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 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 proposals can be compared properly. The proposals with highest scores are our final output. Specifically, we explain our scale/aspect-ratio quantization scheme, and investigate the effects of combinations of ℓ_1 and ℓ_2 regularizers in cascade SVMs with/without ranking constraints in learning. Comprehensive experiments on VOC2007 dataset are conducted, and our results achieve the state-of-the-art performance with high object recall rate and high computational efficiency. Besides, our method has been demonstrated to be suitable for not only class-specific but also generic object proposal generation.

Index Terms—Generic/Class-specific object proposal generation, Scale/Aspect-ratio quantization, Cascade SVMs, Linear filters

◆

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 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 proposals, 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.

● *Dr. Z. Zhang is currently with the Department of Electrical and Computer Engineering, Boston University, Boston, MA 02215, US. Prof. P.H.S. Torr is with the Department of Engineering Science, University of Oxford, Oxford OX1 3PJ, UK.*
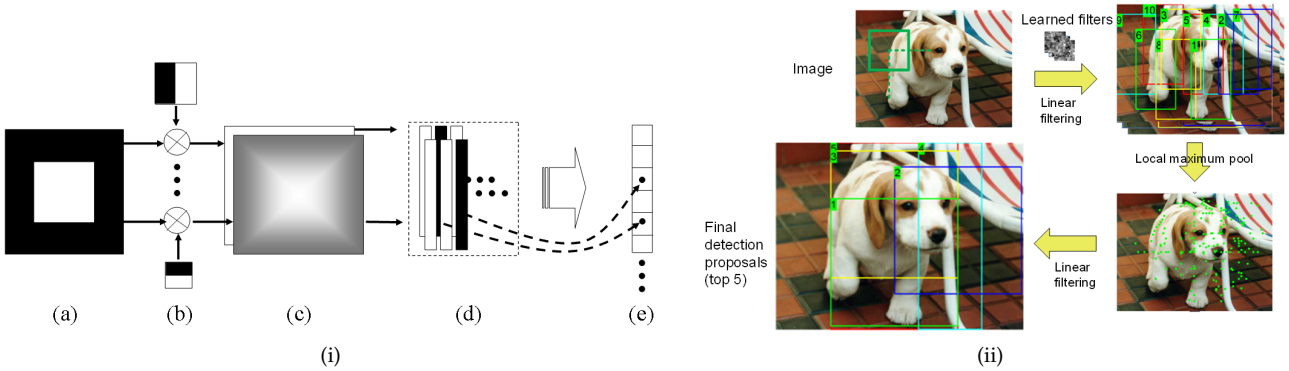*E-mail: zzhang14@bu.edu, philip.torr@eng.ox.ac.uk*

Fig. 1. (i) Summary of our cascaded method. An image (a) is first convolved with a set of linear classifiers at varying scales/aspect-ratios (b) producing response images (c). Local maxima are extracted with non-max suppression from each response image, and the corresponding windows with top ranking scores are forwarded to the second stage of the cascade. Each proposed window is associated with a feature vector (d), and a second round of ranking orders these proposals (e) so that the true positives (marked as black) are pushed towards the top during training. Our method outputs the top ranking windows in this final ordering. (ii) An example of generating proposals for detecting the dog in the image is shown, which explains the steps in (i). The numbers at the corners of windows in the bottom-left image indicate the ranks of windows.

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, and particularly demonstrate the capability of our method for *generic* object proposal generation. We explore the effects of combinations of ℓ_1 and ℓ_2 regularizers in two-stage cascade SVMs with/without ranking constraints in learning. Interestingly, our comprehensive comparison on the VOC2007 [12] dataset suggests that *in general, the cascade where ℓ_1-SVMs, which perform feature selection [13], are utilized in both stages without ranking constraints consistently works best*.

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, and list some implementation details in Section 5. Finally Section 6 shows our experimental results and Section 7 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 [7], [8], [5] 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. [14] 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 [14]. 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 [15] strongly suggests that we should explicitly learn different detectors for different scales.

Several recent works [3], [4], [16], [17], [18], [19] 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. The idea of grouping super-pixels/segments to generate proposals is also used in [16], [17], [18], [19] with different grouping criteria. More empirical comparisons of different proposed object proposal

generation methods are presented in [20].

The major differences between our method and these related work above are:

- From the view of features, our method only takes simple image gradients as features for learning and testing, while all of the related work above utilize multiple visual cues in images;
- From the view of ranking proposals, our method utilizes the classification scores (*i.e.* margins) generated by the learned linear classifiers, rather than the scores from super-pixels [16], prior object localization distributions [3], or the combination of multiple visual cues [4], [3], which involves more heuristics in general;
- From the view of learning, our method formulates the problem using the cascade SVM framework, which is much easier to understand and implement.

As a result, our method achieves state-of-the-art performance with high object recall and high computational efficiency.

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 s and a ground-truth bounding box of an object t, $o(s, t)$, is defined as their intersection area divided by their union area. Clearly, $0 \leq o(s, t) \leq 1$, and the higher $o(s, t)$ is, the better the localization of the object t with the bounding box s is.*

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

Definition 3 (Maximum Overlap). *Given an image I and the ground-truth bounding boxes of multiple objects $g_{1 \cdots m_I}$ in I, the maximum overlap of a window s in I is defined as $o_s = \max_{i \in \{1, \cdots, m_I\}} o(s, g_i)$, where $o(s, g_i)$ denotes the overlap score between s and g_i.*

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

Definition 5 (Quantized Scale/Aspect-ratio). *Given an overlap score threshold η, a window s in an image can be quantized into a quantized scale/aspect-ratio $\mathcal{T}$ if and only if $\exists t \in \mathcal{T}$ such that s can be localized to η-accuracy, where t is a window with the quantized scale/aspect-ratio.*

3.2 Quantization Scheme

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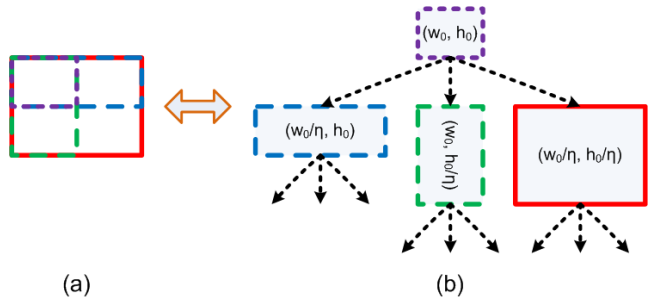

Fig. 2. Illustration of hierarchical representation of our scale/aspect-ratio quantization scheme with overlap threshold $\eta = 0.5$. (a) superimposes the four window scales in a mini-quantization scheme, and (b) unfolds the scales into a tree structure. The relative widths and heights of the windows are represented by the (w, h) pairs. Such a hierarchy can represent all windows to η-accuracy.

