Object Proposal Generation using Two-Stage Cascade SVMs

Ziming Zhang, and Philip H.S. Torr, Senior Member, IEEE

Abstract—Object proposal algorithms have shown great promise as a first step for object recognition and detection. Good object proposal generation algorithms require high object detection recall rate as well as low computational cost, because generating object proposals is usually utilized as a preprocessing step. The problem of how to accelerate the object proposal generation and evaluation process without decreasing recall is thus of great interest. In this paper, we propose a new object proposal generation method using two-stage cascade SVMs, where in the first stage linear filters are learned for predefined quantized scales/aspect-ratios independently, and in the second stage a global linear classifier is learned across all the quantized scales/aspect-ratios for calibration, so that all the windows from the first stage can be compared properly. The windows with highest scores from the second stage are kept as inputs to our new efficient proposal calibration algorithm to improve their localization quality significantly, resulting in our final object proposals. We explain our scale/aspect-ratio quantization scheme, and investigate the effects of combinations of $\ell_1$ and $\ell_2$ regularizers in cascade SVMs with/without ranking constraints in learning. Comprehensive experiments on VOC2007 dataset are conducted, and our method is comparable with the current state-of-the-art methods with much better computational efficiency.

Index Terms—Object proposal generation, Scale/Aspect-ratio quantization, Cascade SVMs, Linear filters, 2D convolution

1 INTRODUCTION

For object proposal generation, we are interested in providing a small set of windows (i.e. bounding boxes) containing object instances probably with high object detection recall as well as high computational efficiency. Recent research has demonstrated that object proposal, as a data pre-process step, can be involved successfully in complex computer vision systems to help reduce the computational cost significantly while achieving state-of-the-art performance, e.g., in object recognition [1] and object detection [2]. In these methods, a small number of object proposals are needed to summarize all the objects in images that will be utilized further by the methods. Therefore, the need to accelerate the evaluation process as well as achieving high object recall is thus becoming more important for a successful computer vision system, and this problem has been attracting more and more attention [3], [4], [5], [6], [7], [8], [9].

The main difficulties in object proposal generation are three-fold. First, the search space for localizing object proposals may be huge: Take a $(W \times H)$-pixel image for example. Considering all possible locations and scales/aspect-ratios in the image, the number of proposal candidates is roughly $O(W^2H^2)$. Second, finding a proper object representation is challenging, because of the change of imaging factors, huge intra-class and inter-class variations, many object categories, etc. Third, there may be multiple correct proposals for a single object instance of interest, leading to unnecessary spatial clusters of proposals. Thus, developing a highly computationally efficient yet accurate object proposal generation algorithm becomes very challenging.

Our previous work appeared as [10], where we proposed a ranking based two-stage cascade model for class-specific object proposal generation. To reduce the search space, we first proposed a scale/aspect-ratio quantization scheme in log-space, which guarantees any possible instance of objects in images can be located using at least one bounding box defined in the scheme. Then we learn linear classifiers at each stage in our cascade, all of whose scores can be utilized for ranking purposes. Ranking support vector machines (SVMs) [11] are used for ranking the windows, which are normal SVMs with additional ranking constraints added into the learning to guarantee that some data should be classified with a higher score than others based on the ground-truth ranking order (e.g. those windows that better overlap the object ground-truth bounding boxes). In this way, our two-stage cascade enables us to incorporate variability in scale and aspect ratio by training a linear classifier for each quantized scale/aspect-ratio in the first stage, and another linear classifier in the second stage to calibrate the scores of the windows proposed from the first stage for final proposals. Finally, the usage of simple gradient features, linear convolution, and non-max suppression makes our method achieve the state-of-
the-art performance in terms of object recall vs. number of proposals with high computational efficiency. Fig. 1 summarizes the cascaded model and gives an example of generating proposals using this method.

This paper extends our work in [10]. Specifically, we
- explain in detail our scale/aspect-ratio quantization scheme;
- investigate more general usage of cascade SVMs by exploring the effects of combinations of \( \ell_1 \) and \( \ell_2 \) regularizers in two-stage cascade SVMs with/ without ranking constraints in learning;
- propose a new efficient proposal calibration algorithm as a post-processing step to improve proposal localization quality;
- and demonstrate the capability of our method for generic object proposal generation.

The rest of the paper is organized as follows. We first review some related work in Section 2. Then we explain the details of scale/aspect-ratio quantization scheme in Section 3. Next we formulate our two-stage cascade SVMs based on the proposed scale/aspect-ratio quantization scheme in Section 4. Further we propose a new proposal calibration algorithm as a post-processing step to improve the localization of our proposals in Section 5. We list some implementation details in Section 6. Finally Section 7 shows our experimental results and Section 8 concludes the paper.

2 RELATED WORK

Various methods have been proposed to handle the proposal generation problem. Branch and bound techniques [6], [9] for instance limit the number of windows that must be evaluated by pruning sets of windows whose response can be bounded. The efficiency of such methods is highly dependent on the strength of the bound, and the ease with which it can be evaluated, which can cause the method to offer limited speed-up for non-linear classifiers. Alternatively, cascade approaches [5], [7], [8] use weaker but faster classifiers in the initial stages to prune out negative examples, and only apply slower non-linear classifiers at the final stages. In [5] a fast linear SVM is used as a first step, while the jumping window approach [7] builds an initial linear classifier by selecting pairs of discriminative visual words from their associated rectangle regions. Felzenszwalb et al. [12] propose a part-based cascaded model using a latent SVM in which part filters are only evaluated if a sufficient response is obtained from a global “root” filter, and [8] propose a combination of cascade and branch and bound techniques. Such approaches have been proved to be efficient, and have generated state-of-the-art results [12]. However, the fact that in [8] the decision scores for detections must be compared across the training data may limit the efficiency of the early cascade stages, where we only need to compare the scores of a classifier at any level of the cascade within a single image. Further, such approaches learn a single model which is applied at varying resolutions. Recent work [13] strongly suggests that we should explicitly learn different detectors for different scales.

Several recent works (e.g. [3], [4], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23]) are closely related to ours. Objectness measure [4] combines multiple visual cues to score the windows, and then produces the object proposals by sampling windows with high scores. Based on [4], Rahtu et al. [3] proposed another category-independent cascaded method for proposal generation, where the proposal candidates are sampled from super-pixels, which are generated using a segmentation method, according to a prior object localization distribution and then ranked using structured learning with learned features. Further in [14], Blaschko et al. investigated the effect of the non-max suppression step in [3] to improve the performance.

