Experiment 3: Training GAN for Super-Resolution Task

I. Experiment Introduction

In this experiment we need to learn how to train a GAN network for dataset **ParisStreet** and fulfill the task of super resolution. The given **ParisStreet** dataset consists of 10000 training images and 1000 test images, each of which is 256*256 size with R, G, B channels. Therefore, we need to downsample the images before training.

In this experiment I still choose CoLaboratory (CoLab) developed by Google to train networks. Since I have described the details of CoLab and how to deploy it, I will straightly start with the networks used this time.

II. Model Implementation

2.1 ESRGAN

ESRGAN[1] is an improved model for SRGAN[2] (Super-Resolution Generative Adversarial Networks). The SRGAN is a seminal work that is capable of generating realistic textures during single image super-resolution.

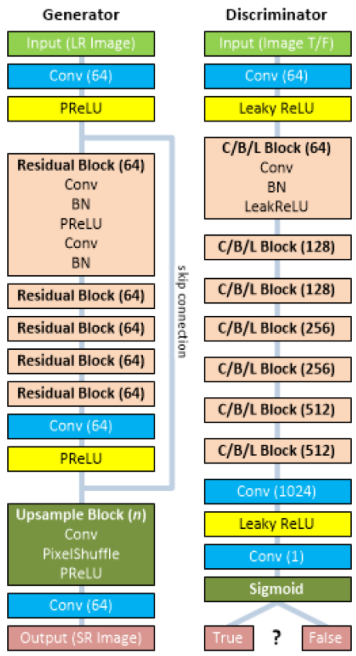

Fig.1 Architecture of SRGAN

To further enhance the visual quality, authors of ESRGAN thoroughly study three key components of SRGAN network architecture, adversarial loss and perceptual loss, and improve each of them. In particular, they introduce the Residual-in-Residual Dense Block (RRDB) without batch normalization as the basic network building unit. Moreover, they borrow the idea from relativistic GAN to let the discriminator predict relative realness instead of the absolute value. Finally, they improve the perceptual loss by using the features before activation, which could provide stronger supervision for brightness consistency and texture recovery.

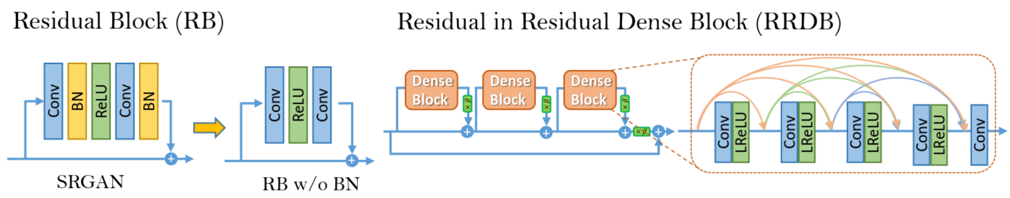

Fig.2 Architecture of ESRGAN

2.2 RRDB

ESRGAN keeps the high-level architecture design of SRGAN (as shown in Fig.1), and uses a novel basic block Residual-in-Residual Dense Block (RRDB) as depicted in Fig.2. Based on the observation that more layers and connections could always boost performance, the proposed RRDB employs a deeper and more complex structure than the original residual block in SRGAN. Specifically, the proposed RRDB has a residual-in-residual structure, where residual learning is used in different levels.

In this experiment, I used two versions of ESRGAN. One is the model presented in [1], the other[3] is the finally updated version that has better performance in PSNR comparison.

III. Results and Evaluation

Before analyzing the experiment results and evaluating the model performance, I have to claim that since I have limited computing resources (CoLab can be used to train networks no more than 12 hours at a time and there could be unpredictable disruption due to connecting problems, which really made me upset), I can only finish training of 10 epochs of ESRGAN at a time. Therefore, the following analysis will be based on the model of ESRGAN with only 10-epoch training and the pre-trained ESRGAN model in [3]. I choose PSNR as the indicator of the recovering quality of pictures achieved by models in this super-resolution task

