Image2StyleGAN++

Euna Lee Luke Tao Kayla Ng Arthi Vijayakumar Johnry Zhao

https://github.com/KaylaNg1/CS4782-Final

1. Introduction

In this project, we aimed to reproduce results from *Image2StyleGAN++* by Rameen Abdal, Yipeng Qin, and Peter Wonka, which extends Image2StyleGAN. The core problem addressed in this paper is how to enable fine-grained and localized image editing in the latent space of StyleGAN. By solving this problem, one is able to make realistic, targeted local edits, such as altering facial features, without having to retrain the model. The three main contributions are optimizing noise separately from optimizing the latent space, using the global latent space (W^+) to make local edits with masks on images, and combining W^+ embedding with activation tensors. Together, these contributions enhance the reconstruction and editability of the existing Image2StyleGAN framework.

2. Chosen Result

For our project, we aimed to tackle all three contributions as stated above. For implementing contribution 1, we aimed to achieve a greater PSNR value post noise compared to only optimizing the latent space. We used the algorithms specified in Sec 6. and 7 of the paper to implement the foundation of these contributions. We chose to implement all three contributions to evaluate how they improve the reconstruction of images in comparison to Image2StyleGAN as well as enable realistic edits of images.

3. Methodology

Our first step in reimplementing the three contributions was to acquire the images the paper used. The input data to the Image2StyleGAN++ were images and images with masks (blocking out certain areas to be changed). We wanted to recreate the paper's results under the same image dataset, which resulted in us acquiring the images through screenshots from the paper(as were we unable to find copies of all the images).

Contribution 1 aimed to optimize the noise maps after optimizing W+ (the latent) to improve the PSNR scores of the original Image2StyleGan framework and generate a higher quality final image. To do this, we first passed the image through the W+ embedding until we reached a sufficient PSNR score above 20DB. This was necessary as without it, noise optimization would be ineffective at raising image quality as the larger structures of the image may still be out of place. We then freeze the W+ embedding to optimize noise maps separately to generate finer details such as hair and pores. Optimizing these separately allows us to get more realistic details without disturbing the overall structure of the image. The noise optimization was trained for 3000 steps and normalized every 20 steps to prevent overfitting and preserve the stochasticity of the noise.

Contribution 2 focused on running localized optimizations in masked and unmasked areas. There were multiple applications of this, and we aimed to recreate two: Local Edits with Scribble and Style Transfer. The goal for Local Edits with Scribble was for the scribble to become part of the features of the person (e.g drawing a red line would generate a scar). The goal for Style Transfer was for a reference style to be overlaid over the masked area, while preserving the original features of the image. To aid in this, we implemented masking by introducing a function that took in an image and masked off either black (style transfer) or red scribbles (local edits with scribble). We modified the loss functions (MSE, perceptual loss, style loss) to calculate loss with the masked area. Additionally, we used gaussian smoothing after extracting the mask to allow for better quality results. This allowed us to control which parts of the image would be edited or where style transfer would occur.

Masked W^+ optimization (W_l): This function optimizes $w \in W^+$, leaving n constant. We use the following parameters in the loss function (L) Eq. 1: $\lambda_s = 0$, $\lambda_{mse_1} = 10^{-5}$, $\lambda_{mse_2} = 0$, $\lambda_p = 10^{-5}$. We denote the function as:

$$W_l(M_p, M_m, w_m, w_{ini}, n_{ini}, x) = \arg\min_{w_m} \lambda_p L_{percept}(M_p, G(w,n), x) + \frac{\lambda_{mse_1}}{N} \|M_m \odot (G(w,n) - x)\|_2^2 \quad (2)$$

Masked Noise Optimization (Mk_n): This function optimizes $n \in N_s$, leaving w constant. The Noise space N_s has dimensions $\{\mathbb{R}^{4\times4}, \ldots, \mathbb{R}^{1024\times1024}\}$. In total there are 18 noise maps, two for each resolution. We set following parameters in the loss function (L) Eq. 1: $\lambda_s = 0$, $\lambda_{mse_1} = 10^{-5}$, $\lambda_{mse_2} = 10^{-5}$, $\lambda_p = 0$. We denote the function as:

$$Mk_n(M, w_{ini}, n_{ini}, x, y) = \arg\min_{n} \frac{\lambda_{mse_2}}{N} \|M_m \odot (G(w,n) - x)\|_2^2 + \frac{\lambda_{mse_1}}{N} \|(1 - M_m) \odot (G(w,n) - y)\|_2^2 \quad (3)$$

Local editing with scribbles required an input image as well as the same image but masked. We then extracted the mask before applying Image2StyleGAN++:the mask to the input image. We then ran masked W+ optimization and optimized the noise within the mask to produce the final image.

Algorithm 5: Local Edits using Scribble

Input: image $I_{scr} \in \mathbb{R}^{n \times m \times 3}$; masks M_{blur}
Output: the embedded code (w_{out}, n_{out})

$(w^*, n_{ini}) \leftarrow$ initialize();
$w_{out} = W_l(1, 1, w_m, w^*, n_{ini}, I_{scr}) + \lambda \|w^* - w_{out}\|_2$;
$n_{out} = Mk_n(M_{blur}, w_{out}, n_{ini}, I_{scr}, G(w_{out}))$;

To implement style transfer, we take an image, reference style, and a masked image. From these we first extract the mask and apply the mask to the input image. Then, we run masked W+ optimization. Next, we then ran masked style transfer optimization which was similar to W+ but optimized with reference to the style image. This required us to write our own style loss function as the paper did not provide details into theirs. We implemented the style loss function based on gram matrices and squared loss. Similarly to local edits with scribbles, we finished by running noise optimization to fine tune the final image.

