for the types of classifiers in the cascade, we utilize binary linear SVM due to its high speed[2].

The first stage of the cascade reduces the initial number of ∼100k of all considered windows roughly by a factor of 10. During the second stage a linear SVM classifier produces final window scores on the basis of computationally more expensive features. The last step consists of a special type of non-maximum suppression (NMS) termed ARNMS that optimizes average recall (AR)[3]. Details about ARNMS are provided in Section 9. The features that describe each bounding box are:

**CNN-SPP:** We follow up on the success of convolutional neural networks on the both object detection and image categorization [22, 15, 31, 16] and use them as our primary bounding box descriptor. To maintain computation efficiency and thus high speed, we employ the fast deep feature extraction technique from [16] that is able to process several thousand bounding boxes per second.

**BEV:** (stage 2 only) Since various edge statistics are a useful objectness cue [27, 1, 42] we introduce a novel Boundary Edge Vector feature (BEV) inspired by the Boundary Edge distribution introduced in [27].

**EB:** Due to an immense speed of the extraction of the EdgeBoxes score [42], we include it as another type of an edge statistic feature.

Additionally, we *speed up extraction of BEV and CNN-SPP features by employing a group-normalized SVM based feature selection algorithm* that automatically collects the set of spatial bins which are the most important for the final classification decision; details are provided in Section 7.

A schematic illustration of our method is presented in Figure 1. In what follows, both classifier stages are discussed in detail. An in-depth explanation of the aforementioned features is provided in sections 4, 5, 6.

### 3.1. Classifier cascade: Stage one

We propose two ways of producing an initial set of regions during the first stage either of which can be used.

**Selective Search + EdgeBoxes70**: Selective Search (SS) [34] regions (using its "fast mode") merged with EdgeBoxes70 (EB70) [42] bounding boxes.

**EdgeBoxes only**: Due to the relative slowness of the Selective Search proposals in comparison with other parts of our pipeline, the second initialization type employs only EdgeBoxes. We set its $\alpha$ parameter controlling the density of the bounding box sampling to a relatively high value of 0.75 to force the generation of an overcomplete pool of regions. EdgeBoxes $\beta$ parameter was set to 1 effectively removing the non-maximum suppression step. This setting produces around 50k regions in 0.5 seconds per image. Subsequently, we take only 30k highest scoring regions according to the EB score. On this set, CNN-SPP descriptors are extracted and appended to the EB scores to form a final stage-1 descriptor which is later scored by an SVM. The ARNMS based on the SVM scores reduces the output to the desired number of 10k boxes.

The second stage is common to both types of stage-one initialization. Two independent versions of our approach can be considered depending only on the chosen initialization type. We term the pipeline that is initialized by EB70 in combination with SS proposals **SSPB+SS** and the pipeline that utilizes EdgeBoxes only **SSPB**.

### 3.2. Classifier cascade: Stage 2

A fixed length descriptor, consisting of the three types of features (EB, BEV, CNN-SPP - described in sections 4, 5, 6) concatenated into a single vector is extracted from each of the 10k bounding boxes and scored using a fast linear binary SVM. The maximum number of 10k input bounding boxes was experimentally found to give good trade-off between the speed and recall. ARNMS which is specifically tuned for the amount of requested proposals is the final step of our method.

## 4. EdgeBoxes feature (EB)

The fastest feature type is the contour score that EdgeBoxes assign to its proposals. Note that in the case of SSPB+SS, besides retaining the score of the extracted EdgeBoxes70, we further use the publicly available EdgeBoxes code to obtain the EB score of the additional Selective Search proposals (without performing the region refinement).

## 5. CNN-SPP feature

The max-pooled activations of the rectified CNN filters coming from the last convolutional layer of the ZF-5 CNN network [39] are another utilized proposal descriptor. This method, originally proposed by He *et al.* [16], rapidly extracts features from spatial bins of several thousand bounding boxes per second. We $\ell_2$ normalize the CNN-SPP fea-

[2]Test showed that the usual choice of AdaBoost [30] is inferior.

