Find Tiny Instance Segmentation

Yueqing Zhuang*
Peking University
zhuangyq@pku.edu.cn

Zeming Li*
Tsinghua University
lizm15@mails.tsinghua.edu.cn

Gang Yu
Megvii Inc. (Face++)
yugang@megvii.com

Abstract

Autonomous driving is an accurate application which is based on computer vision and multi-sense. Scene parsing is a comprehensive analysis of an image need by autonomous driving. As the importance of autonomous driving' security, pixel-accurate environmental perception in computer vision is expected to be exploited. Unlike other applications such as intelligent surveillance, the inaccuracy of perception system which leave out tiny objects would lead to disaster. As tiny-scale objects are hard to detect and segment, in this paper, we exploit a more accurate Tiny Instance Segmentation (TIS) adapted to autonomous driving to get precise boundary for tiny object, which has got 1st place in WAD competition. Moreover, extensive experiments show the effectiveness of each components.

1. Introduction

Instance segmentation, which assigns pixel-wise mask for each object, is one of fundamental computer vision tasks. This task is important for intelligent surveillance, autonomous driving, robotics and so on.

With the development of deep convolutional neural networks, which dominates on computer vision, several solutions were proposed to handle this task. Though detector like MegDet [1] and Light-Head RCNN [2] have improved accuracy and speed a lot at the COCO dataset [3], it's more significant to predict the contour of an object according to specific characteristics of autonomous driving. Mask R-CNN [4] is a typical solution to solve this problem. However, autonomous driving is an accurate application which needs precision for detecting an object especially for tiny objects, while common object detector would fail in this case.

Several released dataset like COCO[3], CityScapes[5] have a large room for improvement. However, none of them aims at autonomous driving. Due to the particularity of autonomous driving, Apollo dataset[6] was released to solve this task. CVPR Workshop on Autonomous Driving uses part of Apollo dataset, whose small objects are dominant (about 60% of objects' scale is less than 32 pixels as Figure 1).

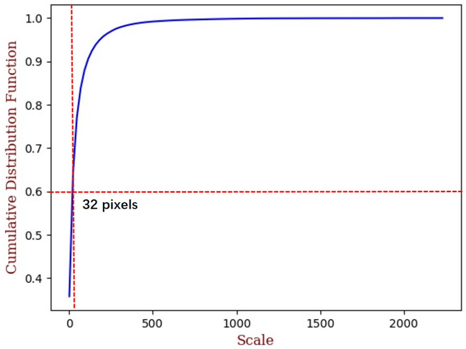

Figure 1. Cumulative distribution function with respect to scale. About 60% scales of objects is under 32 pixels in WAD dataset (Small Objects is the majority of WAD dataset).

In this paper, to address the challenge in autonomous driving, we exploit a better and effective Tiny Instance Segmentation (TIS) to find small scale's object in an image for autonomous driving. Our final submission is an ensemble of 3 models and got 1st place in WAD competition.

2. Methods

2.1. Better Anchor Design

Our method is based on Mask-RCNN [4]. Different from [4], to hand large scales' objects, we adding max pooling to generate an extra RPN feature map at top of region proposal feature as Fig. 2. In this way, network has power to detect large scale object (Fig. 1). To make the best use of ground-truth/anchor ratios and make same-size's proposals only have one choice to pool feature (In FPN[7], each scale of RoIs pools feature from unique stages of ResNet), we design a *Pyramid Anchor (PA)*, whose size of anchor is $8 * [[1,2],[4,8],[16,24],[32,64],[128,256]]$ so that each stage of RPN in FPN has 2-scale anchors, which promise to generate enough small size's anchor to cover small objects.

*The first two authors are equally contributed. Work down while interning ad Megvii.

