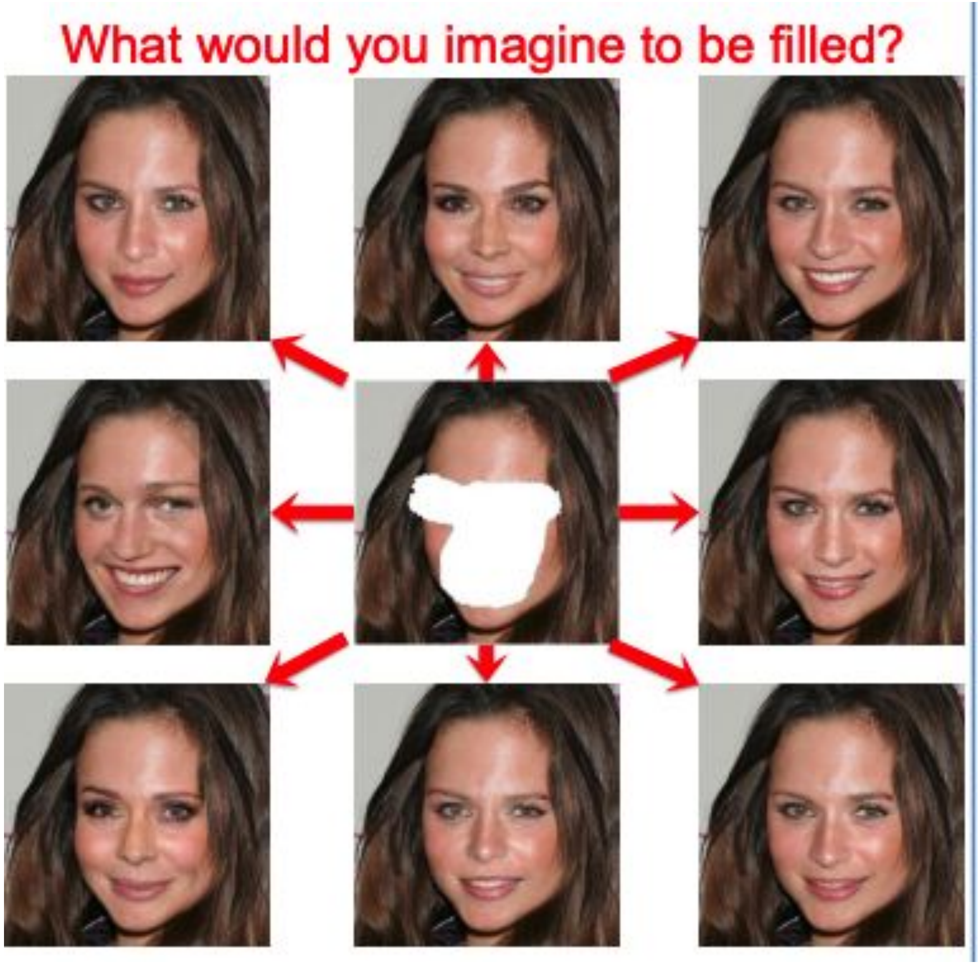
Performance Evaluation of Image Inpainting Algorithms Proposed in Pluralistic Image Completion Paper

By Yunting Chiu and Huong Doan

CSC-676 Computer Vision / Spring 2021 / American University

Outline



Source: Pluralistic Image Completion

Introduction

- Only one optimal result is typically produced in existing image inpainting
- Many methods overly focus on reconstructing the original image
- **PICNet** introduces random noises for generating diverse results with a deep generative network [1].

Goal: Multiple and Diverse Plausible Results (50 Outputs)

