COMP.SGN.300 Advanced Image Processing Project Report

Article Paper: Free-Form Image Inpainting with Gated Convolution [1]

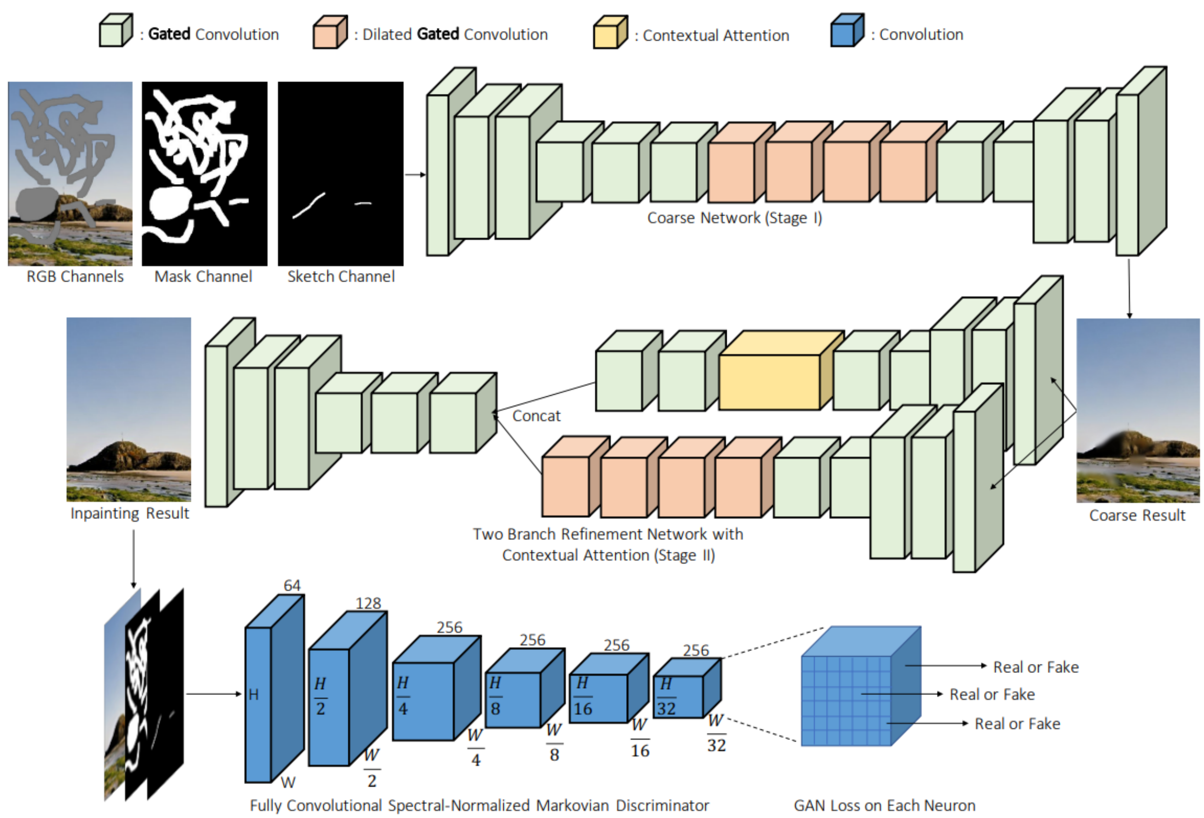
Code modification: https://github.com/nntk072/generative_painting.git

1. Introduction

Image inpainting is a technique used in image processing to fill in missing or corrupted parts of images. This report presents an evaluation of a Convolutional Neural Network (CNN) model for free-form image inpainting, which applies the concept of the Gated Convolutional Layer for the model implementation. There are two image quality metrics: the Structural Similarity Index (SSIM) and the Peak Signal-to-Noise Ratio (PSNR). Moreover, this report will also tell the problems and limitations of this method for the improvement of the model architecture.

