Term Project: Super Resolution

Restore noisy images Using Real-ESRGAN

110611043 林揚森, 110705013沈昱宏

Abstract

Our main method of restoring the noisy image is using the model "Real-ESRGAN" (for super-resolution). The pre-trained model works perfectly on objects with distinct edges (such as coffee cups or animation characters). However, the model is not effective enough to restore the images of human faces and animals (such as pandas). Therefore, we use the built-in function, GFPGAN, to enhance human face restoration and fine-tune the model to improve the restoration of panda images.

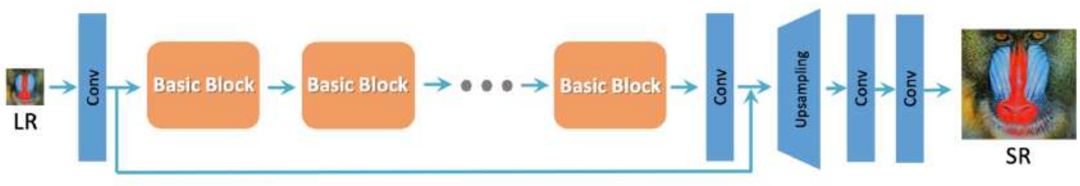
Methods

- ESRGAN (Enhanced Super resolution Generative Adversarial Networks)
- Real-ESRGAN

ESRGAN

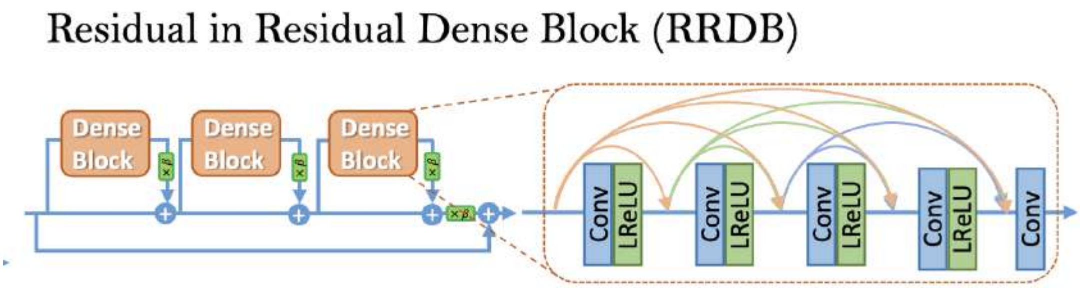
a. **Structure**:

ESRGAN uses the basic architecture of SRRedNet. We could select or design the "Basic Blocks" (e.g. residual block, dense block, RRDB) for better performance.



b. **Basic Blocks:**

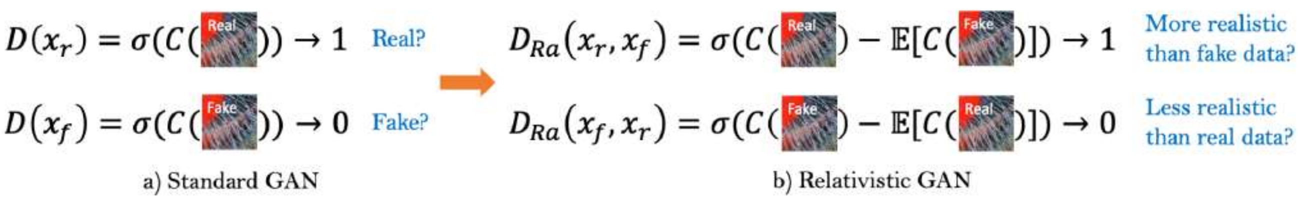
The model uses the "**Residual-in-Residual Dense Block (RRDB)**" as Basic Blocks, which combines multi-level residual network and dense connections as depicted in the following figure.



c. Relativistic Discriminator

In the GAN model, the discriminator works as the "judge", which is used to choose which model to update. Rather than the standard discriminator, they use the **relativistic discriminator** to improve the performance of ESRGAN. The Relativistic Discriminator is deciding which data is "more" realistic, rather than choosing which one is "fake".

