**Project: Follow Me**

**Rubric Points**

**Here I will consider the rubric points individually and describe how I addressed each point in my implementation.**

**Write-up**

**1. Provide a write-up / README document including all rubric items addressed in a clear and concise manner. The document can be submitted either in either Markdown or a PDF format.**

In this write-up I describe the architecture of my neural network as well as the hyper parameters that I have used to meet the specifications. It also conveys my understanding of Fully Convolutional Neural Networks and Semantic Segmentation. I address the reason I chose the parameters and additional changes that could be made in the future to improve upon the project.

**2. The write-up conveys an understanding of the network architecture.**

I chose to employ a seven layer fully convolutional network to complete the semantic segmentation task that is required for the “Follow Me” Project.

To choose the parameters that are described below I first used my intuition and understanding of neural networks gleaned from the lessons to pick some starting values. I then followed a process of trial and error in the Semantic Segmentation Lab to verify and refine my choices. Once I had desirable results with that data I used the same parameters on my captured data and continued to refine the parameters until I achieved the desired results.

The first three layers are encoder layers which consist of two-dimensional separable convolution followed by a batch normalization process. We perform the normalization at the end of each layer so that the following layer has well-conditioned data to start out with. This allows the gradient decent to proceed more efficiently. Each encoder layer increases the depth of the model by increasing the number of filters.

* **Layer 1** – The input of Layer 1 is the current image from the batch. I used 32 filters with a stride of 2. The stride decreased the width and height of the model but changed the depth from 3 to 32.
* **Layer 2** – The normalized output of Layer 1 is used as the input to Layer 2. I used 64 filters and stayed with a stride of 2. This again decreased the width and height dimensions by a factor of two but also doubled the depth.
* **Layer 3** – This is my final encoder layer. The input is the normalized output of Layer 2. I again double the depth and cut the height and width in half.

The fourth layer is a 1x1 convolution. This layer has a four-dimensional output tensor which retains the special information needed for semantic segmentation.

* **Layer 4 –** The 1x1 convolution layer ha a kernel and stride of 1 (as per the definition of a 1x1 convolution) and I chose a depth of 512.

The final 3 layers make up my decoder.

**3. The write-up conveys the student's understanding of the parameters chosen for the neural network.**

I chose to employ a seven layer fully convolutional network to complete the task of

**4. The student has a clear understanding and is able to identify the use of various techniques and concepts in network layers indicated by the write-up.**

I chose to employ a seven layer fully convolutional network to complete the task of

**5. The student has a clear understanding of image manipulation in the context of the project indicated by the write-up.**

I chose to employ a seven layer fully convolutional network to complete the task of

**2. The student displays a solid understanding of the limitations to the neural network with the given data chosen for various follow-me scenarios which are conveyed in the write-up.**

I chose to employ a seven layer fully convolutional network to complete the task of

**Model**

**1. The model is submitted in the correct format.**

In this write-up I describe the architecture of my neural network as well as the hyper parameters that I have used to meet the specifications. I address the choices that I made and additional changes that could be made in the future to improve upon the project.

**2. The neural network must achieve a minimum level of accuracy for the network implemented (40% IoU).**

I chose to employ a seven layer fully convolutional network to complete the task of

**Future Enhancements**

**Conclusion**

In this project I tried to keep the code clean by breaking each step out into its own function. This was helpful when I decided to change the order of some of the filters (in particular I moved the voxel downsampling before the noise reduction to save computation time). The first two pick lists could benefit from capturing more samples of the objects in order to increase the success rate of object recognition. The object recognition could fail if the object of interest was obscured from view or was outside of the small window I selected for this project. If I were to continue with this project I would try to make the object recognition more robust, to identify the drop boxes as a drop box rather than obscuring them from my area of interest. I would also like to consider the “confidence” of the svm for each object to better ensure I do not pick up the wrong object.