Hindawi
Mobile Information Systems
Volume 2017, Article ID 3175186, 11 pages
https://doi.org/10.1155/2017/3175186

Research Article

Improved Object Proposals with Geometrical Features for Autonomous Driving

Yiliu Feng, Wanzeng Cai, Xiaolong Liu, Huini Fu, Yafei Liu, and Hengzhu Liu

College of Computer, National University of Defense Technology, Changsha, China

Correspondence should be addressed to Yiliu Feng; fengyiliu11@nudt.edu.cn

Received 13 February 2017; Accepted 22 March 2017; Published 26 April 2017

Academic Editor: Zhengguo Sheng

This paper aims at generating high-quality object proposals for object detection in autonomous driving. Most existing proposal generation methods are designed for the general object detection, which may not perform well in a particular scene. We propose several geometrical features suited for autonomous driving and integrate them into state-of-the-art general proposal generation methods. In particular, we formulate the integration as a feature fusion problem by fusing the geometrical features with existing proposal generation methods in a Bayesian framework. Experiments on the challenging KITTI benchmark demonstrate that our approach improves the existing methods significantly. Combined with a convolutional neural net detector, our approach achieves state-of-the-art performance on all three KITTI object classes.

1. Introduction

Object detection has been developed in many years and there are a variety of robust approaches [1–5]. In the early years, most of them follow the sliding-window paradigm. But enormous numbers of windows would waste a large amount of efforts on no-object areas. In order to overcome this problem, an effective framework is proposed: object proposals generation followed by a classifier. Most of the methods are designed to generate object proposals for general object detection, such as Edgeboxes [6] and Selective Search [7]. They both work well on the PASCAL VOC dataset [8].

However these methods would suffer a great performance degradation, when they are applied to autonomous driving scene, such as the challenging KITTI benchmark [9], which contains many small objects, occlusion, high saturated areas, and even shadows.

In this paper, we propose an effective approach to improve the results of object proposals in autonomous driving scene. Our work is motivated by the following observations. First, there are three primary objects, in autonomous driving scene, Car, Cyclist, and Pedestrian. These three objects usually lie on the ground with different height. So the proposals should lie on the ground. Second, the real-world size of objects in one category would vary far less than their image-world size, but the real-world size of different categories are also different. It is helpful to use the object size prior of object as an indicator to generate proposals. The details are discussed in Section 3.

This paper has two fundamental contributions.

(1) We propose two new geometric features, AR and SD2, to represent the object size prior. We exploit D2R as an indicator to constraint the proposals lying on the ground. These features are demonstrated to be effective for generating fewer proposals with higher recall.

(2) We deeply analyze the four geometric features, AR, SD2, DMD, and D2R, and propose a method to combine these features with existing methods efficiently. The final results on the KITTI object detection benchmark achieve the state-of-the-art performance in stereo-based methods.

Since it is inevitable to use the depth information to compute the geometric features, we assume a stereo image pair as an input and obtain depth information via the state-of-the-art approach by Yamaguchi et al. [10].

2. Related Work

The main idea of object proposal method is to generate relatively fewer number of bounding boxes that contain the

objects in an image that we are interested in with high recall. Existing proposal generation methods are often based on low-level image features, which can be divided into two categories generally: grouping methods and window scoring methods.

2.1. Grouping Methods. Grouping proposal methods aim to generate multiple segments that are likely to correspond to objects. To cover different objects with various size, most methods attempt to merge the output of a hierarchical image segmentation algorithm. The decision to merge segments is designed manually typically based on superpixel shape, appearance features, and boundary estimates.

Selective Search [7] is one of the most well-known grouping methods which greedily merges superpixels to generate proposals. The method has no learned parameters and has been broadly used as the proposal method of choice by many state-of-the-art object detectors, such as the R-CNN detector.

In order to detect objects with different size, MCG [11] propose an algorithm for fast computing multiscale hierarchical segmentation. They merge the segments based on edge strength and ranking the results using appropriate features.

Since SS and MCG both need an initial image segmentation which impacts the object proposal results, CPMC [12] does not have initial segmentations and uses graph cut directly on pixels. Then it ranks the resulting segments based on a large pool of features.

2.2. Window Scoring Methods. Window scoring methods are to score each candidate window to indicate how likely an object of interest is contained in it. Compared to grouping approaches these methods usually directly return bounding boxes with fast speed. However, they tend to generate proposals with low localization accuracy unless the window sampling is performed very densely.

Objectness [13, 14] is one of the earliest window scoring proposal methods. A model is trained to distinguish objects from the background and an initial set of proposals is generated from salient locations in an image. Then each proposal is scored by a Bayesian framework combining several image features including color, edge density, saliency, and superpixels straddling.

BING [15] is an extremely fast object proposal method (300 fps/s on CPU). Gradient features are used to train a simple linear classifier to detect object proposals in a sliding-window framework which can yield 96.2% recall with 1000 proposals at the IOU threshold of 0.5. Meanwhile BING needs to resize the candidate window to $8 * 8$ which leads to low localization accuracy when mapping the $8 * 8$ window back to the original image. The recall drops rapidly when the IOU threshold gets larger.

Edgeboxes [6] is a very fast and efficient region proposal method, which can generate millions of candidate boxes in a fraction of one second and achieve nearly 96% recall at overlap threshold of 0.5 by using 1000 proposals on the PASCAL VOC dataset. The main contribution of the method is that the number of contours wholly falling into a bounding box is indicative of the possibility of a box covering an object. All of the bounding boxes are generated by sliding-window algorithm and then scored by measuring the number of edge groups that exist in the box minus some of them that overlap the box's boundary.

However most previous methods are designed for general objects; they do not perform well in a particular scene such as the KITTI [9] benchmark. 3DOP [16] is an excellent proposal generation method which exploits object size priors, ground plane, and several depth informed features such as free space, point densities inside the box, visibility, and distance to the road to place proposals in the form of 3D bounding boxes. After generating a large number of proposals, the method scores every proposal by minimizing an energy function. The energy function encodes object size priors, ground plane, and a variety of depth informed features. Their final results achieve a 25% higher recall with 2,000 proposals than the state-of-the-art RGB-D method MCG-D [17] on the KITTI benchmark.