Algorithm 6: Local Style Transfer

Input: images $I_1, I_2 \in \mathbb{R}^{n \times m \times 3}$; masks M_{blur}
Output: the embedded code (w_{out}, n_{out})

$(w^*, n_{ini}) \leftarrow$ initialize();
$w_{out} = W_l(M_{blur}, M_{blur}, 1, w^*, n_{ini}, I_1) + M_{st}(1 - M_{blur}, w^*, n_{ini}, I_2)$;
$n_{out} = Mk_n(M_{blur}, w_{out}, n_{ini}, I_1, G(w_{out}))$;

Contribution 3 involved combining activation tensors of two images. We focused on the spatial combination method which the authors of the paper identified as producing the best results. This meant directly copying the activations from the masked part of image one to the same part of image two. We also tried average copying which involved forming a linear combination of the image activation tensors and channel-wise copying where we created a new tensor by copying specific channels from each images' activation tensors. Similar to the paper, we found the best results when using spatial copying. However, the authors found the improved results when combining activation tensors at the fourth layer, but we found much better blending results at earlier layers. Since our results did not match the paper, we modified the original methodology by blending latent features. We took the weighted average of the W+ latent space of image one and the W+ latent space of image two.

Masked activation tensor operation (I_{att}): This function describes an activation tensor operation. Here, we represent the generator $G(w, n, t)$ as a function of W^+ space variable w, Noise space variable n, and input tensor t. The operation is represented by:

$$I_{att}(M_1, M_2, w, n_{ini}, l) = G(w, n, M_1 \odot (A_l^{I_1}) + (1 - M_2) \odot (B_l^{I_2})) \quad (5)$$

where $A_l^{I_1}$ and $B_l^{I_2}$ are the activations corresponding to images I_1 and I_2 at layer l, and M_1 and M_2 are the masks downsampled using nearest neighbour interpolation to match the $H_l \times W_l$ resolution of the activation tensors.

4. Results & Analysis

Contribution 1: In the end, our PSNR scores before noise optimization reached around 26DB while post optimization boosted to around 35-37 DB, successfully replicating noise optimization's improvement on image reconstruction. We did not have access to the images the paper used and therefore took screenshots to replicate the results. Therefore, the quality of the images we used were worse in comparison which impacted the reconstruction and PSNR scores of our output images. We hypothesize that with higher quality input images, we would have received a similar score to what they achieved in the paper.

Pre Noise Optimization

Post Noise Optimization

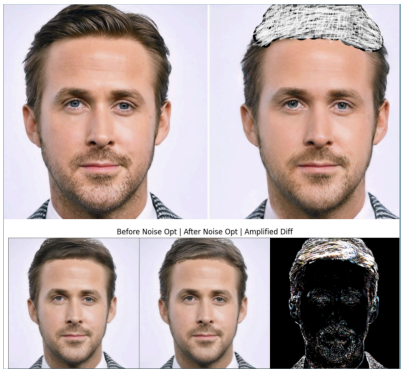
Contribution 2:

- Local Edits with Scribbles: Following the algorithm pseudocode specified in the paper, we were able to successfully implement this application to a similar degree. Although the paper doesn't

specify any specific quantitative results, we are able to visualize the scribble semantically embedding itself into the image, creating realistic edits. One challenge faced was ensuring that the entire scribble, no matter the color, was successfully masked. This was especially the case for when the scribbles were colored similarly to the skin. The paper did not specify how they masked the scribbles.

Paper Results:

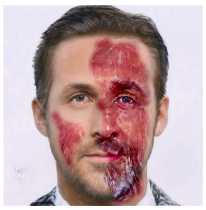
Our Results:



- Style Transfer: The paper was vague in their methodology such as their implementation of style loss which was necessary to match their performances.

Paper Results:

Our results:



Contribution 3: We are unsure if there were additional steps not specified in the paper to generate its results since replicating their methodology was not sufficient enough. However, the combination of spatial copying and feature blending resulted in a faithful mix of two images with details of each individual input source being identifiable which fulfilled the goal of this contribution.

5. Reflection

In summary, our re-implementation was successful, and we learned how Image2StyleGAN++ could be a very useful tool for image editing. We were able to reimplement all three contributions and get comparable results to the paper. There were differences present which can be attributed to specific implementation details in the paper, and quality of our images.

One potential direction highlighted in the paper is to extend this framework to edit images. Another improvement our team came up with is turning scribbles into semantic concepts (e.g. U shaped scribble is a smile). Additionally, we could extend this to other mediums such as 3D art and videos.

6. References

13.12. Neural Style Transfer — Dive into Deep Learning 0.15.0 documentation. (n.d.). D2l.ai. https://d2l.ai/chapter_computer-vision/neural-style.html.

Abdal, R., Qin, Y., & Wonka, P. (2020). Image2StyleGAN++: How to Edit the Embedded Images? *CVPR*, 8296-8305. https://openaccess.thecvf.com/content_CVPR_2020/papers/Abdal_Image2StyleGAN_How_to_Edit_the_Embedded_Images_CVPR_2020_paper.pdf.

Abdal, R., Qin, Y., & Wonka, P. (2020). Image2StyleGAN++: How to Edit the Embedded Images? -Supplementary Material-. *CVPR*. https://openaccess.thecvf.com/content_CVPR_2020/supplemental/Abdal_Image2StyleGAN_How_to_CVPR_2020_supplemental.pdf

Bhat, Z. (2021). Image2StyleGAN [Pretrained Model]. https://github.com/zaidbhat1234/Image2StyleGAN.

Mohd, A. (2021, February 8). *Neural style transfer using PyTorch*. DEV Community. https://dev.to/aquibpy/neural-style-transfer-using-pytorch-3d5l.