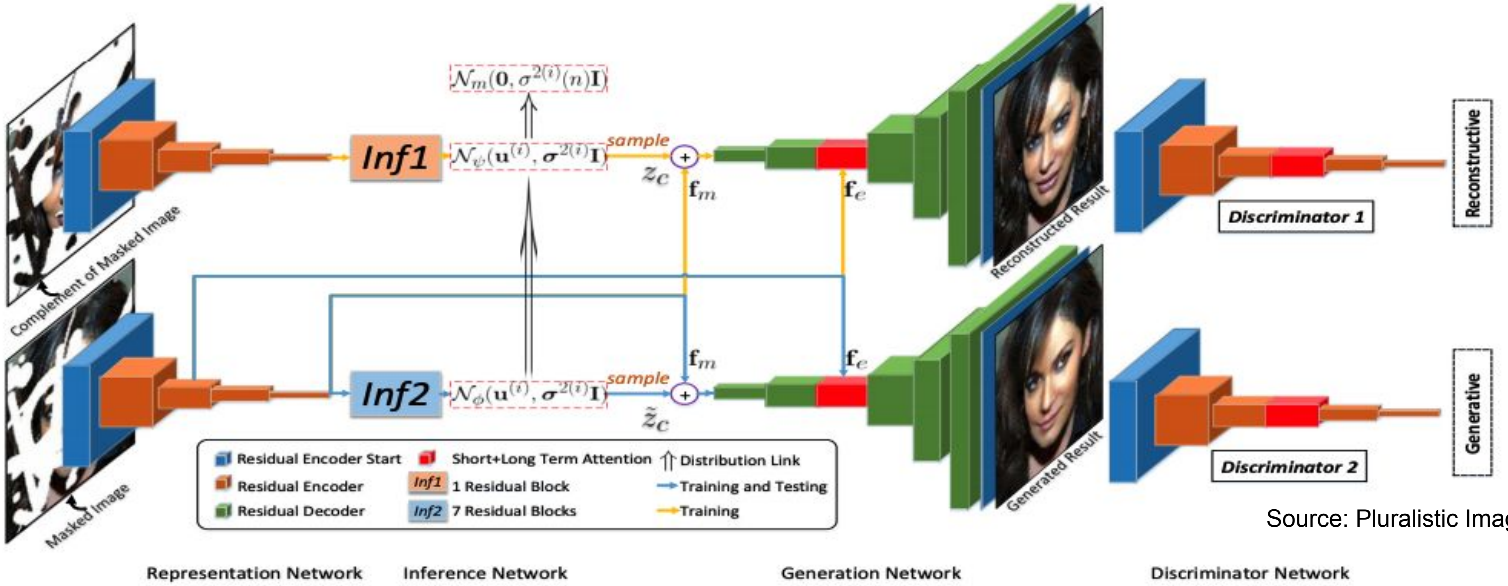
Related Works - T-MAD Model

- Texture Memory-Augmented Deep Patch-Based Image Inpainting [1] is the lated paper (2020) about image inpainting.
- This paper proposed a T-MAD model which is not only for image inpainting but also for object removal and high-resolution results.
- The method PICNet is also compared with the T-MAD method and other inpainting methods.

	random rectangle mask					random irregular mask				
	$l_1\downarrow$	PSNR$\uparrow$	SSIM$\uparrow$	TV$\downarrow$	FID$\downarrow$	$l_1\downarrow$	PSNR$\uparrow$	SSIM$\uparrow$	TV$\downarrow$	FID$\downarrow$
PatchMatch	14.795	15.038	0.819	**10.93**	11.630	11.276	17.387	0.839	**13.02**	10.751
GL[†]	13.806	15.659	0.821	12.04	10.379	10.269	17.403	0.855	13.50	8.295
PICNet	12.722	16.068	0.801	12.64	9.638	9.477	18.097	0.860	13.35	8.097
Edge	11.105	16.690	0.858	11.30	8.176	9.368	18.249	0.869	13.44	8.097
DeepFill	10.829	16.843	0.859	11.35	8.148	9.372	18.230	0.871	13.42	8.079
CRA	10.830	16.839	0.861	11.40	8.150	**9.260**	18.226	0.870	13.33	8.071
Our T-MAD	**10.334**	**17.203**	**0.867**	11.04	**8.131**	**9.261**	**18.351**	**0.873**	13.27	**8.058**

Source: Texture Memory-AugmentedDeep Patch-Based Image Inpainting

Methodology



Source: Pluralistic Image Completion

- Ig is the original image, and Im is the masked image. The method is mapping Ig to Im
- Define Ic as the converse of Im, which is constructed from the masked image
- This paper final goal is to take sample from p(Ic|Im) to recover images

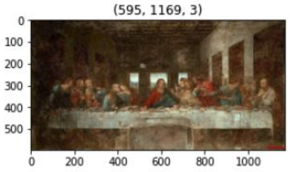
Approach - Datasets

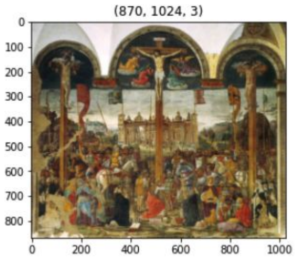
FFHQ-Thumbnail dataset:

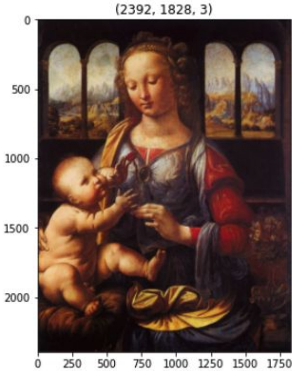
- Training: 65536 images
- Testing: 26 images (25+1)

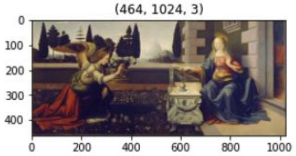
Art work dataset:

- Training: 8 images
- Testing: 8 images









PC: NVIDIA Research Projects & Best Artworks of All Time in Kaggle

Approach - Strength

- The authors did not use Art images to train and test PICNet
- **PICNet** excels at inpainting people's faces.
- Outputs are natural, realistic-looking
- One image can generate 50 diverse images with plausible content

Approach - Limitation

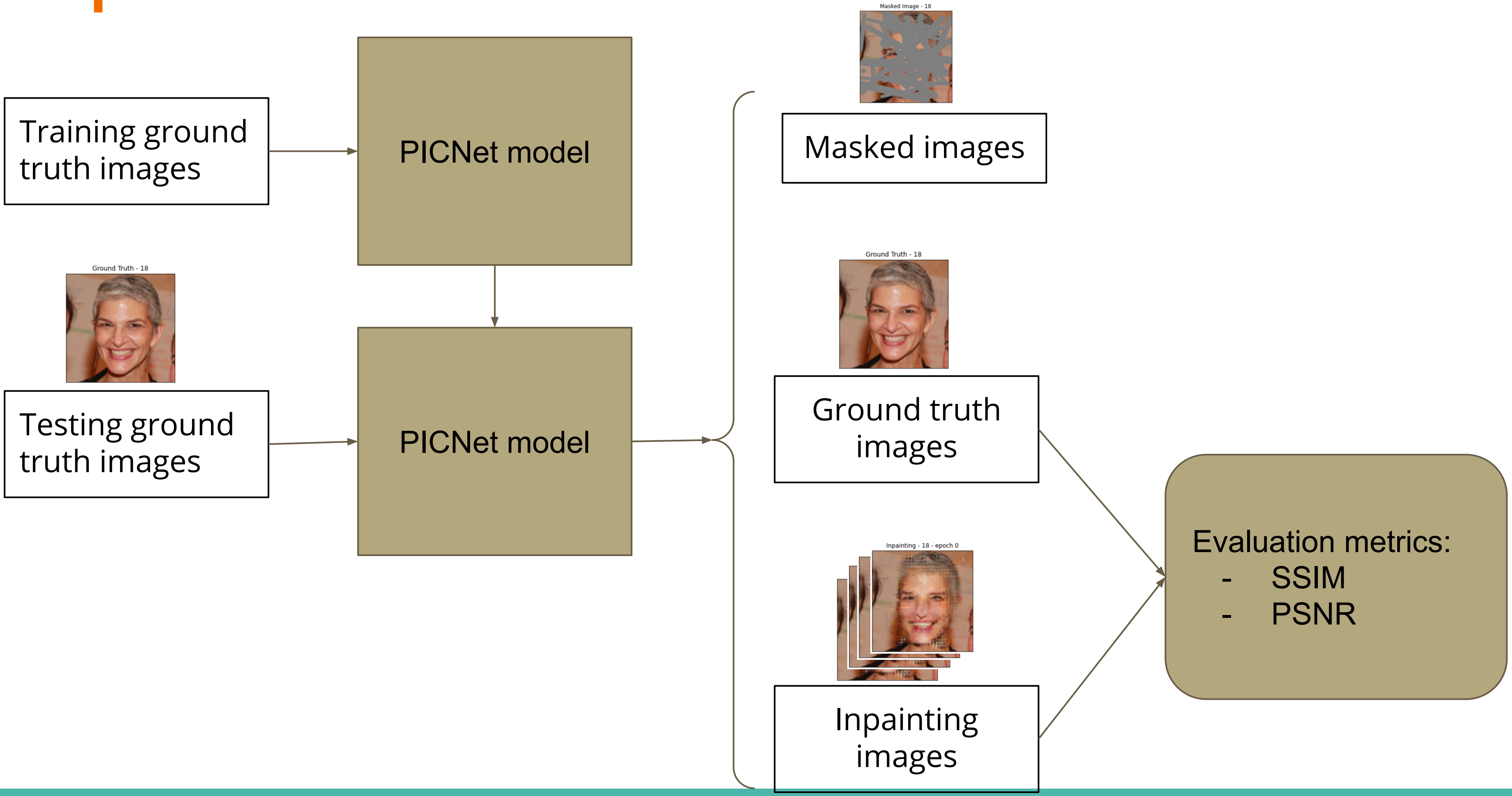
- Before running, the code must be modified

