IEICE TRANS. INF. & SYST., VOL.E99–D, NO.2 FEBRUARY 2016

PAPER

Improved Edge Boxes with Object Saliency and Location Awards

Peijiang KUANG[†a)], ***Nonmember***, **Zhiheng ZHOU**[†b)], ***Member***, ***and*** **Dongcheng WU**[††c)], ***Student Member***

SUMMARY Recently, object-proposal methods have attracted more and more attention of scholars and researchers for its utility in avoiding exhaustive sliding window search in an image. Object-proposal method is inspired by a concept that objects share a common feature. There exist many object-proposal methods which are either in segmentation fashion or engineering categories depending on low-level feature. Among those object-proposal methods, Edge Boxes, which is based on the number of contours that a bounding box wholly contains, has the state of art performance. Since Edge Boxes sometimes misses proposing some obvious objects in some images, we propose an appropriate version of it based on our two observations. We call the appropriate version as Improved Edge Boxes. The first of our observations is that objects have a property which can help us distinguish them from the background. It is called object saliency. An appropriate way we employ to calculate object saliency can help to retrieve some objects. The second of our observations is that objects 'prefer' to appear at the center part of images. For this reason, a bounding box that appears at the center part of the image is likely to contain an object. These two observations are going to help us retrieve more objects while promoting the recall performance. Finally, our results show that given just 5000 proposals we achieve over 89% object recall but 87% in Edge Boxes at the challenging overlap threshold of 0.7. Further, we compare our approach to some state-of-the-art approaches to show that our results are more accurate and faster than those approaches. In the end, some comparative pictures are shown to indicate intuitively that our approach can find more objects and more accurate objects than Edge Boxes.

key words: *detection proposals, saliency, object location, Edge Boxes*

1. Introduction

The goal of object detection is to determine whether an object exists in an image, and if so where in the image it occurs. The traditional approach to this problem is formulated as a classification problem in the well-known sliding window paradigm where the classifier evaluates over an exhaustive list of positions, scales, and aspect ratios. A typical sliding window detector requires about 10^6 classifier evaluations per image. One alternative approach to balance the tension between computational tractability and high detection quality is called object-proposal. The approach is inspired by the assumption that all objects of interest share common visual properties which distinguish them from the background. Instead of searching for an object at every image location, scale and aspect ratio, a set of object proposals, which are likely to contain objects, is first generated for reducing the set of positions that need to be further analyzed. If high recall can be reached with about 10^4 or less windows, significant efficiency improvement can be achieved, enabling the use of even more sophisticated classifiers. Several state-of-the-art object detection algorithms [1]–[3] use the framework of object proposals, which include the winners of the 2013 ImageNet detection challenge and top methods on the PASCAL VOC dataset [4].

High recall is a critical property of an object proposal generator. If a proposal is not generated nearby an object, the object can not be detected. An effective generator is able to obtain high recall with a modest number of candidate bounding boxes, typically ranging $10^3 \sim 10^4$ per image. There is also some speculation that the use of a small number of object proposals may even improve detection accuracy due to reduction of spurious false positive [5].

For this reason, many notable object-proposal methods strive to get high recall performance. For example, Geodesic Object Proposals [6] identifies critical level sets in geodesic distance transforms computed for seeds placed in the image. Those seeds are placed by specially trained classifiers that are optimized to discover objects. Objectness [7], [8] estimates a score based on a combination of multiple cues such as saliency, color contrast, edge density, location and size statistics, and how much such windows overlap with superpixel segments. In efficiency aspect, this method takes seconds per image. Selective Search [1], [9] is a method that carefully engineers features and scoring function, using the notion of superpixels as well. MCG [10] is one of the most recent methods proposing an improved multi-scale hierarchical segmentation, a new strategy to generate proposals by merging up to 4 segments, and a new ranking procedure to select the final object proposals. However, Selective Search and MCG take magnitudes of ten seconds. Edge Boxes [11] is a scoring function evaluated in a sliding window framework. It uses object boundaries as feature for the scoring which reaches a very high recall and efficiency performance. It only cost about 0.3 seconds per image.

Among those methods, Edge Boxes outperforms others in recall performance. It is worthy of promoting the recall performance of Edge Boxes since high recall is a critical property of an object-proposal method. We try to improve Edge Boxes with our observations, and call our approach as

Manuscript received June 9, 2015.
Manuscript revised September 19, 2015.
Manuscript publicized November 12, 2015.

[†]The authors are with the Electronic Information Engineering Institute, South China University of Technology, No.381 Wushan Road, Guangzhou, China.
[††]The author is with the Huiding Technology Co. Ltd., Shenzhen, China.
a) E-mail: pjkuang@foxmail.com
b) E-mail: zhouzh@scut.edu.cn
c) E-mail: dongcheng.wdc@gmail.com
DOI: 10.1587/transinf.2015EDP7222

Improved Edge Boxes.

In this paper, Sect. 2 briefly introduces the basic theory of Edge Boxes. Section 3 states our observations for promoting the the recall performance. Section 4 demonstrates the comparison between our approach, Edge Boxes and other well-known object-proposal methods. Section 5 concludes our work.