Most grouping and scoring methods mentioned above either purely use RGB appearance features or only use depth informed geometric features which ignore their complement of those two features. Although some methods, such as MCG-D, use RGB and depth features simultaneously, it is not suitable for autonomous driving because of the complex outdoor environment. In this paper, we propose a method to exploit both the appearance features and the geometric features. Our work formulates the problem by fusing those two complementary features in a Bayesian framework for obtaining high-quality object proposals in autonomous driving.

3. Methodology

As mentioned in previous sections, geometric features are important for improving the quality of object proposals. We introduce four geometric features: aspect Ratio, diagonal multiplication distance, area multiplication of the square of the object depth, and distance to the road.

3.1. Geometric Features

3.1.1. Aspect Ratio (AR). Objects in different classes usually have magnificent difference on appearance while those in the same class vary far less. Since an object is tightly bounded by a square box whose aspect ratio of the same class should vary in a specific range, based on this intuition, we use AR as a feature to assess the possibility of an image window covering a specific class. The aspect ratio of a square box is calculated as follows:

$$\mathrm{AR} = \frac{w_b}{h_b}, \tag{1}$$

where w_b is the width of a given bounding box while h_b is the height.

3.1.2. Area Multiplication of the Square of the Object Depth (SD2). Objects' sizes in the image can be measured by the bounding boxes covering them and they vary significantly across the dataset. Meanwhile, the real-world size of objects in the same class varies far less as mentioned in [18]. According to the optical imaging principles, the real-world size A_T and the image size A_I of the object have a specific relationship.

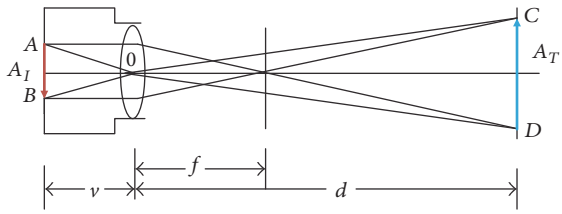

Figure 1: The imaging principle of the camera.

As shown in Figure 1, by using the homothetic triangle theory, the relationship between A_T and A_I can be described as follows:

$$A_T = \frac{d^2}{v^2} A_I, \tag{2}$$

where d is the real-world distance of the object and v is the camera focal length which is usually considered to be fixed.

Depth information has been utilized for object detection in recent years; it can be computed from disparity map or directly obtained by depth sensors, such as Kinect. In this paper, we use a stereo image pair as an input, compute the disparity map via the state-of-the-art approach by Yamaguchi et al. [10], and then calculate the depth by binocular vision theory:

$$\text{depth} = \frac{f * l}{\text{disparity}}, \tag{3}$$

where f is the focal length of the two lenses, l is the distance between two optical centers, and disparity is the horizontal disparity of two stereo-corresponding points. After calculating the depth of all pixels for each image, the average depth of a $3 * 3$ area around the center is used to approximate the depth of an object enclosed by the box:

$$d_{\text{box}} = \frac{1}{9} \sum_{x_i=x_c-1}^{x_c+1} \sum_{y_i=y_c-1}^{y_c+1} \text{depth}(x_i, y_i), \tag{4}$$

where $\text{depth}(x_i, y_i)$ is the depth of point (x_i, y_i) in image and $x_c = x_l + w_b/2$ and $y_c = y_l + h_b/2$ is the center point of the box.

As mentioned above, the relationship between the image size and the depth information of the object can be utilized as a proxy for the real-world object size approximately. The camera focal length can be ignored as it is considered to be a constant. Inspired by the observation of the relationship between real-world size and image size, we use the product of area of the bounding box and the square distance to the camera as an approximate representation of the object size in real-world. The SD2 can be written as

$$\text{SD2} = w_b * h_b * {d_{\text{box}}}^2, \tag{5}$$

where $w_b * h_b$ is the area of the bounding box and can be used as a representation of an object image size approximately.

3.1.3. Diagonal Multiplication Distance (DMD). DMD is the feature that could approximately represent the real-world object size [18].

$$\text{DMD} = \sqrt{{w_b}^2 + {h_b}^2} * d_{\text{box}}, \tag{6}$$

where $\sqrt{{w_b}^2 + {h_b}^2}$ is the diagonal of a bounding box and d_{box} is the depth of the box.

The distributions of DMD and SD2 on Car, Cyclist, and Pedestrian are shown in the second and the third row in Figure 2. It is obvious that DMD and SD2 vary a few in the same class and vary in different ranges which prove the analysis we discussed before.

3.1.4. Distance to the Road (D2R). Since all the annotated objects in the KITTI benchmark are on the ground, the ground plane can be used as an important indicator to predict the possibility that a proposal contains an object. It is more likely to cover an object when the proposal is close to ground plane and is less likely when the proposal is far away from the ground plane. We use the same method in [16] to compute the distance of every pixel to the ground. Then, as in (5), the average of a $3 * 3$ area around the center is used to measure the distance to the road of an object enclosed by the box:

$$\text{D2R} = \frac{1}{9} \sum_{x_i=x_c-1}^{x_c+1} \sum_{y_i=y_c-1}^{y_c+1} \text{Dist}(x_i, y_i). \tag{7}$$

The distribution of D2R on Car, Cyclist, and Pedestrian is shown in the last row in Figure 2.

3.2. Bayesian Framework. As the four proposal features are relatively complementary, using some of them at the same time may appear promising. AR gives only the proportion of object projection size in the image. DMD or SD2 is the replacement for the real-world object size, but either of them depends on precise depth calculated from disparity map. D2R denotes the distance to the road, which can roughly distinguish positive examples from negative examples.

To combine these features (AR, SC, DMD, SD2, and D2R), we train a Bayesian classifier to distinguish between positives and negatives. SC is the initial result of the existing method. For each training image, we sample all the proposals that have an IOU $\geq$ 0.6 with any ground truth as positive W_{obj} and IOU $<$ 0.35 as negative W_{bg}. As there are too many negative proposals we just select 600 randomly for each training image. In this paper, we choose a Naive Bayes approach [13, 14]. In the Naive Bayes model, the features are independent, so training consists of estimating the priors $p(\text{obj})$, $p(\text{bg})$ by relative frequency and the individual feature likelihoods $p(\text{feature} \mid c)$, feature $\in C$ and $c$obj, and bg from the training set we chosen before.

