Ultra-Realistic: Generating Ultra-realistic Super-resolution Imageswith A-ESRGAN

Dushyant Singh Udawat (ds35) , Christopher Cai (cdcai2) , Pratikshit Singh (ps71)

Abstract

In our project, we are planning to implement A-ESRGAN, which is the state of the art GAN model for blind image super-resolution. This version of SR is built upon improvements over previous models such as SRGAN, ESRGAN, and RealSRGAN. It modifies the discriminator used in the aforementioned models by using a multiscale attention U-net discriminator that helps the generator generate more detailed images and reduce the blurring of edges apparent in the previous models. We will use the DIV2K dataset, which contains 800 images for training, 100 images for validation, and 100 images for testing. We also plan to test our implementation on two other datasets (more information in the datasets heading) and compare our results with those of other popular super-resolution models such as **SRGAN, ESRGAN, and Real-ESRGAN on** non-reference natural image quality evaluator(NIQE) [2] metric.

Change of Project Objectives

We believe that we are on track to complete our current objectives.

Current Efforts

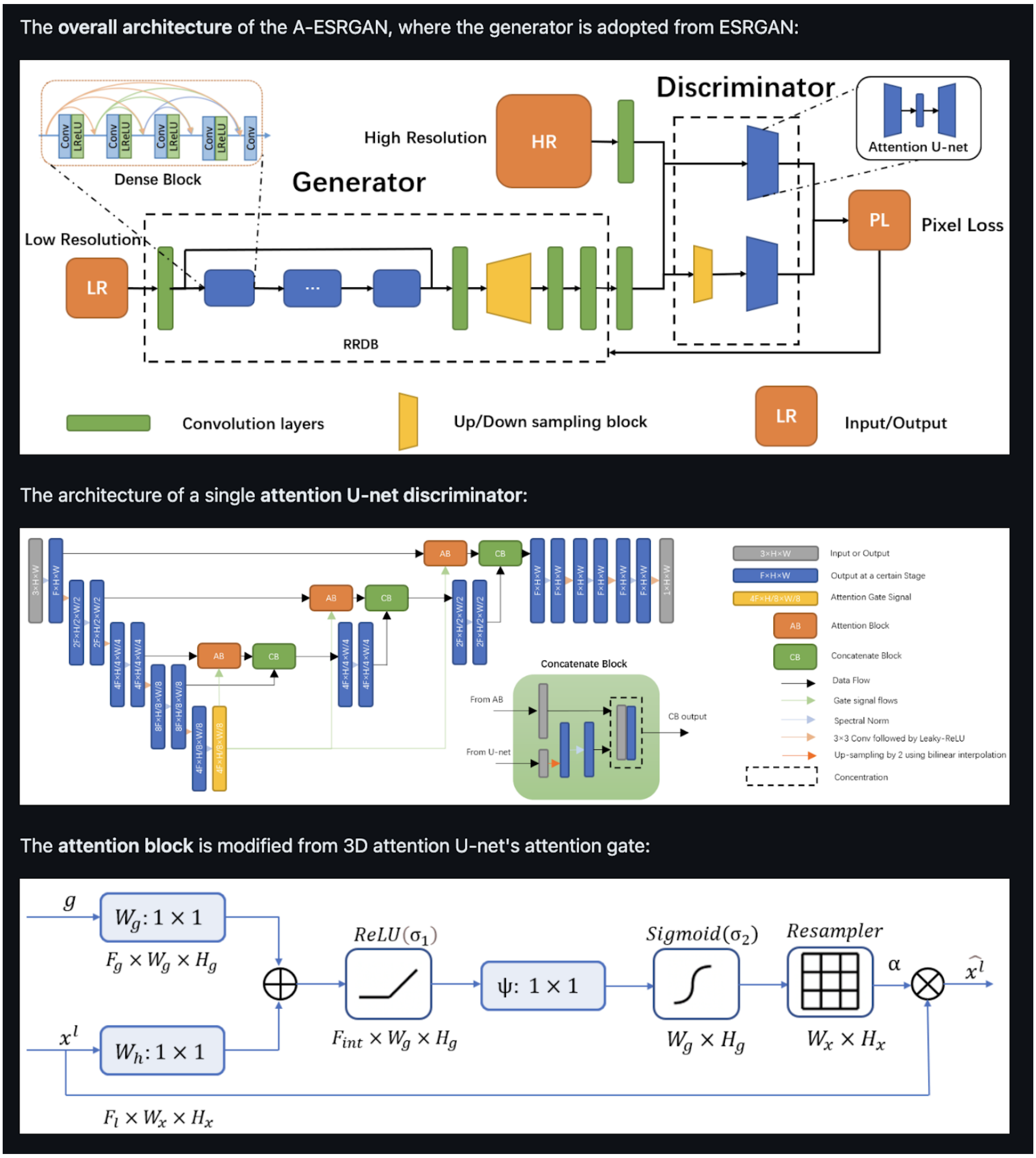
1. **Data-Loader Prepared**

   We were able to download the dataset used in the paper (DIV2K) which we plan to train the model on, but we have not yet decided on which other datasets we will use to further evaluate the model. We have also designed a dataloader that we will use in the train pipeline to feed the images to the model. The dataloader will scan through a provided directory and load the file paths for each image in the directory.

   When retrieving an image, the dataloader will load the image from the file and process it to fit the degradation process. It will then perform the 2-stage degradations as performed in the paper to resemble real-world image degradation. First, the image is sharpened through USM sharpening to create the ground truth image. The sharpened image is then randomly rescaled and has either gaussian or poisson (determined randomly) noise added to it. The image is then passed through JPEG compression and the full degradation process is repeated a second time. In our implementation, we do not include the final sinc filter that the authors use.

2. **Generator and Discriminator Models Defined.**

   We have designed the generator and discriminative models for training and testing purposes. We have made the train pipeline for our model and trained it on a small dataset. We have gotten rid of all the errors and bugs but we are still not getting good results. The reason for that is that the data we have trained on is too small. We will increase the data size to contain a wider variety of images and a larger number of images. The architectures used for the discriminator and generator model are based on the ones that have been described in the paper. Their pictorial representation is as described below-



3. Running the baseline model on images of different degradations.

For this step, we degraded the images using our degradation function and passed them through the pre-trained A-ESRGAN model. The result can be seen in the images below.

Instance 1 -

Ground Truth -

Degraded image -

A-ESRGAN Output -

Instance 2 -

Ground Truth -

Degraded Image -

A-ESRGAN Output -