Learning to Generate Object Segmentation Proposals with Multi-modal Cues

Haoyang Zhang[1,2], Xuming He[2,1], Fatih Porikli[1,2]

[1]The Australian National University, [2]Data61, CSIRO, Canberra, Australia
{haoyang.zhang,xuming.he, fatih.porikli}@anu.edu.au

Abstract. This paper presents a learning-based object segmentation proposal generation method for stereo images. Unlike existing methods which mostly rely on low-level appearance cue and handcrafted similarity functions to group segments, our method makes use of learned deep features and designed geometric features to represent a region, as well as a learned similarity network to guide the grouping process. Given an initial segmentation hierarchy, we sequentially merge adjacent regions in each level based on their affinity measured by the similarity network. This merging process generates new segmentation hierarchies, which are then used to produce a pool of regional proposals by taking region singleton, pairs, triplets and 4-tuples from them. In addition, we learn a ranking network that predicts the objectness score of each regional proposal and diversifies the ranking based on Maximum Marginal Relevance measures. Experiments on the Cityscapes dataset show that our approach performs significantly better than the baseline and the current state-of-the-art.

1 Introduction

Object proposal generation, which aims to produce a set of high-quality object candidates in an image, has become a core component in modern object detection [1–3] and segmentation pipelines. By focusing on a relatively small set of object-like regions, it enables us to use better object representations and improves significantly the accuracy of target vision tasks. While most work in object proposal generation focus on generating bounding boxes for object detection [4–7], object segments or region proposals play an important role in semantic segmentation and object segmentation [8, 9].

Compared to bounding box proposals, generating object segment candidates is more challenging due to inaccuracies in bottom-up segmentation processes. Early work incorporate boundary consistency and smoothness priors through superpixel grouping [5, 9] or MRF-based segmentation [8, 10, 11]. They rely on handcrafted image features to group pixels into region proposals. More recent approaches use deep ConvNets to learn the feature representation and directly predict class-agnostic object masks [12, 13]. However, such end-to-end learning of a deep network makes it difficult to incorporate additional input data from other sensor modalities, such as depth cues [14, 15]. It may require retraining of the full system using a large dataset with instance-level annotations, which can be expensive and time-consuming.

In this work, we consider the problem of generating object segmentation proposals with stereo image inputs. To efficiently incorporate the depth cues computed from the stereo, we take an alternative deep learning approach, and learn an iterative merging process for generating a diverse set of high-quality region proposals. Unlike the previous global approaches, we mainly focus on learning a representation for object-driven perceptual grouping, which is an easier problem due to its local nature and potential to be modeled by a simpler network. More importantly, it enables us to design a late fusion strategy to incorporate the noisy depth cues into grouping without retraining the full deep network pipeline.

Specifically, our method consists of two stages. We start from an initial segmentation hierarchy of the left image and sequentially merge neighboring regions in each level of the hierarchy based on an affinity score predicted by a learned similarity network. This merging process generates new hierarchies of image segments, which is used to produce a pool of regional proposals by taking single, pair, triple and 4-tuple neighboring segments from the hierarchies. We then learn a ranking network that predicts the objectness score of each region proposal. Our similarity and ranking network use a combination of learned deep features for intensity and designed geometric features for depth cue. While the similarity network predicts how likely two regions belong to the same object instance or the same background class, the ranking network estimates the overlap ratio with respect to the ground-truth for each candidate region.

We evaluate our algorithm on the Cityscapes dataset[16] with comparisons to Selective Search baseline and one of the stat-of-the-art methods, Multiscale Combinatorial Grouping (MCG). Our results show that we achieves significant improvement over the MCG method. The main contributions of our work are three folds: first, we propose a deep learning approach to the multi-modal object segmentation proposal generation; second, we design an alternative method to produce region proposals with a learned merging network and ranking network; and finally, our method achieves superior performance to the strong baselines on the challenging Cityscapes dataset.

2 Related Work

Generating high-quality object proposals plays an important role in the recent advance of object detection [1, 2], and has drawn much attention in computer vision literature [17]. Many early work use handcrafted features to score bounding box hypotheses [4, 6, 7] and generate a set of object candidates mainly for detection task. More recent work in object detection literature [3], however, begin to learn the proposal generation as an integrated component of detection networks and do not rely on a preprocessing step for bounding box generation any more.

In contrast to bounding box generation, much less progress has been made in object segmentation proposal generation. One strategy is to formulate the problem as a series of foreground segmentation tasks [8, 10, 11, 18]. By solving multiple graph-cut problems with diverse seeds, they can generate a large set of region proposals. In particular, Lee et al. [19] learns a parametric energy function to combine handcrafted mid-level cues in order to generate a diverse set of regional proposals. An alternative approach is to group superpixels into a hierarchical segmentation and seek semantic meaningful region proposals [5, 9, 20]. The MCG [9] generates multiscale UCMs and takes singletons, pairs, triplets and 4-tuples in their hierarchical segmentations as object proposals. [21] integrates both global foreground and local grouping strategies. Yanulevskaya et al. [22] also learns a grouping method for proposal generation, but their method uses manually designed appearance features and predefined similarity metrics. By contrast, our method directly learns a similarity neural network to group regions.

