Improvement in MetaF2N

Raj Agrawal (210809)

Dataset Preprocessing Changes

Upon reviewing the results from my initial model, I observed unwanted lines appearing on both the faces and backgrounds of the images. Further investigation led me to a GitHub repository that highlighted dataset misalignment as a possible cause, specifically with the FFHQ dataset. To address this issue, I reprocessed the dataset to ensure alignment, using the scripts `ffhq-align.py`, `generate_multiscale.py`, and `generate_tfrecord.py`. This preprocessing step was intended to improve image quality and reduce artifacts in the final model output.

Training details

This time I trained my model over 10,000 aligned_FFHQ images and 1,000 Multiscale_DF2K images over 10,000 iterations. The average training time for 100 iteration is around 160 seconds. The total training time is around 5-6 hours on NVIDIA 24 GB GPU. I trained my model several times by changing the hyper-parameter of the added SSL loss component and the best results are obtained for $\lambda = 0.05$.

GitHub and Google Drive Links

The Updated code and environment setup details can be found in the following GitHub repository: **MetaF2N**.
My new Model can be found on the Google drive at the the following link: **New Model**.
The updated preprocessed data can be found on Google drive at the following link:**Preprocessed Data**.
The test Data images on which I tested the model can be found on Google Drive at the following link:**Test Data**.
The results of the test data on which I tested the model can be found on Google drive at the following link: **Result Data**.

New Idea Implementation

In our approach, we experimented with the concept of self-similarity loss, based on the observation that real-world images exhibit similarities between neighboring patches. To incorporate this idea, I updated the `train.py` script by adding a self-similarity loss function, which captures the self-similarity within images. To compute this loss, I created a separate file named `ssl.py`.
As detailed in the `Mathematical_POC file`, I first generated the similarity maps for both the ground truth (GT) and the output image. I then applied a mask to these maps, considering only similarity values above a specified threshold, set at 20. This resulted in a masked similarity map. The self-similarity loss was then calculated as the absolute difference between the masked similarity maps of the output and GT images.
Although I also attempted to use the Kullback-Leibler (KL) divergence for calculating the self-similarity loss, the results were less effective compared to using the absolute difference. Consequently, the final loss function is defined as:

$$\mathcal{L}_{total} = \mathcal{L}^{T_i} + \lambda \mathcal{L}_{SSL} \tag{1}$$

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_1 + \lambda_2 \mathcal{L}_{\text{LPIPS}} + \lambda_3 \mathcal{L}_{\text{adv}} + \lambda_4 \mathcal{L}_{\text{reg}} + \lambda \|\bar{S}_{SR} - \bar{S}_{HR}\|_1) \tag{2}$$

Observations

For an optimal model, the PSNR metric should be high, while other loss metrics, including LPIPS, FID, and NIQE, should ideally remain low. Upon analyzing the results from my new_model, I observed that the loss values are comparable to, and in certain cases, even outperform those reported in the original paper. Notably, NIQE and FID loss metrics have decreased in some instances relative to the paper's baseline, suggesting improved model accuracy despite training for only 10,000 iterations compared to the 100,000 iterations used in the reference study. Furthermore, additional improvements in performance could potentially be achieved through hyper-parameter tuning, particularly by adjusting the value of λ.

Additions

- I added code to generate multiscale Df2k images, as this functionality was not provided in the GitHub repository. The code is in the script `generate_multiscale.py`.
- I added the `LPIPS` folder, which was also missing from the repository, and updated the `lpips.py` code.
- I added the code for the alignment of FFHQ dataset available online. The code is in the script `ffhq-align.py`
- I added the code for SSL loss which is self similarity loss in the folder named as SSL. The code is in the script `ssl.py`
- I updated the `calculate_metric.py` script, making it compatible for both GPU and CPU environments.

Test Results:Figures and Metrics

Performance metrics during training:

Epoch No.	Time(s) per 100 Epoch	Epoch Loss (Post)
1	84	0.62
100	150	0.498
500	160	0.44
1000	159	0.32
2000	155	0.37
5000	166	0.39
10000	168	0.35

Table 1: Performance metrics for each epoch during training.