The idea of grouping super-pixels/segments to generate proposals is widely used in the literature with different grouping criteria [15], [16], [17], [18], [19], [20], [21]. Carreira and Sminchisescu [15] proposed to generate object hypotheses by solving a sequence
of Constrained Parametric Min-Cut problems (CPMC) on a regular image grid. Uijlings et al. [16] proposed selective search by combining the strength of both an exhaustive search and segmentation and being guided by the image structure. Manen et al. [17] proposed a randomized version of Prim’s algorithm on the superpixel connectivity graphs. Endres and Hoiem [18] proposed to rank the regions (i.e., a set of segments) using structured learning based on various cues. Yanulevskaia et al. [19] proposed to group regions in a hierarchical fashion based on Random Forest. Rantalankila et al. [20] proposed to build a bottom-up segmentation hierarchy by greedily merging adjacent parts of superpixels and also perform graph cut on a superpixel graph. Krähenbühl and Koltun [21] proposed to identify critical level sets in geodesic distance segmentation hierarchy by greedily merging adjacent pairs of superpixels and also perform graph cut on a superpixel graph. Correspondingly, with simple gradients and edges as features, our method achieves much better computational efficiency, leading to relatively long running time during testing.

To speedup the proposal generation algorithms, different features, rather than segments, have been proposed. Zitnick and Dollár [22] proposed the EdgeBox algorithm to fast generate proposals based on edges and contours while achieving good localization quality. Cheng et al. [23] proposed the BING algorithm with binary features so that it can run 300fps as well as achieving reasonable localization quality. More empirical comparisons of different object proposal generation methods can be found in [24].

To distinguish our method from these related work above, we propose

- a scale/aspect-ratio quantization scheme to reduce the proposal searching space from linear-space to log-space,
- a systematic yet efficient approach to learn linear models with much fewer tricks and parameters,
- and an efficient proposal calibration algorithm to improve the proposal localization quality significantly.

Correspondingly, with simple gradients and edges as features, our method achieves much better computational efficiency with comparable object detection recall rates to the current state-of-the-art methods.

3 Scale/Aspect-Ratio Quantization Scheme

3.1 Preliminaries

Before explaining the details of our scale/aspect-ratio quantization scheme, we first introduce some definitions that are used later.

**Definition 1 (Bounding Box Overlap Score).** The overlap score between a bounding box \( t \) and a ground-truth bounding box of an object \( s \), \( o(t,s) \), is defined as their intersection area divided by their union area. Clearly, \( 0 \leq o(t,s) \leq 1 \), and the higher \( o(t,s) \) is, the better the localization of the object \( s \) with the bounding box \( t \) is.

**Definition 2 (\( \eta \)-Accuracy).** We say that a window \( t \in T \) can be localized by another window \( s \in S \) to \( \eta \)-accuracy if \( o(t,s) \geq \eta \), \( 0 \leq \eta \leq 1 \).

**Definition 3 (Maximum Overlap).** Given an image \( I \) and the ground-truth bounding boxes of multiple objects \( g_1 \ldots g_m \) in \( I \), the maximum overlap of a window \( t \) in \( I \) is defined as \( o_t = \max_{g \in \{g_1 \ldots g_m\}} o(t,g_i) \), where \( o(t,g_i) \) denotes the overlap score between \( t \) and \( g_i \).

**Definition 4 (Correct Object Proposals).** Given an overlap score threshold \( \eta \), a window \( t \) is considered as a correct object proposal in an image if and only if \( o_t \geq \eta \).

**Definition 5 (Quantized Scale/Aspect-ratio).** Given an overlap score threshold \( \eta \), a window \( t \) in an image can be quantized into a quantized scale/aspect-ratio \( S \) if and only if \( \exists s \in S \) such that \( o(t,s) \geq \eta \), where \( s \) is a window with the quantized scale/aspect-ratio.

For ease of explanation of our approach, we list the main notation in the following sections in Table 1.

### 3.2 Quantization Scheme

We design our quantization scheme so that in each image any window \( t \in T \) can be represented by at least one window \( s \in S \) in our quantization scheme.

**Fig. 2** gives an intuitive representation of our scheme. Given the smallest size (width and height) of windows in the scheme \( (w_0, h_0) \), we include in our scheme all quantization levels of the form \( S(w_0/\eta^a, h_0/\eta^b) \), where \( a \in \{0,1,\ldots,A\} \) and \( b \in \{0,1,\ldots,B\} \) are naturally limited by the image size.
Fig. 3. An example of our method [10] on demonstrating the localization quality with increase of the number of quantized scales/aspect-ratios, \( K \), on the VOC2006 [25] dataset using the object recall-overlap evaluation. Recall-overlap curves are plotted for individual classes, based on Eq. 1, where \( \eta \) is set to 0.5. For more details, please refer to [10].

As a result, the quantization levels can be thought of as forming a tree structure, as illustrated in Fig. 2.

Next, we will introduce some very important properties of our quantization scheme to explain its essence of reducing the search space for object proposal generation.

**Proposition 1 (Existence of Quantization Scheme).**
Given an overlap score threshold \( \eta \) and a minimum size \((w_0, h_0)\) of objects that can be found in images, any window \((w, h)\) with \( w \) and \( h \) can be localized to \( \eta \)-accuracy by at least one window \( s(w_0, h_0) \) in our scale/aspect-ratio quantization scheme with parameter \( \eta \leq \bar{\eta} < 1 \).

Proof: According to Fig. 2, we can construct a window \( s((w_0, h_0)^a, (w_0, h_0)^b) \) \((a, b \in \{0, -1, -2, \ldots \})\) with quantized scale/aspect-ratio \((w_0, h_0)^a\). Based on Definition 1, we can calculate the overlap between windows \( t \) and \( s \) as follows:

\[
o(t, s) \geq \min \{w_t, w_0^a\} \cdot \min \{h_t, h_0^a\} \cdot \max \{w_t, w_0^b\} \cdot \max \{h_t, h_0^b\}\]

That is, \( t \) can be localized to \( \eta \)-accuracy by \( s \).

