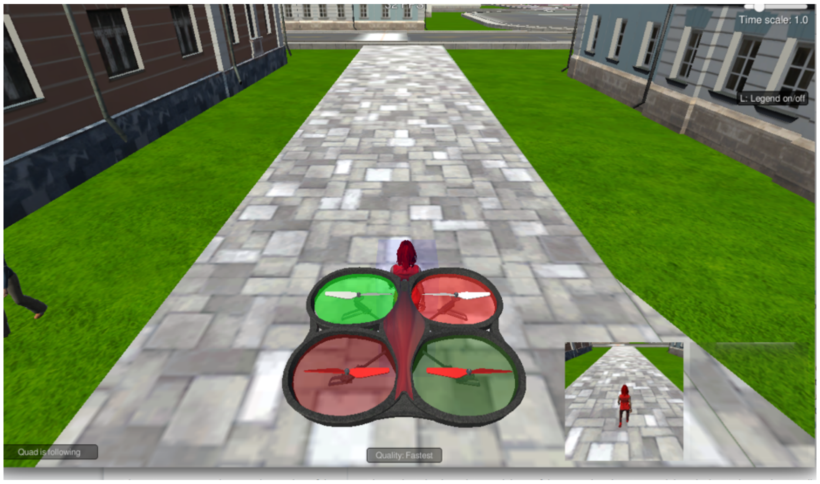
Follow Me Project

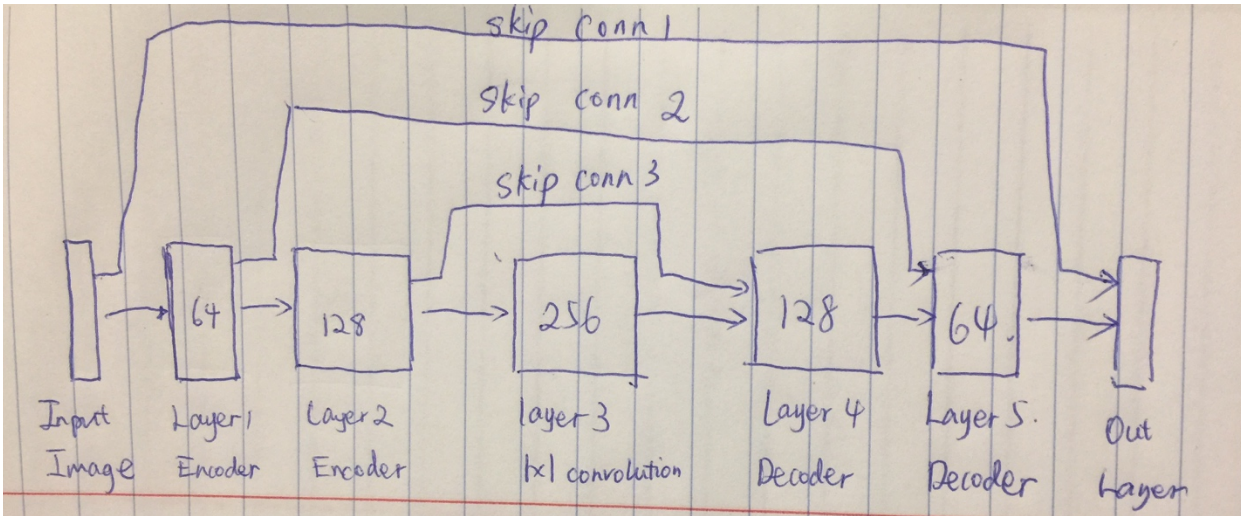
Introduction

In this project, I trained a deep neural network to identify and track a target in simulation as shown in Figure 1 . The target is the woman in red. The trained neural network model is used in application so that the drone can identify the target in the simulated scene and follow the target. The model achieved IoU (Intersection Union) final accuracy of 0.45. In this project, the training dataset are images captured by a patrolling drone in the simulated environment. The report includes discussion of network architecture, parameter tuning and limitations.



Network Architecture

In this project, an encoder-decoder architecture of a fully-convolutional network is used. The encoding stage and decoding stage are connected using a 1x1 convolutional layer.
The encoder extracts feature from images and learns details about images for classification. It does this via multiple layers and gradually learn more and more complex shape and structure. The encoder has pooling that down-samples the images to increase generalization and avoid overfitting. But it loses some information and it reduces spatial dimension.
The decoding stage up-sample the features into the same dimensions of the original image. The upsampling uses bilinear interpolation. The 1x1 convolutional layer maintains spatial information and connect encoding and decoding stage.
Along the way, there are also skip connections which connect some non-adjacent layers together. For example, the output of the first encoder layer is directly connected to the input of the final decoder layer. This has the advantage of retaining more information that may be lost during multi-layered encoding and increasing accuracy of the final segmentation. The final stage after the decoding is a convolutional output layer with softmax activation to make decision.



The network architecture is depicted in the Figure 2. To sum up, it has the features:

- Encoder with pooling
- 1x1 convolutional layer
- skip connections
- Decoding: Bilinear interpolation up-sampling with transposed convolutional layer

Training and Parameter Tuning

The data for training and testing and validation are downloaded from Udacity website. Without much fine tuning, the network can achieve approximately 0.38 accuracy. So the network architecture is unchanged and the task of increasing accuracy goes to fine tune the parameters. The parameters are learning rate, number of epochs, batch size, steps per epoch, validation steps per epoch

Number of epochs, steps per epoch, validation steps per epoch

Number of epochs is the number of iteration of training on the same data set. A large number will overfit the data and a small number may not underfit the data. The final number is chosen to be 100.

Steps per epoch and validation steps per epoch are predetermined by the number of training images and validation images. Steps per epoch is number of training images over batch size. Validation steps per epoch is number of validation images over batch size.

Batch size

To run training over batches instead of the entire dataset can save memory and reduce training time. Batch size of 40 is chosen. Batch size and learning rate is related.

Learning rate and optimizer

Learning rate starts with 0.001 and is slightly increased to 0.005. A low learning rate may not learn enough from the training set and be stuck at local optimum and a large learning rate may overshoot. To achieve the best performance, the Adam optimizer is chosen.

Limitation and Conclusion

The trained network model can only identify the target woman in red. If the goal is to track other targets, the new data sets have to be recollected and the model has to be retrained.

In this project, only the Udacity datasets are used for training and validation and it achieved 45%. By collecting more datasets, the model will be more accurate in predicting.