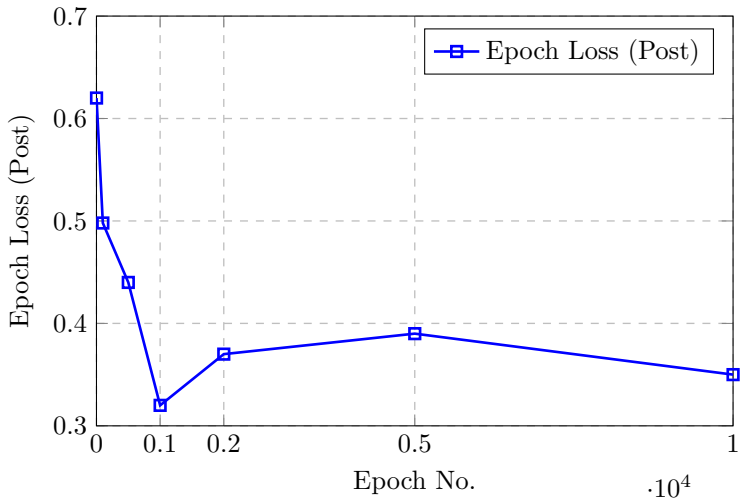


Figure 1: Epoch Loss (Post) over different Epoch Numbers.

Loss Metric

Loss metrics can be calculated using calculate_metrics.py script.
Table 2 : The Loss metrics for my model- Old and New.
Table 3 : The Loss metrics provided in the paper.

My model	Old				New			
Dataset	**PSNR**	**LPIPS**	**FID**	**NIQE**	**PSNR**	**LPIPS**	**FID**	**NIQE**
CelebA_iid	17.26	0.4616	130.54	5.56	25.72	0.310	46.5	4.17
CelebA_ood	17.38	0.4648	116.99	5.61	24.88	0.313	44.47	4.44
FFHQ_iid	16.93	0.4589	204.77	5.32	26.13	0.301	46.10	3.94
FFHQ_ood	17.35	0.4536	111.86	5.42	25.34	0.303	45.40	4.21
FFHQ_Multi_iid	16.79	0.4570	134.55	5.23	25.61	0.308	43.85	3.66
FFHQ_Multi_ood	16.85	0.4491	127.59	5.23	25.11	0.312	43.31	3.65

Table 2: Loss Metrics for Various Datasets: My Model

Dataset	PSNR	LPIPS	FID	NIQE
CelebA_iid	25.76	0.289	45.22	4.10
CelebA_ood	25.00	0.297	46.23	4.47
FFHQ_iid	26.30	0.279	44.51	3.94
FFHQ_ood	25.65	0.283	44.30	4.36

Table 3: Loss Metrics for Various Datasets: Mentioned in the paper

Figures

Some of the pairs provided below serve as references for evaluating the performance of my model.

(a) LQ (b) Old_Model (c) New_Model (d) GT

Figure 2: Test Results: FFHQ_iid

(a) LQ (b) Old_Model (c) New_Model (d) GT

Figure 3: Test Results: FFHQ_ood

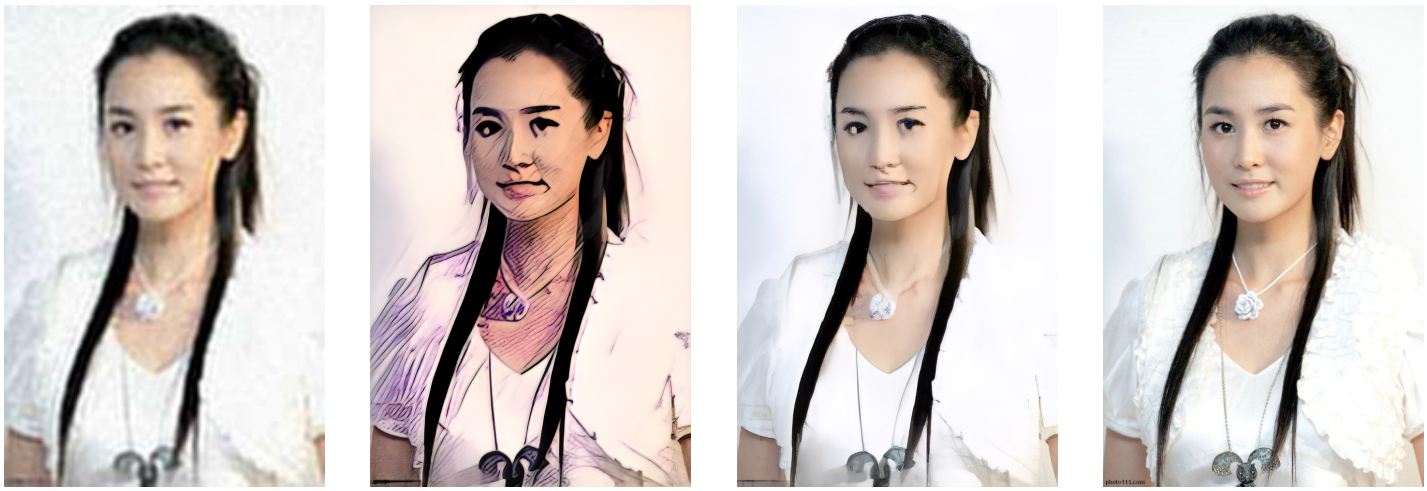
(a) LQ (b) Old_Model (c) New_Model (d) GT