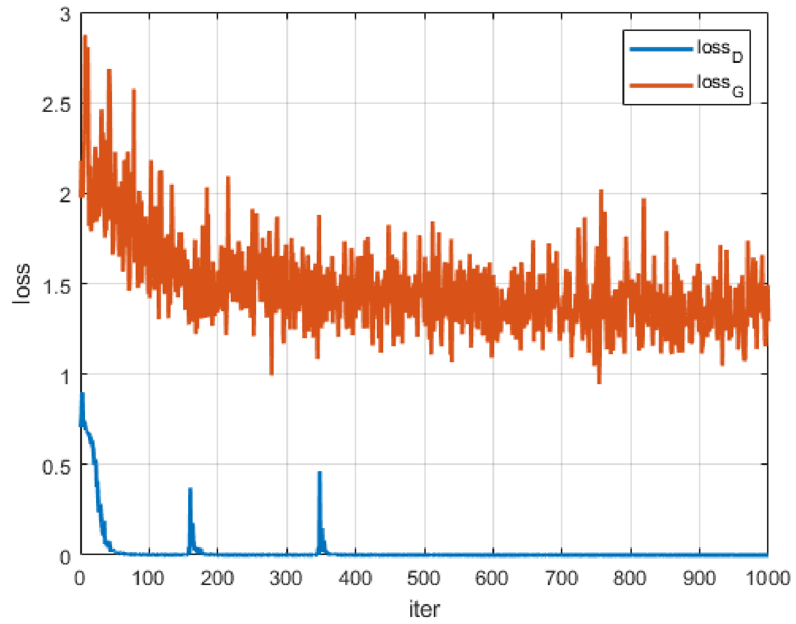

Fig.3 Loss of Discriminator and Generator in epoch 1~1000

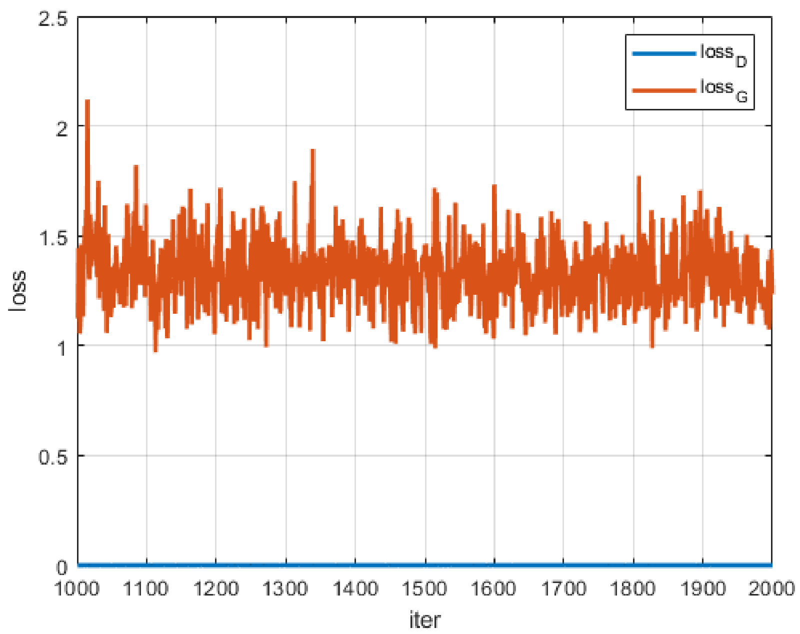

Fig.4 Loss of Discriminator and Generator in epoch 1001~2000

From above pictures we can find that after some iterations of training, the situation became very obvious, the discriminator outperformed than the generator. This means that the Generator cannot hoodwink the discriminator, and this means that the networks might be trapped in a local minimal point. Hence, we need to change the hyper-parameters and retrain the network. Otherwise, the network is less likely to get out of this point only by us adding the number of training epoch.