  self.img = self.img.cuda(self.gpu_ids[0], async=True)

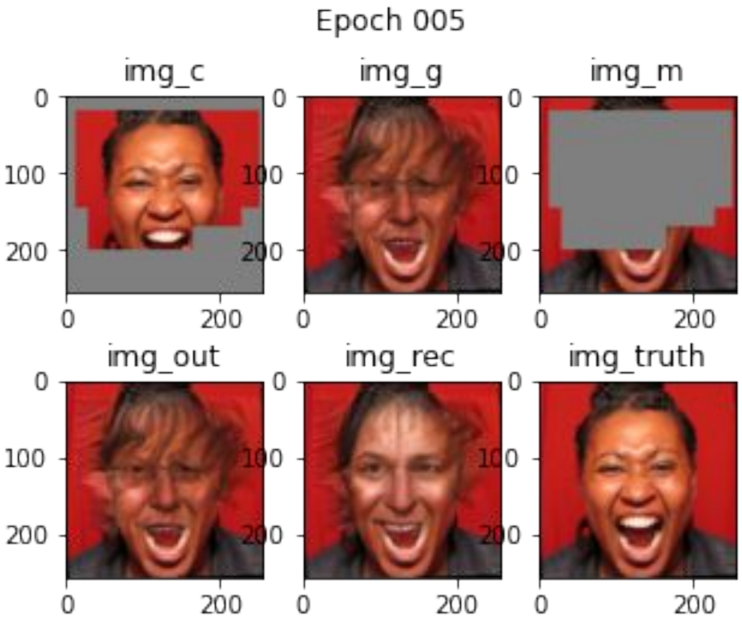
  change from *async=True* to *non_blocking=True*

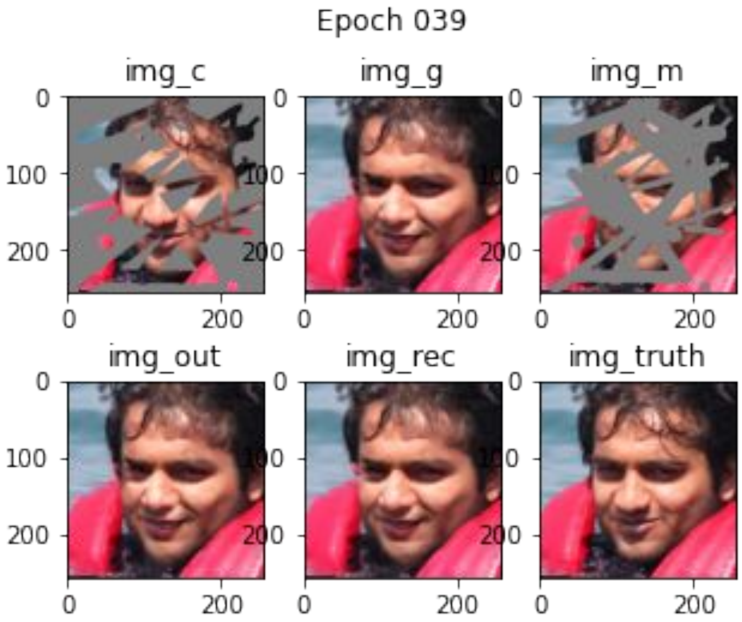
- Colab Pro is crashed after 12 hours of running time
- The maximum epoch is 41 (epoch is the number of passes of the entire training dataset)
- Training images: FFHQ (65536) > Celeba (24183)
- Running training model takes a long time
- Smaller image sizes are preferred

