AI6126: Advanced Computer Vision

Project 2
Blind Face Super-Resolution

Ron Kow Kheng Hui

ID: G1903451J

22 April 2022

School of Computer Science and Engineering

Nanyang Technological University

1 Background

This project is an image super-resolution competition. The objective is to generate high-quality face images from low-quality ones. The data are from FFHQ[1], which was originally created as a benchmark for Generative Adversarial Networks (GAN) [1]. 4000 training images and 400 images each in the validation and test sets are provided.

2 Specifications and Configurations

2.1 Machine Specifications for Model Training

- Operating system: Ubuntu 18.04
- GPU: Nvidia GeForce RTX 2060, 6 GB memory
- Number of GPUs: 1

2.2 Model Configurations and Number of Parameters

The competition imposed a maximum number of 1,821,085 parameters for each model. The SRResNet [2] configuration is provided as a baseline. We modifed the network depth by increasing the number of blocks to 20 in this SRResNet configuration. The details of the configuration are as follows:

- Original configuration: `srresnet_ffhq_300k.py`
- Training samples per GPU: 8
- Original number of parameters: 1,517,571
- Number of parameters after modification: 1,812,995

For our experiments, we used MMEditing[2] [3], the open source image editing toolbox from the OpenMMLab project[3]. In addition to SRResNet, we modified and trained the basic Residual-in-Residual Dense Block network (RRDBNet) [4]. This model uses only L1 loss and is intended to be used as pre-training before training the ESRGAN model with perceptual loss function.

In order for the ESRGAN model to satisfy the competition limit, we reduced the number of mid-channels from 64 to 16, the number of blocks from 23 to 16, and the number of growth channels from 32 to 7. This would give the ESRGAN model 1,698,364 parameters. Using the same number of channels and blocks, the number of parameters for RRDBNet is reduced to 636,947.

[1] https://github.com/NVlabs/ffhq-dataset
[2] https://github.com/open-mmlab/mmediting
[3] https://openmmlab.com/

Unfortunately, due to insufficient time before the end of the competition, we did not manage to train the ESRGAN model. We will only report the results of the pre-training on RRDBNet for comparison with SRResNet. The details of the configuration are as follows:

- Original configuration: `esrgan_psnr_x4c64b23g32_g1_1000k_div2k.py`
- Training samples per GPU: 16
- Original number of parameters: 16,697,987
- Number of parameters after modification: 636,947

2.3 Loss Function and Hyperparameters

The following hyperparameter configurations were used:

- Loss function (SRResNet, RRDBNet): L1 loss
- Optimizer (SRResNet, RRDBNet): Adam (Initial learning rate: 2×10^{-4})
- Learning rate schedule (SRResNet): Cosine annealed restart (Minimum learning rate: 1×10^{-7}, period: 150k iterations)
- Learning rate schedule (RRDBNet): Step [50k, 100k, 200k, 300k]

2.4 Annotation File

For both models, we used `SRAnnotationDataset` type for training. We created an annotation file containing information on the image filenames and the shape of the ground truth images. The first three rows are:

```
00000.png (512,512,3)
00001.png (512,512,3)
00002.png (512,512,3)
```