Deep learning based methods have been applied to object proposal generation, including bounding boxes [23, 3, 24] and segmentation [12, 13]. Deep segmentation proposal methods usually take an end-to-end framework and use a global network to produce final candidate masks. By contrast, we adopt a learning-to-merge strategy and learn a simpler network structure. Such strategy has been successfully applied to semantic segmentation [25]. We note that our similarity network does not compute a distance metric between two regions [26]. Instead, it is an affinity score for grouping.

Few work have addressed the multi-modal object proposal generation task. Chen et al. [15] use depth information in an energy minimization framework to generate object proposals in a form of 3D bounding boxes. Bleyer et al. [14] design an iterative labeling strategy to segment object proposals from stereo images, which is computationally expensive.

Most of existing object segmentation proposal methods use ranking to improve the quality of the candidate pool. For example, [8, 9] both learn a Random Forest with a set of low-level features to rank the object proposals. By contrast, we learn a neural network to rank object proposals.

We build our work on top of several existing techniques. Our model uses FCN network [27] and Hypercolumn feature [28] trained on PASCAL-Context dataset[29]. We extract deep features using "feature masking" technique introduced in [30].

3 Our Method

We aim to generate a set of object segmentation proposals and their objectness scores from a pair of stereo images. To this end, we design a segmentation proposal generation pipeline that learns to fuse multimodal cues and to merge oversegmentation into object candidates. Figure 1 illustrates an overview of our approach. We first estimate a dense depth map of the scene using the stereo images, and build a segmentation hierarchy of the left image. Given the initial segmentation hierarchy, we represent all the regions in the hierarchy using convolutional and depth features. We then train a neural network to predict regions' affinity, and refine the segmentation hierarchy by incrementally merging

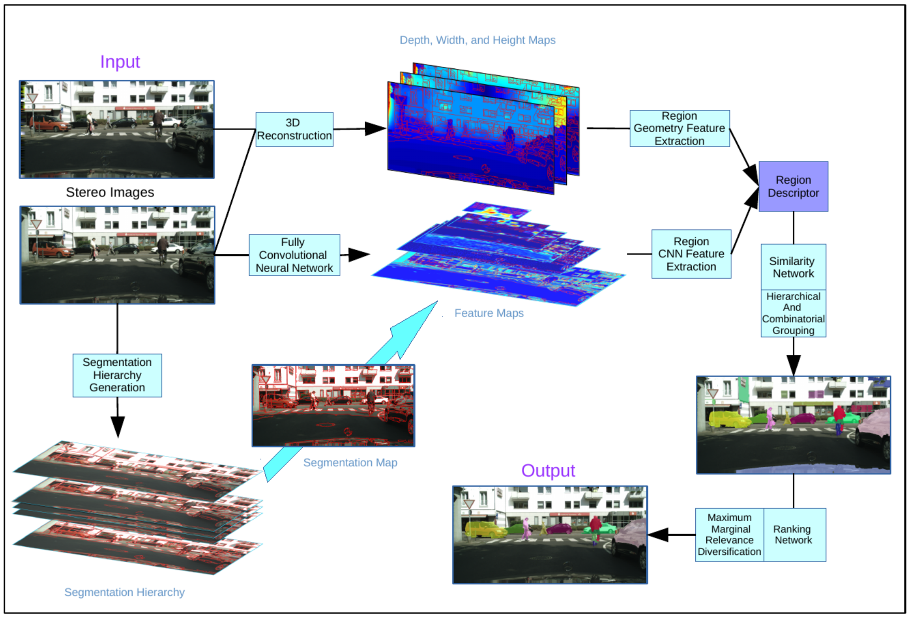
Input
Stereo Images
3D Reconstruction
Depth, Width, and Height Maps
Region Geometry Feature Extraction
Region Descriptor
Fully Convolutional Neural Network
Region CNN Feature Extraction
Feature Maps
Similarity Network
Hierarchical And Combinatorial Grouping
Segmentation Hierarchy Generation
Segmentation Map
Segmentation Hierarchy
Output
Maximum Marginal Relevance Diversification
Ranking Network

Fig. 1. Overview: Our system takes as input a pair of stereo images. We first generate a segmentation hierarchy, compute the convolutional feature maps and reconstruct the 3D scene. Then, we extract descriptors for regions in the segmentation hierarchy. Next, we iteratively merge adjacent regions based on their affinity score predicted by a similarity network to generate object proposals. Finally, we rank these object proposals through a ranking network and diversify the ranking.

adjacent regions from the bottom level based on the estimated similarity. From the new segmentation hierarchies, we extract region singletons, pairs, triplets and 4-tuples as object segment proposals. Finally, we rank these object proposals through a learned ranking network and diversify the ranking based on Maximum Marginal Relevance measures [8]. We now describe each stage of our pipeline in detail.

3.1 Initial Segmentation Hierarchy Generation

The first step of our method constructs an initial segmentation hierarchy of the left image. To generate the segmentation hierarchy, we use the Structured Edge Detection [31] on the image to obtain an edge map for its efficiency and accuracy. An Ultrametric Contour Map (UCM) [9] is generated based on the estimated edge probability map. Then we threshold the UCM at five different levels to create the segmentation hierarchy. The thresholds are chosen such that the numbers of regions from the base level to the top level are roughly 1024,768,512,384 and 256, respectively. For every region, we also record its child regions in the hierarchy, which enables efficient propagation of descriptors of regions from the base level to higher levels through the hierarchy.

