Machine Learning Engineer Nanodegree

Capstone Proposal

Grégory D'Angelo
April 5, 2018

CVPR 2018 WAD Video Segmentation Challenge

Domain Background

Building self-driving cars and autonomous vehicles are one of the most challenging AI project of our days. Several companies, from car manufacturers to tech giants through a plethora of startups, are all competing in race to build the future of mobility. A crucial part of autonomous driving is perception to acquire an accurate understanding of the environment in which the car is operating. Indeed, the recent tragic accident involving a self-driving car reaffirms how important it is to sense the world around the car.

Hence, as an aspiring self-driving car engineer, I'm going to tackle this problem as part of my Machine Learning Nanodegree capstone project. With the knowledge gained from this nanodegree and the Self-Driving Cars Nanodegree, I will participate to the *CVPR 2018 WAD Video Segmentation Challenge* on Kaggle. This challenge is a unique opportunity to work on a tremendously high value and high profile problem for autonomous driving.

Problem Statement

Sensing the surrounding environment is very important for an autonomous vehicle to operate safely in it. A fully attentive human-being can easily and reflexively differentiate between objects/instances, such as person vs. a stop sign, when driving. However, for autonomous vehicles, this task is not as trivial as it is for human drivers. Hence, this Kaggle challenge aims to solve this environmental perception problem for autonomous driving.

In this challenge, I'll have a set of video sequences with fine per-pixel labeling, in particular instances of moving/movable objects such as vehicles and pedestrians are also labeled. The goal is to evaluate the state of the art in video-based object segmentation, a task that has not been evaluated previously due to the lack of fine labeling. The average moving/movable instances per frame can be over 50, in comparisons, only up to 15 cars/pedestrians are labelled in the KITTI dataset. Some very challenging environments, such as harsh traffic and lighting conditions, have been captured as shown in the following images (center-cropped for visualization purpose).

Figure 1: Challenging lighting condition

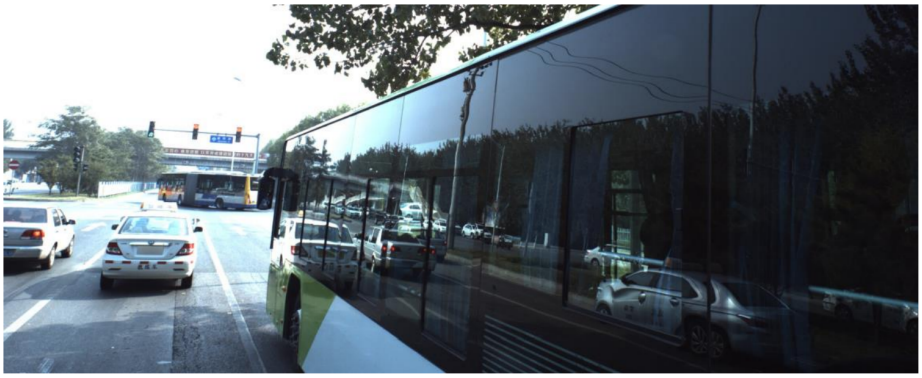

Figure 2: Reflection on bus windows

Figure 3: Harsh Traffic

Datasets and Inputs

To solve the scene parsing problem, I'll use the ApolloScape dataset provided by Baidu, Inc. It contains survey grade dense 3D points and registered multi-view RGB images at video rate, and every pixel and every 3D point are semantically labelled. In addition precise pose for each image is provided. The subset used for this challenge has around 60K image frames and corresponding instance-level annotations. For details about the class definitions and dataset structure please refer to ApolloScape website.

The authors equipped a mid-size SUV with high resolution cameras and a Riegl acquisition system. The dataset is collected in different cities under various traffic conditions. The number of moving objects, such as vehicles and pedestrians, averages from tens to over one hundred. Image frames in the dataset are collected every one meter by the acquisition system with resolution 3384 x 2710.

The dataset is divided into three subsets for original training images, training images labels and test sets images respectively. See below an example of a labelled image of the training set (cropped for visualization purpose).