After training, when given a proposal we calculate its posterior probability using the following equation:

$$p\left(\text{obj} \mid C^1\right) = \frac{p\left(C^1 \mid \text{obj}\right) p\left(\text{obj}\right)}{p\left(C^1\right)}, \tag{8}$$

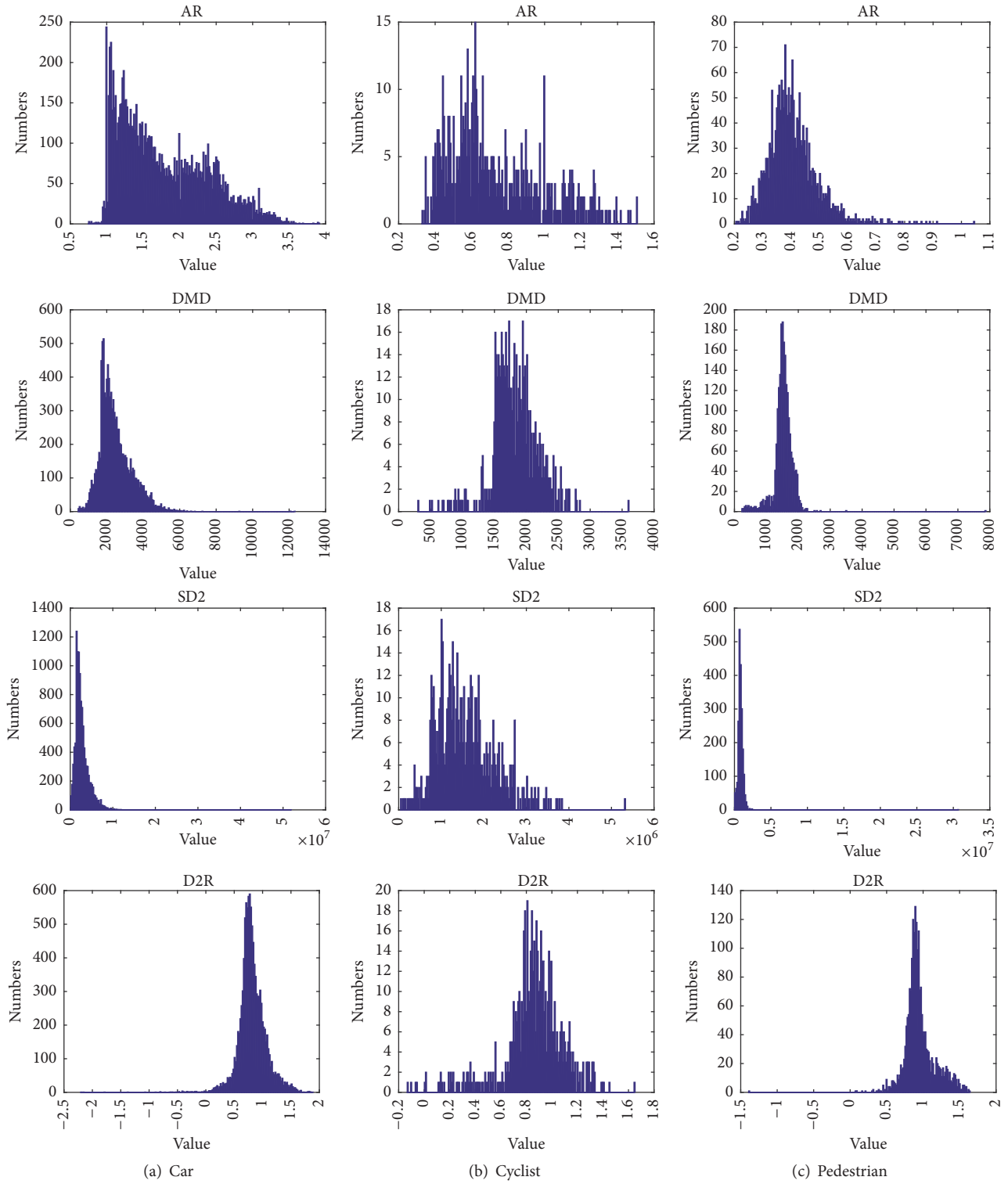

FIGURE 2: Statistic of four object features. For each object class, Car, Cyclist, and Pedestrian, from top to down the features are AR, DMD, SD2, and D2R. We could normalize them to zero mean and unit variance (mean subtraction and division by the standard deviation).

where $C^1 \subseteq C$. This posterior probability constitutes the final proposal score, which is used as the indication of the possibility of a proposal that tends to cover an object.

3.3. Implementation Details. After a large number of positive and negative proposals are sampled, the distribution of their image features (AR, SC, DMD, SD2, and D2R) is demonstrated via the histogram (we sample all the proposals that have an IOU ≥ 0.6 with any ground truth as positive and 600 negative proposals that have IOU < 0.35 for each image). The values of the feature$_i$ V(feature$_i$ | c) are divided into K bins in a range $[V_{\min}, V_{\max}]$. Therefore, the priors p(feature$_i$ | c) are set by relative frequency:

$$p\left(\text{feature}_i \mid c\right) = \frac{N_{\text{Bin}}j}{N}, \quad 1 \leq j \leq K, \tag{9}$$

where $N_{\text{Bin}}j$ is the number of V(feature$_i$ | c) falling into the Bin^j and N is the total number of V(feature$_i$ | c). When any proposal is given, the bin which the value V of the (feature$_i$ | c) falls into is first determined. Then, the individual feature likelihood p(feature$_i$ | c) is roughly equivalent to (9) for each proposal. And the final posterior probability can be calculated according to (8). Noted that (8) allows us to combine any subset C of features, for example, pairs of features $C = \{\text{AR}, \text{SD2}\}$, triplets $C = \{\text{AR}, \text{SD2}, \text{D2R}\}$, or all features $C = \{\text{AR}, \text{SD2}, \text{SC}, \text{DMD}, \text{D2R}\}$. Function (8) can combine any subset rapidly without recomputing the likelihoods.

4. Experiments and Analysis

In this section, we evaluate our method on the challenging KITTI benchmark [9] for all three object classes, which contains 7481 right, 7481 left training images, and 7518 test images. Since the test images do not have any annotations, we split the KITTI training set into train (3,712 images) and validation (3769 images) sets as described in [16]. Bayes model is trained on the train set. All the experiments results are reported on the validation set in three regimes: easy, moderate, and hard, which are defined according to the occlusion and truncation levels of objects.

Following [6], we evaluate the quality of object proposals by using the recall metric. Recall is calculated as the fraction of ground truth objects covered above an IOU threshold. We use curve of the recall versus the number of proposals to depict accuracy at different proposal budgets and recall versus IOU curve to show the variety of recall over different localization precision. In addition, in order to measure the overall accuracy of proposals, we use Area Under the Curve (AUC), which is the area under "recall versus the number of proposals" curve. AUC is a canonical metric which has been shown in [6].

