Vehicle Detection and Segmentation With Mask-R-CNN

Zhe Zhou, Jiaxuan Sun, Jen Wang
Course: ANLY-590 Neural Networks
Instructor: Dr. Keegan Hines
10th Dec. 2019

***Abstract*—The objective of this project is to identify and segment vehicles in an image. Mask-R-CNN is used as the primary method for training the detector and running inference. The dataset utilized is a self-created synthetic data set, including 15000 training images and 3000 validation images. The results are very promising with the network being able to attain a detection precision of 0.9. What is more surprising is that our model is able to detect, segment and even assign correct primary label on those vehicles it has never seen.**

I. Introduction

Recent advances in deep learning and computation infrastructure (cloud, GPUs etc.) have made computer vision applications leap forward very fast. Convolutional neural networks (CNN), the driver behind computer vision, are fast evolving with advanced and innovative architectures to facilitate computer vision applications especially those related to pattern recognition. Since 2018, Mask-R-CNN has been the latest state of art in terms of instance recognition and segmentation. In this project, we propose to use Mask-R-CNN network architecture to conduct vehicle recognition and segmentation. We use Matterports (MIT) implementation of Mask-R-CNN with different sets of weights from transfer learning to train on our own synthetic dataset, then use the trained weights to run inference on new images.

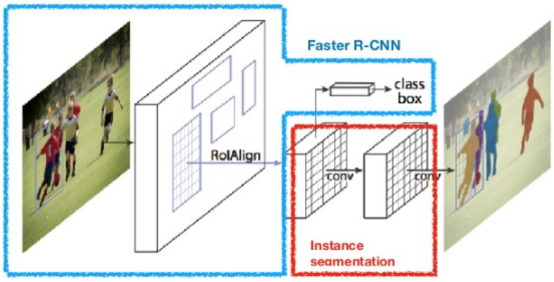


Fig. 1: Mask R-CNN is Faster R-CNN model with image segmentation

II. Related Work

Object detection and instance segmentation is an active area of research in computer vision applications. These advances have been largely driven by architectures like Fully Convolution Network, Fast/Faster R-CNN. R-CNN used selective search to select proposal in a image. Each proposal is sent through the deep learning model and a 2048 vector is extracted. Independent classifiers are trained for each class on these vectors to classify the objects[1]. Fast R-CNN removed training on SVM classifiers and used a regression layer and classification layer[2]. They also have applied selective search on feature map instead of image, thus eliminating the need of sending each proposal through the entire network. Faster R-CNN removed selective search and used a deep convolution network called RPN (region proposal network) to generate proposals thus allowing to train a end to end neural network in a single stage[3]. Mask R-CNN is an extension of Faster R-CNN with an additional module to generate high quality segmentation masks for each image[4].

Fig. 2: Pattern Samples (SUV-02 showcase)

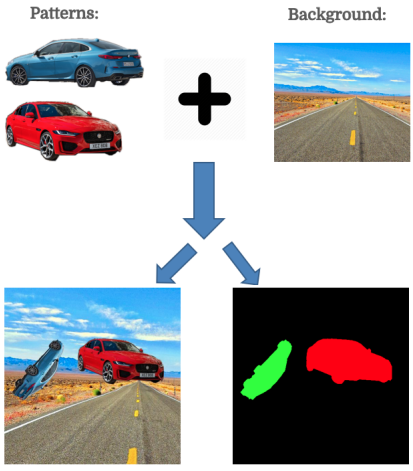


Fig. 3: Synthetic Image Generation Steps

III. Dataset

Patterns and background images have to be prepared first manually. The idea is to crop certain number of vehicles from online images with different view angles, then our dataset-generation program would randomly take certain number of patterns and one background image to generate sufficient number of COCO-like image sets. Here COCO-like image set means an image with its corresponding pattern mask. Then another program would take the mask to generate pattern coordinate info and store in a JSON file. Due to the short of time we only cropped 20 patterns (from different angles) for each of the 4 vehicles(BMW-235, Jag-XE, Volvo-XC60, Land Rover Discovery). Each vehicle has a primary label and secondary label. For example, the label for the Jag-XE would be sedan-01, where the 'sedan' is the primary label, and '01' is the secondary label. Figure 2 and Figure 3 illustrate this idea for better understanding.

