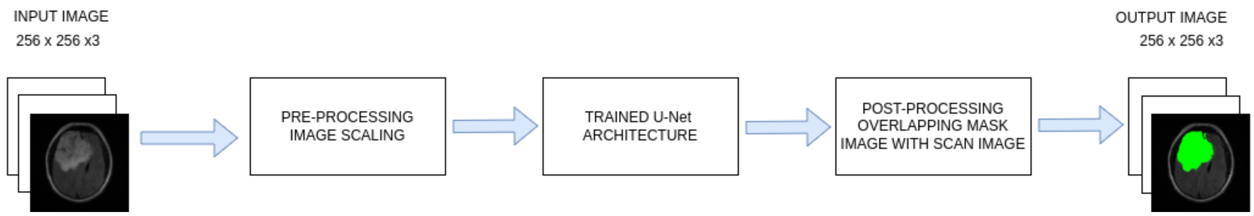
Inference Pipeline.

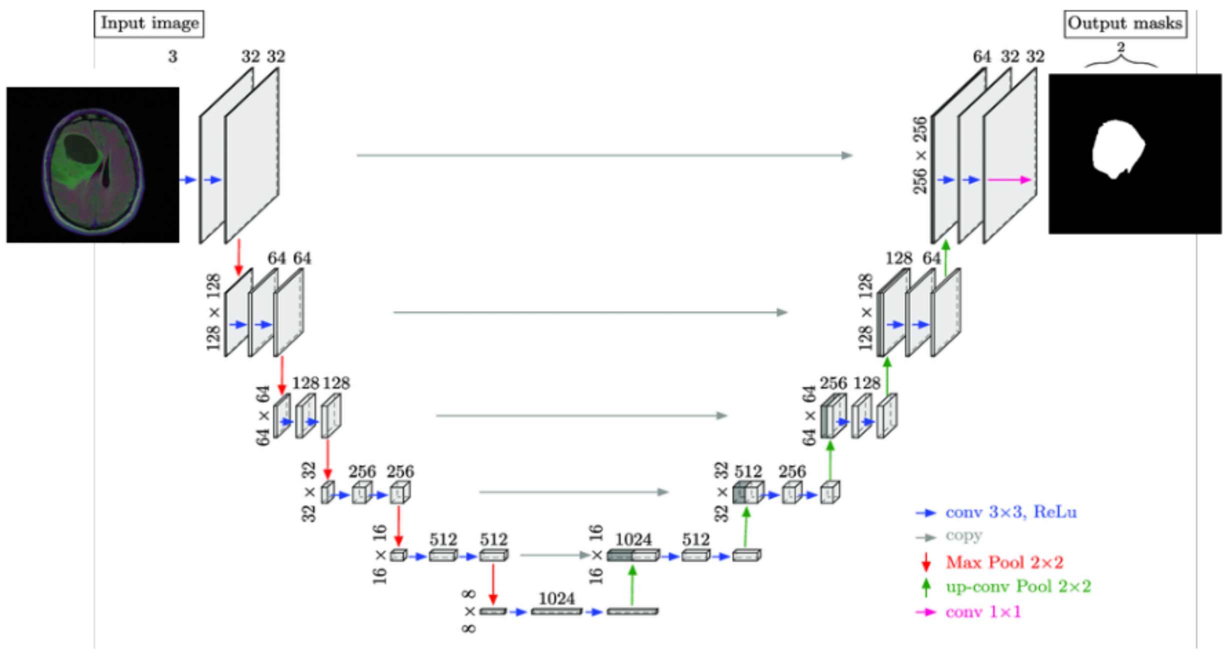
I was given 110 patient data files in separate folders in this case study, which include their brain mri scan and their respective mask image for finding abnormalities in the mri scan. To clean up our data, I create two folders, one for scan images and the other for mask images.

Then we did some image analysis to gain some insights from the data we were utilising, and as a result, we came up with some interesting ideas such as the location of abnormality in the scan, the dominant hue of abnormality in the scan, the percentage of pixel classified as abnormality, and so on.

Then we created a dataframe containing scan image path, mask image path, and a label that tells us whether the scan image is abnormal or not. The purpose of introducing this column is that, because our dataset is imbalanced between abnormal and normal images, our train and test distributions may change if we randomly split the data. To counteract this, we introduced a label column so that we can stratify split our dataset. Then, on an imagenet dataset, we load a pre-trained U-Net architecture and freeze its encoder weights. Then we import our dataset from the image generator (flow from DataFrame), do prior image scaling, and augment the training image.

Then, for 50 epochs, we train our Unet architecture, resulting in a train IOU score of 0.42 and a test IOU score of 0.4.





Conclusion

After examining the model's predictions, I discovered that the best forecasts were made for scans in which the abnormality is separated in region by a considerable bold hue and can also be easily classified by our nacked eye. Despite the fact that our model performs admirably in almost all of the scan, with extremely high precision.