A two-stage cascaded foreground seeds generation for parametric min-cuts

Shao-Mei Li, Jun-Guang Zhu, Chao Gao and Chun-Wei Li
National Digital Switching System Engineering and Technological Research and Development Center
Zhengzhou, Henan - China
[e-mail: lishaomei_may@126.com; zjgndsc@126.com;275237266@qq.com; 2093227141@qq.com]
*Corresponding author: Shao-Mei Li

Received March 7, 2016; revised September 1, 2016; accepted October 24, 2016; published November 30, 2016

Abstract

Parametric min-cuts is an object proposal algorithm, which can be used for accurate image segmentation. In parametric min-cuts, foreground seeds generation plays an important role since the number and quality of foreground seeds have great effect on its efficiency and accuracy. To improve the performance of parametric min-cuts, this paper proposes a new framework for foreground seeds generation. First, to increase the odds of finding objects, saliency detection at multiple scales is used to generate a large set of diverse candidate seeds. Second, to further select good-quality seeds, a two-stage cascaded ranking classifier is used to filter and rank the candidates based on their appearance features. Experimental results show that parametric min-cuts using our seeding strategy can obtain a relative small pool of proposals with high accuracy.

Keywords: Object proposal, foreground seed , two-stage cascaded ranking , parametric min-cuts

This research was supported by a research grant from National Science and Technology Support Plan of China (No.2014BAH30B01), and National Natural Science Foundation of China(No.61521003).

http://dx.doi.org/10.3837/tiis.2016.11.020 **ISSN : 1976-7277**

1. Introduction

Object Proposal [1-4] is a fast object location method. It is always used as a pre-process step in many computer vision areas, such as object detection [5], text detection [6], target tracking and image segmentation [5-8]. Compared with exhaustive sliding window search, objet proposal can help to reduce the search space from 10^6 to 10^3, which can greatly improve computation efficiency.

Current object proposal methods can be classified into three types [1,9] : objectness scoring [10-12], superpixel merging [3,5,13-15] and seed segmentation [16-18]. Objectness scoring ranks candidates and assigns a resulting 'objectness' score to each proposal, then a pre-defined score threshold is used to decide which proposals can be outputed. In superpixels merging methods, a given image is first over-segmented into small superpixels, after which proposals are obtained by merging each adjacent superpixel pair from bottom to up. All the seed segmentation methods start with multiple seed regions and generate a separate foreground-background segmentation for each seed [19]. Compared with objectness scoring and superpixels merging, seed segmentation can generate proposals with higher quality, but it's the most time-consuming [19].

Parametric min-cuts [4] is a typical seed segmentation method. It first enumerates foreground seeds (individual superpixels that are likely to be located inside objects) at different image locations to creat multiple seed graphs, then performs parametric min-cuts on these seed graphs. Since each foreground seed represents a small region of the object, and parametric min-cut is conducted on the seed graph to find the object segment that included in it. Obviously, using a large set of diverse foreground seeds can increase the odds of finding object proposals. However, on one hand, a large number of seeds means that many parametric min-cuts need to be solved, which slows down the algorithm. On the other hand, wrong foreground seed locations may lead to incorrect object proposals. So generating good-quality foreground seeds is a key factor to improve the performance of parametric min-cuts.

To solve this problem, this paper proposes a new framework for generating foreground seeds. First, saliency detection at multiple scales is used to generate a set of candidate foreground seeds. Second, based on the low and mid-level appearance features, these seeds are ranked by a pre-trained two-stage cascaded ranking classifier to predict how object-like a seed is.When used in parametric min-cuts, the foreground seeds generated by our method can achieve more accurate object location on the Pascal VOC 2011 segmentation dataset [20] with less proposals compared with existing state of the art methods.

The rest of this paper is organized as follows. Section 2 overviews the related work of object proposal and parametric min-cuts. Section 3 describes our proposed foreground seeds generation and its application in parametric min-cuts. Experiments and results are presented in Section 4 and we conclude this paper in Section 5.

2. Related Work

Many foreground seeds positioning strategies have been proposed for parametric min-cuts to get more accurate object proposals. Since similar object shapes are shared among different

categories, [21] proposes to generate foreground seeds by shape matching. To reduce the number of proposals, [22] presents to generate regions using the mid-level grouping cues of closure and symmetry, which are modeled by the color Gaussian Mixture Models. Based on [22], [23] proposes a novel structured learning framework to cast perceptual grouping and cue combination. [24] proposes a detection-based method which uses deformable part models [25] to detect part regions and extract foreground seeds from small part regions. [18] firstly over-segments a given image into small superpixels, then merges these hierarchical superpixels to generate foreground seeds. Recently, two seed placement methods are presented in [26]. One is a heuristic approach, and the other is a learning based approach that uses trained classifiers. And the learning strategy outperforms the heuristic method.

Inspired by the existing methods, we propose a new framework for generating effective foreground seeds. First of all, to encourage diversity among the seeds, we use saliency detection at multiple scales to generate large quantity of candidate foreground seeds. Then a pretrained two-stage cascaded ranking classifier, which is composed of partial ranking and complete ranking, is used to rank these candidate seeds.

Though the idea of ranking is also used in [17] and [26], but our work differs from them. [26] trains a linear ranking classifier for the placement of each seed, and only place one seed at every iteration. Different from it, we rank the foreground seeds in non-iterative manner to improve efficiency and use the two-stage cascaded framework to guarantee accuracy. Compared with [17], on one hand, we rank candidate foreground seeds while [17] ranks proposing regions. On the other hand, our ranking framework is novel. In [17], only one structural SVM is applied, but more ranking classifers are learned in our work, and they compose a two-stage cascaded framework to achieve our goal of discovering good-quality foreground seeds.

3. A two-stage cascaded foreground seeds generation

An effective foreground seeds generation strategy must possess two attributes. First, considerably more foreground seeds are required to guarantee high recall. To achieve this goal, we propose to produce candidate foreground seeds by saliency detection at multiple scales, which is different from the single scale scheme used in existing work. Second, though generating a large set of candidate foreground seeds makes it reliable, many seeds may be redundant or not good. These ineffective foreground seeds may lead to wrong proposal location and add computation burden for generating object proposals. So we should select good-quality foreground seeds. To achieve this goal, we propose a learning method based on a two-stage cascaded ranking classifier. In the first stage, a set of ranking SVM models for seeds with different sizes are used to rank all positive seeds above all negative ones. In the second stage, the resulting candidate foreground seeds with different sizes are jointly fed into a structural SVM to obtain a complete ranking result from which final foreground seeds are generated.

The pipeline of our method is illustrated in **Fig. 1**. To be simple for illustration, three sizes are used in saliency detection, and the results of different sizes are labeled in red, blue and green rectangles respectively. The number labeled in the top left corner of each rectangle is the ranking index, which implies the ranking result of the region in the rectangle. In the first stage, the candidate seeds are grouped based on their sizes, and the seeds in each group are ranked by themselves. In the second stage, the selected foreground seeds from all sizes are ranked together. The details of our method are described below.

