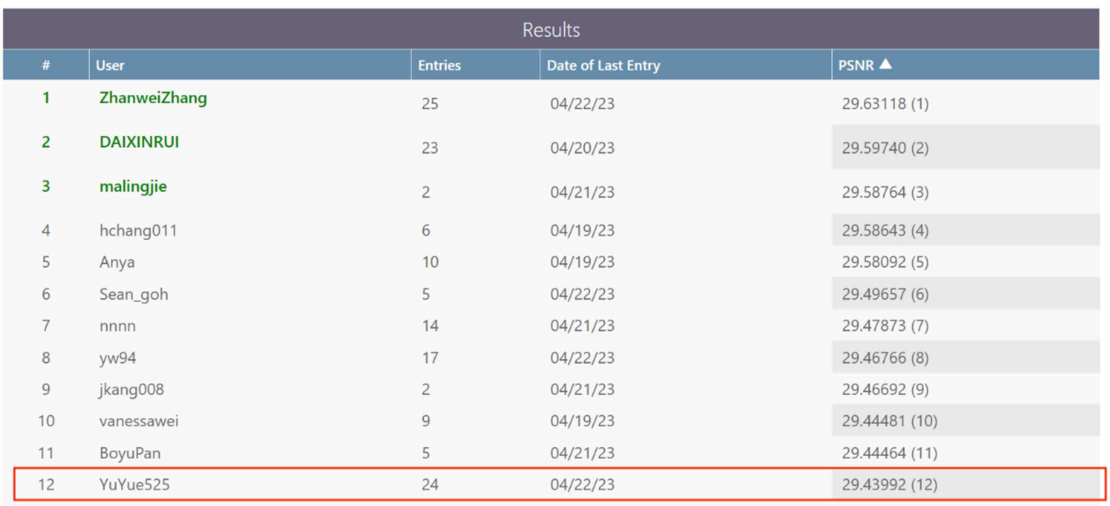
AI6126 Project 1 Report
Blind Face Super-Resolution

Yu Yue
G2202151A

Introduction

In this project, we generate high-quality face images from the corrupted low-quality face images by implement SRResNet [1] model to achieve high PSNR score, and we also implemented SRGAN model to generate more natural high-quality face images. Our implementation achieves **29.4399 PSNR** on the test set and ranked 12^h till 14:00 p.m. 22/4/2023. All the code can be downloaded from GitHub. We introduce our implementation details in this report.



Results

#	User	Entries	Date of Last Entry	PSNR ▲
1	ZhanweiZhang	25	04/22/23	29.63118 (1)
2	DAIXINRUI	23	04/20/23	29.59740 (2)
3	malingjie	2	04/21/23	29.58764 (3)
4	hchang011	6	04/19/23	29.58643 (4)
5	Anya	10	04/19/23	29.58092 (5)
6	Sean_goh	5	04/22/23	29.49657 (6)
7	nnnn	14	04/21/23	29.47873 (7)
8	yw94	17	04/22/23	29.46766 (8)
9	jkang008	2	04/21/23	29.46692 (9)
10	vanessawei	9	04/19/23	29.44481 (10)
11	BoyuPan	5	04/21/23	29.44464 (11)
12	YuYue525	24	04/22/23	29.43992 (12)

Model

In our implementation, we use SRResNet implementation from MMEditing v1.0 codebase [2] for convenience. We extend the baseline SRResNet model from 16 blocks to 24 blocks to achieve higher PSNR score. The number of parameters is **2,108,419** which is less than the limitation of the training parameters. Then we use SRGAN to generate much natural face images. For the SRGAN part, we use the implementation from MMEditing v0.0 because of some bugs in MMEditing v1.0 codebase. Besides, for the generator of the SRGAN, we also extend the number of blocks from 16 to 26, and the number of training parameters of the generator is **2,256,131** which is also less than the limitation.

Loss Functions

When training the SRResNet model, we only use **L1 loss** as the pixel loss. Besides, for the SRGAN, the **Vanilla loss** is introduced as the GAN loss and the perceptual loss is also introduced.