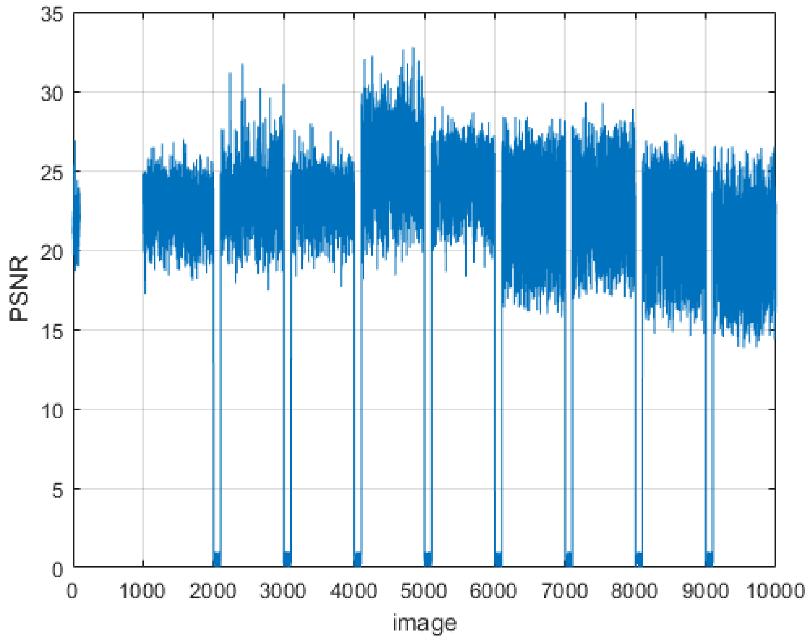

Fig.5 PSNR in training process

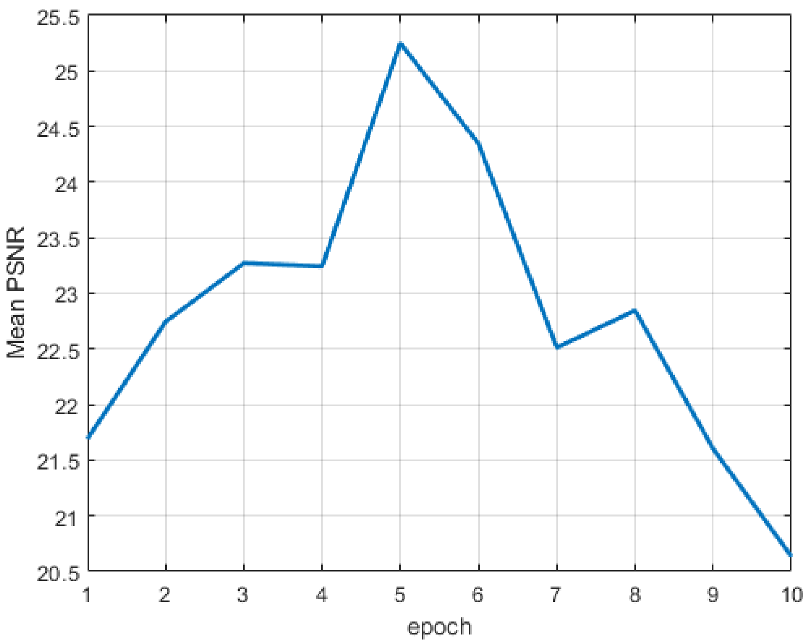

Fig.6 Mean PSNR of each epoch in training process

Using PSNR as the indicator of the quality of recovered super-resolution image, we can see that obviously 10 epochs of training is not satisfied for this task. There are, of course, the PSNR value of some super-resolution images can reach as high as over 30, but there are also images have lower PSNR value of 15. The zeros are because the training settings, not the bad performance of ESRGAN.

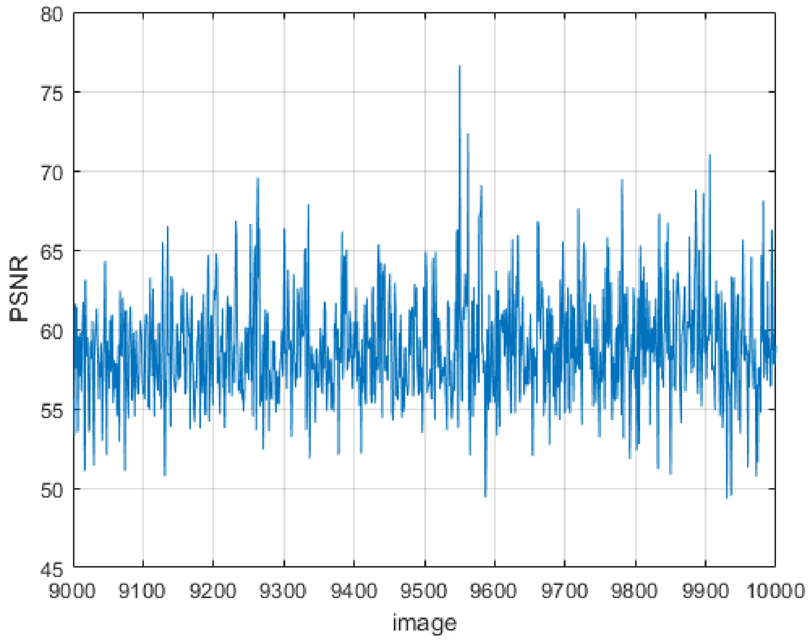

Fig.7 PSNR of training images (part of training set)