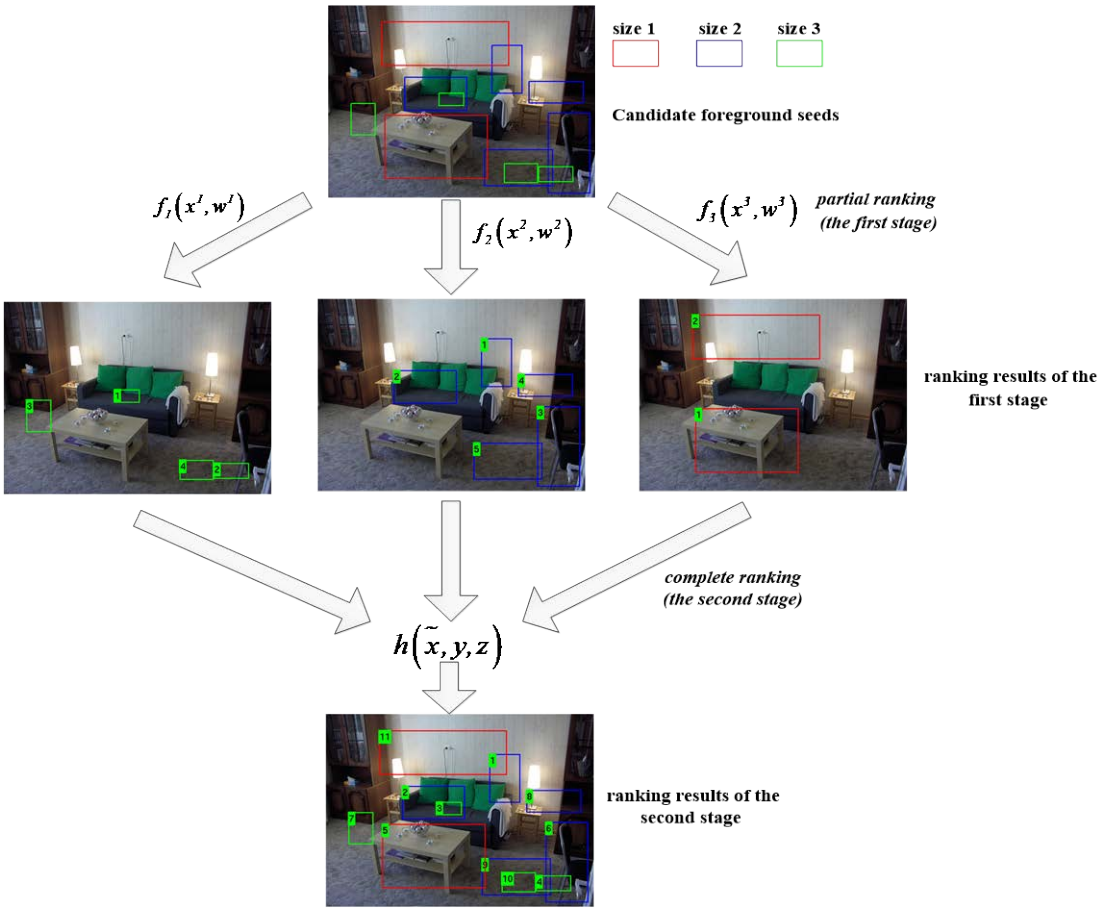


Fig. 1. The pipeline of our method

3.1 Generating Candidate Foreground Seeds at Multiple Scales

In most parametric min-cuts algorithms, foreground seeds are placed on densely sampled regular grids as shown in **Fig. 2**. Such generated foreground seeds are with a single scale which may bias towards some object with certain size. Different from it, we use saliency detection [2] at multiple scales to generate foreground seeds. On one hand, saliency detection can generate more accurate seeds. On the other hand, as shown in **Fig. 3**, multi-scale can find more salient foreground objects at different scales which can improve recall.

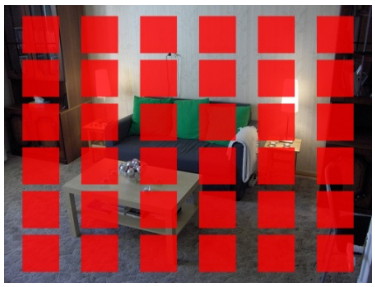


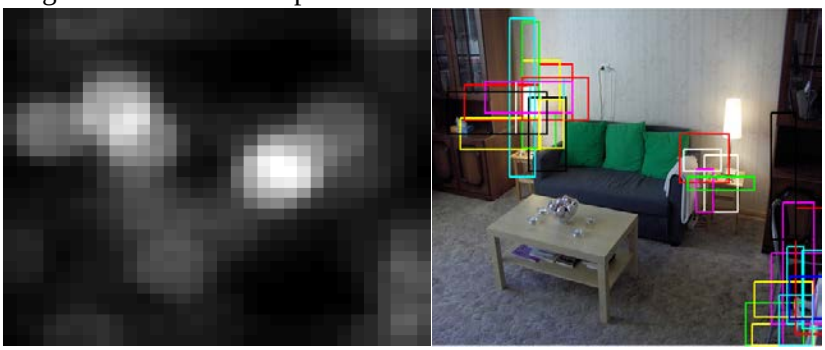
Fig. 2. The illustration of foreground seeds(labeled by red rectangle) placed on regular grids

The procedure is as follows. For a given image, saliency detection [27] is conducted for each scale s to obtain a saliency map I_s which defines the salience for every pixel p. And the saliency of a window x at scale s is defined as

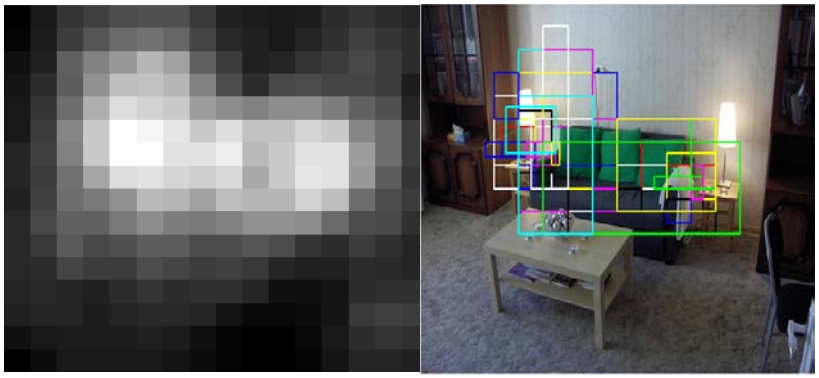
$$MS\left(x,\theta_{MS}^{s}\right)=\sum_{\{p\in x|I_s(p)\geq\theta_{MS}^{s}\}}I_s(p)\times\frac{|\{p\in x\,|\,I_s(p)\geq\theta_{MS}^{s}\}|}{|x|^{l}} \tag{1}$$

Where θ_{MS}^{s} is the scale-specific threshold [2]. The windows including higher density of salient pixels have higher saliency. And considering that saliency has a bias towards larger windows, the saliency of window x is normalized by window size. But different from [2], which use $|x|$ as the denominator of Equation (1), we use $|x|^{l}$ here, where l is a number bigger than 1 and it is set as 1.5 experientially in this paper. The reason is that in [2], saliency detection is used to locate the whole object, but it is used to generate foreground seeds which are parts of object with smaller size here.

Then the windows with higher saliency are chosen as foreground seeds which may cover the object. The left figure in **Fig. (3-a)** is the saliency map for a high scale and the foreground seeds generated based on it is shown in the right. Meanwhile, the left figure in **Fig. (3-b)** is the saliency map for a low scale and the foreground seeds generated based on it is shown in the right. As shown in **Fig. (3-a)**, when the scale is high, the region with small object, such as the stool beside the sofa, looks obvious in the saliency map. And by contrast, the big object region, such as sofa, looks obvious in the saliency map with low scale as shown in **Fig. (3-b)**. Since windows covering different objects in the image may score highest at different scales, generating foreground seeds at multiple scales is needed.



(3-a)



(3-b)

Fig. 3. Foreground seeds generated based on saliency detection at multiple scales. Image (3-a) is the saliency map for a high scale (the left one) and the foreground seeds generated based on it (the right one). Image (3-b) is the saliency map for a low scale and the foreground seeds generated based on it.

