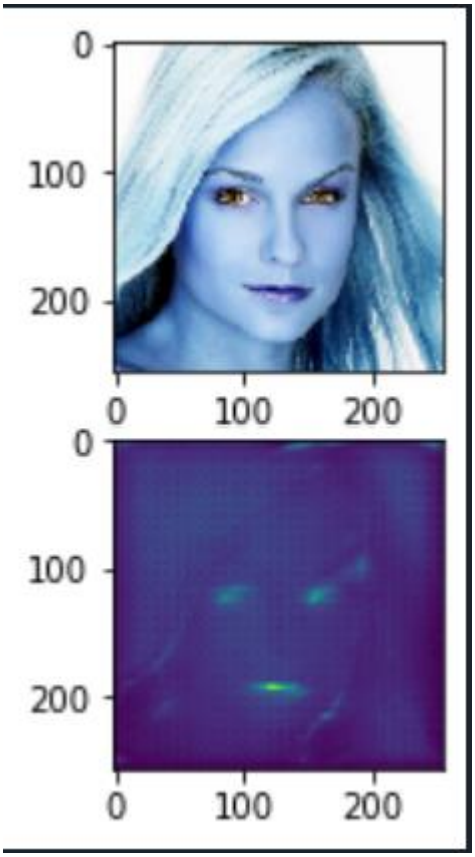
Eyes and lips Segmentation

Approach: I have used UNet with Resnet 34 as encoder for my model. UNet can be divided into 2 parts one part down samples the image and other part up samples the image. For down sampling instead of using vanilla UNet architecture, I have used pre trained Resnet34 and I have coded the whole model from the scratch . I have used SGD optimizer with learning rate of 3.00000002e-03, momentum=0.8, weight_decay=0.0003 and 25 epochs. I have used Dice_coeff loss as a loss function(during training I have tried with multiple loss functions like binary cross entropy, BCEwithlogisticloss) and IOU score for testing.

Data Preparation: I have trained the models on colab, with data on the google drive. Due to space constrain I have taken 2000 images in total for training and testing i.e 1900 images for training and 100 for testing. I have created np array for images, masks and saved them to the drive. I haven't done much of data pre-processing except for normalizing the images using mean and standard deviation of the images.

Results: After training the model, we got masks as the output. Below is the image of one of the results.



But overall, the results are not very promising. The IOU score on the test set 0.22 with maximum 0.4. This shows model is not performance is not good. Due to time constrain, I haven't done anything substantial to improve the performance of the model, but I have few ideas which could help in improving the results.

Steps to improve the Results:

- The Resnet 34 model which I have used for encoding was trained on image net, so instead of using mean and standard deviation of images of dataset I could have used mean and standard deviation of imagenet images. This could have improved the results.
- Initially I did freeze all the layers of resnet 34 and trained model by updating weights of up sampling part, but it didn't give me good results so I unfreeze all the layers. But instead of unfreezing all the layers I should have tried to unfreeze only bottom layers as in pretrained model's top layers are already great in extracting curves and edges features from the image
- Currently I am treating this as a single class problem. But if I treat this as multiclass problem, like left eye as one class and right eye as one so in total I will have 4 classes. So, in this way we will have multiclass segmentation problem. Maybe I should have started with multiclass instead of single class, that would have made more sense.
- In some scenario removing sigmoid function from last layer could help in getting good masks.

About the files uploaded:

Convert_Image.py file is used to generate npy file for images and mask.

model.py file have the code for the model.

Trainer.ipynb is a colab file used for training the model.

Inference.py file is used to generate output by using the trained model. Output results are obtained Just by providing path of the image.

UNetWithResnet34.rar is a the trained model, for using it kindly unzip the file first