Learning to Segment Object Candidates via Recursive Neural Networks

Tianshui Chen, Liang Lin, Xian Wu, Nong Xiao, and Xiaonan Luo

***Abstract*— To avoid the exhaustive search over locations and scales, current state-of-the-art object detection systems usually involve a crucial component generating a batch of candidate object proposals from images. In this paper, we present a simple yet effective approach for segmenting object proposals via a deep architecture of recursive neural networks (ReNNs), which hierarchically groups regions for detecting object candidates over scales. Unlike traditional methods that mainly adopt fixed similarity measures for merging regions or finding object proposals, our approach adaptively learns the region merging similarity and the objectness measure during the process of hierarchical region grouping. Specifically, guided by a structured loss, the ReNN model jointly optimizes the cross-region similarity metric with the region merging process as well as the objectness prediction. During inference of the object proposal generation, we introduce randomness into the greedy search to cope with the ambiguity of grouping regions. Extensive experiments on standard benchmarks, e.g., PASCAL VOC and ImageNet, suggest that our approach is capable of producing object proposals with high recall while well preserving the object boundaries and outperforms other existing methods in both accuracy and efficiency.**

***Index Terms*— Object proposal generation, object segmentation, region grouping, recursive neural networks, deep learning.**

I. Introduction

OBJECT proposal generation, which aims to identify a small set of region proposals where objects are likely to occur, benefits a wide range of applications such as generic object detection [1], [2], object recognition [3]–[5] and object discovery [6], [7]. Usually, a good object proposal method is desired to be capable of not only recalling all existing objects over scales and locations but also preserving their boundaries, for example in Figure 1.

Manuscript received October 11, 2016; revised July 2, 2017 and January 6, 2018; accepted July 12, 2018. Date of publication July 23, 2018; date of current version September 4, 2018. This work was supported in part by the National Natural Science Foundation of China under Grant 61772149, in part by the Program for Guangdong Introducing Innovative and Entrepreneurial Teams under Grant NO. 2016ZT06D211, in part by the National Natural Science Foundation of China under Grant 61622214, in part by the Science and Technology Planning Project of Guangdong Province under Grant 2017A020208041, and in part by the Guangdong Natural Science Foundation Project for Research Teams under Grant 2017A030312006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Weisi Lin. *(Corresponding author: Xiaonan Luo.)*

T. Chen, L. Lin, X. Wu, and N. Xiao are with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, China and also with the Engineering Research Center for Advanced Computing Engineering Software, Ministry of Education, Beijing 100816, China (e-mail: tianshuichen@gmail.com; linliang@ieee.org; sysuwuxian@gmail.com; xiaon6@mail.sysu.edu.cn).

X. Luo is with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China (e-mail: luoxn@guet.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2018.2859025

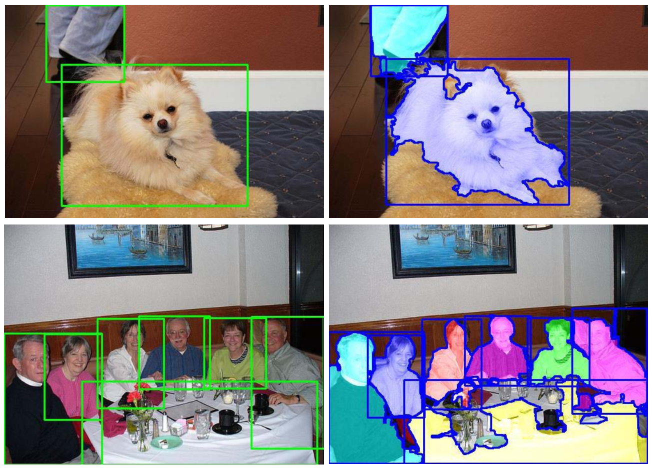


Fig. 1. Some object proposals (indicated by the blue boxes) generated by our approach. Our results match well with the ground-truth (indicated by the green boxes), and also preserve the object boundaries (indicated by the blue silhouettes inside the boxes).

The challenges of object proposal lie in the presence of severe occlusion, variations in object shapes, and the lack of category information. Most of the current methods [8]–[10] tackle these difficulties through bottom-up region grouping or segmentation. Those methods mainly involve two crucial components, i.e., cross-region similarity metric and region merging algorithm. The similarity metric is utilized to measure whether two adjacent regions should be merged, and the merging algorithm performs the inference process that groups pairs of regions into super-regions and finally generates object proposals. Thus, object proposal generation methods based on region grouping basically follow the pipeline: they assign a higher similarity score to the adjacent regions if it is confident that the regions belong to the same class, and recursively merge the adjacent regions with highest score. Despite of acknowledged successes, these approaches usually require elaborative tuning or setting (e.g., manually designed cross-region similarity metric), limiting their performance in complex environments.

In this work, we develop a novel hierarchical region grouping approach for generating and segmenting object proposals by learning a recursive neural network (ReNN). In our ReNN architecture, we incorporate the cross-region similarity metric learning into bottom-up region merging process for end-to-end training. In particular, we define a structured loss that penalizes the incorrect merging candidates by measuring the similarity of adjacent regions and the objectness. In this way, our model explicitly optimizes the cross-region similarity learning and objectness prediction within the recursive iterations. Interestingly, the forward process of ReNN finely accords with the traditional bottom-up region grouping pipeline, leading to a very natural embedding of the two crucial components (i.e., cross-region similarity metric and merging algorithm). Moreover, the objectness score is also learned with the ReNN training, bringing the benefit of fast rejecting the false positive samples.

Obviously, the greedy merging algorithms, that recursively merge two regions with highest merging scores, can be applied for inference with the ReNN model [11]. However, the performance of greedy methods depends heavily on the accuracy of merging scores, since greedy merging is generally sensitive to noise or local minima. In the task of object proposal generation, once a segment of an object is incorrectly merged with the background or other objects, this object has little possibility to be recalled. In addition, we experimentally found that greedy merging leads to incorrect object proposals easily, especially when one segment of an object has similar appearance with background or other surrounding objects. To alleviate this issue, we propose a randomized merging algorithm that introduces randomness in the recursive inference procedure. Instead of merging a pair of neighbouring regions with highest similarity score, we search for k pairs with top k highest similarities, and then randomly pick one pair according to a distribution constructed by their scores. The process is repeated for K times, thus that errors occurred at one random merging process can be corrected in other processes. In this way, it can help to recall more incorrectly merged objects. Figure 1 shows some examples of object proposals generated by our approach.

The key contribution of this work is a deep architecture of recursive neural networks for generating object proposals and preserving their boundaries. This framework jointly optimizes the cross-region similarity and objectness measure together with the hierarchical region grouping process, which is original in literature of object segmentation and detection. Moreover, we design a randomized region merging algorithm with the recursive neural network learning, which introduces randomness to handle the inherent ambiguities of composing regions into candidate objects and thus causes a notable gain in object recall rate. Extensive experimental evaluation and analysis on standard benchmarks (e.g., PASCAL VOC and ImageNet) are provided, demonstrating that our method achieves superior performances over existing approaches in both accuracy and efficiency.

The remainder of the paper is organized as follows. Section II presents a review of the related works. We then introduce our approach and optimization algorithm in detail in Section III and Section IV, respectively. Experimental results, comparisons and analysis are exhibited in Section V. Section VI concludes the paper.

II. Related Work

Many efforts have been dedicated to object proposal generation. Here we roughly divide existing methods into two categories: top-down window-based scoring and bottom-up region grouping, according to their computation process.

A. Window-Based Scoring

This category of methods [12]–[16] attempt to distinguish object proposals directly from the surrounding background through assigning an objectness score to each candidate sub-window. The objectness measures are usually defined in diverse ways, and object proposals generated by sliding windows are then ranked and thresholded by their objectness scores. As a pioneer work, Alexe *et al.* [15] employed saliency cue to measure the objectness of a given window, which was further improved by [17] with learning methods and more complicated features. However, these methods may suffer from expensive computational cost, since they require to search over all locations and scales in images. Recently, to address this problem, BING [14] and Edge Box [13] exploited very simple features such as gradient and contour information to score the windows, and achieved very high computational efficiency. Alternatively, Ren *et al.* [12] proposed a deep learning method based on fully convolutional networks (FCNs) [18] to score windows over scales and locations efficiently. Nonetheless, this method may not locate object accurately, since experimental results show that the recall rate deteriorates as the Intersection over Union (IoU) threshold increases.