2. Edge Boxes

In this section, we briefly introduce Edge Boxes [11]. The basic concept of Edge Boxes is that object proposals are ranked based on a score computed by a scoring function. The score, which counts the number of contours wholly enclosed by a bounding box, indicates the likelihood of the box containing an object.

2.1 The Scoring Function of Edge Boxes

Given an image, the method first computes edges responses and groups them into a set S. For $s_i, s_j \in S$, their affinity $a(s_i, s_j)$ is formulated by:

$$a(s_i, s_j) = \mid cos(\theta_i - \theta_{ij})cos(\theta_j - \theta_{ij}) \mid^2 \tag{1}$$

where θ_{ij} is the angle between the mean positions of s_i and s_j, θ_i and θ_j is the mean orientations of s_i and s_j respectively.

To score a candidate bounding box b is based on edge groups affinities. The approach of Edge Boxes categorizes edge groups into overlapped groups, outside groups, wholly contained groups based on the location relationship between the bounding box and the edge group.

For overlapped groups (denoted as S_b) and outside groups, their weights $w_b(s_i)$ are set to 0 indicating they are not wholly contained by the box. For wholly contained groups, although they are inside the box, they may be generated by the object outside the box. So it needs a measurement to calculate their weights.

$$w_b(s_i) = 1 - \max_T \prod_j^{|T|-1} a(t_j, t_{j+1}) \tag{2}$$

where T is an ordered path of edge groups with a length of $\mid T \mid$ that begins with some $t_1 \in S_b$ and ends at $t_{|T|} = s_i$. This is to find the maximum affinity between s_i and an edge on the box boundary. For instance, if the maximum affinity of s_i with an edge on the box boundary is 0.6, it means 0.4 of magnitude of s_i is wholly contained by the bounding box b, and it also means s_i do a contribution proportional to 0.4 of its magnitude to the score.

Finally, a score of candidate bounding box b is formulated as:

$$h_b = \frac{\sum_i w_b(s_i)m_i}{2(b_w + b_h)^\kappa} \tag{3}$$

where the width and height of box is b_w and b_h respectively, $\kappa = 1.5$ for offsetting the bias of larger windows containing more edges on average. Besides, the authors of Edge Boxes observes that the edges in the center of the box are of less importance than those near the border of the box. So, subtracting the edges magnitudes from a box centered in b is finally used.

$$h_b = \frac{\sum_i w_b(s_i)m_i}{2(b_w + b_h)^\kappa} - \frac{\sum_{p \in b^{in}} m_p}{2(\frac{b_w}{2} + \frac{b_h}{2})^\kappa} \tag{4}$$

where m_p is the magnitude of edge pixel, and $\frac{b_w}{2}$ and $\frac{b_h}{2}$ is the width and height of the center box in b respectively.

3. Improve Recall Performance by Extra Object Awards

3.1 Overview of Our Approach

Edge Boxes believes that the number of contours in the box represents the probability of containing an object. However, sometimes objects are flat or smooth, so objects do not contain strong contours. For this reason, Edge Boxes sometimes misses proposing some visually obvious objects in some images. So, we propose two extra object properties to help retrieve those objects.

Edge Boxes ranks object proposals based on the score computed from scoring function (4). The method sets a threshold and rejects those proposals with a low score. Due to some reasons described as above, some rejected object proposals with a low score may contain objects. We classify those rejected proposals into two classes. One is proposals with objects (false-negative proposals), the other is proposals without objects (true-negative proposals). If those false-negative proposals can be retrieved, recall performance can be improved.

We propose a simple rescoring approach to the false-negative proposals. The new scoring function gives more scores to the false-negative proposals than the true-negative proposals. As a result, the scores of the false-negative proposals are higher than the score threshold and they can be retrieved.

Guided by this idea, we propose two extra scores called as object awards. They are object saliency award and object location award. Object saliency means that the object is salient compared to its background. Object saliency award of false-negative proposal should be greater than that of true-negative proposal. And the object location award is based on the observation that objects 'prefer' to appear at the center part of the image. If a candidate bounding box appears around the center part of the image, it may be more likely to contain an object. So, it should be assigned an extra score. We will give some results in Sect. 4 to prove our assumptions, and we state these two awards in the following.

3.2 Object Saliency Award

By object saliency award we mean a degree that an object differs from background. In this paper, a patch-based model

is employed to measure the object saliency. An image is partitioned into several image pathes. Intuitively, if many patches are classified to the same class, those patches should be regarded as patches with low saliency. Conversely, if just one patch is classified to a certain class, it can be treated as a patch with high saliency. Inspired by this assumption, a simple random forest method is used [12] to calculate the saliencies of patches for efficiency.

In a typical object recognition data, we find that, the degree of color variation of object is relatively greater than that of background. Based on this observation, a decision tree T_k can be constructed by using color variation as splitting function.