2. Model Training and Testing

Moreover, the model architecture will be represented as follows [1]:



Moreover, the training parameters can be demonstrated as follows [1]:

Parameters	Value
Adversarial loss	WGAN_GP, with λ = 0,0005
WGAN_GP Gradient Penalty Loss	λ = 10
Random Crop	True, 10
Padding	Same

Iteration	1000000
Batch Size	16
Image Shapes	[256, 256, 3]
Box Shaping (Randomly distributed) (Height, Width)	[128±32, 128±32]
Autoencoder Loss	True, the multiplication of L1_LOSS × $\|\|$Batch Relation$\|\|_1$

Due to the computational resource limitations of the CNN model, the training procedure could not be processed. Therefore, a pre-trained model from the author was chosen. The testing procedure was conducted on a GPU Tesla V100 32GB/16GB, with 8 CPU cores, and 64 GB RAM per CPU. Each image test was processed within 2.7 seconds. In the testing, there are 3 main purposes: Generating the output from the model, evaluating the model by using PSNR and SSIM evaluation metrics.

3. Dataset

5000 ImageNet images were selected as the dataset for testing the project. The original image from the dataset went through a mask to produce a necessary inpainted image.

There will be two masks for the evaluation: a random Line/Circle/Curve combination mask and a box mask.