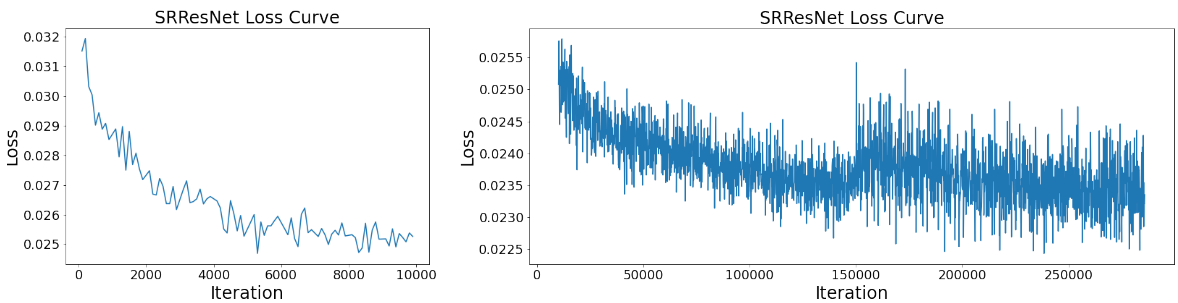
3 Training and Results

3.1 Evaluation Metric

The validation and test sets are evaluated on the Peak Signal-to-Noise Ratio (PSNR) metric. PSNR, which is based on accuracy of pixel values, does not reflect human perception of an image well. Studies have shown that models using the perceptual loss function [2] (which results in lower PSNR scores as compared to using L1 loss) produces images that are perceptually more similar to the source images.

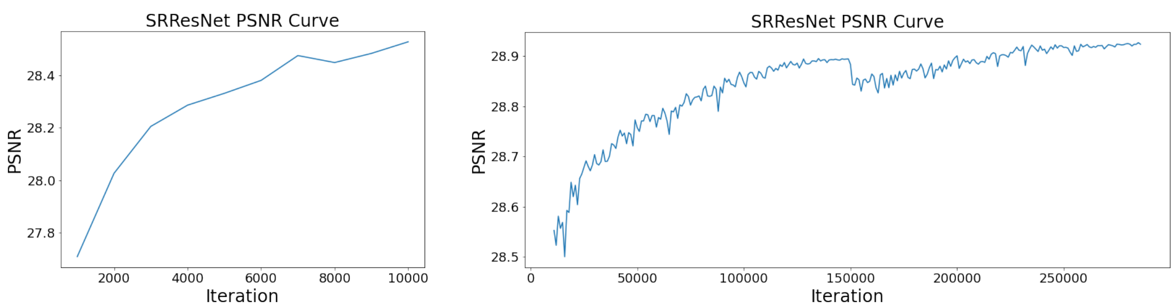
3.2 Training Curves: SRResNet

The loss and PSNR curves for SRResNet are shown in Figure 1 and Figure 2 respectively.



(a) **(b)**

Figure 1: Loss curves (SRResNet): (a) First 10k iterations (b) 10k to 286k iterations.

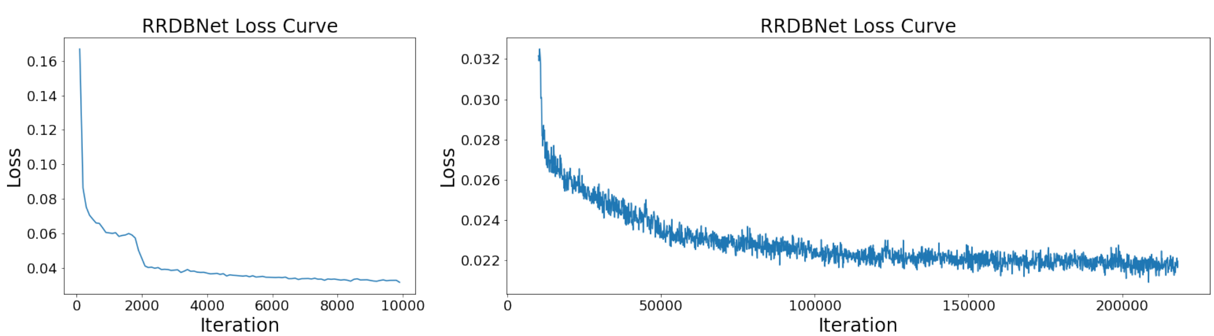


(a) **(b)**

Figure 2: PSNR curves (SRResNet): (a) First 10k iterations. (b) 10k to 286k iterations.

3.3 Training Curves: RRDBNet

The loss and PSNR curves for RRDBNet are shown in Figure 3 and Figure 4 respectively.



(a) **(b)**

Figure 3: Loss curves (RRDBNet): (a) First 10k iterations (b) 10k to 218k iterations.