**Proposition 2 (Sufficient Number of Quantized Scales/Aspect-ratios).**
Given an overlap score threshold \( \eta \), a minimum size \((w_0, h_0)\) and a maximum size \((w, h)\) of objects that can be found in images, the number of quantized scales/aspect-ratios that is sufficient to localize any object to \( \eta \)-accuracy is bounded by

\[
(1 + \log_{\eta} \frac{w}{w_0}) (1 + \log_{\eta} \frac{h}{h_0}).
\]

Proof: Let the smallest quantized scale/aspect-ratio in our scheme is \((w_0, h_0)\). Based on the proof in Proposition 1, we can construct a scale/aspect-ratio quantization scheme which limits \( a \in \{0, \ldots, \log_{\eta} \frac{w}{w_0}\} \) and \( b \in \{0, \ldots, \log_{\eta} \frac{h}{h_0}\} \). Therefore, the number of quantized scales/aspect-ratios that is sufficient to localize all possible objects in images is bounded by

\[
(1 + \log_{\eta} \frac{w}{w_0}) (1 + \log_{\eta} \frac{h}{h_0}).
\]

**Proposition 3 (Search Space for Object Localization).**
Given an overlap score threshold \( \eta \), the minimum size of quantized scale/aspect-ratio \((w_0, h_0)\), and the maximum image size \((W, H)\), the search space for localizing an arbitrary object in images to \( \eta \)-accuracy using quantized windows is

\[
O(W \cdot \log_{\eta} \frac{w}{w_0} \cdot H \cdot \log_{\eta} \frac{h}{h_0}).
\]

Proof: According to Proposition 2, the search space for scales/aspect-ratios of objects is reduced to \(O([\log_{\eta} \frac{w}{w_0}] [\log_{\eta} \frac{h}{h_0}])\) using our quantization scheme, while the search space for positions of objects keeps the same \( O(W \cdot H) \) as sliding window methods. Therefore, the search space for object localization using our scheme is

\[
O(W \cdot \log_{\eta} \frac{w}{w_0} \cdot H \cdot \log_{\eta} \frac{h}{h_0})).
\]

3.3 Discussion

3.3.1 Object Representation

Instead of constructing larger and larger quantized scales/aspect-ratios in the quantization scheme, we utilize a same small window size (e.g. 8 x 8 pixel windows) for all the quantized scales/aspect-ratios by rescaling images accordingly. In this way, we represent all possible objects in images using a fixed small window size.

The intuitions behind this image rescaling are as follows. The objects of interest in images are usually well-defined with clear boundary (i.e. high-contrast
edges) between them and background. At low resolution, these high-contrast edges preserve the discrimination between objects and background, while the details inside the object regions become blur or even fade away. This allows us to avoid modeling very complex object variations, making every object instance look similar to each other. Our method indeed tries to localize these boundary information using linear filters. In our recent work [23], this intuition was shared as well.

Note that we did compare the performance using the quantization scheme illustrated in Fig. 2 with that using our image-scaling quantization scheme, and the former is worse than the latter in terms of not only object recall but, more importantly, computational efficiency. The reason is that with larger quantized scales/aspect-ratios (i.e. higher resolutions) the appearance details of objects become more important to distinguish them from the background, which brings much trouble in designing an efficient generic object representation such like object boundary.

### 3.3.2 Localization Quality

From Proposition 1, we can see that the localization quality of a given quantization scheme is dependent on the parameter \( \bar{\eta} \) (not overlap threshold \( \eta \)), which is chosen to construct the quantization scheme. For instance, in VOC object detection challenges, the overlap score threshold for correct localization is set to 0.5, i.e. \( \eta = 0.5 \). However, to construct our quantization scheme, we can choose an arbitrary value for the parameter \( \bar{\eta} \) as long as \( \eta \leq \bar{\eta} < 1 \), say \( \bar{\eta} = 0.6 \). Then our method can generate better object proposals than those using \( \bar{\eta} = 0.5 \), in general. In order to generate proposals with better localization, we have to create more quantized scales/aspect-ratios (based on Proposition 2), leading to larger search space and higher computational cost accordingly (based on Proposition 3).

We have verified this situation in [10], Fig. 3 is cited from [10], where \( K \in \{36, 121, 196\} \), the total number of quantized scales/aspect-ratios in our scheme, corresponds to \( \bar{\eta} \in \{ \frac{1}{3}, \frac{2}{3}, \frac{3}{3} \} \), respectively, for constructing the quantization schemes. As we see, with increase of \( K \), all the curves are pushing towards the top-right corner, in general. This indicates that increasing \( K \) does help localize objects better, with observations of larger area-under-the-curve (AUC) scores. On the \( K = 36 \) graph, we see that the curves are high for \( \eta \leq 0.5 \), but then drop quickly. However, the curves for the \( K = 121 \) and \( K = 196 \) drop at the corresponding later points, around \( \eta = 0.6 \), leading to better localization and thus partially demonstrating Proposition 1. Fig. 4 is also cited from [10], showing that larger \( K \) does result in higher computational time under the same parameter setting.

### 4 Two-stage Cascade SVMs

Cascade classifiers have a decade history in object detection [26], [27], [28], [29], especially the very successful Viola and Jones’s method for face detection [26]. Cascaded classifiers are good tools for handling extremely imbalanced data, that is, too many negatives and too few positives. Object detection is one of the applications with extremely imbalanced data, where the objects of interest in an image are very few but the non-object are many, considering the huge structural search space of windows. In the cascade, only “positives” are passed on as outputs of each stage, which have higher ranks than those “negatives”.

In our training data, each image is annotated with the bounding boxes of the objects of interest. Our goal is to give higher ranks to the correct object proposals, given the overlap score threshold parameter \( \eta \), than the wrong ones in a very efficient way, such that the windows at the top of the ranking list can be taken as our final object proposals.

#### 4.1 Stage I: Scale/Aspect-ratio Specific Ranking

The first stage of our cascade aims to pass on a number of object proposals based on different sliding windows at each of a set of quantized scales and aspect ratios to the next stage. This is done by learning a linear classifier for each quantized scale/aspect-ratio separately.

#### 4.1.1 Individual Classifier Learning

Given \( \eta \) and a set of quantized scales/aspect-ratios, for each scale \( k \) we wish to learn a linear classifier \( f_k(x_t; w_k) = w_k \cdot x_t \), as suggested in [13], to rank the window \( t \in T_k \), whose feature vector is denoted as \( x_t \), among all the windows in \( T_k \).

