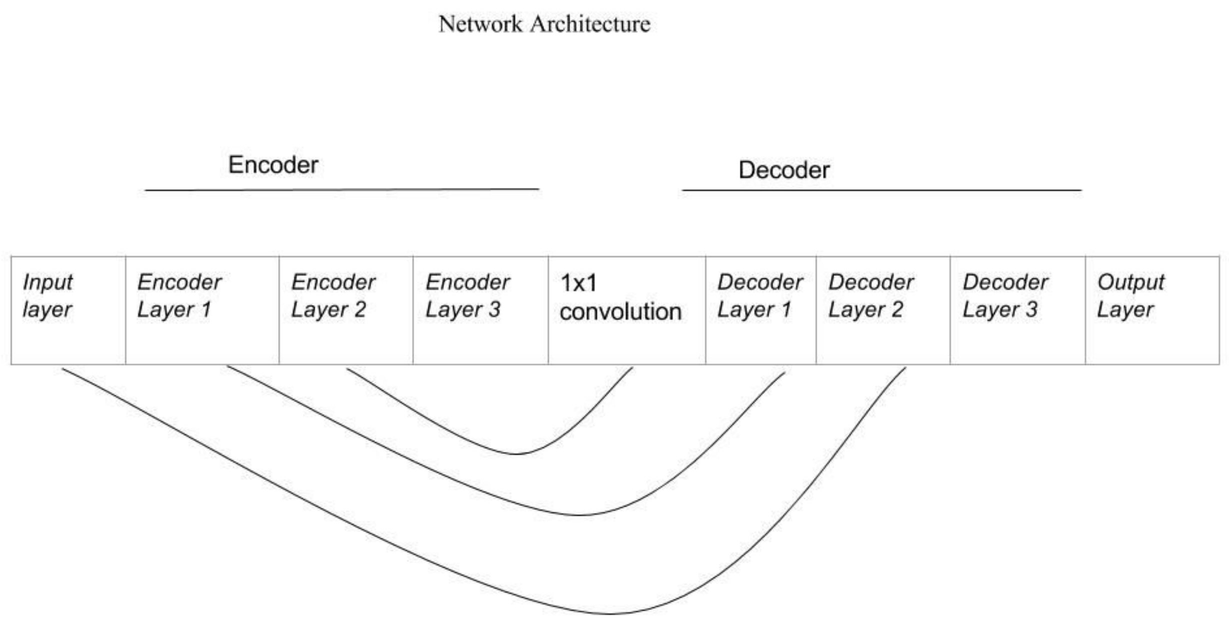
Network architecture :

The network is composed of three encoder layers, three decoder layers and one 1x1 convolution layer between them.



The encoder layers extract the features from the input, the encoder output is then passed to the 1x1 convolution layer. After that the decoder layer takes the output and up-sample the extracted features using transposed convolution.The last thing is the skip connection which is demonstrated by the curved line connecting two non-adjacent layers. The skip connection provide a way to restore some of the information lost during encoding.

HyperParameters:

Learning Rate: The learning rate value reflects the step size of the gradient descent algorithm, high learning rate value means we are taking large step each time.
Batch Size : The batch size is the number of images in one forward/backward pass, Increasing the batch size depends on the available memory. On the other hand smaller batch size make the estimate of the gradient less accurate.
Number of epochs: The number of epochs is how many times the algorithm pass through the whole data.
Steps per epochs : the number of batches of training images that propagate through the network in one epoch.
Validation steps : the number of batches of validation images that propagate through the network in one epoch.

To choose the values of the hyper parameters several trials where made.

Trial 1 :