Experiments - Overview

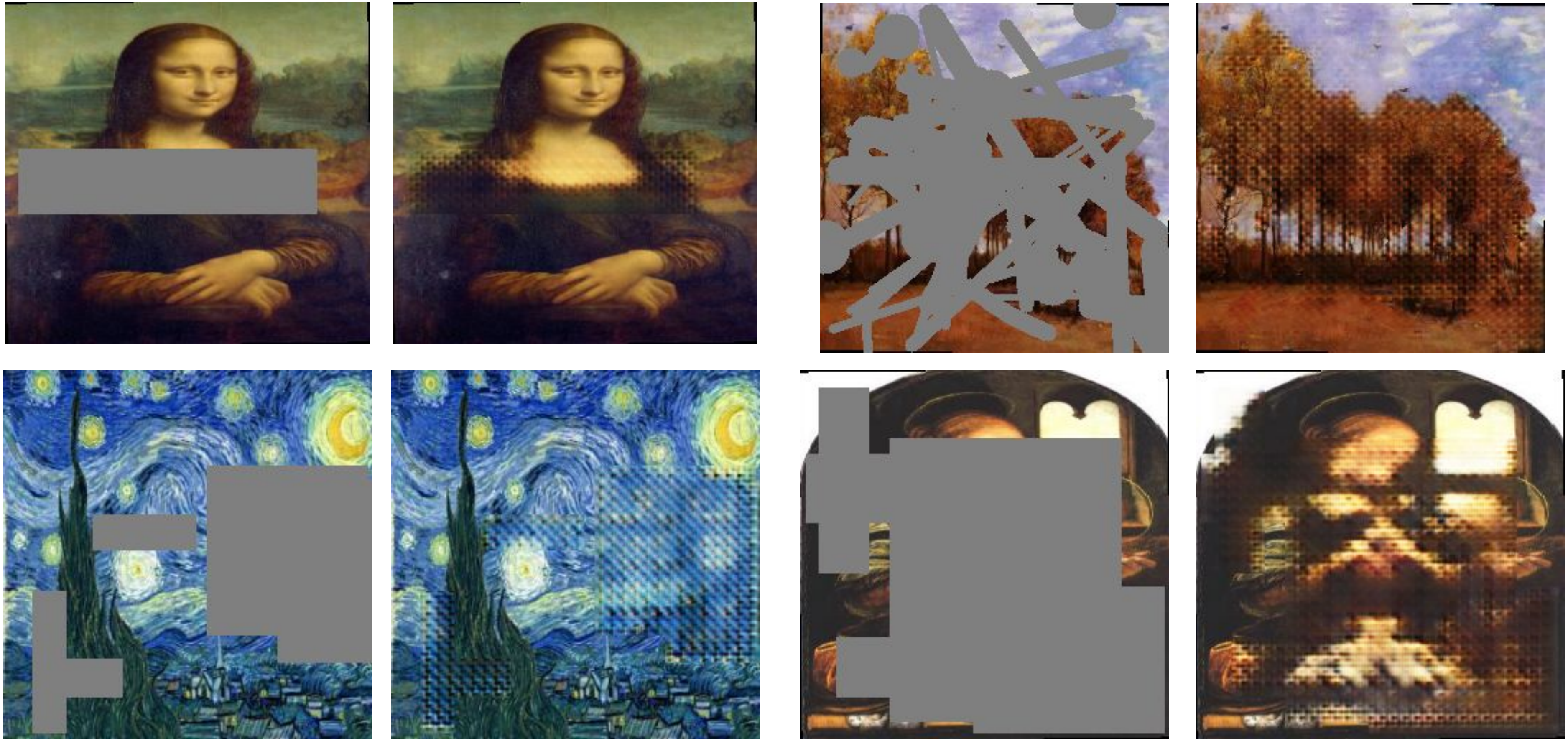


Approach- FFHQ Training Process





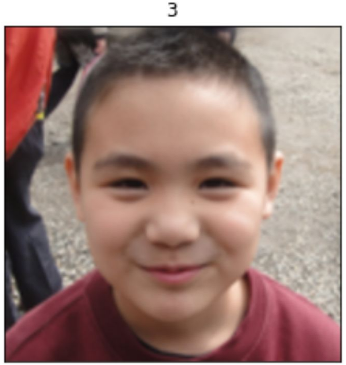
Approach - Art Work Training Process



Approach - FFHQ Testing Process

26 ground truth images are used as input.