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 $\mathcal{S}(w_0/\eta^a, h_0/\eta^b)$, where $a \in \{0, 1, \cdots, A\}$ and $b \in \{0, 1, \cdots, B\}$ are naturally limited by the image size, and $\mathcal{S}(\cdot, \cdot)$ denotes the set of windows with the specific width and height. 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 η_0 and a minimum size of objects (w_0, h_0) that can be found in images, any window s with window size (w_s, h_s) can be localized to η_0-accuracy by at least one window t in our scale/aspect-ratio quantization scheme with parameter $\eta \geq \eta_0$.*

Proof: According to Fig. 2, we can construct a subset of windows in our quantized scheme by computing $a \in \left\{ \lfloor \log_\eta \frac{w_s}{w_0} \rfloor, \cdots, \lceil \log_\eta \frac{w_s}{w_0} \rceil \right\}$ and $b \in \left\{ \lfloor \log_\eta \frac{h_s}{h_0} \rfloor, \cdots, \lceil \log_\eta \frac{h_s}{h_0} \rceil \right\}$, where $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ denote the floor and ceiling operations, respectively. Letting $t(w_t, h_t)$ be a window with quantized scale/aspect-ratio (w_t, h_t), the overlap between s and t can be calculated as follows:

$$\exists a, b, \ o\left(s, t\left(w_0 \eta^a, h_0 \eta^b\right)\right) \tag{1}$$
$$= \frac{\min\{w_s, w_0\eta^a\} \cdot \min\{h_s, h_0\eta^b\}}{\max\{w_s, w_0\eta^a\} \cdot \max\{h_s, h_0\eta^b\}}$$
$$= \eta^{\left|a - \log_\eta \frac{w_s}{w_0}\right| + \left|b - \log_\eta \frac{h_s}{h_0}\right|} \geq \eta^{0.5 + 0.5} = \eta \geq \eta_0.$$

That is, s can be localized to η_0-accuracy by t. $\square$

Proposition 2 (Sufficient Number of Quantized Scales/Aspect-ratios). *Given an overlap score threshold*
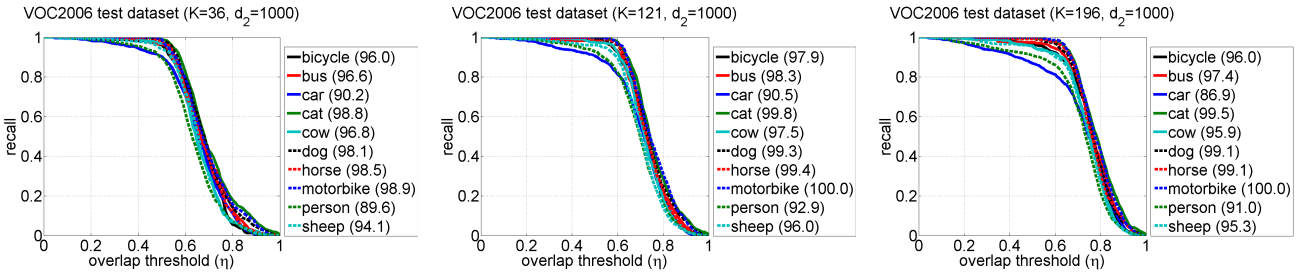
Fig. 3. An example of our method [10] on demonstrating the localization quality with increase of the number of quantized scales/aspect-ratios, K, on the VOC2006 [21] dataset using the object recall-overlap evaluation. Recall-overlap curves are plotted for individual classes using $d_2 = 1000$ final proposals from left to right, and $K \in \{36, 121, 196\}$ from top to bottom. The numbers shown in the legends are the recall percentages when the overlap score threshold for correct localization, η, is set to 0.5. For more details, please refer to [10].

η, *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 is bounded by $\left(1 + \lceil \log_\eta \frac{w_0}{w} \rceil\right)\left(1 + \lceil \log_\eta \frac{h_0}{h} \rceil\right)$.*

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 $a \in \{0, \cdots, \lceil \log_\eta \frac{w_0}{w} \rceil\}$ and $b \in \{0, \cdots, \lceil \log_\eta \frac{h_0}{h} \rceil\}$. Therefore, the number of quantized scales/aspect-ratios that is sufficient to localize all possible objects in images is bounded by $\left(1 + \lceil \log_\eta \frac{w_0}{w} \rceil\right)\left(1 + \lceil \log_\eta \frac{h_0}{h} \rceil\right)$. □

Proposition 3 (Search Space for Object Localization). *Given an overlap score threshold η, 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 using quantized windows is $O\left(W \cdot \lceil \log_\eta \frac{w_0}{W} \rceil \cdot H \cdot \lceil \log_\eta \frac{h_0}{H} \rceil\right)$.*

Proof: According to Proposition 2, the search space for scales/aspect-ratios of objects is reduced to $O(\lceil \log_\eta \frac{w_0}{W} \rceil \lceil \log_\eta \frac{h_0}{H} \rceil)$ 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 $O\left(W \cdot \lceil \log_\eta \frac{w_0}{W} \rceil \cdot H \cdot \lceil \log_\eta \frac{h_0}{H} \rceil\right)$. □

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 (i.e. 8×8 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 [22], this intuition was shared as well.

3.3.2 Localization Quality

From Proposition 1, we can see that *the localization quality of a given quantization scheme in our method is dependent on the parameter η (NOT η_0), 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.* $\eta_0 = 0.5$. However, to construct our quantization scheme, we can choose an arbitrary value for the parameter η as long as $\eta_0 \leq \eta < 1$, say $\eta = 0.6$. Then our method can generate better object proposals than those using $\eta = 0.5$, 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 $K \in \{36, 121, 196\}$ corresponds to $\eta \in \{\frac{1}{2}, \frac{2}{3}, \frac{3}{4}\}$, 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. 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

Cascaded classifiers have a decade history in object detection [23], [24], [25], especially the very successful Viola and Jones's method for face detection [23]. Cascaded classifiers are good tools for handling extremely imbalanced data, that is, too many negatives