4. Results

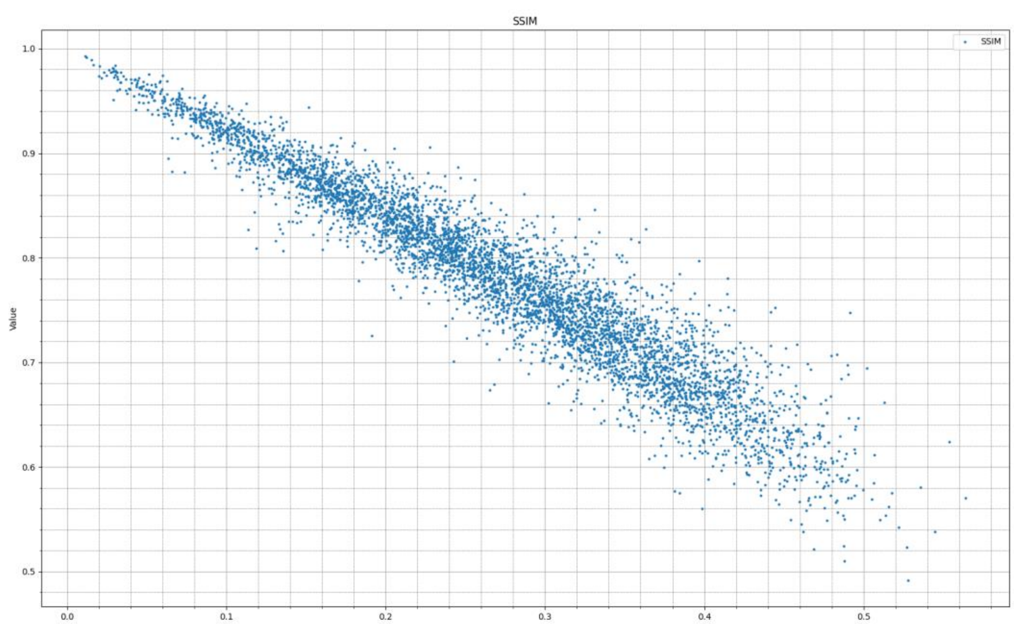


Figure 1: SSIM distribution of Line/Curve/Circle combination mask

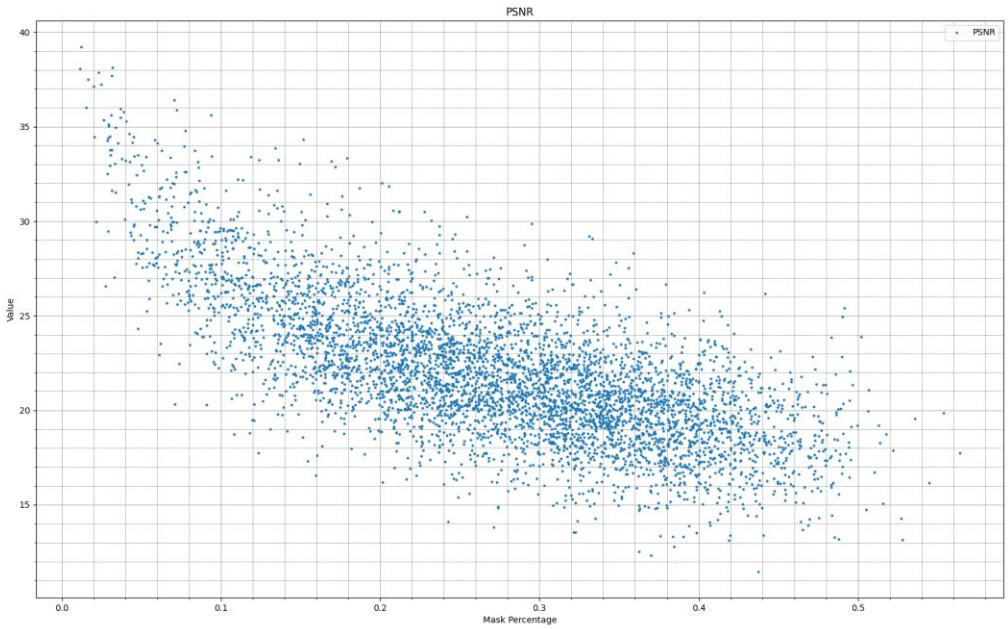


Figure 2: PSNR distribution of Line/Curve/Circle combination mask

