Signal Processing: Image Communication 53 (2017) 110–122

Contents lists available at ScienceDirect

Signal Processing: Image Communication

journal homepage: www.elsevier.com/locate/image

Robust object proposals re-ranking for object detection in autonomous driving using convolutional neural networks

Cuong Cao Pham, Jae Wook Jeon*

College of Information and Communication Engineering, Sungkyunkwan University, Suwon, Republic of Korea

ARTICLE INFO

Keywords:
Object proposals
Autonomous driving
Object detection
Convolutional neural networks
Stereo vision.

ABSTRACT

Object proposals have recently emerged as an essential cornerstone for object detection. The current state-of-the-art object detectors employ object proposals to detect objects within a modest set of candidate bounding box proposals instead of exhaustively searching across an image using the sliding window approach. However, achieving high recall and good localization with few proposals is still a challenging problem. The challenge becomes even more difficult in the context of autonomous driving, in which small objects, occlusion, shadows, and reflections usually occur. In this paper, we present a robust object proposals re-ranking algorithm that effectivity re-ranks candidates generated from a customized class-independent 3DOP (3D Object Proposals) method using a two-stream convolutional neural network (CNN). The goal is to ensure that those proposals that accurately cover the desired objects are amongst the few top-ranked candidates. The proposed algorithm, which we call DeepStereoOP, exploits not only RGB images as in the conventional CNN architecture, but also depth features including disparity map and distance to the ground. Experiments show that the proposed algorithm outperforms all existing object proposal algorithms on the challenging KITTI benchmark in terms of both recall and localization. Furthermore, the combination of DeepStereoOP and Fast R-CNN achieves one of the best detection results of all three KITTI object classes.

1. Introduction

Developing autonomous driving systems that can assist drivers in making decisions is one of the most active and challenging research areas [1]. The goal is to improve safety, reduce traffic accidents, and move closer towards fully autonomous cars and intelligent transportation systems. Among the solutions that have been developed over the past few years, the computer vision-based approach offers the most cost effective solution as it uses cameras rather than other types of more costly sensors. In this paper, we focus on object detection in autonomous driving.

State-of-the-art object detectors employ the exhaustive "sliding window" paradigm [2–4], in which a large number of bounding boxes generated from various scales and ratios is used for classification. This approach has been widely used as a standard object detection framework for many years. Recently, the use of more sophisticated and powerful classifiers [5–11] has improved detection accuracy. However, given the significant increase of computation time per window, the use of the "sliding window" paradigm has become infeasible. An alternative approach with the use of "object proposals" has been successfully introduced to gain both computational efficiency and high detection accuracy [12–15]. The key idea is to generate a moderate set of candidate bounding box proposals that are likely to contain objects and use this set for further classification instead of searching for objects at every image location and scale. This approach facilitates the ease of detection and can also improve accuracy by pruning away false positives before classification.

Since its remarkable discovery [12–15], the development of object proposals has quickly evolved since many methods have been innovated and improved. The goal is to introduce an algorithm that is able to achieve high recall and good localization. Extensive survey and evaluation can be found in [16,17]. In this paper, we briefly outline the existing work and review current methods not covered in [16,17]. Existing methods can be categorized into two main approaches according to their strategy used for generating proposals: grouping and scoring [17]. Grouping approaches typically generate multiple segments that are likely to contain objects by merging similar small regions based on diverse cues. On the other hand, scoring approaches tend to be faster by first initializing a set of bounding box proposals, and then scoring each proposal using an objectness function.

High recall and good localization are the most important properties of an object proposals algorithm [18,19,17]. A robust generator must be

* Corresponding author.
E-mail addresses: cuongpc@skku.edu (C.C. Pham), jwjeon@yurim.skku.ac.kr (J.W. Jeon).

http://dx.doi.org/10.1016/j.image.2017.02.007
Received 12 November 2016; Received in revised form 26 January 2017; Accepted 16 February 2017
Available online 20 February 2017

able to obtain high recall across various intersection over union (IoU) thresholds using a modest number of proposals, ranging from hundreds to a few thousands per image. While each existing algorithm has its own strengths and weaknesses, all algorithms have two common limitations. First, the proposals they produce are not well-localized since the recall drops significantly as the IoU threshold increases, especially for scoring approaches. Second, they are not able to preserve recall, which is relatively high for a large number of proposals but low for a small number of proposals.

Regarding autonomous driving, in which KITTI [1] represents the state-of-the-art benchmark, none of the existing algorithms work well until recently [19]. The existing algorithms not only require a very large number of proposals in order to achieve reasonable recall, but the recall also drops dramatically. Apart from the aforementioned limitations, the lack of success of existing algorithms can also be explained by the fact that KITTI images are more challenging, since they contain small objects, occluded objects, reflections, and shadows [1,19].

Chen et al. [19] first tackle object proposals for autonomous driving by introducing the class-specific 3DOP method, which is able to obtain high recall across various IoU thresholds. Notably, the generated proposals are also well localized since the recall drops gradually and slowly. While 3DOP is the best object proposal generator to date, its disadvantages are twofold. First, it must be run seperately with regard to each object class in order to obtain high recall, which is not efficient. This class-specific property leads to an increase in the processing time of the classification stage, in which the number of needed proposals is linearly increased with respect to the number of object classes. For example, Chen et al. [19] used a total of 6000 proposals for three object classes, while our class-independent approach achieves slightly better accuracy with only 2000 proposals. In fact, generating object proposals is usually referred to as a class-independent task [12,14,17], which measures the likelihood of a window containing an object without considering its class. Second, we observe that even though the results of 3DOP with around 10,000 proposals achieve very high recall, its top-ranked proposals with fewer candidates do not preserve recall effectively. This is because the Markov Random Field (MRF) ranking model of 3DOP is not very robust and because 3DOP only uses depth features, while visual RGB features are not exploited.

In this paper, we present a robust object proposals re-ranking algorithm for autonomous driving using convolutional neural networks (CNN) [20]. Considering the robustness of 3DOP in generating well-localized proposals and the success of deep learning in the last few years, we aim to present a learning algorithm that is able to overcome the aforementioned limitations of 3DOP. The goal is to introduce a class-independent algorithm that is able to achieve high recall and good localization with few candidates. Specifically, we propose a lightweight two-stream CNN that exploits both RGB features and depth features to re-rank proposals, which are generated from a customized class-independent 3DOP. Here, the depth features include disparity map and distance to the ground, which can be computed from a stereo image pair. We call our algorithm DeepStereoOP. Fig. 1 shows the block diagram of the proposed approach. The experiments show the effectiveness of DeepStereoOP, achieving the highest recall across all IoU thresholds and occlusion levels. Ultimately, the combination of DeepStereoOP and the state-of-the-art Fast R-CNN object detector [11,19] achieves one of the best detection results of all three KITTI object classes.

The remainder of this paper is structured as follows. Section 2 presents related work including existing object proposal algorithms, recent improvements, and top-performing object detectors. Section 3 presents the proposed approach, while Section 3.1 presents the experimental results with KITTI dataset to compare our approach to those in the literature. Section 3.2 concludes the paper.

2. Related work

Object detection has undergone a fundamental shift from the traditional sliding window paradigm to the object proposals approach. Therefore, instead of searching for objects at every image location and scale, the classifier just focuses on a set of candidate bounding boxes generated from an object proposals algorithm. Notable pioneer works that shape this research include Objectness [12], CPMC [13], Endres [14], and SelectiveSearch [15]. Subsequent to such pioneering work, many novel algorithms ranging from objectness-based scoring to similarity-based grouping as well as supervised learning and other improvements have also been introduced.

Regarding scoring approaches, Alexe et al. [12,21] first proposed Objectness by sampling an initial set of proposals according to a saliency map, and ranking them using several objectness cues such as saliency, contrast, edge density, and superpixel straddling. Rahtu et al. [22] and Zhang et al. [23] then extended Alexe et al.'s work using structured output ranking and cascaded ranking SVMs, respectively. Cheng et al. [24] introduced BING, which uses a simple linear classifier with the learned normed gradient features to rank proposals generated from sliding windows. Although BING is very fast, its localization is cursory due to the weak discrimination of simple gradient features [25]. Zhang et al. [26] then introduced BING++ to overcome this weakness. Recently, Zitnick et al. [18] proposed EdgeBoxes, which rapidly scores millions of windows by measuring the relationship between the contours completely enclosed within the box and those overlapping the box's boundary. Similar to BING, EdgeBoxes suffers from localization bias [17]. Lu et al. [27] proposed ContourBox to reject proposals that do not have explicit closed contours. Kuo et al. [28] introduced DeepBox, which re-ranks proposals generated from EdgeBoxes using CNN, so that it can achieve better recall with fewer proposals. However, the localization bias issue remains. In summary, the main limitation of these scoring approaches is their strong localization bias; while they achieve high recall at a low IoU threshold, the high recall is barely maintained as the IoU threshold increases. Chen et al. [29] proposed a refinement approach to reduce this bias using multi-thresholding straddling expansion, which erodes and dilates bounding boxes based on superpixels tightness. However, the localization bias issue was not completely resolved.