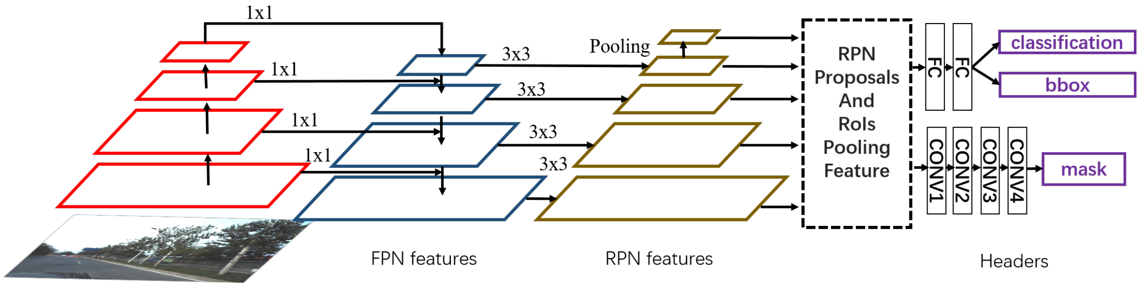

Figure 2. Visualization of our re-implemented Mask R-CNN baseline.

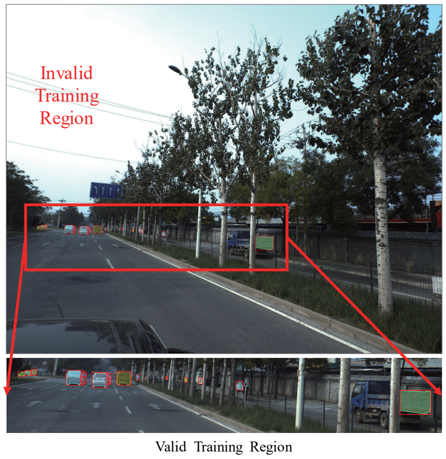

Figure 3. The valid region which contains objects is cropped for training. *VRT* refers to *Valid Training Region* at training proceed.

2.2. Valid Training Methods

To handle small objects in the dataset, we found it difficult to use full image training dataset even we use sublinear memory[8] to save GPU-Memory. It's obvious that using large-scale image can improve results (eg. Table 1) as small object's activation may lost in high-level feature because of image resizing. Another interesting observation is that large-area region in an image doesn't contains any objects (sky and trees and so on eg. Fig.3), it wastes training memory and time to make network learning these backgrounds. Discarding all backgrounds may cause object-similar patches is recognized to another object by mistake. It's necessary to adding some false-samples in training proceed. Therefore, we randomly sample *Valid Training Region* and *Full Image* to make our network learn more diversity.

3. Experiments

3.1. Dataset and Local Validation

The dataset of this challenge is part of Apollo dataset[6], which consist of 39222 images for training (19 videos * 2) and 1917 images for testing (12 short videos). For simplicity, we choose 3 videos (which is including 4622 images and from different roads) for local validation apart from training data. Our experiments are partly based on training-sub dataset (except local validation from training dataset), which is used mAP metric in COCO dataset [3] for checking the effectiveness of our methods. In the end, we use all training data to train our model and submit to kaggle's server to get final results.

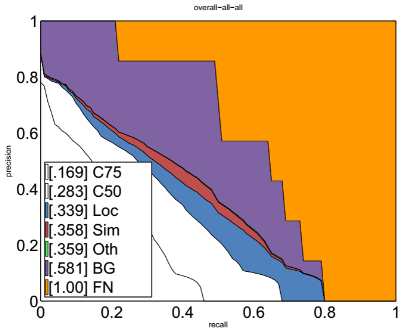

Figure 4. Analysis of coco pre-trained Mask R-CNN on local validation set. Loc: PR at IoU=.10 (localization errors ignored, but not duplicate detections). Sim: PR after supercategory false positives (fps) are removed. Oth: PR after all class confusions are removed. BG: PR after all background (and class confusion) fps are removed. FN: PR after all remaining errors are removed (trivially AP=1). This figure shows Mask R-CNN's recall is low.

For a practical deep learning system, the devil is always in the details. We use the same set of hyper-parameters as in Mask-RCNN[4], except learning rate schedule. For COCO pre-trained model, we train our model for 5 epochs with learning rate 0.01, and another 4 epoches with learning rate 0.001. For ImageNet pre-trained model, we set learning rate 0.01 at first 12 epoche, and decreases to $1/10$, $1/100$ at $12, 15$ epochs respectively.