The Discriminator is expressed as:



The adversarial loss for generator is in a symmetrical form is :

$$L_G^{Ra} = -\mathbb{E}_{x_r}[\log(1 - D_{Ra}(x_r, x_f))] - \mathbb{E}_{x_f}[\log(D_{Ra}(x_f, x_r))],$$

d. Network Interpolation

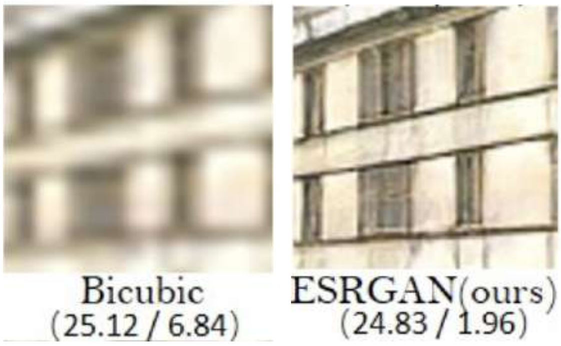
ESRGAN uses network interpolation to remove the unpleasant noise in GAN-based methods. They use two networks: PSNR-oriented network "G_{PSNR}" and Gan-based network "G_{GAN}" to derive an interpolated model "G_{INTERP}".

The equation of their parameters:

$$\theta_G^{\text{INTERP}} = (1 - \alpha)\ \theta_G^{\text{PSNR}} + \alpha\ \theta_G^{\text{GAN}}$$

e. Results:

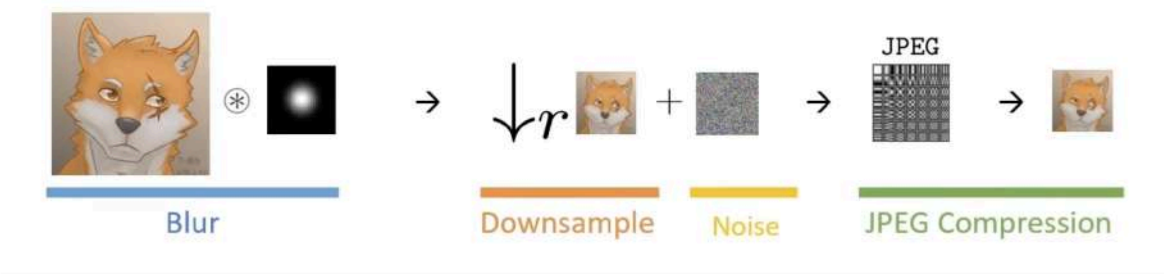
Blurred ⇔ ESRGAN



Real-ESRGAN

a. Why Real-ESRGAN?

As the synthetic images only contain few degradation processes (see the following figure), ESRGAN works great on animated images or vector images etc.

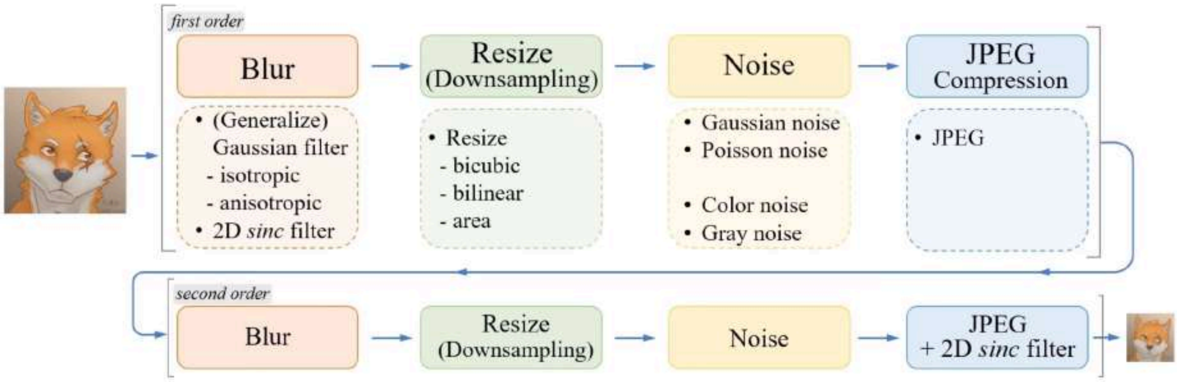


However, the noises in the real world are much more complicated than the synthetic ones (see the following figure), which means that there's still space for ESRGAN to improve. Real-ESRGAN is the improved version of ESRGAN, which is targeted to improve ESRGAN's performance on real world images.