Training and Configurations

Due to the time limitations of our GPU cluster (maximum 8 hours per task submission), we trained the model with the cosine annealing learning rate decay, the cycle of the cosine annealing is 30,000 iterations and we trained the model for 5 cycles, totally 150,000 iterations. For the first four cosine annealing cycles, the learning rate decays from 1e-4 to 1e-7, and decays from 1e-5 to 1e-9 in the final cycle. We use Adam optimizer, and the batch size is set to 12. We record the training curves of the first 30k iterations and we also provided the training log files at GitHub for your reference. Finally, we achieve **29.7315 PSNR** on the validation set and **29.4399 PSNR** on the test set. Figure 2 shows the results of the best model on some validation images.

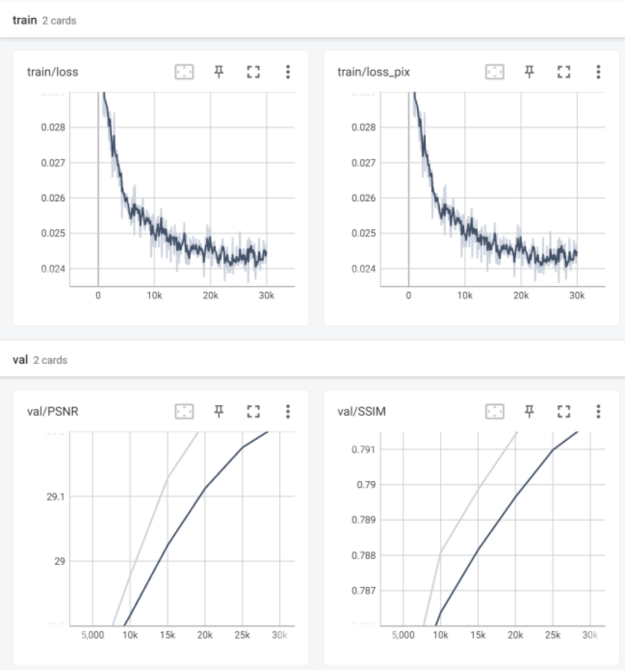


Figure 1. Training curves

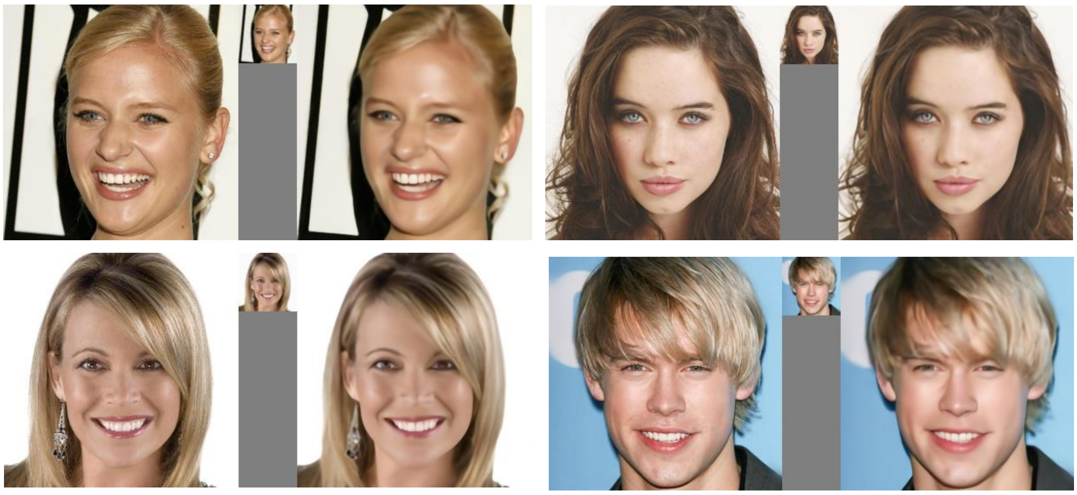


Figure 2. validation results of the SRResNet model. left: ground truth, middle: low-quality right: high-quality result.