3.2 Multimodal Region Representation

For each region in the segmentation hierarchy, we compute two types of features to capture its appearance and 3D geometric properties. We take a bottom-up manner to compute the region features of all the hierarchy levels efficiently. We only calculate those features explicitly for the base level regions and use max-pooling or weighted average-pooling to obtain features of higher level regions recursively.

Appearance Features We extract a set of rich deep features to encode the appearance of a region. We first feed the left image into a Fully Convolutional Network (FCN) [27] to generate multiple layers of feature maps for the entire image. We choose the FCN-8s model trained on PASCAL-Context dataset [29] for the scene labeling task because of its superior performance and diverse set of 59 classes such as $sky, ground, grass, building, road, person,$ *bicycle and car etc.*. The feature map outputs from $pool1, pool2, pool3, pool4, pool5\ and\ fc7\ layers$ are used as our representation, inspired by the "Hypercolumns" concept proposed by Hariharan et al. [28].

Given the feature maps, we compute the appearance features of a region by masking and max-pooling. As the feature maps of different layers are not of the same size, the deep features of a region cannot be directly masked out from these maps. A straightforward way to solve this problem is to upsample the feature maps to the same size as the image [28]. However, due to high dimensionality and varying sizes of the feature maps, e.g the output from $fc7\ layer$ has 4096 dimensions and a very small size (17×33 in our case), such upsampling is very time-consuming and memory-costly.

To tackle this issue, we adopt the convolutional feature masking technique proposed by Dai et al. [30]. Specifically, we first compute the receptive field for every activation in each layer output according to the receptive field geometry [32]. Then we project each activation onto the image plane, which is located at the center of its receptive field. We define a "domain of influence" for an activation on the image plane, which has the same center as its receptive field and a smaller size such that neighboring activations do not overlap. For example, the activation at location $(1, 1)$ in $pool5\ layer$ may have a square "domain of influence" with its center at $(16, 16)$ and its side length as 32 in the image domain. If over 50% of the "domain of influence" of an activation is covered by a region mask, we label this activation as active for this region and it will be included in the calculation of region feature. By this labeling process, we project the base level region masks in the image plane onto the feature maps and then we do max-pooling in the projected masks on the feature maps to extract regions' deep features. Figure 2 (left) shows an example of our feature computation process. This generates a 5568-dimensional feature to encode the region's appearance. When computing

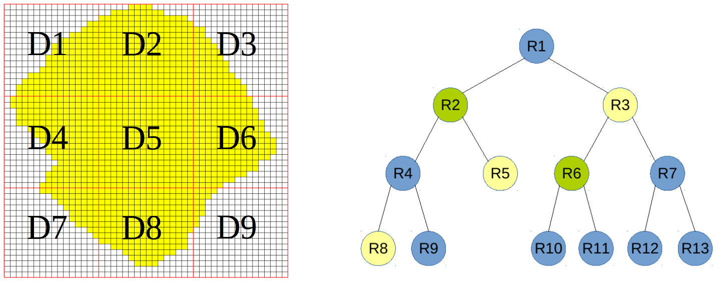

Fig. 2. Left: Illustration of "domain of influence" and feature masking. D1∼D9 red rectangles are the domains of influence of activations A1∼A9 in *pool*5 *layer*. The yellow mask is a region and only A2, A4, A5, A6 and A8 are activated by this region, as over half of their domain of influence is overlapped by this region. **Right:** Illustration of combinatorial grouping. Singletons:R1∼R13. Pairs:(R2,R6). Triplets:(R3,R5,R8).

the deep features of regions in higher levels, we only need to do max-pooling among their child region features.

3D Geometric Features To encode geometric properties of a region, we extract two sets of 3D geometric features. We first estimate the dense depth map using the method [33] and convert it into a point cloud representation in the camera coordinate system according to the camera parameters.

Given the point cloud and a base level region mask, we segment out the subset of the point cloud using the mask. The subset is used to compute two sets of features to describe the region's geometric properties. Denoting the position of a 3D point as (x, y, z), we first compute the center of the region as one set of features, including $mean\ x, mean\ y, and\ mean\ z$. Another set of features describe the spatial distribution of the point cloud, consisting of three histograms, one for each dimension of the point cloud. Specifically, for the width, height and depth dimension, we evenly divide the spatial ranges $[-50m, 40m]$, $[-40m, 3m]$ and $[1m, 100m]$ into 256 $bins$, 128 $bins$ and 256 $bins$ in the log space, respectively. These spatial ranges are obtained from the statistics of the point clouds in the training set. The spatial histograms are computed based on these bins and then normalised by their L_1 norm. The two sets of features are concatenated to form a 643-dimensional feature G_{r_i} to encode the region's 3D geometric properties. The geometric features also can be efficiently computed through the hierarchy simply by weighted average-pooling of child region features as follows,

$$G_{r_{parent}} = \frac{\sum_{r_i \in children} G_{r_i} \times area(r_i)}{\sum_{r_i \in children} area(r_i)} . \tag{1}$$

3.3 Similarity Network

Given the segmentation hierarchy, we want to learn a merging process that generates a high-quality object candidate set from the initial oversegmentation.

