Image Inpainting with Masked Self Attention and Semantic Loss Function

Hamza Pehlivan, Hakan Sivuk

***Abstract*—Color inconsistencies are still a problem in image inpainting tasks. To deal with them, we present an attention mechanism named Masked Self-Attention, which takes an alternative approach to deal with masked inputs. In addition to that, we adopt contextual attention idea for our attention module. We also propose a new loss function called Semantic Attention Loss, which is used to establish connection between semantic labels and attention scores. The source code is available at: https://github.com/hamzapehlivan/cs559_project.**

I. Introduction

Image inpainting, the task of filling missing pixels, is a non trivial task that requires both global and local consistency. It is already known that straightforward CNN based architectures lead to color inconsistencies between painted patches and rest of the image. Different mechanisms are proposed to handle this artifact like attention maps [13], inference time optimization [10] and image blending after training [4]. In this work, we also focus on attention mechanisms by introducing two ideas:

1) We developed a new attention mechanism called Masked Self-Attention in which we dynamically adjust attention scores based on input masks.
2) We adopted the contextual attention idea for our attention module.
3) We developed Semantic Attention Loss, which is used to crate a connection between semantic labels and attention maps.

II. Related Work

The most recent works for inpainting benefits from deep neural networks, in which invalid pixels are initialized with constant placeholder values. Among those, [4] uses a generative model with dilated convolutions, which is known to increase the receptive field for the masked areas. It also uses two discriminators, one for global and the other for local consistency. Some other methods utilize attention mechanism [13] to refine the blurry results that convolution neural nets produces. As these attention mechanisms in image inpainting networks can be contextual like [11], it can be also utilized as self-attention. Self-Attention module which is defined in [14] is used as a basis for our attention module. Our Masked Self-Attention module is different than the Self-Attention because of the nature of the masked images and it will be explained in detail in the next sections. Also, contextual attention idea is adapted from [11] and applied on the attention mechanism in our project.

To enable users to interact with the end result of deep learning algorithms, semantic maps [3], sketches [12] and color maps [5] were used. Motivated by these studies, we also use semantic maps to guide our generation process because it provides pixel level control for the users. A similar work to ours is SESAME [8], which has a SPADE [9] like generator and a two branch discriminator.

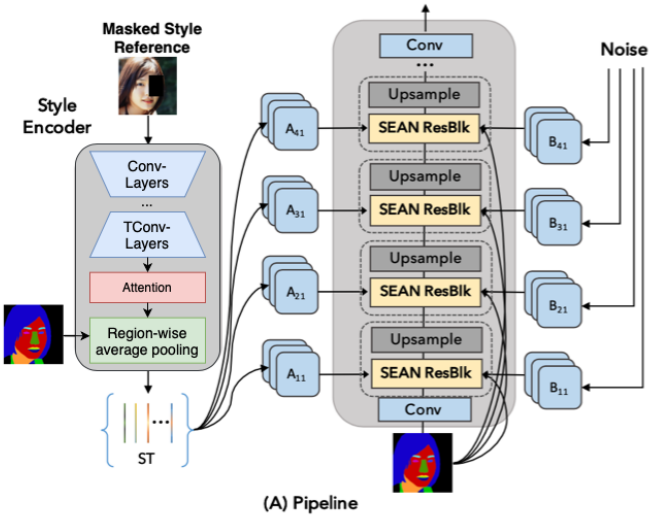

Fig. 1. Slightly modified SEAN network. Attention mechanism is deployed after transposed convolutions in the style encoder. We feed a masked style reference and a segmentation map to the style encoder and it outputs a style matrix. This style matrix is fed to the several layers of generator with segmentation map and random noises.

III. Method

Network. We used [16] as a base architecture for this project. We basically used the same architecture as it is proposed in their paper. The differences are can be seen from Fig. 1, we feed a masked image to the style encoder as the style image and deploy an attention mechanism after transposed convolution layers. In that way, we want to generate a style matrix based on the given segmentation layout and masked style reference. Furthermore, implemented attention mechanism tries to capture global context better and to generate more realistic output images.

