

IPR PROJECT-1

MetaF2N: Blind Image Super-Resolution by Learning Efficient Model Adaptation from Faces

Raj Agrawal (210809)

1 Test Results: Figures and Metrics

The results of the test data images on which I tested the model can be found on Google Drive at the following link: **Result Data**.

1.1 Figures

Some of the pairs provided below serve as references for evaluating the performance of my model. For more you can check **Result Data**.



(a) LQ_celebA_iid



(b) LQ_celebA_ood



(c) LQ_FFHQ_multi_iid



(d) LQ_FFHQ_iid



(e) Result_celebA_iid



(f) Result_celebA_ood



(g) Result_FFHQ_multi_iid



(h) Result_FFHQ_iid

Figure 1: Test Results

1.2 Loss Metric

For calculating the loss metrics such as LPIPS, FID, PSNR, and NIQE, I first updated the `calculate_metrics.py` script and made the necessary changes. Then, I computed these four metrics for each dataset, including CelebA_iid, CelebA_ood, FFHQ_iid, FFHQ_ood, FFHQ_Multi_iid, and FFHQ_Multi_ood. The results are presented in the Table 1: My model and Table 2: Results Provided in the paper.

Dataset	PSNR	LPIPS	FID	NIQE
CelebA_iid	17.26	0.4616	130.54	5.56
CelebA_ood	17.38	0.4648	116.99	5.61
FFHQ_iid	16.93	0.4589	204.77	5.32
FFHQ_ood	17.35	0.4536	111.86	5.42
FFHQ_Multi_iid	16.79	0.4570	134.55	5.23
FFHQ_Multi_ood	16.85	0.4491	127.59	5.23

Table 1: 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 2: Loss Metrics for Various Datasets: Mentioned in the paper

1.3 Observations

For a good model, the PSNR should be high, while all other loss metrics (LPIPS, FID, and NIQE) should be low. When comparing my results with the ones presented in the paper, I observed slightly less accurate results. This can be attributed to two reasons:

1.3.1 Training Iterations

I trained my model for only 1,200 iterations, whereas the authors of the paper trained their model for 100,000 iterations.

1.3.2 Dataset Size

I used a subset of the dataset, approximately one-third of the dataset size used by the authors during the model's training. However, despite these differences, the order of the metrics is nearly the same. Additionally, I would like to highlight that, as shown in the results, the model performs well on images of different sizes. This is because I trained it on the `multiscale_Df2k` sample data.

1.4 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 updated the `calculate_metric.py` script, making it compatible for both GPU and CPU environments.