	K = 36				121				196			
	(W,H)=(2,2)	(4,4)	(8,8)	(16,16)	(2,2)	(4,4)	(8,8)	(16,16)	(2,2)	(4,4)	(8,8)	(16,16)
R=1	0.10±0.02	0.10±0.02	0.14±0.03	0.24±0.08	0.22±0.06	0.24±0.07	0.30±0.13	0.54±0.34	0.37±0.10	0.38±0.11	0.46±0.18	0.75±0.43
2	0.10±0.02	0.11±0.02	0.15±0.04	0.32±0.11	0.23±0.06	0.26±0.09	0.35±0.17	0.70±0.47	0.37±0.10	0.40±0.12	0.51±0.22	0.98±0.61
4	0.10±0.02	0.11±0.02	0.19±0.05	0.47±0.17	0.24±0.07	0.28±0.10	0.43±0.25	1.01±0.76	0.40±0.10	0.43±0.14	0.63±0.32	1.40±1.01

Fig. 4. Comparing the speed of our method [10] in seconds at various parameter settings in forms of "mean±standard deviation", where K denotes the number of quantized scales/aspect-ratios, (W, H) denotes the filter size, and R denotes the number of feature channels. The code is written using a mixture of Matlab and C++, and run on a single core with 3.33 GHz. The highlighted (red) numbers are close to the running time in [6], one of the state-of-the-art cascaded classifiers. For more details, please refer to [10].

TABLE 1
Some notations used in explanation of our cascade SVMs.

Notation	Definition
$\mathcal{T}$	The set of all possible windows in an image.
$\mathcal{S}$	The set of all possible windows in our window quantization scheme.
$\mathcal{S}(w,h)$	The set of all the windows in an image with width w and height h.
$o(t,s)$	The overlap between window $t \in \mathcal{T}$ and window $s \in \mathcal{S}$ (see Def. 1).
o_t	The maximum overlap for window $t \in \mathcal{T}$ in an image (see Def. 3).
$\eta \in [0,1]$	Overlap score threshold for proposal generation.
k	A given scale/aspect-ratio combination in our quantization scheme.
$\mathcal{S}_k$	The set of all the windows which can be represented to η-accuracy at quantized scale/aspect-ratio k.
$\mathbf{w}_k$, $\mathbf{z}_k$	Learned linear classifiers at Stage I and II, respectively, for quantized scale/aspect-ratio k.
$\mathbf{v}$	A channel response feature vector used in Stage II for learning $\mathbf{z}$.

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".

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

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 η, 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 η and a set of quantized scales/aspect-ratios, for each scale k^1 we wish to learn a linear classifier $f_1(\mathbf{x}_s; \mathbf{w}_k) = \mathbf{w}_k \cdot \mathbf{x}_s$, as suggested in [15], to rank the window $s \in \mathcal{S}_k$, whose feature vector is denoted as $\mathbf{x}_s$, among all the windows in $\mathcal{S}_k$.

Ideally, we expect that within image I the ranking score for any window $s_i \in \mathcal{S}_k \bigcap \mathcal{T}_I$ with $o_{s_i} \geq \eta$ is always higher than that of any window $s_j \in \mathcal{T}_I$ with $o_{s_j} < \eta$. That is, for $\mathbf{w}_k$ we require that within the image I all the corresponding positive training windows $\mathcal{I}_k^+ = \{s_i \in \mathcal{S}_k \bigcap \mathcal{T}_I | o_{s_i} \geq \eta\}$ should be ranked above all the training negatives $\mathcal{I}^- = \{s_j \in \mathcal{T}_I | o_{s_j} < \eta\}$. Naturally this leads us to formulate the problem as a ranking SVM as follows:

$$\min_{\mathbf{w}_k, \boldsymbol{\xi}} \quad \frac{1}{p}\|\mathbf{w}_k\|_p^p + C \sum_{i,j,n} \xi_{ij}^n \qquad (2)$$

$$\text{s.t.} \quad \forall n, i \in \mathcal{I}_{kn}^+, j \in \mathcal{I}_n^-, \ \mathbf{w}_k \cdot (\mathbf{x}_i^n - \mathbf{x}_j^n) \geq 1 - \xi_{ij}^n,$$
$$\xi_{ij}^n \geq 0, \quad p \in 1,2.$$

Here, $\mathbf{x}_i^n$ and $\mathbf{x}_j^n$ are the feature vectors associated with positive window i and negative window j in training image I_n respectively, $\boldsymbol{\xi}$ are the slack variables, $C \geq 0$ is a predefined regularization parameter, and $\|\cdot\|_p$ denotes the ℓ_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. 2 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. 2 more general, we introduce a dummy feature $\mathbf{0}$ 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. 2 into a standard SVM without ranking constraints. We denote the solution of Eq. 2 with ranking constraints as "ℓ_p-w/r", and the solution of Eq. 2 without ranking constraints as "ℓ_p-o/r", respectively.

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

---
1. In the following sections, we refer to scale k as quantized scale/aspect-ratio k for short.

local maxima in the response image of classifier $\mathbf{w}_k$ 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, $d_1 \in \{1, \cdots, 1000\}$ 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, $\mathbf{v}$, which consists of the channel responses of the classifier at the first stage. For instance, $\mathbf{v}$ could be a 4-dimensional feature vector if feature $\mathbf{x}$ is divided into 4 segments without overlaps, each of which gives a response to the corresponding classifier. The reason for splitting $\mathbf{x}$ into different segments is that we could make full use of information in different segments to improve the calibration performance.

Based on $\mathbf{v}$, we can re-rank each window i by the decision function $f(\mathbf{v_i}) = \mathbf{z}_{k_i} \cdot \mathbf{v}_i + e_{k_i}$, where k_i denotes the quantized scale/aspect-ratio associated with window i, $\mathbf{z}_{k_i}$ is a set of coefficients for scale k_i 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. 3:

$$\min_{\mathbf{z}, \mathbf{e}, \boldsymbol{\xi}} \quad \frac{1}{p} \|\mathbf{z}\|_p^p + C \sum_{i,j,n} \xi_{ij}^n \qquad (3)$$

$$\text{s.t.} \quad \forall n, i \in \hat{\mathcal{I}}_n^+, j \in \hat{\mathcal{I}}_n^-,$$
$$\mathbf{z}_{k_i} \cdot \mathbf{v}_i^n - \mathbf{z}_{k_j} \cdot \mathbf{v}_j^n + e_{k_i} - e_{k_j} \geq 1 - \xi_{ij}^n,$$
$$\xi_{ij}^n \geq 0, \quad p \in \{1, 2\}.$$

Here, $\hat{\mathcal{I}}_n^+$ and $\hat{\mathcal{I}}_n^-$ denote the positive and negative windows in image I_n forwarded from the first stage of the cascade across different quantized scales/aspect-ratios. Similar to Eq. 2 with the dummy feature, we continue to use the same notations for the solutions of Eq. 3.