A. Attention

In this section, different attention mechanisms are investigated to leverage relationship between pixels to complete the masked area are explained. First of all, the self attention mechanism which is proposed for GANs by [14] will be explained briefly. After that, our proposed attention modules

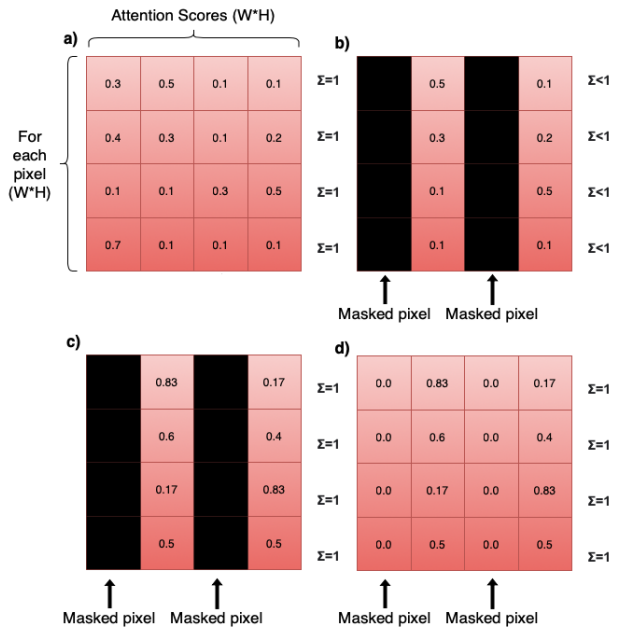

Fig. 2. Modifications on the attention map for masked self-attention mechanism. (a) Original attention map version used for self-attention mechanism. (b) Attention scores of masked pixels are set to 0. (c) Remaining attention scores are rescaled so that sum of each row is equal to 1. (d) Final attention map version used for masked self-attention mechanism.

which use self attention as a basis will be explained. Self attention idea was modified so that it is more suitable for image inpainting task.

Self-Attention This attention mechanism proposed by [14] is used a basis for our novel attention mechanisms. To make it feasible, 3 1x1 convolution operations are applied on the feature map to form query, key and value matrices. To get the attention map, we perform matrix multiplication on query and key and softmax operation on the result. Finally, the attention map is multiplied with the value matrix and the resultant self-attention feature maps are added to the initial feature maps.

Masked Self-Attention To make self-attention mechanism more suitable for image inpainting task, some modifications are needed. The main problem with the original method is that when we form the self-attention feature maps, pixels of the masked area is also used. Masked pixels are black at the beginning and have no information about the original image. Therefore, it can cause a performance degradation. To prevent this performance degradation, while we are forming the self-attention feature maps, we can ignore the masked pixels. To ignore these pixels, we can follow the process as it is shown in Fig. 2. What we have done is basically, setting masked pixels' columns 0 in the attention map matrix, and re-scaling the remaining numbers so that sum of each row is equal to 1 and expected value is preserved for the attention feature maps. By doing that, we get attention scores for only valid pixels. Therefore, when we construct the self-attention map, for each pixel in the self-attention map, only valid pixels are used.

Contextual-Attention We can go a step further with the masked self-attention idea and only get attention feature maps

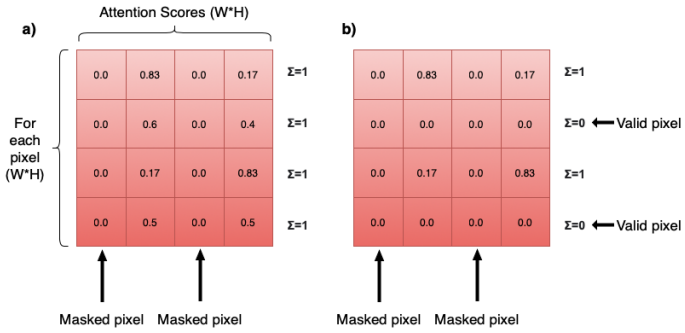

Fig. 3. Modifications on the attention map for contextual attention mechanism. (a) Attention map which is used for masked self-attention mechanism. (b) Final attention map which is used for contextual attention mechanism where attention vectors of valid pixels are all set to 0.

for masked pixels. Instead of getting a attention feature maps for all pixels, we can form it only for masked pixels by using valid pixels from value matrix. To do so, we can basically set valid pixels' rows in the attention map to 0, as it is shown in Fig. 3

B. Loss Functions

Following [16], we used VGG and GAN losses. In addition to them, we utilized pixel loss to increase supervision and proposed a new loss function called Semantic Attention Loss.

