Deep learning (Follow Me) project

Subject: Robotics Software engineer

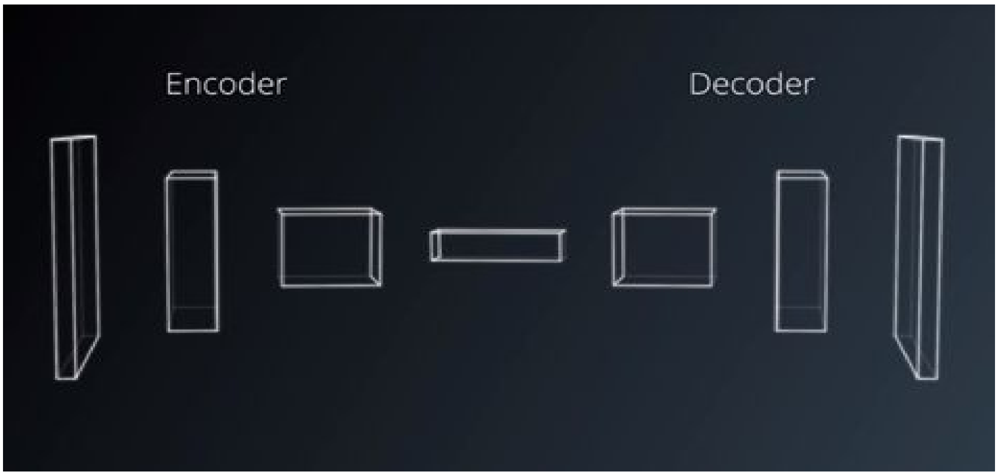
Name: Pratirodh Prasai

#### The write-up for the project

### Understanding of the Network Architecture

In the class, we discuss about different type of neural network; Deep Neural Networks, Convolutional Neural Networks and Fully Convolutional Networks. To complete this project, I chose Fully Convolutional Neural Network(FCNs). FCNs method is effective here to track a hero, it can preserve the spatial information throughout the whole network.
As discussed in class, typical convolutional neural network could be great architecture for a classification task. But here we want to find the person within a frame location. This kind of task is much more difficult to solve since fully connected layers don't preserve spatial information. However, if we change the C from connected to convolutional, we can integrate convolutions directly into the layer to create a fully convolutional network. It solves the problem to find person. furthermore, FCNS don’t care about size of object and it will work on images of any size.

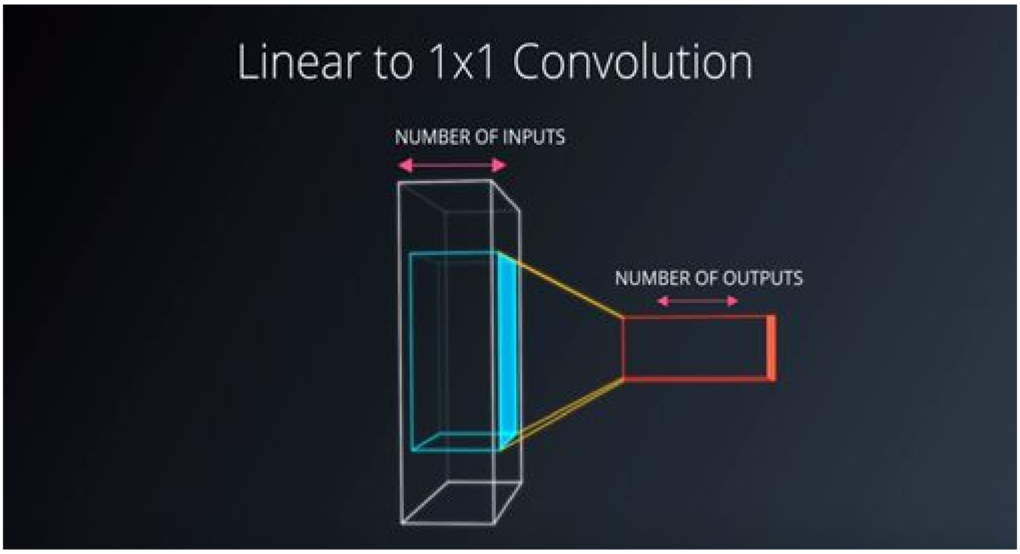
##Encoder and Decoder



Fully Convolutional Networks can have dived in two-part, encoder and decoder. These two processes comprises three special techniques, these are 1x1 convolutional layers, transposed convolutional layers and Skip Connection method.