B. Region Grouping

This branch of researches [8], [9], [19]–[23] cast the object proposal generation as a process of hierarchical region segmentation or partition. Starting from an initial over-segmentation, these methods usually adopt a cross-region similarity / distance metric [24], [25] that works together with region merging algorithms. As a representative example of these methods, Uijlings *et al.* [8] leveraged four types of low-level features (e.g., color, texture etc.) for similarity computing and generated object proposals via hierarchical greedy grouping. Using similar features with [8], Manen *et al.* [9] learned the merging probabilities and introduced a randomized prim algorithm for region grouping. Following similar hierarchical grouping methods, Wang*et al.* [26] proposed a multi-branch hierarchical segmentation method via learning multiple merging strategies at each step. Arbeláez *et al.* [21] constructed hierarchical segmentations and explored the combinatorial space to combine multi-scale regions into proposals. Xiao *et al.* [10] proposed a complexity-adaptive distance metric for grouping the neighbouring super-pixels. It combined a low-complexity distance and a high-complexity distance to adapt different complexity levels. Krähenbühl and Koltun [19] trained classifiers to adaptively place seeds to hit the objects in the image, and identified a small set of level sets as object proposals for each seed. This method was further improved by ensembling multiple models to generate more diverse proposals [27]. Rantalankila*et al.* [22] integrated local region merging and global graph-cut to generate proposals. Due to their high localization accuracy, they are adopted in many state-of-the-art object detection [1], [2] and object

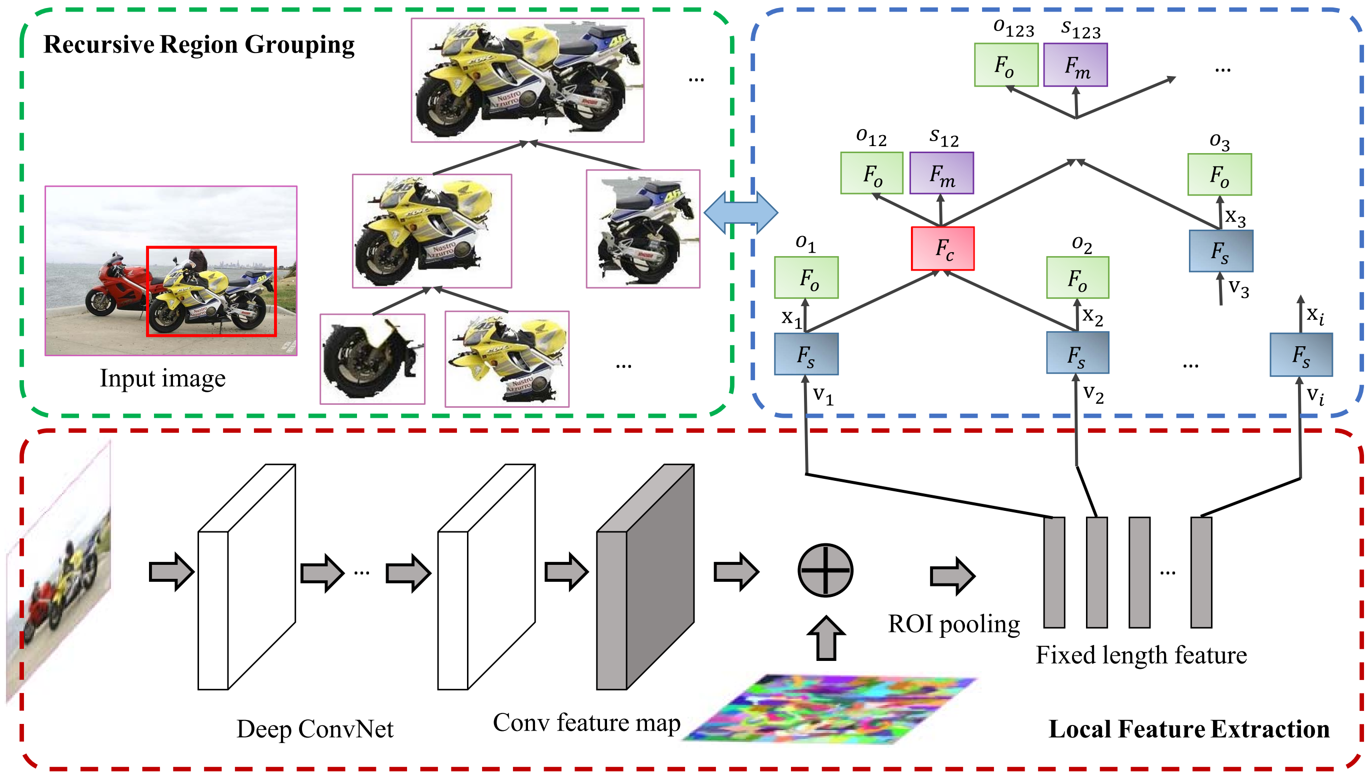


Fig. 2. An overview of our proposed object proposal segmentation framework. The bottom shows local feature extraction, and the top illustrates bottom-up recursive region grouping process. The four modules, F_s, F_c, F_m and F_o, work cooperatively to group regions for generating object proposals.

discovery [7] algorithms. However, these mentioned methods mainly adopt fixed similarity measures for merging regions or finding object proposals, leading to suboptimal performances when handling complex cases. In contrast, our approach adaptively learns the region merging similarity and the objectness measure during the process of hierarchical region grouping. Moreover, our method also introduces randomness into the bottom-up searching of region composition and yields significant improvement over existing methods.

III. Framework of Segmenting Object Proposals

In this section, we introduce our approach in detail. The input image is first over-segmented into N regions with the efficient graph-based method [28]. The Fast R-CNN [29] is used to extract local features for each region. We then design a recursive neural network to group regions and simultaneously predict the associated objectness scores for corresponding proposals. Furthermore, we propose a randomized merging algorithm, which introduces randomness into recursive inference procedure to cope with the inherent ambiguities in the process of merging regions. Figure 2 gives an illustration of our proposed framework.

A. Local Feature Extraction

Since deep features have shown significant improvement than hand-crafted features on various vision tasks [30]–[35], we utilize the Fast RCNN [29] architecture to extract deep local features for each region. The architecture consists of 16 convolutional layers, the same as VGG16-net [30], followed by the region of interests (ROI) pooling layer. Specifically, given an input image, our approach first over-segments it into N regions with the efficient graph-based method [28] and obtains the box for each region that tightly bounds this region. To achieve a better trade-off between speed and accuracy, we follow [29] to resize the input image, thus that the short side of the image is 600, remaining the aspect ratio unchanged. The sixteen convolutional layers take the resized image as input, and produce a pooling of corresponding size feature maps. The ROI pooling layer subsequently extracts a fixed length feature vector for each region.

B. Recursive Neural Networks

We first present some notations that would be used throughout this article. Let $\mathbf{v}_i$ denote the local features of the i-th region, and $\mathbf{x}_i$ denote the corresponding semantic features. $\sigma(\cdot)$ denotes the rectified linear unit (ReLU), where $\sigma(x) = \max(0, x)$.

The core of this framework is the ReNN, which aims to group the regions and simultaneously predict the objectness scores for corresponding proposals in a recursive manner. The ReNN architecture is depicted in Figure 3. The ReNN comprises four modules, i.e., semantic mapper, feature combiner, merging scorer and objectness scorer. Semantic mapper transforms the local features to semantic space which can be further propagated to their parent nodes. Feature combiner computes the joint semantic representations of all neighbouring

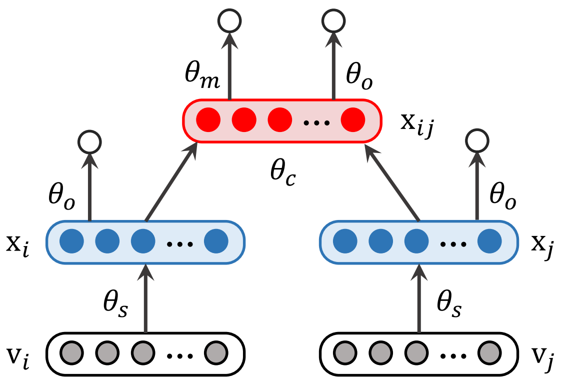


Fig. 3. Illustration of the recursive neural network in our proposed framework. This network computes the scores for merging decision and objectness scores of all regions.

child nodes. Given joint semantic representations, merging scorer calculates the score indicating the confidence that two nodes should be merged. Feature combiner merges the neighbouring nodes according to merging scores, and obtains a hierarchical tree structural segmentations, each of which corresponds to a candidate of proposal. Objectness scorer computes a score which estimates the likelihood of the candidate containing an object. These four modules work cooperatively for proposal segmentation, as illustrated in Figure 2. We describe these four modules in the following.

1) Semantic Mapper: Semantic mapper F_s is a simple feed-forward operator to map the local features into the semantic space in which the combiner operates on. It can be expressed as,

$$\mathbf{x}_i = F_s(\mathbf{v}_i; \theta_s) = \sigma(W_s \mathbf{v}_i + \mathbf{b}_s), \quad (1)$$

F_s captures the region semantic representation, and propagates it to its parent regions through the tree hierarchical structure. To better balance the computational efficiency and accuracy, we empirically set the dimensionality of local features $\mathbf{v}_i$ as 18,432 ($6 \times 6 \times 512$), and that of semantic features $\mathbf{x}_i$ as 256. Hence, the semantic mapper is a one-layer fully-connected network, with 18,432 input and 256 output neurons, followed by the rectified linear unit. $\theta_s = \{W_s, \mathbf{b}_s\}$ are the learnt parameters, in which W_s and $\mathbf{b}_s$ are the weight matrix and bias of the fully-connected layer, respectively.

2) Feature Combiner: Feature combiner F_c recursively takes the semantic features of its two child nodes as input, and maps them to the semantic features of the parent node, formulated as,

$$\mathbf{x}_{i,j} = F_c\left([\mathbf{x}_i, \mathbf{x}_j]; \theta_c\right) = \sigma\left(W_c[\mathbf{x}_i, \mathbf{x}_j] + \mathbf{b}_c\right), \quad (2)$$

F_c aggregates the semantic information of the two child nodes and obtains the semantic representation of the merged node. It takes semantic features of the original regions as leaf nodes, and recursively aggregates them to the root node in a bottom-up manner. In order to ensure the recursive procedure can be applied, the dimensionality of parent node features is set the same as that of child node features. Thus, the architecture of the feature combiner is identical to that of the semantic mapper, except that it has 512 (2×256) input neurons. Similarly, $\theta_c = \{W_c, \mathbf{b}_c\}$ are its learnt parameters, where W_c and $\mathbf{b}_c$ are the weight matrix and bias, respectively.