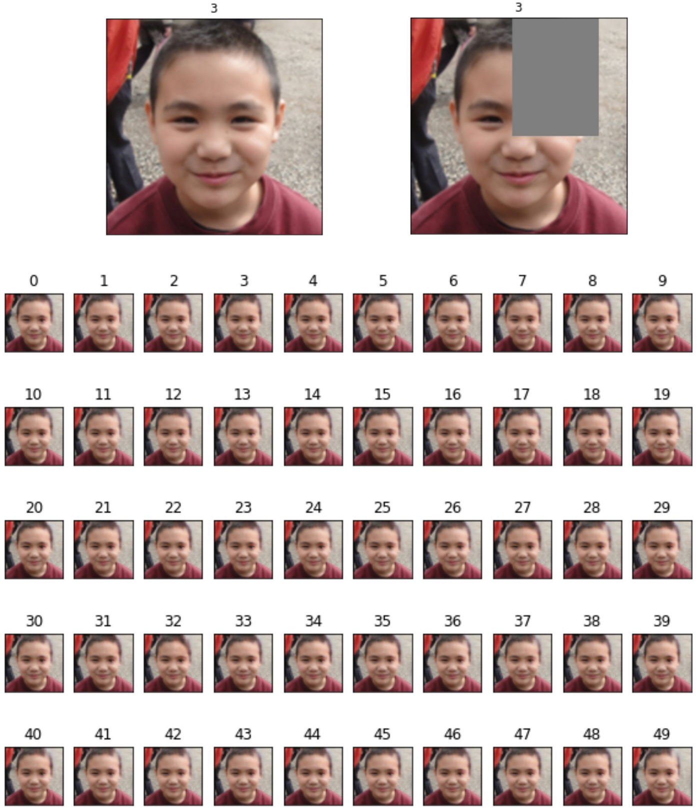
Each ground truth input image produces 50 diverse inpainting results with plausible content.