The results of analyzing features and features integration are tested on the hard validation set for all three objects, while the comparison results to the state of the art are on all three object classes and three regimes which use the same metrics depicted in previous section.

4.1. Various Features Integration. We first verify the effectiveness of all the geometric features independently. As our goal is to analyze the performance of each of the features and their combinations which is independent of the baseline method, we only evaluate our method based on Edgeboxes. The results of the baseline method are named SC. As shown in Figure 3, we analyze the baseline and the four proposed geometric features independently to observe the performance of these features. The first row of Figure 3 is the recall versus IOU curve on 500 proposals while the second row is curve of the recall versus the number of proposals on different IOU threshold. For Car, the IOU threshold is 0.7, and it is 0.5 for Cyclist and Pedestrian. We find that all the four proposed features work better than the baseline in which we only use a single feature to generate the proposals. Based on experiments on the three objects we find that D2R is the most useful feature while our proposed feature SD2 is second. DMD has a similar performance to SD2, because they both catch the constancy of object size in real-world. AR is also a useful feature.

Then we combine those geometric features and SC together in a Bayesian framework using different combination to find the best way for fusion of these features. In order to use Bayesian function, the prior probabilities p(obj), p(bg), and p(feature | c) should be first computed. The p(obj) and p(bg) are constant value which are computed in the training stage. The probability p(feature | c) is calculated by using histogram as described in (9). Before we construct the histogram, we normalize them to zero mean and unit variance (mean subtraction and division by the standard deviation). The mean and standard deviation values of each feature are computed on the entire training set.

We combine the five features in a Bayesian framework with all possible combinations. The combinations include 10 ways for any pairs of features, 10 for any triplets, 5 for any four, and 1 for all features together. We have evaluated all the combinations. Since plotting all the combinations is difficult to observe, we only choose 2 top results from pairs of features combinations and triplets of features combinations, 1 from four features combinations, and 1 for all five features. The results are shown in Figure 4. It can be seen that the best performance is obtained by the combination of {AR, D2R, SD2, DMD} and {AR, D2R, SD2}. The results also hint that D2R is the most effective feature, followed by SD2 and DMD, which is consistent with previous observations in Figure 3. We also can find that SD2 and DMD are highly dependent on each other. So we just use SD2 because SD2 is lightly better than DMD. Usually more features make better results. However, it is noteworthy that when combining SC with all other four features the SC does not improve the performance but depresses it. A possible reason is that boxes with larger SC do not mean having higher possibility of containing Car, Cyclist, or Pedestrian. Finally, as shown in Table 1 we summarize the statistics accuracy measures including Area Under the Curve (AUC), the top recall the method can reach (recall), and the number of proposals to achieve recall = 0.75 (M).

4.2. Comparison to the State of the Art. Based on the analysis on the features in previous section, we choose D2R, SD2, and AR as our final choice. As our method can be integrated into any object proposal generation method, we verify its