3.2 Foreground Seeds Selection based on a Two-stage Cascaded Ranking Classifier

After the step described in Section 3.1, we usually get thousands of foreground seeds for an image and many of these foreground seeds are redundant or not good. To obtain good object proposals, we need to sort these seeds and get a handful of object-like seeds without the prior knowledge about the object class. Since the appearances of different objects are extremely diverse, this is a challenging problem.

To solve this problem, we propose a two-stage cascaded ranking framework. In the first stage, partial ranking is used to filter out ineffective seeds. Since the initial candidate foreground seeds are generated by saliency detection at different scales and they have great differences in size, we group them based on their sizes and learn different ranking classifier for each size group. In the second stage, the left effective seeds from different size groups are ranked as a whole to get a complete ranking result, which are the final foreground seeds.

A. Partial Ranking

In the first stage, our goal is to filter out the ineffective seeds, which means to rank the effective seeds above the ineffective seeds. Since the ordering within the effective seeds is not concerned in this stage, it is called partial ranking. As the approach presented in [28] for text information retrieval, a set of ranking SVMs [29] are trained to accomplish this goal. The ranking process is illustrated in **Fig. 4**.

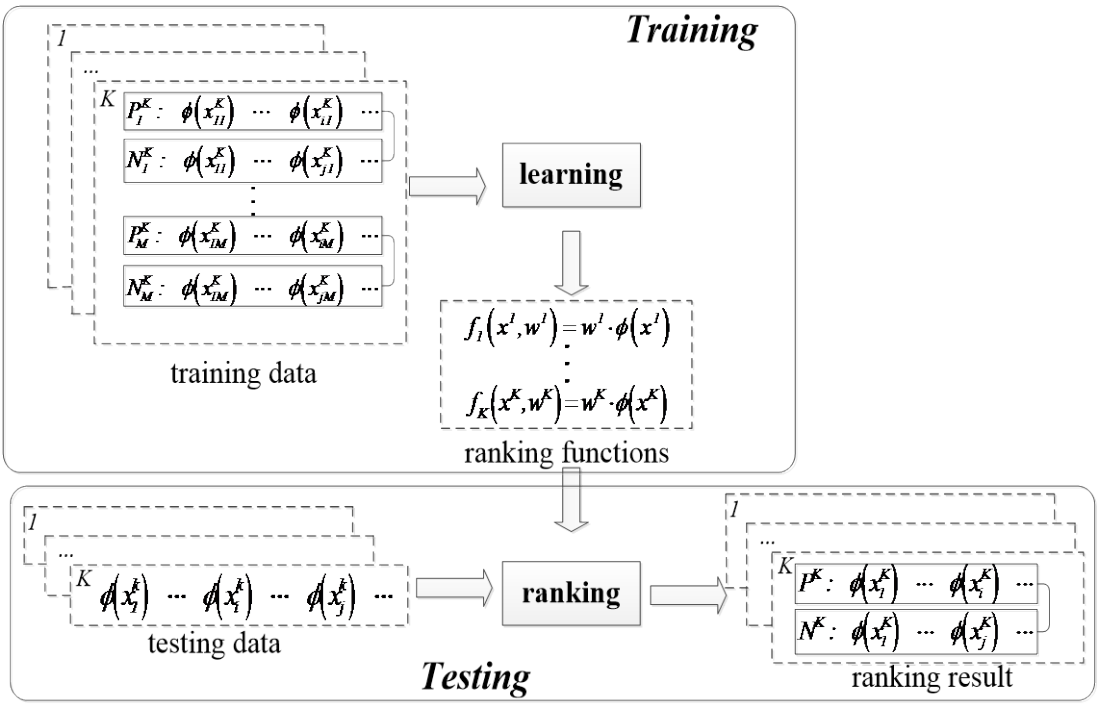


Fig. 4. Partial ranking in the first stage

The details are described as follows.

(a-1).Collect an image set $\{I_1,\ldots,I_M\}$ composed of M images, conduct multi-scale saliency detection as described in Section 3.1 on these images to obtain a set of windows $\{X_1,..,X_M\}$ with high saliency scores; group these windows according to their heights and widths into K groups. Suppose the selected window set of image I_m is X_m, then the windows in X_m can be grouped into $X_m=\{X_m^1,\ldots,X_m^K\}$, where $X_m^k=\{x_{1m}^k,\ldots,x_{im}^k,\ldots\}$ is the subset of candidate foreground seeds numbered as k in image m.

(a-2).For each seed set $X_m^k=\{x_{1m}^k,\ldots,x_{im}^k,\ldots\}$, analyze whether x_{im}^k has overlap with the object region which is labeled as G_m in image I_m. The seeds who have overlap with G_m are called effective seeds and the left seeds are ineffective seeds. The effective seeds compose positive seed set $P_m^k=\{x_{im}^k\in X_m^k \,|\, |x_{im}^k\cap G_m|>0\}$ and the ineffective seeds compose negative seed set $N_m^k=\{x_{jm}^k\in X_m^k \,|\, |x_{jm}^k\cap G_m|=0\}$.

(a-3).Extract the appearance feature $\phi(x_{im}^k)$ for each seed x_{im}^k. Based on the research in [30], each rectangle is firstly partitioned into 4×4 grids, then calculating the color, texture and shape features to get a rich representation for each grid [31]. The appearance features of seeds in P_m^k are labeled as $\phi(P_m^k)=\{\phi(x_{im}^k)\,|\,x_{im}^k\in P_m^k\}$ and used as positive samples. Accordingly, the appearance features of seeds in N_m^k are labeled as $\phi(N_m^k)=\{\phi(x_{im}^k)\,|\,x_{im}^k\in N_m^k\}$ and used as negative samples.

(a-4).Use each appearance feature pair $\phi(x^k)=\{\phi(x_{im}^k),\phi(x_{jm}^k)\}$ in $\phi(P_m^k)$ and $\phi(N_m^k)$ to training the ranking function for group k, which is modeled by Ranking SVM in the form of $f_k(x^k,w^k)=w^k\cdot\phi(x^k)$. And the parameter w^k should satisfy that for each training image, the positive samples score higher than the negative samples. The condition can be described below with the number of image used for training is M,

$$\forall x_{i1}^k\in P_1^k,\ x_{j1}^k\in N_1^k:\ f(x_{i1}^k,\ w^k)>f(x_{j1}^k,\ w^k)$$

…

$$\forall x_{iM}^k\in P_M^k,\ x_{jM}^k\in N_M^k:\quad f(x_{iM}^k,\ w^k)>f(x_{jM}^k,\ w^k) \tag{1}$$

The value of w^k can be obtained by solving the following function with the maximum marginal gain:

$$\begin{aligned}
&\min_{w}\frac{1}{2}\|w^k\|^2+C\sum_{m=1}^{M}\xi_m^k\\
&s.t.\ \forall x_{i1}^k\in P_1^k,\ x_{j1}^k\in N_1^k:\ w^k\cdot\phi(x_{i1}^k)-w^k\cdot\phi(x_{j1}^k)\geq1-\xi_1^k\\
&\qquad\qquad\ldots\\
&\forall x_{iM}^k\in P_M^k,\ x_{jM}^k\in N_M^k:\ w^k\cdot\phi(x_{iM}^k)-w^k\cdot\phi(x_{jM}^k)\geq1-\xi_M^k\\
&\forall\xi_m^k\geq0
\end{aligned} \tag{2}$$

where margin $\|w^k\|^2$ is the distance between the two closest projections within target rankings, ξ_m^k is slack variable, and C is a constant that allows tradeoff between the margin size and training error.

(a-5).Repeat the procedure from step (a-2) to step (a-4) until the Ranking SVMs for all groups are obtained.

After training, we can get the parameters of all the Ranking SVMs. Then in the partial ranking stage, for each testing image I_t , following steps are conducted:

(b-1).Conduct saliency detection at multiple scales for I_t as described in Section 3.1 to get a set of candidate foreground seeds .