Ideally, we expect that within image \( I \) the ranking score for any window \( t_i \in T_k \cap T_l \) with \( \alpha_i \geq \eta \) is

1. In the following sections, we refer to scale \( k \) as quantized scale/aspect-ratio \( k \) for short.
always higher than that of any window \( t_j \in \mathcal{T}_I \) with \( a_{ij} < \eta \). That is, for \( w_k \) we require that within the image \( I \) all the corresponding positive training windows \( \mathcal{I}_k = \{ t_i \in \mathcal{T}_I | a_{ti} \geq \eta \} \) should be ranked above all the training negatives \( \mathcal{I}^- = \{ t_i \in \mathcal{T}_I | a_{ti} < \eta \} \). Naturally this leads us to formulate the problem as a ranking SVM as follows:

\[
\min_{w_k, \xi} \frac{1}{p} \left\| w_k \right\|^p_p + C \sum_{i,j,n} \xi_{ij}^n
\]

\[
\text{s.t.} \quad \forall n, i \in \mathcal{I}_k^+, j \in \mathcal{I}_n^-, w_k \cdot (x_i^n - x_j^n) \geq 1 - \xi_{ij}^n,
\]

\[
\xi_{ij}^n \geq 0, \quad p \in \{1,2\}.
\]

Here, \( x_i^n \) and \( x_j^n \) are the feature vectors associated with positive window \( i \) and negative window \( j \) in training image \( I_n \) respectively, \( \xi = \{ \xi_{ij}^n \} \) are the slack variables, \( C \geq 0 \) is a predefined regularization parameter, and \( \| \cdot \|_p \) denotes the \( \ell_p \) norm of vectors.

Recall that the purpose of learning the individual classifier is to build the proposal pool for further usage, so the constraints in Eq. 3 are restricted to one quantized scale in one image. Therefore, the (local) ranking scores from each classifier are incomparable across scales/aspect-ratios, necessitating the second stage in the cascade.

**Remarks:** In order to make Eq. 3 more general, we introduce a dummy feature \( 0 \) and define that its rank is higher than negatives but lower than positives. Then only comparing positive/negative features with the dummy feature turns Eq. 3 into a standard SVM without ranking constraints. We denote the solution of Eq. 3 with ranking constraints as “\( \ell_p\)-w/r” and the solution of Eq. 3 without ranking constraints as “\( \ell_p\)-w/o/r”, respectively.

Differently from some other ranking objective functions in object proposal generation such like [3] which learns a ranker for all the scales/aspect-ratios, here we learn a classifier/filter individually for each quantized scale/aspect-ratio. The filter responses for different quantized scale/aspect-ratio are not comparable directly since the constraints in Eq. 3 do not contain the ranking comparison among different quantized scales/aspect-ratios. That is why we need a second ranking stage in our method.

### 4.1.2 Proposal Selection with Non-Max Suppression

To decide which proposals to forward from the first stage to the second of the cascade, we look for the local maxima in the response image of classifier \( \omega_k \) as illustrated in Fig. 1(i,c), and set a threshold on the maximum number of windows to be passed on. The first stage thus has two controlling parameters. The first, \( \gamma \in [0,2] \) specifies the ratio between the size of the neighborhood over which we search for the local maxima, and the reference window size for each classifier. This is the non-max suppression parameter. The second, \( d_1 \in \{1, \ldots , 1000\} \) specifies the maximum number of windows, which are the top \( d_1 \) ranked local maxima, as illustrated in Fig. 1(i,d), that can be passed on from any scale. This non-max suppression step is utilized to deal with the difficulty of multiple correct proposals per object.

### 4.2 Stage II: Ranking Score Calibration

The first stage of the cascade generates a number of proposal windows at each scale \( k \) for image \( I \). The second stage then re-ranks these windows globally, so that the best proposals across scales are forwarded. To achieve this, we introduce a new feature vector for each window \( v \), which consists of the segment responses of the classifier at the first stage. Fig. 5 illustrates the final ranking score calculation process, where a 12-dim feature vector \( x_i \) is divided into four 3-dim vectors (i.e. segments), and by multiplying with their counterparts in \( w_k \), the learned filter at Stage I, the outputs (i.e. segment responses) form a new 4-dim vector \( v_i \). Each segment may have a physical meaning, e.g. the gradient orientation. At Stage I the calibration weight for each segment is equal to 1. However, since we cannot compare the ranking scores of outputs from Stage I directly, we need to learn new segment calibration weights \( z_k \)'s at Stage II. Then the calibrated ranking score is taken as the inner product between \( v_i \) and \( z_k \).

Based on \( v \), we can re-rank each window \( i \) by the decision function \( f(v_i) = z_k \cdot v_i + e_{k_i} \), where \( k_i \) denotes the quantized scale/aspect-ratio associated with window \( i \), \( z_k \) is a set of coefficients for scale \( k \) that we would like to learn, and \( e_{k_i} \) is the corresponding bias term. Similarly, we formulate this learning problem as a multi-class ranking SVM as shown in Eq. 4:

\[
\min_{x, e, \xi} \frac{1}{p} \sum_{k_i} \left\| z_k \right\|^p_p + C \sum_{i,j,n} \xi_{ij}^n
\]

\[
\text{s.t.} \quad \forall n, i \in \mathcal{I}_k^+, j \in \mathcal{I}_n^-, z_k \cdot v_i^n - z_k \cdot v_j^n + e_{k_i} - e_{k_j} \geq 1 - \xi_{ij}^n,
\]

\[
\xi_{ij}^n \geq 0, \quad p \in \{1,2\}.
\]
I
ˆ
Here, the red solid bounding box denotes the ground-truth annotation for “dog”, the green dotted bounding box denotes an output proposal from the two-stage cascade SVMs, and the cyan solid bounding box denotes the corresponding output proposal after proposal calibration. The original image is from VOC2007 [30].

Fig. 6. Illustration of the effect of our proposal calibration, where the red solid bounding box denotes the ground-truth annotation for “dog”, the green dotted bounding box denotes an output proposal from the two-stage cascade SVMs, and the cyan solid bounding box denotes the corresponding output proposal after proposal calibration. The original image is from VOC2007 [30].

4.3 Worst-case Computational Complexity

Our method involves the application of simple linear classifiers to the images, and as such is dominated by the complexity of 2D convolution which has to be applied to each image. The worst-case complexity can thus be approximated as \( O(K \cdot R \cdot (WH) \cdot (W_1H_1)) \), where \( K \) denotes the number of individual classifiers learned in Stage I, \( R \) denotes the number of segments used in Stage II, \( (W, H) \) denotes the filter size, and \( (W_1, H_1) \) denotes the resized image size. We note that our complexity is therefore (largely) independent of the number of potential proposals let through at each stage \((d_1, d_2)\), unlike methods which include non-linear classifiers [6], [5]. Also, our algorithm is quite suitable for parallel computing, which will reduce the running time dramatically.