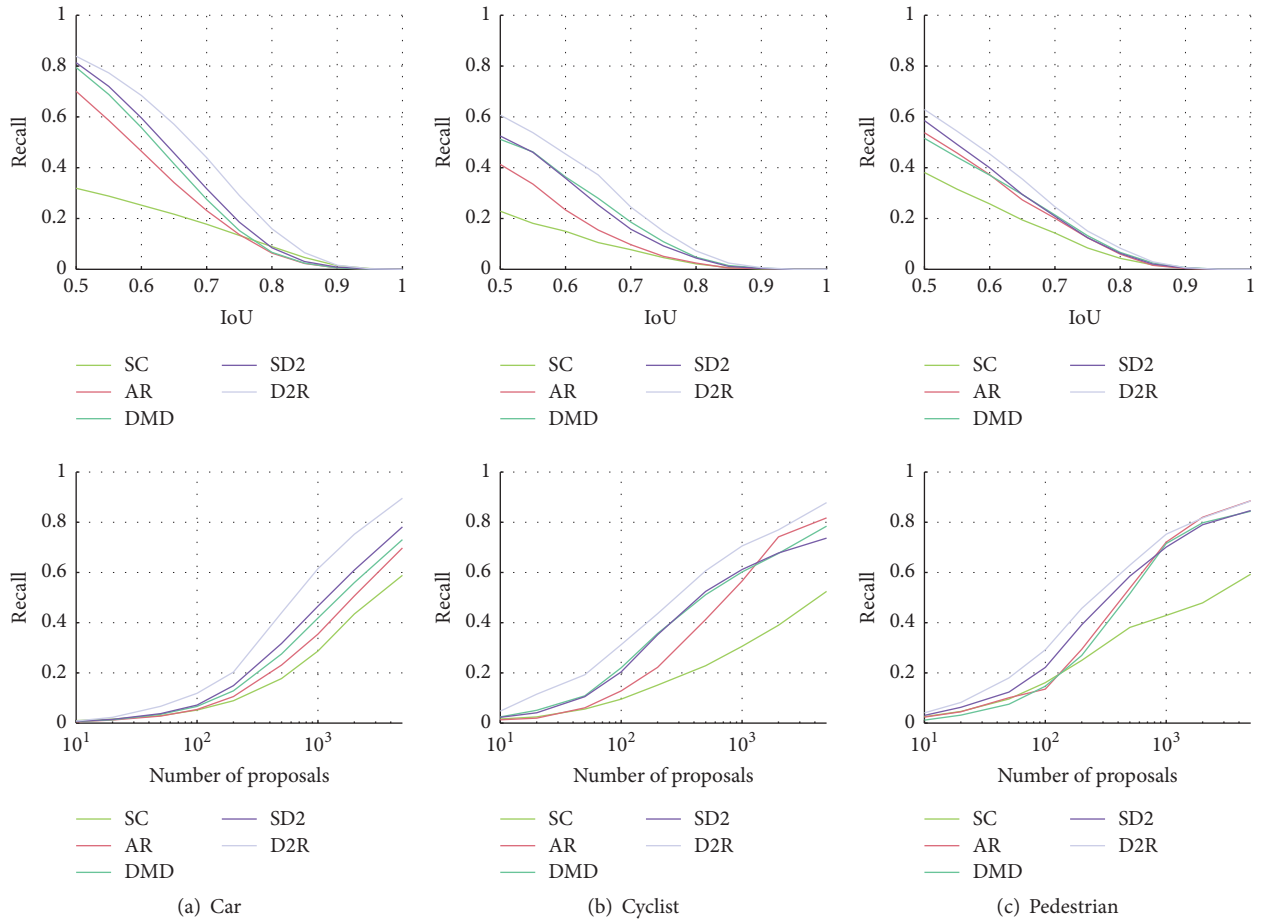

Figure 3: Single feature results: the first row is the recall versus IOU curve on 500 proposals while the second row is curve of the recall versus the number of proposals on different IOU threshold. For Car the IOU threshold is 0.7, and it is 0.5 for Cyclist and Pedestrian. We analyze the original results and the four proposed features independently to observe the usefulness of these features. We find that all the four proposed features work better than the original result when we just use a single feature to generate the proposals. With experiments on the three objects we find that D2R is the most useful feature while our proposed feature SD2 ranks second. DMD have similar performance with SD2, because they both catch the constancy of object size in real-world. AR is also a useful feature.

Table 1: Results on the hard validation sets for all three object classes. AUC is the abbreviation for Area Under the Curve, recall is the maxima recall the method can achieve, and M is the number of proposals when the recall reaches 75%. Inf means the maxima recall cannot reach 75%.

Features		Cars			Cyclist			Pedestrian		
		AUC	Recall (%)	M	AUC	Recall (%)	M	AUC	Recall (%)	M
Single features	AR	0.13	59	Inf	0.14	52	Inf	0.2	89	Inf
	SC	0.15	70	Inf	0.24	82	2209	0.29	89	1226
	DMD	0.17	73	Inf	0.27	78	3735	0.28	84	1337
	SD2	0.19	78	4426	0.27	74	Inf	0.31	85	1463
	D2R	**0.25**	**90**	**1977**	**0.33**	**88**	**1616**	**0.34**	**88**	**986**
Single features	D2R + DMD	0.39	92	682	0.43	89	832	0.46	90	307
	D2R + SD2	0.41	92	509	0.42	88	927	0.46	89	286
	D2R + DMD + AR	0.45	92	413	0.47	89	564	0.53	90	609
	D2R + SD2 + AR	**0.46**	**92**	**352**	**0.47**	**89**	**536**	0.54	89	667
	D2R + DMD + SD2 + AR	0.46	92	392	0.47	89	625	**0.54**	**91**	**129**
	All	0.45	92	423	0.44	89	568	0.52	91	234

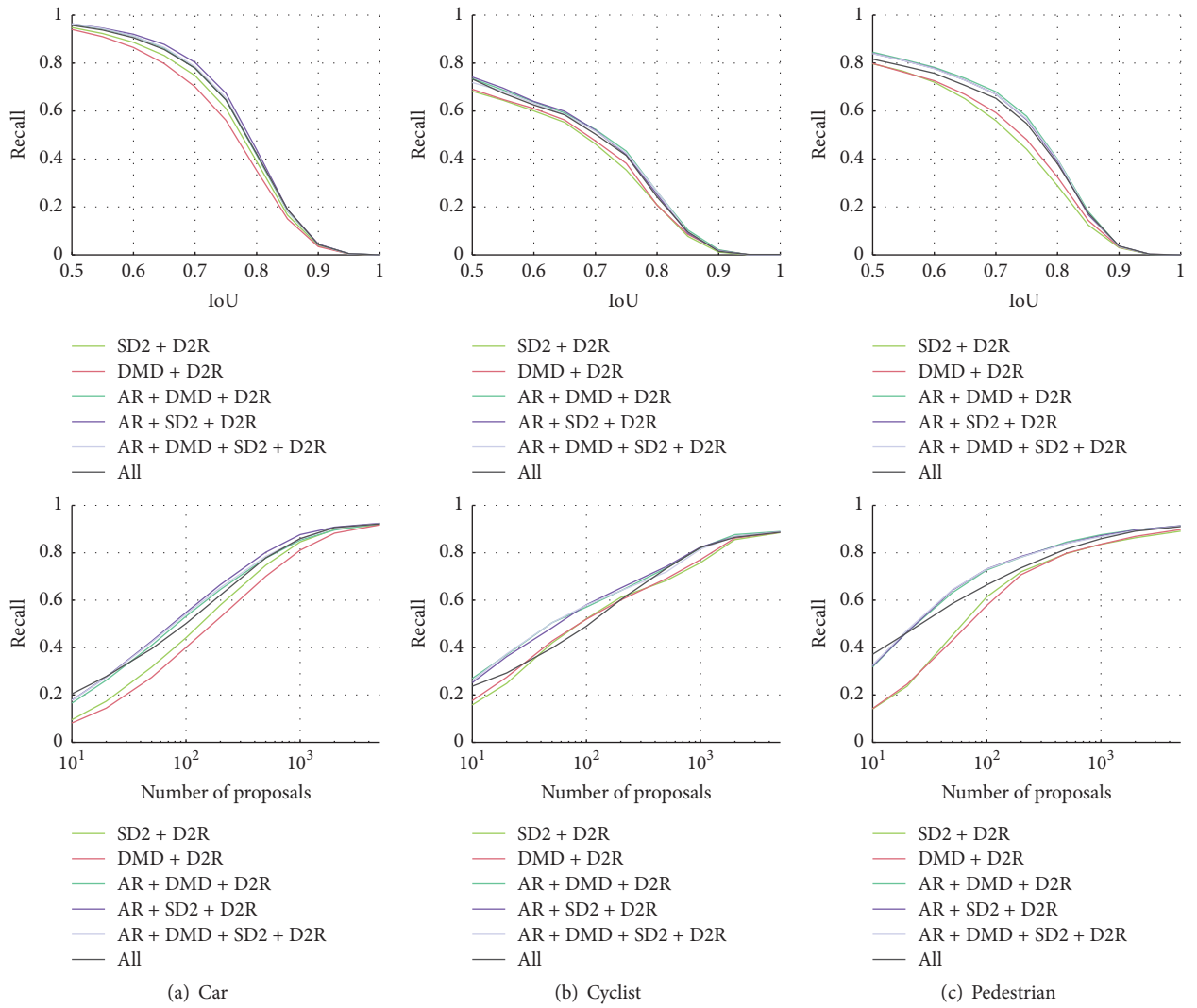

Figure 4: Features combination results: The first row is the recall versus IOU curve on 500 proposals while the second row is curve of the recall versus the number of proposals on different IOU threshold. For Car the IOU threshold is 0.7, and it is 0.5 for Cyclist and Pedestrian.

effectiveness on two representativeness methods: EB (Edge-boxes) and SS (Selective Search). Correspondingly, we name their improved versions Our-EB145 and Our-SS145, where 1 represent AR, 2 represent SC, 3 represent DMD, 4 represent SD2, and 5 represent D2R. Our-EB145 means the results obtained by fusing those three geometric features, AR, SD2, and D2R, with EB in a Bayesian framework. In the paper, we just use Our-EB instead of Our-EB145, the same to Our-SS. We also compare our results to 3DOP because it is the state-of-the-art method that exploits geometric features to generate object proposals.