3) Merging Scorer: Given the joint semantic features of two neighbouring nodes, merging scorer F_m computes a score that indicates the confidence that whether two nodes should be merged, expressed as

$$s_{i,j} = F_m\left(\mathbf{x}_{i,j}; \theta_m\right) = W_m \mathbf{x}_{i,j} + \mathbf{b}_m, \quad (3)$$

The scores determine the pair that should be merged first in both learning and inference stages. It consists of one simple fully connected layer which takes 256 dimensionality combined features as input and produces one scores. $\theta_m = \{W_m, \mathbf{b}_m\}$ are the learnt parameters, where W_m and $\mathbf{b}_m$ are the weight matrix and bias of the fully-connected layer, respectively.

4) Objectness Scorer: Each node of the tree is related to the semantic information of the corresponding region, i.e., the semantic features. Objectness scorer F_o directly predicts objectness scores in semantic feature space.

$$o_i = F_o(\mathbf{x}_i; \theta_o) = \phi\left(W_{o\_1}\sigma\left(W_{o\_0}\mathbf{x}_i + \mathbf{b}_{o\_0}\right) + \mathbf{b}_{o\_1}\right), \quad (4)$$

where $\phi(\cdot)$ is the softmax operation. Our approach rejects candidate proposals that have low scores without compromising the recall rate. We experimentally found that one fully connected layer (merely consisting of 512 parameters) is so simple that it can not well fit thousands of proposals. Thus, we utilize two stacked fully connected layers to implement the objectness scorer, in which the first one is 256 to 256, followed by the rectified linear unit, and the second one is 256 to 2, followed by a softmax layer for objectness prediction. $\theta_o = \{W_{o\_0}, W_{o\_1}, \mathbf{b}_{o\_0}, \mathbf{b}_{o\_1}\}$ are the learnt parameters, where W_{o_0} and $\mathbf{b}_{o\_0}$ are the weight matrix and bias of the first fully-connected layer, while W_{o_1} and $\mathbf{b}_{o\_1}$ are those of the second one.

C. Randomized Merging Algorithm

As discussed above, greedy merging groups the neighbouring regions with the highest similarity score for each iteration. Once a segment of an object mistakenly merges with a neighboring segment that belongs to surrounding objects or background, this object would have little chance to be found. Figure 4 presents an example as an illustration. Given an image with a brown cat and a black-white one, the brown cat is successfully detected using the greedy merging processing, as it is distinguishable from the background (red bounding box in Figure 4). However, the white segment of the other cat incorrectly merges with a piece of background as they have more similar appearance (red circle in Figure 4). In this case, the subsequent merging process misses this cat inevitably. We propose a randomized merging algorithm to alleviate this problem. Instead of merging the neighbouring regions with the highest similarity score for each iteration, our approach selects one pair to merge among the top k highest pairs according to a distribution constructed based on their scores. The randomized merging process can be repeated for several times to increase

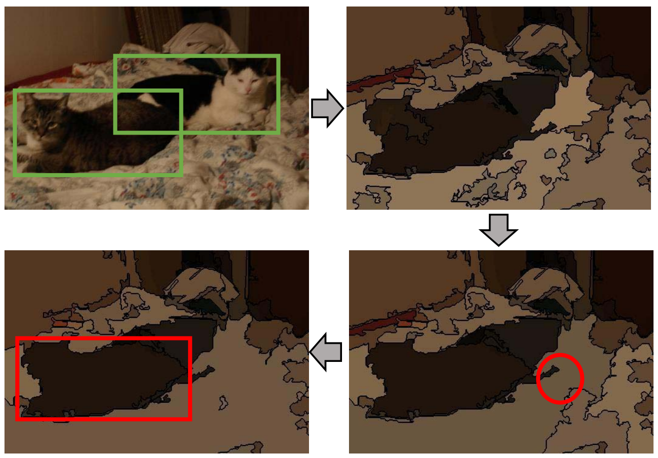


Fig. 4. An example of incorrect merging using the greedy merging algorithm. Top left: Input image; top right: over-segmentation; bottom right: incorrect merging; bottom left: merging result. The black-white cat is lost because its white part incorrectly merges with the background.

the diversity of the generated proposals. This helps to recall more incorrectly merged objects, as explained in Section V-E.

The randomized merging algorithm works as follows. Starting from the semantic features $\{\mathbf{x}_i\}_{i=1}^{N_{seg}}$ and over-segmented regions $\mathcal{R} = \{r_i\}_{i=1}^{N_{seg}}$ where N_{seg} is the number of segments, our approach first computes the merging scores of all neighbouring regions using the feature combiner and merging scorer. Our approach then re-ranks the merging scores to obtain the k pairs of neighbouring regions $\{(r_{i_t}, r_{j_t})\}_{t=1}^{k}$ with the top-k highest scores $\{s_{i_t,j_t}\}_{t=1}^{k}$, and further constructs a multinomial probability distribution according to the k merging scores, expressed as

$$(i_t, j_t) \sim Mult\,(\rho)\,, \tag{5}$$

where

$$\rho_{i_t,j_t} = \frac{\exp\left(s_{i_t,j_t}\right)}{\sum_{t=1}^{k} \exp\left(s_{i_t,j_t}\right)}, \quad t = 1, 2, \cdots, k, \tag{6}$$

where ρ_{i_t,j_t} indicates the probability that the t-th pair of regions can be selected. Our approach randomly draws one pair of regions $\left(r_{i_{t'}}, r_{j_{t'}}\right)$ according to the probability distribution $Mult\,(\rho)$, merges these two regions together, and then computes new merging scores between the resulting region and its neighbours. The process is repeated until the whole image becomes one region. The general process is detailed in Algorithm 1. As the candidate object proposals, we consider the bounding boxes that tightly enclose the segments throughout the hierarchy. Then the objectness scores, learned by the objectness scorer, are used to rank the candidate proposals and the ones with low scores are rejected to get a certain number of proposals.

Algorithm 1 Randomized Merging Algorithm

```
Input: Initial region set $\mathcal{R} = \{r_i\}_{i=1}^{N_{seg}}$
Output: Set of object proposal $\mathcal{P}$
1:  Initialize merging score set $\mathcal{S} = \emptyset$
2:  for all neighbouring region pair $(r_i, r_j)$ do
3:      Calculate merging score $s_{i,j}$
4:      $\mathcal{S} = \mathcal{S} \cup s_{i,j}$
5:  end for
6:  while $\mathcal{S} \neq \emptyset$ do
7:      Get the $k$ highest merging scores $\{s_{i_t,j_t}\}_{t=1}^{k}$
8:      Construct multinomial distribution $Mult\,(\rho)$
9:      Select randomly $t'$-th pair according to $Mult\,(\rho)$
10:     Merge corresponding regions $r_{t'} = r_{i_{t'}} \cup r_{j_{t'}}$
11:     Remove scores regarding $r_{i_{t'}}$: $\mathcal{S} = \mathcal{S} \setminus s_{i_{t'},*}$
12:     Remove scores regarding $r_{j_{t'}}$: $\mathcal{S} = \mathcal{S} \setminus s_{j_{t'},*}$
13:     Compute merging score set $\mathcal{S}_{t'}$ between $r_{t'}$ and its
        neighbours
14:     Update merging score set $\mathcal{S} = \mathcal{S} \cup \mathcal{S}_{t'}$
15:     Update region set $\mathcal{R} = \mathcal{R} \cup r_{t'}$
16: end while
17: Extract object proposals $\mathcal{P}$ from all regions in $\mathcal{R}$
```

IV. Optimization

Suppose that we have the training set $\mathcal{X} = \{(I_i, c_i, b_i)\,|i = 1, 2, ..., N\}$, where N is the number of training samples; I_i is the i-th input sample, including the local features of all regions and the adjacency matrix (as shown in Figure 5(a) and (b)); c_i and b_i are the corresponding class labels of regions and ground truth object bounding boxes, respectively. Our model is jointly trained with two objectives: 1) the merging loss $\mathcal{L}_m$ penalizes incorrect region grouping in the hierarchical tree structure; and 2) the objectness loss $\mathcal{L}_o$ helps to learn the objectness scorer. Therefore, we define the structured loss as

$$\mathcal{L} = \mathcal{L}_m + \lambda\mathcal{L}_o + \frac{\eta}{2}||\theta||_2^2, \tag{7}$$

where $\theta = \{\theta_s, \theta_c, \theta_m, \theta_o\}$ are the set of parameters to learn and $||\theta||_2^2$ is the L2 norm regularization term. λ and η are two balance parameters.

A. Merging Loss

Given an input image I, its bottom-up merging process can be presented as $RN\,(\theta, I, t)$, and it produces a binary tree $t \in \mathcal{T}\,(I)$, where $\mathcal{T}\,(I)$ is the set of all possible binary trees constructed from input I. In the learning stage, the class labels of all the segmented regions are available. We further define $\mathcal{T}\,(I, c)$ as the set of all possible correct trees. Here, a tree is regarded as correct if any region merges with the one belonging to the same class before other regions from different classes. Figure 5 presents some examples of generating correct and incorrect trees from an image.

Inspired by [11], [36], we define a margin loss function $\Delta L : \mathcal{I} \times \mathcal{C} \times \mathcal{T} \rightarrow \mathbb{R}^{+}$, where $\Delta L\,(I, c, t)$ measures the penalty of the construction of a parsing tree t for input I with label c. In the context of recursive merging process, the loss increases when a segment merges with the one from different class before those with the same class label. We denote $N\,(t)$ as the set of non-terminal nodes of tree t, and $subtree\,(d)$ as a subtree underneath the non-terminal node for each $d \in N\,(t)$.

