Semantic Segmentation

Patrick Cleeve

Objective:

The objective of the project is to train a fully connected network to detect road areas in images using semantic segmentation.

The code for the project is available here:
https://github.com/patrickcleeve/CarND-Semantic-Segmentation

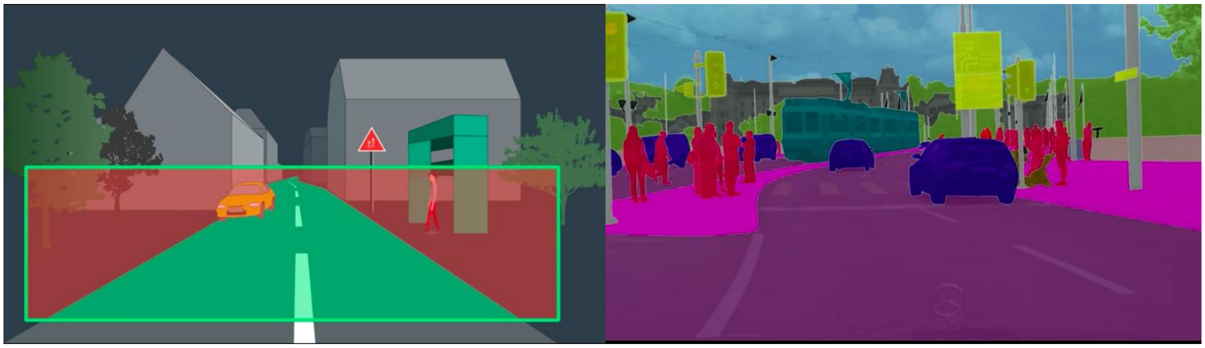
Fully Convolutional Networks

Standard Convolutional Neural Networks use a series of convolutional layers, connecting to fully connected layers and into an activation function. This pyramid structure is a good solution for classifying an image (e.g. hotdog/not hotdog). However, they are unable to locate where in an image an object might be, because the fully connected layers don't retain spatial information. They reduce the image down to a classification. In addition, the size of the input image is fixed (or has to be resized) due to the fixed size of the fully connected layers.

Fully convolutional networks replace fully connected layers with convolutions. This preserves the spatial information (how pixels are related to another), and outputs an image. We can use these networks to try to detect the location of objects in an image, and attempt to understand what is taking place in a image.

Scene Understanding

A simple and powerful method for understanding what is in a scene is through the use bounding boxes. We can train a network to recognise an object and draw a rectangular box around it. This method works well for detecting objects such as cars, people or traffic signs. State of the art networks such as YOLO, perform well even at high frames per second (fps). However, bounding boxes are less useful when detecting more complex, curvy shapes or areas such as the road or sky. As shown below the bounding box is unable to accurately capture the true shape of the road.



Bounding Boxes and Semantic Segmentation (Udacity Lecture)

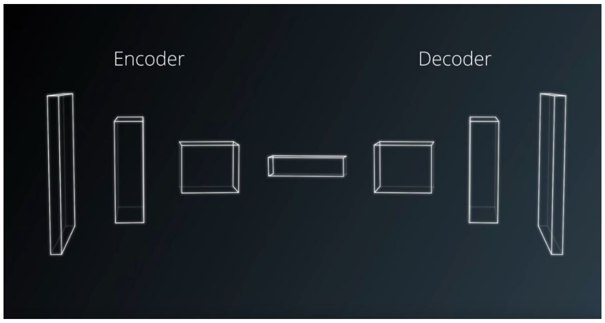
Semantic segmentation allows the network to classify each pixel of the image, therefore gathering valuable information from each pixel. However, the improved understanding comes at a significant performance costs, with segmentation inference limited to 4-7 fps compared to 40-70 fps for bounding box methods. These performance constraints are particularly important to autonomous vehicle applications as large cloud computing resource cannot be used due to latency (lag).