Figure 5 shows recall versus IOU on 500 proposals and we can see that Our-EB and Our-SS obtain significant improvement compared to the original EB and SS. For Car, our method is better than 3DOP when the IOU is below 0.7, while, with the IOU getting larger, 3DOP obtain better results. This phenomenon also appears in Cyclist. A possible reason is that the original results are good enough when the IOU is high. However, for Pedestrian Our-EB always shows better performance than 3DOP.

Figure 6 shows recall versus the number of candidates. For Car, we can achieve nearly 90% recall when the number of candidates is 1000 for moderate and hard regimes while for easy regimes we only need 200 candidates to get the same results. However, the baseline cannot achieve 90% recall no matter how many candidates are used. For Cyclist and Pedestrian our results show similar improvements over the baselines. Compared to 3DOP our method obtains different degrees of improvements. For example, by using 100 proposals for Pedestrian our method achieves 89%, 80%, and 70% recall for easy, moderate, hard regimes while the 3DOP is around 70%, 60%, and 52%. However when the number of proposals gets larger our method achieves similar result with 3DOP.

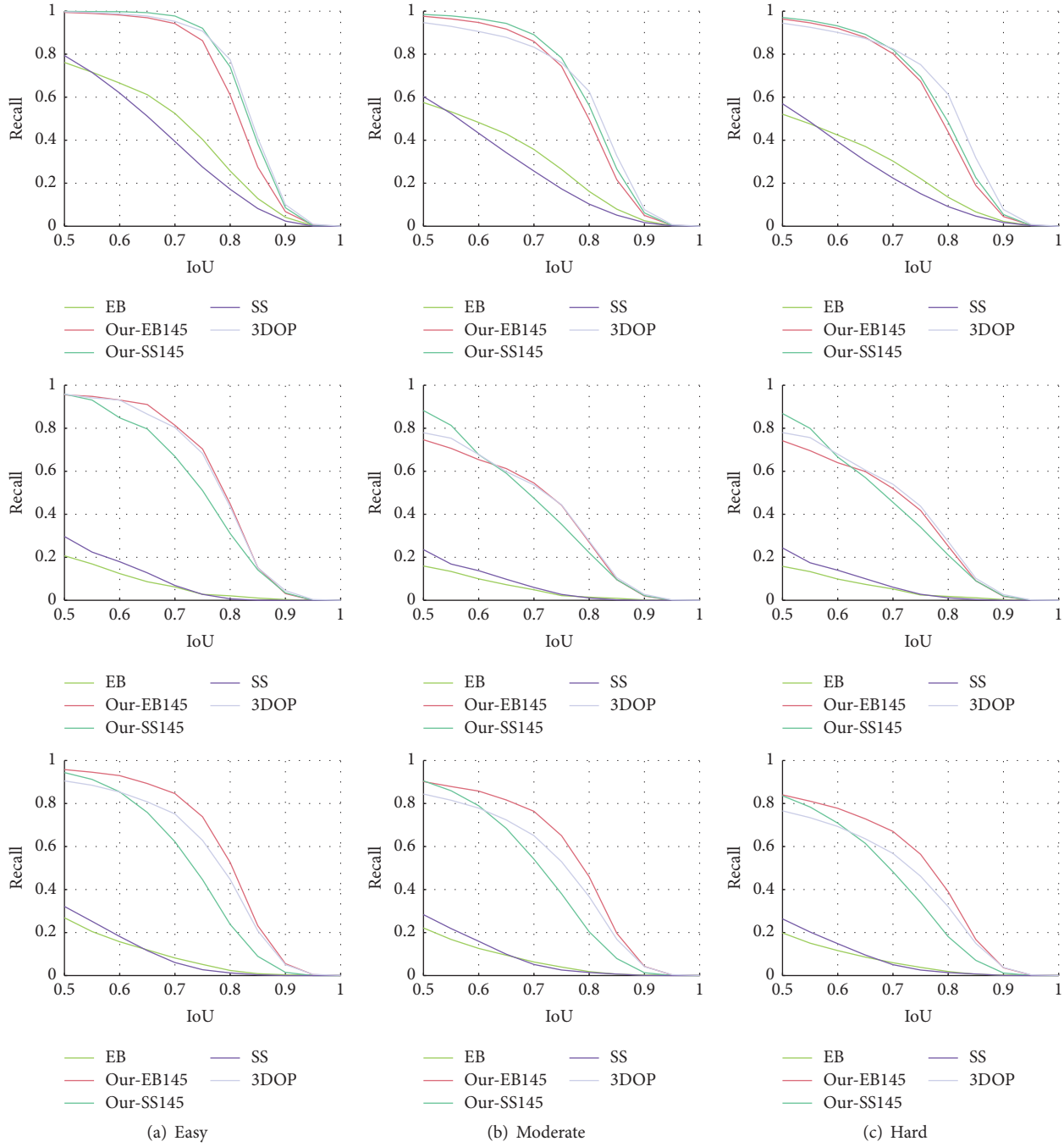

Figure 5: Recall versus IOU for 500 proposals in three regimes. From top to down: Car, Cyclist, and Pedestrian.

4.3. Running Time. Given the depth map, our features can be computed efficiently. Combined with the existing method, our approach can obtain significant improvement with only 0.2 s additional runtime on a single core by MATLAB. Table 2 shows the running time of different proposal methods.

4.4. Object Detection. To evaluate the object detection performance based on our proposal generation method, we apply the state-of-the-art fast R-CNN object detector on the bounding box proposals generated by our method, as 3DOP in [16]. We report results on the validation set of the KITTI benchmark and compare our methods (Our-EB and Our-SS) with those whose bounding box proposals are generated by Selective Search and Edgeboxes. Experiments show that the detection performance can be improved around 18% and 15%, respectively. We also compare our results with that of 3DOP. The results are presented in Table 3. Our approach can achieve comparable or better performance across all three categories.