We use 8 images (1 image per GPU) in one image batch, $Valid\ Training\ Region\ (VTR)$ and $Full\ Image\ (FI)$ are randomly chosen, whose shorter edge randomly sampled from $\{1500,\ 1800\}$ and longer edge is set to 3384. Thanks to sublinear memory technology [8], we can train our network with the limitation of 11G-memory such as 1080ti.

Table 1. The influence of training scale in Mask R-CNN for WAD dataset. *VTR* means only using *Valid Training Region* at training stage. Results below is mask AP(%).

$Network$	$Scale$	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l	mAP
Mask R-CNN [COCO]	[1500, 1800]	28.1	16.8	7.5	26.9	36.8	16.9
Mask R-CNN [COCO]	[1800, 2400]	29.9	18.3	9.2	27.5	40.0	18.3
Mask R-CNN [COCO]	VTR	30.3	19.5	9.1	30.5	42.8	18.9

Table 2. The roadmap to final model in Mask R-CNN under backbone of ResNet-50. *PA* represents *Pyramid Anchor*, *VTR* means using *Valid Training Region* in training proceed. *FI* denotes adding *Full image* at training stage to learn background as well. Results below is mask AP(%), backbone is based on ResNet-50.

$Network$	$COCO$	PA	VTR	FI	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l	mAP
Mask R-CNN					25.4	15.1	7.1	23.1	35.9	15.2
Mask R-CNN	✓				28.1	16.8	7.5	26.9	36.8	16.9
Mask R-CNN	✓	✓			29.9	18.3	9.2	27.5	40.0	18.3
Mask R-CNN	✓	✓	✓		30.3	19.5	9.1	30.5	42.8	18.9
Mask R-CNN	✓	✓	✓	✓	34.3	21.5	10.6	33.1	44.0	21.2

3.2. Experiments Results

We first train our network using our re-implemented Mask-RCNN baseline, the scale in this setting is set to $\{1500,\ 1800\}$. As Table 3 shows, compared with ImageNet pre-trained model, COCO pre-trained model outperforms by 1.7 points.

The character of apollo dataset is that minor scale of objects is under 8-pixels while max scale of objects is 75% of image in images. (bus is ahead from camera). As Figure 4 shows, recall in COCO pretrained Mask R-CNN is quite low. In Table 3, we use two setting to improve recall of Region Proposal Network. *CA* represents *Cluster Anchor*, in which we use K-means to cluster 5 anchor scale $7 * [2, 8, 18, 38, 80]$ in rpn anchor size. In this way, we can improve recall on Region Proposal Network to increase mAP by 0.3. However, by analyzing Average Recall in local validation, we found recall is also not enough. As table 3 shows, our special *PA* is more suitable for improving results.

Table 3. The design in Mask R-CNN for WAD dataset. *CA* means *Cluster Anchor*, *PA* is equal to *Pyramid Anchor*. Results below is mask AP(%).

$Network$	mAP_{50}	mAP_{75}	mAP_s	mAP_m	mAP_l	mAP
Mask R-CNN	25.4	15.1	7.1	23.1	35.9	15.2
+ [COCO]	28.1	16.8	7.5	26.9	36.8	16.9
+ CA	28.6	17.0	7.4	28.0	40.0	17.2
+ PA	29.4	17.8	8.2	27.9	40.2	17.7

For submission, we use all training data (39222 images) to train our model with different backbone such as ResNet [10], SENet [11]. It is noticed that we replace the first *7x7* convs with two consecutive *3x3* convs as PSPNet [9] in the final model, which makes small-scale objects sensitive to be detected because *7x7* may smooth small objects resulting in blur of feature map. Table 2 shows our roadmap to best result on local validation. For single ResNet-PSPNet-152, tiny objects and crowd objects(Fig 5 and Fig 6)can be detected.