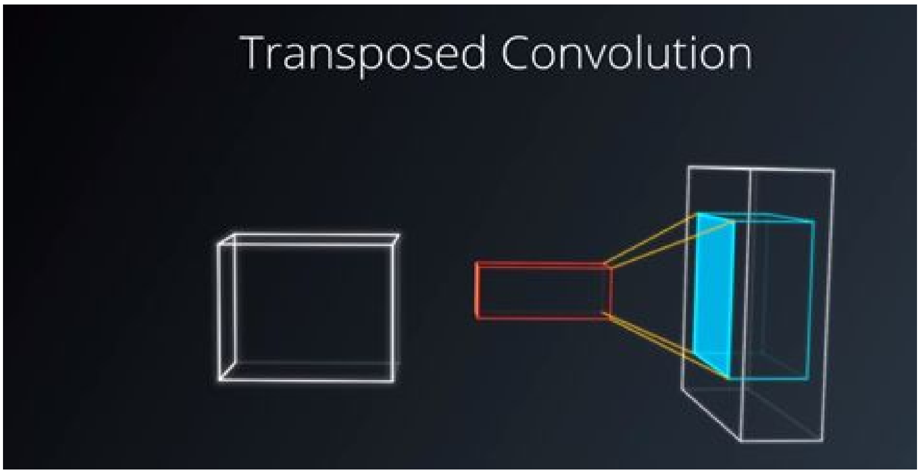
Encoder is series of convolution layers like VGG and Reset. Encoder used to get the feathers of the image from the model, where decoder help to recover the loss spatial dimension and bring image in original size.

#1x1 Convolution Layer



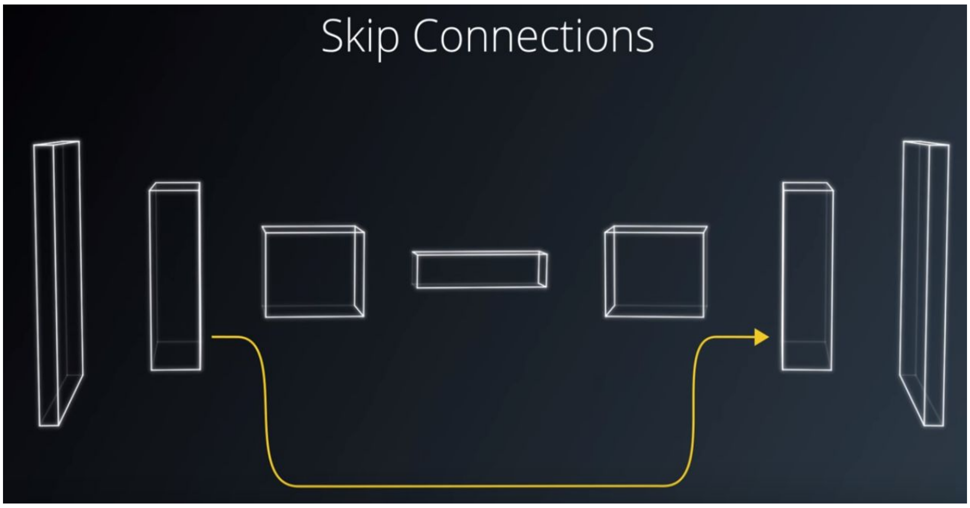
This is a regular convolution, with a kernel and stride of 1. It allows the network to be able to retain spatial information from the encoder. Its allows to reduced the dimension for classification while retaining spatial information.

#Transpose Convolution



It is the part of decoder process in FCNs, can define as a reverse convolution in which the forward and backward passes are swapped

#Skip Connections



It helps the network to retain information from prior layers that were lost in subsequent convolution layers, as a result, the network can make more precise segmentation decisions. The use of skip connection in FCNs, for example, the output from the first encoder is connected to the input of the final decoder.

