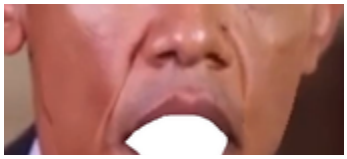
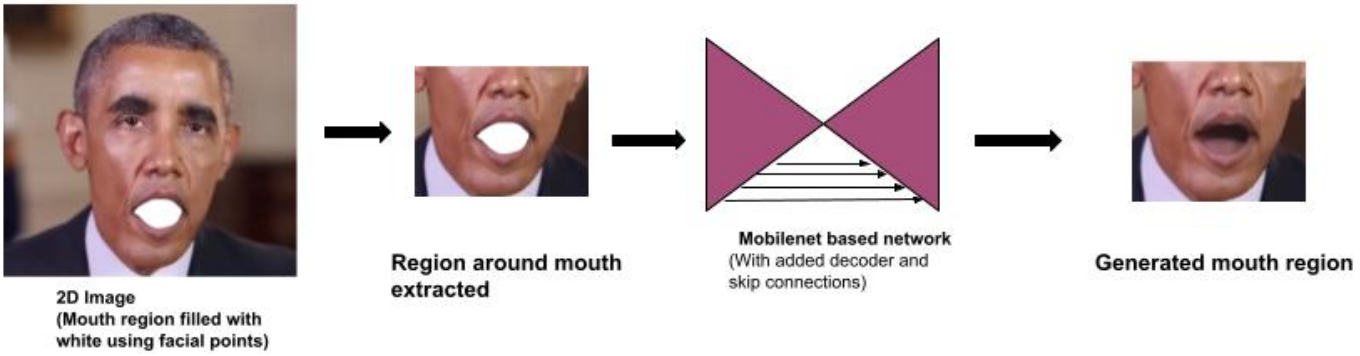
Mouth Interior Synthesis

In this section we will discuss about generation of interior region of mouth. For transferring of expression from source to target we need to transfer the mouth movement to the target image (or video).



After generating 2D image using the "*Face Parameter Extraction Model*" we crop out the mouth region from that image (we can do that because we have 50,000 3D facial points), after cropping the mouth area we fill area inside the mouth with white color (see image beside).

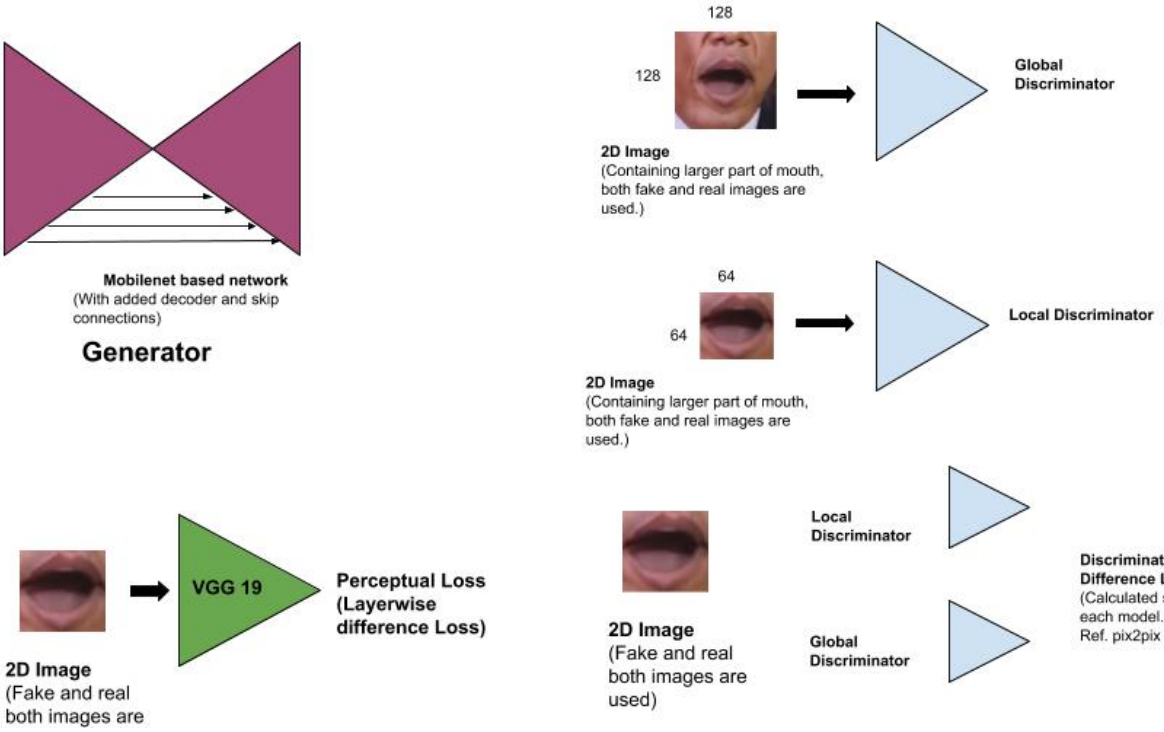
Overall, synthesis of mouth interior region looks like below



Mouth region is filled with white color, it is then cropped and fed to Mobilenet based efficient architecture to generate interior region of mouth which can be used on the target image (or video) for new expression.

Generator network is based on Mobilenet for faster real time processing (since the Xpression app will be used on mobile phones). We further added skip connections i.e. encoder and decoder network is connected through a "tunnel" (similar to what it is in U-Net).

We are using 2 discriminator models for training, we feed larger area of the mouth to one of the discriminator and we call that discriminator as "*glo bal discriminator*", on other discriminator we feed mouth region tightly bound in the mouth area (without any region nearby) and call that discriminator as "*local discriminator*".



Network is trained using the principle of "*Generative Adversarial Network*", we use L1 loss to compare pixel values between provided and generated image. We also use perceptual loss as better training practice, pretrained VGG19 model was used to compare layer values for calculation of perceptual loss. Also, for calculating L1 loss, we provide higher weight to the boundary of the mouth region. Similar to perceptual loss calculated with vgg19, we also calculate "*Discriminator LayerWise Difference Loss*" where we calculate difference of layers separately for local and global discriminator (refer to pix2pix HD paper for more information).

Detailed code can be found in https://github.com/embodyme/deep-image-inpainting “deep-image-inpainting” repository.

- Implementation of Mobilenet (Generator) can be found at “networks/mobilenet_generator.py”
- Implementation of Discriminator can be found at “networks/discriminator.py”
- Loss function implementation is available at “networks/losses.py”