Similarity to [13], we use RGB and Lab color spaces are used to together to represent the color feature when an image is input. Patches are grid-sampled from top-left to bottom-right of the image without overlap and then reshaped into column vectors respectively. The length of each vector is $6r^2$($r * r$ is the size of one patch, $r = 16$ in our paper) and further we denote all vectors as $P = \{p_1, p_2, \ldots, p_n\}$. If the width and height of the image are not divisible by r, the remaining pixels are left aside and assigned a mean saliency score at the end. Random forest is an ensemble of T trees, formulated as $F = \{T_1, T_2, \ldots, T_T\}$ and each tree in F is a decision tree.

The splitting function at node n in tree T_k is formulated by:

$$f_n(S_n, h_1, h_2) = \begin{cases} p_i \in S_l & \text{if } d_i(h_1, h_2) \le \theta_{h_1,h_2} \\ p_i \in S_r & \text{otherwise} \end{cases} \quad (5)$$

where h_1, h_2 are two different random generating indices, $d_i(h_1, h_2) = (p_i(h_1) - p_i(h_2))^2$, S_n is the patches contained in node n while S_l and S_r are the patch sets contained in node n's left and right child respectively, θ_{h_1,h_2} is the mean $d_i(h_1, h_2)$ of S_n.

Splitting process is terminated at a node when it reaches the max depth or it only contains one patch. The nodes in the max depth or the nodes, which contain only one patch, are called leaf nodes. After the random forest is built, saliency score function is formulated by:

$$S_{saliency}(p_i) = \frac{1}{\frac{1}{T}\sum_{k=1}^{T} n_k} \quad (6)$$

where n_k is the number of patches in the leaf node where p_i resides in tree T_k, and T is the number of trees in forest F. Worthy of mentioning that each pixel in image patch p_i shares the saliency score.

In Eq. (5), we define the decision criteria using $d_i(h_1, h_2)$ so that the patch with high color-variation goes into the same group. Because patches with high color variation are usually less than those with low color variation in natural images, the splitting process in the group where those patches with high color variation reside will be always terminated quickly. As a result, the patch with high color variation resides in the leaf which contains either a patch or a few patches. In other words, n_k in the Eq. (6) is a rather small value. So, patches with high color variation can be scored high by Eq. (6). Namely, patches with high color variation belonging to object will be scored high by Eqs. (5) and (6).

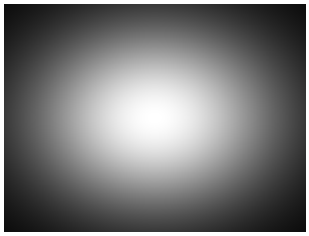


Fig. 1 Location Prior. The brighter it is, the higher score it gets. If a bounding box appears around the center part of the image, it gets a high location award.

3.3 Object Location Award

The concept of location award mainly comes from our intuition, but we have some evidences to support our intuition in Sect. 4. In many cases, objects will be more likely to appear at center part than other parts of images. Many images, pictures or photos are taken for capturing objects. As a result, object may always appear around the center part of the image. Inspired by this observation, a candidate bounding box which appears at the center part of the image ought to be scored higher than those which appear far away from the center part. Under this assumption, we use a Gaussian-function to model this award.

We model the location award using two-dimensional Gaussian:

$$\begin{aligned} g(\mathbf{x} - \mathbf{x_c}) &= exp(-\frac{1}{2}(\mathbf{x} - \mathbf{x_c})^T \Sigma^{-1} (\mathbf{x} - \mathbf{x_c})) \\ \Sigma &= \begin{bmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{bmatrix} \end{aligned} \quad (7)$$

σ_1 controls the horizontal width of kernel function, and σ_2 controls the vertical width of kernel function. In practice, $\mathbf{x}$ and $\mathbf{x_c}$ are normalized to 1 and $\sigma_1 = \sigma_2 = 0.05$ for better results.

In general cases, $(\frac{1}{2}, \frac{1}{2})$ can be assigned to $\mathbf{x_c}$, and $\mathbf{x}$ can be the center point of the candidate bounding box. When the candidate bounding box gets close to the center part of an image, the distance between $\mathbf{x}$ and $\mathbf{x_c}$ decreases and this candidate bounding box gets a higher object location award. Figure 1 shows the location prior. The brighter it is, the higher score it gets. Namely, if a bounding box appears around the center part of the image, it gets a high location award.

Letting $\mathbf{x}$ be the center of a candidate bounding box, we define the location award of a candidate bounding box by:

$$\begin{aligned} S_{location}(\mathbf{x}) &= g(\mathbf{x} - \mathbf{x_c}) \\ &= exp(-\frac{1}{2}(\mathbf{x} - \mathbf{x_c})^T \Sigma^{-1} (\mathbf{x} - \mathbf{x_c})) \end{aligned} \quad (8)$$

3.4 Combination of Three Scores

The origin score function is:

$$h_b = \frac{\sum_i w_b(s_i)m_i}{2(b_w + b_h)^\kappa} - \frac{\sum_{p \in b^{in}} m_p}{2(b_w + b_h)^\kappa} \tag{9}$$

We modify the scoring function Eq. (9) by combing Eqs. (6) and (8) to give the scoring function:

$$S = h_b + \alpha S_{location} + \beta S_{saliency}; \tag{10}$$

where α and β are set to control the balance between h_b and awards. The way to choose parameters α and β is shown in Sect. 4.