input

PICNet model

output



Results - Overview

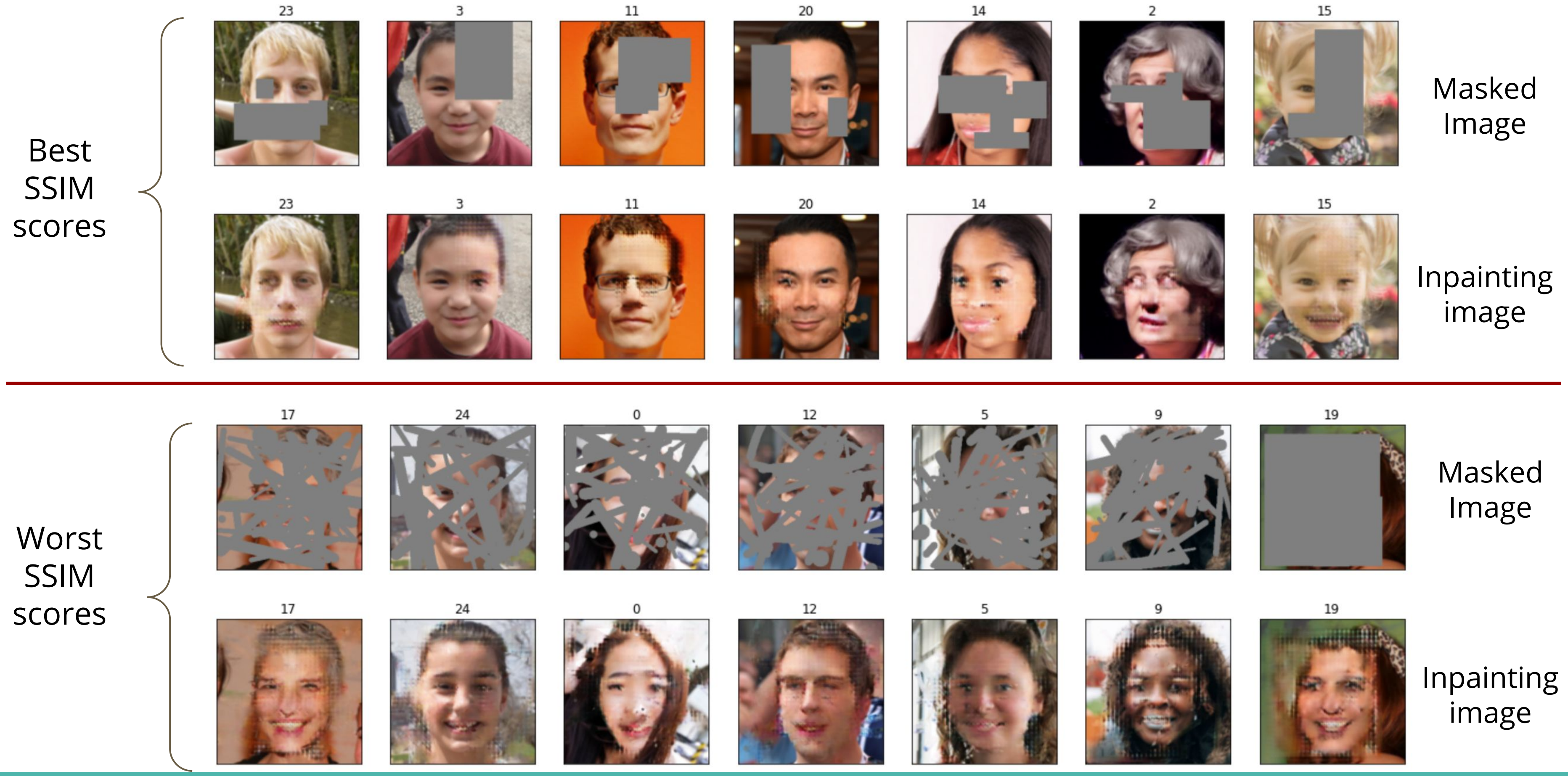
Best SSIM and PSNR scores

Worst SSIM and PSNR scores

Results - SSIM Score

image_id	mean	0	1	...	46	47	48	49
23	0.918419	0.917711	0.915873	...	0.917085	0.920520	0.917907	0.918097
3	0.910127	0.906360	0.908306	...	0.910367	0.909299	0.903683	0.908254
11	0.894535	0.899388	0.892184	...	0.893571	0.896297	0.888303	0.882920
20	0.894138	0.891464	0.897670	...	0.888794	0.890198	0.894168	0.892992
...	...	...	...	...	...	...	...	...
17	0.691193	0.693020	0.689691	...	0.695928	0.695544	0.694331	0.692799
25	0.676537	0.694302	0.679272	...	0.683070	0.678657	0.678499	0.662027
9	0.615126	0.618748	0.618975	...	0.619402	0.618387	0.618162	0.608343
19	0.520405	0.522354	0.530161	...	0.512296	0.540669	0.501832	0.522253