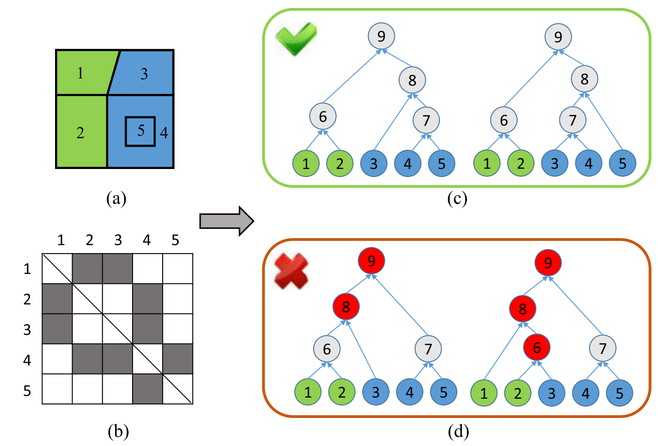


Fig. 5. Examples of generating correct and incorrect trees. (a) Input image, green and blue indicate differently labelled regions. (b) Adjacent matrix of image regions; (c) correct trees; (d) incorrect trees.

Naturally, we formulate the loss by penalizing the incorrect subtrees

$$\triangle L(I, c, t) = \sum_{d \in N(t)} \mathbf{1}\{subtree(d) \notin \mathcal{T}(I, c)\}, \tag{8}$$

where $\mathbf{1}\{\cdot\}$ is an indicator function whose value is 1 when the expression is true and 0 otherwise. Figure 5(d) illustrates two examples of incorrect trees, in which the margin losses are 2 and 3, respectively.

Our goal is to learn a function $f_\theta(\cdot)$ with small expected loss on the unseen inputs. Similar to [11] and [36], we consider the following forms

$$f_\theta(I) = \arg\max_{t \in \mathcal{T}(I)} \{s(RN(\theta, I, t))\}, \tag{9}$$

where $s(\cdot)$ predicts the score for a tree by summing up merging scores of all the merged neighbouring pairs. In the optimization procedure, we aim to learn a score function that assigns higher scores to correct trees than incorrect ones. Given the parameters θ, we first define the margin between the correct tree t_i and another tree t for I_i,

$$s(RN(\theta, I_i, t_i)) - s(RN(\theta, I_i, t)). \tag{10}$$

Intuitively, the margin will be enlarged as the margin loss function $\triangle L(I, c, t)$ increases, expressed as

$$s(RN(\theta, I_i, t_i)) - s(RN(\theta, I_i, t)) \geq \kappa \triangle L(I, c, t), \tag{11}$$

where κ is a parameter. The merging loss can be thus defined as

$$\mathcal{L}_m = \sum_{i=1}^{N} \mathcal{L}_m^{(i)}, \tag{12}$$

where

$$\mathcal{L}_m^{(i)} = \max_{t \in \mathcal{T}(I_i)} \{s(RN(\theta, I_i, t)) + \kappa \triangle L(I_i, c_i, t)\} - \max_{t_i \in \mathcal{T}(I_i, c_i)} \{s(RN(\theta, I_i, t_i))\}. \tag{13}$$

Optimizing the merging loss can maximize the correct trees' scores while minimizing the scores of the highest scoring but incorrect trees. Following [11], we utilize the greedy merging to approximatively find an tree with maximum scores among $\mathcal{T}(I_i)$, and a correct tree with maximum scores among $\mathcal{T}(I_i, c_i)$. The gradients are computed and back propagated based on these two selected trees.

B. Objectness Loss

One of the main advantages of our approach is that it can simultaneously predicts an objectness score for each proposal candidate, which can be used for proposal ranking and rejecting the ones with low scores. We simply employ a softmax classifier with the semantic features of each node. We generate positive and negative samples from all of the regions as follows. Given a region, we first calculate the IoU scores between the box that tightly bounds this region with each ground truth bounding box. If the maximum IoU is larger than 0.5, this region is considered as positive; and if the maximum IoU is smaller than 0.2, it is used as a negative sample. All these regions are considered as useful regions to define the objectness loss. We simply ignore other regions since they may not provide discriminative information. For the i-th useful region, the loss function can be defined as

$$\mathcal{L}_o^{(i)} = -\sum_{l=0}^{1} \mathbf{1}\{l_i = l\} \log\left(p_{i,l}\right), \tag{14}$$

where $p_{i,l}$ is the score corresponding to the likelihood of the region belonging to label l. Hence

$$\mathcal{L}_o = \sum_{i=1}^{N_u} \mathcal{L}_o^{(i)}, \tag{15}$$

where N_u is the number of useful regions.

The model is jointly trained by the stochastic gradient descent (SGD) with momentum [37].

V. Experiment

In this section, we present the extensive experimental results to compare with state-of-the-art methods, demonstrating the superiority of the proposed methods, and analyze the benefit of introducing the randomized merging algorithm for object proposals generation.

A. Experimental Setting

1) Datasets: We first conduct the experiments on the PASCAL VOC2007 dataset [38], which consists of 9,963 images from 20 categories of objects. The model is trained using 422 images of the PASCAL VOC2007's segmentation set. We compare the performance of our approach with those of state-of-the-art methods, and evaluate the contribution of randomized merging algorithm using the 4,952 test images that contain 14,976 objects, including the "difficult" ones. To better demonstrate the effectiveness of the proposed method, we also conducted experiments on the PASCAL VOC 2012 validation set, which contains 15,787 objects in 5,823 images. As our model is trained with 20 object categories on PASCAL VOC, we further investigate the generalization ability of our method to unseen object categories on

ImageNet 2015 validation dataset [39], which contains about 20,000 images of 200 categories, without re-training the model using the training samples from ImageNet.

2) Evaluation Metrics: One of the primary metrics is the Intersection over Union (IoU) measure, where the IoU is defined as the intersection area of the proposal, and the ground truth bounding box divided by their union area. For a fixed number of proposals, the recall rate (the fraction of ground truth annotations covered by proposals) varies as the IoU threshold increases from 0.5 to 1, so that a recall-IoU curve can be obtained. Besides, the curves indicating the recall rate with reference to the number of ranked proposals, are also given, with IoU fixed as both 0.5 and 0.8, respectively. This is widely adopted by many proposal works [26], [40] for evaluation. We also compare the average recall (AR), defined as the average recall when IoU ranges from 0.5 to 1 [40], [41], since AR is considered to be strongly correlated with detection performance.

3) Implementation Details: Following [8], we adopte the efficient graph-based method [28] to produce initial over-segmentations with four parameter values (i.e., $k = 100, 150, 200, 250$), respectively. We implement the proposed model using Caffe open source library [42], and train it by stochastic gradient descent (SGD) with a batch size of 2, momentum of 0.9, weight decay of 0.0005. The learning rate is initialized as 10^{-5} and divided by 10 after 20 epochs. The balance parameter λ in Equation 7 is simply set as 1. Note that it is indeed possible to perform joint training for the fast RCNN and the ReNN. We do not train the model in this way, because only 422 images are provided for training, which easily leads to over-fitting. Therefore, we simply use the fast RCNN model pre-trained on the PASCAL VOC 2007 detection dataset for local feature extraction, and then train our ReNN model alone. In the random merging algorithm, the parameter k is set as 5. To improve the quality of generated proposals, we perform the random merging process for K times ($K = 8$ in our experiments), and rank all the generated proposals using the objectness scores. The proposals with low scores are rejects to get a certain number of proposals for evaluation.

B. Comparison With State-of-the-Art Approaches

In this subsection, we compare our method with recent state-of-the-art methods, including BING [14], Randomized Prim (RP) [9], EdgeBox (EB) [13], Multiscale Combinatorial Grouping (MCG2015)[1] [43], Selective Search (SS) [8], Faster R-CNN (RPN) [12], Complexity Adaptive Distance Metric (CADM) [10], Multi-branch Hierarchical Segmentation (MHS)[2] [26], Geodesic Object Proposals (GOP)[3] [19], Learn to Propose Objects (LPO) [27]. In our experiments, we use Edgebox70 (optimal settings for an IoU threshold of 0.7) for EB, and default settings for others, in order to ensure the best overall performance for these methods. In addition, we follow [40] to control the number of candidates to a specific value for a fair comparison. Since BING, MCG2015, SS, CADM, RPN, EB, MHS and GOP provide sorted proposals, we select n proposals with top n highest scores for evaluation. However, RP and LPO does not provide the scores to rank the proposals, so we simply select the first n proposals in our experiments.

We first analyze the experimental results on the PASCAL VOC 2007 dataset, as depicted in Figure 6. It can be observed that window-scoring-based methods (e.g., EB and RPN) achieve competitive recall rates with a relatively low IoU threshold. This mainly benefits from the exhaustive search over locations and scales, and the accuracy of rejecting the non-objects by the window-scoring-based methods. However, their recall rates drop significantly when the IoU threshold increases. In contrast, region-grouping-based methods yield better performance as the IoU threshold increases. It is shown that MCG2015 performs best among region-grouping-based methods, but it is very time-consuming (over 30s per image) and may not practical especially for real-time object detection systems. It is noteworthy that our method runs 7× faster than MCG2015, and meanwhile outperforms MCG2015 overall, particularly when the number of object proposals is strictly constrained (e.g., with 100 or 500 proposals).