4. Results and Comparisons

In this section, we first show that our saliency score indeed meets our goal. Next, we verify our location assumption in Sect. 3.3. Finally, we study the influence of parameters.

4.1 Results of Saliency Score

Figure 2 visualize saliency scores. Intuitively, we can see that patches belonging to an object are brighter than those belong to background. In other words, patches belonging to objects have higher saliency score than that of background patches.

We do another more convincing experiment based on PASCAL VOC 2007 dataset. We calculate the saliency score of each image. In each image, we compute the mean saliency score of the object and the background respectively. We compute the mean saliency score of the object and the background in the following manner: firstly, we compute the mean saliency score of the object. Next, we subtract saliency scores of those patches belonging to objects from the saliency score image, so the mean saliency score of the remaining patches is the saliency score of the background.

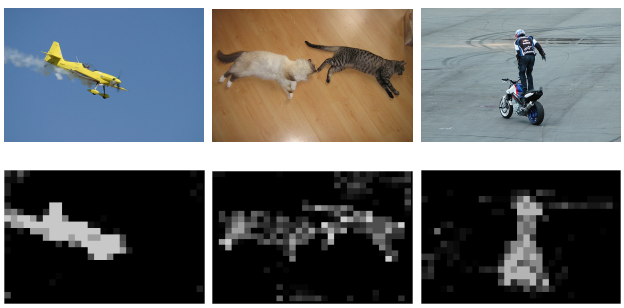


Fig. 2 Saliency score. The top are the original images. The bottom are the visualization of their saliency score.

Table 1 shows saliency score of objects and non-objects of several categories. Taking aeroplane as an example, the saliency score of the object aeroplane is much greater than that of background because patches belonging to background, blue sky, are almost the same. We observed this tendency in other categories too.

So, a conclusion summarized from Table 1 is that no matter what category the object is, its saliency score is greater than that of background using our saliency computing method. In other words, if a candidate box has a high saliency score, it may contain an object. Therefore, adding saliency score $S_{saliency}$ to h_b can give more awards to those bounding boxes which are more likely to contain objects. In this way, we can retrieve some objects which are not found in Edge Boxes to promote recall performance.

4.2 Analysis of Location Awards

The location awards are base on our assumption. We believe that many images are taken for capturing objects. As a result, objects may always occur near the center part of the image. In this subsection, we want to prove that our assumption is right to some degree. We analyze objects' locations in PASCAL VOC 2007 dataset. If the center point of an image is denoted as $\mathbf{p_c}$ and the center point of the object's bounding box is denoted as $\mathbf{p_b}$, we record the distance between $\mathbf{p_c}$ and $\mathbf{p_b}$. Our goal is to see how much percentages of objects are in the circle with radius r and what its density is. $\mathbf{p_c}$, $\mathbf{p_b}$ are normalized to 1. Results shown as Table 2.

Table 2 shows the percentage and the density of objects in the circle with radius r. And 'density' tells us the percentage of aeroplanes in a unit area. For instance, the category 'aeroplane' in Table 2 means that there are 76.21% of aeroplanes in the circle with radius 0.3, and as the radius increases, the percentage increases too. However, there are about 2.70% of aeroplanes in a unit area when the circle's radius is 0.3. And as the radius increases, the density decreases. This pattern can be seen in other categories. It means that no matter what category the object is, object densely appears at the center part of an image. Reversely, we can say if a candidate box appears near the center parts of images, it is likely to contain an object.

Table 2 indicates that our assumption is correct. No matter what category the objects are, objects densely appear at the center part of the image. Namely, objects are likely to appear around the center part of images. So, if a

Table 1 Saliency comparison between objects and non-objects

	aeroplane	**bicycle**	**bird**	**boat**	**bottle**	**bus**	**car**	**cat**	**chair**	**cow**
objects	0.176	0.105	0.105	0.140	0.119	0.115	0.118	0.080	0.088	0.085
non-objects	0.030	0.035	0.036	0.043	0.043	0.043	0.044	0.036	0.037	0.032
	dinning table	**dog**	**horse**	**motorbike**	**person**	**potted plant**	**sheep**	**sofa**	**train**	**tv monitor**
objects	0.082	0.083	0.081	0.098	0.097	0.088	0.096	0.070	0.090	0.104
non-objects	0.033	0.035	0.045	0.035	0.040	0.040	0.037	0.030	0.033	0.040