[3]The area under the recall-overlap curve evaluated for IoU thresholds ranging from 0.5 to 1 [17]

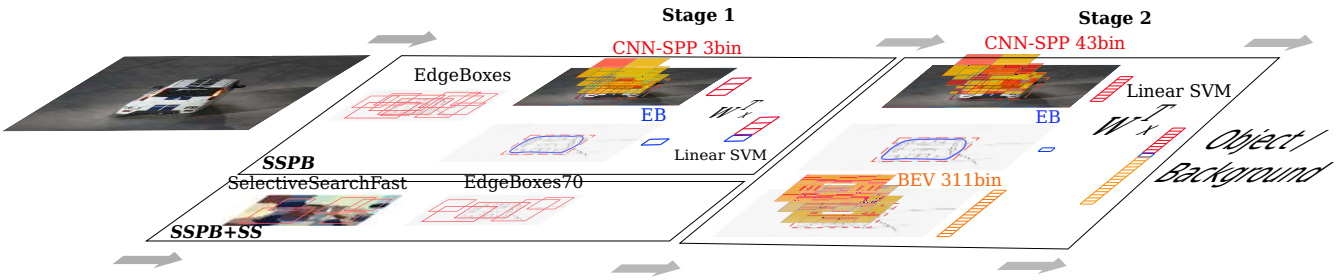



Figure 1. **An overview of our method.** The first stage of the cascaded approach consists of either extracting EdgeBoxes70 together with Selective Search (fast mode) proposals (SSPB+SS) or filtering a large input set of dense EdgeBoxes proposals using SVM that utilizes fast CNN-SPP features (SSPB). During the second stage three descriptor types are extracted from each window and scored by a linear SVM to obtain the final objectness score.

tures to facilitate the convergence of the later used SVM classifier.

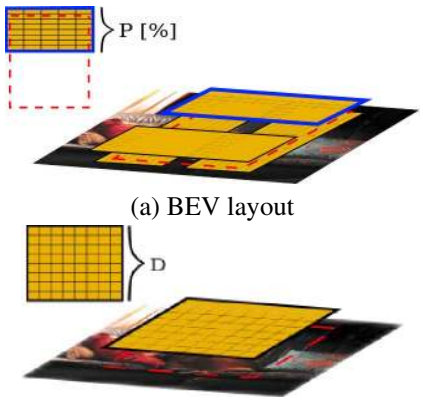
The layout of bounding box subdivisions is the same as in [16], *i.e*. the bounding box is split to $D^2$ equally sized divisions that cover a box uniformly without overlap (Figure 2). We set multiple $D$ parameters such that 10 different split layouts are created corresponding to $D = \{1, 2, 3, ..., 10\}$, giving 385 bins in total. However, in practice *we pool conv5 features only from the bins selected by the feature selection approach* which is thoroughly described in Section 7.

## 6. Boundary Edge Vector feature (BEV)

Boundary Edge Vector exploits the EdgeBoxes edge map (*i.e*. the output of the Structured Edge Detector [9]) for pooling edge statistics inside individual bounding box spatial bins. More precisely, all edgels residing inside a spatial bin are quantized to 4 equally wide orientation bins. After that a 4-dimensional bin descriptor is formed by utilizing integral images to accumulate the edgel intensities that correspond to each of the orientation bins. All these bin descriptors are then concatenated into a single vector which is later $\ell_2$ normalized to form the final BEV descriptor.

The layout of BEV spatial bins is depicted in Figure 2. First, in order to include information about the edges that cross the bounding box boundary, the bounding box dimensions are both enlarged by 10% prior to creating the spatial subdivisions. Then, eight stripes collinear with each of the bounding box sides are all divided across to five divisions to form 40 spatial bins per bounding box side in total. The stripe octet's width is set, such that it covers $P\%$ of the bounding box side. Several different layouts each corresponding to different values of $P$ ($P = \{0.16, 0.18, 0.22, 0.24, 0.28, 0.32, 0.36\}$) are used. Additionally, *feature selection (explained in Section 7) is again used to pick the most informative spatial bins* and thus only the selected ones are again chosen for extraction of the bin descriptors.

The Boundary Edge Vector resembles the Boundary




(a) BEV layout

(b) CNN-SPP layout

Figure 2. **The layout of spatial bins used for pooling descriptors**. (a) BEV is pooled in 40 bins arranged along each of the bounding box sides. (b) CNN-SPP descriptor spatial bins cover the bounding box uniformly without overlap.

