Instance Segmentation with Mask R-CNN for Self-Driving Vehicles in Urban Environments

Xiao Wei & Shangyin Gao & Sen Wang & Fengze Han

Abstract

Our objective is to investigate the state-of-art neural network Mask R-CNN for instance segmentation specifically in the urban self-driving scenarios. We adopted transfer learning to train on Cityscapes dataset. To improve the segmentation performance, we looked into the network architecture, trained our model in multiple steps and fine-tuned with various hyper-parameters. Inspection results were concluded. Reducing the mask loss is our biggest challenge and more efforts are required in future research.

Dataset

Mask R-CNN is orignially trained on Microsoft COCO dataset by facebook. In our project, we use Cityscapes dataset instead as it offers various street view for urban self-driving tasks. Main differences between these two datasets are shown in Table 1.

Dataset	Classes	Image size	Scenes	Image number
Cityscapes	35	2048×1024	street	3k
COCO	80	multiple sizes	different	35k

Table 1: Difference between Cityscapes and COCO

Figure 1: Left image from COCO, right from Cityscapes

8 categories of objects of our interest are shown in Table 2. They are vital for our detection task for self-driving cars in urban scenarios. Also, they appears in both datasets.

person	rider	car	truck	bus	train	m-cycle	bicycle
17.9k	1.8k	26.9k	0.5k	0.4k	0.2k	0.7k	3.7k

Table 2: Picture number on each class

Problems encountered: some false labels in the Cityscapes can lead training loss to increase

Mask R-CNN Network Architecture

- ResNet101 and Feature Pyramid Networks (FPN) backbone
- Parallel head structure: bounding box classification and mask regression
- Structure detail shows in Figure 2 and Figure 3

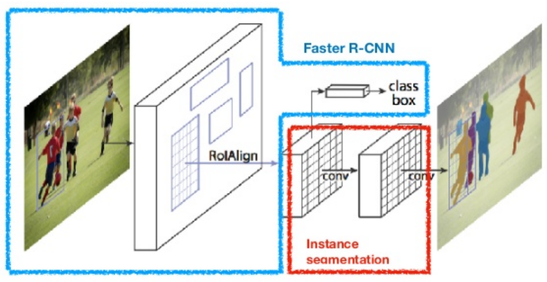

Figure 2: Structure of Mask R-CNN

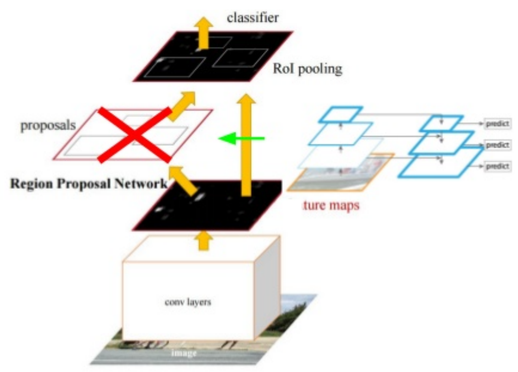

Figure 3: Structure detail of FPN

Model Training

Training Schemes

- Transfer Learning Characteristics:
  Data domain: from general scenarios to urban driving scenarios.
  Task domain: instance segmentation (stays the same).
- $Loss = Loss_{classification} + Loss_{boundingbox} + Loss_{mask}$
- Optimizer: Stochastic Gradient Descent
- Pre-training:
  Pre-trained Mask R-CNN with ResNet101 and Feature Pyramid Networks backbone on COCO by matterport[1]. The first 4 stages of ResNet101 backbone were fixed during the whole training process, providing good feature extraction.
- Multi-Stage Training:
  Step 1: Fix the bounding box prediction and classification head, and train from ResNet101 5th Stage up to the top for 5 epochs with learning rate as 0.01.
  Step 2: Train from ResNet101 5th Stage up to the top for 10 epochs with learning rate as 0.005.
  Step 3: Train the same layers and decrease the learning rate to 0.001.

Experiment Inspection

- Learning Curves
  In order to validate our training pipline, we first tried to overfit the model on a randomly choosen image with 100 iterations. The corresponding learning curve is shown in Figure 4. Training loss quickly converged, especially the classification and bounding box loss. However, validation loss kept increasing apart from the bounding box loss. This indicates that the pre-trained network is capable of generalizing the bounding box prediction to some degree.

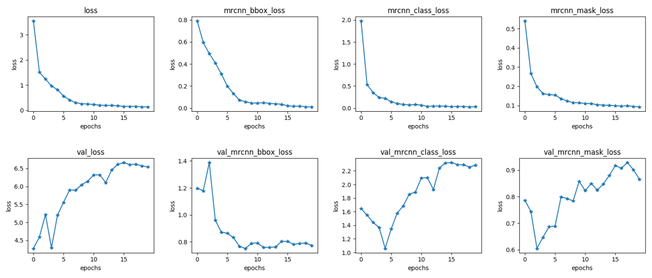

Figure 4: Overfitting on a Single Image

Next, we trained our model on a larger dataset, i.e. 100 images, for 2000 iterations. The result is shown in Figure 5. In contradiction to the overfitting curve, the validation losses also dropped. However, as the the training data and iterations were limited, validation losses still remained at a relative high level.