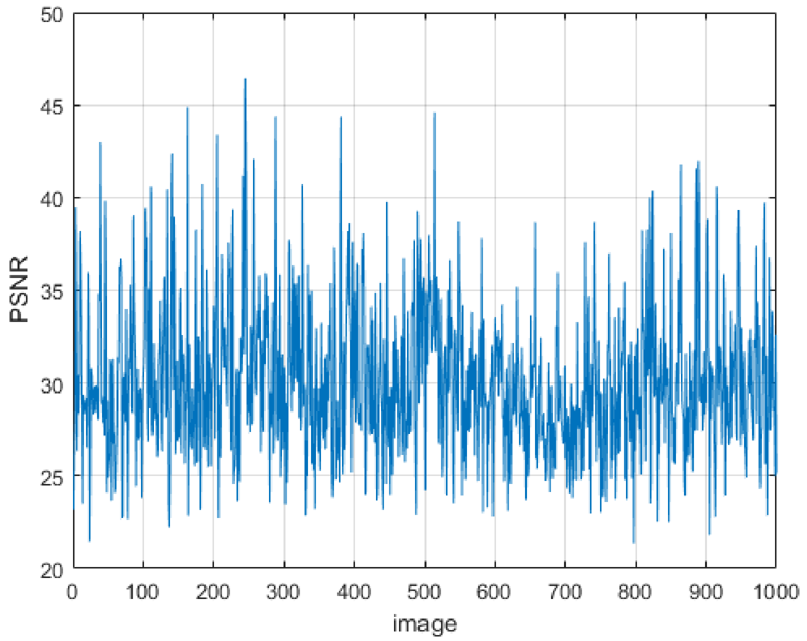

Fig.8 PSNR of test images

Above two figures show the PSNR of super-resolution pictures generated by the second model of ESRGAN I implemented in this experiment and can be found in fold /exp3/ESRGAN_PSNR. We can see that the PSNR values are obviously higher than that of the first model with an average around 60 in training set and 30 in test set. This is because the model implemented here is a fully-trained model in several datasets not just in the given **ParisStreet** dataset, which is the key factor in this experiment.

Here are some comparisons of low-resolution pictures and super-resolution pictures generated by ESRGAN.

Fig.9 Comparison of LR and SR images

IV. Conclusions

In this experiment I learned about the basic principals of GAN and implemented the model of ESRGAN for the super-resolution task. The idea of letting generator and discriminator fight with each other during training and finally improve the performance of certain network is really marvelous!

In order to fulfill the task with better performance, I read some papers of different models of GAN and gained more knowledge of this kind of networks, which really helped me during training the network.

However, the deepest impression of GAN I had during this experiment is that this kind of networks are too hard to train. In last experiment, I can finish 300-epoch training of ResNet-18 on CoLab in 5 hours. Now, it took me 12 hours and only finished 10 epochs of training for ESRGAN. This is a game for players with abundant computing resources, people like me cannot even try some ideas of improving the network because of limited computing resources. Anyway, I still learned much from this experiment.

This is the last experiment of this Pattern Recognition course. Though we had this course all online with Zoom, I felt no difference with studying in classroom. Looking back at the 16-week of studying, I built a systematic structure of pattern recognition gradually and implemented many algorithms, which in turn deepened my understanding of the theories.

Thanks for our dedicated lecturer, Prof. Guo Zhenhua and our responsible teaching assistant Chen Shengjie. Thank you for your preparation and dedication to this course!

Reference

[1]. Wang X, Yu K, Wu S, et al. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks[C]. european conference on computer vision, 2018: 63-79.

[2]. Takano N, Alaghband G. SRGAN: Training Dataset Matters[J]. arXiv: Computer Vision and Pattern Recognition, 2019.

[3]. ECCV18 Workshops - Enhanced SRGAN. Champion PIRM Challenge on Perceptual Super-Resolution (Third Region) https://github.com/xinntao/ESRGAN