(b-2).Group the above candidate foreground seeds to K groups, $X_{I_t}=\left\{X_{I_t}^1,\ldots,X_{I_t}^K\right\}$,as training step (a-1).

(b-3). As described in step (a-3), extract the appearance features [30] of the candidate foreground seeds in each group $X_{I_t}^k=\left\{x_{1I_t}^k,\ldots,x_{iI_t}^k,\ldots\right\}$ to get $\phi(X_{I_t}^k)=\left\{\phi(x_{1I_t}^k),\ldots,\phi(x_{iI_t}^k),\ldots\right\}$.

(b-4).Input each $\phi\left(x_{iI_t}^k\right)$ into the pre-trained Ranking SVM for *k*th group to decide whether it is an effective foreground seed or not. For each group, the topmost ranked β seeds are kept as effective seeds.

(b-5).Repeat step (b-3) and (b-4) to get the effective foreground seeds for all the groups, and finally we can get the effective seed set of all groups $\tilde{X}=\left\{\tilde{x}_1,..,\tilde{x}_{K\cdot\beta}\right\}$.

As we have no priori about the object region in the image, theoretically better results bias towards bigger β and bigger K . β is set as 50 empirically in this paper and the setting of K is discussed in Section 4.1.

The value of $K\cdot\beta$ is a few hundred, and after partial ranking, the left seeds still contain many ineffective foreground seeds which need to be selected further. Though the seeds in $\tilde{X}$ are the ranking results from multiple Ranking SVMs, we will rank them as a whole in the next Section.

B. Complete Ranking

The effective seed set $\tilde{X}=\left\{\tilde{x}_1,..,\tilde{x}_{K\cdot\beta}\right\}$ from partial ranking is used as the input to the second stage for complete ranking.

Since different from partial ranking which means to filter out the ineffective seeds from each group respectively, the ranking result in the second stage will determine the absolute ordering among effective foreground seeds from all groups, we call the ordering in this stage as complete ranking. We use the ranking function proposed in [17] to accomplish this goal:

$$h\left(\tilde{X},Y,z\right)=\sum_{i=1}^{K\cdot\beta}z_a\cdot\phi\left(\tilde{x}_i\right)-z_b\cdot\psi\left(y_i\right) \tag{3}$$

Equation (3) is a combination of appearance features $\phi(\tilde{x}_i)$ and overlap penalty term $\psi(y_i)$, where $y_i\in Y=\{1,2,...,K\cdot\beta\}$, $i=1,2,...,K\cdot\beta$ indicates the ranking index of seed i , ranging from 1 to the number of seeds, $K\cdot\beta$. $z=\{z_a,z_b\}$ is a weight vector. Our goal is to find the best ranking Y^* which can make the function $h\left(\tilde{X},Y,z\right)$ have the highest score:

$$Y^*=argmax_Y h\left(\tilde{X},\ Y,\ z\right) \tag{4}$$

$\psi(y_i)$ penalizes seeds with high overlap with top ranked seeds [17]. Concretely speaking, it is related with the sum of overlaps this seed has with all the seeds ranked above it. It's calculated in the form of Equation (5):

$$\psi(y_i)=\sum_{\{j|y_j<y_i\}}q\left(ov\left(\tilde{x}_i,\ \tilde{x}_j\right)\right) \tag{5}$$

As shown in Equation (6), the overlap in Equation (5) is defined as the area of two seeds' intersection divided by their union:

$$ov\left(\tilde{x}_i,\ \tilde{x}_j\right)=\frac{\left|\tilde{x}_i \cap \tilde{x}_j\right|}{\left|\tilde{x}_i \cup \tilde{x}_j\right|} \tag{6}$$

Since the intensity of the penalty depends on the amount of overlap, the seed with more overlap should be suppressed more than the seed with less overlap. To measure the overlap, in Equation (5), we use $q(\cdot)$ to equally quantize the overlaps into 10 bins.

Since the ranking function $h\left(\tilde{X},Y,z\right)$ can be regarded as a projection from $\tilde{X}$ to Y in the form of $h\left(\tilde{X},Y,z\right):\tilde{X}\to Y$, it allows us to take advantage of structured learning to obtain the solution of weight vector z. According to the Structural SVM proposed in [32], we construct the following objective function:

$$\begin{aligned}
&\min_z \frac{1}{2}\| z \|^2 + C\sum_{m=1}^{M}\xi_m \\
s.t.\ &\tilde{Y}_1 = argmax_{Y_1} h\left(\tilde{X}_1,\ Y_1,\ z\right),\ \forall Y_1 \neq \tilde{Y}_1 : h\left(\tilde{X}_1,\tilde{Y}_1,z\right) - h\left(\tilde{X}_1,Y_1,z\right) \geq \mathcal{L}\left(G_1,Y_1\right) - \xi_1 \\
&\cdots \\
&\tilde{Y}_M = argmax_{Y_M} h\left(\tilde{X}_M,Y_M,z\right),\ \forall Y_M \neq \tilde{Y}_M : h\left(\tilde{X}_M,\tilde{Y}_M,z\right) - h\left(\tilde{X}_M,Y_M,z\right) \geq \mathcal{L}\left(G_M,Y_M\right) - \xi_M \\
&\forall \xi_m^k \geq 0
\end{aligned} \tag{7}$$

$\tilde{Y}_m$ in Equation (7) means the best ordering which can make the ranking function have highest score with the candidate foreground seeds from image m, while Y_m means general ordering. $\mathcal{L}$ is the loss term which indicates the margin between the ranking scores obtained by the best ordering $\tilde{Y}_m$ and other ordering Y_m. Since in the best ordering, the seed has overlaps with more different object regions should be ranked in front, $\mathcal{L}$ is defined in the form of Equation (8):

$$\mathcal{L}\left(G_m\ ,\ Y_m\right)=\frac{1}{\left|G_m\right|\cdot\left|T\right|}\sum_{t\in T}\sum_{g\in G_m}\alpha(t)\cdot\sum_{\{i|ov(\tilde{x}_i,g)\in t\}} y_i \tag{8}$$

Similar to the definition of penalty term, the overlap between foreground seed $\tilde{x}_i$ and object region $g\in G_m$, which is labeled as $ov(\tilde{x}_i,g)$, is also quantized into 10 bins which are equally partitioned between 0 and 1. Each bin is indexed by t, and the upper limit of t is 10 which is indicated by T in Equation (8). Then we will sum the indexes of y_i s in each bin to construct the relationship between the overlap and the ranking index. Considering that different bin has different impact on the loss term, for example, the foreground seed having more overlap (in the bin with bigger index) with object region but ranked back has stronger impact on the loss term. We use $\alpha(t)$ to weight the sum of ranking indexes in each bin t. $\alpha(t)$ is defined as $\alpha(t)=\exp\left(index_t/\sigma\right)$, where $index_t$ is the index of bin t and $\sigma=0.1$.