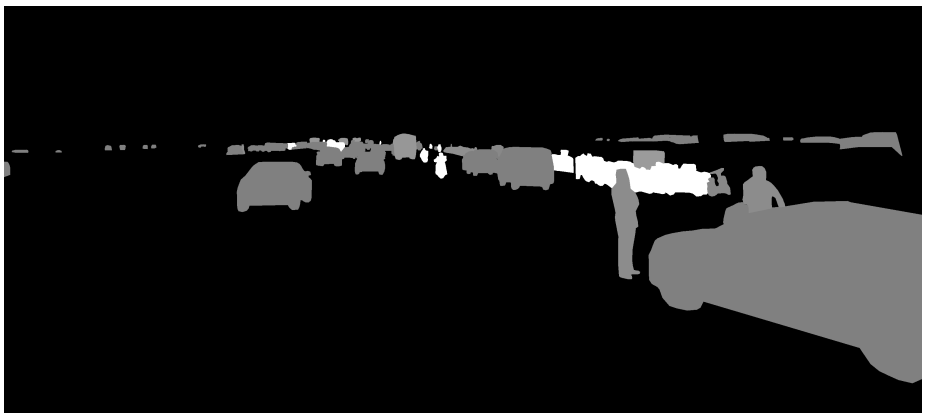

Figure 4: Dense traffic environment

Solution Statement

Recent developments in neural networks architectures and techniques have greatly advanced the performance of computer vision tasks. New applications have become possible thanks to these scientists progress, in particular in the field of self-driving cars. Hence, I'll focus on using end-to-end models, and taking advantage of transfer learning, for solving scene parsing. Specifically, I plan to review state-of-the-art architectures and techniques for semantic segmentation with deep learning.

Benchmark Model

I plan to compare the results of my final solution to the results obtained with a basic Convolutional Neural Network (CNN) that I will design with few layers. I will compare the performance to each other, and also visually compare the output segmentation on the training set. A well-designed solution using state-of-the-art architectures on segmentation task should be able to beat a basic CNN model.

Evaluation Metrics

The evaluation metrics used to quantify the performance of my solution model are provided by the authors of the Kaggle competition.

The interpolated average precision (AP) is used as the metric for object segmentation. The mean AP (mAP) is computed for all the video clips and all the classes at different intersection-over-union (IoU) thresholds. The IoU between a predicted instance A and a ground truth instance B is computed by $IoU(A, B) = \frac{A \bigcap B}{A \bigcup B}$.

To obtain the Precision-Recall curve, the authors choose ten IoU thresholds in range [0.5, 1.0) with step 0.05. They match ground truth instances with predicted instances at different IoU thresholds. For example, given an IoU threshold 0.5, a predicted instance is considered as "matched" if the IoU with a ground truth instance is greater than 0.5. If there are multiple predicted instances matched to a ground truth instance, the predicted instance with the largest IoU is considered as the true positive, and remaining predicted instances are false positives. The predicted instances that are not matched with any ground truth instances are counted as false positives. If IoU between a predicted instance and ignoring labels is larger than the IoU threshold, this predicted instance is removed from the evaluation. Notice that the group classes, such as car group and bicycle group, are also ignored in the evaluation.

Project Design

Before doing any modelling, I will dive into the dataset to get a basic summary and visualize it. Number of training examples vs. testing examples, images shape, class definitions and distributions, are key characteristics that I will explore. Also, displaying some original images and associated ground-truth label images will help me have a good understanding of the dataset.

After playing with the data, I will start by implementing one or more state-of-the-art network architectures such as the well-known U-Net, SegNet/Squeeze-SegNet, PANet, and LinkNet. I will also have a look to the Carvana Image Masking Kaggle Challenge winners approach and architecture, called TernausNet. They have win a Kaggle challenge on similar computer vision task. This could be very

helpul at solving the semantic segmentation problem I'm facing. In terms of software, I'll use Keras or Tensorflow as a deep learning framework and OpenCV for image processing.

At this point, I'll train some of these architectures using pre-trained weights to speed up the training phase and improve performance of semantic segmentation. The goal of training multiple models is to benchmark them and eventually average/combine their results. Then, I will move on to the next step which is testing the model on completely new data it has never seen before. For this competition, it means submitting my solution on the evaluation server on Kaggle.

Moreover, like almost all computer vision task solved with deep learning I'll use data augmentation such as horizontal flips, shifts, rotations, and color transformations. I plan to use the imgaug python library to do so.

At every iteration of my project workflow (e.g. after every submission), I'll finetune my model in order to enhance its performance. One important thing I plan to do is to understand the limitations of my model by performing a visual inspection of the predictions. For the train set, I'll review cases with the lowest performance scores and try to tune the model accordingly.