Compared to scoring methods, grouping methods typically obtain

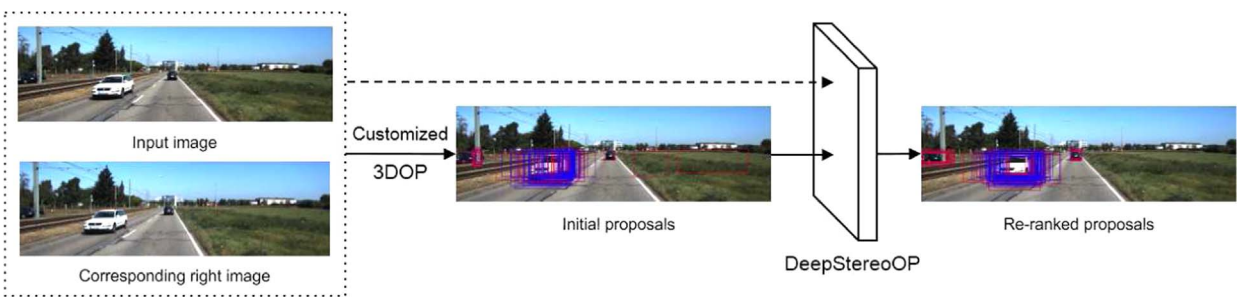

Fig. 1. Block diagram of the proposed object proposals re-ranking approach. Initial proposals with around 10,000 candidates are first generated using a customized class-independent 3DOP, and are then re-ranked using our proposed DeepStereoOP network.

better localization at the expense of increasing computation time. Carreira et al. [13,30] solved a constrained parametric min-cuts (CPMC) problem with several foreground and background seeds to generate proposals that are ranked using Gestalt-like features. Later, Rigor et al. [31] accelerated the computation of CPMC by reusing computation across multiple parametric min-cuts. Endres [14,32] also solves graph-cuts with different seed regions based on hierarchical occlusion boundaries segmentation. Notably, SelectiveSearch [15,33] is the most well-known grouping method and has been widely used in recent top-performing object detectors [9,11,34]. Specifically, it performs hierarchical segmentations in multiple color spaces using Felzenszwalb et al.'s algorithm [35], and greedily merges adjacent superpixels to generate proposals according to low-level features. Manén et al. [36] extended this idea with RandomizedPrim that randomly merges adjacent superpixels with learned probabilities. Rantalankila et al. [37] used different features for superpixel merging and graph-cuts to generate proposals. MCG [38] also uses hierarchical segmentations, but it merges segments based on combinatorial grouping. GOP [39] computes geodesic distance transforms from foreground-background masks and identifies critical level sets as proposals. While most of grouping methods also include a ranking model, the model is not sufficient and requires more reasoning to select top k proposals [28,17]. Some methods do not even provide ranked proposals [37,31,39], while others only provide random scores [36,29]. This ranking issue leads to ineffectiveness in preserving proposal recall when considering a small number of proposals.

Another approach that uses supervised learning has also emerged recently. Krahenbuhl et al. [40] proposed LPO, which trains conditional random field models to generate proposals. Lee et al. [41] introduced a novel structured learning model called parametric min-loss together with a set of mid-level grouping cues to accomplish the task. Multibox [7,42] trains a sophisticated CNN to directly regress a fixed number of ranked proposals. DeepProposal [43] searches for proposals in a sliding window fashion using a coarse-to-fine cascade on multiple layers of CNN features. DeepMask [44] and region proposal networks (RPN) [45] train very deep networks to simultaneously generate proposals and predict scores. Despite using deep networks, these methods are just able to achieve high recall under loose overlap criteria with small IoU thresholds, but not under strict overlap criteria [46]. In contrast, the proposed DeepStereoOP only uses a lightweight two-stream CNN to re-rank proposals generated from a customized class-independent 3DOP, and achieves high recall across various IoU thresholds. In addition, our DeepStereoOP provides 3D object proposals, which is another significant advantage over these methods.

Regarding object proposals for autonomous driving, which is the main focus of this work, Chen et al. [19] reported that none of the existing algorithms works well with the KITTI dataset. To this end, they introduced the class-specific 3DOP that is able to achieve high recall across various IoU thresholds. Specifically, they first sampled a large number of object proposals using 3D object size priors, and then formulated an MRF energy minimization problem encoding depth features to rank candidate proposals. However, their approach has two drawbacks as mentioned earlier. First, 3DOP needs to be run with regard to each object class in order to obtain high recall. Consequently, the number of proposals used for the classification stage is also linearly increased with respect to the number of object classes. Second, the MRF ranking model is not very robust since top-ranked proposals with few hundreds candidates are not able to preserve initial recall of around ten thousands candidates, resulting in missed objects, which cannot be recovered in the subsequent classification stage. Parallel to our work, Chen et al. [47] introduced another algorithm called Mono3D, but the results were only comparable to those of 3DOP. Moreover, Mono3D is still a class-specific approach, while generating object proposals is usually referred to as a class-independent task [12,14,17].

Given these circumstances, we introduce DeepStereoOP to overcome the aforementioned limitations of 3DOP. Specifically, the proposed two-stream CNN architecture encodes both RGB visual features and depth features to re-rank proposals generated from a customized class-independent 3DOP. We compute disparity map using the SPS stereo matching algorithm [48] and estimate distance to the group map using Chen et al.'s method [19]. In fact, object detection and distance estimation using stereo matching are usually coupled in most autonomous driving systems, so that additional computation of depth features is not considered as a computational disadvantage compared to monocular algorithms. Experiments show the robustness of our approach. Above all, no class-independent object proposals algorithm for autonomous driving currently exists that is able to achieve high recall and good localization across all IoU thresholds. The key contribution of this paper is to present such an approach.

Related to object detection for autonomous driving, a great number of vehicle, pedestrian, and obstacle detection algorithms have been proposed. Extensive reviews and evaluations of the state-of-the-art methods can be found in [49–54], while the KITTI website [1] gives a review of the latest methods. The deformable parts model (DPM) [4] is the most well-known algorithm, and many variants have been built on this model to improve its performance [55,54,56,57]. However, many novel algorithms have recently surpassed DPM. According to the KITTI benchmark, the Regionlets [34], 3DVP [58], and Fast R-CNN based detectors [19,45,47] are currently the top-performing algorithms. Ultimately, our combination of DeepStereoOP and Fast R-CNN [11,19] achieves one of the best detection results of all three KITTI object classes.

3. Proposed approach

3.1. Generating initial proposals

The proposed object detection pipeline consists of three steps: generating initial proposals, re-ranking, and detection. Here, a customized class-independent 3DOP is employed to carry out the first step. To begin, we first express each 3D bounding box as $\mathbf{y} = (x, y, z, \alpha, t)$, where (x, y, z) represents the center of the 3D box, α is the azimuth angle, and t denotes a set of 3D size prior templates learned from the training data. Chen et al. [19] claim that the proposal $\mathbf{y}$ should contain a high density of occupied voxels in the point cloud $\mathbf{x}$ and should not overlap with the free space. Here, $\mathbf{x}$ is computed from depth map, which can be generated by a stereo matching algorithm. In this work, we use the SPS algorithm [48] for consistent with [19]. Other stereo matching algorithms [59–61] are also applicable. Apart from the point cloud density and free space criterions, the height of the point cloud inside $\mathbf{y}$ should be close to the average height learned from the data and higher than that of the point cloud surrounding $\mathbf{y}$. Given these constraints, the MRF energy is expressed as follows

$$E(\mathbf{x}, \mathbf{y}) = \mathbf{w}_{pcd}^T \rho_{pcd}(\mathbf{x}, \mathbf{y}) + \mathbf{w}_{fs}^T \rho_{fs}(\mathbf{x}, \mathbf{y}) + \mathbf{w}_{hp}^T \rho_{hp}(\mathbf{x}, \mathbf{y}) + \mathbf{w}_{hc}^T \rho_{hc}(\mathbf{x}, \mathbf{y}) \tag{1}$$

where the weights $\mathbf{w}^T$ are trained using structured SVM [62]. $\rho_{pcd}(\mathbf{x}, \mathbf{y})$, $\rho_{fs}(\mathbf{x}, \mathbf{y})$, $\rho_{hp}(\mathbf{x}, \mathbf{y})$, and $\rho_{hc}(\mathbf{x}, \mathbf{y})$ represent the point cloud density, free space, height prior, and height contrast criterions, respectively. Specifically, the point cloud density criterion computes the ratio of occupied voxels inside the 3D box, and is formally expressed by

$$\rho_{pcd}(\mathbf{x}, \mathbf{y}) = \frac{1}{|\Omega(\mathbf{y})|} \sum_{v \in \Omega(\mathbf{y})} S(v) \tag{2}$$

where $S(v)$ determines whether or not $\mathbf{y}$ contains the voxel v, and $|\Omega(\mathbf{y})|$ denotes the size of the 3D box. In a similar manner, the free space constraint is defined by

$$\rho_{fs}(\mathbf{x}, \mathbf{y}) = \frac{1}{|\Omega(\mathbf{y})|} \sum_{v \in \Omega(\mathbf{y})} F(v) \tag{3}$$

where $F(v)$ determines whether or not the ray from the camera to the

voxel v hit an occupied voxel. The aim of this constraint is to minimize the number free space voxels appearing inside the box. Next, the height prior constraint is encoded as

$$\rho_{hp}(\mathbf{x}, \mathbf{y}) = \frac{1}{|\Omega(\mathbf{y})|} \sum_{v \in \Omega(\mathbf{y})} H(v) \tag{4}$$

with

$$H(v) = \begin{cases} \exp\left[-\frac{1}{2}\left(\frac{h_v - \mu_h}{\sigma_h}\right)^2\right] & S(v) = 1 \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

where h_v denotes the height of the voxel v with respect to the road plane. μ_h and σ_h are the mean height and standard deviation estimated from the training data, respectively. Finally, the height contrast criterion is computed by

$$\rho_{hc}(\mathbf{x}, \mathbf{y}) = \frac{\rho_{hp}(\mathbf{x}, \mathbf{y})}{\rho_{hp}(\mathbf{x}, \mathbf{y}^+) - \rho_{hp}(\mathbf{x}, \mathbf{y})} \tag{6}$$

where $\mathbf{y}^+$ is the enlarged version of $\mathbf{y}$, which is generated by expanding each direction of $\mathbf{y}$ by 0.6 m.

After sampling and scoring all $\mathbf{y}$ using (1), we arrange them with respect to $E(\mathbf{x}, \mathbf{y})$ and perform non-maximal suppression (NMS) with IoU = 0.8 to render the initial output. In [19], Chen et al. used various IoU thresholds with regard to each object class to carry out NMS with the desired number of output proposals of around few thousands candidates. For their class-independent version, which is called 3DOP-generic [19], IoU is set to 0.7 to obtain around 3500 candidates. On the other hand, we set IoU = 0.8 to obtain around 10,000 proposals and re-rank them again using DeepStereoOP before selecting top-ranked candidates for further detection and classification. We empirically found that our customized version has a higher potential to obtain high recall and good localization accuracy after re-ranking compared to 3DOP-generic.