Typically, an IoU threshold of 0.5 is used to measure whether the target object is detected successfully in object detection tasks. However, as suggested in recent works [13], [40], [41], the proposals with an IoU of 0.5 cannot fit the ground truth objects well, usually resulting in a failure of subsequent object detectors. This reveals the fact that the recall rate with an IoU threshold of 0.5 is weakly correlated with the real detection performance. Hence, we also present the curves of recall rate with respect to the number of proposals at a more strict IoU threshold of 0.8, shown in Figure 6(e), to demonstrate the superiority of our method. We believe that our method may be more suitable for object detection systems owing to better localization accuracy and efficiency. Besides, we compare the average recall (AR), considered to have a strong correlation with detection performance, as another important metric for evaluation. As shown in Figure 6(f), our method outperforms other state-of-the-art algorithms, which suggests that it is likely to achieve a better detection performance with the proposals generated by our method.

We also compare the performance on the PASCAL VOC 2012 validation set, as depicted in Figure 7. Note that RPN is trained with the data from both VOC 2007 and 2012 datasets, but RP and our method are learned on the VOC 2007 dataset without re-training here. Even though VOC 2012 is more challenging and larger in size, our method still achieves best performance over other state-of-the-art algorithms, again demonstrating the effectiveness of the proposed method. It is also shown that more improvement over other methods on the VOC 2012 is achieved than that on the VOC 2007.

We present some qualitative examples in Figure 8, including some random samples (top four rows) that contains two or more objects and some samples (the last row) that

[1]MCG2015 is the improved version of original MCG and achieves much better performance.

[2]We only compare with MHS on the PASCAL VOC 2007 dataset, because only the results on this dataset are available.

[3]We only compare with GOP on the PASCAL VOC 2007 and ImageNet datasets, because only the results on these two datasets are available.

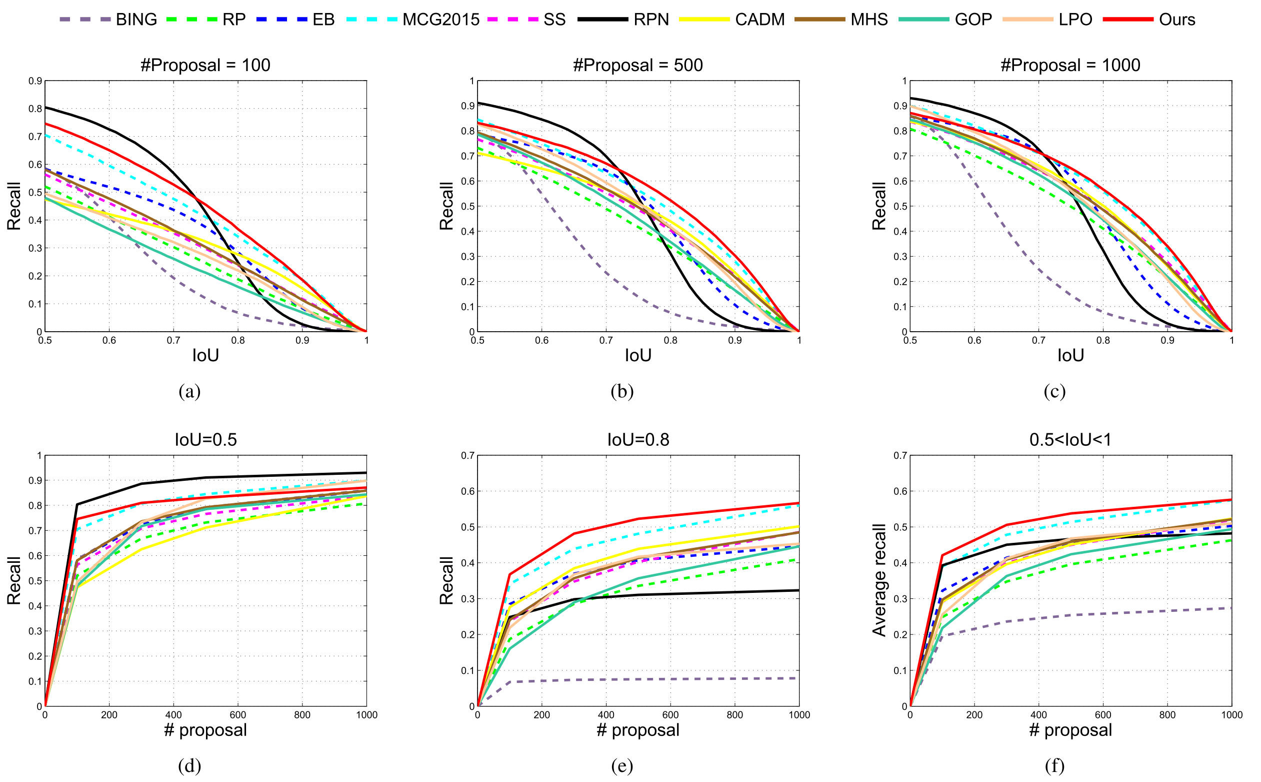


Fig. 6. Comparison of our proposed method and other state-of-the-art approaches on the PASCAL VOC 2007 test set. Best viewed in color.

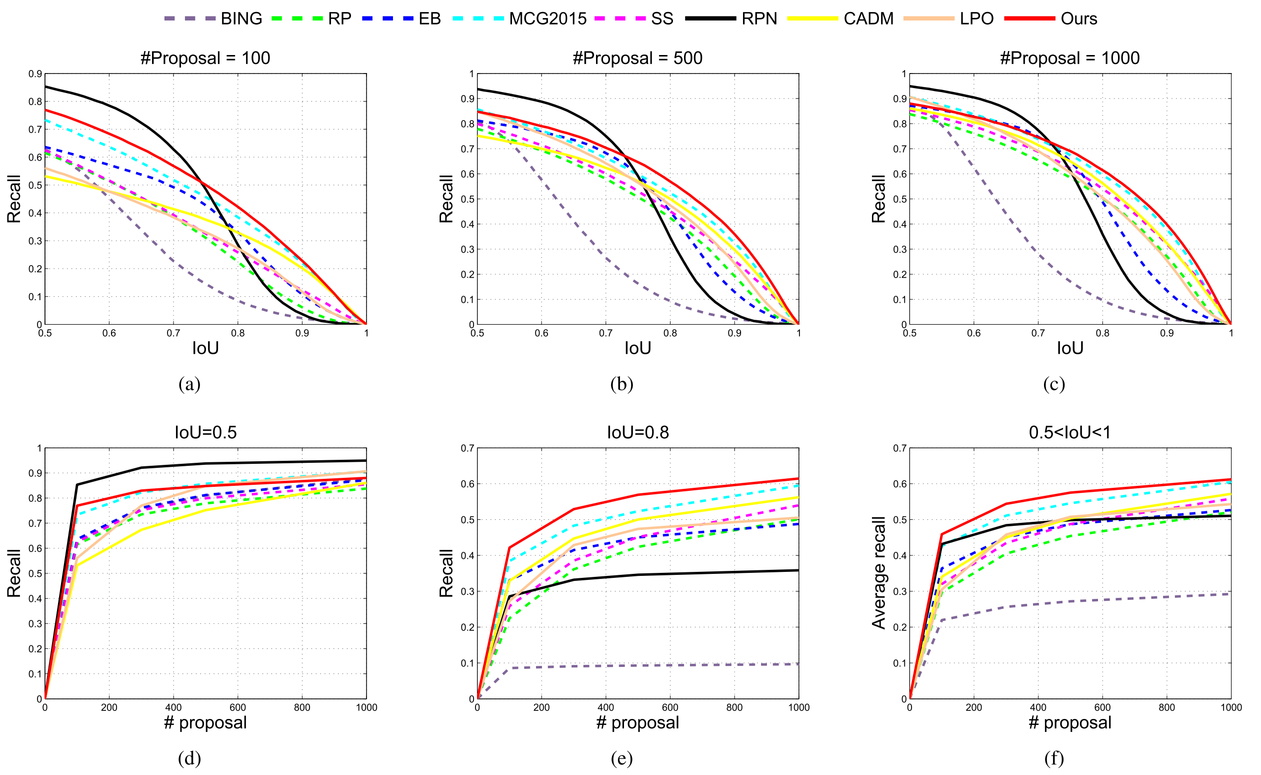


Fig. 7. Comparison of our proposed method and other state-of-the-art approaches on the PASCAL VOC 2012 validation set. Note that our model is trained on PASCAL VOC 2007, but still achieves the best performance against other competitors. Best viewed in color.

challenges our method. We find that, in most cases, our results match well with the ground-truth, and preserve the accurate object boundaries. The missed object are in part tiny ones, e.g., the distant and severely-occluded ones.