Edge distribution (BE) proposed by Rahtu *et al*. in [27]. However, in BE [27], edgels corresponding to only one predefined edgel orientation bin are accumulated inside every spatial bin. Furthermore the accumulated orientation intensities are projected using a predefined set of weights whereas we "unfold" the descriptor into a much higher dimensional vector where all spatial orientations are taken into account. The SVM classifier determines the best weights for each orientation and spatial bin. Finally, we improve the pooling stage, by increasing the number of pooling bins and subsequently learning their optimal layout inside the spatial bin selection algorithm. In the light of these changes our newly introduced feature could be seen as a generalization of the Boundary Edge distribution measure.

## 7. Spatial bin selection

In the case of BEV and CNN-SPP features a large number of spatial bins has to be used in order to obtain state-of-the-art performance. However, this substantially increases

the computational demands. We therefore perform a feature selection step which automatically picks relevant spatial bins that will form the final descriptor.

Our descriptors are created by pooling information from spatial bins, they are formed by groups of values that correspond to spatial subdivisions. To perform selection of bins we use a sparsity-inducing SVM solver [37], that employs the *group lasso* term as a regularizer $\Omega(w)$ [38]. More precisely $\Omega(w) = \sum_{b=1}^{B} \|w_b\|$ where $w$ stands for the set of SVM weights, $w_b$ is the group of weights corresponding to the bin $b$ and $B$ is the overall number of used subdivisions. The value of the $C$ parameter controls the number of zeroed groups $w_b$.

Each spatial bin that corresponds to a group of zeroed SVM weights then plays no role in the final bounding box score and thus could be omitted during the feature extraction step. For BEV and CNN-SPP descriptors the groups of dimensions have size 4 (number of orientation bins) and 256 (number of convolutional filters) respectively.

Our choice of group normalized SVM, instead of e.g. $\ell_1$ regularized SVM which would remove individual descriptor dimensions, is motivated by the computational overheads associated with visiting a single spatial bin: for conv5 features, the spatial bin max-pooling is implemented using SSE instructions thus it is faster to access one continuous block of memory, represented by all convolutional features inside a spatial bin. For BEV, memory addresses to the integral image have to be computed. Thus, by using group normalization, we not only avoid computation of many features but we also decrease the number of costly visits of spatial bins.

Note that the approach consisting of inducing block sparsity to image features was first used in [25] to discover relevant gaussians in the context of Fisher Vector detection pipeline [6, 26].

## 8. SVM and group normalized SVM learning

The standard SVM that combines BEV, CNN-SPP and EB features as well as the group lasso SVM classifiers are learned on the same set of training bounding boxes. The positive examples are all the ground truth regions that contain any of the object classes present in the "train"+"val" sets of the Pascal VOC 2007 detection dataset [12].

The set of negative bounding boxes is composed of two equally sized subsets. While all regions are required to have at most 30% overlap (Pascal intersection-over-union metric) with any of the ground truth objects the first half is sampled from the immediate vicinity of the ground truth regions, while boxes from the second can reside at any location in any training image. The number of negative samples is roughly equal to half of the positive samples.

After the three aforementioned descriptors are obtained from each training region, the sparsity-inducing learning follows. Since the sizes of groups of dimensions that we want to remove are distinct for each of the two feature types (BEV and CNN-SPP), we train two different sparse SVM classifiers separately for each descriptor design. In practice, for the second stage of the detection cascade we select the SVM's regularization parameter such that 43 and 311 spatial regions are selected for CNN-SPP and BEV features respectively. In the case of the first stage of the SSPB classifier, which utilizes CNN-SPP feature, only 3 spatial bins were selected.

After the feature selection step, following [25], training descriptors are stripped of the unused dimensions and $\ell_2$ re-normalized. Additionally, the survivors of the feature selection process are concatenated and the corresponding EB feature is appended to form the final set of training descriptors for the standard $\ell_2$ regularized SVM learning. The $C$ regularization parameter of the $\ell_2$ regularized SVM was set to 1. Hard negative mining tends to worsen the detector performance. We thus stop the pipeline training after the initial mining of random negative samples.

