CVPR 2018 WAD Video Segmentation Challenge

Second Place Solution

Super Camera

1. Overview

The goal of this Kaggle Challenge is to accurately segment objects such as car and pedestrian at instance-level from the video sequences recorded by the autonomous driving car. Our solution is based on Mask R-CNN[5]. We use both Apolloscape and Cityscape datasets and only focus on a particular field of view in the image at the highest possible resolution. The implementations and pretrained model are available on `https://github.com/Computational-Camera/`

2. Dataset

2.1. Classes and annotations

There are several autonomous driving open databases available for instance-level semantic labelling such as KITTI[4], Cityscape[3], Apolloscape[6] and recent released Berkeley Deep Drive[8]. Besides the Apolloscape `http://apolloscape.auto/scene.html` which currently provides 63, 632 frames near 4K resolution instance-level labelled images over 4 different urban roads, we also use Cityscape dataset for training our model as it improves the diversity of the classes, specifically the pedestrian class.

In this competition, seven different instance-level annotations are evaluated including motorcycle, bicycle, pedestrian, truck, bus, and tricycle. Considering the video frames from Apolloscape is densely sampled, we only use one over sixth of total samples to reduce the training time. The number of annotations and portions in different datasets are shown in Tab.1 and Fig.2.

2.2. Statistical Analysis

The image resolution of Apolloscape is 3384 by 2710. However, our interesting objects are not uniformly distributed over the image. Actually, as revealed in the 2D histogram in Fig.21, the distribution is densely concentrated in a narrow stripe of the whole field of view. For example, between $y = 1560$ to $y = 2280$, 99.7% of our interesting objects are localized in this region of interest (ROI). This is our key observation.

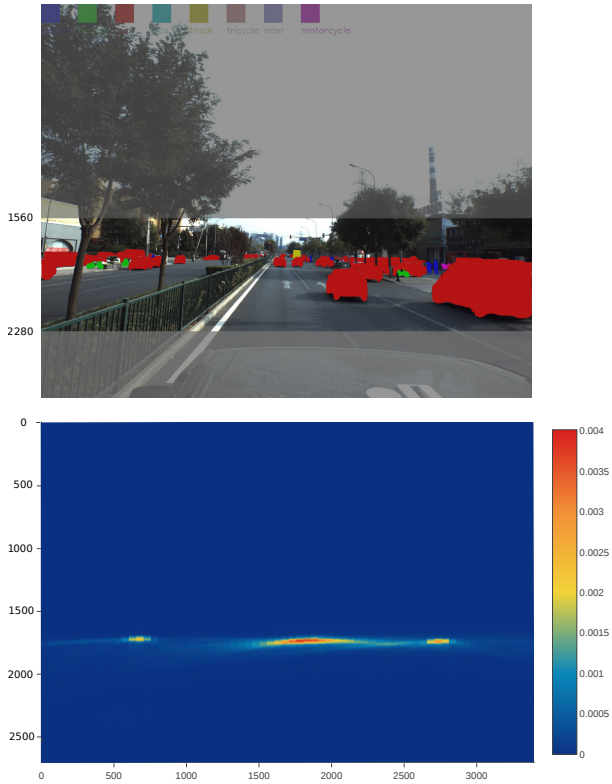

Figure 1. Top, Our prediction result example in the busy urban street. Bottom, The statistics of annotations in Apolloscape.

Class	Apolloscape	Cityscape	Our Merged
Person	241779	21413	68071
Bicycle	27583	4904	10023
Car	653513	31822	152346
Motorcycle	37743	888	8346
Bus	16729	483	3750
Truck	36173	582	7941
Tricycle	36069	0	7445

Table 1. The number of annotations in different datasets.