To achieve this, we design and train a neural network to compute the affinity between two adjacent regions and use the network to merge region pairs recursively. Unlike the manually designed similarity scores used in [5, 22], our network enables us to learn a more effective merging criterion in the multimodal space.

Network Architecture Our similarity network takes a concatenation of feature descriptors from two adjacent regions as input and consists of three fully-connected layers. Each layer has 512 neurons and uses RELU as activation function except the last layer. We add the dropout layer after the first two layers to prevent overfitting. The output is the affinity score between two input regions in the range of $[0, 1]$ and indicates how likely two regions belong to the same object. We use the MatConvNet [32] to implement our networks in this work.

Network Training We obtain the training examples from the initial segmentation hierarchy. As we are learning a similarity network for object proposal generation, we expect that the network is able to output a high similarity score for two regions from the same object instance or the same background class, and to output a low score for two regions from different object instances. This network can be viewed as re-weighing the boundary strength between regions in the original UCM.

We formulate the network training as a binary classification problem. Each positive example is a pair of neighbouring regions overlapped with the same object instance and both regions have an overlap score larger than a threshold $t_{p1} = 0.7$. Here we define the overlap score as the intersection of a region and an object instance divided by the area of that region. We also take pairs overlapped with the same background class. In particular, we hope that regions belonging to the same background class around the object instance can merge together so that they do not interfere with the grouping of those regions from this object instance. To balance the positive examples from the object instance and from the background classes, we keep the proportion between them roughly at $1 : 2$.

For negative examples, we first take pairs of neighbouring regions in which one has an overlap with the object instance higher than $t_{p1} = 0.7$ while the other overlaps with the same object instance less than $t_n = 0.6$. Similar to the positive examples, we also include adjacent background region pairs which satisfy the same aforementioned condition. We keep the proportion of negative examples from the object instance and from the background classes at about $1 : 1$.

To mimic the process of grouping at test time, we scan regions from all levels of the segmentation hierarchy and obtain about $4, 120, 000$ positive and $3, 370, 000$ negative training examples. As there are two ways to concatenating features from two adjacent regions, we use both orders in training the network and the total number of training samples is doubled.

We train the similarity network to minimize the *log loss* using stochastic gradient descent with a batch size of $2, 000$ examples, momentum of 0.9, weight decay of 0.0005 and for 15 epochs. The learning rate we use for each epoch changes from 0.001 *to* 0.000001 evenly in the log space.

3.4 Hierarchical and Combinatorial Grouping

Given the region features in every level of the segmentation hierarchy and a learned similarity network, we generate a set of object proposals by a hierarchical and combinatorial grouping process.

Hierarchical Grouping We start from the initial regions in a single level of the segmentation hierarchy and re-group them by applying the similarity network. Specifically, we first compute the affinities between all adjacent regions via forwarding the feature descriptor of neighbouring regions through the similarity network. Then two most similar regions are merged into a new region and the descriptor for this new region is computed. This can be easily done by max-pooling (for appearance feature) or weighted average-pooling (for geometric feature) as described in Section 3.2. Next the affinities between this new region and its neighbours inherited from its child regions are updated using the similarity network. This merging process is repeated until the whole image becomes a single region. We apply this hierarchical grouping procedure to all five levels of the initial segmentation hierarchy, and take all single regions (region singletons) in the five new segmentation hierarchies as our initial set of object proposals.

Combinatorial Grouping Selecting the region singletons in the segmentation hierarchies only, however, is insufficient to generate a high quality pool of object proposals [9]. We follow a combinatorial grouping procedure similar to [9] to generate a larger object proposal set. In particular, we empirically select 10,000 region pairs, 10,000 region triplets and 5,000 region 4-tuples from every newly generated segmentation hierarchy to expand our object proposal pool, which performs well in our experiments. Figure 2 (right) shows an example of region singletons, pair and triplet in the segmentation hierarchy. We perform Non-Maximal Suppression (NMS) afterwards, which significantly reduces the number of candidates, since those region pairs, triplets and 4-tuples from the same segmentation hierarchy are heavily overlapped. The final pool of object proposals contains less than 10,000 proposals per image on average.

3.5 Ranking Network

In the final step, we want to estimate the quality of each object proposal, or its objectness score. This allows us to obtain good trade-off between the number and the quality of object proposals under various settings. We achieve this by training a ranking neural network to predict the IoU of each object proposal with the matched ground truth as in [8].

Network Architecture Our ranking network is a regression network, which has a similar architecture to the similarity network except the input and output layer. It also consists of three fully-connected layers and each layer has 512 neurons. The input is the feature descriptor of a single object proposal, which can be computed efficiently. The proposals defined by region singletons has their descriptors precomputed during the merging process. For those proposals formed by region pair, triplet or 4-tuple, their descriptor can be computed using the same max-pooling or average-pooling method described before. The output layer of the network is a linear layer that predicts the IoU between the input proposal and the corresponding ground truth. We use the the mean squared loss during network training. In the training stage, we also add a dropout layer after the first two layers to prevent overfitting.