## 9. Non-maximum suppression for optimizing average recall

We discovered that it is suboptimal to perform the standard greedy NMS for discarding redundant high scoring regions, since it tends to either remove many well-located proposals, when its threshold is set to a low value, or completely miss a large portion of regions that are not finely aligned with an object (when the NMS threshold is set to a high value).

To reach a compromise between these two situations, we employ a special type of NMS which we term ARNMS. The goal of ARNMS is to extract a set of candidates that have the best possible average recall given the desired number of output object detections $N$. More accurately, ARNMS runs in $S$ subsequent stages. During stage $s$ the standard greedy NMS is performed with overlap threshold $o_s$ followed by the extraction of $N/S$ highest scoring not suppressed regions. In practice we use $S = 3$ with $o_1 = 1$ (*i.e.* no NMS employed), $o_2 = 0.7$ and $o_3 = 0.5$.

Note that [17] have employed a similar strategy for improving the AR of EdgeBoxes proposals. However their approach is not tuned for specific amounts of outputted proposals thus, when for instance using a very dense sampling of EdgeBoxes during the first stage of our SSPB cascade, the method of [17] would be comparable to running simple greedy NMS with threshold set to 0.9 resulting in low recalls at decreased IoU thresholds.

## 10. Experiments

We test our object proposal methods on two standard object detection benchmarks:

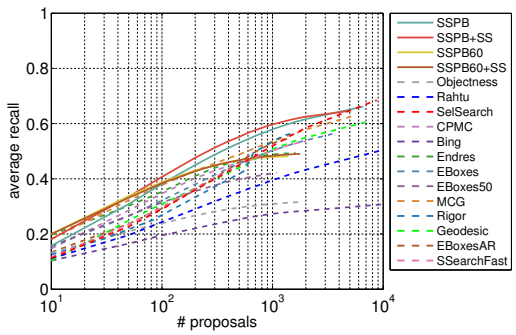



Figure 4. **Average recalls achieved by our (solid lines) and state-of-the-art proposal approaches on VOC07-TEST** as a function of the number of proposals per image.

*VOC07 [12]*: The "test" and in some cases "val" sets of the Pascal VOC2007 dataset were used for evaluation of our methods. The "test" set (VOC07-TEST) consists of 4952 images containing 20 distinct visual object classes together with their bounding box annotations. 2510 images are included in the "val" set (VOC07-VAL) and a similar number of 2501 pictures resides in the "train" set (VOC07-TRAIN).

*ILSVRC2013 [29]*: To check the ability of our method to generalize to unseen data the more challenging ILSVRC2013 DET task's "validation" set was utilized. The amount of images is roughly 20k while there are annotations for 200 object classes.

Note that abbreviations of all competing proposal methods are matched to their original papers in the References section.

**Overlap-recall experiments** In this section, overlap-recall curves obtained using the publicly available benchmark code made by Hosang *et al*. [18], are provided. In case of overlap-recall curves an "oracle" detector that for each ground truth bounding box reports the most overlapping proposal is run. The curve then consists of achieved recalls as a function of minimal required IoU overlaps at which a proposal is regarded as a true positive.

We tested 4 variants of our algorithm. SSPB and SSPB+SS (described in detail in the preceding sections), SSPB60 and SSPB+SS60. SSPB60 differs from SSPB in the final step where ARNMS is replaced by the standard greedy NMS with the overlap threshold set to 0.6. The same applies to SSPB+SS60, which replaces the ARNMS step of SSPB+SS. The two additional methods were introduced because they give compelling performance when a small amount of candidates is requested.

Figure 3 shows the overlap-recall curves of our methods on VOC07-TEST in comparison with state-of-the-art algorithms. Additionally, in Figure 4 we provide average recall measures that have been shown to conveniently quantify the performance of generic object detectors [17].

| | # candidates | | | | | |
|---|---|---|---|---|---|---|
| method | 10 | 50 | 100 | 500 | 1000 | 10000 |
| SSFast | 23.7 | 37.2 | 42.8 | 52.5 | 54.2 | 54.8 |
| EB | 32.3 | 43.0 | 46.1 | 52.1 | 53.3 | 53.1 |
| SSPS (ours) | **36.0** | 46.7 | 50.0 | 53.1 | 56.4 | **56.3** |
| SSPB+SS (ours) | 35.7 | **47.8** | **50.2** | **56.1** | **56.6** | **56.3** |
| DMultiBox [11] | 29.0 | - | - | - | - | - |

Table 1. **RCNN detector mAP as a function numbers of proposals per image** for different proposal methods.

