V for Vehicular Video Vision

John Chuter
Department of Computer Science
Stanford University
jchuter@stanford.edu

Abstract

Autonomous driving needs realtime analysis of a driving environment. One approach is image analysis, the AI for which has moved swiftly. These algorithms perform semantic image segmentation of a driving environment, on objects relevant to a driver (e.g. car, pedestrian). This project tests one of the most recent developments, Mask R-CNN (2017), fine-tuned on a dataset for the 2018 CVPR WAD Video Segmentation Challenge. The result was decreased performance across the board, due to a failed implementation by the author. As a result, this paper serves to explain the dataset and algorithm, but little more.

1 Introduction

Computation time has been a major historic challenge of image analysis. Deep learning has progressed in multiple directions (such as YOLO and region proposal) to efficiently achieve high precision and recall. From sliding window, to Fast and Faster RCNN, to YOLO, and to Mask R-CNN, we move to faster and more precise methods. With further improvements, we approach human performance at autonomous driving and other related milestones. Specifically, this project implements the Mask R-CNN algorithm with pre-training on COCO, and tuning on WAD's newest dataset. The input is an image of a driving environment, and the output is that same image, with masks covering the pixels of classified objects.

2 Related Work

While no paper has been released on this dataset, it would be remiss to not mention the community at Kaggle surrounding this competition, with particularly useful discussion around dataset preprocessing strategies, such as selecting a subset of the data with a high number of classified objects and kernels to execute common tasks, such as notebooks for dataset visualization. [1] https://www.kaggle.com/c/cvpr-2018-autonomous-driving

3 Dataset and Features

3.1 Overview

This dataset [1] was comprised of image pairs, where each input image is from footage of videoed driving scenes, and the associated label is a corresponding image where each pixel value indicates the instance and class of an object. These images are 720p, colored and labeled with 30+ classes. Per the competition, there are 92GB preallocated for a training set, 4GB for a dev set, and private test set.

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

There are three other datasets to compare this to. COCO, the dataset used on which my implementation was pretrained, is another segmented image dataset with over 200k images labeled with generic object used by the original Mask R-CNN paper [2]. CityScapes is another dataset also used by that paper, with 5000 examples of city environments. Finally, previous students of CS230 used the Mapillary dataset for its varied traffic situations [3]. However, the sheer volume of the WAD dataset is roughly 10x these other datasets, and is comprised of ordered video sequences, which introduces potential for unique model architectures.

Figure 1: example sequence of images.

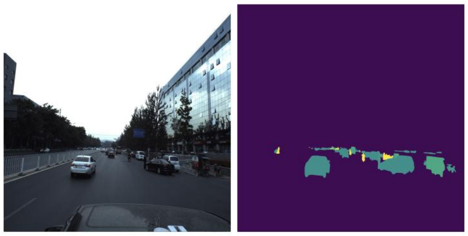

Figure 2: left image as input, right as label

3.2 Preprocessing

There are multiple strategies for preprocessing. Specifically, to improve computational efficiency:

1. Select images with more instances, and prioritize training on these.
2. Trim the upper half of the image, as this typically contains a lower density of relevant objects, and a higher density of sky and/or irrelevant objects, when compared to the lower half.
3. Resize to 1024x1024. Per the Matterport implementation [4], this allows training on multiple images simultaneously.

While the COCO dataset has 80+ categories, and the CVPR 30+, I defined 6 categories of interest: car, motorcycle, bicycle, pedestrian/person, truck, and bus, with the additional background category. These categories correspond to the Kaggle competition's evaluated categories.

3.3 Implementation Configuration

To train Mask R-CNN requires an input image, output class labels, and masks. For this training set, the output image's pixel values capture the instance and class, with the bounding boxes extracted from the extreme coordinate of each object mask. Specifically for the Matterport implementation, the three methods of ADD_CLASS, ADD_IMAGE, and LOAD_MASK take the Kaggle dataset and make it usable by their pretrained model. In LOAD_MASK, incidentally, I notice I fail to correctly assign height and width, and is one potential source of error.

4 Methods

4.1 Mask R-CNN

Mask R-CNN works in two stages, with a total of four parts. Stage 1 identifies regions of interest, in two parts. The image is fed into an FPN + ResNet "backbone", which in image-to-vec fashion outputs a feature map. A Region Proposal Network (RPN) then scans over this feature map, convolutionally evaluating multiple anchors simultaneously and identifying the Regions of Interest, 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. First we take the regions identified by the RPN, pool those with ROI pooling to a consistent size, then classify those regions more deeply into specific categories, with refinements to the corresponding bounding boxes. The masks we generated for objects are then applied to the classified regions.

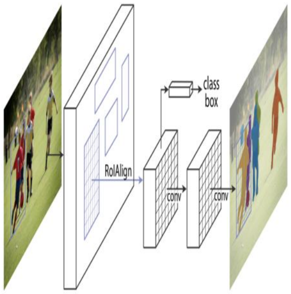

Figure 3: Mask R-CNN [2]

4.2 Matterport Implementation

Specifically, this model uses:

1. ResNet101 + FPN for CNN “backbone”
2. Cost function: $L = L_{class} + L_{box} + L_{mask}$
3. Pre-trained weights from the MS COCO dataset
4. Fine-tuned head layers with WAD
5. Images of dimensions 1024x1024

5 Experiments/Results/Discussion

5.1 Hyperparameter Tuning (in the Abstract)

I had 3 different configurations. One was the default, the other two adjusted the learning rate and mask threshold values. Specifically,

1. learning rate = .001, .005 .01
2. Mask Threshold = 32x32, 64x62, 128x128

Learning rate seemed to be a common hyperparameter to play with, being a parameter that varied dataset to dataset for the original paper, the Matterport default, and other Mask R-CNN projects, and was responsible for drastically different results by those authors. The idea of a Mask Threshold was taken from the previous quarter’s usage of Matterport [3], as it addressed the common issue of very small objects due to the image resizing.

5.2 Evaluation Metric

The evaluation metric is IoU. Loss is calculated from classification, bounding box, and the mask.

5.3 Discussion

Ultimately, this model failed to achieve results. The loss function increased continuously and dramatically. While there are numerous possible reasons, I believe the root cause is a bug in my Dataset subclassing.

6 Conclusion/Future Work

In summary, the WAD dataset is an exciting new resource for image segmentation that merits future exploration. For me, the future would consist of executing the above ideas. To do this, I have a couple reflections. While the Matterport implementation has been used by others to great efficacy, in the future I intend to experiment with the other models due to ease of use. For training such a large dataset, compute power and time are also valuable resources I would take advantage of. For this, while it ultimately didn't matter, I found that the Colaboratory notebooks made cloning a repo, adding Google Authorizations, batching data from Google Drive, and installing dependencies relatively simple to alternatives. I also would collaborate with others, to reduce error and individual workload. Specifically, in the context of this class, I should have followed up with my TA when I didn't receive responses to email, not to mention the multiple other resources available to me. Even though I thought it was fine, I should have gone to OH and checked anyway. Tuesday morning when I started to train, I figured I was basically done. Tuesday night I was despairing as the performance went in the wrong direction. A summer project to enjoy, at least, for the completionist in me.

References

[1] CVPR 2018 WAD Video Segmentation Challenge. https://www.kaggle.com/c/cvpr-2018-autonomous-driving

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

[3] Lu, Xiao, et al. "Mask R-CNN application: Instance Segmentation in Driving Scenes." 2018.

[4] Matterport. "Mask R-CNN for Object Detection and Segmentation". https://github.com/matterport/Mask_RCNN