Network Training We build the training dataset by choosing four types of training examples. The first type includes all the ground truths and the corresponding target IoUs are 1.0. The remaining training examples come from the object proposals generated on the training set. We split these object proposals into three categories according to their IoU to the ground truth:$IoU >= 0.5$, $0 < IoU < 0.5$ and $IoU = 0$. For the first category, we take all proposals in this group as training examples and denote its size as N. As to the latter two categories, we randomly select $3N$ and $3N$ examples from their pools respectively, which balances the training dataset. Finally, we obtain about 5,000,000 training examples in total.

We train the ranking network using stochastic gradient descent with a batch size of $2,000$ examples, momentum of 0.9, weight decay of 0.0005 and for 10 epochs. The learning rate we use for each epoch changes from 0.01 *to* 0.00001 evenly in the log space.

Diversifying the Ranking After assigning every proposal a ranking score, we diversify the ranking to reduce redundancy. Following [8], we achieve this based on Maximum Marginal Relevance measure, which is used to remove redundant object proposals. We apply the same re-ranking procedure as in [8] to lower the rank of the segment proposals that heavily overlap with higher-ranked proposals.

4 Experiments

In this section, we evaluate our multimodal object proposal generation approach on the publicly available Cityscapes dataset [16]. To the best of our knowledge, Cityscapes dataset is the only public dataset with stereo images and object instance segmentation ground truth, which are required by our method for quantitative evaluation.

Dataset Cityscpaes [16] is a newly released large-scale dataset for semantic urban scene understanding. It is comprised of a large diverse set of stereo video sequences recorded on streets from 50 different cities. $5,000$ of these images have high quality instance-level annotations for humans and vehicles and they are split into separate training ($2,975$ images), validation (500 images) and test ($1,525$ images) sets. This dataset is very challenging as it is biased towards busy

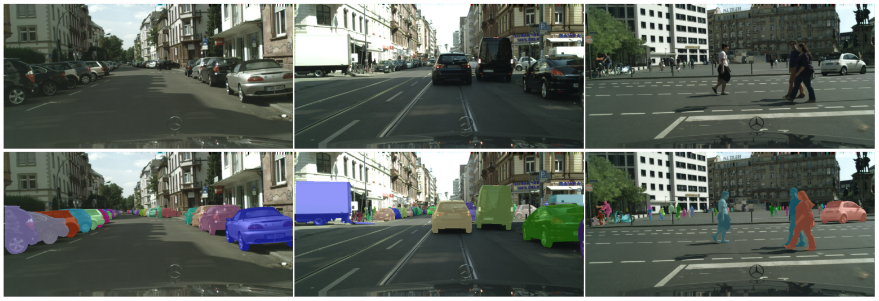

Fig. 3. Illustration of the Cityscapes dataset.**Top:** RGB images. **Bottom:** instance-level ground truth.

and cluttered scenes where many, often highly occluded, objects occur at various scales. Figure 3 shows some examples.

In our experiments, we further split the training set into two subsets: one for training (2,614 images) and the other for validation (361 images taken at Tubingen, Ulm and Zurich). We use their validation set (500 images) to evaluate the approaches, as the ground truth of the test set is withheld and their evaluation server does not provide results on proposal generation. The original image size is 1024 × 2048, which is too large to feed into the GPU memory when forwarding the image through the FCN-8s. So we downscale the original image by a factor of 4 into 512 × 1024. The dataset only provides instance-level annotations for humans (*person and rider*) and vehicles (*car*, *truck*, *bus*, *bicycle*, *motorbicycle*, *caravan and trailer*), which are considered as object proposal ground-truth in our experiments.

Evaluation Measures We employ the recall vs. number of proposals with a fixed IoU threshold and the average recall (AR) as the evaluation metrics. As discussed in Hosang et al.'s work [17], AR has been shown to have a strong correlation with the final detection performance. In our experiments, we compute the AR between IoU 0.5 to 1 and report AR vs. number of proposals.

Baseline and State-of-the-Art As we focus on object segmentation proposals generation, we mainly compare our approach (Ours-Depth-Seg) against two widely-used top-performing segmentation proposal generation methods: MCG [9] and SelectiveSearch [5], as well as our approach without geometric feature (Ours-NoDetph-Seg). In addition, we compare to the more recent 'Geodesic Object Proposals' (GOP) method [34](GOP(200,15) and GOP(140,4)), which has publicly available code.

We use the default parameters in MCG to generate the proposals. For Selective Search, we adopt the parameters used in RCNN [1], and keep the segmentation proposals instead of bounding boxes. The "Quality" version of Selective search (SeSe-Quality-60k) uses four different initial segmentations, five