In this way, all the windows can be ranked in an image. The top d_2 windows are then considered as the final proposals generated at the second stage of our cascade.

4.3 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 must be applied to each image. The complexity can thus be approximated as $O(K \times R \times (W \times H) \times (W_I \times H_I))$, 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_I, H_I) 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 IMPLEMENTATION

We list some details of our implementation[2] of the cascade SVMs as follows.

(1) Scale/aspect-ratio quantization scheme: In our experiments, we test $\eta \in \{0.5, 0.67, 0.75\}$, 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 $\mathbf{w}_k$ 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 η-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 $\mathbf{x}$, 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 $\mathbf{w}_k$ 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 γ, d_1, W and H: Here we follow our work in [10] and keep using the same parameters as before. Precisely, $\gamma = 0.6$, $d_1 = 50$ [3], $W = H = 16$ pixels. Please refer to [10] for the parameter selection details.

(5) SVM solver: We employ LIBLINEAR [26] as our solver. To train ranking SVMs, we take 10^5 samples

---

2. The code is available at https://sites.google.com/a/brookes.ac.uk/zimingzhang/code.
3. 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 $d_1 = 50$.

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 EXPERIMENTS

In [10] we have demonstrated the capability of our method for class-specific object proposal generation, 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. Therefore, it is fair to compare our method with the two very closely related work [3][4] and [4][5]. We did not compare ours with [16] because their outputs are pixels in the proposals rather than bounding boxes, which makes their method better for segmentation measure.

We test our method on PASCAL VOC2007 [12]. 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 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 binary object/non-object filter and output object proposals per image during testing, no matter what classes the object instances belong to. This is our default learn and testing procedure without specific mention.

We measure our performance in terms of *object recall vs. overlap score threshold* (recall-overlap for short) curves [10], [3], [5], [6], *object recall vs. number of proposals* (recall-proposal for short) curves at $\eta = 0.5$, and running speed. We follow the PASCAL VOC challenge and use $\eta = 0.5$ for correct detection.

6.1 VOC2007

We first test different cascade settings on this dataset using different K's and ℓ's, and then compare our method with [3], [4]. We use the training/validation dataset, consisting of 5011 images, to train our model, and test it on the test dataset, comprising 4952 images.

6.1.1 Cascade Setting Comparison

Fig. 5 summarizes the comparison results, where our program runs for three times and we report the mean and standard deviation of our results. From the top 3 settings in each sub-figure, we can see that (1) In general, the performances using different settings are close to each other; (2) In Stage I, the method $\ell_1 - o/r$ seems to work best, which trains ℓ_1-norm SVMs, rather than ranking SVMs; (3) In Stage II,

both methods $\ell_1 - o/r$ and $\ell_2 - w/r$ seem to work better than others, the first training ℓ_1-norm SVMs and the second training ℓ_2-norm ranking SVMs; (4) The method proposed in [10] is slightly worse than the best setting; (5) With a larger K, the AUC score under 1000 proposals becomes larger, while differences of the AUC score under a fewer proposals (*i.e.* 1, 10, 100) are marginal. This also verifies that with a larger K, the localization quality of our object proposals will become better, in general, as stated in Section 3.3.2.

It surprises us that the $\ell_1 - o/r$ SVMs work so well in our cascade, because usually ℓ_2-norm SVMs work better than ℓ_1-norm SVMs [26]. We believe that ℓ_1-norm SVMs actually select the discriminant features and suppress non-discriminant ones between objects and non-objects.

6.1.2 Recall-Overlap Evaluation

The object recall *vs.* overlap score threshold (recall-overlap for short) curves measure the quality of proposals within a fixed number of proposals by varying the overlap score threshold.

Fig. 6 shows our comparison results on VOC2007. We can see here the movement of the curves towards the top-right both as we allow more output proposals ($d_2 \in \{1, 10, 100, 1000\}$) and as we increase $K = \{36, 121, 196\}$ in our quantization scheme. Recall that our quantized scales/aspect-ratios are designed to cover bounding boxes to a particular overlap score threshold of η, so $K \in \{36, 121, 196\}$ corresponds to $\eta \in \{0.5, 0.67, 0.75\}$ respectively. This affects the performance observed, and on the $K = 36$ graph for instance, 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 $\eta = 0.6$, implying our quantization is capturing the desired information.

From the curves, we also can see that our method has a similar behavior to [4], and their AUC values are close to each other, and at $\eta = 0.5$, in most cases our method and [4] achieve higher object recall than [3]. However, in terms of proposal localization quality, [3] is the best among these methods, because its curves drop quickly when η is larger than around 0.75, while the curves of ours and [3] drop when η is larger than around 0.55. This observation indicates that compared to our method and [4], the correct detection proposals outputted by [3] are closer to the ground-truth bounding boxes of objects, which may be caused by the structured learning used in [3].

Fig. 8 breaks down the VOC2007 results in Fig. 6 by classes using 1000 proposals, and displays the recall-overlap curves. Similar observations to Fig. 6 can be made. Table 2 summarizes the AUC score comparison on VOC2007.

4. We downloaded their public code and precomputed windows for VOC2007 from http://www.cse.oulu.fi/CMV/Downloads/ObjectDetection.

5. We downloaded their public code and precomputed windows for VOC2007 from http://groups.inf.ed.ac.uk/calvin/objectness/.

TABLE 2
AUC score comparison on VOC2007 using 1000 proposals in Fig. 6 and Fig. 8.

Methods	AUC (objects)	AUC (classes)
Ours ($\ell_1 - o/r + \ell_1 - o/r$)	64.5%	(65.1±2.2)%
[4]	64.9%	(66.8±4.2)%
[3]	**67.4%**	**(70.8±8.3)%**

TABLE 3
Object recall comparison on VOC2007 using 1000 proposals as shown in Fig. 7 and Fig. 9.

Methods	Recall (objects)	Recall (classes)
Ours ($\ell_1 - o/r + \ell_1 - o/r$)	**93.8%**	**(95.1±3.5)%**
[4]	88.6%	(92.0±6.7)%
[3]	77.7%	(82.8±12.8)%

TABLE 4
Object recall comparison on VOC2007 using different numbers of proposals and the same experimental setting in [4].