Algorithm 1: Structural learning based complete ranking in the second stage

1:Input: $\{\tilde{X}_1,\ldots,\tilde{X}_M\}$ (The foreground seeds of the M images selected in the first stage), $\{G_1,\ldots,G_M\}$ (the ground truth of the M images), C (penalty factor), ε (error accuracy)

2:Output： z (weight vector)

3:Initialization: ξ_m , z , $\mathcal{W}_m=\varnothing$ (workspace set)

4:repeat

5:for $m=1,\ldots,M$, do

6:Exhaustively search the best odering $\tilde{Y}_m=arg\max_{Y_m} h\left(\tilde{X}_m,Y_m,z\right)$

7: Calculate the most violate constraint $Y_m=\arg\max_{Y_m}\mathcal{L}\left(G_m,Y_m\right)+h\left(\tilde{X}_m,Y_m,z\right)-h\left(\tilde{X}_m,\tilde{Y}_m,z\right)$

8:if $\mathcal{L}\left(G_m,Y_m\right)+h\left(\tilde{X}_m,Y_m,z\right)-h\left(\tilde{X}_m,\tilde{Y}_m,z\right)>\xi_m+\varepsilon$, then

9:update workspace set $\mathcal{W}_m=\mathcal{W}_m\bigcup\{Y_m\}$

10:update weight vector and slack variable based on quadratic programming

$$\left(z,\boldsymbol{\xi}\right)=\arg\min_{z,\xi}\frac{1}{2}z^2+C\sum_{m=1}^{M}\xi_m$$

$$s.t.\ \forall\hat{Y}_1\in\mathcal{W}_1:h\left(\tilde{X}_1,\tilde{Y}_1,z\right)-h\left(\tilde{X}_1,\hat{Y}_1,z\right)\geq\mathcal{L}\left(G_1,\hat{Y}_1\right)-\xi_1$$

$$\ldots$$

$$\forall\hat{Y}_M\in\mathcal{W}_M:h\left(\tilde{X}_M,\tilde{Y}_M,z\right)-h\left(\tilde{X}_M,\hat{Y}_M,z\right)\geq\mathcal{L}\left(G_M,\hat{Y}_M\right)-\xi_M$$

11： end if

12： end for

13： until no z has changed during iteration

Since the best ordering $\tilde{Y}$ is unknown, we learn this latent structured model by iteratively finding the rank with the highest score for each image and solving the structured learning problem. The learning process is described as Algorithm 1. As in the sixth row of Algorithm 1, we firstly derive the ranking result with the highest score in the condition of current weights, and output it as the best ranking. Then use a cutting-plane [33] based optimization to learn the Structural SVM from Row 7 to Row 10. First, find the ranking which has the most violate constraint with the current weigths (Row 7). Second, update the ranking with the most violate constraint (Row 8 and Row 9). Finally, update z with the new constraint (Row 10). The above steps are repeated until the change of z is small.

3.3 The Application of Our Method in Parametric Min-cuts

In this section, we describe how to apply the foreground seeds generated by our method to parametric min-cuts, an object proposal method. Here, we integrate our method in the framework of a state-of-the-art parametric min-cuts algorithm, Rigor [34], and the main pipeline is illustrated in **Fig. 5**.

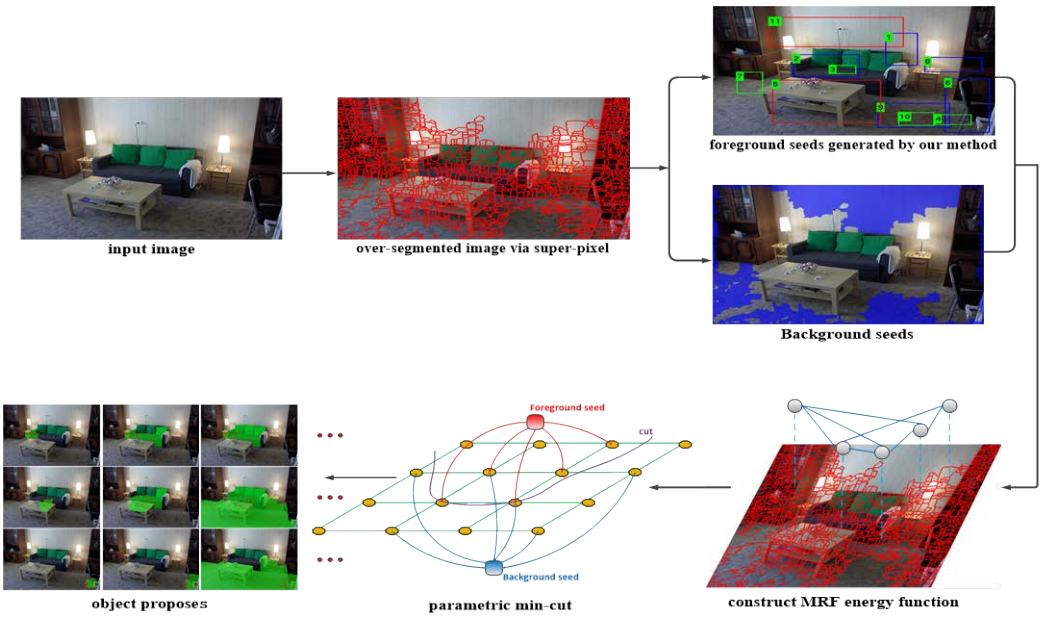


Fig. 5. In the framework of Rigor, Our method is used in the Row 1, Column 3 to generate foreground seeds

In Rigor, first of all, define a graph on superpixels V where the similarity between neighboring superpixel pair (x_u, x_v) is measured and encoded as edges E to get a weighted graph $G=(V,E)$. Then use the foreground and background priories provided by foreground seeds and background seeds respectively to construct a binary energy function with a scaling term as follows:

$$E^{\lambda}(X,Y,s)=w_1\sum_{u\in V}\underbrace{\phi_1(x_u,y_u,s)}_{unary\ \ term}+w_2\sum_{(u,v)\in E}\underbrace{\phi_2(x_u,x_v,y_u,y_v)}_{binary\ \ term}+\lambda\sum_{u\in V}\underbrace{\phi_2(x_u,y_u)}_{scaling\ \ term} \tag{9}$$

The unary term $\phi_1(x_u,y_u,s)$ combines single or multiple appearance cues to describe the similarity between the nodes in the graph model. And $Y=\{y_v\}_{v=1}^{|V|}$ is the set of binary lables $\{0,1\}$ for each superpixel v.

As the scaling term in Equation (9) has a linear variable λ, we can minimize the energy function based on parametric min-cuts [35] to decide the label of each node, which indicates the node belongs to foreground or background. Finally, with the constraint of each foreground seed, we can get a set of segmentation results by gradually changing scales as object proposals.

Since we improve the foreground seeds generation in Rigor algorithm by a two-stage cascaded ranking classifier, to distinguish from the original Rigor algorithm, we call this parametric min-cuts method as Rank2 algorithm below.

4. Experiments

We perform experiments on Pascal VOC [20] dataset, which consists of 20 object categories. The 5011 images in the VOC 2007 test set are used as training data to learning the two-stage cascaded ranking classifier, and the 1449 images in the VOC 2011 segmentation task are used for test. All the experiments are done on a 3.3GHz CPU(Intel(R) Xeon E5-2690) with 64G RAM. To speed up the computation of appearance feature, the gPb contour detection method used in [36] is replaced by a fast contour detection method [37].

4.1 Decide the Number of Ranking SVMs used in the Partial Ranking

As described in Section 3.2.A, to guarantee the recall, we group the candidate foreground seeds obtained by saliency detection according to their sizes, and train ranking SVM for each group. In this experiment, we will test how to choose the number of groups for partial ranking.

First, we specify six sets of boundaries to group the candidate seeds into multiple groups: $\{[16,128]\}$, $\{[16,72],[72,128]\}$, $\{[16,54],[54,90],[90,128]\}$, $\{[16,44],[44,72],[72,100],[100,128]\}$, $\{[16,38],[38,60],[60,82],[82,104],[104,128]\}$, $\{[16,24],[24,32],[32,48],[48,64],[64,96],[96,128]\}$.

Using the grouping strategies in different sets, we can divide the candidate foreground seeds obtained by saliency detection into groups with different group numbers. For example, if we use $\{[16,128]\}$ to partition the seeds, we treat the seeds as a whole with a single group, and we only need to train 1 Ranking SVM in the first stage. And if we use $\{[16,24],[24,32],[32,48],[48,64],[64,96],[96,128]\}$ to group the seeds according to their heights and widths respectively, we can group them into $6\times 6=36$ different groups, and we need to train 36 Ranking SVMs. In a word, for the above 6 grouping strategies, we respectively need to train 1, 4, 9, 16, 25, 36 Ranking SVMs in the first stage. Here we use the original Rigor algorithm as baseline to test our Rank2 algorithms with the above 6 different grouping strategies.

