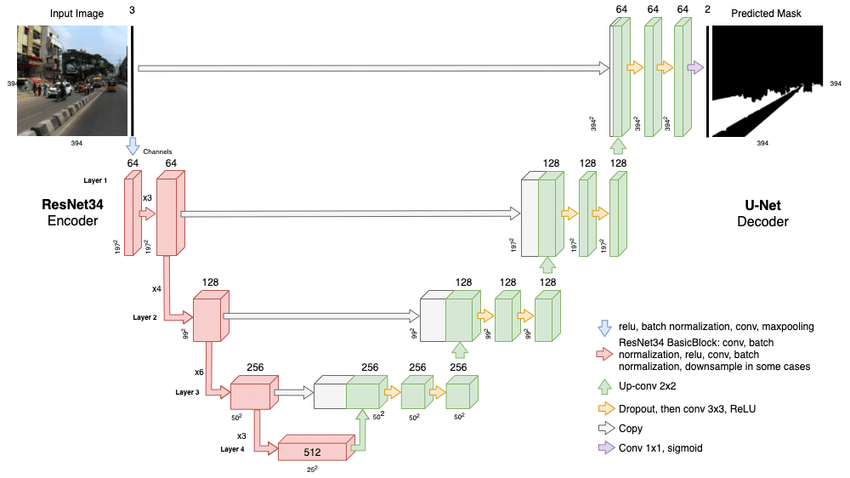
**A concise report**

**Introduction**

This report documents the training process, including model architecture details, training configuration, evaluation metrics and their achieved values, benchmarks and any relevant visualizations.

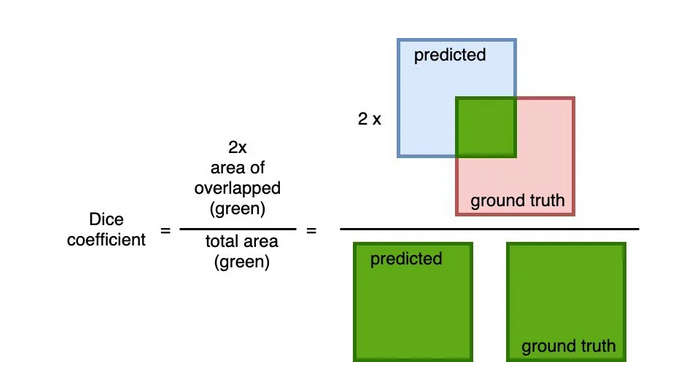
**Architecture**

For this Project I chose **PyTorch**. In my opinion, it has an advantage over others in ease of use. Next, I needed to choose a model architecture for semantic segmentation. And I chose **U-Net** architecture to solve this problem because of its efficient ability to take into account both local and global contexts in images. Due to its architecture, **U-Net** is able to detect and preserve object details, which allows for accurate and clear segmentations. also I used **ResNet34** as an encoder. In my opinion **ResNet34** has an ideal balance between speed then accuracy.



**Loss Function**

For the loss function and evaluation metrics I used **DiceLoss**. From the definition, we can see that the Dice coefficient increases the weight of the overlap in both the numerator and the denominator. This means that as the overlap increases, the Dice loss function provides a greater gradient flow, which encourages more precise segmentation.



**Training**

As optimizer I used **Adam**. Adam is optimal for me. Also, since I have limited computing power, I reduced the images by a factor of 3 and they were trained with a size of 256x256. But I think training will almost not change from this.

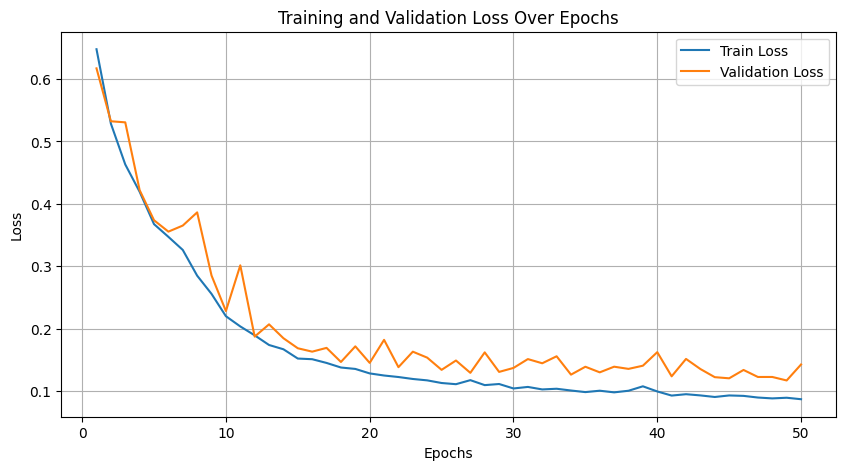
There are **hyperparameters**:

EPOCHS = 50

BATCH\_SIZE = 16

LR = 0.03

I set **BATCH\_SIZE** value as **16** because during the experiment I found out that it most efficiently uses computing resources, and also provides good accuracy.



As we can see, the graph looks adequate and in general we got good results.  
  
Train accuracy: 0.089;

Valid accuracy: 0.117;

Test accuracy: 0.152;  
  
A little bit overfitting, but I think it’s ok.

**Results**

Left – RGB image, center – ground truth, right – predicted by my model