Method	10 Prop.	100 Prop.	1000 Prop.
Ours ($\ell_1 - o/r + \ell_1 - o/r$)	**46.4%**	**78.7%**	**93.1%**
[4]	41.0%	71.0%	91.0%

TABLE 5
Computational time comparison on VOC2007 in second per image with 1000 proposals.

Methods	Computational time
Ours ($\ell_1 - o/r + \ell_1 - o/r$)	**0.20±0.02**
[4]	3.58±0.25
[3]	2.22±0.42

6.1.3 Recall-Proposal Evaluation

As a pre-process step in a system, the object recall with a certain η using a fixed number of proposals is more important, because this recall determines the best performance that objects can be detected. Therefore, we propose another measure using the object recall vs. number of proposals (recall-proposal for short) curves at $\eta = 0.5$.

In Fig. 7 we show how the recalls of different methods are effected as we increase the number of output proposals d_2 from 1 to 1000 on VOC2007. We can see that when d_2 is beyond 200, the curves become flatter and flatter. We believe that this property of our approach is useful for detection tasks, because it narrows down significantly the total number of windows that classifiers need to check while losing few correct detections. From the comparison of the 4 cascade settings, $\ell_1 - o/r + \ell_1 - o/r$ performs best. Therefore, *in the following experiments we use the setting "$\ell_1 - o/r + \ell_1 - o/r$" as our default cascade setting*. Compared with [4], [3], our method has a similar behavior to [4], and both are better than [3] significantly.

Similarly, Fig. 9 breaks down the VOC2007 results in Fig. 7 by classes and displays the recall-overlap curves. As we see, some categories need far fewer proposals to achieve good performance. For instance, for the dog category, 100 output proposals saturate performance. Table 3 lists the object recall comparison on VOC2007.

Particularly, here we also perform a same experiment used in objectness [4]. We divide the 20 object categories into two sets. Same as [4], we use the 14 categories (*i.e.* aeroplane, bicycle, boat, bottle, bus, chair, diningtable, horse, motorbike, person, potted-plant, sofa, train, tvmonitor) as testing categories, and the rest as training categories, which means that these 14 categories are unseen during training. The images containing objects within the training categories in the training/validation dataset are utilized as the training data for learning our models, and the images con-taining objects within the testing categories in the test dataset are utilized as the test data for evaluating our method. This experiment is designed for exploring the generality of the proposal methods. Table 4 lists our comparison results between ours and objectness [4]. Still our method outperforms [4] in terms of object recall given the number of proposals.

6.1.4 Computational Time

The computational time comparison of the three methods is listed in Table 5. Our implementation is a mixture of Matlab and C++, just like [4], [3], and all the programs are run on a single core of Intel Xeon W3680 CPU with 3.33GHz. The computational time shown here includes all the steps at the test stage starting from loading images. As we see, our method is more than 10 times faster than [4], [3], because our method only utilizes the simple gradients in gray images as features, and 2D convolution for classification, which are very efficient.

7 CONCLUSION AND DISCUSSION

We propose a very efficient two-stage cascade SVM method for both class-specific and generic 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 (*i.e.* 8×8 pixels). We learn linear filters in each stage based on SVM formulations, 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.

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.

Our method is both fast and efficient, and we have shown a substantial improvement in speed and recall over two recent related work [4], [3]. Besides object detection, we believe that our work will contribute to many other research areas, such like segmentation [27] and stereo matching [28].

Our recent proposal generation method in [22] achieves the fastest running time among all popular object proposal generation methods [20], [29], and the most repeatable under different imaging conditions (*e.g.* illumination, rotation, scaling, blurring, *etc.*). However, the main issue of our 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. Unfortunately, as we stated above, for our method better localization quality can be achieved at the cost of higher computational cost. Thus, how to reduce such computational burden as well as improving the localization quality will be our future work.

ACKNOWLEDGMENTS

REFERENCES

[1] Y. Wei, W. Xia, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan, "Cnn: Single-label to multi-label," *arXiv preprint arXiv:1406.5726*, 2014.

[2] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *Computer Vision (ICCV), 2013 IEEE International Conference on.* IEEE, 2013, pp. 17–24.

[3] E. Rahtu, J. Kannala, and M. Blaschko, "Learning a category independent object detection cascade," in *ICCV'11*, 2011.

[4] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *PAMI*, 2012.

[5] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman, "Multiple kernels for object detection," in *ICCV'09*, 2009.

[6] C. Lampert, "An efficient divide-and-conquer cascade for nonlinear object detection," in *CVPR'10*, 2010, pp. 1022–1029.

[7] O. Chum and A. Zisserman, "An exemplar model for learning object classes," in *CVPR'07*, 2007, pp. 1–8.

[8] P. Felzenszwalb, R. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *CVPR'10*, 2010, pp. 2241–2248.

[9] C. Lampert, M. Blaschko, and T. Hofmann, "Efficient subwindow search: A branch and bound framework for object localization," *PAMI*, vol. 31, no. 12, pp. 2129–2142, December 2009.

[10] Z. Zhang, J. Warrell, and P. H. Torr, "Proposal generation for object detection using cascaded ranking svms," in *CVPR'11*, 2011.

[11] R. Herbrich, T. Graepel, and K. Obermayer, "Large margin rank boundaries for ordinal regression," in *Advances in Large Margin Classifiers*, 2000, pp. 115–132. [Online]. Available: http://stat.cs.tu-berlin.de/publications/papers/herobergrae99.ps.gz

[12] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results," http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[13] P. Gehler and S. Nowozin, "On feature combination for multiclass object classification," in *ICCV'09*, 2009, pp. 221–228.

[14] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *PAMI*, vol. 32, no. 9, pp. 1627–1645, September 2010.

[15] D. Park, D. Ramanan, and C. Fowlkes, "Multiresolution models for object detection," in *ECCV'10*, 2010, pp. 241–254.

[16] I. Endres and D. Hoiem, "Category independent object proposals," in *ECCV'10*, 2010, pp. 575–588.

[17] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, 2013. [Online]. Available: http://www.huppelen.nl/publications/selectiveSearchDraft.pdf

[18] V. Yanulevskaya, J. Uijlings, and N. Sebe, "Learning to group objects," in *CVPR*, 2014. [Online]. Available: http://www.huppelen.nl/publications/2014CVPR_LearningToGroupObjects_CameraReady.pdf

[19] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search," in *CVPR*, 2014.

[20] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" *arXiv preprint arXiv:1406.6962*, 2014.

[21] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool, "The PASCAL Visual Object Classes Challenge 2006 (VOC2006) Results," http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[22] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *CVPR*, 2014.

[23] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Hawaii, 2001.