3.2. Re-ranking proposals using deepstereoop

3.2.1. RGB and depth features

After performing the first step, we obtain the initial proposals as well as the estimated disparity map and the corresponding height map or distance to the ground map. We further utilize these two depth features to re-rank the initial proposals together with the input RGB image. Since RGB features have been widely used for CNN, the addition of these depth features can provide supplementary geometric information to render the learning more robust. For example, the disparity map can provide hints about object boundaries and compactness, while we can reduce the scores of proposals that include road and sky regions according to the distance to the ground map. Here, we truncate the height to 10 m, and linearly scale its values to the range [0, 255].

3.2.2. Fast network architecture

From a computational point of view, the time taken for generating and re-ranking proposals must be faster than the time taken for the classifier itself. Therefore, the network architecture for this re-ranking module must be lightweight and the time for inference must be fast. We thus select the reduced four-layer version of CaffeNet [11] (i.e. AlexNet [5]) presented in [28] as a starting point. We extend this basic network by adding a depth stream to encode the disparity map and distance to the ground. Fig. 2 shows an overview of the proposed DeepStereoOP architecture.

The first two convolutional layers of the RGB stream are denoted as `conv1` and `conv2`, while the first two convolutional layers of the depth stream are denoted as `conv1_depth` and `conv2_depth`. In the spirit of spatial pyramid pooling [10], a ROI pooling layer [11] with a fixed spatial extent of 6×6 is performed at the end of each stream to obtain proposal-specific features, and features output from two streams are concatenated and fed to the fully connected layer `fc3`. The number of feature maps, kernel size, and stride of `conv1` and `conv1_depth` are set to (96, 11, 4), while `conv2` and `conv2_depth` are set to (256, 5, 1). The number of outputs of `fc3` is 1024. All the aforementioned layers are followed by a ReLU activation function. We also use Dropout regularization [63] with the dropout ratio of 0.5 at `fc3` to avoid overfitting. The final fully connected layer `fc4` embeds only two units representing the object and background probabilities, which are scored using a softmax layer.

3.2.3. Initialization and training

We initialize the weights of `conv1` and `conv2` with the weights of the CaffeNet model [5,11], while the weights of `conv1_depth`, `conv2_depth`, and fully connected layers are initialized randomly from Gaussian distributions. Regarding training samples, a proposal is considered to be a foreground window when its overlap with the ground truth box is higher than 0.7. On the other hand, when the overlap between the proposal and the ground truth box is lower than 0.3, we consider it as a background window. Other proposals with overlap in the range [0.3, 0.7] are ignored during training. This strategy is applied to all bounding box proposals without considering object class.

While upscaling the input image by a factor of 3.5 has been recommended for classification task [19], using very large image increases the computation time while the inference must be fast as stated above. Therefore, we only upscale the input image by a factor of 1.6 at testing in order to trade off between accuracy and computational performance. That said, the size the shortest side of an image is scaled to 600 regarding KITTI dataset. We train the network for 150,000 iterations with a mini-batch size of 128 using three scales of [400, 600, 900]. The initial learning rate is set at 0.001, and we drop it by a factor of 10 every 40,000 iterations. The momentum is set to 0.9, while the weight decay is 0.0005.

The final proposals are then sorted with respect to the new scores, and we can select top-ranked proposals for further detection and classification. It is worth noting that the proposals obtained at this stage contain both 2D and corresponding 3D bounding boxes.

3.3. Detecting objects

The top-ranked proposals generated from the second step are further scored using a robust classifier to detect and classify objects. As in the case of [19,47], we use the network built on Fast R-CNN [11] to accomplish the task. The overall architecture of the network is illustrated in Fig. 3. Intuitively, the network first computes the convolutional features from the entire image and then feeds them into two branches, which learn features from the bounding box proposals and from the contextual boxes. Here, the contextual boxes are generated by expanding the corresponding proposals by a factor of 1.5 as in [64]. Both the proposal-specific and the context-specific features are computed using ROI pooling layers, and two fully connected layers are subsequently added in each branch.

The output features from the two branches are finally concatenated and fed to three sibling output layers, which jointly learn object class probabilities, bounding box regression offsets, and object orientation for the foreground boxes using three equal weighted losses. The losses for object class classification and bounding box regression are similar to those used in Fast R-CNN [11], while smooth L_1 loss is used for the orientation regression. For background boxes, only the loss for object class classification is examined.

For initializing the network parameters, we use the VGG16 model [8] trained on ImageNet as in [11,19,47]. More precisely, we initialize the weights of the convolutional layers and the proposal branch with the weights of VGG16, and copy the weights of the proposal branch to initialize the weights of the corresponding context branch. We carry out

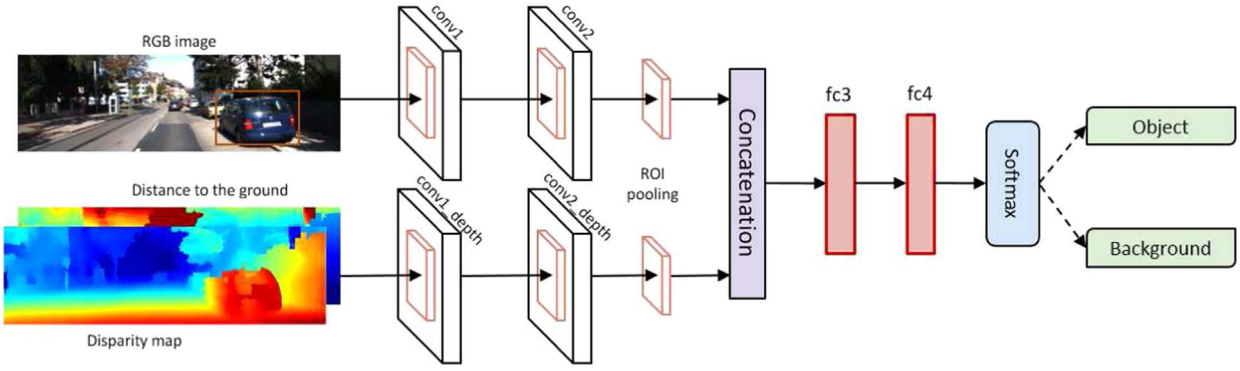

Fig. 2. Proposed DeepStereoOP architecture with a two-stream CNN using RGB and depth features. The output consist of two units encoding objectness and background scores of the proposal **y**. The larger the objectness score is, the higher the probability that **y** will cover an object.

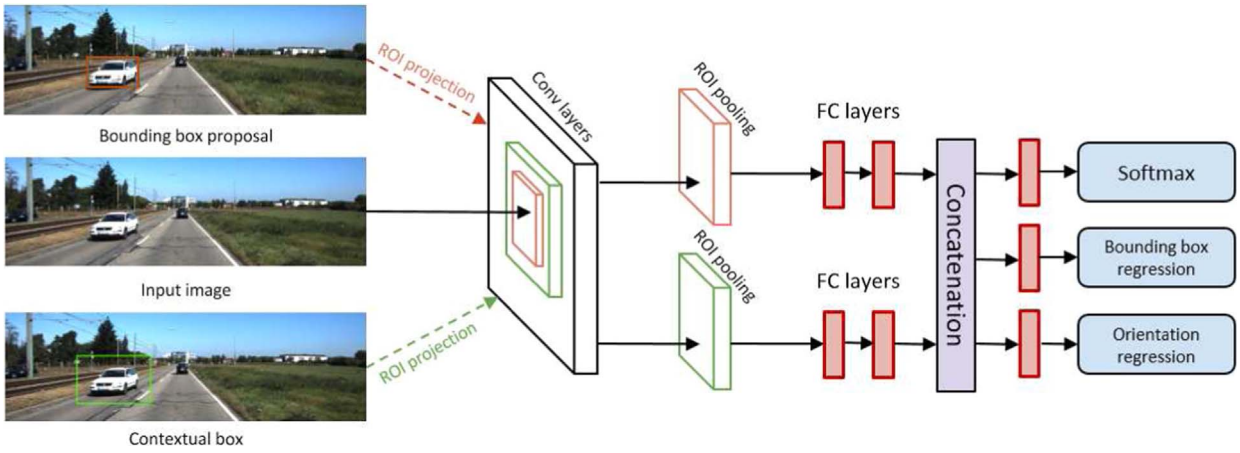

Fig. 3. Fast R-CNN architecture for object detection and orientation estimation adopted from [19]. The left image, 2D objectproposals, and 3D orientation information are used as inputs.

Fig. 4. Qualitative 2D object proposals results of the proposed DeepStereoOP on the validation set. Top scoring proposals are marked in descending order from blue to red.

both training and testing steps in a single scale fashion, in which the input image is upscaled by a factor of 3.5 to handle small objects. This means that the size of the shortest side of an image is 1295, considering the KITTI dataset. We train the network for 40,000 iterations as in [19,47] with the default settings, in which the mini-batch size is set to 128, momentum to 0.9, and weight decay to 0.0005. The base learning rate is 0.001 and we decrease to 0.0001 after the first 30,000 iterations. The KITTI overlap criteria, in which the IoU threshold is set at 0.7 for *Car* and 0.5 for *Pedestrian* and *Cyclist*, is employed to determine positive samples, while proposals with an IoU lower than these thresholds are considered as negative samples.

Ultimately, we only need the top 2000 proposals for testing in all three object classes. This shows the key advantage of our proposed approach compared to [19,47], in which a total of 6000 proposals was used, and this number can be further increased when extending the number of object classes.

Both DeepStereoOP and Fast R-CNN are implemented in Python and C++ using Caffe framework [65].