For the training of SRGAN model, because we use MMEditing 0.0 version, we separately trained the SRResNet generator and then use the pre-trained generator to train the discriminator. We train the generator with the same setting as above and then we train the whole SRGAN model for 40,000 iterations with multi-step learning rate decay. The training loss curves and validation PSNR curve are shown in Figure 3.

As we can see, when training the SRGAN model, the loss values for the discriminator oscillates but generally decreases, and the perceptual loss is continuously decreasing. The GAN loss and the pixel loss are oscillating but generally increases. Besides, the PSNR of validation also oscillates but is lower than the pure SRResNet generator.

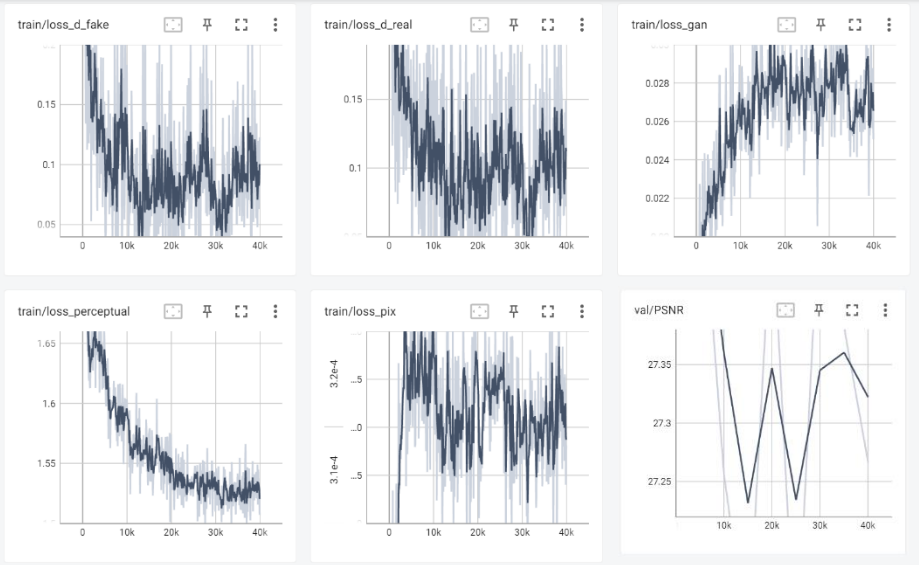


Figure 3. Training loss curves and validation PSNR curve of SRGAN model

Then we compare the performance difference between the SRResNet and the SRGAN model on the given six real test images. Figure 4 shows one of the examples. As we can see from the two results, SRGAN generates more realistic beards, teeth and facial wrinkles. But for the SRResNet, the result is more blur and unreal. For the complete six example results, please refer to Appendix or our GitHub.

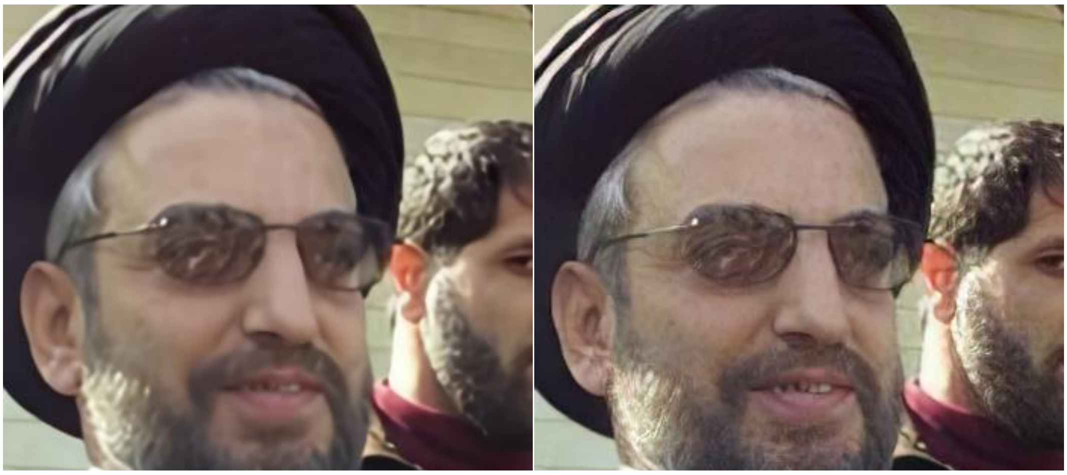


Figure 4. Comparison between SRResNet and SRGAN model on one real test images, left: SRResNet; right: SRGAN.