Figure 9 analyzes the AR with regard to the ground truth objects in different sizes. It is shown that our method performs slightly worse than RPN if we only consider small-sized objects whose areas are less than 5k pixels. Nevertheless, our

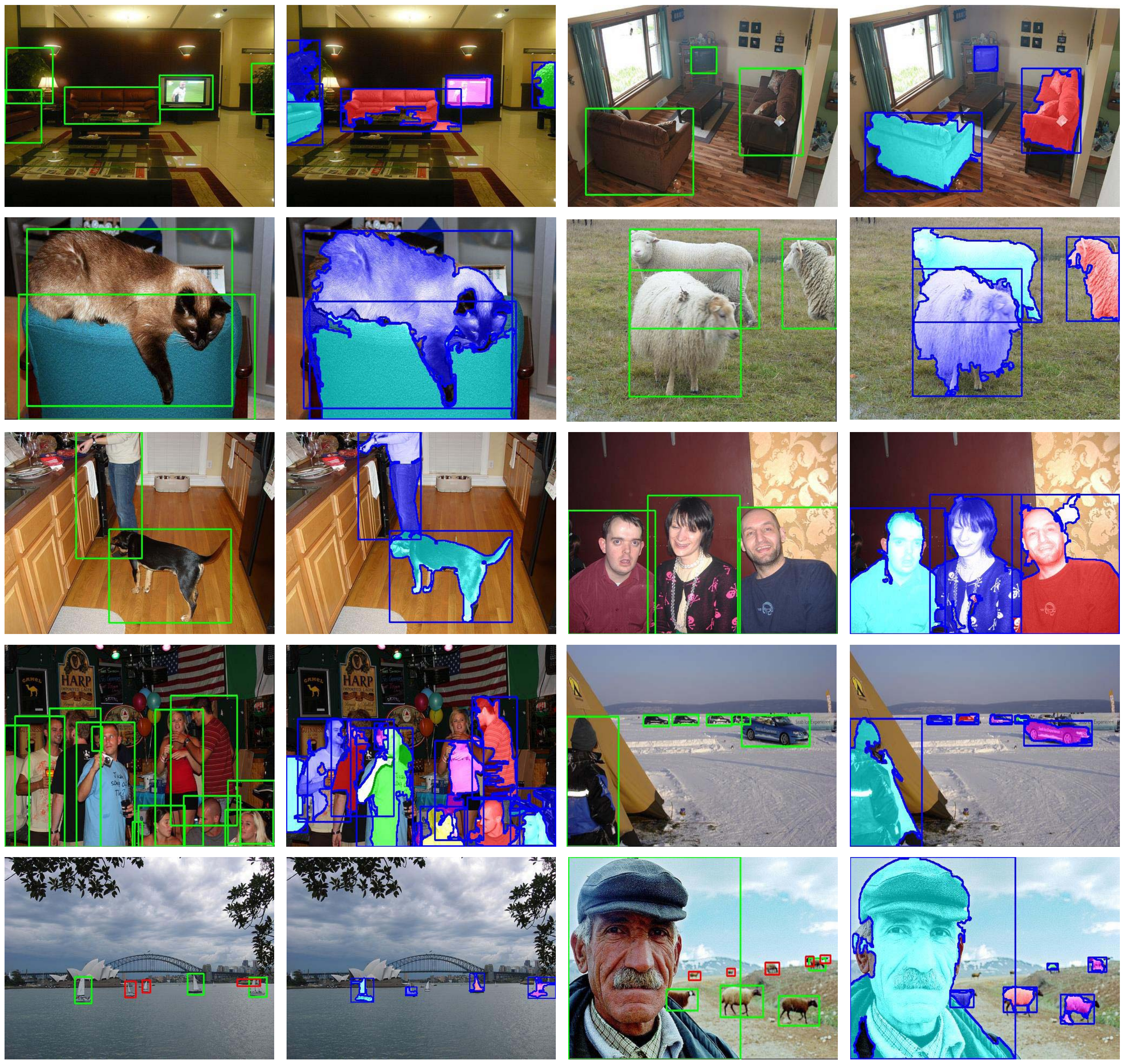


Fig. 8. Qualitative examples of our object proposals. Ground truth boxes are shown in green and red, with green indicating the object is found and red indicating the object is not found. The blue boxes are the object proposals with highest IoU to each ground truth box, and the blue silhouettes are the corresponding object contours. All the samples are taken from PASCAL VOC dataset.

method yields best performance over other competitors in general, especially for recalling objects in larger size. Other grouping-based methods such as SS and CADM show similar results. One possible reason is that grouping-based methods depends heavily on the over-segmentations. In this case, the boundaries of small-sized objects are generally difficult to be well preserved, if the segmentation is not accurate enough. But this problem will not have a significant impact on larger-sized objects. Therefore, region-grouping-based approaches usually exhibit desirable ability to recall objects in relatively large size, but may fail to recall small-sized ones.

C. Object Detection Performance

Since most state-of-the-art object detectors rely on object proposals as a first preprocessing step, it is essential to evaluate the final detection performance with proposals generated by different methods. In this subsection, we conduct experiments to analyze the quality of proposals for object detection tasks. To this end, we use the Fast R-CNN detection framework [29] using both CaffeNet [44] and VGG-Net [30] as the benchmarks. The detectors are trained using PASCAL VOC 2007 trainval set, and tested using the test set for all the experiments here. We compare EB, SS, MCG2015 and RPN with the proposed method. For a fair comparison, we select

TABLE I

COMPARISON OF OBJECT DETECTION PERFORMANCE WITH PROPOSALS GENERATED BY DIFFERENT METHODS. ALL OF THE DETECTORS ARE LEARNED BY THE FAST RCNN ON PASCAL VOC 2007 TRAINVAL SET, AND TESTED WITH 20 CATEGORIES ON THE PASCAL VOC 2007 TEST SET. THE UPPER PART PRESENT THE RESULTS USING FAST R-CNN WITH CAFFENET AND THE LOWER PART SHOWS THOSE USING THE FAST R-CNN WITH VGG-NET

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
SS	**64.8**	67.9	52.9	45.3	20.6	69.9	65.6	70.7	30.0	62.7	59.8	62.3	73.6	64.9	54.8	24.8	49.3	60.2	71.7	55.2	56.4
EB	62.7	68.0	52.5	**45.8**	24.6	65.2	69.2	**70.8**	29.8	**64.2**	59.3	61.7	**74.9**	67.5	59.5	26.8	51.6	55.2	71.1	56.9	56.9
MCG2015	64.2	70.6	50.3	42.5	26.4	70.8	66.4	69.3	29.8	63.7	61.2	61.0	72.7	65.5	57.7	28.1	50.3	61.1	70.3	59.1	57.0
RPN	61.5	**71.2**	53.5	42.6	**29.0**	**72.3**	**72.5**	70.7	32.2	63.5	56.0	62.8	**74.9**	**68.2**	**62.4**	27.6	**53.9**	53.6	66.7	58.5	57.7
Ours	64.6	67.1	**56.1**	44.0	27.7	70.1	66.7	70.5	**35.9**	60.5	**61.8**	**64.0**	74.2	66.1	57.7	**30.2**	52.6	**62.7**	**73.2**	**65.7**	**58.6**
SS	**74.5**	77.7	66.6	**58.3**	32.8	77.9	76.0	**83.8**	43.3	75.9	**70.2**	77.7	80.0	75.7	67.0	32.3	64.3	66.6	**78.1**	65.3	67.2
EB	73.0	**78.1**	67.3	56.2	43.5	**80.2**	77.0	83.0	46.8	74.5	63.6	78.1	80.0	**76.1**	69.5	37.0	67.0	66.4	74.4	63.5	67.7
MCG2015	74.5	77.9	66.6	52.5	42.2	78.4	77.9	80.0	45.1	71.7	66.4	77.5	80.1	74.1	72.4	34.0	67.5	**68.0**	77.6	67.7	67.6
RPN	72.9	77.7	69.2	57.3	**44.8**	78.4	**81.7**	82.5	44.5	**77.3**	62.5	80.4	81.4	74.6	73.6	33.6	**71.3**	65.5	77.1	64.4	68.5
Ours	72.6	77.3	**71.6**	54.0	40.1	78.5	78.2	83.6	**47.6**	74.9	67.3	**81.3**	**83.6**	**76.1**	**74.5**	**39.3**	64.6	65.5	76.7	**72.4**	**69.0**

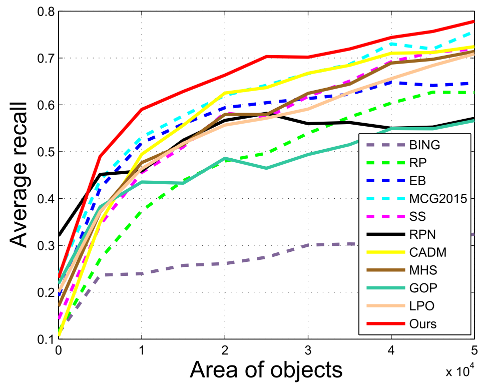


Fig. 9. Comparison of average recall (AR) with respect to different sizes of ground-truth objects on the PASCAL VOC 2007 test set. All of the AR rates are computed with top ranked 500 proposals per image. Best viewed in color.

top-1000 proposals for all of the methods in both training and testing stages. The mean average precision (mAP) and average precision (AP) for each of the 20 categories are shown in Table I. It can be seen that our proposed method achieves the best mAPs of 58.6% and 69.0% using the Fast R-CNN with CaffeNet and VGG-Net, respectively, outperforming other state-of-the-art methods. This also verifies the effectiveness of our method for detection tasks.

D. Generalization to Unseen Categories

In addition, we conduct experiments on the ImageNet 2015 validation set to further evaluate the generalization ability to a wider scope of object categories. Note that all of the learning-based models (e.g., RP, RPN and ours) are trained on the PASCAL VOC dataset, and directly tested on the ImageNet 2015 validation set without re-training. The comparision of the experimental results are shown in Figure 10. It can be seen that our method has comparable performance with MCG2015, and surpasses other methods. No obvious deterioration in performance is observed on the ImageNet 2015 validation set, suggesting that our method does not exclusively fit the 20 specific categories of objects from the PASCAL VOC. In other words, our method is capable of capturing generic objectness information and generalizing to unseen categories. In addition, most state-of-the-art methods achieve similar performance to those on the PASCAL VOC, while RPN suffers from severe performance drop. One possible reason is that category information is exploited to learn class-specific detectors, which makes the RPN overfit 20 categories of objects from the PASCAL VOC.