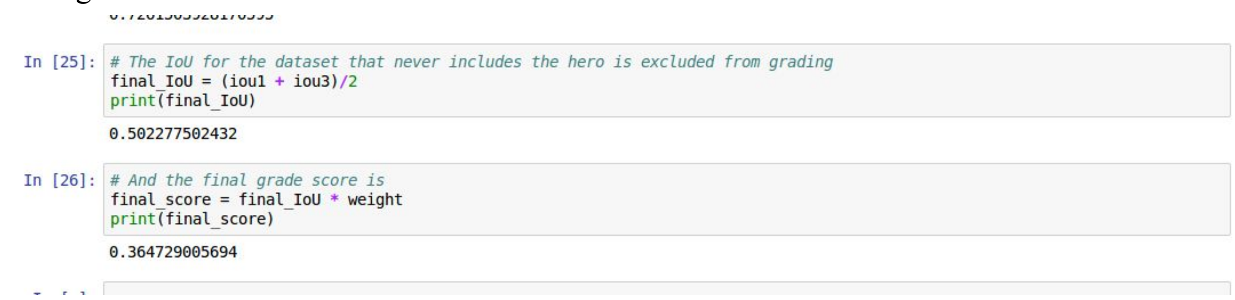
Num_of_epochs = 10

Steps per epoch = 50

Validation steps = 20

Learning rate = 0.01

Final grade = 36 %



```
In [25]: # The IoU for the dataset that never includes the hero is excluded from grading
         final_IoU = (iou1 + iou3)/2
         print(final_IoU)

         0.502277502432

In [26]: # And the final grade score is
         final_score = final_IoU * weight
         print(final_score)

         0.364729005694
```

Trail 2 :

Leaning rate = 0.01

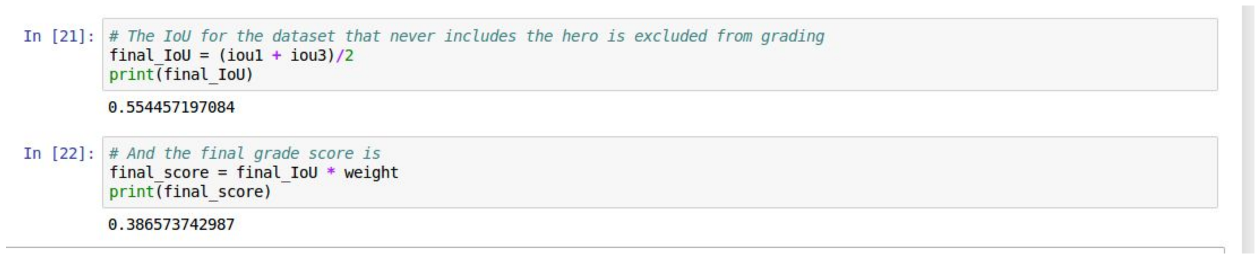
Num_of_epochs = 15

Final grade = 32%

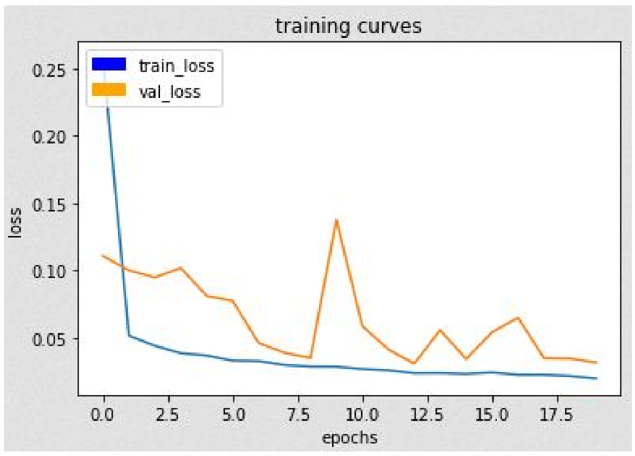
Trial 3:

Num_of_epochs = 20

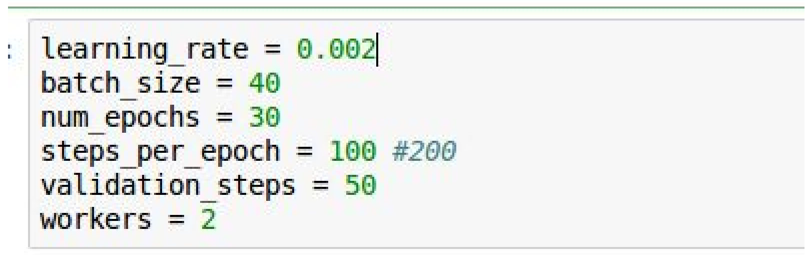
Final grade = 38%



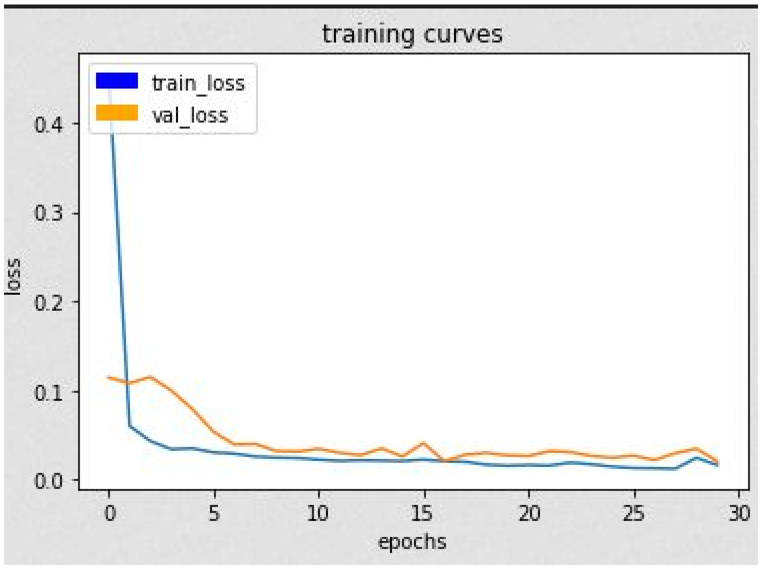
```
In [21]: # The IoU for the dataset that never includes the hero is excluded from grading
         final_IoU = (iou1 + iou3)/2
         print(final_IoU)

         0.554457197084

In [22]: # And the final grade score is
         final_score = final_IoU * weight
         print(final_score)

         0.386573742987
```



Trial 4:



```
: learning_rate = 0.002
  batch_size = 40
  num_epochs = 30
  steps_per_epoch = 100 #200
  validation_steps = 50
  workers = 2
```

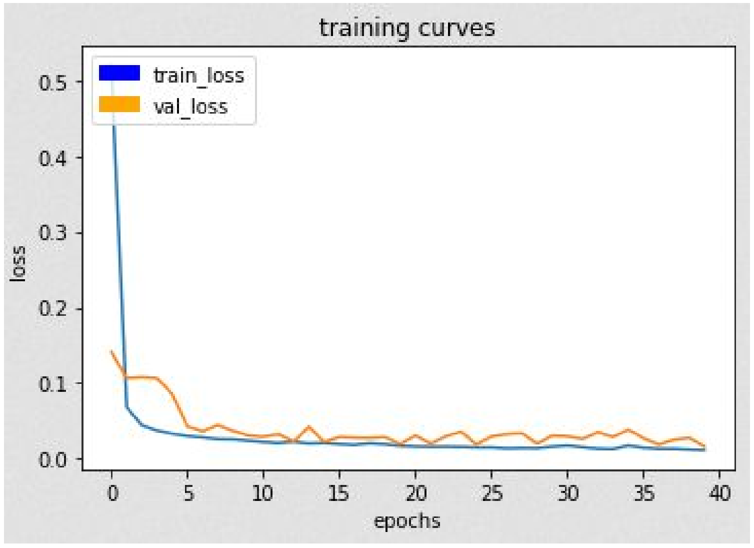


Final grade = 37 %

Trial 5:

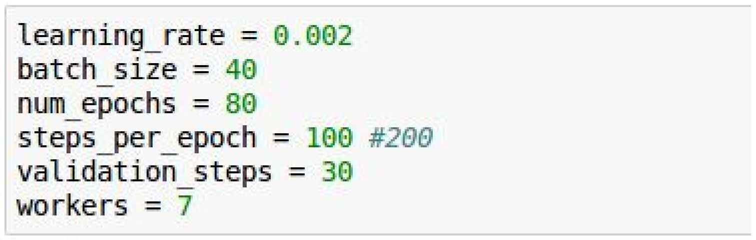
The same Parameter as the previous trial.