[24] G. Heitz, S. Gould, A. Saxena, and D. Koller, "Cascaded classification models: Combining models for holistic scene understanding," in *Advances in Neural Information Processing Systems (NIPS 2008)*, 2008.

[25] M. J. Saberian and N. Vasconcelos, "Learning optimal embedded cascades," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 10, pp. 2005–2018, 2012.

[26] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[27] J. Carreira and C. Sminchisescu, "CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[28] M. Bleyer, C. Rhemann, and C. Rother, "Extracting 3d scene-consistent object proposals and depth from stereo images," in *Proceedings of the 12th European conference on Computer Vision - Volume Part V*, ser. ECCV'12. Berlin, Heidelberg: Springer-Verlag, 2012, pp. 467–481.

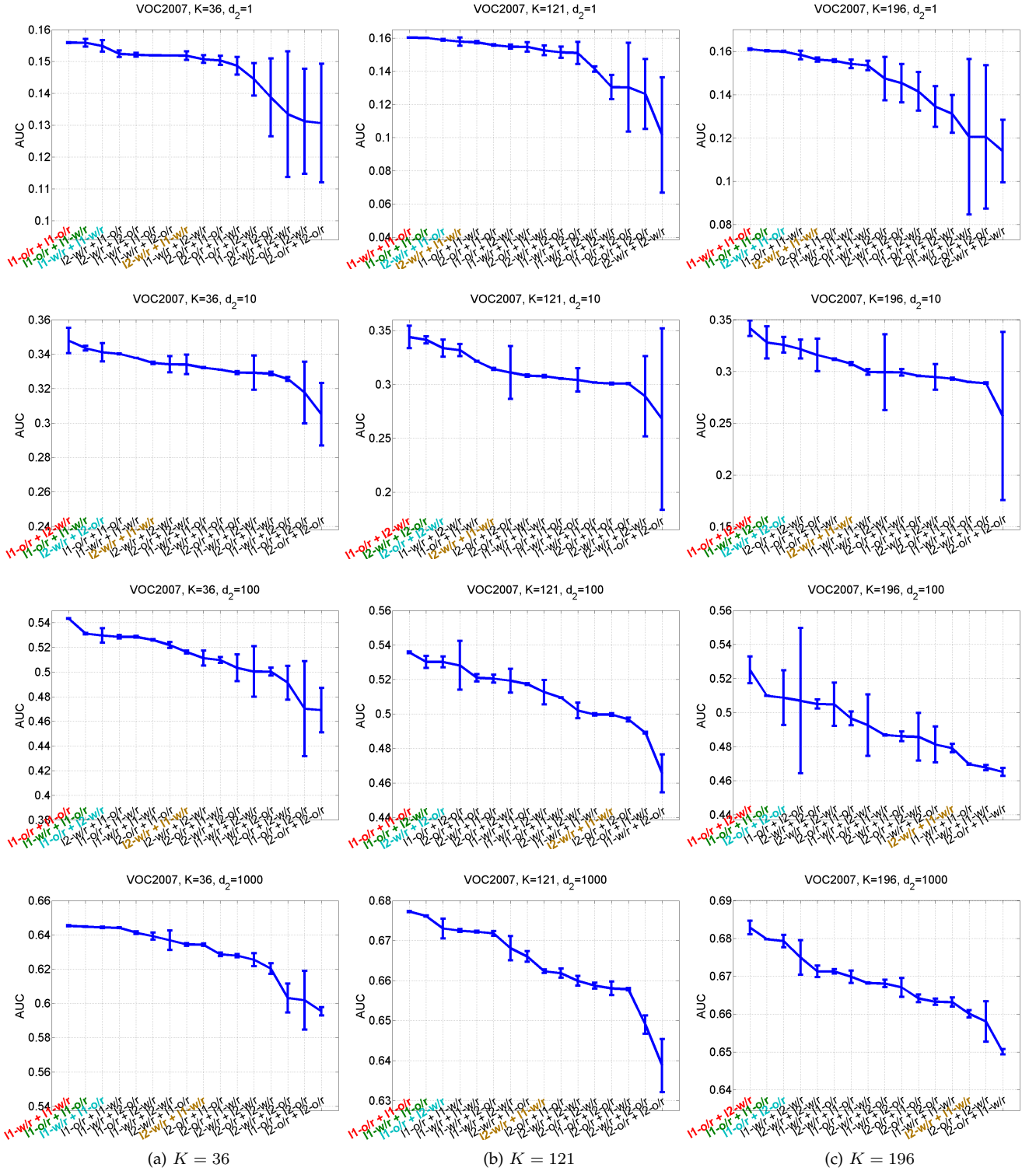[29] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," 2014.