5 Proposal Calibration

Though we can guarantee to localize an object instance to \( \eta \)-accuracy with our proposed scale/aspect-ratio quantization scheme, sometimes the localization quality of the proposals from the two-stage cascade SVMs is poor due to the limited number of quantized scales/aspect-ratios. Fig. 6 shows an example of such case. However, after applying our proposal calibration to the output proposals as a post-processing step, it is clear that their localization quality is improved a lot.

The basic idea here is: since we would like to locate the contours of objects (i.e. the intuition of our method), we can utilize the edges in images to approximate the contours, and keep updating the boundaries of the output proposals based on the edges. As shown in 6(b), the left side of the green dotted bounding box is far away from the contour of the dog, resulting in large irrelevant area \textit{w.r.t.} the ground-truth. Our proposal calibration algorithm pushes the left side towards the closest edges, and simultaneously updates the rest sides. The final proposal of our method is shown as the bounding box with cyan color.

The proposal calibration algorithm is listed in Alg. 1, where functions “edge”, “ind2sub”, “size”, “min”, “max” follow the MATLAB function formats, and matrices \( I, P, E, \) Neighbors, \( X, Y \) follow the MATLAB matrix format as well. Give the image size as \((W_1, H_1)\), the computational complexity of edge detection can be as low as \( O(W_1H_1 \cdot \log(W_1H_1)) \) (e.g. canny edge detector), and the complexity of distance transform [31] is \( O(W_1H_1) \).

Therefore, compared with the computational complexity of the two-stage cascade SVMs in Section 4.3, during testing the overall complexity of our method is dominated by the SVMs, that is, \( O(K \cdot R \cdot (WH) \cdot (W_1H_1)) \).

6 Implementation

We list some details of our implementation of the cascade SVMs as follows.

1) Scale/aspect-ratio quantization scheme: In our experiments, we test \( \bar{\eta} \in \{0.5, 0.67, 0.75\} \), which lead respectively to the maximum numbers of classifiers learned at the first stage \( K \in \{36, 121, 196\} \) by limiting the sizes of windows from 10 to 500 pixels. This enables us to approximate the sizes of the smallest object and the whole image within the hierarchy.

2) Features and data used in Stage I: We use simple gradient features to learn each classifier \( w_k \) at the first stage. In detail, we first convert all the images into gray scale, and represent all the object ground-truth bounding boxes to \( \eta \)-accuracy using
our scale/aspect-ratio quantization scheme to provide positive windows. After randomly selecting negatives across scales, all windows are resized to a fixed feature window size \((W,H)\), and then for each pixel, the magnitude of its gradient is calculated. At test time, to generate features \(x\), we simply resize the image for each scale \(k\) by the ratio of its reference window to \((W,H)\), and then apply the learned classifier \(w_k\) by 2D convolution.

(3) Features used in Stage II: We use the 1D \((i.e. \: R = 1)\) classifier responses \((i.e. \: margins)\) from Stage I as features to train the ranking SVM, because from [10], we can see that the performance gained by increasing the dimension of features in Stage II is marginal, but computational time is boosted significantly, especially for large window size \((W,H)\).

(4) Parameters \(\gamma, d_1, W\ and \ H\): Here we follow our work in [10] and keep using the same parameters as before. Precisely, \(\gamma = 0.6, \: d_1 = 50^2, \: W = H = 16\) pixels. Please refer to [10] for the parameter selection details.

(5) SVM solver: We employ LIBLINEAR [32] as our solver. To train ranking SVMs, we take \(10^5\) samples randomly as the training set, each of which is created by a positive minus a negative. Without tuning, in all the cases, we set the regularization parameter \(C = 10\).

(6) Functions in Alg. 1: We utilize the default MATLAB functions here, except DistanceTransform, which is from VLFeat [33].

7 Experiments

In [10] we have demonstrated the capability of our method for class-specific object proposal generation.

2. When \(K = 36\), we set \(d_1 = 150\) so that our method can select more than \(10^3\) proposals from Stage I. For other \(K\), we still use \(d_1 = 50\).
and partial experimental results are shown in Fig. 3 and Fig. 4. For more details, please refer to [10].

In this paper, our method is extended for generic object proposal generation, and outputs bounding boxes as object proposals. We learn only one object model per quantized scale/aspect-ratio by using all the object instances in the training data as positives to train a single object/non-object filter and output object proposals per image during testing, no matter what classes the object instances belong to.

We measure our performance in terms of (1) object detection recall vs. overlap score threshold (recall-overlap) curves, (2) object recall vs. number of proposals (recall-proposal) curves, (3) average best overlap (ABO), (4) mean average best overlap (MABO), and (5) computational time. We follow the PASCAL VOC challenge and use overlap threshold $\eta = 0.5$ for correct detection.

We test our method on PASCAL VOC2007 [30]. VOC2007 contains 20 object categories, and consists of 9963 natural images with object labels and their corresponding ground-truth bounding boxes released for training, validation and test sets. We use the training/validation dataset, consisting of 5011 images, to train our method, and test it on the test dataset, comprising 4952 images.

We compare our method with [3], [4], [10], [14], [16], [17], [18], [20], [21], [14], [16], [17], [18], [20], [21]. To evaluate the DR and MABO, we download the precomputed proposals for [3], [14], and [4]. We use the default parameter setting for each method since they have been optimized for VOC2007, in general, expect for [21] where we utilize the parameters $(180, 9)$ as highlighted at the author's website. The output proposals from every method are ranked based on the associated scores by the code, if any, otherwise we preserve the same ranking order as the code returns. If there are no enough proposals, say less than 1000, we just use all proposals from this method for the measures using 1000 proposals.

7.1 Cascade Setting Comparison

Fig. 7 summarizes the AUC comparison results, where our program runs for three times and we report the mean and standard deviation of our results. In general, the performances using different settings are close to each other. When the number of proposals $d_2 \in \{1, 10, 100\}$, methods with the total number of quantized scales/aspect-ratios $K = 36$ work better than the other, on average, but when $d_2 = 1000$, methods with $K \in \{121, 196\}$ work equally better than those with $K = 36$. The main reason for this observation is that the number of windows returned from the first stage for $K = 36$ is significantly smaller than those for $K \in \{121, 196\}$. Then when $d_2$ is small, the windows for $K = 36$ have better chance to localize object instances correctly. Clearly, our proposal calibration algorithm does help improve the performance. Especially, the ranking constraints at the second stage are more important for achieving better performance with proposal calibration. When $d_2 = 1000$, all the settings with proposal calibration achieve very close performance in terms of mean and standard deviation. Without specific mention, we will compare our best performance with those of the other methods in the following sections.

