Semantic Segmentation for Lyft Perception Challange

Shreyansh Bhatt

1 Introduction

This short report describes Fully Convolutional Neural Network (FCN) based Semantic Segmentation approach for identifying road and Cars in an image.

Given an image (as shown in Figure 1) left, the task is to identify Drivable portions and other cars in a given image within 1/10 of a second. The idea is to classify each pixel in one of the following three classes, 1. None, 2. Road, and 3. Car.

The evaluation measures are Precision, recall, and image processing time are factors evaluating the solution.

FCN is found to perform well on pixel-to-pixel semantic segmentation task. The basic idea is to preserve global information of where an object is along with the identification of objects. FCNs achieved state-of-the-art accuracy in semantic segmentation task. Hence, I have used FCN for this challenge.

In the rest of the report, I have described the FCN architecture, the way I achieved reported F score, and the way I achieved reported FPS.

2 FCN architecture

I have not changed the existing FCN architecture. Following describes the building blocks of the network. I have used the pre-trained VGG convolutional network and modified it to create FCN as suggested in the original FCN research paper. Specifically,

1. Remove fully connected layers after layer 7 and add a 1x1 convolutional layer.
2. Add a transpose convolutional layer, upsampling layer.
3. Add a skip layer from pooling layer 4 of VGG net to the resulting layer of step 2.
4. Upsample resulting layer of step 3.
5. Add a skip layer from pooling layer 3 of VGG net to resulting layer of step 4.
6. Upsample the resulting layer of step 5.

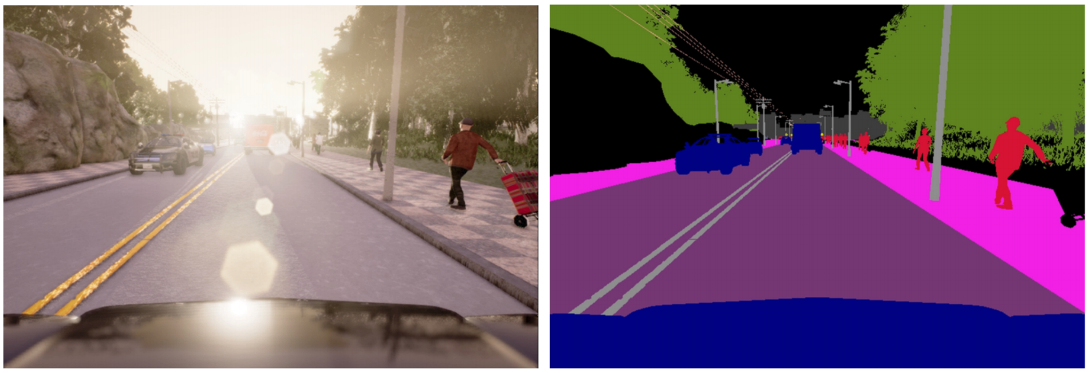

Figure 1: Example image

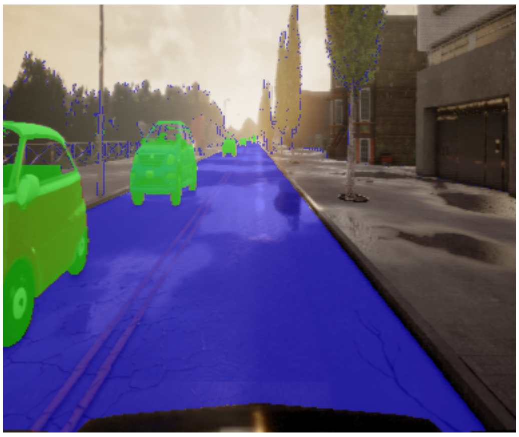

Figure 2: Resize image before removing extra information

The result of layer 6 is of the size of original input (For details on upsampling, please refer to the code). In other words, the result is a probability of 3 classes for each pixel of an image. Softmax is applied to this to get the final class probability.

The idea behind upsampling is to label all pixels within a boundary to the label of a particular pixel. E.g., A portion of the image was identified as having a road (class 1) by CNN. Upsampling is marking the whole region with it. Also, the skip layers are adding global information that is otherwise thrown away by CNN. 1x1 convolution is also applied on the skip layers before adding them in step 3 and 5.