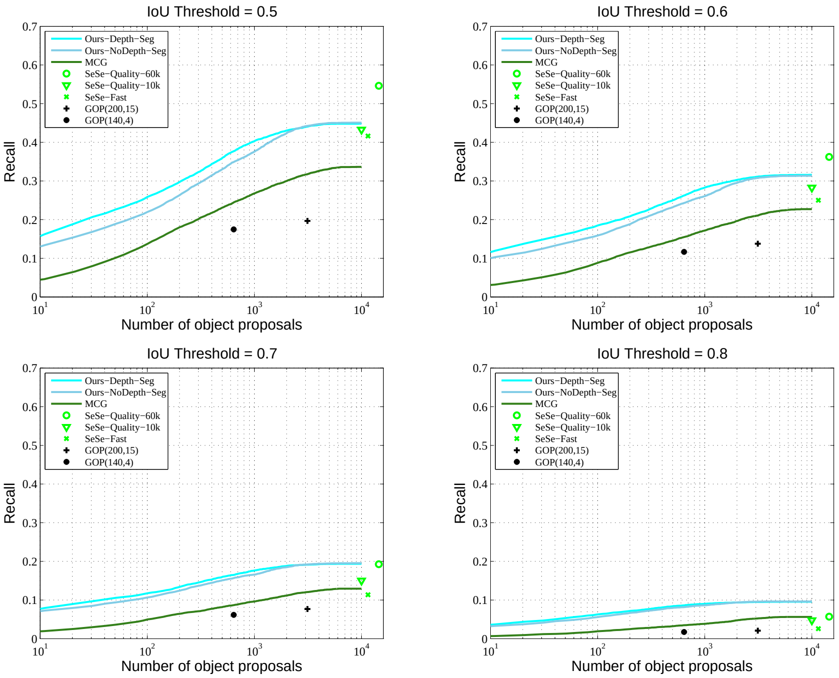

Fig. 4. Recall vs. number of proposals under different IoU thresholds.

color spaces and four similarity functions to diversify object proposals and over 60,000 proposals are generated per image on average. To make a fair comparison, we randomly select 10,000 proposals (SeSe-Quality-10k) from the SeSe-Quatlity-60k and evaluate their quality. We repeat this for 5 times and take the average results as their performance. The "Fast" version (SeSe-fast) uses only two different initial segmentations, two color spaces and two similarity functions for diversification and about 12,000 proposals on average are generated per image.

Furthermore, in order to demonstrate that our method can also generate high-quality bounding-box proposals, we conduct experiments to compare with the EdgeBoxes [7]. We use the tightest boxes enclosing our segmentation proposals as the output to evaluate our method.

Segmentation Results Figure 4 shows the recall rate when varying the number of object proposals under different IoU thresholds. We can see that our approach constantly and significantly outperforms MCG, SeSe-Quality-10k, SeSe-Fast and GOP. The recall of our approach attains 44.8% at about 5000 proposals when IoU threshold is 0.5, while MCG attains 33.6%, SeSe-Fast 41.7%, and SeSe-Quality-10k 44.0%. The performance of both versions of GOP is much lower than the above methods. With 1,000 proposals and IoU threshold as 0.5, the recall of our approach is above 40.0% while MCG just gets 26.8%. When the IoU threshold

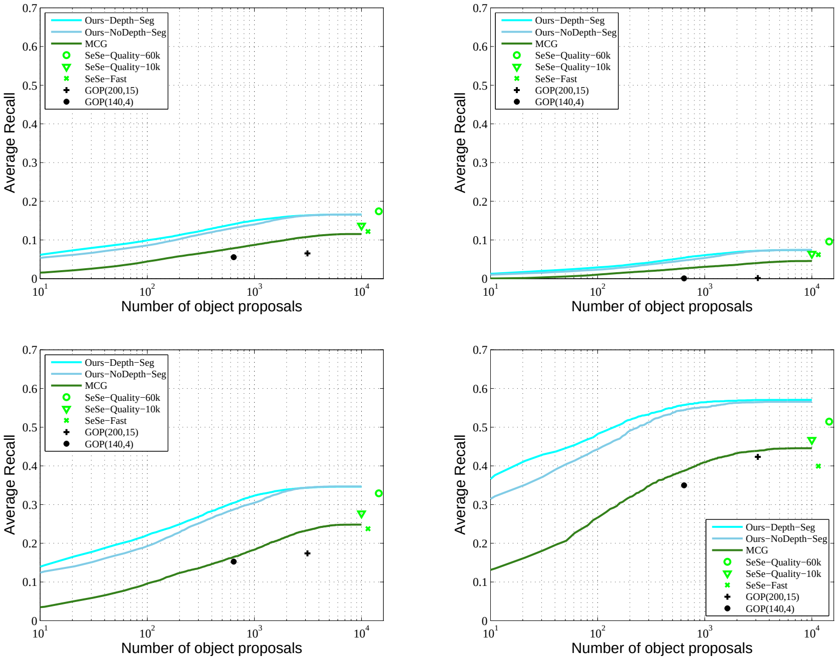

Fig. 5. AR vs. number of object proposals. **Top left:** Overall. **Top right:** Small objects. **Bottom left:** Medium objects. **Bottom right:** Large objects.

increases, we can see that the performance of Selective Search drops much faster than Ours and MCG, and particularly when the IoU threshold equals to 0.7, our method has a similar recall as SeSe-Quality-60k. This indicates that the quality of our proposals is better than Selective Search.

On the other hand, the performance of our approach using geometry information is always better than that without geometry information. Surprisingly, the upper bound of recall is not boosted by geometry information. This might be due to the noisy depth cues computed from the stereo images and that the geometric feature we manually designed is relatively weak. However, the ranking of proposals indeed benefits from the additional geometry information, as geometry information like the 3D height of a region is a good indicator of the objectness in street scenes.