Table 2 Percentage and density of objects in the circle with radius r

	aeroplane		**bicycle**		**bird**		**boat**		**bottle**	
	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)
r=0.3	76.21	2.70	63.58	2.25	66.76	2.37	64.21	2.28	48.91	1.73
r=0.35	82.04	2.14	73.62	1.91	76.09	1.97	75.79	1.97	61.21	1.59
r=0.4	86.89	1.73	82.28	1.64	82.51	1.64	83.51	1.66	73.68	1.47
r=0.45	91.26	1.22	91.14	1.43	90.96	1.43	91.93	1.45	86.92	1.37
	bus		**car**		**cat**		**chair**		**cow**	
	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)
r=0.3	46.62	1.65	47.25	1.67	80.58	2.85	49.87	1.76	65.07	2.30
r=0.35	58.45	1.52	56.72	1.47	88.85	2.31	60.28	1.57	76.08	1.98
r=0.4	69.81	1.39	66.95	1.33	93.17	1.85	74.27	1.48	85.65	1.70
r=0.45	82.37	1.29	80.31	1.25	96.76	1.52	85.57	1.35	94.74	1.49
	dining table		**dog**		**horse**		**motorbike**		**person**	
	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)
r=0.3	48.12	1.70	76.87	2.72	74.32	2.63	58.56	2.07	56.71	2.00
r=0.35	60.62	1.58	85.94	2.23	80.86	2.10	69.39	1.80	67.60	1.76
r=0.4	74.33	1.49	91.61	1.82	85.36	1.70	78.52	1.56	77.69	1.55
r=0.45	85.75	1.35	95.46	1.50	91.67	1.44	90.30	1.42	88.11	1.38
	potted plant		**sheep**		**sofa**		**train**		**tv monitor**	
	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)	percentage(%)	density(%)
r=0.3	50.37	1.78	55.21	1.95	54.45	1.93	69.83	2.47	54.21	1.92
r=0.35	62.64	1.63	68.75	1.79	64.89	1.67	76.86	2.00	67.33	1.75
r=0.4	74.54	1.48	79.69	1.59	79.13	1.57	86.36	1.71	80.20	1.60
r=0.45	86.25	1.36	89.06	1.40	89.57	1.41	93.80	1.47	90.59	1.42

Table 3 Recall performance in training set and test set when generating 5000 proposals

recall	$\sigma = 0$	$\sigma = 0.005$	$\sigma = 0.05$	$\sigma = 0.1$	$\sigma = 0.5$	$\sigma = 1$
training set	87.21%	88.06%	**88.09%**	88.03%	88.02%	87.21%
test set	87.13%	88.00%	**88.04%**	87.96%	87.96%	87.13%

candidate bounding box appears around the center part of an image, it may be more likely to contain an object. So, adding $S_{location}$ to h_b prefers to give an extra award to those candidate bounding boxes which are more likely to contain objects. As a result, this helps us to improve recall performance.

4.3 Parameter Influences

4.3.1 Influences of σ_1 and σ_2 in Location Award

σ_1 and σ_2 in Eq. (7) control the range of the location prior. To keep simpleness, we set $\sigma_1 = \sigma_2 = \sigma$. If σ is too large, the range of the location prior becomes large, which means that everywhere in the image has a location award. It is equal to that everywhere in the image has no location award which makes no improvement in recall performance. Reversely, if σ is too small, the range of the location prior is too small which means that only few location in the image can have a location award. It does not have any effect to promote the location award.

We split the PASCAL VOC 2007 dataset into training set and test set. In training set, σ is varied to find its optimal value. And we compare the recall performance between training set and test set to study its generalization performance. Shown as Table 3.

Table 3 shows that when σ becomes small, for instance $\sigma = 0$, recall performance decreases. Reversely, when σ becomes large, for instance $\sigma = 1$, recall performance decreases too. It is appropriate to set $\sigma = 0.05$ because it can reach a high recall. In other perspective, recall of test set is a little behind that of training set when $\sigma = 0.05$, so $\sigma = 0.05$ has a good generalization performance.

4.3.2 Influences of α and β in Score Function

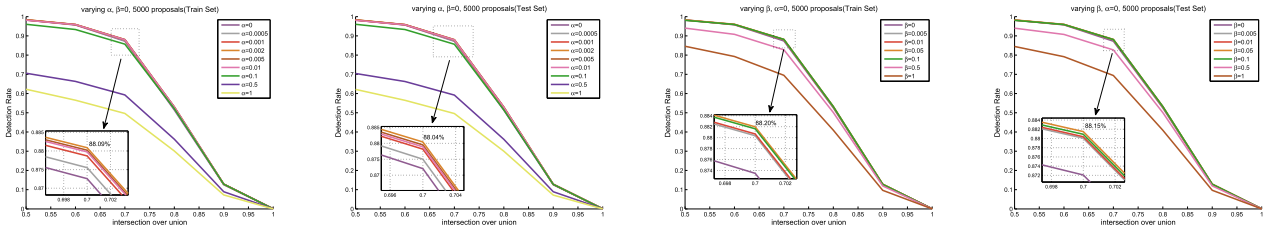
Our final scoring function (10) includes two parameters α and β. Parameters α and β balance h_b and awards. Testing various α and β respectively when generating 5000 object proposals to find the optimal α and β is the goal of this subsection.