Figure 3: Randomly generated output from Line/Curve/Circle combination mask

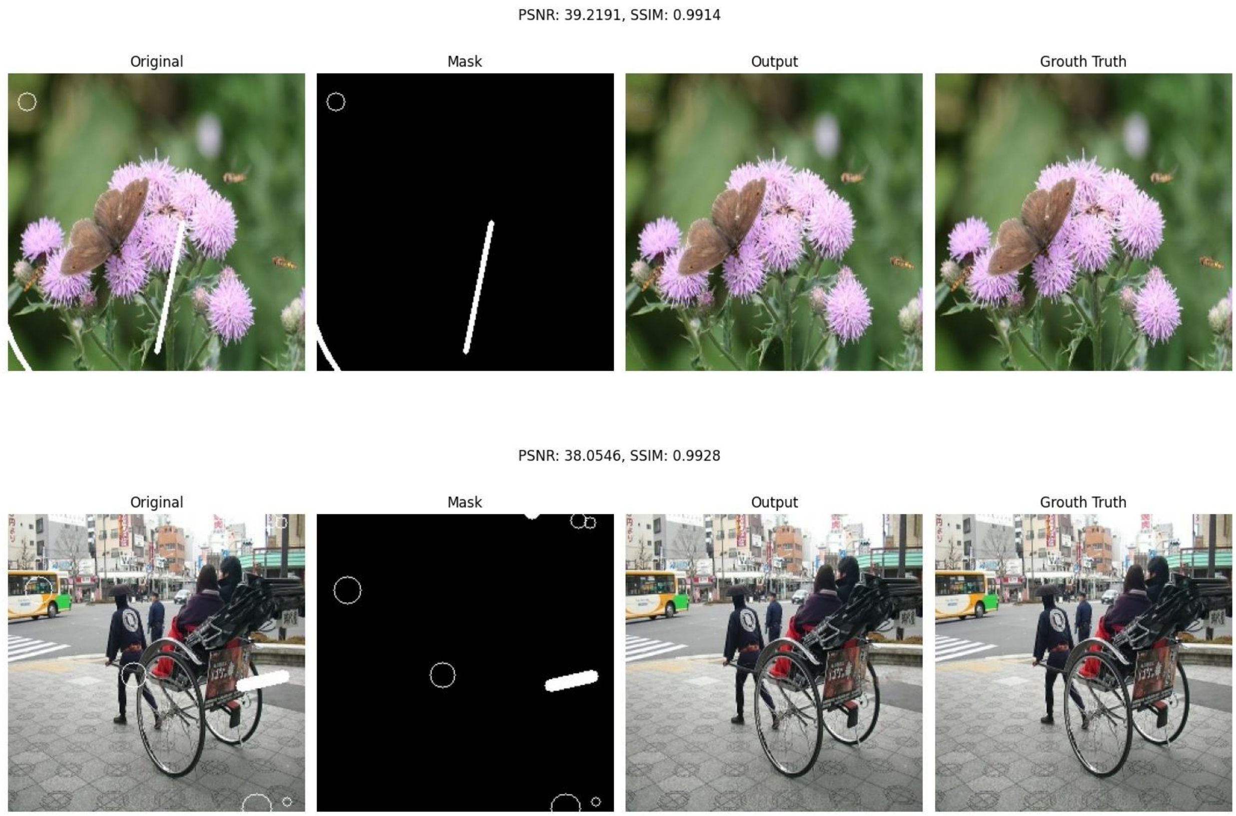


Figure 4: Best PSNR and SSIM generated output from Line/Curve/Circle combination mask