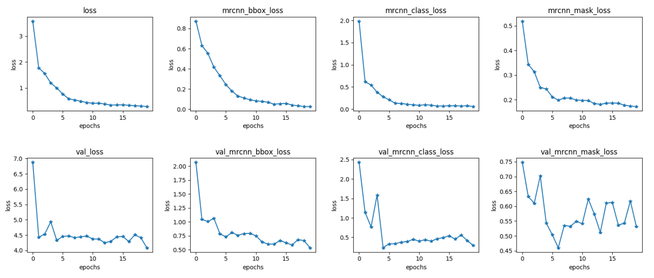

Figure 5: Training on 100 Images for 2000 Iterations

Last but not least, having explored various training schemes, we trained our model on the whole training dataset for 7200 iterations. The learning curve is shown below in Figure 6.

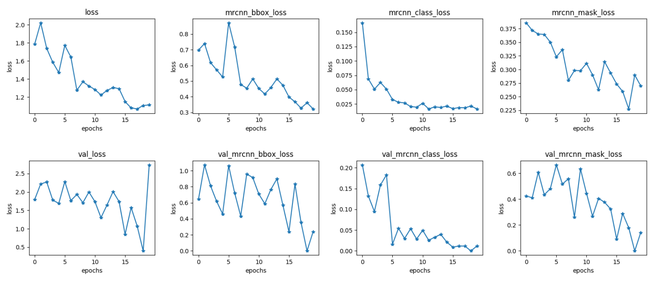

Figure 6: Training on the Entire Training Set for 7200 Iterations

- RPN Anchor Scales
  As Figure 7 shows, big RPN anchor scale tends to detect big object and missing some detail (small cars in the middle of left image). For a small anchor scale, we can detect better small object, which is useful in Cityscapes dataset.

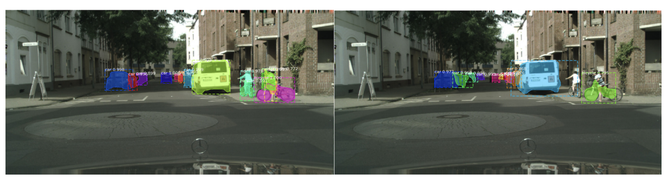

Figure 7: Left image with a high RPN anchor scale, right with a how anchor scale

- RPN NMS Threshold
  As Figure 8 shows, if we use a low NMS threshold, we will lose a lot of proposals that we interested (rider and bicycle). Instead, if we set a relative high threshold, we will have more region proposals which can cover the hold image better.

Figure 8: Left image with a low NMS threshold, right with a high NMS threshold

Conclusions

Efforts Made to Improve the Network:

- Keep the input images in original size rather than down sampling by half, as otherwise some masks will disappear because of resizing.
- Reconstruct the loss function by adding punishment term on the mask loss.
- Fix different parts of the network (bounding box and classification head, mask head) respectively in different multi-stage training schemes in order to reduce the mask loss.
- Random search for learning rate in the range of [0.001, 0.02] on small size training dataset.
- Test different RPN anchor scales.
- Test different thresholds for non-maximun suppression of RPN.
- Add one fully convolutional layer on the Mask head.

Evaluation on Test Set

	training data	AP[val]
InstanceCut [3]	fine & coarse	15.8
DWT [2]	fine	19.8
SGN [4]	fine	29.2
MASK R-CNN	fine	31.5
MASK R-CNN	fine & COCO	36.4
MASK R-CNN(original architecture)	fine & COCO	10.8
MASK R-CNN(added one Conv layer)	fine & COCO	12.6
MASK R-CNN(original architecture, overfitted)	Single Image	58.2

Table 3: Validation results on Cityscapes

Table 3 compares our results with the state-of-art on Cityscapes validation set. With only 7200 iterations of training, our best model yielded 12.6 AP.

Example Output

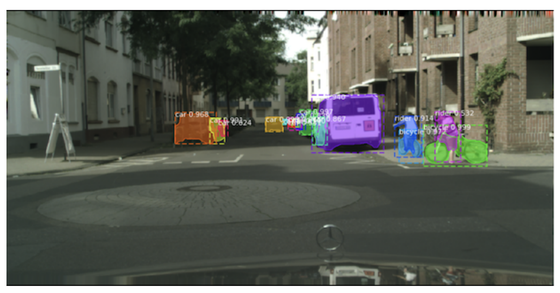

Figure 9: Example output

Summary

Our model achieved good performance with only 7200 iterations on bounding box prediction and classification. However, the mask prediction remained unsatisfying, which is the main cause for relatively low AP. Through experiments, we concluded that some objects that should be detected were ranked with low probabilities among all ROIs. They are further ignored in later detection process. We designed multi-stage training schemes and tuned hyper-parameters and network structure accordingly to cope with this issue. The limit of our model is yet to be determined until large scale training is performed.

References

[1] *https://github.com/matterport/mask_rcnn*. *Github*.

[2] M. Bai and R. Urtasun. Deep watershed transform for instance segmentation. *CVPR*, 2017.

[3] A. Kirillov and E. Levinkov. Instancecut: from edges to instances with multicut. *CVPR*, 2017.

[4] S. Liu and J. Jia. Sgn: Sequen- tial grouping networks for instance segmentation. *ICCV*, 2017.