7.2 Recall-Overlap Evaluation

The recall-overlap curves measure the detection quality of proposals within a fixed number of proposals by varying the overlap score threshold.

Fig. 8 shows our recall-overlap comparison results on VOC2007, where the object detection recall is the average over all the 20 classes. As we see, in general,
when \(d_2\) is small, our method works not as well as the segment-based methods, but when \(d_2 = 1000\) our method is comparable with most of the current state-of-the-art methods. Particularly, when \(d_2\) increases, the localization quality of our proposals is much better than those from CSVM [10] and BING [23] by a large margin. Since these two methods and ours share the same cascade SVM learning strategy, the big improvement comes from our proposal calibration.

Fig. 9 breaks down the VOC2007 results in Fig. 8 by classes using 1000 proposals, and Table 2 summarizes the AUC score comparison on the VOC2007 test data.

### TABLE 2

<table>
<thead>
<tr>
<th>Methods</th>
<th>(d_2 = 1)</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rahlu [3], [14]</td>
<td>11.0</td>
<td>36.0</td>
<td>57.8</td>
<td>71.3</td>
</tr>
<tr>
<td>Objectness [4]</td>
<td>20.3</td>
<td>44.4</td>
<td>59.5</td>
<td>66.8</td>
</tr>
<tr>
<td>CSVM [10]</td>
<td>10.7</td>
<td>33.2</td>
<td>53.7</td>
<td>68.9</td>
</tr>
<tr>
<td>Sel. Search [16]</td>
<td>14.5</td>
<td>39.5</td>
<td>63.7</td>
<td><strong>80.6</strong></td>
</tr>
<tr>
<td>Rand. Prim [17]</td>
<td>12.6</td>
<td>37.7</td>
<td>62.4</td>
<td>77.6</td>
</tr>
<tr>
<td>Endres [18]</td>
<td><strong>23.8</strong></td>
<td><strong>50.6</strong></td>
<td>69.1</td>
<td>77.3</td>
</tr>
<tr>
<td>Rantalankila [20]</td>
<td>17.8</td>
<td>44.6</td>
<td>57.7</td>
<td>74.5</td>
</tr>
<tr>
<td>GoP [21]</td>
<td>3.8</td>
<td>19.7</td>
<td>55.7</td>
<td>78.4</td>
</tr>
<tr>
<td>EdgeBox [22]</td>
<td>19.3</td>
<td>44.7</td>
<td>65.5</td>
<td>78.1</td>
</tr>
<tr>
<td>BING [23]</td>
<td>22.3</td>
<td>37.5</td>
<td>56.7</td>
<td>64.8</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>16.2</td>
<td>37.3</td>
<td>58.5</td>
<td>74.7</td>
</tr>
</tbody>
</table>

### TABLE 3

Object detection recall comparison (%) on VOC2007 in Fig. 10 using different numbers of proposals \(d_2\).

<table>
<thead>
<tr>
<th>Methods</th>
<th>(d_2 = 1)</th>
<th>10</th>
<th>100</th>
<th>1000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rahlu [3], [14]</td>
<td>7.0</td>
<td>32.7</td>
<td>64.7</td>
<td>83.5</td>
</tr>
<tr>
<td>Objectness [4]</td>
<td>17.3</td>
<td>49.5</td>
<td>75.8</td>
<td>92.0</td>
</tr>
<tr>
<td>CSVM [10]</td>
<td>3.5</td>
<td>25.2</td>
<td>63.5</td>
<td>92.0</td>
</tr>
<tr>
<td>Sel. Search [16]</td>
<td>9.7</td>
<td>37.3</td>
<td>71.5</td>
<td>93.5</td>
</tr>
<tr>
<td>Rand. Prim [17]</td>
<td>8.6</td>
<td>35.0</td>
<td>70.4</td>
<td>90.3</td>
</tr>
<tr>
<td>Endres [18]</td>
<td><strong>20.9</strong></td>
<td><strong>55.2</strong></td>
<td><strong>82.8</strong></td>
<td><strong>90.1</strong></td>
</tr>
<tr>
<td>Rantalankila [20]</td>
<td>0.1</td>
<td>0.9</td>
<td>16.2</td>
<td>85.6</td>
</tr>
<tr>
<td>GoP [21]</td>
<td>2.4</td>
<td>13.8</td>
<td>60.2</td>
<td>94.2</td>
</tr>
<tr>
<td>EdgeBox [22]</td>
<td>17.8</td>
<td>45.8</td>
<td>75.4</td>
<td>95.1</td>
</tr>
<tr>
<td>BING [23]</td>
<td>18.2</td>
<td>37.3</td>
<td>73.0</td>
<td><strong>95.2</strong></td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>9.9</td>
<td>34.0</td>
<td>66.9</td>
<td>93.1</td>
</tr>
</tbody>
</table>

### 7.3 Recall-Proposal Evaluation

As a pre-processing step in a system, the object recall with a certain \(\eta\) using a fixed number of proposals is very important, because this recall determines the best performance that objects can be detected. Therefore, we perform another measure, i.e. recall-proposal curves at \(\eta = 0.5\).

Fig. 10 shows our comparison results. When \(d_2\) increases, the curves become flatter and flatter as we expect. Again our method is comparable with the current state-of-the-art methods, and compared with CSVM and BING, our method works better than CSVM but worse than BING.

Similarly, Fig. 11 breaks down the VOC2007 comparison results in Fig. 10 by classes. As we see, some categories need far fewer proposals to achieve good performance. For instance, for the cat category, 100 proposals can achieve around 90% detection recall.