Figure 5 (top left) describes the overall AR when changing the number of object proposals. It shows that our approach is significantly better than MCG, SeSe-Fast, SeSe-Quality-10k and GOP, but slightly worse than SeSe-Quality-60k which uses much more proposals. With 1,000 proposals, our method achieves an AR of 15%, while MCG just 8.7% and this number is consistent with the performance of instance segmentation task reported by Cordts et al. [16] who use MCG object proposals in their experiments.

Table 1. AR at different number of proposals(100, 1,000, 5,000 and total number of proposals(N)), overall AUC (AR averaged across all proposal counts) and also AUC at different scales (small, medium and large objects denoted by superscripts S,M and L).

Method	AR@100	AR@1000	AR@5000	AR@N	AUC	AUC^S	AUC^M	AUC^L
SeSe-Fast	-	-	-	0.122	0.106	0.052	0.206	0.358
SeSe-Quality-10k	-	-	-	0.137	0.108	0.047	0.221	0.402
SeSe-Quality-60k	-	-	-	**0.174**	0.145	**0.077**	0.278	0.451
MCG	0.045	0.087	0.113	0.115	0.107	0.041	0.229	0.432
GOP(200,15)	0.032	0.059	0.065	0.065	0.063	0.001	0.169	0.406
GOP(140,4)	0.032	0.056	0.056	0.056	0.055	0.001	0.151	0.344
Ours-noDepth-Seg	0.086	0.140	0.165	0.166	0.159	0.069	0.335	0.559
Ours-Depth-Seg	**0.099**	**0.150**	**0.165**	0.166	**0.160**	0.070	**0.337**	**0.566**

Following [12], we also report the AR vs. the number of object proposals at different object scales, as the size of object in Cityscapes dataset varies in a quite wide range. We split the ground truth into three sets according to object pixel area a: small ($a < 32^2$), medium ($32^2 \leq a \leq 96^2$) and large ($a \geq 96^2$). Figure 5 describes the performance at each scale. All methods perform poorly on small objects (top right), which leads to the low overall AR. By contrast, when it comes to the categories of medium (Bottom left) and large (Bottom right) object, the AR by all approaches has a considerable increase and our method performs substantially better than MCG, SeSe-Fast, SeSe-Quality-10k and GOP, and also slightly better than SeSe-Quality-60k.

More detailed quantitative results are shown in Table 1, which reports the AR at selected proposal numbers and the averaged overall AR across all proposal numbers (AUC), as well as AUC at different object scales. Finally, examples of generated proposals with the highest IoU to the ground truth on selected images are shown in Figure 6.

Bounding-Boxes Results We also compare our method against bounding-box proposal generation method, the EdgeBoxes [7], using the metric of AR. Figure 7 shows that our approach generate much better bounding-box proposals than the EdgeBoxes. With 1,000 proposals, our approach gets an AR of 27.3%, which is over 2.5× higher than the EdgeBoxes' (10.5%). The upper bound of our method (31.2%) is also much higher than the EdgeBoxes's (25.0%).

5 Conclusion

In this paper, we propose a learning-based object segmentation proposal generation method for stereo images, which exploits both deep features and the depth cue. We extract features from convolutional feature maps and geometry maps to describe a region. We learn a similarity network to estimate the affinity between two adjacent regions, sequentially merge regions from a segmentation

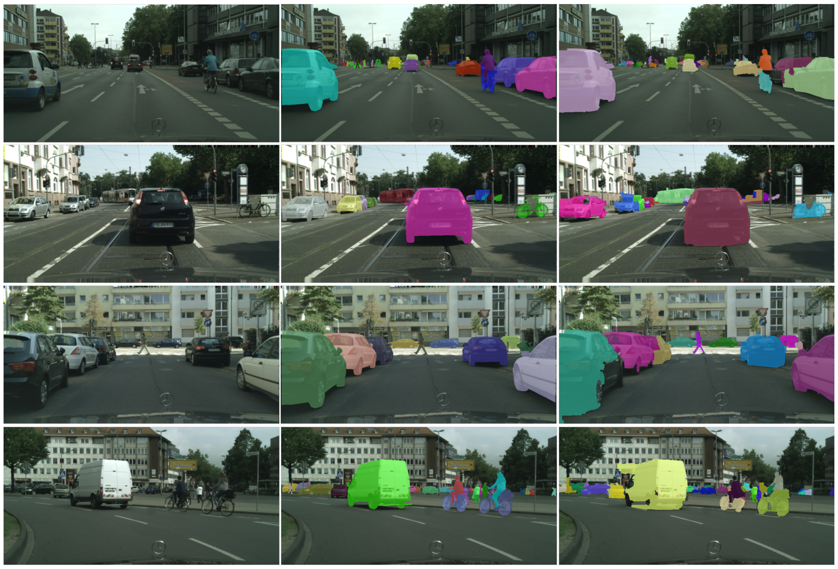

Fig. 6. Qualitative examples of our object proposals. **Left:** RGB images. **Middle:** Ground truth. **Right:** Our best proposals.