It is apparent that our approach performs better or on par with state-of-the-art both in terms of average recall and the individual recalls achieved at most IoU thresholds. It is rivaled only by the Selective Search ("quality mode") when 10000 candidate windows per image are considered. As noted earlier SSPB and SSPB+SS do not give that impressive performance when a small number of candidates is requested, however the decrease of the non-maximum suppression threshold (SSPB+SS60 and SSPB60) puts our approaches again in the leading position. The comparison between SSPB and SSPB+SS is slightly in favor of SSPB+SS, however we note that SSPB is faster due to the skipping of the Selective Search extraction step. Another positive point is that although SSPB is categorized as one of window scoring methods that tend to attain lower recalls at higher IoU thresholds, it is able to produce bounding boxes comparable to those of *e.g*. MCG or Selective Search in terms of localization quality.

**Combination with a class-specific detector.** To check the applicability of our method, we designed an experiment where the state-of-the-art RCNN [15] class-specific object detector utilizes the output of a proposal generation algorithm. The four proposal algorithms that were tested were SSPB, SSPB+SS, EdgeBoxes70 and Selective Search in its "fast mode" (originally used for RCNN). We recorded the achieved RCNN mAPs on the VOC07-TEST set while varying the number of used candidates per window.

Since we empirically discovered that using a proper IoU threshold when executing the non-maximum suppression of SSPB, SSPB+SS and EdgeBox70 candidate windows is crucial for obtaining the best possible final RCNN performance, we validated these optimal NMS thresholds on the VOC07-VAL set for each number of requested proposals separately. Table 1 shows achieved mAP values.

The results indicate that in case of SSPB and SSPB+SS the RCNN mAP decreases the least as the number of candidates is reduced. Also note that RCNN was originally trained using the Selective Seach "fast" proposals, which typically sways the results in favor of this method [17].Yet, SSPB+SS and SSPB is still able to outperform Selective Search "fast"; additionally our methods improve the results of the original RCNN pipeline when 1000 and more proposals are produced per image.

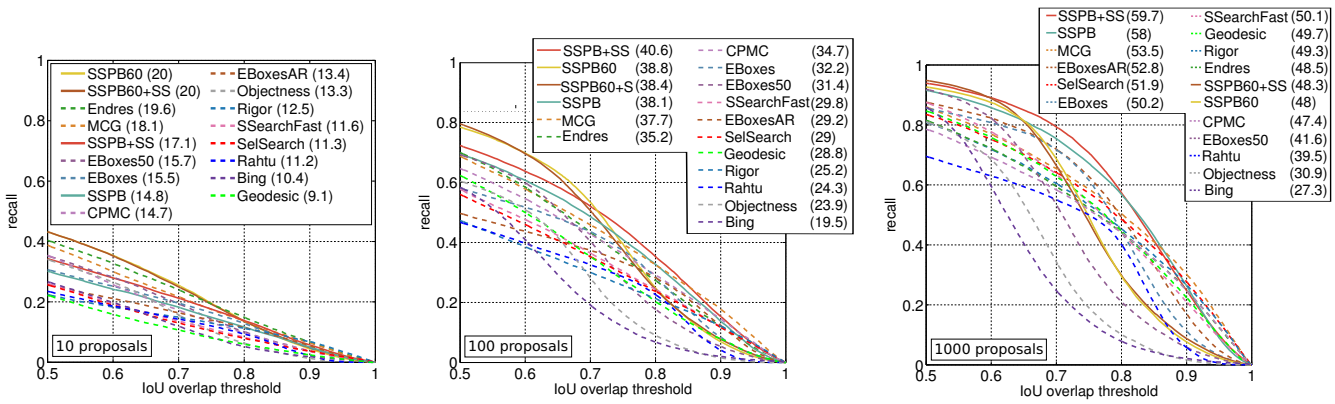



Figure 3. **Overlap-recall curves** of our (solid lines) and state-of-the-art proposal methods on the VOC07-TEST set when 10 (left), 100 (center) and 1000 (right) candidate windows are considered per image. The legends are sorted by the average recalls (in brackets).