VGG loss: Let Φ be the VGG network, I be real image, M be mask, G be generator, D be discriminator, Sem be semantic map. There are N layers that we calculate loss from and they have K dimensions. The loss is calculated as:

$$\mathcal{L}_{vgg} = \mathbb{E}(\sum_{i=1}^{N} \frac{1}{K_i} \|\Phi_i(I * M) - \Phi_i(G(I, Sem, M) * M)\|_1) \tag{1}$$

Pixel Loss. We calculated pixel loss in unmasked regions as follows:

$$\mathcal{L}_{pixel} = \mathbb{E}(\|(I * M) - (G(I, Sem, M) * M))\|_1) \tag{2}$$

Adversarial loss: The adversarial loss is calculated with real and fake images as the following:

$$\mathcal{L}_{adv} = \mathbb{E}(\log D(I) + \log(1 - D(G(I, Sem, M) * M))) \tag{3}$$

Semantic Attention Loss. The main idea of this loss function is the network should adjust attention scores based on the semantic label of the pixels. For instance, if the pixel of interest is an eye, we want our model to pay attention other eye pixels, rather than background. To encourage such a behaviour, we defined a hyper-parameter μ, which pixels with same semantics would want to achieve. We denote the number of pixels that have the same semantic label with the point of interest as C. Attention scores of the pixels are found with the function att. Lastly, we denote the semantic set that the point of interest is a member of as A. There are K point of

interests, and we calculate loss function for each of them as the following:

$$\mathcal{L}_{sal} = (\sum_{i \in A} att(i)) - \mu * C \tag{4}$$

After calculating Equation 5 for each point of interest, we find their mean value, which gives the final semantic attention loss.

$$\mathcal{L}_{sal\_avg} = \frac{1}{K}(\sum_{j} \mathcal{L}_{sal}[j]) \tag{5}$$

We also showed in our implementation that this loss function can be found with efficient vectorized calculations.

With the introduction of the semantic attention loss, the objective of the generator becomes:

$$\min_{G}(\lambda_1 * \mathcal{L}_{vgg} + \lambda_2 * \mathcal{L}_{pixel} + \lambda_3 * \mathcal{L}_{sal\_avg} + \mathcal{L}_{adv}) \tag{6}$$

The objective of the discriminator is:

$$\max_{D} \mathcal{L}_{adv} \tag{7}$$

C. Optimization at Test Time

Another method that we can use is to optimize the style vectors at test time using only unmasked pixels. This method is highly utilized in GAN inversions [1]. First, we create a style vector from the input image and it is fed to generator to obtain an output. We compare the original image with the output using pixel and VGG losses and update the style vectors. The optimization objective is:

$$\min_{style\_vectors} (\lambda_1 * \mathcal{L}_{vgg} + \lambda_2 * \mathcal{L}_{pixel}) \tag{8}$$

Training Details. We use Adam optimizer [2] with $\beta_1 = 0.5$ and $\beta_1 = 0.999$. Learning rate is 0.0002 for every network. Total number of epochs is 40. We set λ_1 = 0.1, λ_2 = 0.5, λ_3 = 0.1 and $\mu = 5 * 10^{-4}$ in the loss function. Before we feed image and mask to the encoder, we multiply them and then we concatenate them channel-wise. After we received the fake image, we modify it as following before we feed it to the discriminator.

$$FinalImage = I * M + F * (1 - M) \tag{9}$$

where M is mask, F is generator output and I is the original image. This is done because we do not want our generator to change valid pixels.

For each model, training takes 3 days. We have trained our models with 4 GPUs (GeForce GTX 1080Ti).

For the test time optimization, we use Adam optimizer with β_1 = 0.9 and β_2 = 0.999. Learning rate is set to 0.01. We optimize each image for 20 iterations. We set $\lambda_1 = 5 * 10^{-5}$ and $\lambda_2 = 5{*}10^{-3}$ in the loss functions of iterative optimization.

Dataset. We used CelebAMask-HQ dataset [6] for our experiments. It contains 19 instance labels (e.g. left eye, right eye, hair...). It has 28000 images for training and 2000 images for validation. We mask the input images with randomly generated free form masks using the algorithm described in ComodGAN [15]. We use 20% as the maximum mask ratio.