According to [26], we use ABO (average best overlap)—#Seed (seed number) curve to measure the validity of foreground seeds. This curve reflects the changing trend of ABO with an increasing number of seeds. The definition of ABO can be found in [3], which is the average value of the overlap rate between each object region and the closest object proposal. Obviously, bigger ABO implies better performance.

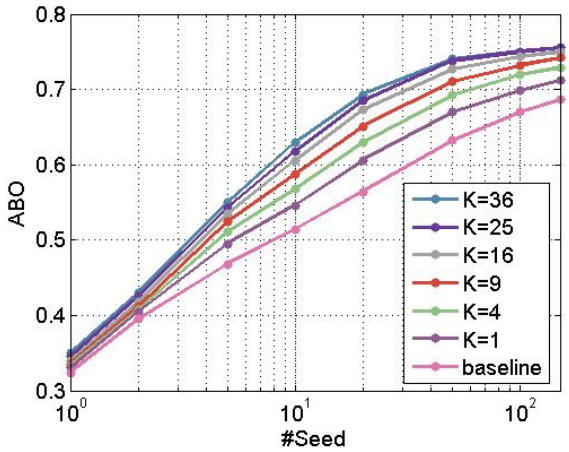


Fig. 6. ABO—#Seed curves with different grouping strategies

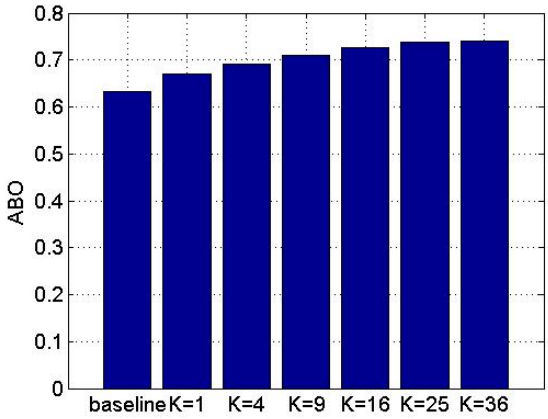


Fig. 7. ABO values with different group numbers while the seed number is 50

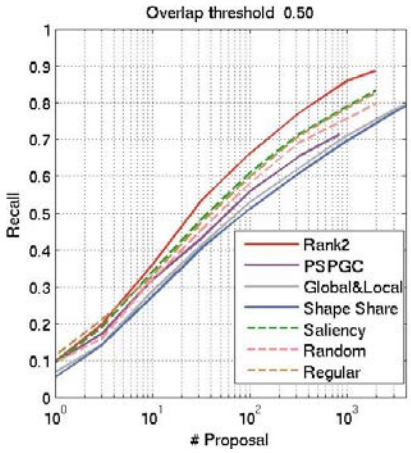
As shown in **Fig. 6** that ABO increases as the number of seeds increases for all the methods. And the curve with bigger K has better performance. The foreground seeds generation by evenly sampling in original Rigor is labelled as "baseline" in **Fig. 6**, and the results of our method with all the *K*s are better than it. Such improvement hinges on two contributions in our method. On one hand, saliency detection at multiple scales can generate huge various candidate foreground seeds. On the other hand, the ranking SVMs we proposed can select effective foreground seeds.

Moreover, with the increasement of K, the curve distance between neighboring Ks becomes small. As it's shown that the curve with $K = 36$ and the curve with $K = 25$ is very closing in the figure. We learn from that it has no gain to set K bigger than 36, so the number of groups is set to 36 in the following experiments. As it's shown in **Fig. 7**, when $K = 36$ and the number of seeds is set to 50, the ABO can achieve 0.742.

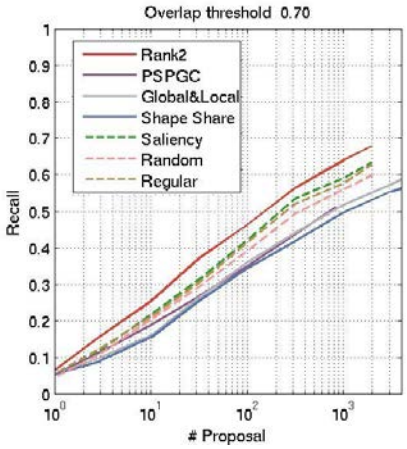
4.2 Comparison to Other Foreground Seeds Generation Methods

In this section, our Rank2 algorithm and some other representative foreground seeds generation methods are respectively used in parametric min-cuts algorithm for comparison. The compared methods include PSPGC [24], Global&Local [18], Shape Share [21] and three methods based on different grid sampling strategies: regular sampling (Regular), random sampling (Random) and saliency-weighted random sampling (Saliency).

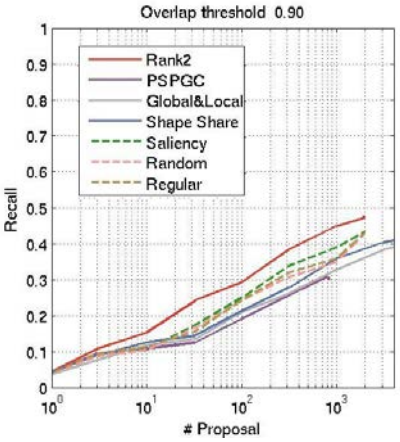
Since the number of foreground seeds generated from some of the above generation methods cannot be controlled, it can't be compared directly. Then we use the number of object proposals obtained by the parametric min-cuts algorithm to indicate it. It's reasonable since the number of foreground seeds decides the number of object proposes in Parametric Min-cuts algorithm. As in [38], we use Recall-#Proposal curve to measure the performance of each method, which reflects the changing trend of recall with the increasing number of object proposals. Since the calculation of recall is related to overlap rate, which implies location accuracy, here we set it to three different thresholds, $ov = 0.5, 0.7, 0.9$, and report the recalls corresponding to them respectively. The results are shown in **Fig. 8**.



(8-a)



(8-b)



(8-c)

Fig. 8. The Recall-#Proposal curves of different foreground seeds generation methods at different overlap rate thresholds

From **Fig. (8-a)** to **(8-c)**, as the overlap rate threshold becomes stringent, the performances of all the methods gradually decline. And our Rank2 algorithm has the best recalls at both overlap rate thresholds $ov = 0.70$ and $ov = 0.9$, which means that Rank2 has better performance in the condition of higher location accuracy. But as shown in 8(a) when overlap rate threshold is set as $ov = 0.5$, Regular has higher recall than Rank2 when the number of object proposals is less than 10. The reason is that with a low seed budget, the seeds generated by Regular are placed more separately, so it can discover more object proposals though the location accuracy may be low.

Moreover, Saliency and Regular, the two improved versions of Rigor perform similarly and have better performances than other methods. The reason is that Rigor is a state-of-the-art objection proposal algorithm, which can provide complementary information for locating object by using multiple unary terms at the same time. But since placing initial candidate foreground seeds at random position can't guarantee they can hit the object region but not the background, Random has worst location quality among these three methods though it is also a Rigor-based method.

As for the other three methods, PSPGC is able to discover more object proposals with less number of seeds since it can accurately pre-locate object region by DPM detection. Global&Local and Shape Share have respective advantages. Shape Share has higher recall while $ov = 0.9$ since shape matching can help to generate object proposes with higher location accuracy. On the contrary, shape matching may miss many anomaly shapes, so Global&Local, which combines features of color and texture outperforms Shape Share at $ov = 0.5$ and $ov = 0.70$.