Recently DeepMultiBox [11] attained 29.0 mAP on VOC07-TEST with their proposals and using a class-specific detector with CNN architecture from [22] while considering just 10 candidate regions per image. Our result is substantially higher, achieving 36.0 mAP when the same number of SSPB regions is proposed in each image while noting that the architecture of RCNN differs from DeepMultiBox only in the type of classifier in the topmost layer.

**Generalization experiments** Since our method is trained on VOC07-TRAIN+VOC07-VAL, which contains only 20 distinct classes, we verified its performance in a more challenging setting with previously unseen classes. We thus tested the proposed method on the ILSVRC2013 validation set of the detection task.

A potential caveat is that the used ZF-5 network was trained on the ILSVRC2013 training set of the classification task which contains some of the images from the ILSVRC2013 validation set of the detection task used for testing our detector. To overcome this problem we removed the 311 images that are located in both sets and tested our detector on this very slightly reduced set (we refer to it as ILSVRC2013-DET-VAL$_R$). Overlap-recall curves of our and state-of-the-art proposal techniques are plotted in Figure 5, again with the help of the software of [17].

Recently, [4] have shown that the object proposal evaluation protocol could be gamed by training a class specific object detector and use its scores as an objectness measure. We trained an objectness SVM classifier on the 20 class Pascal scores produced by the original CNN-SPP detector [16] according to the training protocol from Section 8. We then apply this classifier to ILSVRC2013-DET-VAL$_R$. The method is labeled "Overfit" in Figure 5.

Results show that our approaches outperform other methods in terms of AR as well as in achieved recalls evaluated between 0.5 and 0.83 overlap thresholds, once a larger amount of candidate windows is considered (more than 500). For the lower proposal amounts SSPB and SSPB+SS stay on par with the competition. Our methods also outperform the Overfit proposals.

**Feature selection experiments** demonstrate the ability of the employed feature selection algorithm to decrease the number of spatial bins while maintaining comparable performance of our approach. We trained two different variations of SSPB+SS on VOC07-TRAIN set and tested them on VOC07-VAL. The first solely utilized the CNN-SPP descriptors, whilst the second used only BEV descriptors. Average recalls as a function of the number of used proposals for various numbers of selected spatial bins are reported.

Figure 6 shows that for CNN-SPP as well as for BEV features, the resulting average recall decreases very slowly as the number of effective spatial bins is reduced. This way it is possible to limit the amount of utilized spatial subdivisions to 25 % and 85 % of the original number in the case of BEV and CNN-SPP features respectively, without hurting the quality of the produced proposals. Figure 6 contains performance of the BE feature [27] to show the improvement of the BEV feature over BE.

**Run-time analysis.** The speed of our methods is compared with the algorithms that attained the best average recalls in experiments. Table 2 shows mean processing times on a fixed subset of 200 images sampled randomly from VOC07-TEST. We report both GPU (GeForce GTX TITAN BLACK) and CPU (Intel Xeon E5-2630 v3) times.

The speed of our approach is comparable to the Selective Search "fast mode" and roughly 4$\times$ faster than its "quality mode", while performing better than the "quality mode". Most of the segmentation methods that perform on par with SSPB and SSPB+SS on ILSVRC2013 and are inferior on VOC07-TEST such as MCG, Endres or CPMC are slower by more than an order of magnitude.