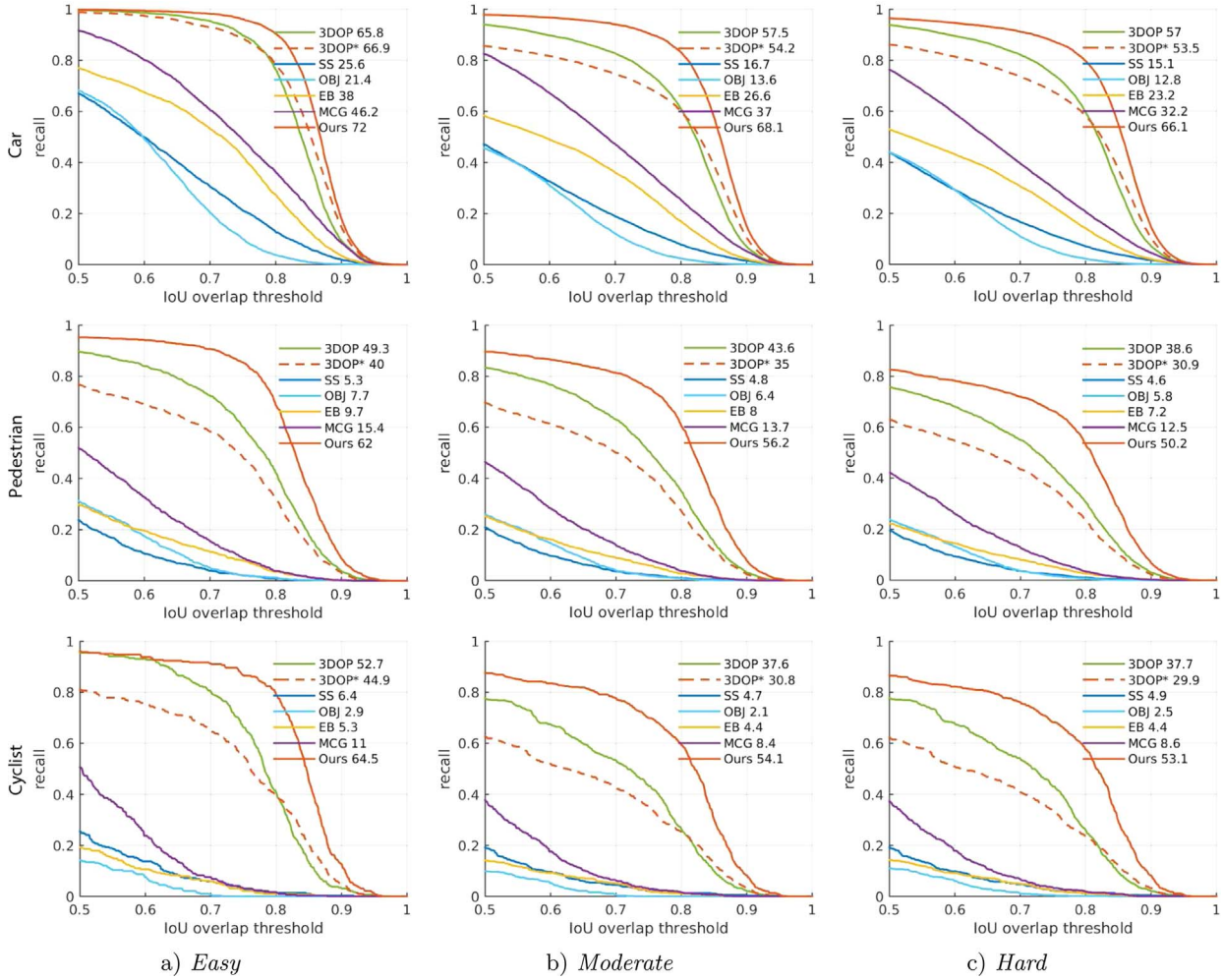

Fig. 5. 2D proposal recall vs. IoU using 500 bounding box candidates. The AR values are shown next to the algorithm labels.

4. Experimental results and discussion

4.1. Experimental setup

4.1.1. Datasets

We carried out the experiment using the challenging KITTI dataset [1], which represents the state-of-the-art benchmark for autonomous driving. Accordingly, the KITTI object detection dataset consists of 7481 training images with available ground truth labels and 7518 testing images. Corresponding rectified right images are also provided to form stereo image pairs.

Three desired object classes are employed: *Car*, *Pedestrian*, and *Cyclist*. The evaluation is carried out with respect to each class in three difficulty levels: *easy*, *moderate*, and *hard*, which are defined according to object height, occlusion level, and truncation scenario. Since the test set contains no ground truth labels, and tends to be used for online evaluation, we used the training and validation sets [19] split from the default training set to train and evaluate the performance of the proposed DeepStereoOP and the competing object proposals algorithms. Specifically, the split training set contains 3712 images, the validation set contains 3769 images, and there is no overlap between the two sets in terms of images extracted from the same sequence.

4.1.2. The competing algorithms

We compare the proposed DeepStereoOP with Objectness (OBJ) [21], SelectiveSearch (SS) [33], MCG [38], EdgeBoxes (EB) [18], and class-specific 3DOP [19]. These methods were selected as they are well-known and have been widely used. Specifically, Objectness and SelectiveSearch represent the state-of-the-art scoring and grouping approaches, respectively. MCG and EdgeBoxes are the top-performing algorithms in the context of common objects [17], and class-specific 3DOP is currently the top-performing object proposals algorithm in the context of autonomous driving [19]. In addition, we report the results of our customized class-independent 3DOP, which DeepStereoOP uses as a baseline. For simplicity, we denote it as 3DOP*. The validation set is used for evaluation. For object detection and orientation estimation, we compare our final object detection pipeline to published methods on the KITTI benchmark [1] using the test set.

4.1.3. Evaluation metrics

Regarding object proposals, we evaluate the performance of the competing algorithms using several proposal recall metrics [12,15,19,17].

- First, we compute the fraction of ground truth bounding boxes covered by proposals as the IoU threshold varies within the range [0.5, 1] for a fixed number of proposals. This is the most important metric and has been used as a primary protocol for most existing evaluations in the literature.
- Second, we compute the recall as a function of number of proposals with fixed IoU threshold. Specifically, we set IoU = 0.7 for *Car*, and IoU = 0.5 for *Pedestrian* and *Cyslist*, following the KITTI overlap criteria. This metric is considered as complementary to the first metric.
- Third, we report the average recall (AR) [17], which has been shown to be well correlated with localization accuracy and object

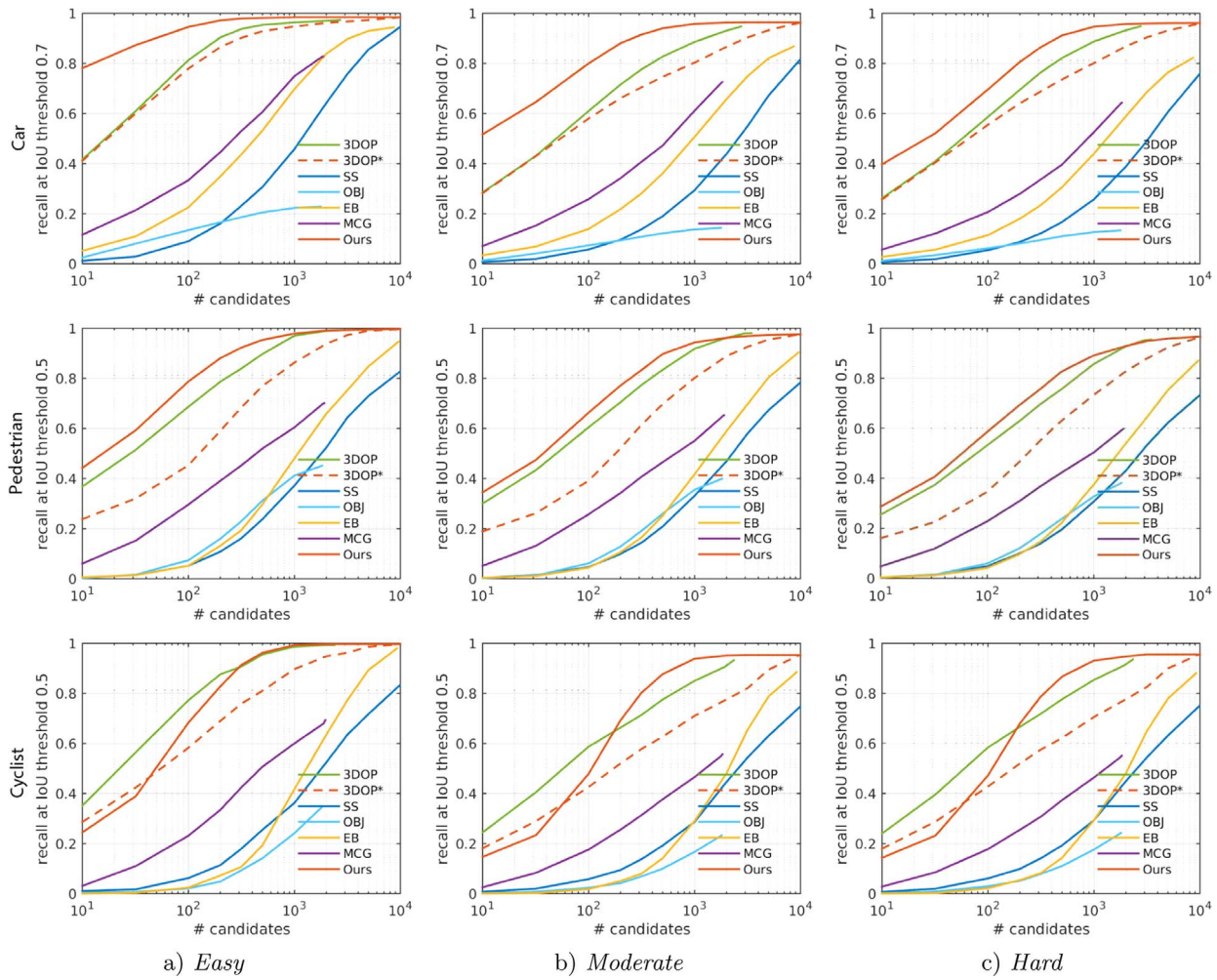

Fig. 6. 2D proposal recall vs. number of bounding box candidates. We use an overlap threshold of 0.7 for *Car*, and 0.5 for *Pedestrian* and *Cyclist*.

detection performance.