4.3 Comparison to Other Object Proposals

In this section, we evaluate the accuracy of object proposals produced by our Rank2 algorithm. **Table 1** compares the accuracy of Rank2 to five state-of-the-art object proposal methods, SS [3], MCG [38], CPMC [4],Cat-Ind OP [17] and Rigor [34]. Among these five methods, SS and MCG are based on the framework of hierarchical superpixel merging while the other three are based on parametric min-cuts. The performance of each algorithm is evaluated by #Proposal, ABO, Covering [26] and Running time.

Table 1. Accuracy and running time for our method and the compared methods

Method	#Proposal	ABO	Covering	Time(s)
Parametric Min-cuts based methods				
CPMC[4]	673.5	0.712	0.826	274.4
Cat-Ind OP[12]	1641.3	0.705	0.813	125.8
Rigor[13](64)	1764.9	0.737	0.832	8.3
Rank2(25)	821.3	0.714	0.812	6.4
Rank2(64)	1773.2	0.751	0.842	8.8
Superpixel merging based methods				
SS [3]	8015.7	0.752	0.840	11.2
MCG [11]	1285.6	0.754	0.850	25.5

*The numbers labeled in the square brackets are the numbers of foreground seeds. The methods without this term use the default settings as the original literatures. The number of partial ordering classifiers used in the first stage is $K = 36$.

The number of foreground seeds used in Rank2 is highly related to its performance. More foreground seeds can get better ABO, but meanwhile generate more proposals and need more running time. To analyze such influence, we test with different seed numbers. 25 topmost ranked foreground seeds and 64 topmost ranked foreground seeds are respectively used in Rank2(25) and Rank2(64). Rank2(25) biases towards computational cost while Rank2(64) biases towards accuracy. As shown in Table 1, Rank2(25) accomplishes similar performance as Rigor in ABO and Covering with less seeds, and the running time is reduced by 7.7%. Rank2(64) uses the same number of seeds as Rigor, and its running time is similar to Rigor, but its ABO is increased by 1.9% and its Covering is increased by 1.3%. In a word, Rank2(25) outperforms Rigor in running time, and Rank2(64) outperforms Rigor in accuracy. Compared with the other two parametric min-cuts based methods, CPMC and Cat-Ind OP, Rank2(25) and Rank2(64) are two orders of magnitude faster.

Moreover, a hierarchical superpixel merging based method, MCG outperforms Rank2(25) and Rank2(64) in ABO and Covering by combining state-of-the-art hierarchical segmentation and multi-scale information, but its running time is more than 3 times to Rank2(25) and Rank2(64). Another hierarchical superpixel merging based method, SS has similar performance as Rank2(25) and Rank2(64) in accuracy and running time, but it generates more than 4 times the number of proposals than all the other methods. More proposals will affect the efficiency of following process, such as segmentation and detection.

In a word, synthetically considering these four metrics, our Rank2 outperforms the other methods.

4.4 Qualitative Analysis

Some qualitative proposal results from different object proposal methods with the highest overlap rates are shown in **Fig. 9**. The regions marked by red and bottle green in the second column are the ground truth. And the regions marked by grass green and purple in the last 7 columns are proposals generated by different methods, with a number displayed on each proposal marking its overlap rate.

For each original image, the proposals with the top 3 overlap rates are labelled by the red rectangle. To rank these proposing results, if an image has only one proposal, we use its overlap rate for ranking directly. And if an image has two proposals, we sum their overlap rates for ranking. As it shown in **Fig. 9**, Rank2(64) outperforms most methods on these 5 images. And it must be noted that Rank2(64) has the best performances on the first three images. Rank2(25) also has good performance on three images, and it fails to achieve top 3 on the third and fifth images whose objects are in big size and have similar color with the background.

In summary, the foreground seeds generated by our two-stage cascaded ranking are effective, but since the low and mid-level appearance features used in ranking are based on color, texture and shape, they are sensitive to complex and clutter background. And using more seeds is an effective way to improve the performance.

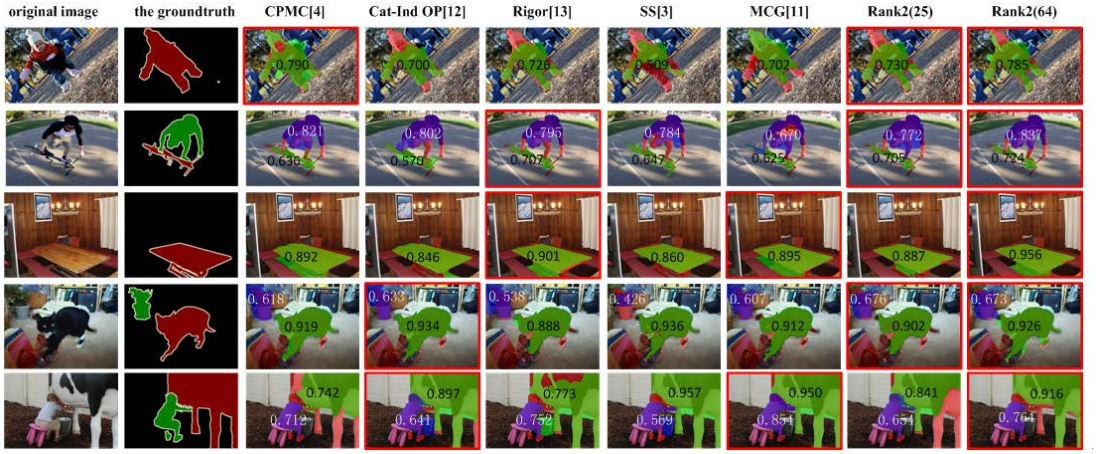


Fig. 9. The comparison of the best proposals on some images generated by different object proposal algorithms. The results with the top 3 overlap rates are labelled by the red rectangle.

5. Conclusion

Foreground seeds generation is a fundamental work in the parametric min-cuts based object proposal algorithms. To get good-quality foreground seeds, we train a two-stage cascaded ranking classifier to filter the candidate foreground seeds based on their appearance features. Considering that foreground seeds have multiple sizes, in the first ranking stage, we use many ranking SVMs trained with foreground seeds in different sizes to rank candidate foreground seeds according to their sizes. Then the left candidate foreground seeds are ranked by a structural SVM to finally select the foreground seeds. Experimental results show that the proposed method can generate good-quality foreground seeds in an efficient way, and these seeds can be successfully used in parametric min-cuts framework.

References

[1] J. Hosang, R. Benenson, P. Dollár and B. Schiele, "What makes for effective detection proposals?" *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol.38, no.4, pp. 6644-6665, August, 2016. Article (CrossRef Link).

[2] B. Alexe, T. Deselaers, V. Ferrari, "What is an object?" in *Proc. of 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 73-80, June 13-18, 2010. Article (CrossRef Link).

[3] KEAVD Sande, JRR Uijlings, T. Gevers and AWM Smeulders, "Segmentation as selective search for object recognition," in *Proc. of the 2011 IEEE Conference on Computer Vision (ICCV)*, pp. 1879-1886, January 12-16, 2011. Article (CrossRef Link).

[4] J. Carreira, C. Sminchisescu, " Constrained parametric min-cuts for automatic object segmentation," in *Proc. of the 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3241-3248, June 13-18, 2010. Article (CrossRef Link).

[5] R. Girshick, J. Donahue, T. Darrell and J. Malik, " Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 580-587, June 23-28,2014. Article (CrossRef Link).

[6] L. Gomez and D. Karatzas, "Object Proposals for Text Extraction in the Wild," in *Proc. of 2015 International Conference on Document Analysis and Recognition*, pp.1786-1812, August 23-26,2015. Article (CrossRef Link).