**Ablation analysis.** Figure 7 presents average recall curves achieved on VOC07-TEST with each stage of SSPB while

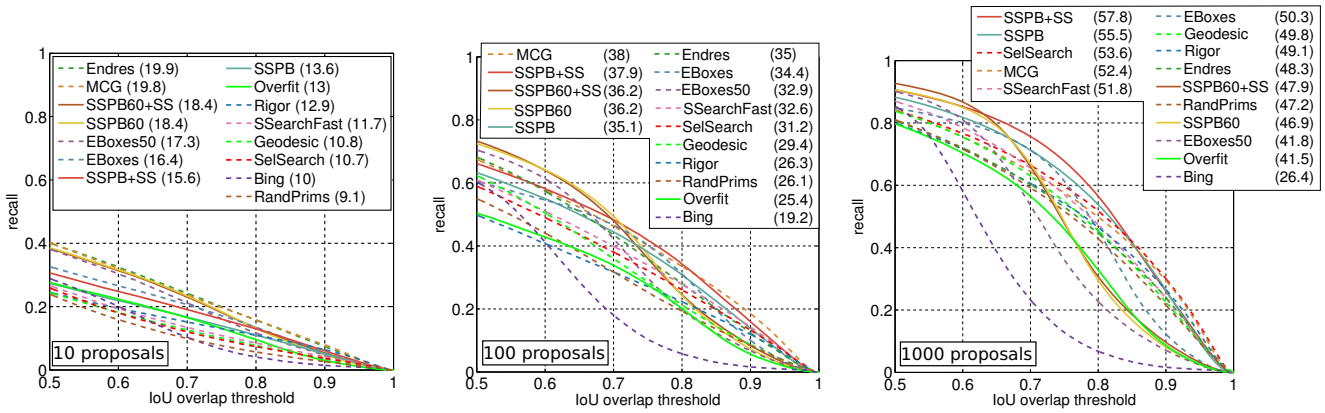



Figure 5. **Overlap-recall curves** of our (solid lines) and state-of-the-art proposal methods on ILSVRC2013-DET-VAL$_R$ set when 10 (left), 100 (center) and 1000 (right) candidate windows are considered per image. The legends are sorted by average recalls (in brackets).

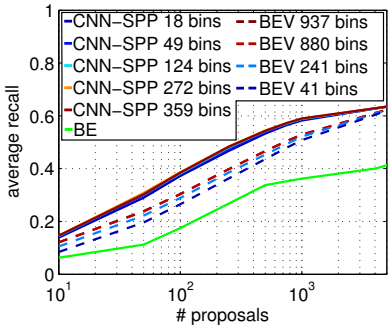



Figure 6. **Average recalls achieved by BEV and CNN-SPP features respectively as a function of the number of proposals** for different numbers of spatial bins selected by group-lasso SVM. The BE feature [27] is included to show its inferiority to BEV.

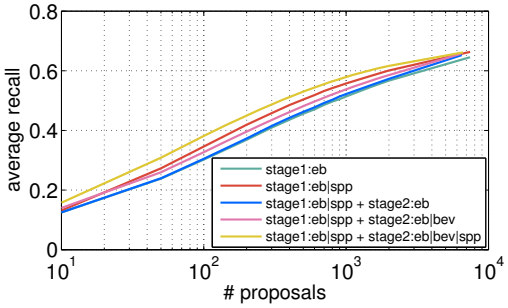



Figure 7. **Ablation analysis.** Performance on VOC07-TEST of both stages of SSPB with different feature combinations.

varying the combination of proposed features. Results show beneficial effects of each added feature/stage.

## 11. Conclusions

We have introduced a novel window scoring method for extraction of object proposals, named SSPB. SSPB uses several fast-to-compute features: deep CNN-SPP features, the EdgeBoxes score and the newly proposed BEV descriptor that accumulates information about edges near bounding box boundaries. We substantially speed up the extraction of these objectness cues by a group normalized SVM based feature selection which does not hurt the final generic object detector performance. The improvement decreased the SSPB processing times below the level of the majority of state-of-the-art proposal approaches.

| Method | time [s] | Method | time [s] |
|---|---|---|---|
| SSFast [34] | 2.52 | MCG† [2] | 30 |
| SSQuality [34] | 11.85 | CPMC† [3] | 250 |
| EdgeBoxes70 [42] | 0.39 | Endres† [10] | 100 |
| SSPB (ours) | 3.16 (11.51) | Rigor† [19] | 10 |
| SSPB+SS (ours) | 4.09 (12.45) | Geodesic† [21] | 1 |

†results taken from [17]

Table 2. **Per image processing times** of our methods and the best performing competition. For SSPB and SSPB+SS, GPU was used for extraction of conv5 features. Full-CPU times are in brackets.

Results on the Pascal VOC2007 dataset indicate that our method delivers state-of-the-art in average recalls and at recall levels at many IoU thresholds for various numbers of candidate windows per image. We obtained similar results on ILSVRC2013. Since SSPB was trained on Pascal, the positive results prove that our method generalizes to previously unseen data.