- Fourth, we report recall versus distance from the ego-car [19]. This is an important metric in the context of autonomous driving, in which it verifies the robustness of object proposals algorithms against small and far objects.
- Finally, for algorithms producing 3D proposals, which is the case for our DeepStereoOP and 3DOP [19], we report 3D recall versus IoU threshold, number of candidates, and distance from the ego-car in the default *moderate* case.

Regarding KITTI online evaluation, which consists of two tasks, object detection, and object detection and orientation estimation, we evaluate our results using the standard KITTI metrics. Specifically, we use the Average Precision (AP) metric for object detection, and the Average Orientation Similarity (AOS) metric for object detection and orientation estimation.

4.2. Recall Results

Under the above experimental setup, the qualitative 2D results of the proposed DeepStereoOP are shown in Fig. 4, and the recall versus IoU threshold of the competing algorithms is shown in Fig. 5. The number of top-ranked proposals was fixed to 500 for evaluation as in [19]. In general, the proposed DeepStereoOP outperforms all other methods in all three object classes, achieving the best recall across all IoU overlap thresholds. Notably, our approach achieves very good localization accuracy since the recall is maintained consistently high regarding IoU < 0.8 and is only dropped under more strict overlap criteria IoU > 0.8. Indeed, the AR values obtained from DeepStereoOP are significantly larger than those obtained from other methods, proving the robustness of our approach in terms of localization accuracy. Although the class-specific 3DOP is the second-best method, we need to run 3DOP seperately three times with respect to each object class, while we only need to run DeepStereoOP once. This shows the advantage of our approach over class-specific 3DOP, especially when increasing the number of classes to detect other objects rather than only the car, pedestrian, and cyclist. For 3DOP*, which practically achieves the same recall as our approach when considering the entire set of proposals, its top 500 candidates hardly preserve that recall. This shows the effectiveness of our proposed re-ranking model compared to the original ranking model using MRF [19]. More precisely, our top-ranked proposals reported in Fig. 5 achieves 19.47 AR better than those of 3DOP* in *moderate* case, which is a significant improvement considering AR values obtained from Objectness, SelectiveSearch, EdgeBoxes, and MCG are only 7.37, 8.73, 13, and 19.7, respectively.

We now discuss recall as a function of the number of proposals as shown Fig. 6, in which we define a proposal to be recalled if IoU exceeds 0.7 for *Car*, and 0.5 for *Pedestrian* and *Cyclist*, following the KIT TI overlap criteria. Accordingly, DeepStereoOP outperforms all other methods in the *Car* and *Pedestrian* categories. Interestingly, we can see that DeepStereoOP achieves around 90% recall for *Car* by using only 500 proposals for *moderate* and *hard* cases. On the other hand, class-specific 3DOP and 3DOP* require more than 1000 proposals in order to reach 90% recall and other methods require significantly more candidates or even become saturated and cannot match that criterion. For the *Cyclist* category, the proposed DeepStereoOP also outperforms other

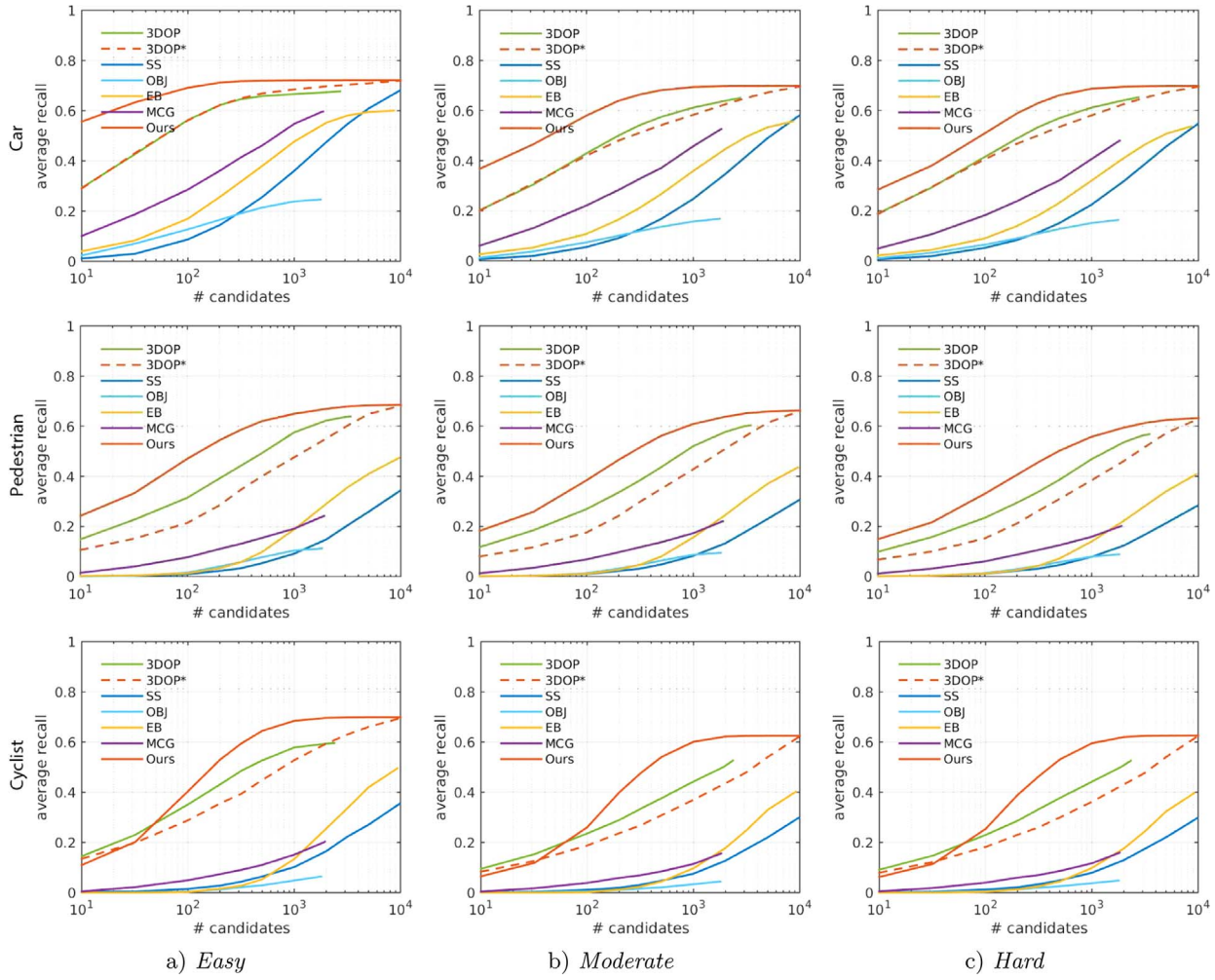

Fig. 7. Average recall (AR) vs. number of 2D bounding box candidates.

methods except class-specific 3DOP, in which it obtained better recall when using less than 500 proposals for the *easy* case, and less than 300 proposals for the *moderate* and *hard* cases. This can be explained by the fact that the KITTI dataset does not contain sufficient amount of *Cyclist* data for training. In particular, the split training set contains only 597 *Cyclist* ground truth labels, compared to 11,017 and 2113 labels for *Car* and *Pedestrian*, respectively. We believe the performance of DeepStereoOP can be further improved when a greater amount of training data is provided. Nevertheless, DeepStereoOP outperforms all other methods in *moderate* and *hard* cases when using more than 300 proposals.

It is noted that Fig. 6 only shows recall at a fixed IoU, which does not reflect the localization accuracy. Hence, we also report AR across IoU $\in [0.5, 1]$ versus number of proposals as shown in Fig. 7. Obviously, our DeepStereoOP outperforms all other methods in *Car* and *Pedestrian* categories, and this is also the case for *Cyclist* when considering more than 100 proposals. This shows the considerable potential of DeepStereoOP in object detection because AR has been shown to be well correlated with the final object detection performance [17]. SelectiveSearch, Objectness, EdgeBoxes, and MCG achieve very low AR accordingly.

Apart from the previous three proposal recall comparisons, which are widely used in the setting of common objects, Fig. 8 shows recall with respect to distance from the ego-car to illustrate the robustness of the competing algorithms against small and far objects, which is an important characteristics in the context of autonomous driving. We can see that DeepStereoOP achieves the best overall recall across various distances. Class-specific 3DOP achieves comparable recall to ours within the range of 40 m, but shows reduced recall when further objects are considered. SelectiveSearch, Objectness, EdgeBoxes, and MCG are not robust against far objects, especially against those further than 20 m.

Regarding 3D object proposals, Fig. 9 shows qualitative 3D results of the proposed DeepStereoOP and Fig. 10 shows recall versus IoU, recall versus distance, and recall versus number of candidates in the *moderate* case. Accordingly, the superior performance of DeepStereoOP over 3DOP with regard to 2D results in *Car* and *Pedestrian* categories also holds for 3D results. For *Cyclist*, DeepStereoOP achieves better results when using more than 200 proposals. Again, the ability to produce 3D bounding box proposals is another advantage of the proposed DeepStereoOP compared to other methods such as SelectiveSearch, Objectness, EdgeBoxes, and MCG.

4.3. Ablation studies

We study the impact of input features in the DeepStereoOP algorithm. Specifically, we aim to study the advantage of the proposed two-stream architecture versus one-stream architecture which only uses RGB, and to identify which of the two depth features, i.e. disparity map or distance to the ground, plays a more important role. With the same parameter initialization and training strategy mentioned in subSection 3.2, we compare proposals re-ranked by four architectures as follows: one-stream architecture with RGB only, two-stream architecture with RGB and disparity map, two-stream architecture with RGB and distance to the ground, and the proposed architecture with all input features.

