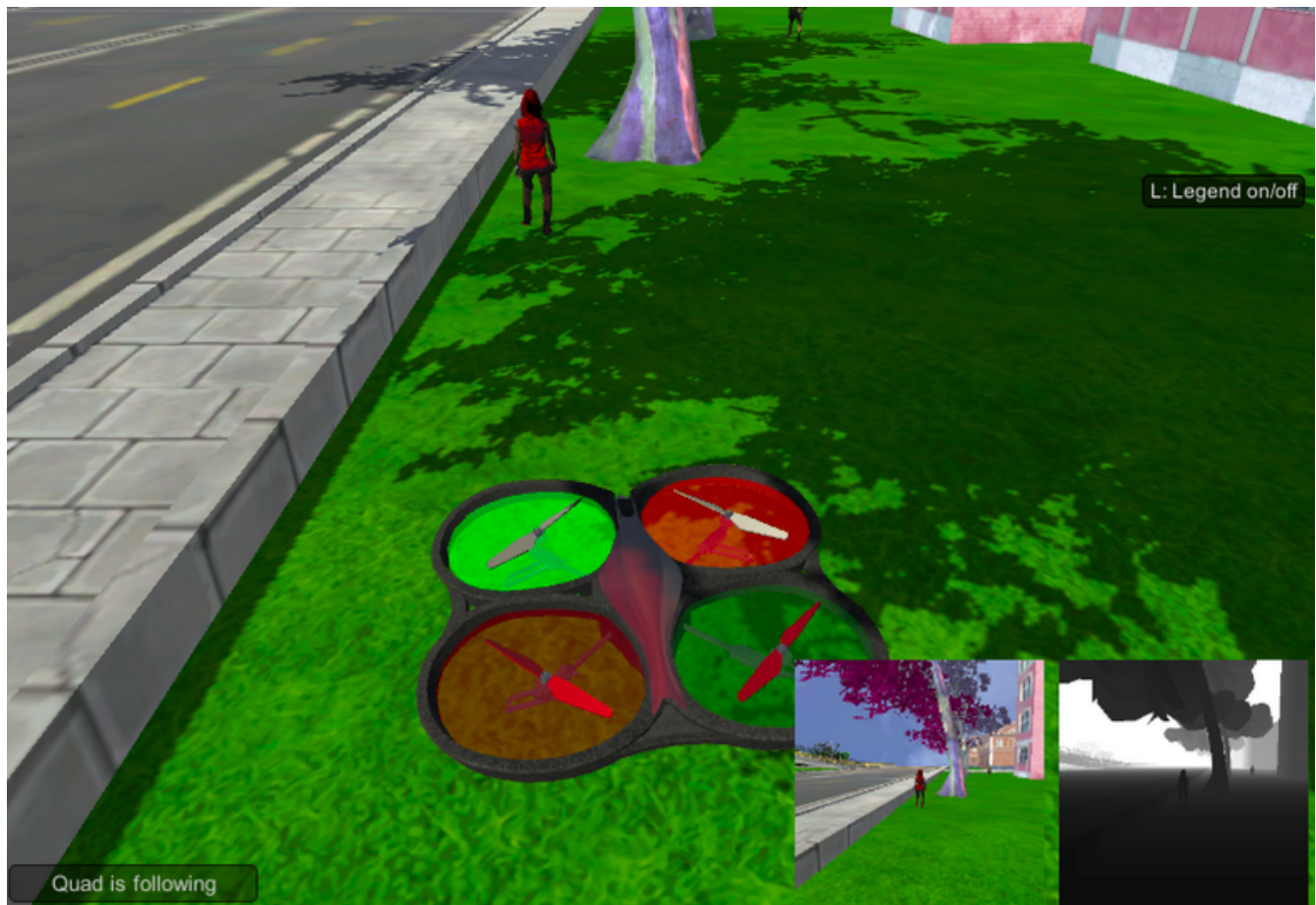


Arturo Polanco Lozano
Self Driving Car – Robotics Engineer
Report Follow Me Project

Introduction

This project uses a fully convolutional neural network to let a drone segment a target in an image and follow it in a virtual environment. This approach allows to paint every single pixel in an image depending on the label of the object it contains and in order to measure the performance of the model is used IOU (intersection over union) which takes the intersection of the prediction pixels and ground truth pixels and divides it by the union of them.



