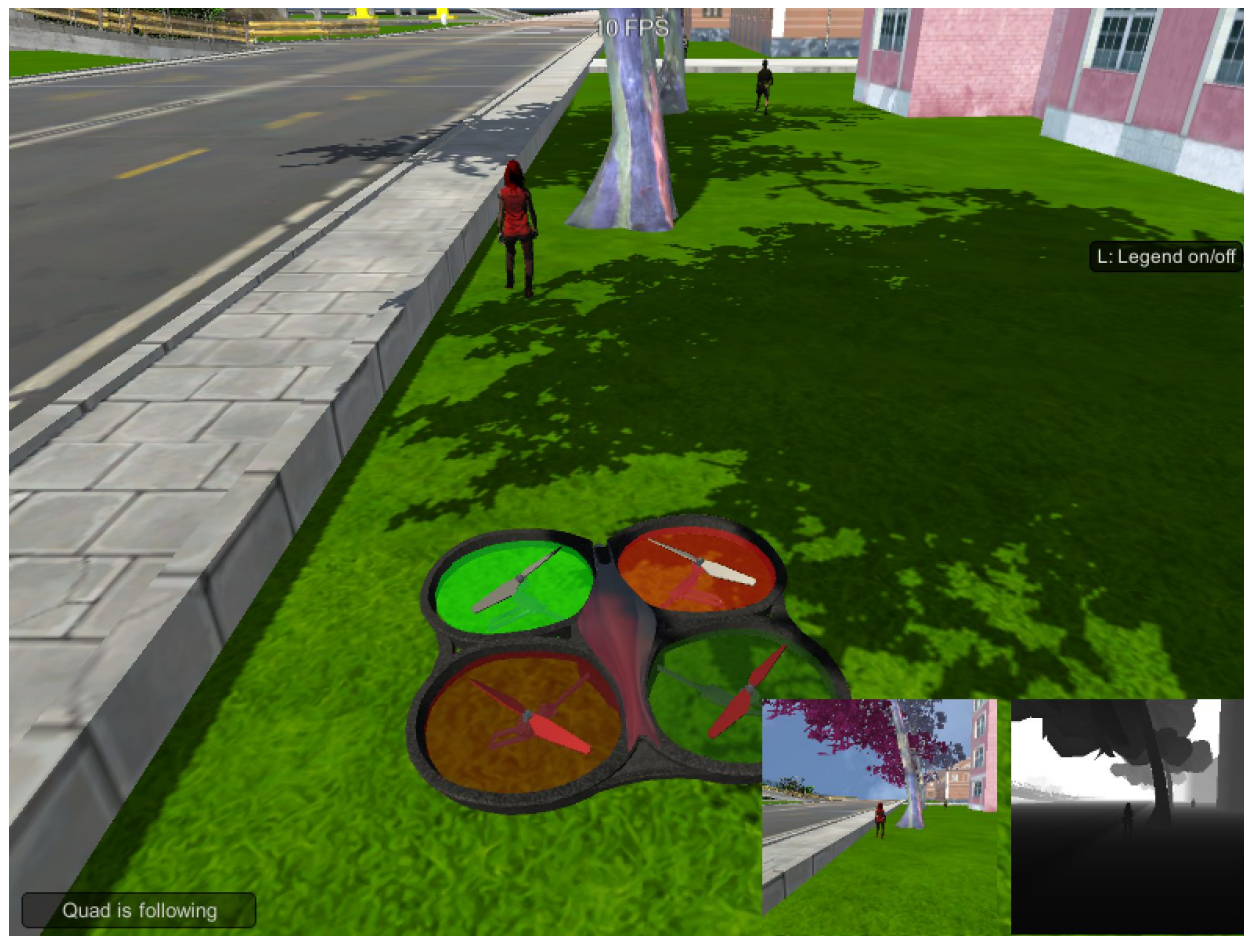


Follow Me

● Overview

- *Designing and training a FCN model that is applied to quadrotor on a local sim which aims to detect a specific person then follow that person around.*