Training Machine

We trained our models on one Nvidia A40 GPU with 30 GB memory usage provided by SCSE GPU clusters of NTU, the time limitation of each submission task is 8 hours.

Reference

[1] C. Ledig et al., Photo-realistic Single Image Super-Resolution using a Generative Adversarial Network, CVPR 2017.

[2] MMEditing. https://github.com/open-mmlab/mmediting/tree/main.

Appendix

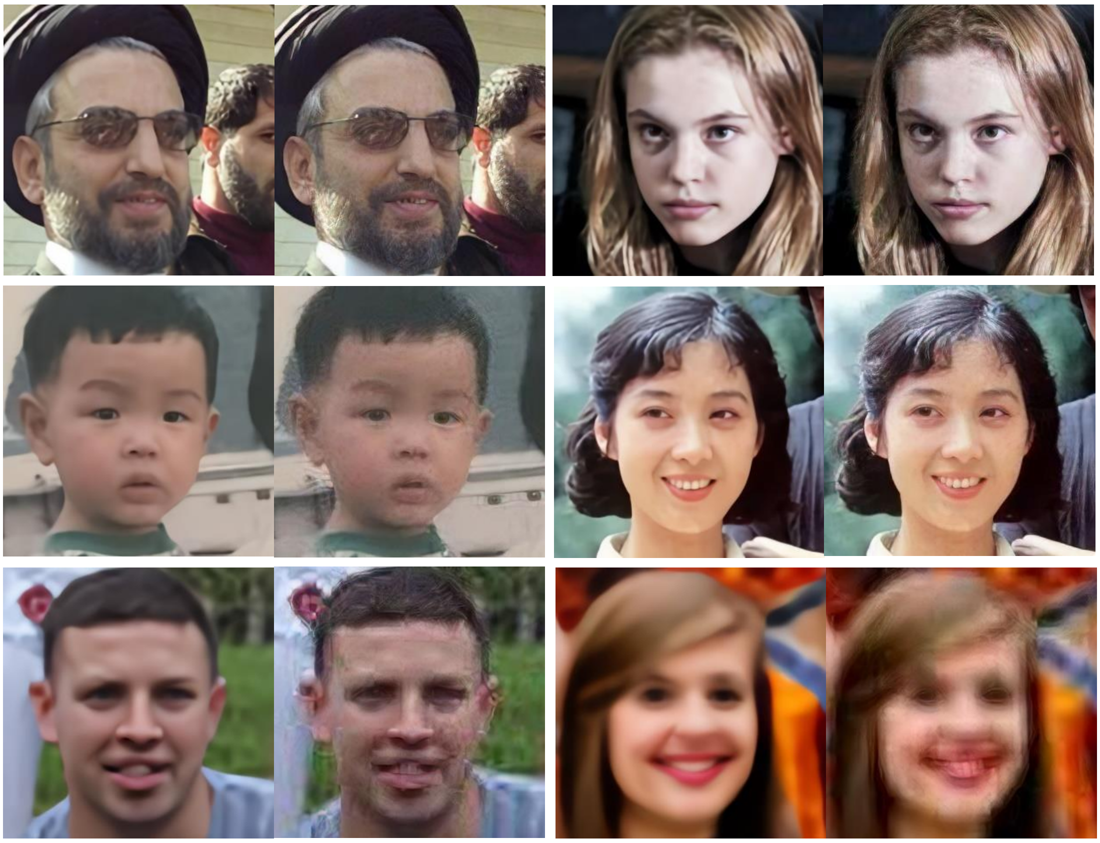


Figure 5. Comparison between SRResNet and SRGAN model on real test images, left: SRResNet; right: SRGAN.