Another key information is scaling these skip layers before a convolution is applied, i.e., weights of these layers are scaled before they are added to the resulting layer as described. This allows the network to achieve better accuracy as well as to train the network fast. I achieved the F score reported in here within ten epochs of training.

3 F-score

I found several choices affecting the f score. Following, I have described the rationale of each choice and its result.

3.1 Image Scaling

The input image is from Carla simulation and has a size of 600x800 resulting in 480,000 pixels. Such an image is huge for Training as well as the inference part. Hence, I resized the images. I tried several image resize options and the one that worked the best was 320x384. Also, the images are resized after removing the hood and sky from the image. I believe the reason why 320x384 working the best (other options were 320x576 and 320x768) was because CNN is found to work the better with squared images (supported by several blog posts).

Another essential information is resizing images before removing other information from the labeled data set, i.e., setting road as class 1, car as class 2, and rest of the objects as 0 and after removing this information. I found that images should be resized after removing this information. Following shows the implication of both.

3.2 Hyper parameter selection

I found the network for this task to be sensitive to the initial weight selection and learning rate selection. The learning rate yielding the best results was 1e-4. The initial weights were selected randomly with mean 0 and standard deviation of 0.001. I did not find the network to be susceptible to the regularization loss parameter.

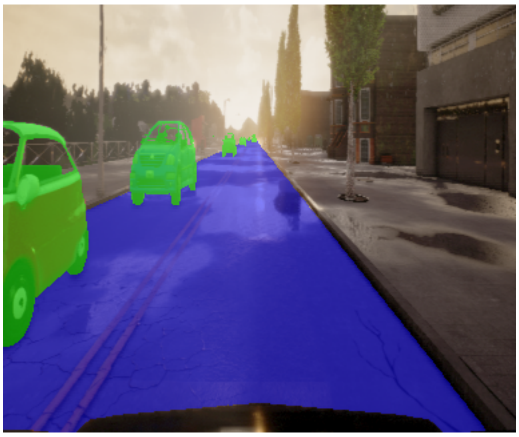

Figure 3: Resize image after removing extra information

3.3 Training data selection

Another critical factor affecting the F-score was the training data. Initial training data (given with the project) had relatively fewer cars. Hence, the network had difficulty identifying cars yet yielding a good F-score for the road. I enhanced the training data with different weather condition and images with cars in the training data. This improved car recall but affected the precision. I then added images with pedestrian so as the network learns on distinguishing pedestrians from cars. I found this improving the precision but ended up compromising the recall. As recall for the car was important, I did not add lots of images with pedestrians. However, I believe the final training data should have more images with pedestrian and then let the network learn for 20+ epochs (which I did not do for this task. I should have).

4 Frames Per Second

Another important aspect of the project was to have an adequate FPS, i.e., the number of images the trained network can process and label in one second. I found that batching 10 images of size 320x384 achieved 10 FPS which as an intended FPS for this project. I believe this is still a realistic condition since a camera captures video of 60 FPS; hence process 10 Frames together can still let a car decide the environment in real time.

Also, compressing the model was a crucial part in achieving 10 FPS. I used freeze model utility of Tensorflow for model compression.

The final F-score achieved was 84.3.

5 Misc

Another important detail I would like to report is regarding binary classification vs. multi-label classification in this task. I started with two models each learning how to identify road and car respectively. I used FCN for both models and used these resulting models to identify road and car feeding single image to two models. I was hoping for this achieve better F-score than creating one FCN model which identifies road and car at the same time. However, it was not true. A single model trained for identifying three classes yielded relatively similar f scores compared to two models doing binary classification. This is very interesting and speaks for the feasibility of FCN in semantic segmentation since binary classification has mostly yielded better performance for ML tasks.

6 Conclusions

To conclude, FCN worked well for identifying roads and car in an image. Learning rate and initial weights are the key tuning parameters in the training process. FCN works satisfactorily for multi-label pixel classification. Training for more epochs would certainly yield a better f score, but training for only ten epochs itself was enough to achieve 80+ f score.