Fig. 5. Comparison of different cascade settings (Stage I + Stage II) on VOC2007 using $K \in \{36, 121, 196\}$ and $d_2 \in \{1, 10, 100, 1000\}$, respectively. In each sub-figure, the cascade settings are sorted in descending order based on the means of different *area under object recall-overlap curves* (AUC) scores, the top 3 settings are colored by red, green, and cyan, respectively. Note that the setting "$\ell_2 - w/r + \ell_1 - w/r$" is the method proposed in [10], colored by yellow.
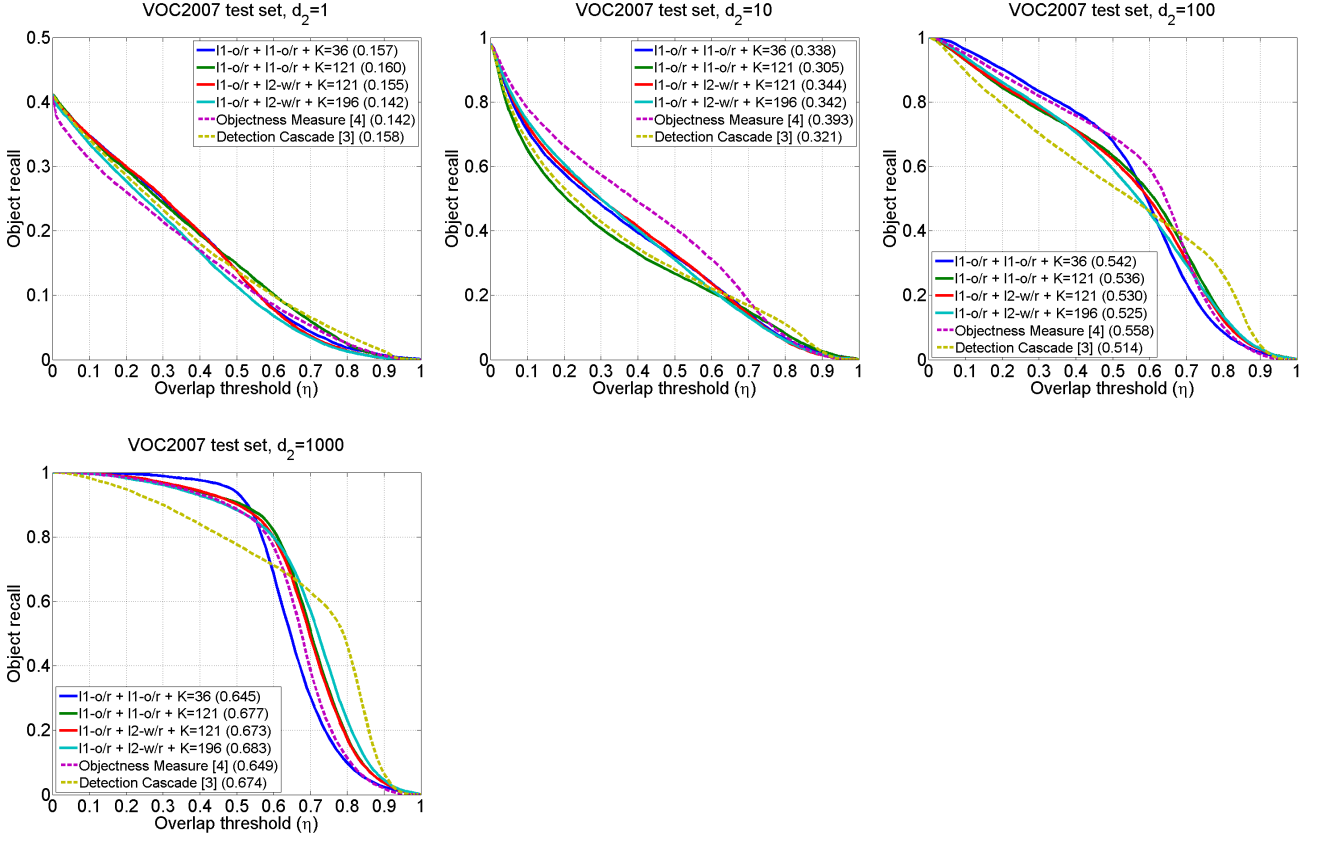
Fig. 6. Comparison of recall-overlap curves using different methods and d_2's on VOC2007. The numbers in the brackets are AUC scores for the methods. Among the 4 cascade settings, our method with larger K achieves better AUC scores using more proposals. Overall, with a same number of proposals, each individual method has similar behaviors. In terms of object recall at $\eta = 0.5$, ours and [4] perform very similarly, and both outperform [3] in most cases. But in terms of localization quality of proposals, [3] performs best.
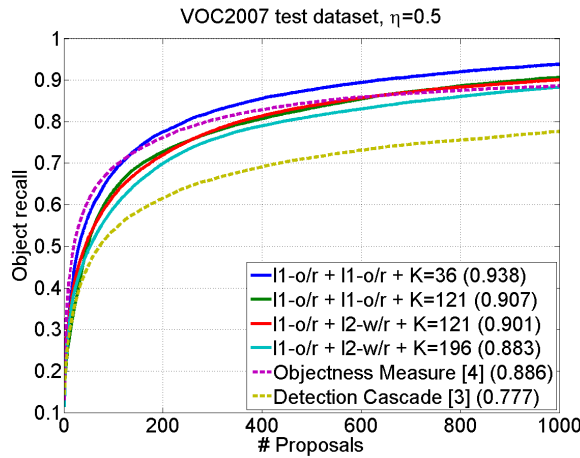