Data Collection

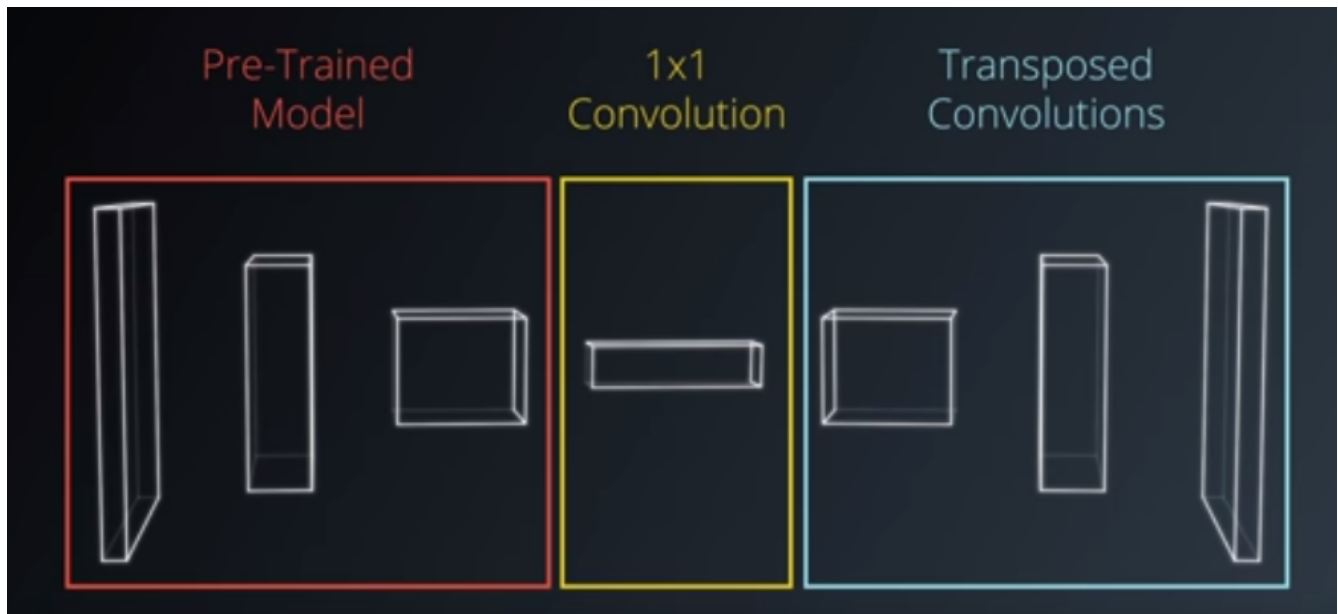
I used the data provided by Udacity.

Network Architecture

The network can be divided into the following stages:

- Input image with a skip connection with the output convolutional layer, skip connections are important due to they allow the network to use information from multiple resolution scales and helps to make more precise segmentation decisions.

- First encoder layer, which extracts characteristics useful to perform segmentation. This encoder has a skip connection with the second decoder.
- Second encoder layer which has a skip connection with the first decoder.
- 1x1 Convolution layer, this is an alternative of a fully connected with the advantage that preserves spatial information.
- First decoder layer, that up scale the encoder output back into the dimensions of the original image.
- Second decoder layer.
- Convolution output with softmax activation function to perform segmentation.



Hyperparameter Tuning

- Optimizer : Adam, is a very famous optimization algorithm and usually outperforms the other possible choices.
- Batch Size : 32, is a typical number.
- Number of epochs : 30, this value is selected by trial and error but since the computational cost is high I decided to give a look to similar exercises and found out 30 would work ok.
- Learning Rate: 0.001, is a recommended learning rate for Adam.
- Steps per epoch : 100, the number of batches of samples before declaring an epoch is finished.
- Validation steps per epoch: 50, since data should be tested in every single epoch this number represents the number of images in validation over batch size.

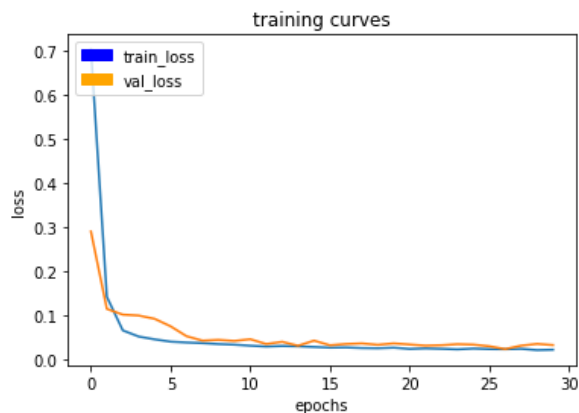
Training Process

Unfortunately I run out of AWS credits so I had to perform the training using my pc, after a very long period of time I could successfully train the neural network with 30 epochs in approximately 20 hours.

Experiments and Results

Since the computational resources are so high for fully convolutional neural networks I only did only one implementation, the metric showed a performance above 0.4 and that was very satisfying. The performance in simulation following the target was so also very good. In order to perform segmentation of ther objects like cats or dogs I would have to retrain the algorithm using additional data of those objects, depending of the application that would make the model more robust and complex.

```
100/100 [=====] - 2139s - loss: 0.0210 - val_loss: 0.0351  
Epoch 30/30  
99/100 [=====>.] - ETA: 18s - loss: 0.0219
```



```
100/100 [=====] - 2158s - loss: 0.0219 - val_loss: 0.0325
```

```
Out[9]: <tensorflow.contrib.keras.python.keras.callbacks.History at 0x7fb1277e2d30>
```