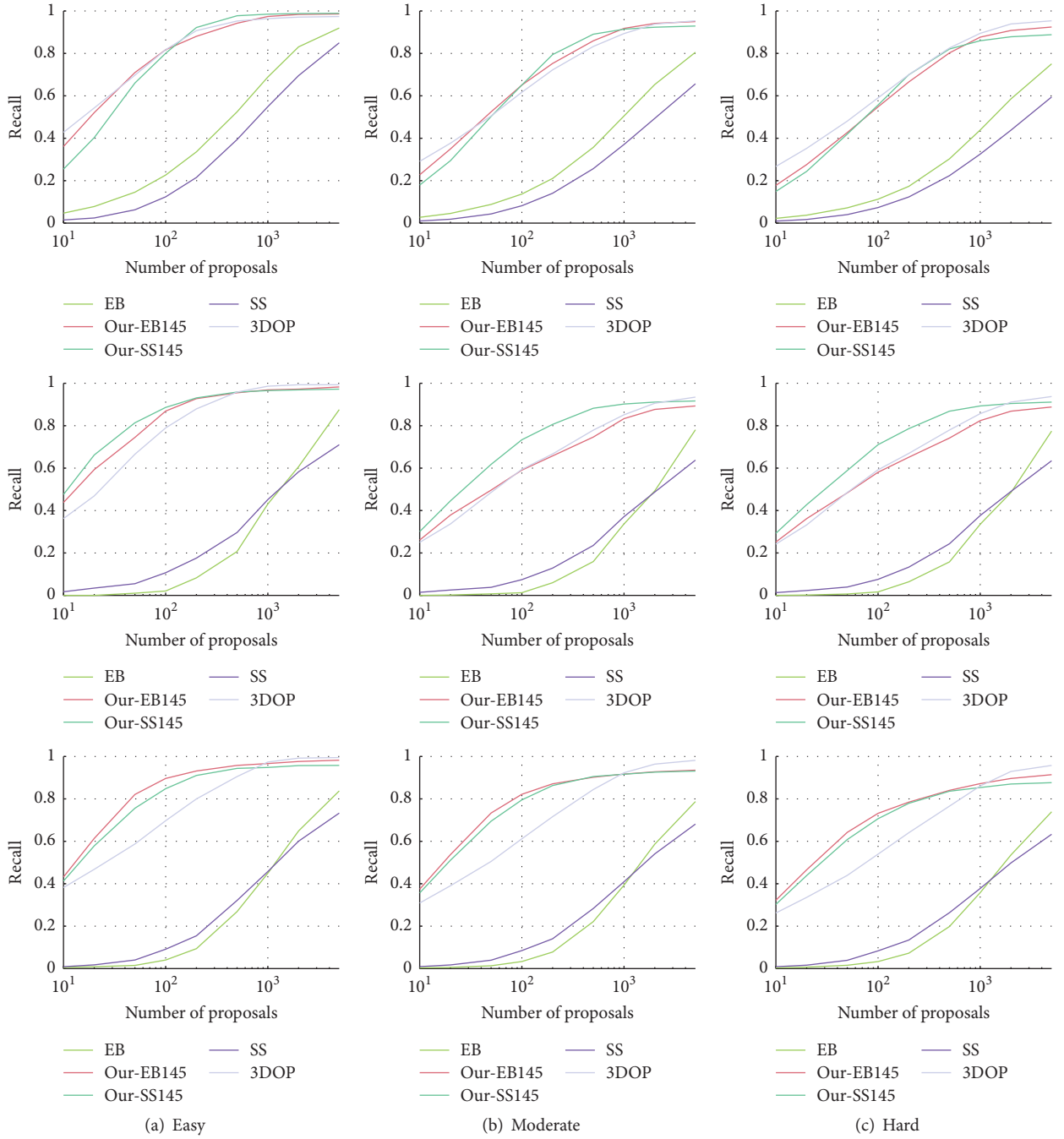

Figure 6: Recall versus number of proposals: the overlap threshold for Car is 0.7, and it is 0.5 for Pedestrian and Cyclist. From top to down: Car, Cyclist, and Pedestrian.

Table 2: Running time of different proposal methods.

Method	Selective Search	Edgeboxes	3DOP	Our-SS	Our-EB
Time (second)	15	1.5	1.2	15.2	1.7

4.5. Visual Results. The visual results of our object detection framework are shown in Figure 7. It would be best to enlarge and view it in color. The odd rows are the ground truth bounding box while the even rows are detection bounding box. Different colors indicate different difficulties. Green means not occluded, yellow means partly occluded, and red means fully occluded. Our approach produces precise detection result even for distant and occluded objects. But it more failed if the object is too distant and fully occluded, since we can not obtain enough depth or appearance information