b. Improvements in Real-ESRGAN

Degradation process of training data:

Real-ESRGAN adopts the 2-nd order degradation to reach the balance of simplicity and effectiveness. Each order includes 4 steps (see following figure).



c. Results

Noisy → Real-ESRGAN

Innovation and Improvement

- GFPGAN (face enhancement)
- fine-tune

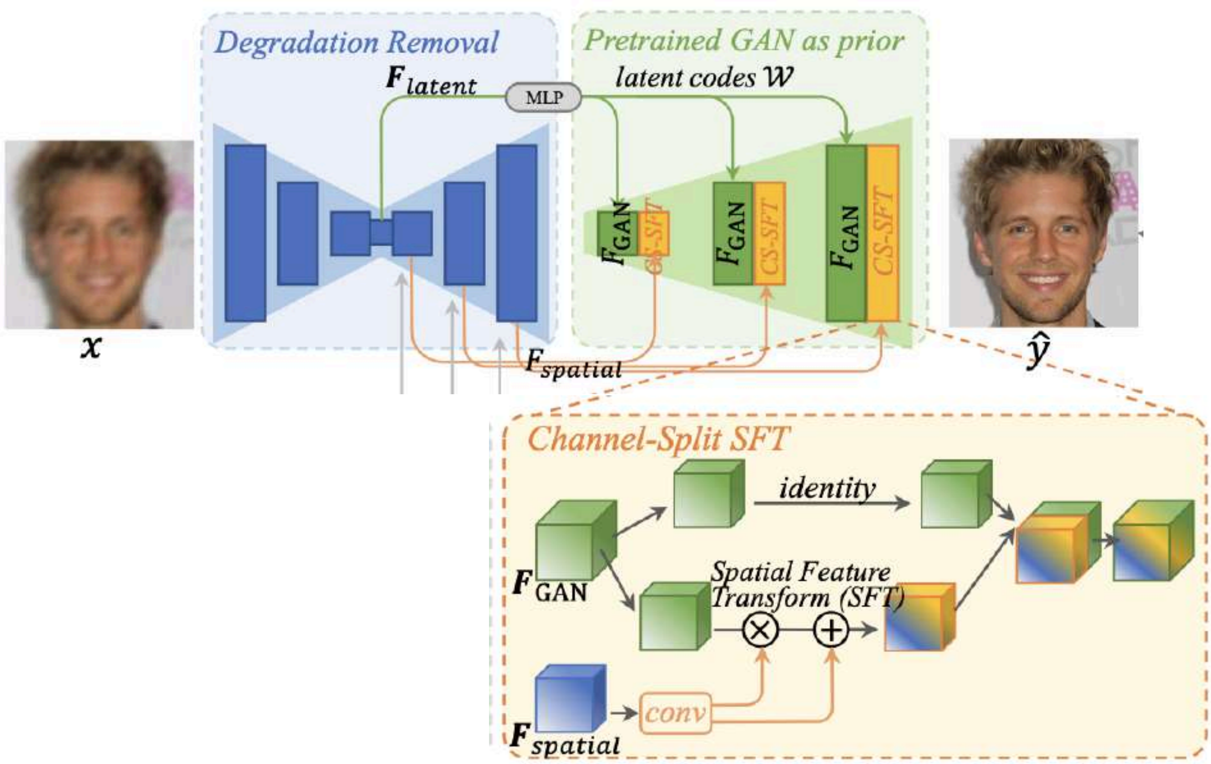
GFPGAN (Generative Facial Prior GAN)

a. Why GFPGAN

Face restoration usually highly relies on facial geometry prior, the restoring performance of ESRGAN is limited by the lack of facial geometry prior. To improve the performance of facial restoration, GFPGAN leverages rich and diverse priors encapsulated in a pre-trained face GAN for blind face restoration.