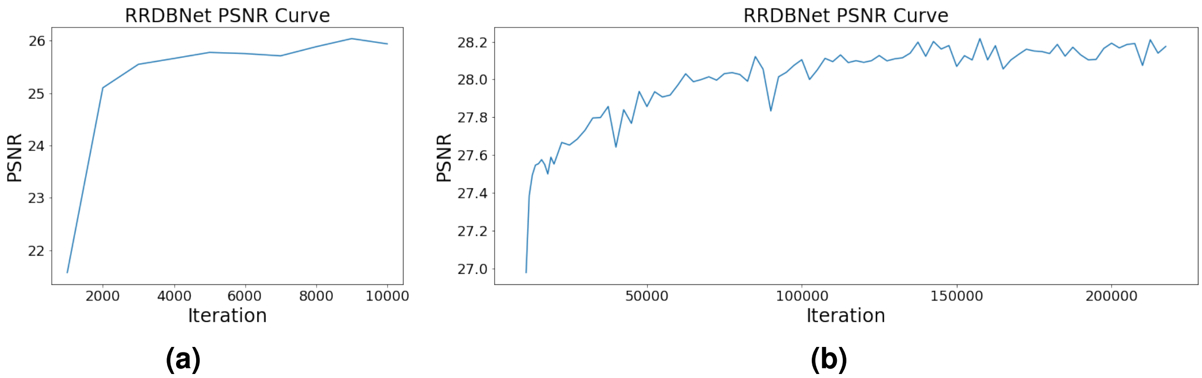


Figure 4: PSNR curves (RRDBNet): (a) First 10k iterations. (b) 10k to 218k iterations.

4 Analysis and Discussion

Due to the much smaller model capacity of RRDBNet, the PSNR score is much lower than that from SRResNet after 200k iterations. We note that in the original configuration, the number of iterations is set at 1000k. For RRDBNet, the original configuration uses cosine annealed restart but we used the simpler step schedule. PSNR was unable to increase after 150k iterations. Perhaps cosine annealed restart would have helped.

For SRResNet, PSNR scores increased steadily until training reached 150k iterations, which is the period set for cosine annealed restart. From 150k iterations, PSNR dropped sharply before increasing again. After a further 50k iterations, PSNR recovered to 150k levels and increased further, albeit very slowly. The loss was much more noisy after the restart at 150k iterations.

We note that while the model is able to restore most of the validation and test images well, a few images have poor results (e.g., Figure 5). As expected, better quality LQ images lead to more accurate HQ images (e.g., Figure 6).

Figure 5: Validation images: (a) LQ image (b) GT image (c) HQ image from SRResNet.

(a) **(b)** **(c)**

Figure 6: Validation images: (a) LQ image (b) GT image (c) HQ from SRResNet.

5 Best Model and Scores

5.1 Best Model and Test PSNR Score

The best PSNR score on the test set is 28.74, from the SRResNet model at 285k iterations. The result on CodaLab is shown in Figure 7:

37	28.7405838681	srresnet_285k.zip	04/22/2022 12:36:26	104255	Finished	✔	+

Figure 7: CodaLab screenshot of the best submission.

5.2 Validation PSNR Scores

The best PSNR scores on the validation set for each model are as follows:

- SRResNet (at 285k iterations): 28.92635
- RRDBNet (at 157.5k iterations): 28.21615

5.3 Links to Checkpoint File and High Quality Images of the Best Model

The checkpoint file for the best model can be downloaded from:
https://drive.google.com/file/d/1Ffcq7V21e2C4iK5Kb1q1Qi_TSaHVTiTV/view?usp=sharing

The HQ images from the best model can be downloaded from:
https://drive.google.com/file/d/1cUQm1J2kTorAcsqoXzRvMB75HSZrKVrn/view?usp=sharing

References

[1] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 1

[2] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, and Zehan Wang. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2016. 1, 2

[3] MMEditing Contributors. MMEditing: OpenMMLab image and video editing toolbox. https://github.com/open-mmlab/mmediting, 2022. 1

[4] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Yu Qiao, and Chen Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European Conference on Computer Vision Workshops(ECCVW)*, pages 0–0, 2018. 1