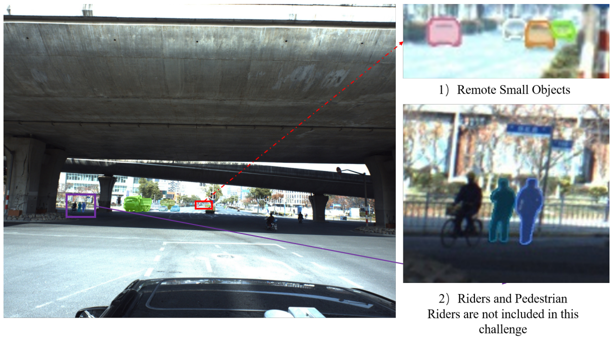

Figure 5. Visual results on remote scene using single ResNet152-PSPNet. Small cars (less than 8 pixels) can be detected as well. At the same time, our detector is able to distinguish pedestrians from riders.

The top performance comes with a few details. For testing, multi-scale testing, horizontal flip testing, bounding box voting in [4] was adopted. For multi-scale testing, we set longer edge of image to 8000 and other ranges from 1810 to 4510 with step 300. For bounding box voting, nms $thresh$ is set to 0.5 and $merge\_thresh$ is set to 0.9. As Table 4 shows, our final submission gets 33.9% in the leaderboard and we have got 1st place in this challenge.

4. Conclusion

In CVPR Workshop on Autonomous Driving, we design a new specifical Tiny Instance Segmentation (TIS) and new training strategy to detect and segment small objects for autonomous driving so that got 1st place exceeding 2nd place

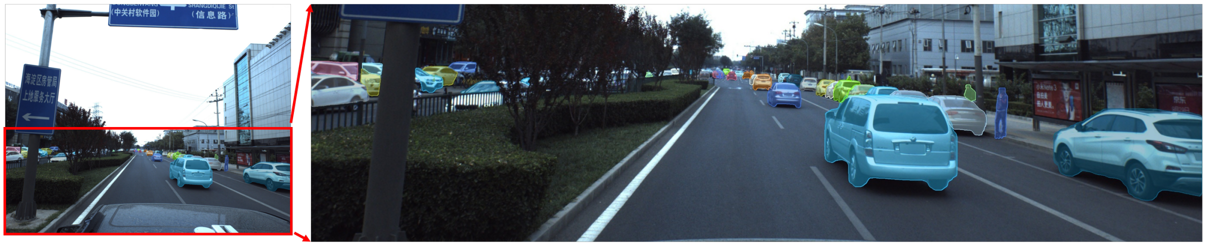

Figure 6. Visual Result on Crowdy Scene.

Table 4. Our Results on WAD testing data. *MS* represents *Multi-Scales Testing*

$Network$	$Backbone$	mAP
Mask R-CNN	ResNet-50	26.7
Our TIS	ResNet-50	29.0
Our TIS	SENet-152	31.9
Our TIS + MS	ResNet-PSPNet-152	32.4
Our TIS + MS	2*ResNet-PSPNet-152	32.8
Our TIS + MS	+SENet 152	**33.9**
Second in leaderboard	unknown	30.2
Third in leaderboard	unknown	26.7

by a large marge (Tables 4). In the future, to address real problem on autonomous driving, we will exploit information of time-continuous information and combination with multi-sense data such as depth map, point clouds and so on.

References

[1] Chao Peng, Tete Xiao, Zeming Li, Yuning Jiang, Xiangyu Zhang, Kai Jia, Gang Yu, and Jian Sun. Megdet: A large mini-batch object detector. *CoRR*, abs/1711.07240, 2017.

[2] Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head R-CNN: in defense of two-stage object detector. *CoRR*, abs/1711.07264, 2017.

[3] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[6] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *CoRR*, abs/1803.06184, 2018.

[7] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[8] Tianqi Chen, Bing Xu, Chiyuan Zhang, and Carlos Guestrin. Training deep nets with sublinear memory cost. *CoRR*, abs/1604.06174, 2016.

[9] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2881–2890, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. *arXiv preprint arXiv:1709.01507*, 2017.