b. Structure of GFPGAN

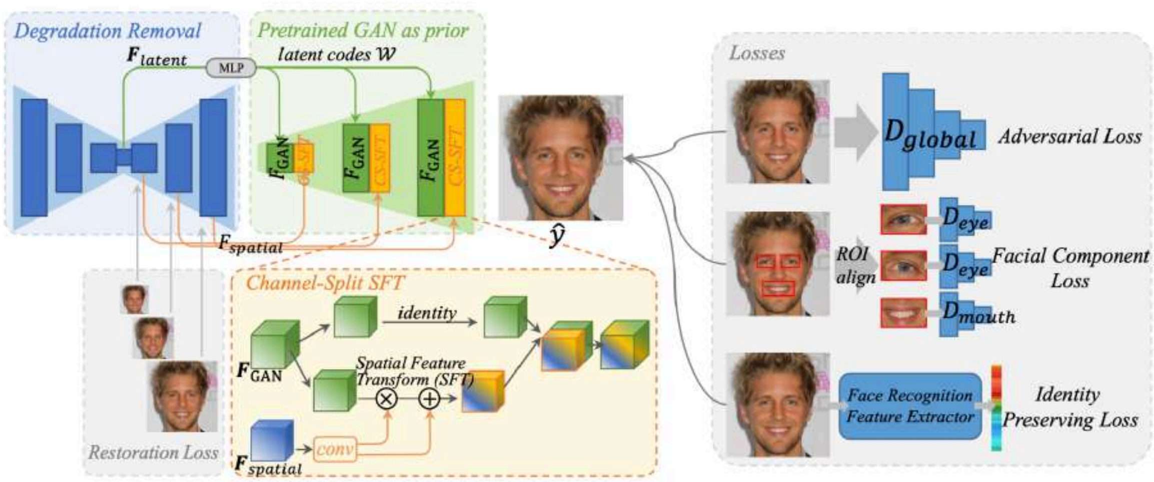
GFPGAN mainly consists of two parts. One is a U-net (degradation removal module), and another is a pre-trained face GAN as prior.
These two components are bridged by a latent code mapping and several CS-SFT(Channel-Split Spatial Feature Transform) layers.



There Are three main mathematical techniques that GFPGAN adopted during the training process.

1. **Intermediate restoration losses**: to remove complex degradation
2. **Facial component loss with discriminators**: to enhance facial details
3. **Identity preserving loss**: to retain face identity.

The role of each technique is shown in the following figure (two gray parts in the following figure).



c. Results

Noisy image ➡ Real-ESRGAN ➡ **Real-ESRGAN + GFPGAN**

Fine-tune the ESRGAN model

a. Fine-tune the pre-trained model

Fine-tuning the pre-trained model mainly consists of three steps.

1. First, prepare the dataset.

We downloaded the panda dataset from kaggle and generated the paired data that follow the methods we've mentioned in Real-ESRGAN. (There's open source code the author of Real-ESRGAN used to generate the degraded data on github.)

2. Download the pre-train model.
3. Finetune

b. Results

Noisy image ➡ Real-ESRGAN ➡ **Fine-tuned Real-ESRGAN**

References

- ESRGAN:
  - Paper: https://arxiv.org/pdf/1809.00219
  - Github: https://github.com/xinntao/ESRGAN?tab=readme-ov-file
- Real-ESRGAN
  - Github: https://github.com/xinntao/Real-ESRGAN-ncnn-vulkan#computer-usages
- GFPGAN
  - Paper: https://arxiv.org/pdf/2101.04061
  - Github: https://github.com/TencentARC/GFPGAN
- dataset (panda)
  - Kaggle: https://www.kaggle.com/datasets/ashishsaxena2209/animal-image-datasetdog-cat-and-panda