Figure 3 illustrates the scoring function's behavior when varying parameters α and β.

Generally, it is reasonable in object detection applications that Intersection of Union (IoU) is 0.7, because the condition that IoU is 0.5 is too loose. IoU is 0.5 means that a bounding box only find half of the object which is obviously not suitable for some applications. Moveover, the condition that IoU is 0.9 is too strict. IoU is 0.9 means that a bounding box nearly finds the whole object. This is hard to achieve in practical. Thus, our following analysis focuses on the recall performance when IoU is 0.7.

The PASCAL VOC 2007 dataset is divided into training set and test set. In training set, α and β are varied to find their optimal values. And the comparison between recall rate of training set and that of test set is to study the generalization performance of α and β.

From Fig. 3 (a), we vary α from 0 to 1 keeping $\beta = 0$ unchanged when generating 5000 proposals. Either α is too large or too small, recall performance decreases in a degree. When $\alpha = 0.002$, recall rate is the best. The similar pattern can be seen in Fig. 3 (c). Recall rate of $\beta = 0.05$ is better



(a) varying α in training set (b) varying α in test set (c) varying β in training set (d) varying β in test set

Fig. 3 The results of varying α and β in training set and test set

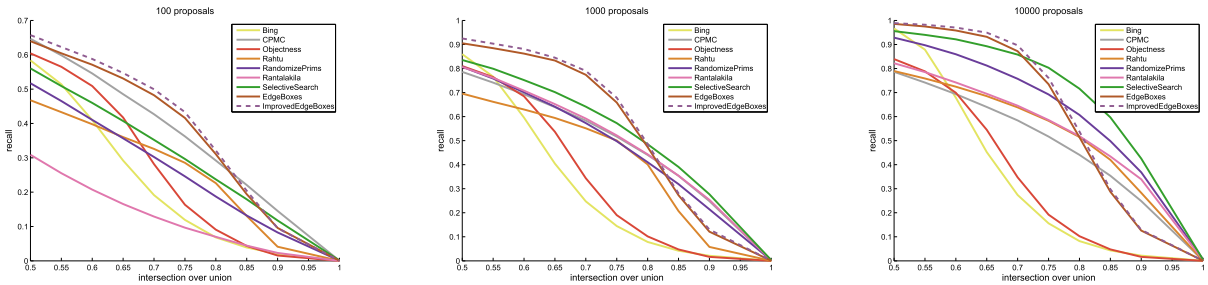


Fig. 4 Comparison between Improved Edge Boxes and various object proposals algorithm

Table 4 Results of our approach compared to other methods

	BING [22]	Rantalankila [18]	Objectness [5]	RandomizePrims [20]	
Recall	29%	68%	39%	80%	
Time	**0.25s**	10s	3s	1s	
	Rahtu [21]	Selective Search [1]	CPMC [19]	Edge Boxes [11]	Improved Edge Boxes
Recall	70%	**87%**	65%	**87%**	**89%**
Time	3s	10s	250s	**0.36s**	**0.43s**

than other values of β. Therefore, $\alpha = 0.002$ and $\beta = 0.05$ in practice can promote the recall performance in a degree.

From Figs. 3 (a) and 3 (b), recall rate of training set is 88.09% when IoU=0.7, $\alpha = 0.002$ and $\beta = 0$ while that of test set is 88.04%. The recall rate of test set is just a little behind that of training set which indicates that $\alpha = 0.002$ doesn't cause serious overfitting in training set and it has a good generalization performance. Similarly, recall rate of training set is 88.20% when IoU=0.7, $\beta = 0.05$ and $\alpha = 0$ while that of test set is 88.15%. $\beta = 0.05$ doesn't seriously overfit the training set which indicates that $\beta = 0.05$ reach a good generalization performance.

4.4 Recall Performance Comparison

We compare our Improved Edge Boxes algorithm against various object proposals algorithms. Results of all competing methods were provided by Hosang et al. [23] in a standardized format. When $\sigma = 0.05$, $\alpha = 0.002$ and $\beta = 0.05$, our approach achieves over 89% object recall at the overlap threshold of 0.7 in test set. And Fig. 4 shows the recall performance when varying the IoU threshold for different numbers of proposals. Compared to Edge Boxes, we improve its recall performance in a degree for different numbers of proposals. Compared to other proposals algorithms, although our recall performance is behind Selective Search [1], Rantalakila [18], RandomizePrims [20] in high IoU, it takes the lead in the low and middle IoU. Because a high IoU like 0.9 is not practical due to this high IoU criteria is too strict. It's difficult to achieve in practice. As a result, it may lose many objects that following steps could never find. In other words, if we choose an appropriate IoU like 0.7, our proposals algorithm, Improved Edge Boxes, can return a satisfactory recall performance.

Finally, we compare the runtime of our approach to other methods in Table 4.