### Table 4

<table>
<thead>
<tr>
<th>Methods</th>
<th># prop. ((d_2))</th>
<th>MABO (%)</th>
<th>Detection rate (%), (\eta = 0.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rahlu [3], [14]</td>
<td>10000.0</td>
<td>80.4</td>
<td>87.7 81.7 73.4 <strong>60.3</strong> <strong>35.5</strong></td>
</tr>
<tr>
<td>Objectness [4]</td>
<td>999.6</td>
<td>68.0</td>
<td>88.6 78.2 41.0 12.0 0.02</td>
</tr>
<tr>
<td>CSVM [10]</td>
<td>1000.0</td>
<td>69.6</td>
<td>91.1 78.9 45.4 15.0 3.4</td>
</tr>
<tr>
<td>CSVM [10]</td>
<td>9990.2</td>
<td>72.5</td>
<td><strong>99.7</strong> 94.8 56.4 17.4 3.5</td>
</tr>
<tr>
<td>Sel. Search [16]</td>
<td>9988.8</td>
<td>81.8</td>
<td>90.0 82.9 71.5 55.7 32.8</td>
</tr>
<tr>
<td>Rand. Prim [17]</td>
<td>1350.4</td>
<td>80.5</td>
<td>89.6 81.8 71.1 56.1 33.0</td>
</tr>
<tr>
<td>Endres [18]</td>
<td>1611.0</td>
<td>79.1</td>
<td>88.4 79.5 67.4 51.6 30.4</td>
</tr>
<tr>
<td>Rantalankila [20]</td>
<td>1131.0</td>
<td>77.4</td>
<td>84.3 75.0 63.5 49.0 30.3</td>
</tr>
<tr>
<td>GoP [21]</td>
<td>1652.6</td>
<td><strong>81.9</strong></td>
<td>94.7 89.0 78.4 58.8 28.9</td>
</tr>
<tr>
<td>EdgeBox [22]</td>
<td>3480.0</td>
<td>80.7</td>
<td>98.2 <strong>96.1</strong> <strong>87.8</strong> 53.1 12.2</td>
</tr>
<tr>
<td>BING [23]</td>
<td>9905.2</td>
<td>66.3</td>
<td>99.5 70.2 29.4 9.4 2.5</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>10000.0</td>
<td>75.5</td>
<td>87.2 78.5 63.4 37.5 12.3</td>
</tr>
<tr>
<td><strong>Ours</strong></td>
<td>9990.2</td>
<td>79.3</td>
<td>98.1 94.8 82.2 47.5 13.3</td>
</tr>
</tbody>
</table>

### 7.4 ABO & MABO

Average best overlap (ABO) is a proposal localization quality measure for a class, and mean average best overlap (MABO) is a measure of the ability of how well the proposals generated by a method can localize object instances. ABO is defined as the average of the best overlap score for each object instance in each class, one ABO per class, and MABO is the mean of ABO’s across all the classes.

Table 5 lists our ABO & MABO comparison results on the VOC2007 test data. Still our method is comparable with others in all the classes, and ours is much better than CSVM and BING, which again demonstrates the usefulness of the proposal calibration in our method. Fig. 12 shows some failure examples of our method for visualization that succeeded in at least one of the methods [16], [17], [18], [21], [22].

We also list the number of total proposals and the corresponding MABO and object detection rate \(\text{i.e.} \text{number of detected objects divided by the total number of objects}\) with different overlap threshold \(\eta\) for each method in Table 5. Our method is comparable with the best performance among the competitors.
### TABLE 4

<table>
<thead>
<tr>
<th>Methods</th>
<th>aer.</th>
<th>bic.</th>
<th>bird</th>
<th>boat</th>
<th>bot.</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>cha.</th>
<th>cow</th>
<th>din.</th>
<th>dog</th>
<th>hor.</th>
<th>mot.</th>
<th>per.</th>
<th>pet.</th>
<th>pot.</th>
<th>she.</th>
<th>sota</th>
<th>tra.</th>
<th>tv</th>
<th>MABO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rahtu [3, 14]</td>
<td>72.6</td>
<td>74.3</td>
<td>69.0</td>
<td>69.7</td>
<td>49.0</td>
<td>79.9</td>
<td>67.2</td>
<td>82.6</td>
<td>72.9</td>
<td>80.5</td>
<td>70.8</td>
<td>78.5</td>
<td>74.3</td>
<td>65.7</td>
<td>58.9</td>
<td>69.8</td>
<td>82.3</td>
<td>79.5</td>
<td>73.8</td>
<td>72.2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Objectness [4]</td>
<td>67.3</td>
<td>69.0</td>
<td>65.6</td>
<td>64.4</td>
<td>57.1</td>
<td>72.9</td>
<td>65.3</td>
<td>70.7</td>
<td>62.8</td>
<td>64.8</td>
<td>76.6</td>
<td>73.3</td>
<td>71.0</td>
<td>67.0</td>
<td>67.6</td>
<td>63.5</td>
<td>61.4</td>
<td>65.6</td>
<td>74.7</td>
<td>70.6</td>
<td>65.5</td>
<td></td>
</tr>
<tr>
<td>CSVM [10]</td>
<td>70.4</td>
<td>70.2</td>
<td>67.6</td>
<td>66.7</td>
<td>58.9</td>
<td>73.6</td>
<td>67.6</td>
<td>76.5</td>
<td>64.9</td>
<td>68.0</td>
<td>74.1</td>
<td>74.7</td>
<td>72.7</td>
<td>71.4</td>
<td>66.6</td>
<td>62.3</td>
<td>67.9</td>
<td>75.6</td>
<td>74.5</td>
<td>66.7</td>
<td>69.6</td>
<td></td>
</tr>
<tr>
<td>Sel. Search [16]</td>
<td>80.9</td>
<td>80.9</td>
<td>74.5</td>
<td>74.3</td>
<td>69.2</td>
<td>83.7</td>
<td>76.3</td>
<td>87.0</td>
<td>74.5</td>
<td>79.8</td>
<td>82.4</td>
<td>87.5</td>
<td>79.5</td>
<td>81.0</td>
<td>70.6</td>
<td>67.1</td>
<td>72.8</td>
<td>89.4</td>
<td>82.5</td>
<td>80.0</td>
<td>78.4</td>
<td></td>
</tr>
<tr>
<td>Rand. Prim [17]</td>
<td>83.5</td>
<td>83.1</td>
<td>80.1</td>
<td>78.1</td>
<td>62.8</td>
<td>85.2</td>
<td>77.7</td>
<td>90.7</td>
<td>77.6</td>
<td>82.2</td>
<td>88.6</td>
<td>89.4</td>
<td>82.9</td>
<td>81.9</td>
<td>72.6</td>
<td>71.2</td>
<td>80.4</td>
<td>90.3</td>
<td>85.7</td>
<td>83.7</td>
<td>81.4</td>
<td></td>
</tr>
<tr>
<td>Endres [18]</td>
<td>71.0</td>
<td>81.0</td>
<td>72.3</td>
<td>65.5</td>
<td>60.9</td>
<td>85.1</td>
<td>79.1</td>
<td>87.9</td>
<td>72.4</td>
<td>80.3</td>
<td>87.0</td>
<td>87.1</td>
<td>82.2</td>
<td>70.5</td>
<td>70.5</td>
<td>67.4</td>
<td>76.2</td>
<td>89.7</td>
<td>84.5</td>
<td>79.1</td>
<td>78.1</td>
<td></td>
</tr>
<tr>
<td>Rantalankila [20]</td>
<td>73.5</td>
<td>74.6</td>
<td>73.5</td>
<td>66.7</td>
<td>54.0</td>
<td>81.3</td>
<td>72.7</td>
<td>89.2</td>
<td>86.8</td>
<td>76.4</td>
<td>83.2</td>
<td>87.4</td>
<td>81.0</td>
<td>76.0</td>
<td>66.2</td>
<td>62.8</td>
<td>72.1</td>
<td>87.1</td>
<td>82.0</td>
<td>77.5</td>
<td>75.3</td>
<td></td>
</tr>
<tr>
<td>GoP [21]</td>
<td>73.5</td>
<td>74.6</td>
<td>73.5</td>
<td>66.7</td>
<td>54.0</td>
<td>81.3</td>
<td>72.7</td>
<td>89.2</td>
<td>86.8</td>
<td>76.4</td>
<td>83.2</td>
<td>87.4</td>
<td>81.0</td>
<td>76.0</td>
<td>66.2</td>
<td>62.8</td>
<td>72.1</td>
<td>87.1</td>
<td>82.0</td>
<td>77.5</td>
<td>75.3</td>
<td></td>
</tr>
<tr>
<td>EdgeBox [22]</td>
<td>76.8</td>
<td>81.6</td>
<td>78.5</td>
<td>76.7</td>
<td>65.8</td>
<td>83.9</td>
<td>76.3</td>
<td>823</td>
<td>76.4</td>
<td>82.2</td>
<td>80.9</td>
<td>83.6</td>
<td>81.3</td>
<td>80.9</td>
<td>73.6</td>
<td>71.7</td>
<td>80.8</td>
<td>82.5</td>
<td>79.8</td>
<td>81.4</td>
<td>78.9</td>
<td></td>
</tr>
<tr>
<td>BING [23]</td>
<td>65.5</td>
<td>66.0</td>
<td>64.0</td>
<td>62.3</td>
<td>60.6</td>
<td>66.5</td>
<td>64.4</td>
<td>69.9</td>
<td>62.6</td>
<td>65.1</td>
<td>69.5</td>
<td>68.3</td>
<td>65.9</td>
<td>65.7</td>
<td>63.8</td>
<td>62.4</td>
<td>64.6</td>
<td>69.0</td>
<td>68.6</td>
<td>63.4</td>
<td>65.4</td>
<td></td>
</tr>
<tr>
<td>Ours</td>
<td>79.9</td>
<td>77.1</td>
<td>75.6</td>
<td>70.9</td>
<td>60.4</td>
<td>79.4</td>
<td>72.0</td>
<td>84.3</td>
<td>69.1</td>
<td>76.7</td>
<td>80.7</td>
<td>82.4</td>
<td>80.5</td>
<td>77.6</td>
<td>69.0</td>
<td>66.3</td>
<td>75.0</td>
<td>82.7</td>
<td>78.0</td>
<td>73.8</td>
<td>75.5</td>
<td></td>
</tr>
</tbody>
</table>