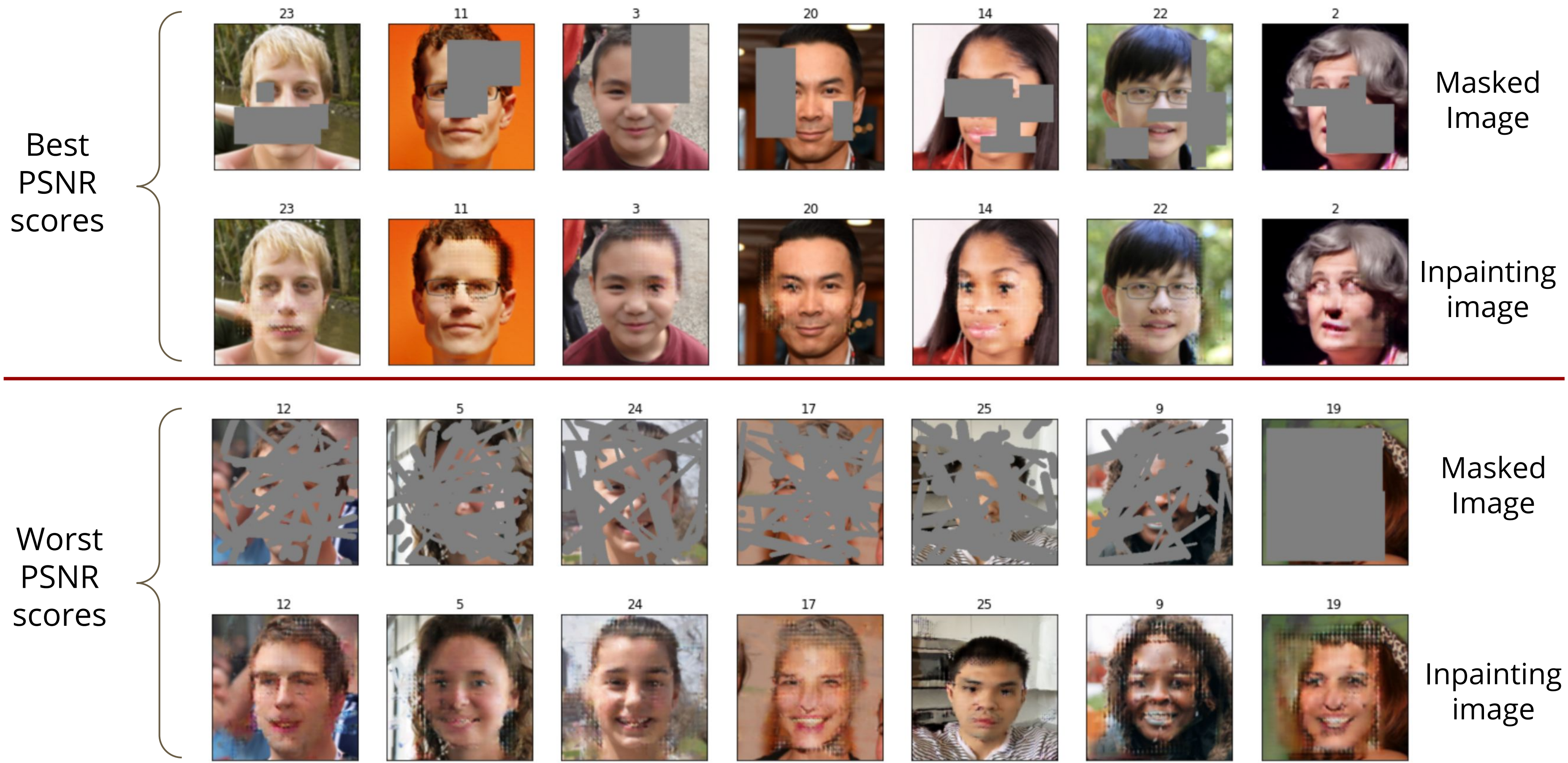
Results - SSIM Score



Results - PSNR Score

image_id	mean	0	1	...	46	47	48	49
23	36.643850	36.694018	36.503304	...	36.533961	36.822991	36.718100	36.613575
11	36.148645	36.271907	36.118319	...	36.125391	36.228952	36.049256	35.743276
3	35.976883	35.900848	35.951420	...	36.033813	35.990388	35.798776	35.860911
20	35.894791	35.865022	35.928127	...	35.789940	35.893077	35.937917	35.863821
...	...	...	...	...	...	...	...	...
12	31.264525	31.234762	31.445818	...	31.106084	31.195558	31.258641	31.213109
5	31.129193	31.176131	31.123171	...	31.212215	31.165599	31.020753	31.111523
9	30.901778	30.916139	30.926596	...	30.925427	30.904244	30.934820	30.847321
19	29.737357	29.748785	29.750040	...	29.769799	29.810887	29.724323	29.748198

Results - PSNR Score



Results - Comparison

image_id	ssim_mean	psnr_mean	ssim_rank	psnr_rank
23	0.918419	36.643850	1.0	1.0
3	0.910127	35.976883	2.0	3.0
11	0.894535	36.148645	3.0	2.0
20	0.894138	35.894791	4.0	4.0
14	0.888750	35.352410	5.0	5.0
...	...	...	...	...
24	0.692855	31.427512	22.0	21.0
17	0.691193	31.462293	23.0	20.0
25	0.676537	31.547474	24.0	18.0
9	0.615126	30.901778	25.0	25.0
19	0.520405	29.737357	26.0	26.0

Conclusion & Discussion

- The **smaller** the mask region is, the **better** the result of inpainting process is.
- The **more** SSIM (similarity scores) is, the **better** the result image is.
- For most of the cases, the **better** result images have **higher** PSRN scores.
- Therefore, the **good** result images will **mostly** have the **high** SSIM and PSNR scores.
- In general, the model is able to generate different results given a masked input image. However, those results did **not** really look **plausible**.

Future Works

Maybe upgrade computing environment or having a better GPU for getting the faster running time.

Continue to complete training and testing process of the artwork dataset and then do the evaluation metrics calculation.

References

[1] Xu, R., Guo, M., Wang, J., Li, X., Zhou, B., & Loy, C. (2020). *Texture Memory-Augmented Deep Patch-Based Image Inpainting.*

[2] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. Pluralistic image completion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438– 1447, 2019

Thank you for listening!
Q&A