Mechanism	PSNR
Without Attention	29.42
Self-Attention	30.26
Masked Self-Attention	30.41
Contextual Attention + Attention Loss	30.49

TABLE I
PSNR SCORES FOR MODELS TRAINED WITH THE SPECIFIED ATTENTION MECHANISM.

	SSIM	PSNR
Without Optimization	0.950	30.71
With Optimization	0.947	30.49

TABLE II
COMPARISON BETWEEN USING OPTIMIZATION AT TEST TIME AND NOT USING.

IV. RESULTS

To evaluate our model, we use PSNR and SSIM as evaluation metrics, besides visual results.

A. Ablation Study on Attention Modules

As it can be seen from Table I, different attention modules are compared based on PSNR metric. Using attention modules improved the result for all attention mechanisms. The best result is observed when the contextual attention module and attention loss are used together. Result for using contextual attention without attention loss is not reported because in that case the training was unstable.

B. Visualization of Attention Maps

The aim of the proposed contextual attention and attention loss is basically completing masked region based on the related valid pixels. Therefore, visualizing attention maps can give useful insights about whether the this aim is achieved. As it can be seen from Fig. 4; when we construct a masked pixel in attention feature maps, it gives more attention to the valid pixels of the same semantic region. When the masked pixel is from hair, it gives the most attention to other valid hair pixels. Similarly, when the masked pixel is from the face, it gives most attention to other valid face pixels. As a result, using contextual attention with attention loss can be beneficial for completing a missing style pixel from the same semantic region.

C. Impact of Optimization

We observed that use of iterative updates increase the evaluation scores. That is why iterative optimization pushes the style vectors to a point in which original image can be constructed more realistically (Table II).