Our proposals work very well in combination with the current state-of-the-art class-specific object detector RCNN [15]. Besides significantly improving RCNN mAP when the number of considered candidates is limited, higher numbers of SSPB proposals also slightly increase class-specific detection above the level of Selective Search.

## References


[1] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, 2010. Objectness 1, 2, 3

[2] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping. In *CVPR*, 2014. MCG 1, 2, 8

[3] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *CVPR*, 2010. CPMC 1, 2, 8

[4] N. Chavali, H. Agrawal, A. Mahendru, and D. Batra. Object-proposal evaluation protocol is 'gameable'. *arXiv:1505.05836*, 2015. 7

[5] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *CVPR*, 2014. Bing 1, 2

[6] R. G. Cinbis, J. Verbeek, and C. Schmid. Segmentation driven object detection with fisher vectors. In *ICCV*, 2013. 5

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

[8] P. Dollár, Z. Tu, P. Perona, and S. Belongie. Integral channel features. In *BMVC*, 2009. 2

[9] P. Dollár and C. L. Zitnick. Structured forests for fast edge detection. In *ICCV*, 2013. 4

[10] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*. 2010. Endres 1, 2, 8

[11] D. Erhan, C. Szegedy, A. Toshev, and D. Anguelov. Scalable object detection using deep neural networks. In *CVPR*, 2014. DeepMultiBox 2, 6, 7

[12] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 5, 6

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 2010. 1

[14] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004. 2

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1, 3, 6, 8

[16] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. 2014. 1, 3, 4, 7

[17] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *CoRR*, 2015. 1, 2, 3, 5, 6, 7, 8

[18] J. Hosang, R. Benenson, and B. Schiele. How good are detection proposals, really? *BMVC*, 2014. 2, 6

[19] A. Humayun, F. Li, and J. M. Rehg. Rigor: Reusing inference in graph cuts for generating object regions. In *CVPR*, 2014. Rigor 2, 8

[20] N. Karianakis, T. J. Fuchs, and S. Soatto. Boosting convolutional features for robust object proposals. *arXiv:1503.06350*, 2015. 2

[21] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *ECCV*. 2014. Geodesic 2, 8

[22] A. Krizhevsky, I. Sutskever, G. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, 2012. 3, 7

[23] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 1

[24] S. Manen, M. Guillaumin, and L. V. Gool. Prime object proposals with randomized prim's algorithm. In *ICCV*, 2013. RandPrim's 1, 2

[25] D. Novotný, D. Larlus, F. Perronnin, and A. Vedaldi. Understanding the fisher vector: a multimodal part model. *arXiv:1504.04763*, 2015. 5

[26] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*. 2010. 5

[27] E. Rahtu, J. Kannala, and M. Blaschko. Learning a category independent object detection cascade. In *ICCV*, 2011. Rahtu 1, 2, 3, 4, 7, 8

[28] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *CVPR*, 2014. 2

[29] O. Russakovsky et.al. ImageNet Large Scale Visual Recognition Challenge, 2014. 2, 6

[30] R. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictions. *Machine learning*, 1999. 3

[31] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *ICLR*, 2013. 3

[32] C. Szegedy, S. Reed, D. Erhan, and D. Anguelov. Scalable, high-quality object detection. *CVPR*, 2014. 2

[33] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun. Support vector machine learning for interdependent and structured output spaces. In *ICML*, 2004. 2

[34] K. E. Van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, 2011. SSearchFast, SelSearch 1, 2, 3, 8

[35] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 1, 3

[36] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 1, 3

[37] H. Yang, Z. Xu, I. King, and M. R. Lyu. Online learning for group lasso. In *IMCL*, 2010. 5

[38] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *J. R. Stat. Soc. Ser. B*, 2006. 5

[39] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*. 2014. 3

[40] Z. Zhang, J. Warrell, and P. Torr. Proposal generation for object detection using cascaded ranking svms. In *CVPR*, 2011. 1, 2

[41] Q. Zhao, Z. Liu, and B. Yin. Cracking bing and beyond. In *BMVC*, 2014. 2

[42] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*. 2014. EBoxes,EBoxes50,EBoxesAR 1, 2, 3, 8