IV. Methodology

A. Mask R-CNN Structure

Mask R-CNN (regional convolutional neural network) is a two-stage framework (Figure 4):

- **Stage 1** identifies Regions of Interest (ROI), in two parts. The image is fed into an FPN + ResNet50 backbone, which outputs a feature map. A Region Proposal Network (RPN) then scans over this feature map, convolutionally evaluating multiple anchors simultaneously and identifying the ROI with the anchor and a simple foreground or background evaluation.
- **Stage 2** analyzes the regions considered foreground, generating masks for objects, and in parallel classifying objects to which the masks can be applied.

FPN improves the standard feature extraction pyramid by adding a second pyramid that takes the high-level features from the first pyramid and passes them down to lower layers. By doing so, it allows features at every level to have access to both, lower and higher level features.

B. Mask R-CNN Loss Function

The multi-task loss function of Mask R-CNN Combines the loss of classification, localization and segmentation mask:

$$\mathcal{L} = \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{box}} + \mathcal{L}_{\text{mask}} \tag{1}$$

where $\mathcal{L}_{\text{cls}}$is:

$$\mathcal{L}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*) \log(1 - p_i) \tag{2}$$

$\mathcal{L}_{\text{box}}$is:

$$\mathcal{L}_{\text{box}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} L_1^{\text{smooth}}(t_i^u - v_i) \tag{3}$$

$\mathcal{L}_{\text{mask}}$is defined as the average binary cross-entropy loss, only including k-th mask if the region is associated with the ground truth class k:

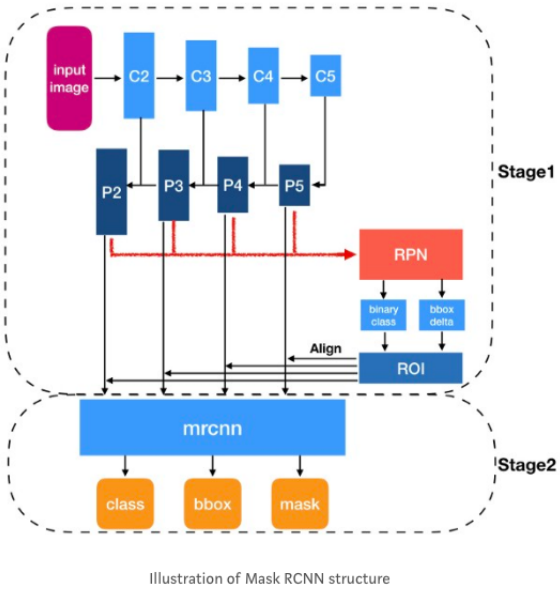


Illustration of Mask RCNN structure

Fig. 4: Two-Stage Mask R-CNN Structure

$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} \left[y_{ij} \log \hat{y}_{ij}^k + (1 - y_{ij}) \log(1 - \hat{y}_{ij}^k) \right] \tag{4}$$

C. Network Training

The network was initialize with weights pre-trained on the MS COCO dataset. Since our images have size of 800 x 700 which is very to the ones in MS COCO dataset, we do not have to retrain all the weights. We only train on the head layer and freeze the rest. After the head layer is trained, then fine tune the rest layers. Initial training of the training dataset showed that a major contribution to the loss function arose from the Mask R-CNN bounding box loss indicating that the network structure head layers required more training than the backbones.

There are quite a few hyper parameter that could be further tuned prior to training to improve the results:

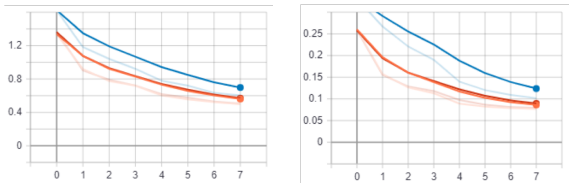
- Learning Rate
- Gradient Clip Norm
- Learning Momentum
- Weight Decay
- Backbone Network
- Backbone Strides
- Scales, Rations, and Anchors per image for the RPN

V. Results

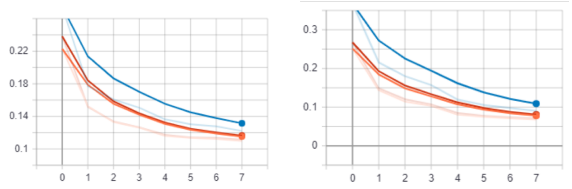
A. Network Training Results

Following Figures illustrate the loss functions, some weights distribution and sample feature visualizations:

According to figure 5, our loss function converges quite well. Total loss is around 0.45, class loss is about 0.12, mask loss is about 0.09 and the box loss is around 0.075. Figure