Num_of_epochs = 40

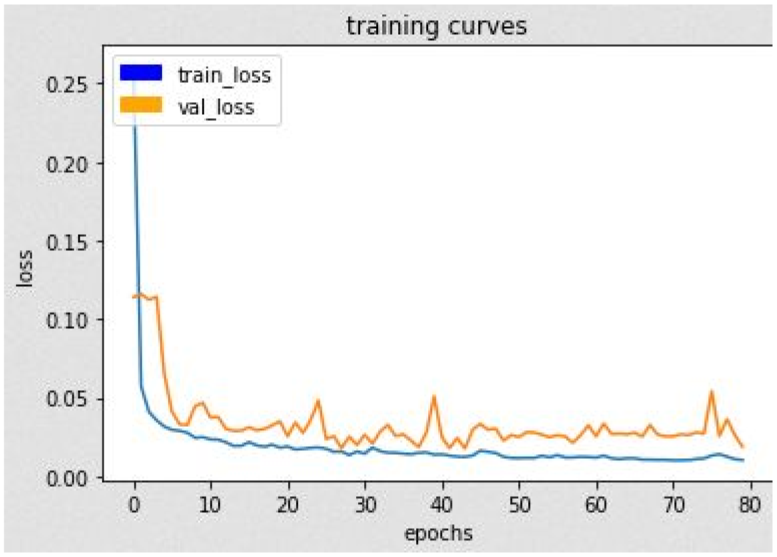


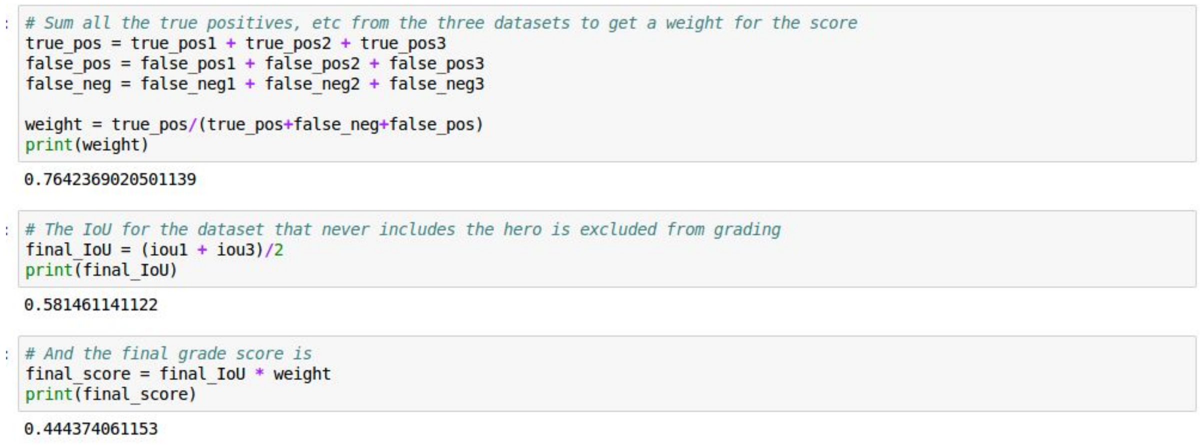
Final grade = 38.9%

Trail 6:



```
learning_rate = 0.002
batch_size = 40
num_epochs = 80
steps_per_epoch = 100 #200
validation_steps = 30
workers = 7
```





```
# Sum all the true positives, etc from the three datasets to get a weight for the score
true_pos = true_pos1 + true_pos2 + true_pos3
false_pos = false_pos1 + false_pos2 + false_pos3
false_neg = false_neg1 + false_neg2 + false_neg3

weight = true_pos/(true_pos+false_neg+false_pos)
print(weight)
```

```
0.7642369020501139
```

```
# The IoU for the dataset that never includes the hero is excluded from grading
final_IoU = (iou1 + iou3)/2
print(final_IoU)
```

```
0.581461141122
```

```
# And the final grade score is
final_score = final_IoU * weight
print(final_score)
```

```
0.444374061153
```

One by one convolution Vs fully connected layer :
Two structures of the Network might be used :
1-Encoder layer followed by fully connected layer.
2-Encoder layer followed by 1x1 convolution layer then decoder layer.
When using the first configuration we can solve classification problems like is this a T-shirt, is it a duck or walrus but we can't answer questions like where in the scene is the duck because in using fully connected layer we concentrated on the features itself and neglected the spatial information.
To answer the second question we need to use the second configuration. In the second configuration we preserve the spatial information because we don't flatten the output like we do in the fully connected layer.

Encoding/Decoding images :
Encoding the image is the process of extracting the features to feed it to a fully connected layer or to 1x1 convolution layer. In encoding the image, we concentrate on the features itself and might lose the bigger picture.
Decoding an image is the process of upsampling the output to a desired dimension, The decoding process is used along with the fully convolutional layer and the skip connection techniques to retain some of the lost information.

Following the dog/cat :

The model we developed can't be used to follow a dog/cat because we didn't train it to recognize dog or cat but we trained it to follow the hero. To follow anything else we need to train another model and feed it with pictures for what we want to follow.

Future Enhancements:

1- Getting more data from the simulation specially for the target at a distance.
2- Increase the depth and maybe adding more encoding/decoding layers.
3- Increasing the number of epochs.
4-Lowering the learning rate (it will take more time but will achieve a better accuracy).

References :
[1]https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
[2]https://stackoverflow.com/questions/4752626/epoch-vs-iteration-when-training-neural-networks
[3]https://stats.stackexchange.com/questions/153531/what-is-batch-size-in-neural-network

[4]https://towardsdatascience.com/epoch-vs-iterations-vs-batch-size-4dfb9c7ce9c9