Discussing the results of the model with the Line/Curve/Circle combination mask, we can see that the model has the distribution of PSNR and SSIM following the linear trendline for the SSIM and exponential trendline for the PSNR. This can show the stability of the model when generating the generalization of the images, although we know that training a CNN model is an abstract idea and the approach of multiple combinations of the image processing and convolutional can lead to fluctuation and instability. Moreover, according to the output images, observed in image generation from the dataset, almost all the characteristics of the object seem to have remained if the specific masking area is not too large in a specific region.

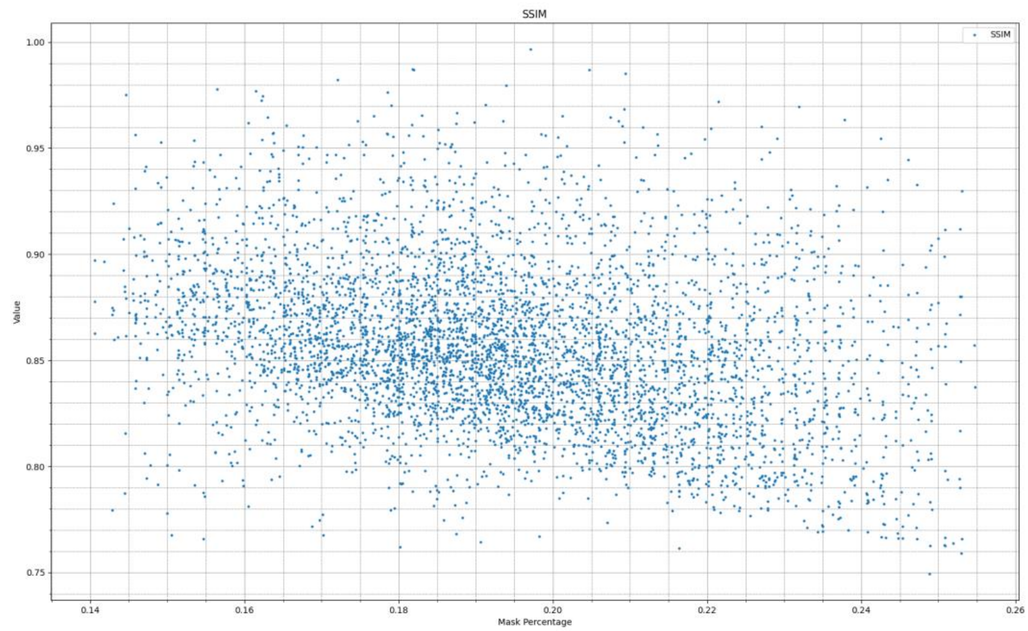


Figure 5: SSIM distribution of box mask

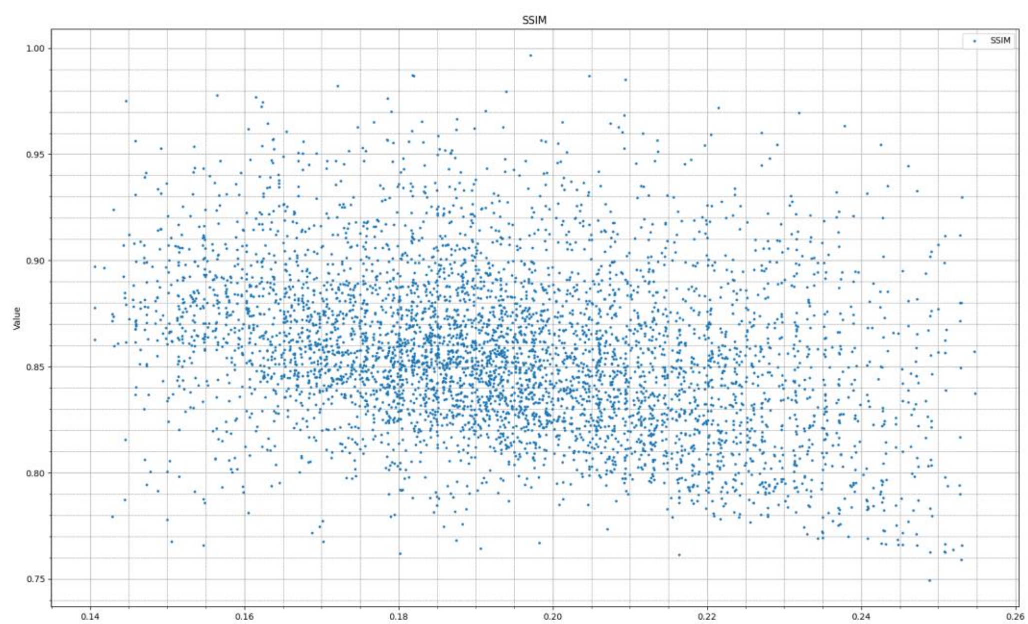


Figure 6: PSNR distribution of box mask

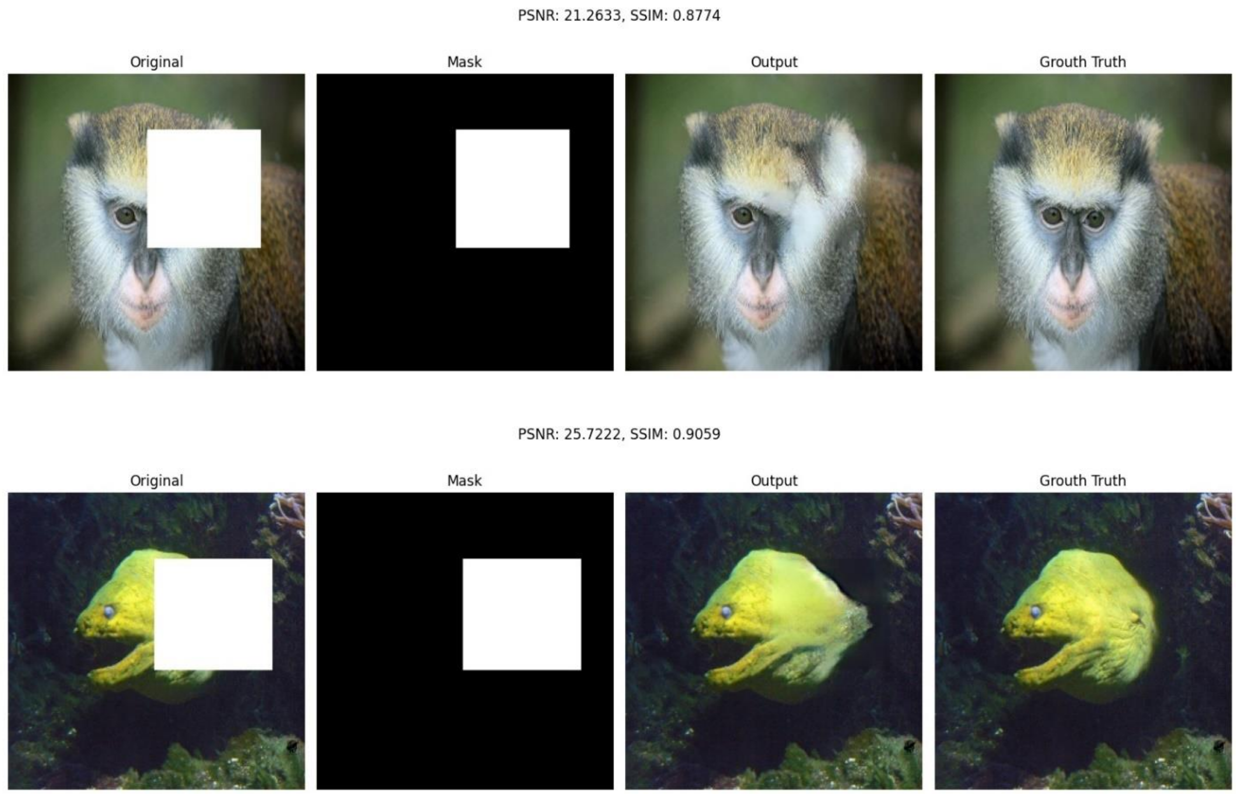


Figure 7: Randomly generated output from box mask

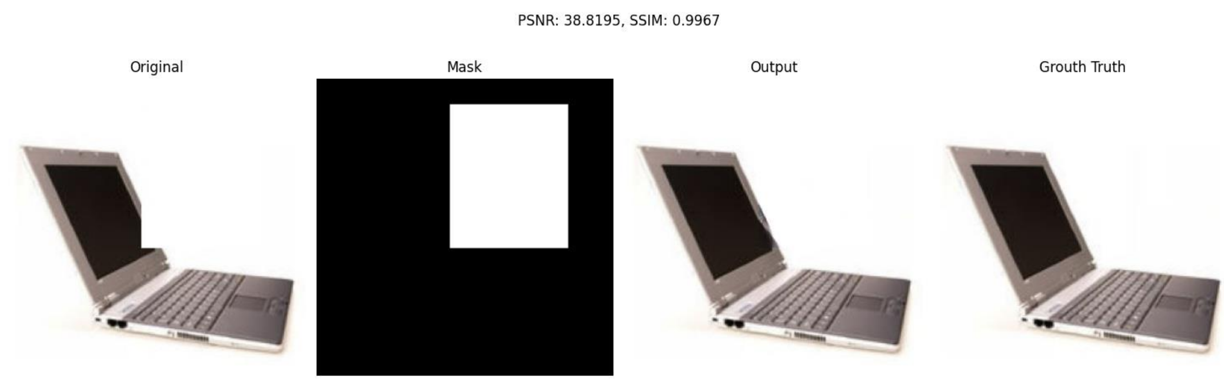


Figure 7: Best PSNR and SSIM generated output from box mask

Discussing the model with the box masking method, it can be shown that the model now is still generating the PSNR and SSIM ranges following the linearly and exponential trendlines, respectively. However, when comparing the slope with the Line/Circle/Curve mask, the slope from the box mask method is lower. The reason for this answer is that the generated area does not have the neighborhood information with the model generation. Moreover, the monkey-generated figure can also show the limitation of the model generation, where the monkey now has only 1 eye instead of 2. Therefore, even though the model generation now is qualified, it does not satisfy the global context of the object, where information can be processed by the same characteristic of the same object.

5. Model Complexity and Limitations

Even though the idea of gated convolution is reasonable and new to approach the problem, there will be some problems and limitations for this method:

- Gated Convolutional Networks combine convolutional networks with a gating mechanism, which increases the complexity of the model, leading to time and resource consumption compared with the traditional method.
- As a basic building block of CNNs, the convolutional layer is designed to extract local patterns and lacks the ability to model global context. Therefore, even though some events/actions/objects have general and global characteristics (the same as the monkey-generated output), they cannot generate as expected.
- Due to the complexity of the model, the batch size needs to be 16 to satisfy the time requirement. However, a batch size of 1 is always the best for accuracy and image quality, leading to the fact that the model now is performing the best image quality.

6. Future Work

Future work includes applying different model architectures that produce better results. An example worth trying is applying a Self-Organized Operational Layer, which uses generative neurons for convolutional [2]. Therefore, the shaping and the complex pattern will be recognized and divided into specific regions, with different filters, leading to better efficiency of the output generation.

Another approach is applying neural style transfer for model architecture [3]. Therefore, the generated output can be a combination of the information from the global output, which has been classified and recognized by the model architecture, by using the max pooling approach and applying this recognition to the generated output.

7. References

[1] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, & Thomas Huang. (2019). Free-Form Image Inpainting with Gated Convolution

[2] Serkan Kiranyaz, Junaid Malik, Habib Ben Abdallah, Turker Ince, Alexandros Iosifidis, & Moncef Gabbouj. (2020). Self-Organized Operational Neural Networks with Generative Neurons.

[3]. Leon A. Gatys, Alexander S. Ecker, & Matthias Bethge. (2015). A Neural Algorithm of Artistic Style