Figure 4: Test Results: FFHQ_Multi_iid

(a) LQ (b) Old_Model (c) New_Model (d) GT

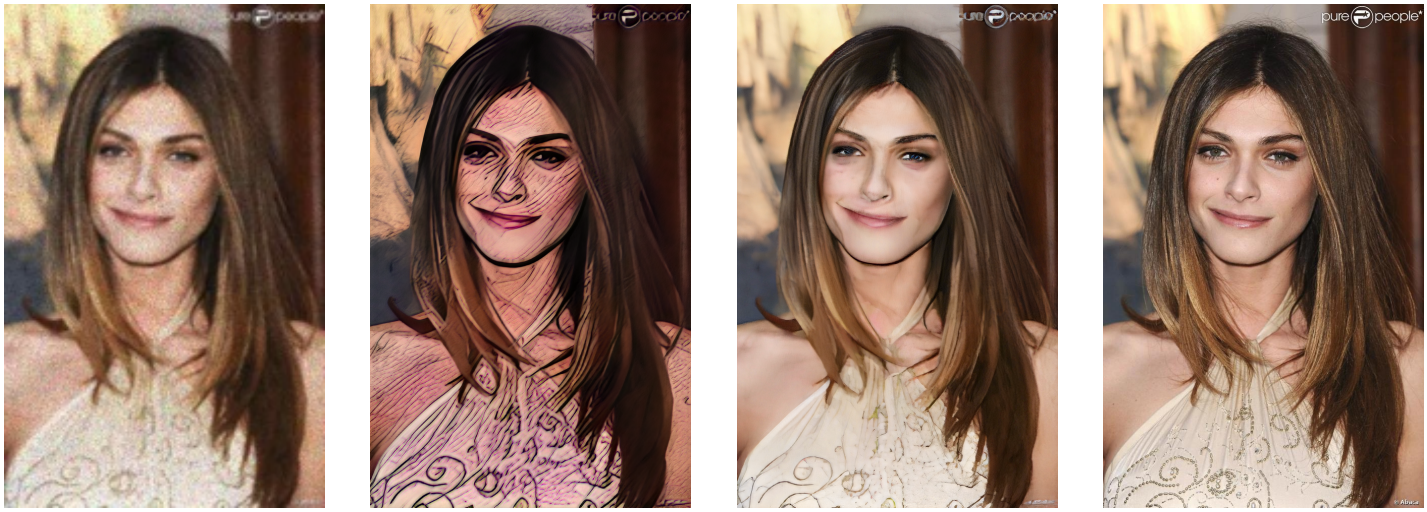
Figure 5: Test Results: FFHQ_Multi_ood

...



(a) LQ (b) Old_Model (c) New_Model (d) GT

Figure 6: Test Results: CelebA_iid



(a) LQ (b) Old_Model (c) New_Model (d) GT

Figure 7: Test Results: CelebA_ood

References

[1] Du Chen, Zhengqiang Zhang, Jie Liang, and Lei Zhang. ***SSL****: A Self-similarity Loss for Improving Generative Image Super-resolution.* Available at `https://arxiv.org/pdf/2408.05713`.-For SSL loss

[2] Richard Zhang, Phillip Isola1, Alexei A. Efros,Eli Shechtman, Oliver Wang. ***LPIPS****: The Unreasonable Effectiveness of Deep Features as a Perceptual Metric.* Available at `https://arxiv.org/pdf/1801.03924`.-For LPIPS loss

[3] FFHQ_Alignment- Available at `https://github.com/happy-jihye/FFHQ-Alignment`.

[4] Xintao Wang, Liangbin Xie, Chao Dong, Ying Shan. ***Real-ESRGAN****: Training Real-World Blind Super-Resolution with Pure Synthetic Data.* Available at `https://arxiv.org/pdf/2107.10833`.-For multiscale DF2K dataset.

[5] Zhicun Yin, Ming Liu, Xiaoming Li, Hui Yang,Longan Xiao, Wangmeng Zuo. ***MetaF2N****: Blind Image Super-Resolution by Learning Efficient Model Adaptation from Faces.* Available at `https://arxiv.org/pdf/2309.08113`.