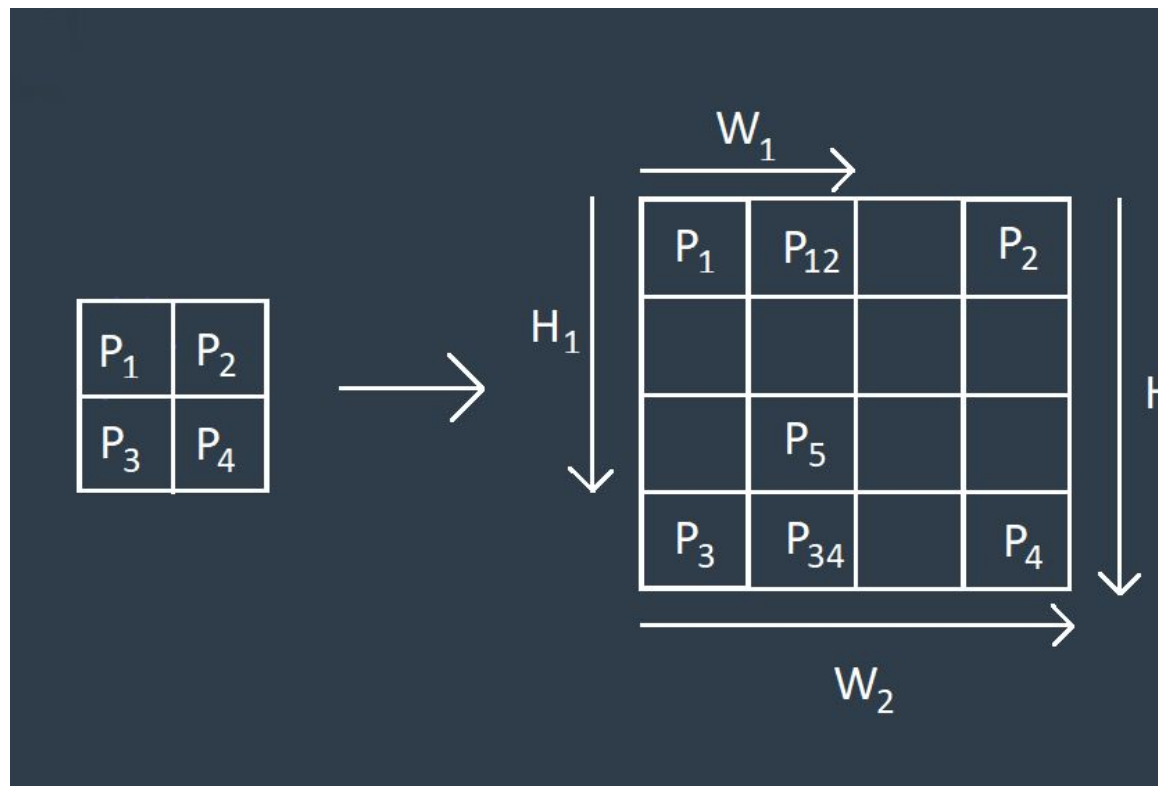
● ***Network Layers***

○ ***Encoder***

- The encoder portion is a convolution network that detect is the person in this image.
- IT apply separable convolutions which comprise of a convolution performed over each channel of an input layer and followed by a 1x1 convolution that takes the output channels from the previous step and then combines them into an output layer.
- This helps to reduce the parameters make separable convolutions quite efficient with improved runtime performance.

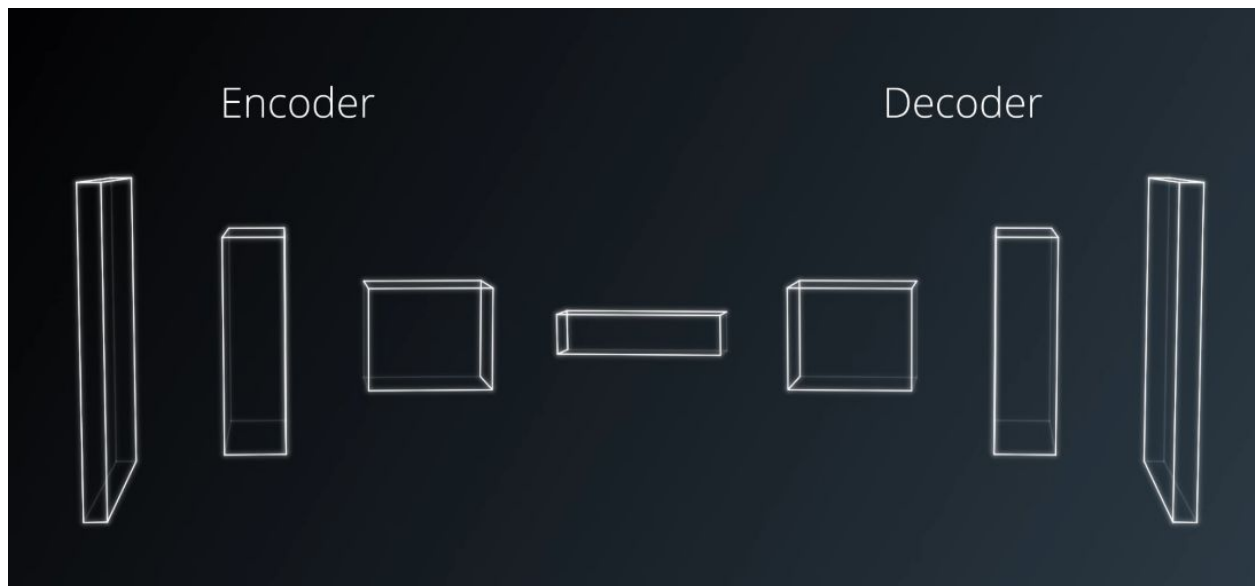
○ **Decoder**

- *Help to answer where is the person as in the encoding layers the spatial information is lost their*
- Bilinear upsampling technique is used then apply concatenation



● **FCN layer construction**

- *I used FCN because in the encoding the network losses a lot of spatial information because it narrows to the details so it losses the big picture so that's why i applied FCN so that my network can answer where is the human in the picture.*
- *First i applied 3x layers in encoding and 3x layers in decoding and after training the accuracy was 32% but after searching for how to improve the accuracy i found that most FCN depend on 2x layers which convert each layer depth to the power of two so i started with 64 for the depth then 128 and at the `conv2d_batchnorm` i went for 256 as a maximum depth*



This is the first arch first layer after encode picture from 3 depth (RGB) to 64 then to 128 then to 256 then to 512

But after that i removed two layers one from encoding and other from decoding

- This network can be applied to classify any other object but must change the training and validating data sets

- ***Hyper parameters***

- *I tried a lot first started with 10 epoch and 0.01 learning rate then i started tuning and seeing how other people tune related networks and these were final result*

```
learning_rate = 0.001  
batch_size = 64  
num_epochs = 50  
steps_per_epoch = 65  
validation_steps = 50  
workers = 120
```

- ***Final accuracy***

- *First accuracy was 32% and i started tuning my hyper parameter but*

*couldn't exceed the 40% then i
applied a major change to my
network to by reducing the encoding
and decoding layers from 3 layers to
2 layers then i could achieve 42%*