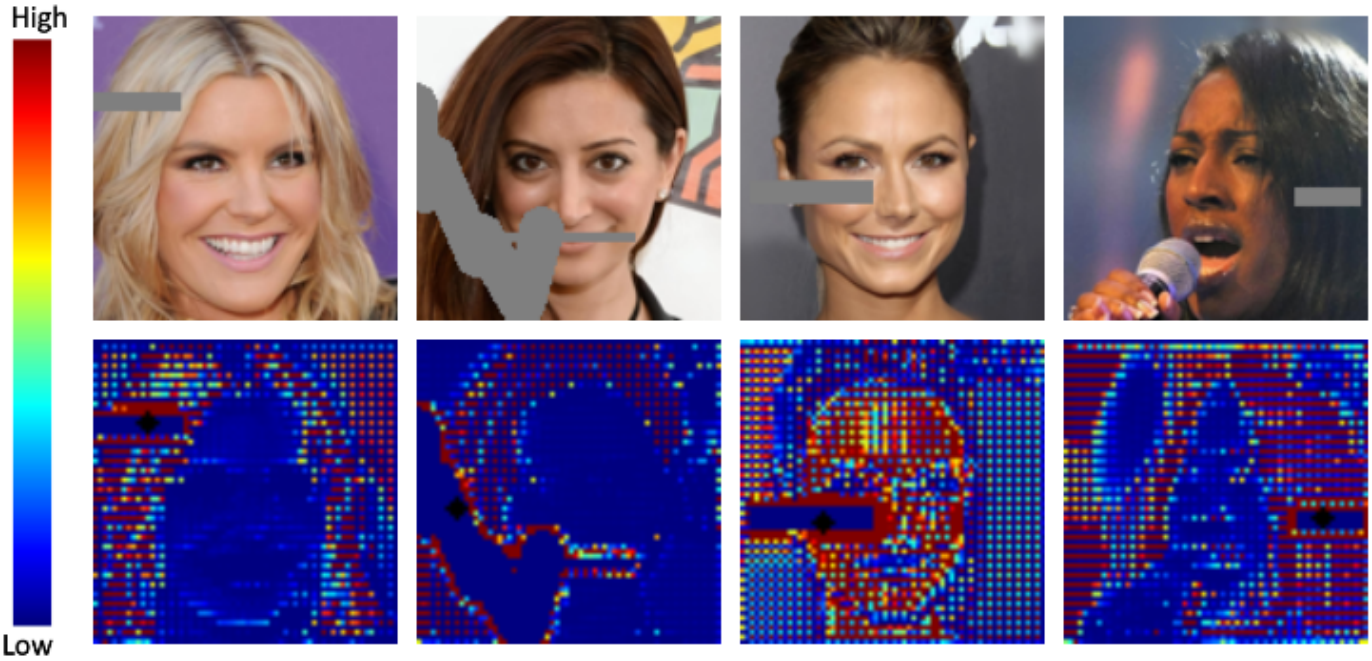

Fig. 4. Visualization of attention maps. Black diamonds denote points of interest and the heatmap denotes the corresponding attention map for that pixel. Red color denotes high attention score, while blue color denotes low attention score. First row shows masked inputs and second row shows their attention maps.

Project	SSIM	PSNR
RFR-Net [7]	0.981	33.56
Our	0.950	30.71

TABLE III
COMPARISON WITH THE STATE OF THE ART MODEL BASED ON SSIM AND PSNR METRICS.

D. Comparison with State of the Art Results

When we compare our model with the state of the art model [7] based on PSNR and SSIM metrics, we found that we are still behind. Further investigation should be conducted to make our network suitable for image inpainting.

E. Visual Results

As it can be seen from Fig. 5, our model is able to generate realistic images based on the given semantic layouts. However, there are also some failure cases where obvious artifacts can be seen in the generated image. For instance, in the third row, when we mask one of the eyes, the masked area is completed in a way that it is not consistent with the other eye. Also, in the forth row, since the mask is close to blurry regions, it is completed with an inconsistent texture which makes the output unrealistic. Therefore, these results show that although our model is capable of generating realistic images, there are also some limitations which should be improved in future works.

V. CONCLUSION

To sum up, we have selected [16] as basis architecture for the project. We added novel attention module called Masked Self-Attention. We tried to adopt contextual attention idea for our attention module. Also, a novel semantic attention loss is proposed as an accompany to contextual attention module. With these different attention mechanisms and attention loss, several experiments have been conducted and it has been observed that using contextual attention and attention loss together yielded the best result. Also, iterative updates during the test time improved the results. At the end of the project, we couldn't achieve the current state of the art results shown in Table III. Our model can generate realistic images, however, some of the generated images include artifacts.

REFERENCES

[1] Rameen Abdal, Yipeng Qin, and Peter Wonka. Image2stylegan: How to embed images into the stylegan latent space? *CoRR*, abs/1904.03189, 2019.
[2] Jimmy Ba Diederik P. Kingma. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2015.
[3] Seunghoon Hong, Xinchen Yan, Thomas E Huang, and Honglak Lee. Learning hierarchical semantic image manipulation through structured representations. In *Advances in Neural Information Processing Systems*, pages 2713–2723, 2018.
[4] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, 36:1–14, 07 2017.
[5] Youngjoo Jo and Jongyoul Park. SC-FEGAN: face editing generative adversarial network with user's sketch and color. *CoRR*, abs/1902.06838, 2019.
[6] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
[7] Jingyuan Li, Ning Wang, Lefei Zhang, Bo Du, and Dacheng Tao. Recurrent feature reasoning for image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.
[8] Evangelos Ntavelis, Andrés Romero, Iason Kastanis, Luc Van Gool, and Radu Timofte. SESAME: Semantic Editing of Scenes by Adding, Manipulating or Erasing Objects. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 394–411, Cham, 2020. Springer International Publishing.

