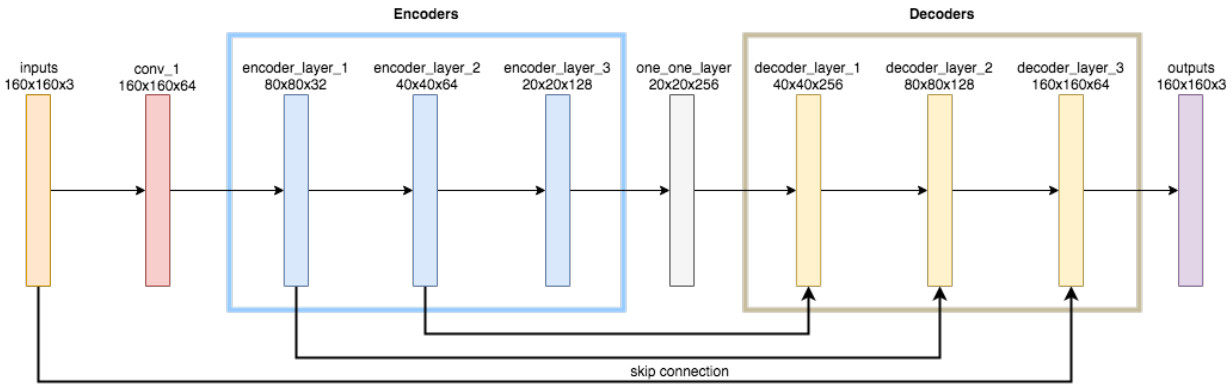
Follow Me Report

I. Network Architecture



The above is the final network architecture designed to be used for the **Follow Me** project which results the final score of 42%.

The data from inputs go through a convolution layer(using conv2d_batchnorm), then three encoder layers. After being convolved by one_one_layer, it is decoded by the next three decoder layers and then finally be convolved to the build the output image.

Layer's information:

- inputs: the binary data from 160x160(width x height) image and the depth 3 of three color channels.
- conv_1: after being convolved by (filters=64, kernel_size=4, strides=1) the result layer had the same width x height as the input layer but with the depth of 64.
- encoder_layer: using separable_conv2d_batchnorm to reduce the width and height of the activation by the factor of 2. The separable convolution helps reducing the number of parameters by multiplying two individual kernels. The relu activation will convert linear function to non-linear function in a simple, easy to be calculated way.
- one_one_layer: the fully connected layers require many parameters but don't preserve the spatial information of the activation layer. The one one layer could replace the fully connected layer with less parameters and also preserve the spatial information of encoder layer.
- decoder_layer: after go though the one one layer, data be decoded by decoder_layer which used bilinear_upsample to scale the activation by the factor of

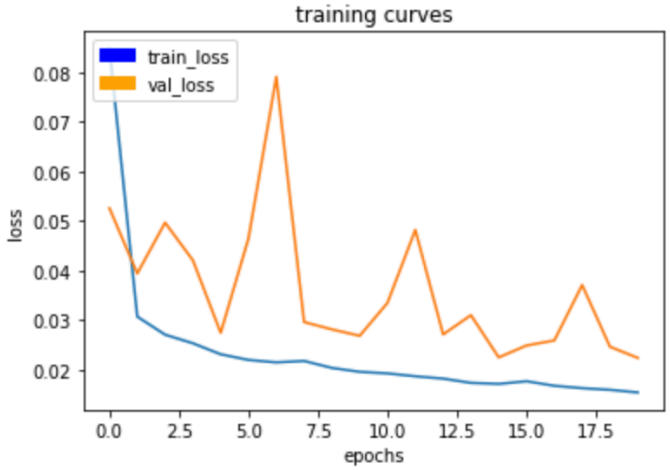
two. And in-order to retain information loss from encoder layer, the skip connection is directly tied up with decoder layer from encoder layer or input layer. The first two inner layers in decoder_layer is not learnable(only transform the shape of the activation), so there is the last separate convolution layer which is the learning part of decoder layer.
- ouputs: the output image with the same shape of input data is formed by using convolution and optimizer.

II. Optimizing

In order the have the final score above 40%, many different network architectures and hyper parameters are tried. The first results mostly show either over-fitting(too many parameters) or under-fitting(too small number of parameters).

- To avoid over-fitting or under-fitting, the three encoder layers, three decoder layers are finally chosen, three skip connections also used to retain spatial information of activations.
- Separable convolution is prefer to regular convolution because it use less parameters(form big kernel by the multiplication of two smaller kernels) thus may avoid over-fitting.
- Batch normalization will normalize activations before going through the next layer which boost the training process since gradient descent could go down faster than with non-normalized data.
- There is also more data be added into training set(3000 more images) and validation set(1000 more images), the data is captured in various of sizes, angles with and without the hero.
- The final learning rate is 0.005, the original value 0.001 is too small and usually lead to over-fitting.
- The number of epochs is 20 which usually don't affect the training result a lot, the model is optimized mostly after 5-7 epochs.
- The batch size is 20, increasing the batch size usually slow down the training process and lead to over-fitting. Smaller batch size can be trained faster but results under-fitting.
- Steps per epoch is 465 ~ 9247(images) / 20(batch size)
- Validation steps is 75 ~ 1500(images) / 20(batch size)

The final training and validation graph look like:




```
465/465 [==============================] - 227s - loss: 0.0154 - val_loss: 0.0223
```

III. Application

In this project, the application of using fully convolutional neutral network and 1x1 convolutional is to segment the area of concerns in the input image. By using 1x1 convolutional layer instead of fully connected layer, the network could answer the question where is the hero in the image by retaining spatial information rather than flatting all parameters. Moreover, fully convolutional neutral network could re-produce the same image's shape by down-sampling and up-sampling the input data so we could know where exactly the hero in the original image.
Beside, the above techniques is also used for image reconstruction, image compression.

IV. Improvements

With the result training graph, we may realize that the model is still a little bit under-fitting, the validations loss is much more higher than the training loss. In-order to improve that, we may add more convolutional layer, one more encoder layer, decoder layer or add more inner convolutional layer inside decoder layer. The above actions may lead to over-fitting. If this is the case, the dropout(preferable) or max-pooling could be the techniques to rescue us and reduce over-fitting.

Obtaining more training, validation data could also helps since neutral network could work well with pre-existing trained patterns, if they encounter new patterns, they are likely to predict wrongly.
One more limitation of neutral network is it is very depended on hyper parameters, some set of data could only work well with some set of hyper parameters and network architectures due to the uncertainty of data and optimizer. To overcome it, we could try to automate the hyper parameters selection phase and let the computer chose the best for us, but it will also result in more training time and iteration required.