3.1. Image Alignment:

3.2. GAN Inversion:

Original:

After 200 steps:

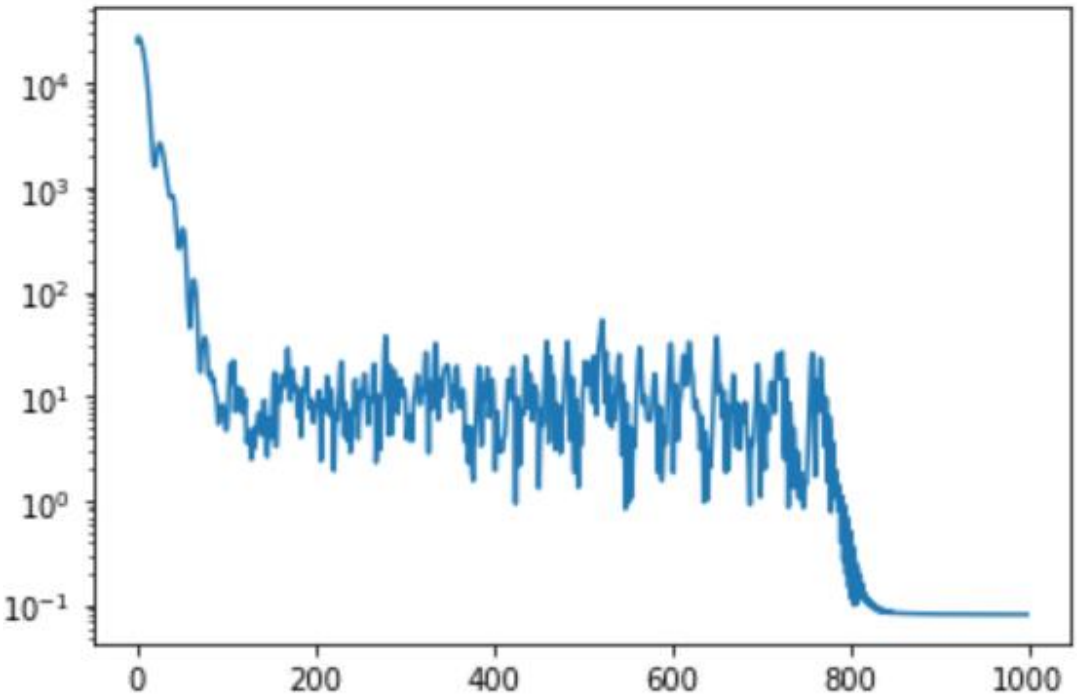
After 400 steps:

After 600 steps:

Final:

Image generated by initial z:

Plot loss:

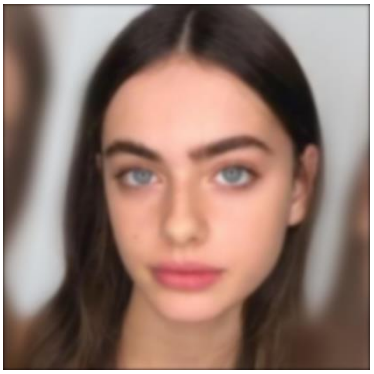


- The latent_dist_reg weight and num_steps affect the results in a generative model by controlling the regularization and number of optimization steps during training. A higher latent_dist_reg weight results in closer-to-mean latent representations, reducing overfitting. A larger num_steps value decreases loss by giving the model more optimization steps.

3.3.1 Image Deblurring:

Original:

Blurred:



Deblurred:

Original:

Deblurred:

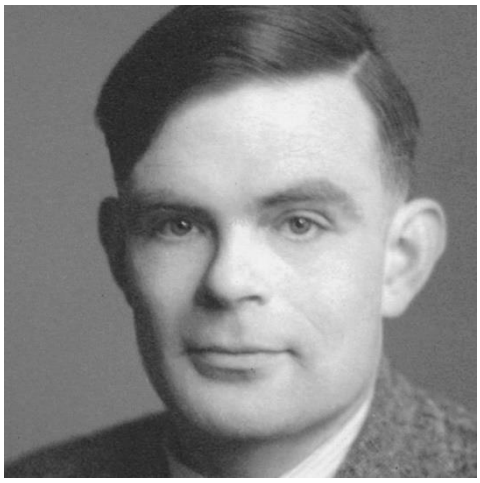
- My solution implements Gaussian blur for an image using PyTorch's conv2d function. It takes an image tensor and two hyperparameters (kernel_size and sigma) as inputs. The kernel_size sets the size of the kernel for the convolution operation and sigma determines the standard deviation for the Gaussian distribution used to generate the kernel. The kernel is expanded to 3 channels and repeated to match the image size before the convolution is performed with padding set to "same" and groups to 3 to preserve image shape and channels.
- I had some difficulties understanding PyTorch, especially with building the kernel and fixing bugs related to dimensions. I resolved these issues by seeking explanations online and discussing with friends about errors that I had.
- The kernel size has a direct effect on the amount of blurring applied to the image. A larger kernel size results in more smoothing and loss of fine details. The sigma parameter determines the standard deviation of the Gaussian distribution used to generate the kernel values, and affects the sharpness of the transition between blurred and unblurred areas.

3.3.2 Image Colorization:

Original: Grayscale: Colorized:



Original: Colorized:



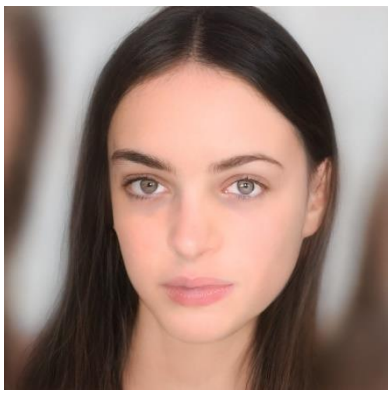

- The function takes a grayscale image tensor as input and calculates its mean along the 1st dimension to get a single channel representation of the image. This mean tensor is then expanded to 3 channels using the expand function to match the RGB representation of a color image. The mean tensor of the grayscale image serves as the input to the generator of the GAN network, allowing it to generate a colorized version of the image.

3.3.3 Image Inpainting:

Original:

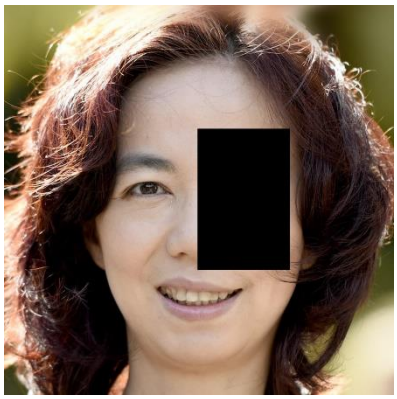
Masked:

Inpainted:



Original:

Masked:



Inpainted:

- My solution was to multiply the original image by a mask image that erases part of the image, creating a "hole" to be filled in by the inpainting algorithm, then repeat this step at each iteration. The motivation behind this approach is to preserve the original semantic information in the undamaged parts of the image while reconstructing the missing parts in a way that fits.