#### 7.5 Computational Time

We show the computational time comparison in Fig. 13. Our implementation is based on the code of CSVM, a mixture of Matlab and C++. All the methods run on an Intel Xeon E2696 v2 CPU with 2.50GHz. The computational time shown here includes all the steps at the test stage starting from loading images. In general, segment-based methods run relatively slow. For instance, the fastest segment-based method, i.e. GoP [21] needs 1.26s, while the slowest one, i.e. Rantalankila [20] needs 23.72s. CSVM and BING run at almost the same speed (0.06s), and our method needs 0.07s per image. The proposal calibration step needs the extra 0.01s.

### 8 CONCLUSION AND DISCUSSION

We propose a very efficient two-stage cascade SVM method for object proposal generation. To achieve better computational efficiency, we propose a scale/aspect-ratio quantization scheme to reduce the bounding box search space into log-space. To represent each object instance, we utilize the simple gradients within small fixed-size windows (e.g. $8 \times 8$ pixels). We learn linear filters in each stage using linear SVMs, resulting in applying fast 2D convolution to localizing object proposals during testing. Non-max suppression is used to select proper proposals in the first stage. To further improve the localization quality of windows returned from the second stage, we propose a new proposal calibration algorithm as a post-processing step. Comprehensive experiments on VOC2007 demonstrate that our method is not only comparable with the current state-of-the-art methods in terms of both object detection recall, ABO, and MABO, but also can run as fast as 0.07s per image, which is at least an order of magnitude faster than the current state-of-the-art methods.

We envisage that the cascaded model can be used as the initial stage in complex systems. Our framework naturally incorporates scale and aspect ratio information about objects, which are treated separately in the first stage of the cascade, and we emphasize the flexibility of the framework, where different types of features could easily be incorporated at this stage. Besides object detection, we believe that our work will contribute to many other research areas, such like segmentation [15], stereo matching [34], and video analysis [35].

Our recent proposal generation method, BING [23],
achieves the fastest running time among all popular object proposal generation methods [24], and the most repeatable under different imaging conditions (e.g. illumination, rotation, scaling, blurring, etc.). However, the main issue of this method seems that the localization quality of our proposals are worse quantitatively compared to other methods. This is mainly because of our scale/aspect-ratio quantization scheme. The proposal calibration algorithm in this paper may be a cure for BING.

ACKNOWLEDGMENTS

We acknowledge support of the EPSRC and financial support was provided by ERC grant ERC-2012-AdG 321162-HELIOS.

REFERENCES


Ziming Zhang received his MS degree in 2010 from Simon Fraser University, Canada. Later he received his PhD degree in 2013 from Oxford Brookes University, UK, under the supervision of Prof. Philip H. S. Torr. He is currently holding a research position in Boston University, MA. His research areas includes object recognition and detection, machine learning, optimization, large-scale information retrieval, visual surveillance, and medical imaging analysis. His works have appeared in CVPR, ECCV, and NIPS.

Philip H.S. Torr received the PhD degree from Oxford University. After working for another three years at Oxford, he worked for six years as a research scientist for Microsoft Research, first in Redmond, then in Cambridge, founding the vision side of the Machine Learning and Perception Group. He is now a professor at Oxford University. He has won awards from several top vision conferences, including ICCV, CVPR, ECCV, NIPS and BMVC. He is a Royal Society Wolfson Research Merit Award holder.
Fig. 9. Comparison of class-specific recall-overlap curves using different methods on the VOC2007 test data with 1000 proposals.
Fig. 11. Comparison of class-specific recall-proposal curves using different methods on the VOC2007 test data. Here we utilize the overlap threshold $\eta = 0.5$. 