Image Processing - Exercise 5

Areen Mansour, areen0507, 212900211

Introduction

The exercise focuses on utilizing StyleGAN2 for image reconstruction tasks, introducing concepts such as GAN inversion, latent optimization, and image priors. The primary goal is to reconstruct images using StyleGAN2 and explore its capabilities. By understanding GAN inversion, where the latent vector generating a given image is sought, and latent optimization, where the latent vector is optimized while keeping the generator weights fixed, we aim to achieve accurate reconstructions. Leveraging StyleGAN2's prior knowledge of the training domain, we treat it as an image prior, guiding the reconstruction process. Thus, the exercise familiarizes us with key techniques in image reconstruction using GANs.

Algorithm

The algorithm used for image reconstruction in this context involves several key steps:

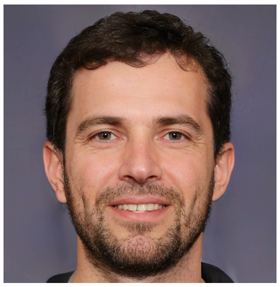
1. **Initialization**: The process starts with initializing the generator network of StyleGAN2, which has been pre-trained on a dataset of human faces. Additionally, the target image for reconstruction is loaded into memory.
2. **Latent Optimization**: The algorithm begins latent optimization by iteratively adjusting the latent vector of the generator while keeping the network weights fixed. This optimization process aims to find the optimal latent code that minimizes a predefined loss function.
3. **Loss Calculation**: During each iteration of latent optimization, the algorithm calculates a loss function that measures the similarity between the features of the generated image and the target image. This loss function incorporates perceptual features extracted from the images, such as those obtained from a pre-trained VGG16 network.
4. **Regularization**: To ensure stability and prevent overfitting, the algorithm applies regularization techniques during the optimization process. This may include noise regularization to encourage smoothness in the latent space and regularization on the distance between the optimized latent vector and a reference latent vector.
5. **Optimization Steps**: The optimization process iterates for a predefined number of steps, gradually adjusting the latent vector to minimize the loss function. Each optimization step involves updating the latent vector based on the gradient of the loss function with respect to the latent vector.
6. **Image Generation**: At each optimization step, the algorithm synthesizes an image from the optimized latent vector using the generator network. This generated image is then compared to the target image to compute the loss and guide further optimization.
7. **Convergence**: The optimization process continues until a stopping criterion is met, such as reaching a maximum number of iterations or achieving satisfactory convergence of the loss function.
8. **Output**: Once the optimization process is complete, the algorithm outputs the final reconstructed image along with the corresponding inverted latent vector that produced it.

Overall, the algorithm leverages latent optimization and perceptual loss functions to iteratively reconstruct an input image using a pre-trained GAN model. It effectively exploits the learned representations of natural images encoded in the GAN's latent space to guide the reconstruction process towards producing visually plausible results.

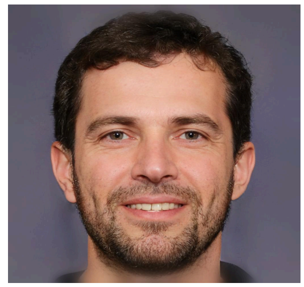
Results

Results of 3.1

-Example 1



Before



After

-Example 2

Before

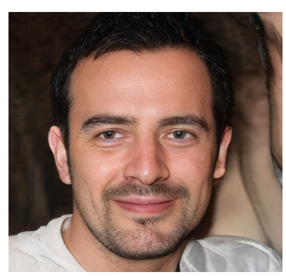
After

Results of 3.2

-Example:

Original Image

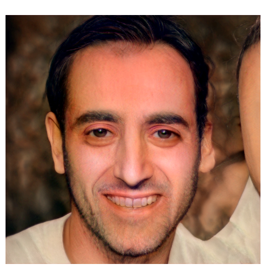
Final Optimized Image



The optimization process:

