Image Processing - Exercise 5

Amir Kelman, amir.kelman, 208288811

Introduction

The goal of this exercise was to perform image reconstruction using StyleGAN2, a generative model trained on the FFHQ dataset, by inverting real images into the latent space and reconstructing them under various degradation modes. Specifically, the task involved image alignment (Section 3.1), reconstructing synthetic images (Section 3.2), and recovering degraded images under three modes: Gaussian blur (Section 3.3.1), grayscale conversion (Section 3.3.2), and inpainting (Section 3.3.3). The main techniques used were latent space optimization and perceptual loss minimization, leveraging concepts from generative modeling and optimization learned in class. I employed StyleGAN2's latent space to represent images as latent vectors, optimized these vectors using a perceptual loss (via VGG16), and applied degradation transformations to simulate real-world corruptions, aiming to reconstruct the original images.

Algorithm

The general algorithm for image reconstruction using StyleGAN2 consists of the following conceptual steps:

1. **Latent Space Initialization**: Start with an initial latent vector w in StyleGAN2's W space, the average w vector computed from a large sample of random latent codes. This provides a good starting point for optimization.
2. **Image Synthesis**: Use StyleGAN2's synthesis network to generate an image from the latent vector w, producing a synthetic image.
3. **Degradation Application**: Apply a degradation transformation (e.g., Gaussian blur, grayscale, inpainting, or none) to the synthesized image or target image, depending on the stage, to simulate corruption.
4. **Perceptual Loss Computation**: Extract features from both the target image (or its degraded version) and the synthesized image (or its degraded version) using a pretrained VGG16 network. Compute the perceptual loss as the mean squared error between these features.
5. **Regularization**: Add regularization terms to the loss, such as a latent distance regularization (to keep w close to the average w) and noise regularization (to smooth noise inputs in StyleGAN2).

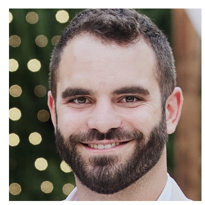
6. **Optimization**: Use an optimizer (e.g., Adam) to iteratively update the latent vector w and noise inputs to minimize the total loss, reconstructing the target image over several steps.
7. **Visualization and Saving**: Periodically save intermediate and final reconstructed images, along with the corresponding latent vectors, for evaluation.

This algorithm was adapted for each degradation mode by modifying the degradation step and ensuring proper normalization during loss computation and visualization.

Results

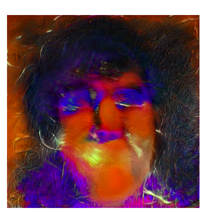
Results of 3.1

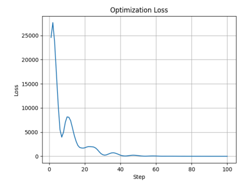
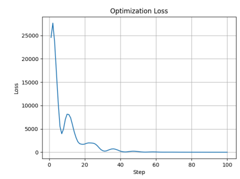
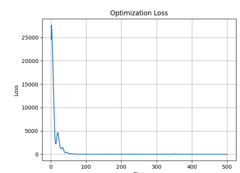
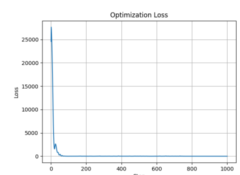
In Section 3.1, I processed real images to align so that the facial parts are in the same location in each image

Original Image :  Aligned Image: 

Results of 3.2

In Section 3.2, I inverted a synthetic image back into StyleGAN2's latent space to validate the reconstruction process. The hyperparameters num_steps and latent_dist_reg_weight significantly affected results: higher num_steps (500-1000) improved detail fidelity but risked overfitting at extreme values (2000+), while latent_dist_reg_weight balanced naturalness and uniqueness—high values (1.0) produced generic faces, low values (0.0001) yielded unique but sometimes inaccurate features. Optimal results came from num_steps=1000 and latent_dist_reg_weight=0.001, showing stable convergence and balanced detail retention.

Original image: **, Initial z:** 

Latent_dist_reg_weight & num_steps	**Generated image**	**Loss plot**
generated reg = 0.0001, steps = 100		
generated reg = 0.1, steps = 100		
generated reg = 0.1, steps = 500		
generated reg = 0.1, steps = 1000		

Results of 3.3.1

In Section 3.3.1, I applied Gaussian blur (GAUSSIAN_BLUR_DEGRADATION) to “Wedding_image_Amir_Aligned.jpg” and reconstructed it. I used a kernel size of 21 and sigma of 6.0, optimized for 1000 steps with a latent distance regularization weight of 0.001.