We add saliency score and location score to h_b which needs extra computing time. However, Table 4 shows that our approach, Improved Edge Boxes, is still faster than other methods except Edge Boxes and BING. But, BING has the worst accuracy of all evaluated methods at IoU of 0.7. The methods with comparable accuracy are Selective Search, but it is considerably slower. Compared to Edge Boxes, our runtime performance is behind Edge Boxes, but we lead in the recall performance. In other words, we improve the recall performance with a minor loss in runtime. However, runtime of our approach is still fast compared to other approaches.

Finally, qualitative comparison results between Edge Boxes and our approach are shown in Fig. 5. In Fig. 5, Column (a) and (c) are the results of Edge Boxes, and Column (b) and (d) are the results of our approach. Bounding boxes

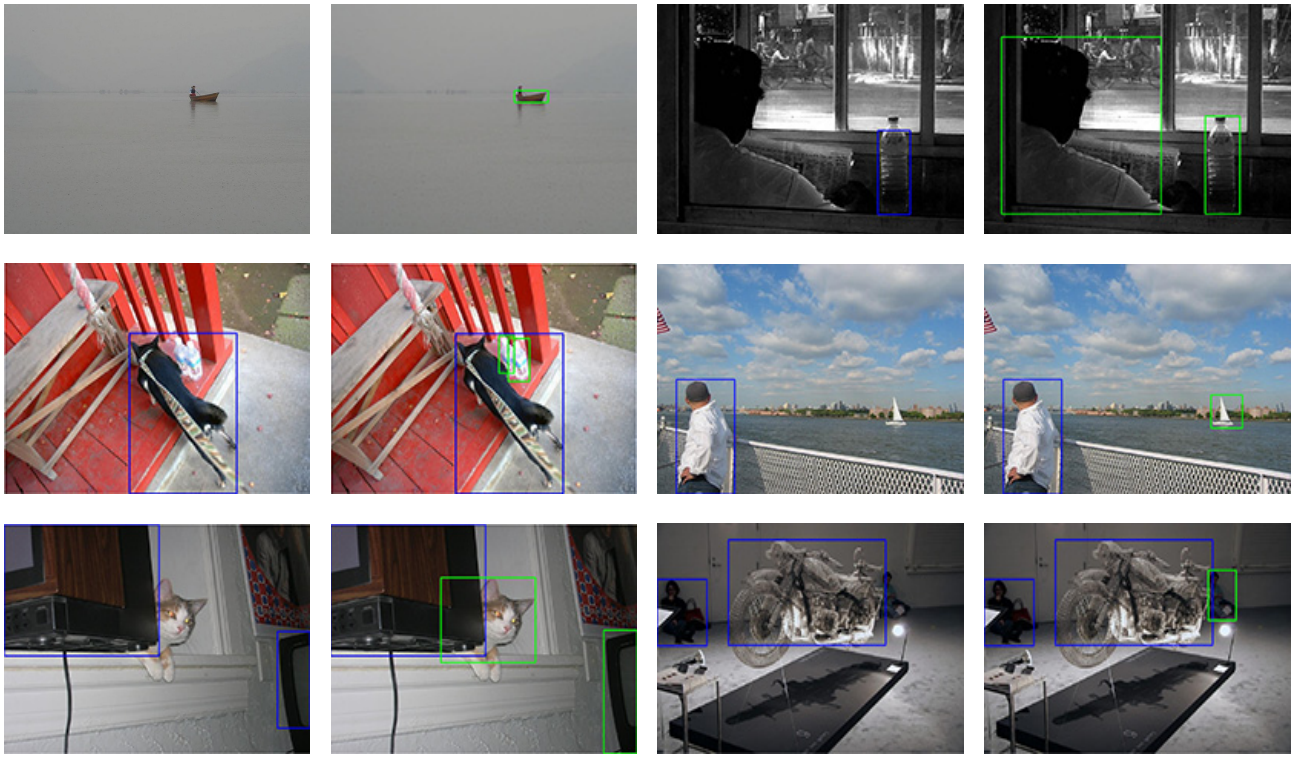


Fig. 5 Qualitative comparison examples between Edge Boxes and our approach. Column (a) and (c) are the results of Edge Boxes, Column (b) and (d) are results of our approach. A blue bounding box indicate an object is found by both approaches, and a green bounding box indicates an object is found by our approach.

are shown in blue and green. A blue bounding box indicates an object is found by both approaches, and a green bounding box indicates an object which can not be found by Edge Boxes is found by our approach.

5. Conclusion

In this paper we describe the recall performance of an effective method, Edge Boxes, for finding object proposals in images. We propose two extra observations: object saliency and object location. We state two scoring function to compute object saliency score and object location score. Both of them serve as extra awards to the original score. Last, we combine the original score, object saliency awards and object location awards to evaluate the possibility that indicates a candidate box contains an object.

Recall performance is the most important in object proposals approaches, because if an object can not be found in object proposed step, it will never be found in the following steps. For this reason, we try to promote the recall performance of Edge Boxes. Compared to other approaches, we indeed improve recall performance, sacrificing a minor runtime.

Acknowledgements

The work is supported by National Natural Science Foundation of China (61372142, U1401252, 61571005), Guangdong Province Science and technology plan (2013B010102004, 2013A011403003).