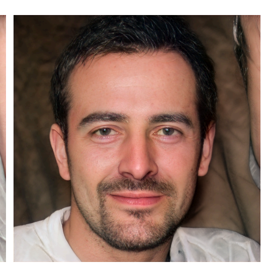
Intermediate 0

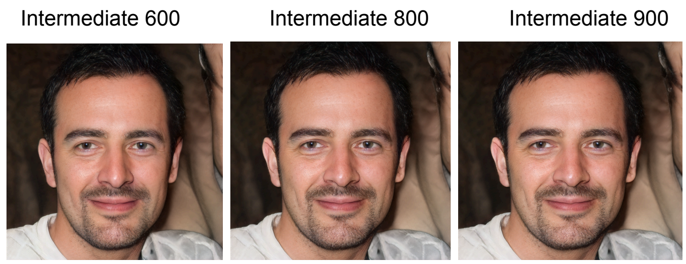
Intermediate 100



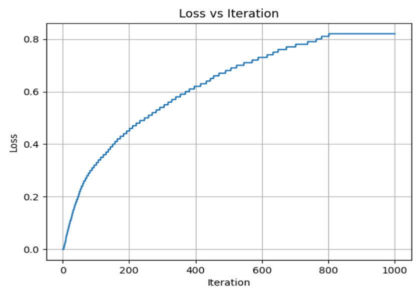
Intermediate 200

Intermediate 400

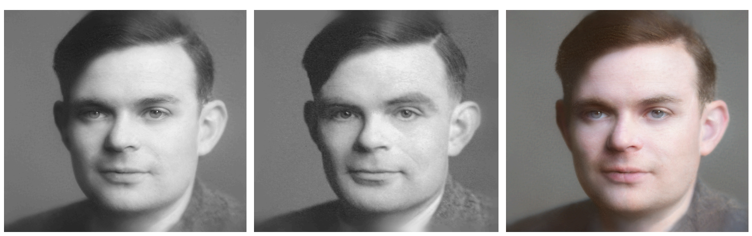




plot of the optimization loss:



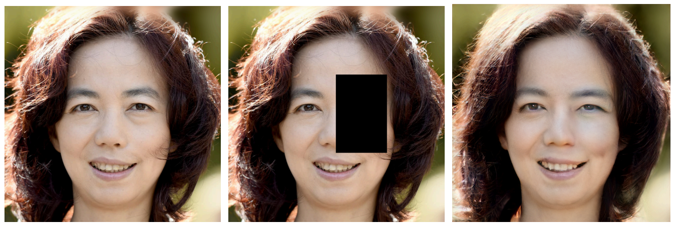
- Results of 3.3.2



• Discussing Solution for Grayscale Degradation and Latent Dist Regularization:

- Grayscale Degradation Adaptation:
    - Modified code to accommodate GRAYSCALE_DEGRADATION mode.
    - Ensured preservation of all three RGB channels while converting images to grayscale.
    - Achieved by averaging channels for grayscale representation and replicating it across RGB channels.
- Motivation for Adaptation:
    - Aimed to enhance the colorization process by retaining structural details from grayscale images.
    - Preservation of all three channels facilitates effective utilization of structural information for realistic colorization.
- Handling Latent Dist Regularization:
    - Faced challenges in selecting an optimal `latent_dist_reg_weight` parameter.
    - Struggled to balance between maintaining structure and introducing variation.
    - After experimentation, settled on a value of 1 to strike a balance between preserving the original essence and introducing realistic color variations.

-Results of 3.3.3



- Code Modifications for Inpainting:
    - Introduced adjustments enabling support for image inpainting, where missing areas are reconstructed.
    - Integrated functionality to utilize a mask image, defining the regions for inpainting.
    - Adapted the optimization process to incorporate the mask image, ensuring accurate inpainting.
- Motivation and Implementation:
    - Incorporated code to load the mask image and apply it during optimization for the inpainting degradation mode.
    - Converted the mask image into a tensor and adjusted its values to control selective modification of synthesized images.
    - This adaptation enables the optimization process to focus on reconstructing only the masked areas, leading to precise and semantically meaningful inpainted images.
- Encountered Issues and Solutions:
    - Determining the appropriate weight for latent distance regularization (latent_dist_reg_weight) posed a challenge.
    - Resolved by running optimization for an extended period (3000 steps) to ensure high-quality results.
    - Experimentation and result comparison helped settle on an optimal value of 0.4 for latent_dist_reg_weight.
    - Fine-tuning this parameter enhanced performance and produced visually appealing inpainted images.


if you have tried different hyper-parameters: num steps = 1000, latent sit reg weight = 1.(i got a completely different picture)

Conclusion

In this exercise, we delved into GAN inversion and latent optimization using StyleGAN2, aiming to reconstruct input images under various degradation scenarios. We treated the images generated by the GAN as belonging to the target domain, leveraging them for reconstruction tasks. By optimizing the latent vector while keeping the generator weights fixed, we successfully achieved realistic image reconstruction. Key elements included using perceptual features from a pre-trained VGG16 network, incorporating noise and latent distribution regularization, and simulating degradation modes like inpainting, grayscale, and Gaussian blur. Overall, this exercise showcased StyleGAN2's capability in image manipulation and restoration, offering insights into its practical applications.