[7] A. Milan, L. Leal-Taixé, K. Schindler, et al, "Joint tracking and segmentation of multiple targets,"

in *Proc. of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5397-5406, June 7-12,2015. Article (CrossRef Link).

[8] J. Carreira, R. Caseiro, J. Batista, et al, "Semantic segmentation with second-order pooling," in *Proc. of 2012 European Conference on Computer Vision (ECCV)*, pp. 430-443, October 7-13, 2012. Article (CrossRef Link).

[9] A. Borji, D. N Sihite, L. Itti, "Salient object detection: A Benchmark,"*IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706-5722, October, 2015. Article (CrossRef Link).

[10] B. Alexe, T. Deselaers, V. Ferrari, "Measuring the objectness of image windows," *PAMI*, vol. 34, no.11, pp. 2189-2202, November, 2012. Article (CrossRef Link).

[11] E. Rahtu , J. Kannala, M. Blaschko, "Learning a category independent object detection cascade," in *Proc. of the 2011 International Conference on Computer Vision (ICCV)* , pp. 1052-1059, November 6-13, 2011. Article (CrossRef Link).

[12] M.M. Cheng, Z. Zhang, W.Y. Lin and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3286-3293, June 23-28,2014. Article (CrossRef Link).

[13] S. Manen, M. Guillaumin, L.V. Gool, "Prime object proposals with randomized prims algorithm," in *Proc. of the 2013 International Conference on Computer Vision (ICCV)*, pp. 2536-2543, December 1-8,2013. Article (CrossRef Link).

[14] X. Wang, M. Yang, S. Zhu and Y. Lin, "Regionlets for generic object detection," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol.37, no.10, pp.17-24, January, 2015. Article (CrossRef Link).

[15] P.F. Felzenszwalb, D.P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol.59, no.2, pp.167-181, September, 2004. Article (CrossRef Link).

[16] J. Carreira, C. Sminchisescu, "Cpmc: Automatic object segmentation using constrained parametric min-cuts," *PAMI*, vol.34, no.7, pp. 1312-1328, December, 2011. Article (CrossRef Link).

[17] I. Endres, D. Hoiem, "Category-independent object proposals with diverse ranking," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.36, no.2, pp. 222-234, June, 2014. Article (CrossRef Link).

[18] P. Rantalankila, J. Kannala, E. Rahtu, "Generating object segmentation proposals using global and local search," in *Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2417-2424, June 23-28, 2014. Article (CrossRef Link).

[19] C. L. Zitnick, P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. of 2014 European Conference on Computer Vision (ECCV)*, pp. 391-405, September 6-12, 2014. Article (CrossRef Link).

[20] M. Everingham, S. M. A. Eslami, L. Van Gool, et al, "The pascal visual object classes challenge: A retrospective,"*IJCV*, vol.111, no.1, pp. 98-136, January, 2015. Article (CrossRef Link).

[21] J. Kim, K. Grauman, "Shape sharing for object segmentation," in *Proc. of 2012 European Conference on Computer Vision (ECCV)*, pp. 444-458, October 7-13,2012. Article (CrossRef Link).

[22] T. Lee, S. Fidler, S. Dickinson, "Multi-cue mid-level grouping," in *Proc. of 2014 Asian Conference on Computer Vision (ACC*V), pp. 376-390, April 16-20, 2014. Article (CrossRef Link).

[23] T. Lee, S. Fidler, S. Dickinson, "Learning to Combine Mid-level Cues for Object Proposal Generation," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1680-168, February 18, 2015. Article (CrossRef Link).

[24] B. Singh, X. Han, Z. Wu, et al, "PSPGC: Part-Based Seeds for Parametric Graph-Cuts," in *Proc. of 2014 Asican Conference on Computer Vision (ACCV)*, pp. 360-375, April 16, 2014. Article (CrossRef Link).

[25] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, et al, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.32, no.9, pp. 1627-1645, September, 2009. Article (CrossRef Link).

[26] P. Krähenbühl, V. Koltun, "Geodesic object proposals," in *Proc. of 2014 European Conference on*

Computer Vision (ECCV), pp. 725-739, September 5-8, 2014. Article (CrossRef Link).
[27] X. Hou, L. Zhang, "Saliency detection: A spectral residual approach," in *Proc. of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, June 17-22, 2007. Article (CrossRef Link).
[28] T. Y. Liu, *Learning to rank for information retrieval*, 2nd Edition, Springer Berlin Heidelberg, 2011. Article (CrossRef Link).
[29] T. Joachims, "Optimizing search engines using clickthrough data," in *Proc. of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 133-142, August 12-15,2002. Article (CrossRef Link).
[30] C. Gu, J. J. Lim, P. Arbeláez, et al, "Recognition using regions," in *Proc. of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1030-1037, June 20-25, 2009. Article (CrossRef Link).
[31] N. Dalal, B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. of the 2005 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 886-893, June 20-25, 2005. Article (CrossRef Link).
[32] I. Tsochantaridis, T. Joachims, T. Hofmann, et al, "Large margin methods for structured and interdependent output variables," *Journal of Machine Learning Research*. vol.6, no.2, pp. 1453-1484, January, 2005. Article (CrossRef Link).
[33] T. Joachims, T. Finley, C. N. J. Yu, "Cutting-plane training of structural SVMs," *Machine Learning*, vol.77, no.1, pp. 27-59, October, 2009. Article (CrossRef Link).
[34] A. Humayun, F. Li, J. M. Rehg, "RIGOR: Reusing inference in graph cuts for generating object regions," in *Proc. of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 336-343, June 23-28, 2014. Article (CrossRef Link).
[35] V. Kolmogorov, Y. Boykov, C. Rother, "Applications of parametric maxflow in computer vision," in *Proc. of the 2007 International Conference on Computer Vision (ICCV)* , pp. 1-8, Octorber 14-21, 2007. Article (CrossRef Link).
[36] M. Maire, P. Arbeláez, C. Fowlkes, et al, "Using contours to detect and localize junctions in natural images," in *Proc. of the 2008 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1-8, June 23-28,2008. Article (CrossRef Link).
[37] P. Dollár, C. L. Zitnick, "Fast edge detection using structured forests," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol.37, no.8, pp.1558-1570, December, 2015. Article (CrossRef Link).
[38] J. Pont-Tuset, P. Arbelaez, J. T. Barron, et al, "Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, pp. 1-1, March, 2016. Article (CrossRef Link).

Shaomei Li received the B.S. degree from Information Engineering University, Zhengzhou, China, in 2004. Received the M.S. and Ph.D degrees in communication and Information System from the National Digital Switching System Engineering and Technological Research and Development Center, Zhengzhou, China, in 2007 and 2011 respectively. She is currently a Lecturer with National Digital Switching System Engineering and Technological Research and Development Center , Zhengzhou, China. Her current research interests include pattern recognition and computer vision, especially human action analysis, target tracking and image understanding

Junguang Zhu received the B.S. degree from Nan Jing University, Nanjing, China, in 2013, and is currently pursuing the M.S. Degree with National Digital Switching System Engineering and Technological Research and Development Center, Zhengzhou, China. His current research interests include object detection, computer vision and pattern recognition.

Chao Gao received his MS degree in system engineering from National University of Defense Technology (China), in 2008. He is currently a lecturer with the National Digital Switching System Engineering and Technological Research and Development Center, Zhengzhou, China. His research interests include multi-class object detection, image categorization and semantic segmentation.

Chunwei Li received the B.S. degree from Zhe Jiang University, Zhejiang, China, in 2013, and is currently pursuing the M.S. Degree with National Digital Switching System Engineering and Technological Research and Development Center, Zhengzhou, China. His current research interests include object detection, computer vision and pattern recognition.