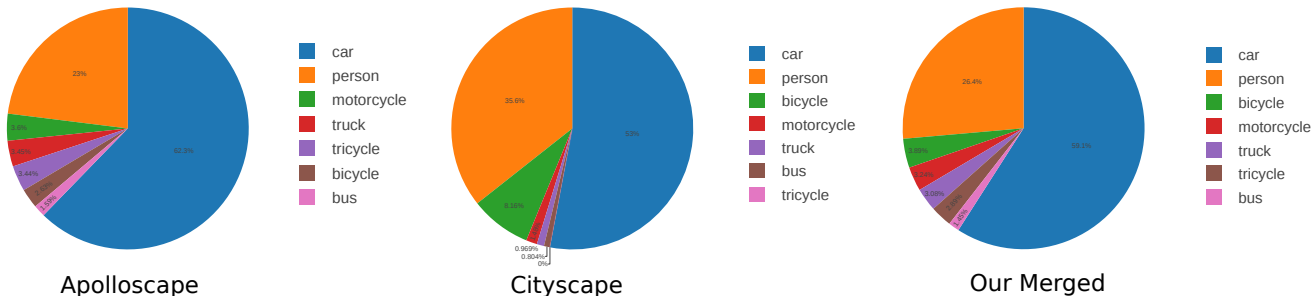

Figure 2. Instance portions in different datasets

3. Instance Segmentation

The core algorithm in our solution is Mask R-CNN[5]. There are a few open source implementations on the shelf, we use Detectron `https://github.com/facebookresearch/detectron`, the object detection library developed by the Mask R-CNN original authors Kaiming He, Piotr Dollar, Ross Girshick from Facebook AI Research team.

As we learned from the previous Kaggle image segmentation competition[1], to achieve better performance, higher image resolution for training and inference is always preferred. However, due to the GPU memory size constraint and our statistical analysis in the previous section, we crop and resize the image to 3360 by 720. With a Nvidia P100 graphical card, training a 9-class Mask R-CNN FPN model with ResNet X101 backbone network requires around 1560 MB GPU memory when the batch size is 1. However, to have a reliable solution, a full field of view prediction at a lower resolution should be considered to recognize those objects that are not in our ROI coverage. In practice, we find it does not improve the performance.

Another interesting finding is the threshold for the object detection. Through experiment, we found that 0.25 is the optimal threshold to obtain the highest leader board score.

We use the pretrained model from COCO[7] dataset provided by the Detectron model zoo`https://github.com/facebookresearch/Detectron/blob/master/MODEL_ZOO.md`.

The loss and precision curves over iterations are shown in Fig.3. We use the training result after 70,000 iterations (nearly 6 epochs). The initial learning rate is 0.001. The whole training process takes around 30 hours with a single Nvidia P100 GPU.

The individual class mean average precision (mAP) score is shown is in Tab.2.

4. Further Work

Due to the tight schedule, we were not able to explore the direction of improving the prediction result using the temporal information among the video frames, similar to the recent work by Chen et.al[2]. I think this direction is promising, as the detection error introduced by the perspectives, the scales of the object can be reduced by the temporal information.

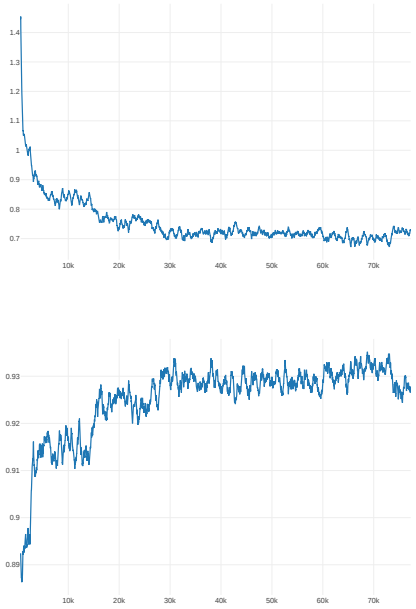

Figure 3. The loss and precision curves over iterations. (a) Loss Curve. (b) Precision Curve.

Class	Private Leader Board
Car	0.06799
Bus	0.06421
Person	0.05243
Truck	0.04773
Bicycle	0.02817
Tricycle	0.01810
Motorcycle	0.01320
Total	0.30571

Table 2. The individual mAP of 7 evaluated classes.

References

[1] Carvana image masking challenge. https://www.kaggle.com/c/carvana-image-masking-challenge, 2017.

[2] K. Chen, J. Wang, S. Yang, X. Zhang, Y. Xiong, C. C. Loy, and D. Lin. Optimizing video object detection via a scale-time lattice. In *CVPR*, 2018.

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. CVPR, 2016.

[4] A. Geiger, P. Lenz, and R. Urtasun. Kitti vision benchmark suite. http://www.cvlibs.net/datasets/kitti/, 2012-2015.

[5] R. Girshick, G. Gkioxari, P. Dollár, and K. He. Mask R-CNN. ICCV, 2017.

[6] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang. The apolloscape dataset for autonomous driving. arXiv: 1803.06184, 2018.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollr, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[8] F. Yu. Deepdrive berkeley. https://deepdrive.berkeley.edu/, 2018.