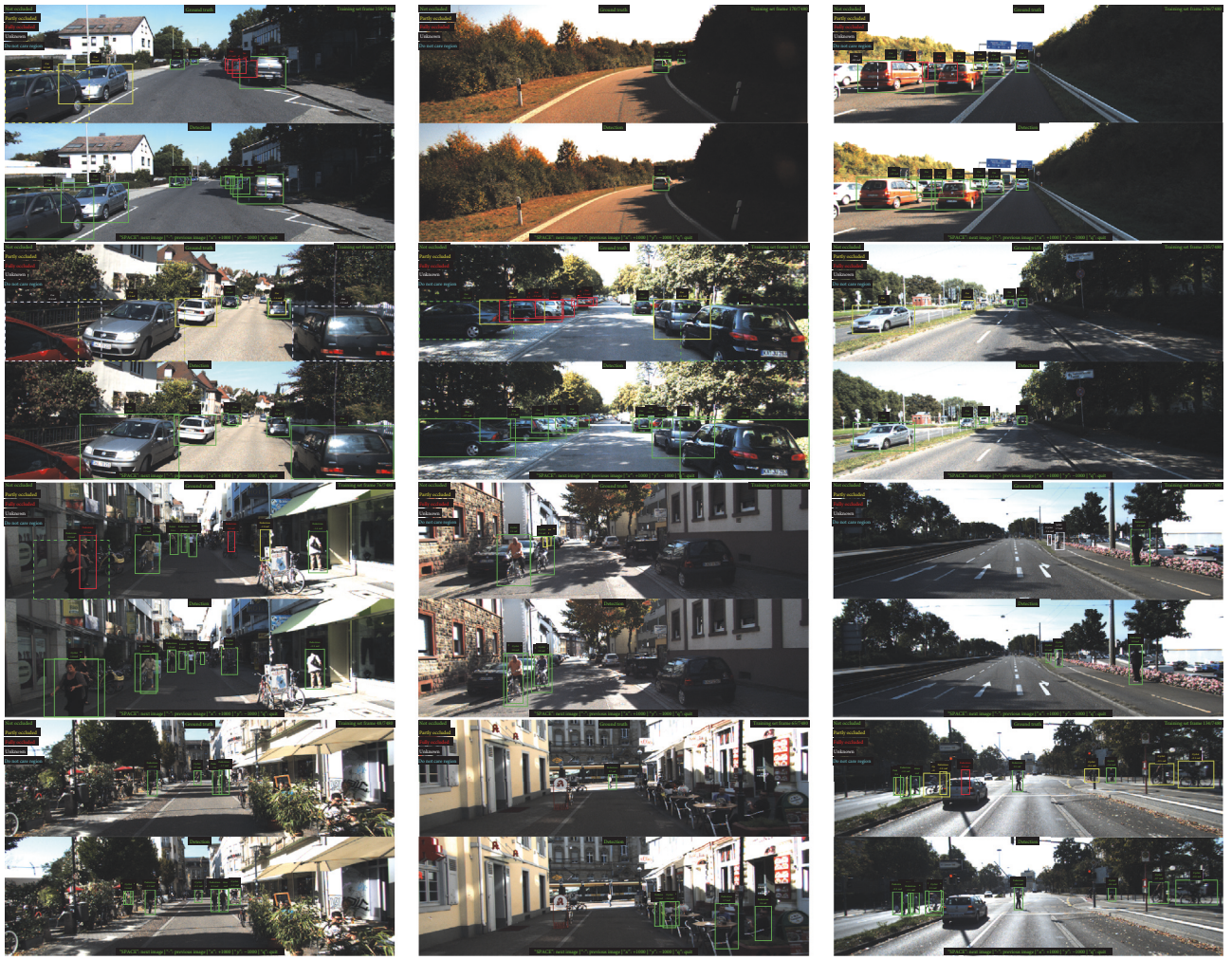

Figure 7: The visual results of our object detection framework. The odd rows are the ground truth bounding box while the even rows are detection bounding box. Different colors indicate different difficulties. Green means not occluded, yellow means partly occluded, and red means fully occluded. The first four rows are the results of Cars, while the second four rows are the results of Pedestrian and Cyclist.

Table 3: Average Precision (AP) (in %) on the validation set of the KITTI object detection benchmark with 1000 proposals, while, for EB and SS, the number of proposals is 2000.

Metric	Method	Cars			Cyclist			Pedestrian		
		Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard
AP	SS [16]	75.91	60.00	50.98	56.23	39.16	38.83	54.06	47.55	40.56
	EB [16]	86.81	70.47	61.16	55.01	37.87	35.80	57.79	49.99	42.19
	3DOP	94.47	87.09	**78.72**	84.65	57.38	55.63	72.47	65	57.24
	Our-SS	**95.36**	**87.84**	78.57	**84.71**	**57.74**	**55.8**	74.23	66.54	57.9
	Our-EB	88.92	87.40	78.43	83.38	57.72	55.69	**74.39**	**66.73**	**58.17**

for object detection. And when a person rides a Cyclist, the ground truth just has an annotation of Cyclist while our method gives two detection results, Cyclist and Pedestrian, as shown in the sixth row. People sitting in a chair are detected; however they are not marked as ground truth in the KITTI datasets.

5. Conclusion

In this paper, we propose several geometric features which are suitable for object proposals in the autonomous driving scene and integrate them with existing object proposal generation methods in a Bayesian framework. We deeply analyze

the effectiveness of each geometric feature and different combinations of features. Experiments on the challenging KITTI benchmark demonstrate that, by integrating these geometric features into existing object proposal methods, we achieve significant improvement on all three object classes. Subsequently we improve the object detection performance. Our future work will focus on integrating geometric features into a totally CNN framework for boosting their performance in the autonomous driving scene.

Conflicts of Interest

The authors declare that they have no competing interests.

References

[1] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '05)*, vol. 1, pp. 886–893, IEEE, June 2005.

[2] P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *Proceedings of the 26th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '08)*, pp. 1–8, June 2008.

[3] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627–1645, 2010.

[4] S. Ravishankar, A. Jain, and A. Mittal, "Multi-stage contour based detection of deformable objects," in *Proceedings of the European Conference on Computer Vision (ECCV '08)*, pp. 483–496, Springer, 2008.

[5] P. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.

[6] C. L. Zitnick and P. Dollár, "Edge boxes: locating object proposals from edges," in *Proceedings of the European Conference on Computer Vision*, pp. 391–405, Springer, 2014.

[7] K. E. A. Van De Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, "Segmentation as selective search for object recognition," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV '11)*, pp. 1879–1886, November 2011.

[8] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: a retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, 2015.

[9] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the KITTI vision benchmark suite," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, pp. 3354–3361, June 2012.

[10] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *Computer Vision—ECCV 2014*, Springer, 2014.

[11] P. Arbeláez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, "Multiscale combinatorial grouping," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 328–335, IEEE, June 2014.

[12] J. Carreira and C. Sminchisescu, "Constrained parametric min-cuts for automatic object segmentation," *IEEE Transactions on Software Engineering*, vol. 23, no. 3, pp. 3241–3248, 2010.

[13] B. Alexe, T. Deselaers, and V. Ferrari, "What is an object?" in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '10)*, pp. 73–80, June 2010.

[14] B. Alexe, T. Deselaers, and V. Ferrari, "Measuring the objectness of image windows," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2189–2202, 2012.

[15] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300fps," in *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR '14)*, pp. 3286–3293, June 2014.

[16] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3d Object proposals using stereo imagery for accurate object class detection," https://arxiv.org/abs/1608.07711.

[17] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from RGB-D images for object detection and segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV '14)*, pp. 345–360, Springer, 2014.

[18] A. Janoch, S. Karayev, Y. Jia et al., "A category-level 3-D object dataset: putting the kinect to work," in *Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCV '11)*, pp. 1168–1174, November 2011.