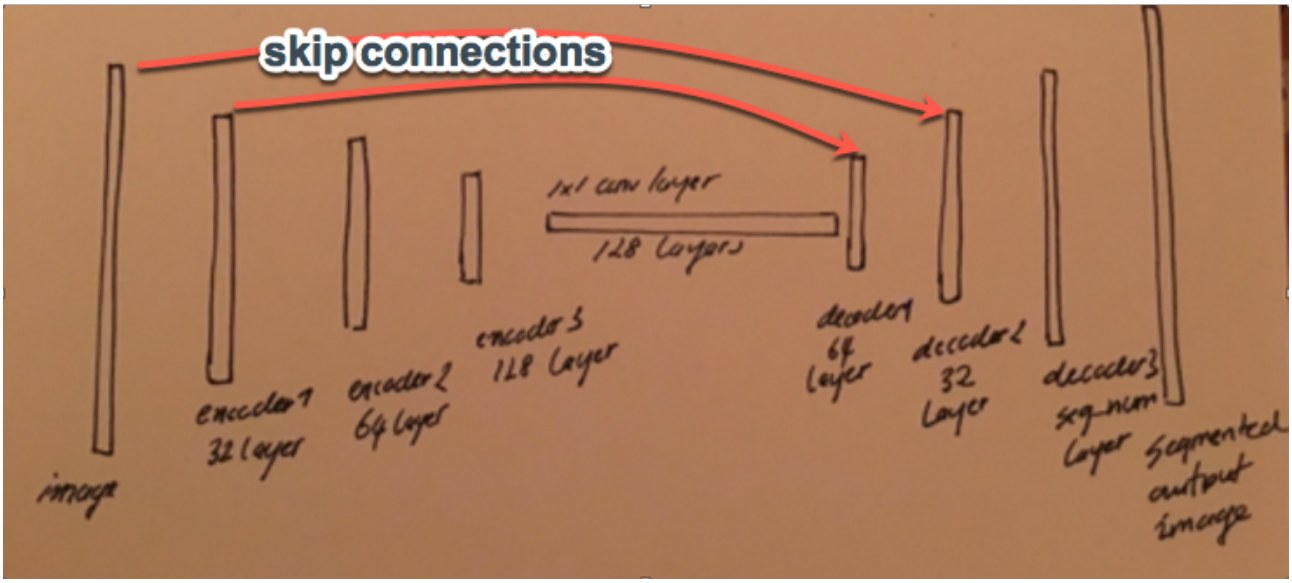
In this context, it is not enough to classify result as true or false. At the end, we need to apply semantic segmentation and put each object into a category. So, basic CNN is not enough to get this result. In CNN, with fully connected layer we can get simply boolean result, true or false. So, I have applied a FCN with encoder and decoder. With the help of the decoder, the images were upsampled and spacial information gathered.



I have decided on the following parameter set after many iteration. When I used learning rate as 0.01, After 10 epochs, model started to overfit. training accuracy continue to get better but validation accuracy get worse. I decided that 0.01 is too much and model training is too fast with this parameter. I have tried several other learning rate. I first tried 0.01 and then 0.001 and none of them works good enough for me. Then I chose 0.003 and it works fine, it seems neither too small nor too much. I finally produce good enough result with the following parameter sets.

```
learning_rate = 0.003
batch_size = 100
num_epochs = 22 (first 20, and then 2 more)
steps_per_epoch = 200
validation_steps = 50
workers = 2
```

FCN is comprised of an encoded and decoder. The encoder is a convolutional neural network that reduces to a deeper 1x1 convolution layer. 1x1 convolution helped in reducing the dimensionality of the layer. A fully-connected layer of the same size would result in the same number of features. However, replacement of fully-connected layers with convolutional layers presents an added advantage that during inference (testing your model), you can feed images of any size into your trained network. in contrast to a fully connected layer which could be for basic classification of images. The difference is the effect of preserving spacial information from the image.

In FCN, The output from the convolutional and pooling layers represent high-level features of the input image. The purpose of the fully connected layer is to use these features for classifying the input image into various classes based on training data.

FCN got really got results in computer vision task such as semantic segmentation. FCN use three special techniques. first is one by one convolutional layer. Second, up-sampling through the use of transposed convolutional layer, and third skip connection.

Lets look skip connection in more details. Skip connections allow the network to to use information from multiple resolution scales. As a result network can make more precise segmentation decision. As a result of convolution or encoding in general is you narrow down the scope by looking closely at some pictures and lose the bigger picture as a result.So, even if we decode the output of the encoder to the original image size some information has already been lost. Skip connection are a way of retaining the information easily. The way skip connection works to connect the output of the one layer to a non-adjacent layer. These skip connection allow the network to use information from the multiple resolution.

The goal of the encoder is to extract features from the image. It is basically a CNN and it can be replaced with famous CNN architecture such as ResNet to apply transfer learning.

In FCN, Decoder is used for upsampling the data. Decoder upsample layers to higher dimensions or resolutions. There are several ways to achieve upsampling such as bilinear upsampling.Bilinear upsampling is a resampling technique that utilizes the weighted average of four nearest known pixels, located diagonally to a given pixel, to estimate a new pixel intensity value. The weighted average is usually distance dependent.

we need to have spacial information at the end of the model. We used FCN and encoder/decoder.

Since we trained the model only with humans, the model works well with humans. Our quadcopter could follow a human and identify it in a human crowd. But, it could not probably work as well with dogs, cars or any other objects other than human. Because during training, we only use data of humans. But we can improve the model and train it with other objects as well. Then, model would work better and would be more generic.

You can find the model in this path(It has more than 40% accuracy):
https://github.com/clockworks/nd-robond-followme-deeplearning-project/tree/master/data/weights.h5

You can find jupyter notebook for implementaion.
https://github.com/clockworks/nd-robond-followme-deeplearning-project/blob/master/code/model_training.ipynb

You can find model_training.html which shows the final accuracy here:
https://github.com/clockworks/nd-robond-followme-deeplearning-project/blob/master/model_training.html

SOURCES:
[1] - https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/