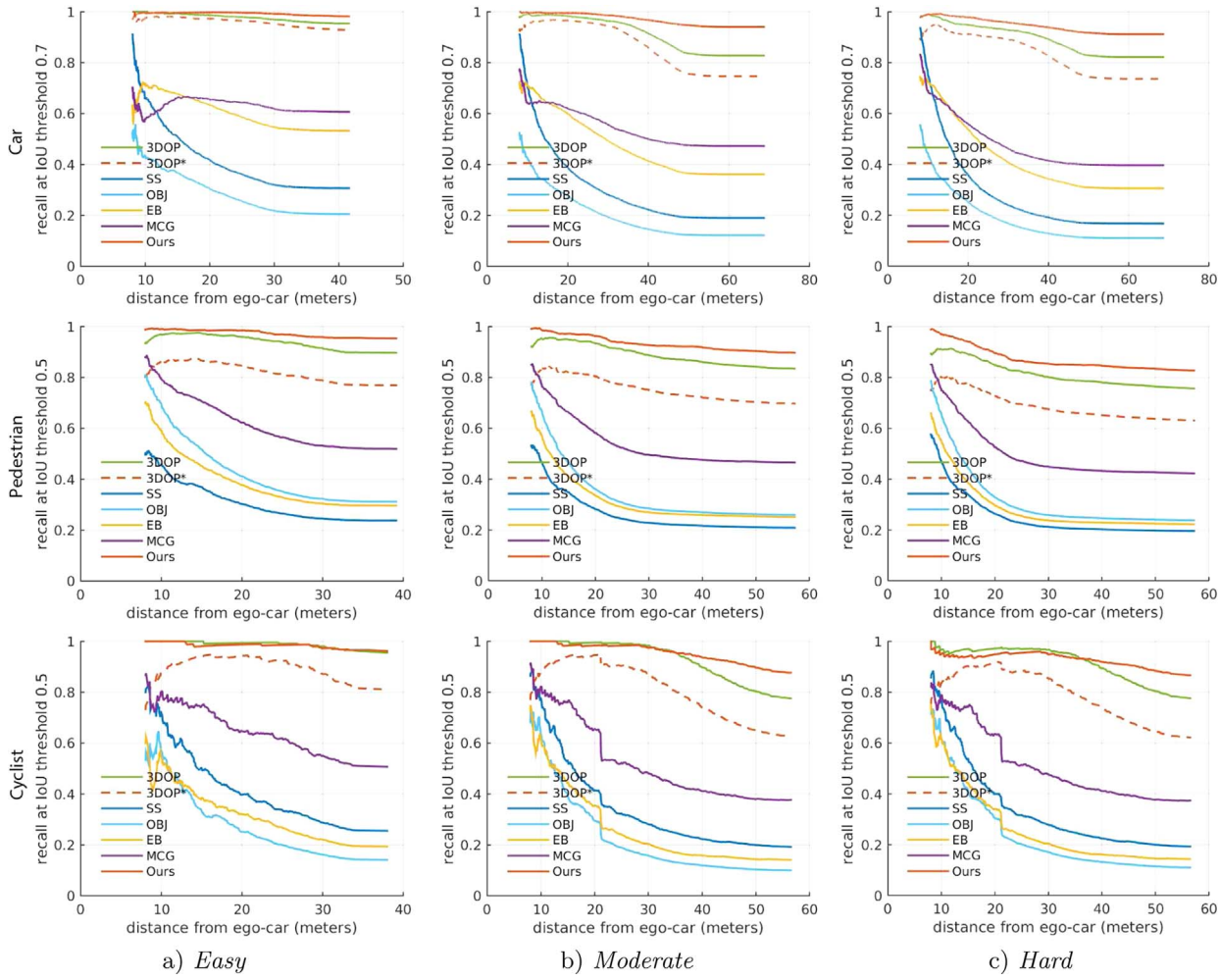

Fig. 8. 2D proposal recall vs. distance from ego-car using 500 candidates. We use an overlap threshold of 0.7 for *Car*, and 0.5 for *Pedestrian* and *Cyclist*.

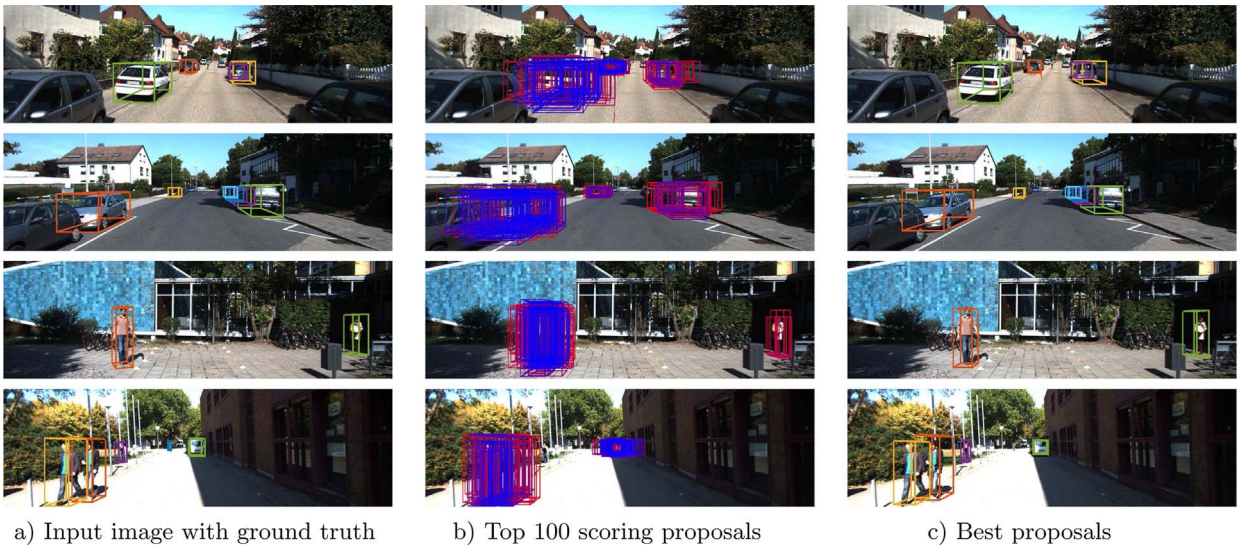

Fig. 9. Qualitative 3D object proposals results of the proposed DeepStereoOP on the validation set. Top scoring proposals are marked in descending order from blue to red.

Comparisons of different architectures are shown in Table 1. Accordingly, we compare AR computed from 2D proposal recall vs. IoU using 500 bounding box candidates. Those AR values are corresponding to those shown in Fig. 5. In general, all three two-stream architectures yield better results than one-stream architecture with RGB only. For disparity map and distance to the ground comparison, two-stream architecture with disparity map shows better AR in *Pedestrian* category, while two-stream version with distance to the ground is better in *Cyclist* category. This shows the importance of both features. As a result, the proposed two-stream architecture with the combination of all features yields the best results.

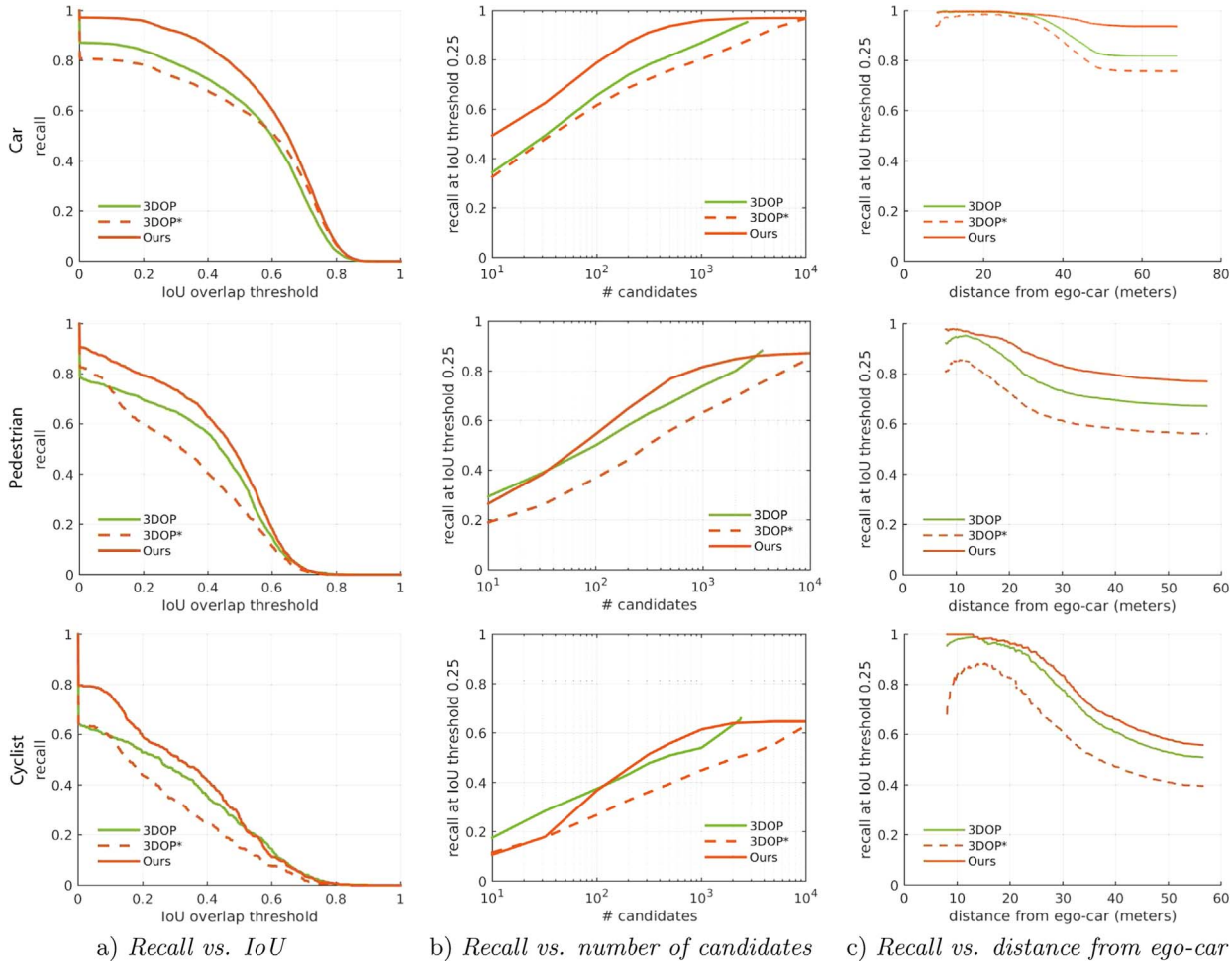

Fig. 10. 3D proposal recall in *moderate* case. We use 500 bounding box candidates in recall vs. IoU, and use IoU = 0.25 for all object classes in recall vs. distance from ego-car and recall vs. number of candidates.