Original Image:	Original Degraded Image:	Generated Image:	Generated degraded Image:

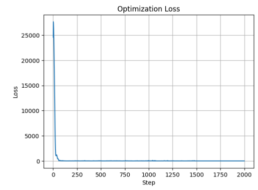
			
			

Solution and Motivation: I used StyleGAN2's latent optimization to deblur images, applying a Gaussian blur (kernel 21, sigma 6.0) and minimizing perceptual loss (VGG16). A strong blur was chosen because weak blurs (e.g., kernel 5, sigma 1.0) resulted in blurry outputs, indicating the original image was blurrier than the synthetic blur, necessitating a noticeable degradation for effective deblurring.

Issues and Solutions: first, it generated an image not similar to Luen, but the loss was acceptable- that was interesting. According to the data, I suspect this is the image closest to the average image. I fixed this by adding an apply_gaussian_blur_for_run_latent_optimization that returns the generated blurred image as a tensor that saves the gradients.

Bad example:

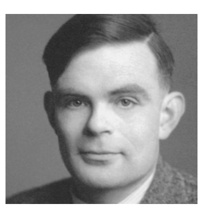
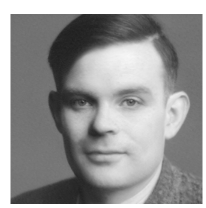
Generated image	Generated degraded image	this session loss plot

Analysis of Blur Kernel Size and Hyperparameters: A weak blur (kernel 5, sigma 1.0) had minimal effect, while a strong blur (kernel 21, sigma 6.0) enabled deblurring but risked over-smoothing.

Results of 3.3.2

In Section 3.3.2, I converted "Wedding_image_Amir_Aligned.jpg" to grayscale (GRAYSCALE_DEGRADATION) and reconstructed it, using the same hyperparameters as Section 3.3.1.

Original Image:	Original Degraded Image:	Generated Image:	Generated degraded Image:
			
			

Solution and Motivation: I used StyleGAN2's latent optimization, converting the target image to grayscale using luminance weights (0.299R + 0.587G + 0.114B) and optimizing the latent vector to minimize perceptual loss (VGG16) between the grayscale target and synthesized image. The motivation was to leverage StyleGAN2's generative capabilities to infer plausible colors based on the FFHQ dataset's human face distribution, ensuring realistic skin tones and features.

Issues and Solutions: A challenge arose where the colorized image initially had unnatural hues (green, blue and yellow). This was resolved by adjusting latent_dist_reg_weight from 0.001 to 0.5, which balanced the latent vector's proximity to the dataset's average, producing more natural colors
(attached- the bad example latent_dist_reg_weight == 0.001)

Results of 3.3.3

In Section 3.3.3, I applied inpainting (INPAINTING_DEGRADATION) to “Wedding_image_Amir_Aligned.jpg” by masking a central rectangular region (40% of the image dimensions) and reconstructed it. I also tested ”fei_fei_li_masked.png” with the same mask.

Solution and Motivation: I applied a binary mask (0/1) to erase central image areas, then optimized the latent vector to minimize VGG16-based perceptual loss between masked targets and synthesized images. This leveraged StyleGAN2's ability to infer missing regions that align naturally with unmasked areas.

Issues and Solutions: Minor normalization mismatches between StyleGAN2's [-1,1] range and standard [0,255] caused visual differences in plots but didn't affect the loss calculation and the optimization results.
Additionally, after latent_dist_reg_weight = 0.01 worked poorly, I applied 0.001, and it worked better

Original Image:	Original Degraded Image:	Generated Image:	Generated degraded Image:

Conclusion

This exercise demonstrated the power of StyleGAN2 for image reconstruction tasks, leveraging latent space optimization and perceptual loss to invert and recover degraded images. Key findings include:

- Gaussian blur recovery (Section 3.3.1) successfully restores some sharpness but struggles with fine details, suggesting a need for additional regularization or more steps.
- Grayscale recovery (Section 3.3.2) plausibly reconstructs colors, highlighting StyleGAN2's ability to infer missing information, though exact color matching remains challenging.
- Inpainting (Section 3.3.3) effectively reconstructs masked regions with surprisingly good results, creating coherent facial features that maintain semantic consistency with visible areas.

Overall, the techniques learned—latent space manipulation, perceptual loss, and degradation modeling—proved robust for image reconstruction, with potential for improvement in handling complex degradations.