Fig. 7. Comparison of recall-proposal curves using different methods on VOC2007. The numbers in the brackets are the object recalls using 1000 proposals. Among the 4 cascade settings, $\ell_1 - o/r + \ell_1 - o/r$ performs best. Still our method and [4] have similar behaviors on both datasets, and both outperform [3] significantly. Using 1000 proposals, our method outperforms [4], [3] by 5.2% and 16.1% on VOC2007, respectively.
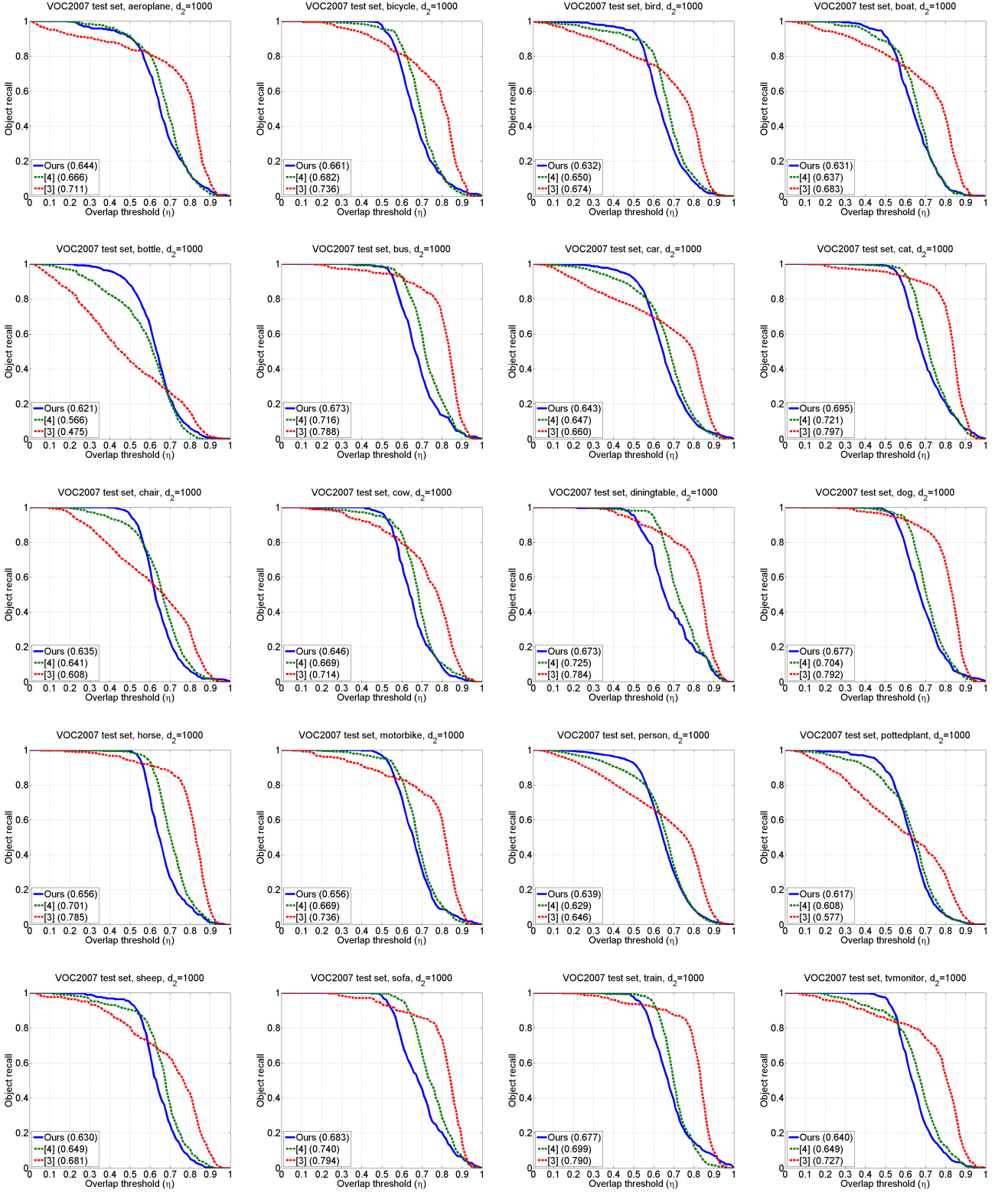
Fig. 8. Comparison of recall-overlap curves using different methods on each class in the test dataset of VOC2007. The numbers in brackets are the AUC scores for each method. In general, our method (*i.e.* $\ell_1 - o/r + \ell_1 - o/r$) performs similarly to [4], and when $\eta > 0.5$ [3] seems better than ours and [4] in terms of localization quality of proposals. The mean and standard deviation of AUC scores for our method, [4], [3] are $(65.1 \pm 2.2)\%$, $(66.8 \pm 4.2)\%$, and $(70.8 \pm 8.3)\%$, respectively.
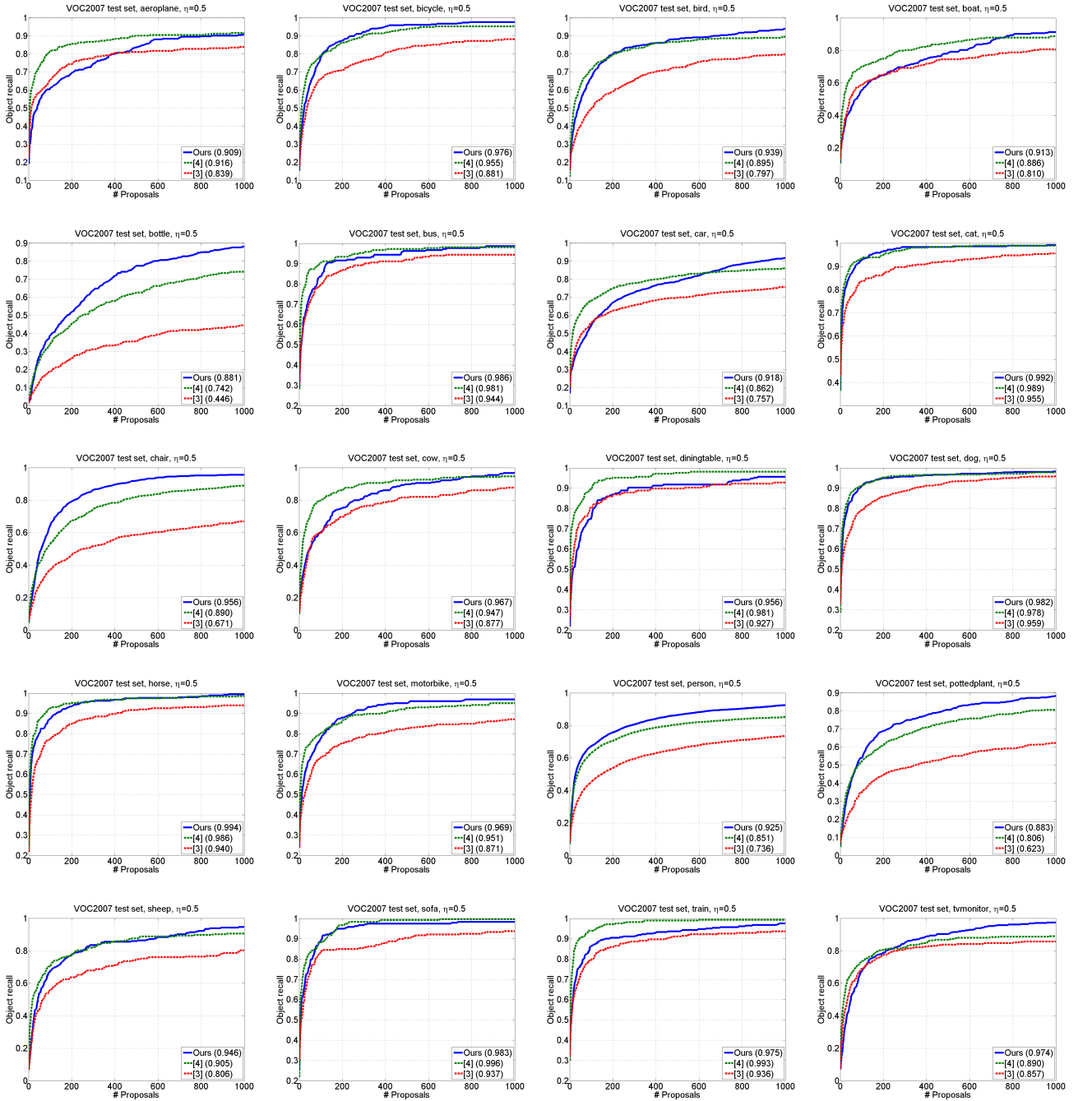
Fig. 9. Comparison of recall-proposal curves using different methods and $\eta = 0.5$ on each class in the test dataset of VOC2007. The numbers in brackets are the object recall values for each method using 1000 proposals. Still, in general our method (*i.e.* $\ell_1 - o/r + \ell_1 - o/r$) and [4] have similar behaviors, and both outperform [3] using 1000 proposals. The mean and standard deviation of the object recall values for our method, [4], [3] are $(95.1 \pm 3.5)\%$, $(92.0 \pm 6.7)\%$, and $(82.8 \pm 12.8)\%$, respectively.