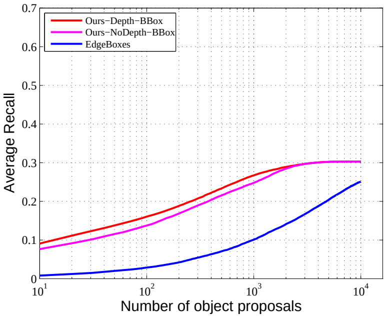

Fig. 7. AR vs. number of object proposals for Bounding Box proposals.

hierarchy based on the affinity to generate object proposals and learn a ranking network to predict the objectness of a proposal. Experiments on the Cityscapes dataset show that our approach achieves much better average recall than the state-of-the-art and depth cue can improve the ranking of proposals.

Acknowledgment Data61 is part of the Commonwealth Scientific and Industrial Research Organisation (CSIRO) which is the federal government agency for scientific research in Australia. The Tesla K40 used for this research was donated by the NVIDIA Corporation.

References

1. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Region-based convolutional networks for accurate object detection and segmentation. Pattern Analysis and Machine Intelligence, IEEE Transactions on (2015)
2. Girshick, R.B.: Fast R-CNN. CoRR **abs/1504.08083** (2015)
3. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems. (2015) 91–99
4. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE (2010) 73–80
5. Uijlings, J.R., van de Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. International journal of computer vision **104** (2013) 154–171
6. Cheng, M.M., Zhang, Z., Lin, W.Y., Torr, P.: Bing: Binarized normed gradients for objectness estimation at 300fps. In: IEEE CVPR. (2014)
7. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: Computer Vision–ECCV 2014. Springer (2014) 391–405
8. Carreira, J., Sminchisescu, C.: Cpmc: Automatic object segmentation using constrained parametric min-cuts. Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012) 1312–1328
9. Pont-Tuset, J., Arbeláez, P., Barron, J., Marques, F., Malik, J.: Multiscale combinatorial grouping for image segmentation and object proposal generation. In: arXiv:1503.00848. (2015)
10. Endres, I., Hoiem, D.: Category independent object proposals. In: Computer Vision–ECCV 2010. Springer (2010) 575–588
11. Krähenbühl, P., Koltun, V.: Learning to propose objects. In: CVPR. (2015)
12. Pinheiro, P.O., Collobert, R., Dollar, P.: Learning to segment object candidates. In: Advances in Neural Information Processing Systems. (2015) 1981–1989
13. Dai, J., He, K., Sun, J.: Instance-aware semantic segmentation via multi-task network cascades. arXiv preprint arXiv:1512.04412 (2015)
14. Bleyer, M., Rhemann, C., Rother, C.: Extracting 3d scene-consistent object proposals and depth from stereo images. In: Computer Vision–ECCV 2012. Springer (2012) 467–481
15. Chen, X., Kundu, K., Zhu, Y., Berneshawi, A., Ma, H., Fidler, S., Urtasun, R.: 3d object proposals for accurate object class detection. In: NIPS. (2015)
16. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. arXiv preprint arXiv:1604.01685 (2016)
17. Hosang, J., Benenson, R., Dollár, P., Schiele, B.: What makes for effective detection proposals? arXiv preprint arXiv:1502.05082 (2015)
18. Humayun, A., Li, F., Rehg, J.M.: The middle child problem: Revisiting parametric min-cut and seeds for object proposals. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1600–1608
19. Lee, T., Fidler, S., Dickinson, S.: Learning to combine mid-level cues for object proposal generation. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 1680–1688
20. Wang, C., Zhao, L., Liang, S., Zhang, L., Jia, J., Wei, Y.: Object proposal by multi-branch hierarchical segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3873–3881

21. Rantalankila, P., Kannala, J., Rahtu, E.: Generating object segmentation proposals using global and local search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 2417–2424
22. Yanulevskaya, V., Uijlings, J., Sebe, N.: Learning to group objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2014) 3134–3141
23. Kuo, W., Hariharan, B., Malik, J.: Deepbox: Learning objectness with convolutional networks. arXiv preprint arXiv:1505.02146 (2015)
24. Ghodrati, A., Diba, A., Pedersoli, M., Tuytelaars, T., Van Gool, L.: Deepproposal: Hunting objects by cascading deep convolutional layers. In: Proceedings of the IEEE International Conference on Computer Vision. (2015) 2578–2586
25. Sharma, A., Tuzel, O., Liu, M.Y.: Recursive context propagation network for semantic scene labeling. In: Advances in Neural Information Processing Systems. (2014) 2447–2455
26. Zagoruyko, S., Komodakis, N.: Learning to compare image patches via convolutional neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 4353–4361
27. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3431–3440
28. Hariharan, B., Arbeláez, P., Girshick, R., Malik, J.: Hypercolumns for object segmentation and fine-grained localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 447–456
29. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2014)
30. Dai, J., He, K., Sun, J.: Convolutional feature masking for joint object and stuff segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2015) 3992–4000
31. Dollár, P., Zitnick, C.L.: Fast edge detection using structured forests. Pattern Analysis and Machine Intelligence, IEEE Transactions on **37** (2015) 1558–1570
32. Vedaldi, A., Lenc, K.: Matconvnet: Convolutional neural networks for matlab. In: Proceedings of the 23rd Annual ACM Conference on Multimedia Conference, ACM (2015) 689–692
33. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: Computer Vision–ECCV 2014. Springer (2014) 756–771
34. Krähenbühl, P., Koltun, V.: Geodesic object proposals. In: Computer Vision–ECCV 2014. Springer (2014) 725–739