TABLE II

COMPARISON OF GREEDY MERGING AND RANDOMIZED MERGING ON THE PASCAL VOC 2007 TEST SET. WE REPORT THE RESULTS USING TOP RANKED 500 PROPOSALS. R@0.5 AND R@0.8 INDICATE THE RECALL RATES WITH AN IOU THRESHOLD OF 0.5 AND 0.8, RESPECTIVELY, AND AR IS THE AVERAGE RECALL

	R@0.5	R@0.8	AR
Greedy	0.872	0.423	0.489
Random	0.870	0.405	0.478

E. Evaluation of Randomized Merging Algorithm

In this subsection, we evaluate the contribution of the proposed randomized merging algorithm. We compare the performance of conventional greedy and the proposed randomized merging algorithms on the PASCAL VOC 2007. In the first setting, we allow the randomized merging algorithm to be performed only one time in each recursion step, and the experimental results are shown in Table II. It can be seen that greedy merging and one-time randomized merging achieve comparable results. However, with the increase of randomized times, our merging algorithm generates more diverse proposals. Here we conducted the experiments and compare the results obtained with different randomized times. The number of object proposals are fixed as 500 and 1000, respectively. Figure 11 clearly shows that the AR rate improves as the random times increases, and then goes near saturation eventually. This approach provides a notable gain in recall rates compared to greedy merging strategy. It is also shown that the recall rate with an IoU threshold of 0.5 first keep fixed and then drops as the random times increases, while that with 0.8 boosts consistently. This suggests that the predicted objectness scores are not accurate enough with small IoU values with the ground truth bounding boxes.

Another critical issue is that whether stable performance can be achieved by our proposed method, since we introduce randomness to the merging process. To better clarify this

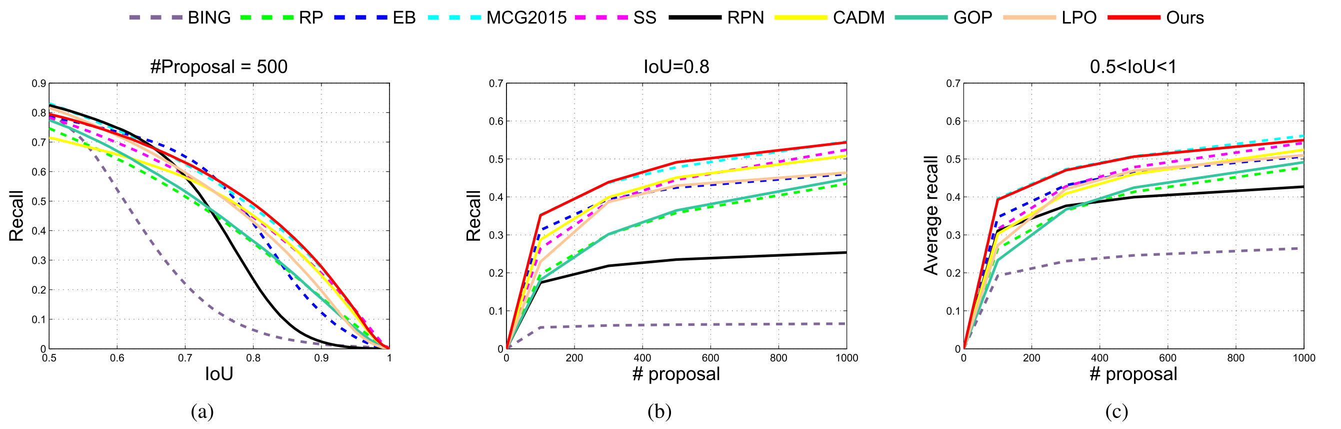


Fig. 10. Comparison of our proposed method and other state-of-the-art approaches on the ImageNet 2015 validation set. Best viewed in color.

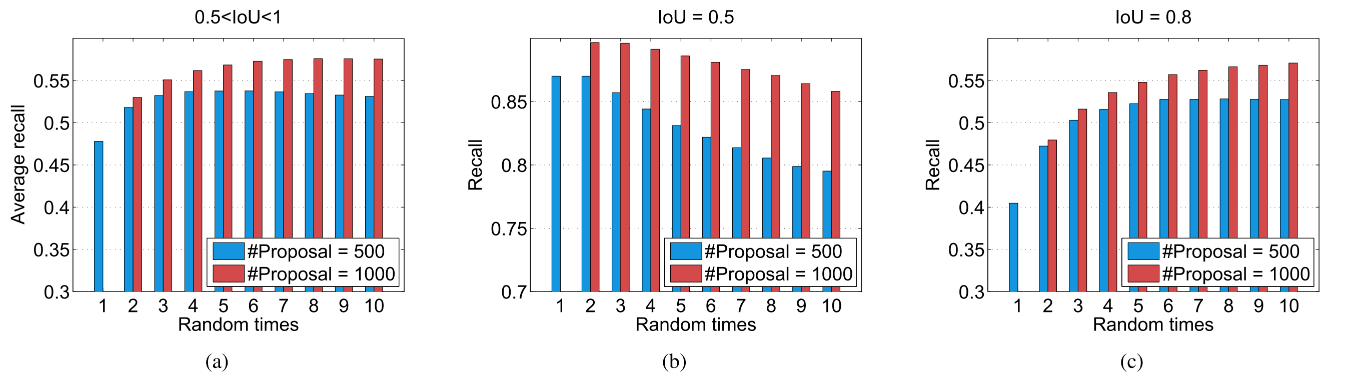


Fig. 11. Comparison of recall rates with different randomized times on the PASCAL VOC 2007 test set. We report the results of both top 500 and 1000 proposals for a fair comparison. Note that the number of proposals generated by one-time randomized merging is less than 1000, so we cannot provide this result here. Best viewed in color.

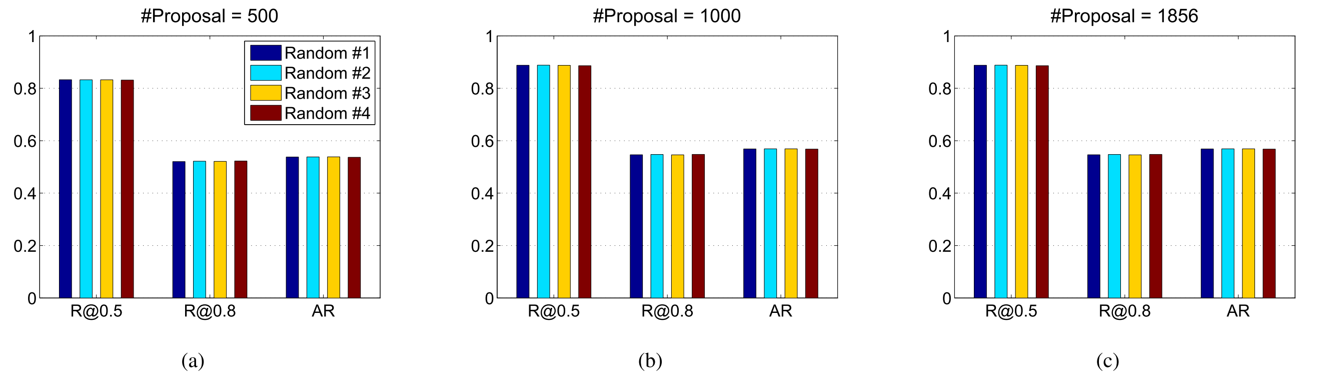


Fig. 12. Comparison of recall rates in four groups of randomized merging experiments on the PASCAL VOC 2007 test set. We report the results using top ranked 500, 1000 and all of the proposals. R@0.5 and R@0.8 mean the recall rates with an IoU threshold of 0.5 and 0.8, respectively, and AR is the average recall. Best viewed in color.

problem, we conduct four groups of experiments, and we repeat the randomized merging process for five times for each group. As shown in Figure 12, our method also exhibits great stability in recall rates and average recall with different numbers of object proposals.

F. Efficiency Analysis

In this subsection, we present the comparison of the efficiency of our model and the state-of-the-art methods. The execution time of MHS [26] are directly taken from [26] as its codes are not available. The deep-learning-based methods (i.e., ours and RPN) are conducted on a single NVIDIA TITAN X GPU, and the rest are carried out on a desktop with an Intel i7 3.4GHz CPU and 16G RAM. The average running time of all the methods for generating 1,000 proposals on the PASCAL VOC 2007 dataset are reported in Table III. It can be seen that window scoring methods achieve relatively high computational efficiency because of using very simple

TABLE III

COMPARISON OF THE AVERAGE RUNNING TIME (SECOND PER IMAGE) FOR GENERATING 1000 PROPOSALS AND THE AVERAGE RECALL (AR) ON THE PASCAL VOC 2007 TEST SET

Type	Method	Time	AR (%)
Windows scoring	Bing [14]	0.2	27.38
	RPN [12]	0.2	48.19
	EB [13]	0.3	50.30
Regions grouping	RP [9]	1.0	46.30
	GOP [19]	1.0	49.38
	LPO [27]	1.1	50.98
	MHS [26]	2.8	52.08
	SS [8]	10.0	51.91
	CACD [10]	22.0	52.30
	MCG2015 [43]	30.0	57.45
	Ours	4.2	**57.60**

features and efficient scoring methods. Among those region grouping methods, RP, GOP and LPR run slightly faster than our method, but their performance are much inferior than ours on both PASCAL VOC and ImageNet datasets (see Figure 6, 7 and 10). MCG2015 achieves comparable results, but it is extremely time-consuming. It is noteworthy that our method achieves the best performance among all methods while sharing quite high running efficiency. Specifically, for our method, it takes about 0.2s for feature extraction, and about 0.5s for one-time random merging. The random merging process is repeated for 8 times, thus the running time is 4.2s per image.