Table 1
AR Comparison of Different Architectures Using 2D Proposal Recall vs. IoU with 500 Proposals.

Method	Car			Pedestrian			Cyclist		
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
Ours	**72.0**	**68.1**	**66.1**	**62.0**	**56.2**	**50.2**	**64.5**	**54.1**	**53.1**
RGB + Disparity map	*72.0*	68.1	*66.1*	*61.0*	*54.9*	*49.0*	62.9	51.7	50.9
RGB + Distance to the ground	71.9	68.1	66.0	60.2	54.3	48.4	*63.9*	*53.1*	*52.1*
RGB, *one-stream*	71.9	67.9	65.9	59.5	53.6	47.7	61.8	50.4	49.7

4.4. Object detection

Regarding our full object detection pipeline, which combines DeepStereoOP and Fast R-CNN, we used the default training set to carry out the training procedures. The results on the test set are reported and compared to the state-of-the-art object detectors in Tables 2, 3. Specifically, Table 2 shows AP values for object detection, while Table 3 shows AOS values for object detection and orientation estimation. Qualitative 2D and corresponding 3D detection results are shown in Figs. 11 and 12.

In terms of AP, our approach achieves significant improvement for all object classes and difficulty levels over the state-of-the-art Faster R-CNN [45], which used deep RPN for generating proposals, as well as other algorithms ranging from spLBP [66] to LSVM-MDPM-sv [55]. Compared to 3DOP and Mono3D, which used the same Fast R-CNN architecture as that used in our approach, the proposed algorithm produces slightly better results for *Car*, and comparable results for *Pedestrian* and *Cyclist*. It is worth noting that we only used 2000 class-independent proposals, while the other two methods used a total of 6000 class-specific candidates. This is the essential difference as well as the advantage of class-independent algorithms over class-specific approaches in terms of number of proposals used for detection.

Comparing AOS among these approaches, our approach significantly outperforms those ranging from 3DVP [58] to AOG [67] as shown in Table 3. Moreover, our approach outperforms Mono3D for all three object classes and difficulty levels even though we used only 2000 proposals, while Mono3D used 6000 proposals. Compared to 3DOP, we achieve better results for *Car* and comparable results for *Pedestrian*. While 3DOP obtains better results for *Cyclist*, we believe our approach can obtain more accurate results when KITTI provides more sufficient data for training as mentioned earlier.

In terms of computational performance, our proposed approach is also fairly efficient. In particular, the customized class-independent 3DOP takes about 1,32 s to generate initial proposals on a single core of

Table 2
Quantitative KITTI Object Detection Evaluation Using AP (in %).

Method	Car			Pedestrian			Cyclist		
	Moderate	Easy	Hard	Moderate	Easy	Hard	Moderate	Easy	Hard
Ours	**89.04**	**93.45**	**79.58**	67.32	**81.82**	**65.12**	65.84	**79.58**	57.90
3DOP [19]	88.64	93.04	79.10	**67.47**	81.78	64.70	**68.94**	78.39	**61.37**
Mono3D [47]	88.66	92.33	78.96	66.68	80.35	63.44	66.36	76.04	58.87
Faster R-CNN [45]	81.84	86.71	71.12	65.90	78.86	61.18	63.35	72.26	55.90
spLBP [66]	77.39	87.18	60.59	–	–	–	–	–	–
Regionlets [34]	76.45	84.75	59.70	61.15	73.14	55.21	58.72	70.41	51.83
AOG [67]	75.94	84.80	60.70	–	–	–	–	–	–
3DVP [58]	75.77	87.46	65.38	–	–	–	–	–	–
CompACT-Deep [68]	–	–	–	58.74	70.69	52.71	–	–	–
DeepParts [69]	–	–	–	58.67	70.49	52.78	–	–	–
FilteredICF [70]	–	–	–	56.75	67.65	51.12	–	–	–
R-CNN [71]	–	–	–	50.13	61.61	44.79	–	–	–
SubCat [53]	75.46	84.14	59.71	42.34	54.67	37.95	–	–	–
OC-DPM [54]	65.95	74.94	53.86	–	–	–	–	–	–
DPM-VOC + VP [57]	64.71	74.95	48.76	44.86	59.48	40.37	31.08	42.43	28.23
SquaresICF [72]	–	–	–	44.42	57.33	40.08	–	–	–
MDPM-un-BB [4]	62.16	71.19	48.43	–	–	–	–	–	–
LSVM-MDPM-sv [55]	56.48	68.02	44.18	39.36	47.74	35.95	27.50	35.04	26.21

Table 3
Quantitative KITTI Object Detection and Orientation Estimation Evaluation Using AOS (in %).

Method	Car			Pedestrian			Cyclist		
	Moderate	Easy	Hard	**Moderate**	Easy	Hard	**Moderate**	Easy	Hard
Ours	**86.86**	**92.04**	**77.34**	59.28	72.82	56.85	55.69	69.20	48.95
3DOP [19]	86.10	91.44	76.52	**59.80**	**72.94**	**57.03**	**58.68**	**70.13**	**52.35**
Mono3D [47]	86.62	91.01	76.84	58.15	71.15	54.94	54.97	65.56	48.77
3DVP [58]	74.59	86.92	64.11	–	–	–	–	–	–
SubCat [53]	74.42	83.41	58.83	34.18	44.32	30.76	–	–	–
OC-DPM [54]	64.42	73.50	52.40	–	–	–	–	–	–
DPM-VOC + VP [57]	61.84	72.28	46.54	39.83	53.55	35.73	23.17	30.52	21.58
LSVM-MDPM-sv [55]	55.77	67.27	43.59	35.49	43.58	32.42	22.07	27.54	21.45
AOG [67]	30.77	33.79	24.75	–	–	–	–	–	–

Fig. 11. Qualitative car detection results on the test set.

a PC with an Intel Core i7-4790 processor, while DeepStereoOP and Fast R-CNN take around 0,38 s and 1,7 s to accomplish object proposals re-ranking and object detection on a NVIDIA Titan X GPU, respectively.

After all, the robustness of the proposed class-independent object proposals re-ranking algorithm, which is fast and is able to achieve high recall and good localization, makes it particularly useful in many circumstances.

5. Conclusion

We presented a robust object proposals re-ranking algorithm for object detection in autonomous driving. The proposed DeepStereoOP re-ranks proposals generated from a customized class-independent 3DOP method using a lightweight two-stream CNN architecture. The proposed algorithm exploits not only RGB features as in the conven-

Fig. 12. Qualitative pedestrian and cyclist detection results on the test set.

tional CNN architecture, but also depth features including disparity map and distance to the ground, both of which can be computed from a stereo image pair using a stereo matching algorithm. The experimental results showed that our proposed algorithm outperforms the state-of-the-art object proposals algorithms in terms of both proposal recall and localization accuracy. According to the KITTI benchmark, the combination of DeepStereoOP and Fast R-CNN achieves one of the best object detection results on all three object classes.

We highly anticipate that our contributions will be used in the development of object proposals and object detection in autonomous driving.

Acknowledgment

This work was supported by Samsung Research Funding Center of Samsung Electronics under Project Number SRFC-IT1402-12.

