Image Segmentation Kaggle Challenge

Yokila Arora, Hubert Lu, Eley Ng

{yarora, hubertlu, eleyng}@stanford.edu

Motivation

We perform **object segmentation** in the CVPR 2018 Kaggle Competition for autonomous driving using images taken from a car camera. In our approach, we have adopted the **Mask R-CNN** model for this task. Assisted by transfer learning via pre-trained weights from the COCO dataset, we have trained the Mask R-CNN model and successfully predicted objects in the image.

Data

- **Images** (provided by Baidu, Inc.)
  - 39,222 RGB images
  - Their corresponding labels (i.e. 39,222 masks)
  - 7 target classes, 6 of which overlap with COCO dataset classes
- **Preprocessing**
  - Downsized from 3384×2710 to 1024x1024 (same aspect ratio)
  - Mapped the object classes from the labelled images to the number of classes
  - For images without classes, an empty mask is applied

Table 1. Train, Dev, and Test Split

Train	Dev	Test
99%	1%	1917 images

References

[1] He, Kaiming, et al. "Mask r-cnn." *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017.

[2] "Cvpr 2018 wad video segmentation challenge," https://www.kaggle.com/c/cvpr-2018-autonomous-driving, accessed: 2018-05-12.

[3] "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask RCNN, accessed: 2018-05-12.

Model

- Mask RCNN is the most recent addition to the class of R-CNN models, Mask R-CNN inherits Faster R-CNN (for classification and detection) and performs a pixel-wise segmentation by adding a mask proposal generation with the ROIAlign
- The backbone of the Mask R-CNN is ResNet50-FPN (Feature Pyramid Network)

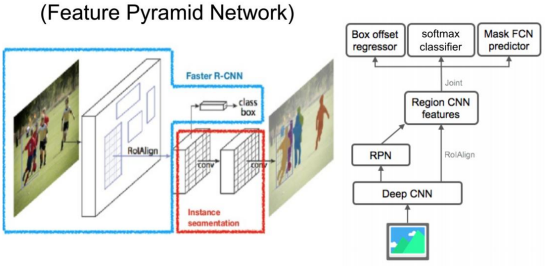

Figure 1. Mask R-CNN Overview [1]

Results

The following results are obtained by training the head layers

Baseline: results with pre-trained coco weights

Model 1: ResNet50 8500 steps

Model 2: ResNet101 3000 steps

Table 2. Training Loss, Validation Loss, and Kaggle Submission Score

	Training Loss	Dev Loss	Submission Score
Baseline	—	—	0.02538
Model 1	1.4182	1.4644	0.03607
Model 2 (LR=0.001)	1.5459	1.4794	0.03098
Model 2 (LR=0.005)	1.9589	2.1058	0.02763

Discussion

- Compare losses of different models
- Detection of images are successful
- Some images do not detect instances of target classes, particularly if far away (example in Fig. 2)

Figure 2. Missing "bicycle" inference

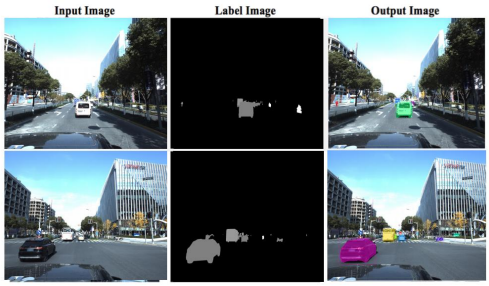

Figure 3. ResNet50 results on test set (top), ResNet101 results (bottom)

Future Work

- Train sequential images using LSTM layers alongside the existing Mask RCNN model to improve the prediction performance
- Train with more images
- Use the average precision (AP) metric for comparison with other works