VI. CONCLUSION

In this paper, we have presented a simple yet effective approach to hierarchically segment object proposals by develping a deep architecture of recursive neural networks. We incorporate the similarity metric learning into the bottom-up region merging process for end-to-end training, rather than manually designing various representations. In addition, we introduce randomness into the greedy search to cope with the ambiguity in the process of merging regions, making the inference more robust against noise. Extensive experiments on standard benchmarks demonstrate the superiority of our approach over state-of-the-art approaches. In addition, the effectiveness of our method for real detection systems is also verified. In future work, this proposed framework can be tightly combined with category-specific object detection methods.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 346–361.

[3] Y. Wei *et al.*, "HCP: A flexible CNN framework for multi-label image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1901–1907, Jun. 2015.

[4] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2017, pp. 464–472.

[5] T. Chen, Z. Wang, G. Li, and L. Lin, "Recurrent attentional reinforcement learning for multi-label image recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6730–6737.

[6] Y. J. Lee and K. Grauman, "Learning the easy things first: Self-paced visual category discovery," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1721–1728.

[7] M. Cho, S. Kwak, C. Schmid, and J. Ponce. (2015). "Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals." [Online]. Available: https://arxiv.org/abs/1501.06170

[8] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.

[9] S. Manen, M. Guillaumin, and L. Van Gool, "Prime object proposals with randomized Prim's algorithm," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2013, pp. 2536–2543.

[10] Y. Xiao, C. Lu, E. Tsougenis, Y. Lu, and C.-K. Tang, "Complexity-adaptive distance metric for object proposals generation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 778–786.

[11] R. Socher, C. C. Lin, C. Manning, and A. Y. Ng, "Parsing natural scenes and natural language with recursive neural networks," in *Proc. Int. Conf. Mach. Learn.*, Jun. 2011, pp. 129–136.

[12] S. Ren, K. He, R. Girshick, and J. Sun. (2015). "Faster R-CNN: Towards real-time object detection with region proposal networks." [Online]. Available: https://arxiv.org/abs/1506.01497

[13] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 391–405.

[14] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300 fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 3286–3293.

[15] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2010, pp. 73–80.

[16] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2189–2202, Nov. 2012.

[17] Z. Zhang, J. Warrell, and P. H. Torr, "Proposal generation for object detection using cascaded ranking SVMs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2011, pp. 1497–1504.

[18] J. Long, E. Shelhamer, and T. Darrell. (2014). "Fully convolutional networks for semantic segmentation." [Online]. Available: https://arxiv.org/abs/1411.4038

[19] P. Krähenbühl and V. Koltun, "Geodesic object proposals," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014, pp. 725–739.

[20] J. 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, Jul. 2012.

[21] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2014, pp. 328–335.

[22] P. Rantalankila, J. Kannala, and E. Rahtu, "Generating object segmentation proposals using global and local search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2417–2424.

[23] M. Van den Bergh, G. Roig, X. Boix, S. Manen, and L. van Gool, "Online video SEEDS for temporal window objectness," in *Proc. IEEE ICCV*, Dec. 2013, pp. 377–384.

[24] L. Lin, G. Wang, W. Zuo, X. Feng, and L. Zhang, "Cross-domain visual matching via generalized similarity measure and feature learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1089–1102, Jul. 2017.

[25] S.-Z. Chen, C.-C. Guo, and J.-H. Lai, "Deep ranking for person re-identification via joint representation learning," *IEEE Trans. Image Process.*, vol. 25, no. 5, pp. 2353–2367, May 2016.

[26] C. Wang, L. Zhao, S. Liang, L. Zhang, J. Jia, and Y. Wei, "Object proposal by multi-branch hierarchical segmentation," in *Proc. CVPR*, Jun. 2015, pp. 3873–3881.

[27] P. Krähenbühl and V. Koltun, "Learning to propose objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1574–1582.

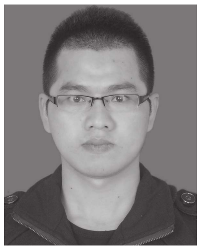
[28] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[29] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, Sep. 2015, pp. 1440–1448.

[30] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

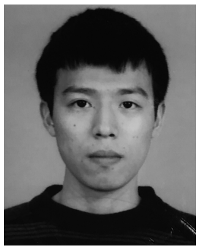
[31] T. Chen, L. Lin, L. Liu, X. Luo, and X. Li, "DISC: Deep image saliency computing via progressive representation learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 6, pp. 1135–1149, Jun. 2016.

[32] L. Lin, K. Wang, W. Zuo, M. Wang, J. Luo, and L. Zhang, "A deep structured model with radius–margin bound for 3D human activity recognition," *Int. J. Comput. Vis.*, vol. 118, no. 2, pp. 256–273, 2016.
[33] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang, "Active self-paced learning for cost-effective and progressive face identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 7–19, Jan. 2018.
[34] T. Chen, L. Lin, R. Chen, Y. Wu, and X. Luo, "Knowledge-embedded representation learning for fine-grained image recognition," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 627–634.
[35] Z. Wang, T. Chen, J. Ren, W. Yu, H. Cheng, and L. Lin, "Deep reasoning with knowledge graph for social relationship understanding," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018, pp. 2018–2021.
[36] B. Taskar, D. Klein, M. Collins, D. Koller, and C. D. Manning, "Max-margin parsing," in *Proc. EMNLP*, 2004, pp. 1–3.
[37] L. Bottou, "Stochastic gradient descent tricks," in *Neural Networks: Tricks of the Trade*. Berlin, Germany: Springer, 2012, pp. 421–436.
[38] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
[39] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2014.
[40] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. (2015). "What makes for effective detection proposals?" [Online]. Available: https://arxiv.org/abs/1502.05082
[41] X. Chen, H. Ma, X. Wang, and Z. Zhao, "Improving object proposals with multi-thresholding straddling expansion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2587–2595.
[42] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, Nov. 2014, pp. 675–678.
[43] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik. (2015). "Multiscale combinatorial grouping for image segmentation and object proposal generation." [Online]. Available: https://arxiv.org/abs/1503.00848
[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1097–1105.



Tianshui Chen received the B.E. degree from the School of Information and Science Technology, Sun Yat-sen University, Guangzhou, China, in 2013, where he is currently pursuing the Ph.D. degree in computer science with the School of Data and Computer Science. His current research interests include computer vision and machine learning.

Liang Lin (M'09–SM'15) was a Post-Doctoral Fellow with the University of California at Los Angeles, Los Angeles, CA, USA, from 2008 to 2010. From 2014 to 2015, he was a Senior Visiting Scholar with The Hong Kong Polytechnic University and The Chinese University of Hong Kong. He is the Executive Research and Development Director of the SenseTime Group Ltd., and a Full Professor with Sun Yat-sen University. He is the Excellent Young Scientist of the National Natural Science Foundation of China. He currently leads the SenseTime Research and Development teams to develop cutting-edges and deliverable solutions on computer vision, data analysis and mining, and intelligent robotic systems. He has authored and co-authored over 100 papers in top-tier academic journals and conferences (e.g., 12 papers in TPAMI/IJCV and 50+ papers in CVPR/ICCV/NIPS/IJCAI). He is a fellow of IET. He was a recipient of the Best Paper Dimond Award at the IEEE ICME 2017, the Best Paper Runners-Up Award at ACM NPAR 2010, the Google Faculty Award in 2012, the Best Student Paper Award at the IEEE ICME 2014, and the Hong Kong Scholars Award in 2014. He served as an Area/Session Chairs for numerous conferences, such as ICME, ACCV, and ICMR. He has been serving as an Associate Editor for the IEEE TRANSACTIONS ON HUMAN-MACHINE SYSTEMS, *The Visual Computer*, and *Neurocomputing*.



Xian Wu received the B.E. degree from the School of Software, Sun Yat-sen University, Guangzhou, China, in 2015, where he is currently pursuing the M.Sc. degree in software engineering with the School of Data and Computer Science. His current research interests include computer vision and machine learning.

Nong Xiao (M'97) received the B.S. and Ph.D. degrees in computer science from the College of Computer, National University of Defense Technology (NUDT), China, in 1990 and 1996, respectively. He is currently a Professor with the School of Data and Computer Science, Sun Yat-sen University. His current research interests include large-scale storage system, network computing, and computer architecture. He has over 130 publications to his credit in journals and international conferences, including the IEEE TSC, IEEE TMM, JPDC, JCST, HPCA, ICCAD, MIDDLEWARE, MSST, IPDPS, CLUSTER, SYSTOR, and MASCOTS. He is a member of ACM.

Xiaonan Luo was the Director of the National Engineering Research Center of Digital Life, Sun Yat-sen University, Guangzhou, China. He is currently a Professor with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, China. He received the National Science Fund for Distinguished Young Scholars granted by the National Nature Science Foundation of China. His research interests include computer graphics, CAD, image processing, and mobile computing.