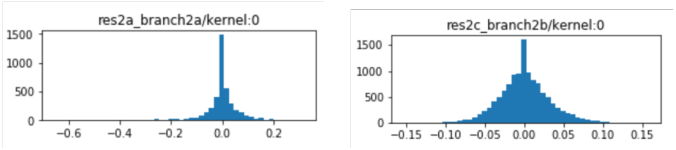


(a) Overall Loss (b) Mask Loss



(c) Class Loss (d) Box Loss

Fig. 5: Loss Function Visualization



(a) Res2a (b) Res2c

Fig. 6: Weights Distribution Example

6 shows weights distribution that were taken out from Res2a layer and Res2c for example illustration, and the weights are indeed normally distributed. Figure 7 just shows the feature map from the first convolutional layer on a sample image.

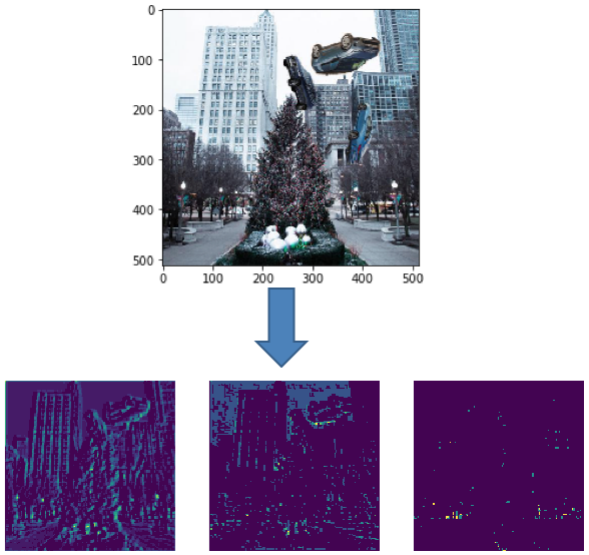


Fig. 7: Feature Visualization (Res2a Layer example)

B. Pattern Inference Results

Figure 8 shows the pattern inference on the synthetic training image, all the patterns was clearly identified and segmented, and correctly labeled (both primary and secondary). Figure 9 shows the pattern inference on the real image with trained vehicle but not that specific angle on that image. That vehicle was clearly identified and segmented, and correctly labeled (both primary and secondary), which is very promising. Figure 10 shows the pattern inference on the real image with four vehicles that were never trained in the model. As we can see, all the four vehicles were successfully segmented and even correctly assigned the primary label, which is 'SUV'.

Fig. 8: Inference on synthetic Image (patterns trained)

Fig. 9: Inference on actual Image (pattern trained)

Fig. 10: Inference on actual Image (patterns not trained)

VI. Conclusion

For trained patterns, our model can reach 93.2 percent vehicle identification rate and 96 percent vehicle classification rate, which is pretty good. Given the fact we only have the time to prepare 4 specific models of cars, this high classification rate

TABLE I: Identification Rate (VGG50)

Pre-trained Weights	Accuracy
MS COCO	93.2
ImageNet	89

TABLE II: Identification Rate (VGG100)

Pre-trained Weights	Accuracy
MS COCO	92
ImageNet	87.6

is not that representative. However, once we have more kinds and models of cars prepared, classification will play a more important role. What is more promising is that our model was able to identify and segment vehicles which have never been trained in the model, as shown on the left.

VII. Future Work

The fundamental challenge here for instance segmentation is the lack of data. With 20000 training images or more which derived from 4 vehicles, it is very easy to cause overfiting. The only way to tackle this issue is to prepare more different training patterns.

Other CNN network like VGG16 and VGG19 will be used for backbone to see how would the result change from the ResNet50 and ResNet100 backbone that we have uesed. Also, different combination of hyper-parameter will be further explored.

VIII. References

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proc. IEEE Conf. on computer vision and pattern recognition (CVPR), pp. 580-587. 2014.

[2] Ross Girshick. Fast R-CNN. In Proc. IEEE Intl. Conf. on computer vision, pp. 1440-1448. 2015.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (NIPS), pp. 91-99. 2015.

[4] Kaiming He, Georgia Gkioxari, Piotr Dollr, and Ross Girshick. Mask R-CNN. arXiv preprint arXiv:1703.06870, 2017.

[5] A Brief History of CNNs in Image Segmentation: From R-CNN to Mask R-CNN by Athelas.