Semantic Segmentation

The semantic segmentation network is composed of two sections:

Encoder

The encoder is used to extract features from the image. Similar to transfer learning, we can use an already pre-trained image classification model to form the encoder section of the network. For this project we use VGG-16, but other models are common (e.g. ResNet).



Decoder:

Instead we use a Decoder, to upscale (sort of interpolate) the output of the encoder into an image

We use three different techniques in implementing the semantic segmentation model:

1x1 Convolution:
We can replace the fully connected layers with 1x1 convolutions to preserve the spatial information.

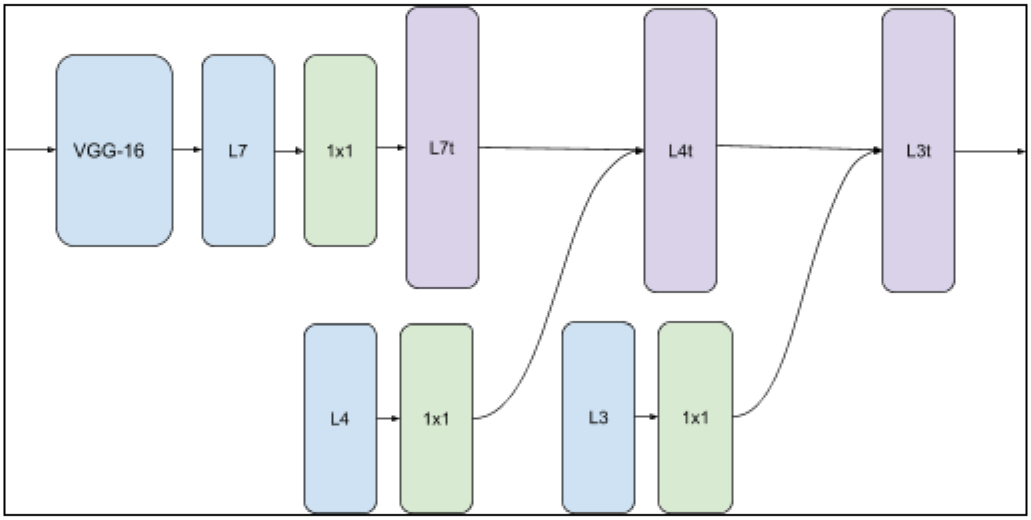
Skip Connections:
As convolutions are applied to the image, we narrow in on specific parts of the image, at the expense of losing the bigger picture. Even if decoding the ouput, we lose information in the process. We use skip connections to feed the output of an earlier layer to a later one, to preserve information from multiple different resolutions, allowing the network to make more precise segmentations.

Transposed Convolution (Deconvolution):
Convolutions reduce the spatial dimensions of the image as they are applied. The transposed convolution has the opposite effect, it up-samples the image to restore spatial information. We transpose data onto a sparse matrix and interpolate the areas between based on the information.

Project Architecture

For the project we implement the FCN-8 VGG16 from Fully Convolutional Networks for Semantic Segmentation. The model was trained on the Udacity Workspace using Tesla K80 GPU.

Blue: VGG-16 layer
Green: 1x1 convolution
Purple: Transposed Convolution (Upsample)



FCN8-VGG16 Architecture (Berkeley)

Final Model Parameters:

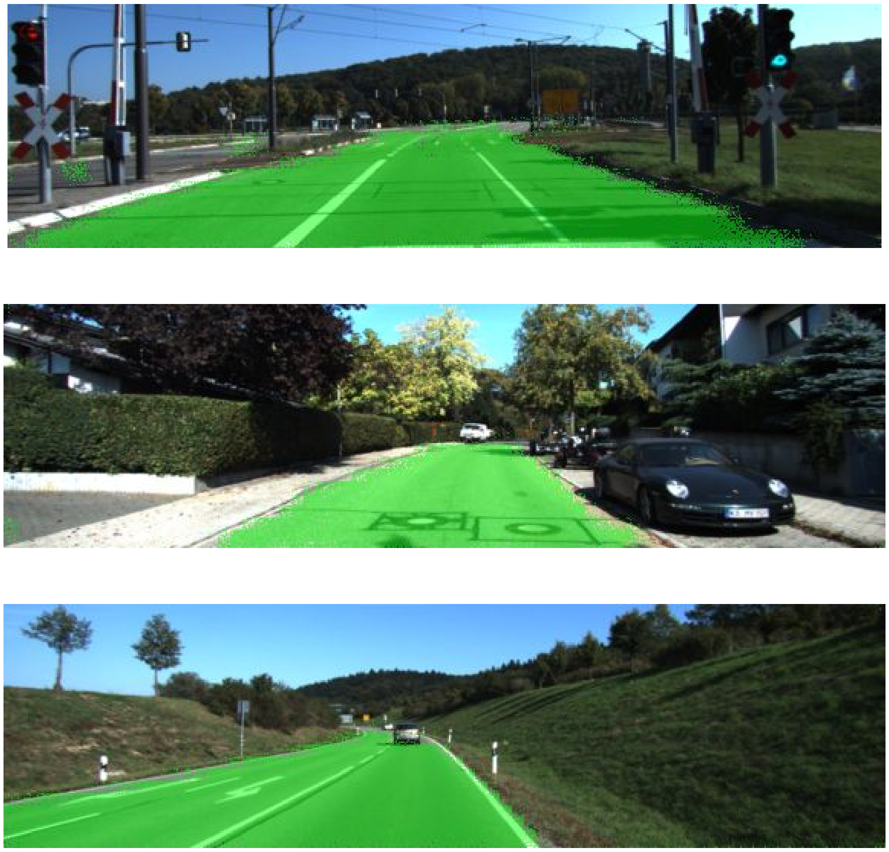
Epochs	25
Batch Size	5
Keep Probability	0.5
Learning Rate	0.0005

Discussion

Semantic segmentation is a powerful technique for classifying objects or areas with complicated shapes, such as roads. The model was very computationally intensive to train, and was the most complex neural net (VGG, FCN) I have had to implement. The training would not be possible (that is economical) without the powerful GPUs in the cloud. Even using a Tesla K80 (approx $1800 AUD), the training of 25 epochs took about 20-30 minutes.

Whilst the model performs reasonably well, it has a lot of room for improvement. In particular, it did not perform well in shaded areas, and does not show the road boundaries very cleanly. You are able to see individual pixels near the edge of the road, which is likely problematic for driving.

In addition, for use in actual vehicle inference the performance would need to be significantly improving using techniques such as model freezing, fusion and quantisation discussed in lecture. It might also be useful to train the model for multiple classes such as road, cars, pedestrian and signs, or with multiple decoders to do separate tasks such as depth measurement.



Semantic Segmentation Output

Reference

Traffic Sign Classification
https://github.com/patrickcleeve/CarND-Term1-TrafficSignClassifer-P2/blob/master/Traffic_Sign_Classifier.ipynb

Behavioural Cloning
https://github.com/patrickcleeve/CarND-Term1-BehavioralCloning-P3/blob/master/model.ipynb

Udacity Classroom, (Advanced Deep Learning)
https://classroom.udacity.com/nanodegrees/nd013/parts/6047fe34-d93c-4f50-8336-b70ef10cb4b2

Udacity, Project Walkthrough
https://www.youtube.com/watch?v=5g9sZIwGubk

Udacity, Project Slack Channel
S-t3-p-semantic-segmentation, no url available

One By One Convolutions:
https://iamaaditya.github.io/2016/03/one-by-one-convolution/

https://www.quora.com/What-is-a-1X1-convolution

https://stats.stackexchange.com/questions/194142/what-does-1x1-convolution-mean-in-a-neural-network

https://stackoverflow.com/questions/39366271/for-what-reason-convolution-1x1-is-used-in-deep-neural-networks

https://jhui.github.io/2017/03/16/CNN-Convolutional-neural-network/

https://medium.com/nanonets/how-to-do-image-segmentation-using-deep-learning-c673cc5862ef