References

[1] A. Geiger, P. Lenz, R. Urtasun, Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2012, pp. 3354–3361. ⟨http://www.cvlibs.net/datasets/kitti/eval_object.php⟩.
[2] P. Viola, M. Jones, Robust Real-Time Face Detection, J. Comput. Vis. 57 (2) (2004) 137–154.
[3] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), vol. 1, 2005, pp. 886–893.
[4] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object Detection with Discriminatively Trained Part-Based Models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[5] A. Krizhevsky, I. Sutskever, G. Hinton, Imagenet Classification with Deep Convolutional Neural Networks, in: Proceedings Adv. Neural Inf. Process. Syst. (NIPS), 2012.
[6] X. Wang, M. Yang, S. Zhu, Y. Lin, Regionlets for Generic Object Detection, in: Proceedings International Conference on Comput. Vis. (ICCV), 2013, pp. 17–24.
[7] C. Szegedy, S. Reed, D. Erhan, D. Anguelov, Scalable, high-quality object detection, ⟨http://arXiv:1412.1441arXiv:1412.1441⟩.
[8] K. Simonyan, A. Zisserman, Very Deep Convolutional Networks for Large-Scale Image Recognition, ⟨http://arXiv:1409.1556arXiv:1409.1556⟩.
[9] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 580–587.
[10] K. He, X. Zhang, S. Ren, J. Sun, Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition, in: Proceedings ECCV2014, vol. 8691 of LNCS, 2014, pp. 346–361.
[11] R. Girshick, Fast R-CNN, in: Proceedings International Conference on Comput. Vis. (ICCV), 2015.
[12] B. Alexe, T. Deselaers, V. Ferrari, What is an object?, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2010, pp. 73–80.
[13] J. Carreira, C. Sminchisescu, Constrained parametric min-cuts for automatic object segmentation, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2010, pp. 3241–3248.
[14] I. Endres, D. Hoiem, Category Independent Object Proposals, in: Proceedings EC CV2010, vol. 6315 of LNCS, 2010, pp. 575–588.
[15] K.E.A. van de Sande, J.R. Uijlings, T. Gevers, A.W.M. Smeulders, Segmentation As Selective Search for Object Recognition, in: Proceedings International Conference on Comput. Vis. (ICCV), 2011.
[16] J. Hosang, R. Benenson, B. Schiele, How good are detection proposals, really? in: BMVC, 2014.
[17] J. Hosang, R. Benenson, P. Dollár, B. Schiele, What makes for effective detection proposals? IEEE Trans. Pattern Anal. Mach. Intell. 38 (4) (2016) 814–830.
[18] C.L. Zitnick, P. Dollár, Edge Boxes: Locating Object Proposals from Edges, in: Proceedings ECCV2014, vol. 8693 of LNCS, 2014, pp. 391–405.
[19] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, R. Urtasun, 3D Object Proposals for Accurate Object Class Detection, in: Proceedings Adv. Neural Inf. Process. Syst. (NIPS), 2015, pp. 424–432.
[20] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, Backpropagation applied to handwritten zip code recognition, Neural Comput. 1 (4) (1989) 541–551.
[21] B. Alexe, T. Deselaers, V. Ferrari, Measuring the Objectness of Image Windows, IEEE Trans. Pattern Anal. Mach. Intell. 34 (11) (2012) 2189–2202.
[22] E. Rahtu, J. Kannala, M.B. Blaschko, Learning a category independent object detection cascade, in: Proceedings International Conference on Comput. Vis. (IC CV), 2011.
[23] Z. Zhang, J. Warrell, P.H.S. Torr, Proposal generation for object detection using cascaded ranking SVMs, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2011, pp. 1497–1504.
[24] M.-M. Cheng, Z. Zhang, W.-Y. Lin, P. Torr, BING: Binarized Normed Gradients for Objectness Estimation at 300fps, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 3286–3293.
[25] Q. Zhao, Z. Liu, B. Yin, Cracking BING and beyond, in: BMVC, 2014.
[26] Z. Zhang, Y. Liu, T. Bolukbasi, M.-M. Cheng, V. Saligrama, BING++: A Fast High Quality Object Proposal Generator at 100 fps, ⟨http://arXiv:1511.04511arXiv:1511.04511⟩.
[27] C. Lu, S. Liu, J. Jia, C. K. Tang, Contour Box: Rejecting Object Proposals without Explicit Closed Contours, in: Proceedings International Conference on Comput. Vis. (ICCV), 2015, pp. 2021–2029.
[28] W. Kuo, B. Hariharan, J. Malik, DeepBox: Learning Objectness with Convolutional Networks, in: Proceedings International Conference on Comput. Vis. (ICCV), 2015, pp. 2479–2487.
[29] X. Chen, H. Ma, X. Wang, Z. Zhao, Improving Object Proposals with Multi-Thresholding Straddling Expansion, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 2587–2595.
[30] J. Carreira, C. Sminchisescu, CPMC: automatic Object Segmentation Using Constrained Parametric Min-Cuts, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1312–1328.
[31] A. Humayun, F. Li, J.M. Rehg, RIGOR: Reusing Inference in Graph Cuts for Generating Object Regions, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 336–343.
[32] I. Endres, D. Hoiem, Category-independent object proposals with diverse ranking, IEEE Trans. Pattern Anal. Mach. Intell. 36 (2) (2014) 222–234.
[33] J.R. Uijlings, K.E.A. van de Sande, T. Gevers, A.W.M. Smeulders, Selective Search for Object Recognition, Int. J. Comput. Vis. 104 (2) (2013) 154–171.
[34] X. Wang, M. Yang, S. Zhu, Y. Lin, Regionlets for Generic Object Detection, IEEE Trans. Pattern Anal. Mach. Intell. 37 (10) (2015) 2071–2084.
[35] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient Graph-Based Image Segmentation, International J. Comput. Vis. 59 (2).
[36] S. Manén, M. Guillaumin, L.V. Gool, Prime Object Proposals with Randomized

Prim's Algorithm, in: Proceedings International Conference on Comput. Vis. (ICCV), 2013, pp. 2536–2543.
[37] P. Rantalankila, J. Kannala, E. Rahtu, Generating Object Segmentation Proposals Using Global and Local Search, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 2417–2424.
[38] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, J. Malik, Multiscale Combinatorial Grouping, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 328–335.
[39] P. Krahenbuhl, V. Koltun, Geodesic Object Proposals, in: Proceedings ECCV2014, vol. 8693 of LNCS, 2014, pp. 725–739.
[40] P. Krahenbuhl, V. Koltun, Learning to Propose Objects, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 1574–1582.
[41] T. Lee, S. Fidler, S. Dickinson, Learning to Combine Mid-level Cues for Object Proposal Generation, in: Proceedings International Conference on Comput. Vis. (IC CV), 2015.
[42] D. Erhan, C. Szegedy, A. Toshev, D. Anguelov, Scalable Object Detection Using Deep Neural Networks, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 2155–2162.
[43] A. Ghodrati, A. Diba, M. Pedersoli, T. Tuytelaars, L.V. Gool, DeepProposal: Hunting Objects by Cascading Deep Convolutional Layers, in: Proceedings International Conference on Comput. Vis. (ICCV), 2015, pp. 2578–2586.
[44] P.O. Pinheiro, R. Collobert, P. Dollár, Learning to Segment Object Candidates, in: Proceedings Adv. Neural Inf. Process. Syst. (NIPS), 2015, pp. 1981–1989.
[45] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in: Proceedings Adv. Neural Inf. Process. Syst. (NIPS), 2015.
[46] Z. Jie, X. Liang, J. Feng, W. F. Lu, E. H. F. Tay, S. Yan, Scale-aware Pixel-wise Object Proposal Networks, ⟨http://arXiv:1601.04798v2arXiv:1601.04798v2⟩.
[47] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, R. Urtasun, Monocular 3D Object Detection for Autonomous Driving, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2016.
[48] K. Yamaguchi, D. McAllester, R. Urtasun, Efficient Joint Segmentation, Occlusion Labeling, Stereo and Flow Estimation, in: Proceedings ECCV2014, vol. 8693 of LNCS, 2014, pp. 756–771.
[49] Z. Sun, G. Bebis, R. Miller, On-road vehicle detection: a review, IEEE Trans. Pattern Anal. Mach. Intell. 28 (5) (2010) 694–711.
[50] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: an evaluation of the state of the art, IEEE Trans. Pattern Anal. Mach. Intell. 34 (4) (2012) 743–761.
[51] S. Sivaraman, M.M. Trivedi, Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking and behavior analysis, IEEE Trans. Intell. Transp. Syst. 14 (4) (2013) 1773–1795.
[52] R. Benenson, M. Omran, J. Hosang, B. Schiele, Ten Years of Pedestrian Detection, What Have We Learned?, in: Proceedings ECCV2014 Workshops, vol. 8926 of LNCS, 2014, pp. 613–627.
[53] E. Ohn-Bar, M.M. Trivedi, Learning to detect vehicles by clustering appearance patterns, IEEE Trans. Intell. Transp. Syst. 16 (5) (2015) 2511–2521.
[54] B. Pepik, M. Stark, P. Gehler, B. Schiele, Occlusion patterns for object class detection, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2013, pp. 1–8.
[55] A. Geiger, C. Wojek, R. Urtasun, Joint 3d estimation of objects and scene layout, in: Proceedings Adv. Neural Inf. Process. Syst. (NIPS), 2011.
[56] J. Yebes, L. Bergasa, R. Arroyo, A. Lzaro, Supervised learning and evaluation of KIT TI's cars detector with DPM, in: Proceedings IV, 2014.
[57] B. Pepik, M. Stark, P. Gehler, B. Schiele, Multi-View and 3D Deformable Part Models, IEEE Trans. Pattern Anal. Mach. Intell. 37 (11) (2015) 2232–2245.
[58] Y. Xiang, W. Choi, Y. Lin, S. Savarese, Data-driven 3d voxel patterns for object category recognition, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 1903–1911.
[59] F. Guney, A. Geiger, Displets: Resolving Stereo Ambiguities Using Object Knowledge, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 4165–4175.
[60] J. Zbontar, Y. LeCun, Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches, ⟨http://arXiv:1510.05970arXiv:1510.05970⟩.
[61] C.C. Pham, V.Q. Dinh, J.W. Jeon, Robust non-local stereo matching for outdoor driving images using segment-simple-tree, Signal Process. Image Commun. 39 (2015) 173–184.
[62] I. Tsochantaridis, T. Hofmann, T. Joachims, Y. Altun, Support Vector Learning for Interdependent and Structured Output Spaces, in: Proceedings International Conference Machine learning (ICML), 2004.
[63] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neuralnetworks from over tting, J. Mach. Learn. Res. 15 (1) (2014) 1929–1958.
[64] Y. Zhu, R. Urtasun, R. Salakhutdinov, S. Fidler, segDeepM: Exploiting Segmentation and Context in Deep Neural Networks for Object Detection, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 4703–4711.
[65] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, http://arXiv:1408.5093arXiv:1408.5093.
[66] Q. Hu, S. Paisitkriangkrai, C. Shen, A. van den Hengel, F. Porikli, Fast detection of multiple objects in traffic scenes with a common detection framework, IEEE Trans. Intell. Transp. Syst. 17 (4) (2016) 1002–1014.
[67] T. Wu, B. Li, S.-C. Zhu, Integrating context and occlusion for car detection by hierarchical and-or model, in: Proceedings ECCV2014, vol. 8694 of LNCS, 2014, pp. 652–667.
[68] Z. Cai, M. Saberian, N. Vasconcelos, Learning Complexity-Aware Cascades for Deep Pedestrian Detection, in: Proceedings International Conference on Comput. Vis. (IC CV), 2015, pp. 3361–3369.
[69] Y. Tian, P. Luo, X. Wang, X. Tang, Deep Learning Strong Parts for Pedestrian Detection, in: Proceedings International Conference on Comput. Vis. (ICCV), 2015, pp. 1904–1912.
[70] S. Zhang, R. Benenson, B. Schiele, Filtered Channel Features for Pedestrian Detection, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2005, pp. 1751–1760.
[71] J. Hosang, M. Omran, R. Benenson, B. Schiele, Taking a deeper look at pedestrians, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2015, pp. 4073–4082.
[72] R. Benenson, M. Mathias, T. Tuytelaars, L.V. Gool, Seeking the Strongest Rigid Detector, in: Proceedings IEEE International Conference Comput. Vis. Pattern Recognit. (CVPR), 2013, pp. 3666–3673.