References

[1] J.R.R. Uijlings, K.E.A. van de Sande, T. Gevers, and A.W.M. Smeulders, "Selective Search for Object Recognition," IJCV, vol.104, no.2, pp.154–171, Sept. 2013.
[2] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for Generic Object Detection," Proc. IEEE International Conf. on Computer Vision, Sydney, Australia, pp.17–24, 2013.
[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, USA, pp.580–587, 2014.
[4] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," IJCV, vol.88, no.2, pp.303–338, June 2010.
[5] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the Objectness of Image Windows," IEEE Trans. Pattern Analysis and Machine Intelligence, vol.34, no.11, pp.2189–2202, Nov. 2012.
[6] P. Krähenbühl and V. Koltun, "Geodesic Object Proposals," Proc. 13th European Conf. Computer Vision, Zurich, Switzerland, vol.8693, pp.725–739, Sept. 2014.
[7] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?," Proc. IEEE Conf. Computer Vision and Pattern Recognition, San Francisco, USA, pp.73–80, June 2010.
[8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," Proc. IEEE Conf.

Computer Vision and Patter Recognition, Miami, USA, pp.248–255, June 2009.
[9] K.E.A. van de Sande, J.R.R. Uijlings, T. Gevers, and A.W.M. Smeulders, "Segmentation as selective search for object recognition," Proc. IEEE Conf. Computer Vision, Barcelona, Spain, pp.1879–1886, Nov. 2011.
[10] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale Combinatorial Grouping," Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, USA, pp.328–335, June 2014.
[11] C.L. Zitnick and P. Dollár, "Edge Boxes: Locating Object Proposals from Edges," Proc. 13th European Conf. on Computer Vision, Zurich, Switzerland, vol.8693, pp.391–405, Sept. 2014.
[12] L. Breiman, "Random forests," Machine Learning vol.45, no.1, pp.5–32, Oct. 2001.
[13] A. Borji and L. Itti, "Exploiting local and global patch rarities for saliency detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, USA, pp.478–485, June 2012.
[14] C. Gu, J.J. Lim, P. Arbelaez, and J. Malik, "Recognition using regions," Proc. IEEE Conf. Computer Vision and Pattern Recognition, Miami, USA, pp.1030–1037, June 2009.
[15] T. Deselaers, B. Alexe, and V. Ferrari, "Localizing Objects While Learning Their Appearance," Proc. 11th European Conf. Computer Vision, Heraklion, Crete, Greece, pp.452–466, Sept. 2010.
[16] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, USA, pp.733–740, June 2012.
[17] G. Yu, J. Yuan, and Z. Liu, "Unsupervised random forest indexing for fast action search," Proc. IEEE Conf. Computer Vision and Pattern Recognition, Providence, USA, pp.865–872, June 2011.
[18] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating Object Segmentation Proposals Using Global and Local Search," Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, USA, pp.2417–2424, June 2014.
[19] Joo Carreira, and C. Sminchisescu. "CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts," IEEE Trans. Pattern Anal. Mach. Intell., vol.34, no.7, pp.1312–1328, 2012.
[20] S. Manen, M. Guillaumin, and L.V. Gool, "Prime Object Proposals with Randomized Prim's Algorithm," Proc. IEEE International Conf. Computer Vision, Sydney, Australia, pp.2536–2543, Dec. 2013.
[21] E. Rahtu, J. Kannala, and M. Blaschko, "Learning a category independent object detection cascade," Proc. IEEE International Conf. on Computer Vision, Barcelona, Spain, pp.1052–1059, Nov. 2011.
[22] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "Binarized normed gradients for objectness estimation at 300fps," Proc. IEEE Conf. Computer Vision and Pattern Recognition, Columbus, USA, pp.3286–3293, June 2014.
[23] J. Hosang, R. Benenson, and B. Schiele, "How good are detection proposals, really?" Proc. British Conf. Machine Vision, Nottingham, Britain, 2014.
[24] P. Viola and M.J. Jones, "Robust Real-Time Face Detection," IJCV, vol.57, no.2, pp.137–154, May 2004.
[25] J. Canny, "A Computational Approach To Edge Detection," PAMI, vol.8, no.6, pp.679–698, Nov. 1986.

Peijiang Kuang received the B.E. degree in information engineering and is M.E. degree in reading in information engineering in South China University of Technology, Guangzhou, China. He is currently a student with South China University of Technology. His research orientations are image processing and machine learning.

Zhiheng Zhou received the B.S. and M.S. degrees in applied mathematics and the Ph.D. degree in electronic and information engineering from South China University of Technology, Guangzhou, China, in 2000, 2002, and 2005, respectively. He is currently a Professor with South China University of Technology. His research interests include image processing and image and video transmission.

Dongcheng Wu received the B.E. degree in information engineering and is M.E. degree in reading in information engineering in South China University of Technology, Guangzhou, China. He is currently a system development algorithm Engineer with Huiding Technology Company in Shenzhen, China. His research orientations are image processing and objects tracking.