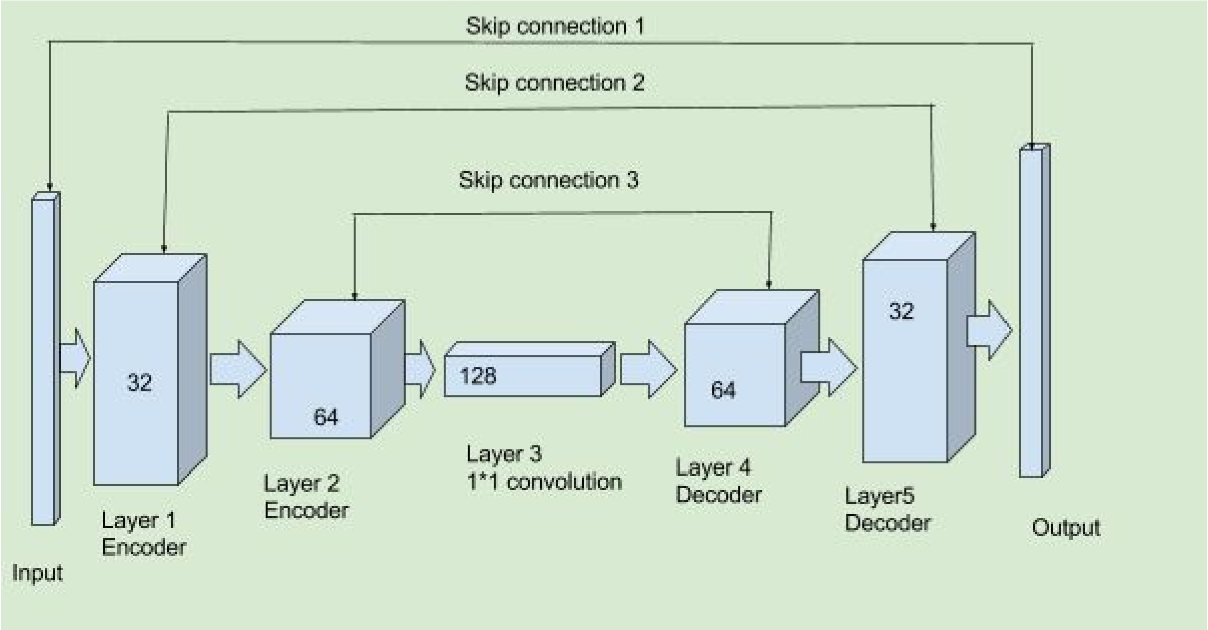
##Code for Encoder and Decoder

```
def fcn_model(inputs, num_classes):
   # TODO Add Encoder Blocks.
     # Remember that with each encoder layer, the depth of your model (the number of
filters) increases.
   encoder_1 = encoder_block(inputs, 32, strides=2)
   encoder_2 = encoder_block(encoder_1, 64, strides=2)
   # TODO Add 1x1 Convolution layer using conv2d_batchnorm().
   conv2d_batchnormed = conv2d_batchnorm(encoder_2, 128, kernel_size=1, strides=1)

   # TODO: Add the same number of Decoder Blocks as the number of Encoder Blocks
   decoder_1 = decoder_block(conv2d_batchnormed, encoder_1, 64)
   decoder_2 = decoder_block(decoder_1, inputs, 32)
    # The function returns the output layer of your model. "x" is the final layer obtained from
the last decoder_block()
                      return layers.Conv2D(num_classes, 1, activation='softmax',
padding='same')(decoder_2)
```

### Design of neural network

I used FCNs model of two layers of Encoder and Decoder with 1*1 convolution layers. The detail of Fully Convolutional Network model is shown in figure below.



The value of filter size, stride for first and second can see above in code section, where I put the standard kernel and stride size value 1.

my choices for model hyperparameters
learning_rate = 0.0015
batch_size = 100
num_epochs = 40
steps_per_epoch = 200
validation_steps = 50
workers = 2

The first part of script was taken from Segmentation-Lab practice from class room and i also reach out to slack community for better solution. Initially, I chose batch size of 25 and num of epochs 15, where outcome of final score less than 30%. After that i increase batch size to 100, epoch to 25 and learning rate to 0.01, then performance was increase to IOU 39.3 %.Which is just below the submission requirement . later try to modify the number of filters from 34 to 64, and 64 to 128 of layers with same parmenter but the result get more worse.

As i read on slack, one friend suggest not to Increasing the steps per epoch and validation to high because it rarely enhance increased the model accuracy, and it significantly increased the training time. Thus it did not increase them over 200 and 50, respectively.

Finally, increasing to number of epoch to 40 and slowing learning rate to 0.0015 give result of 41.3 %.

###Discussions

Here I use the image date provided by Udacity. In project, I need to track the “hero” as target person from the many people. Here, Intersection over union(IOU) metric used to measure the accuracy level of performance. The model was trained using an AWS EC2 p2.xlarge instance, The final score of my model is **41.3%**.

Here the model is only trained using classified images of people to identify target in red outfit. I believe the model doesn't work in same state. To follow other objects, we should provide the architecture with classified images of that object, such as cat,dog, car ete. Here we only sumiluate the classified image of humans to distinguish between the red outfit and other people.

## Future improvement

first i decided to training model on my CPU (3.4 GHz i7, 16 GB Ram), but it took 24 hours . it works well without crashing but the performance was bad. After i set up account with AWS, i switch to GPU to train the model. So it's not practical to use CPU method.

As suggest from reviewer and students form slack Robotics group , The network architecture of the project could be enhance having more deeper network with more layers. It would help learning smaller detail which could improve to find the target form far away. As I try to add some extra convolution layer, while i train the model throw few error, so I use previous model. However, If we apply more layer, it could improve performance.

I think, The amount of right data is importantly to train the model, we could collect more data form simulator targeting the “hero” which could enhance the output of model.

Thank you!