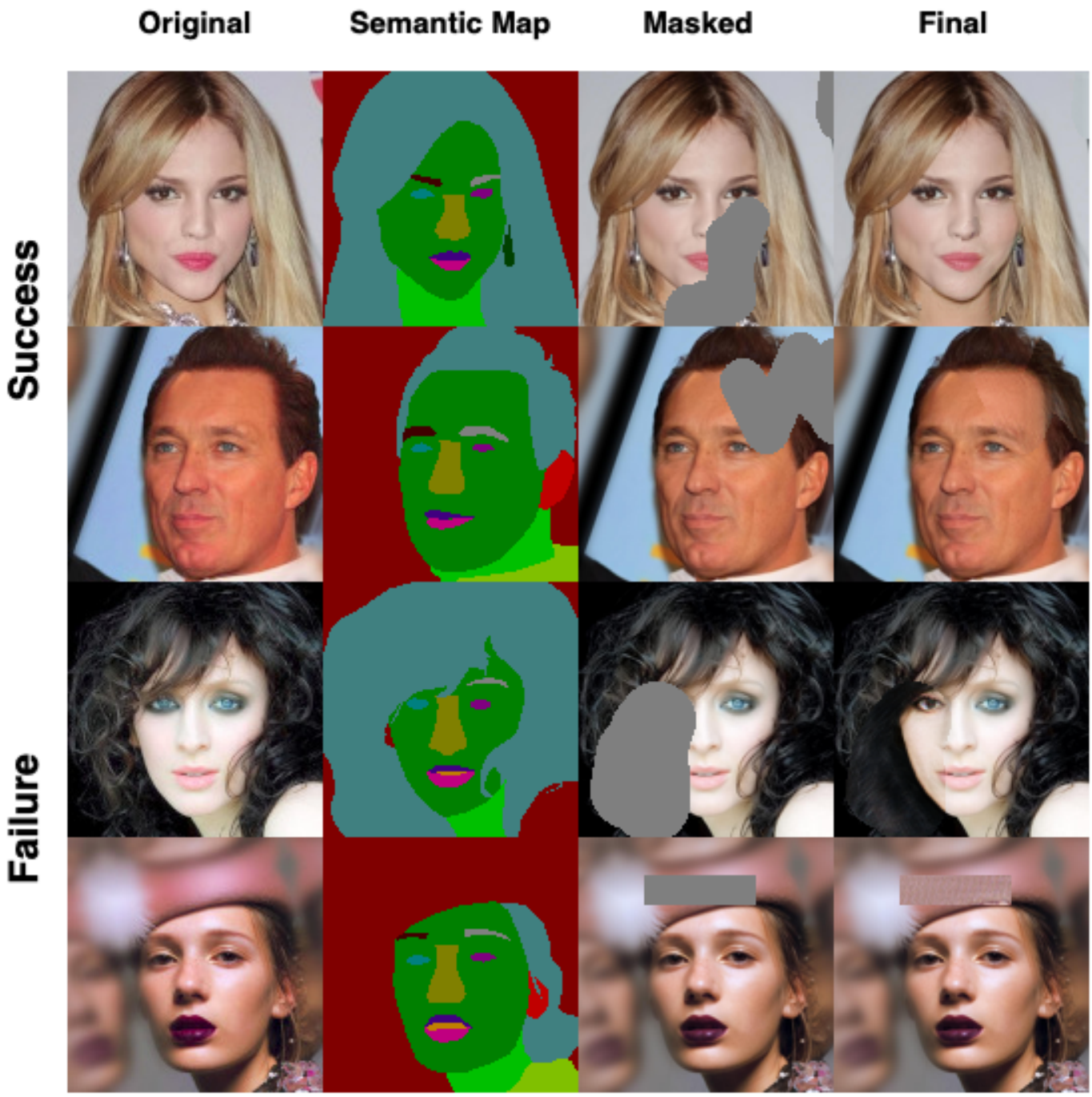

Fig. 5. Some success and failure cases for the generated images. Images are shown in the following column order: original image, semantic map, masked image, final image.

[9] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[10] Tengfei Wang, Hao Ouyang, and Qifeng Chen. Image inpainting with external-internal learning and monochromic bottleneck. *CoRR*, abs/2104.09068, 2021.

[11] Zili Yi, Qiang Tang, Shekoofeh Azizi, Daesik Jang, and Zhan Xu. Contextual residual aggregation for ultra high-resolution image inpainting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7508–7517, 2020.

[12] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. *arXiv preprint arXiv:1806.03589*, 2018.

[13] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.

[14] Han Zhang, Ian Goodfellow, Dimitris Metaxas, and Augustus Odena. Self-attention generative adversarial networks. In *International conference on machine learning*, pages 7354–7363. PMLR, 2019.

[15] Shengyu Zhao, Jonathan Cui, Yilun Sheng, Yue Dong, Xiao Liang, Eric I Chang, and Yan Xu. Large scale image completion via co-modulated generative adversarial networks. In *International Conference on Learning Representations (ICLR)*, 2021.

[16] Peihao Zhu, Rameen Abdal, Yipeng Qin, and Peter Wonka. Sean: Image synthesis with semantic region-adaptive normalization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.