Ultra-Realistic: Generating Ultra-realistic Super-resolution Images with A-ESRGAN

Pratikshit Singh
ps71@illinois.edu

Dushyant Singh Udawat
ds35@illinois.edu

Christopher Cai
cdcai2@illinois.edu

Problem Statement: The objective of this project is 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[3] on non-reference natural image quality evaluator(NIQE)[5] metric.

Figure 1: Left: Low resolution original image; Right: Super-resolution image using our model

This report is presented for the final project
of Deep learning for Computer Vision(CS444)

Department of Computer Science
University of Illinois Urbana-Champaign
United States of America
May 10, 2023

1 Introduction

The field of computer vision has a persistent challenge known as blind image super-resolution (SR), which involves the enhancement of low-resolution (LR) images that have been subjected to intricate and unknown distortions. More specifically, super resolution refers to the process of enhancing the resolution and quality of images and videos beyond their original resolution. A-ESRGAN, or Attention-Enhanced Super Resolution Generative Adversarial Network, is a state-of-the-art deep learning architecture that has shown remarkable performance in super resolution tasks. In this graduate student course project, we will explore the details of A-ESRGAN and its implementation for super resolution tasks. Specifically, we will analyze the architecture of A-ESRGAN, compare its performance with other state-of-the-art methods, and examine the factors that affect its performance. Moreover, we will experiment with A-ESRGAN on various datasets and evaluate its performance on different image resolutions and quality levels. Through this project, we aim to gain a deeper understanding of A-ESRGAN and contribute to the development of more effective super resolution techniques.

A-ESRGAN is based on previous work on Basic SRGAN[1] and ESRGAN[4]. The work also depicts the use of U-net as GAN models and thus utilizes the power of U-nets as well for producing better results. GAN based approach for super-resolution has been proved to generate better results than deep-learning networks like convolution based approaches. Deep learning based networks while are able to generate images with high especially in Peak Signal-to-Noise Ratio (PSNR) value are known to lose high frequency details leading to over smoothen results. Generative models show that they do not suffer from this issue when used for super-resolution.

A Generative Adversarial Network (GAN) for super-resolution consists of two interconnected networks: a generator and a discriminator. The generator receives low-resolution (LR) images as its input and works to produce high-resolution (HR) images that are as close as possible to the original image. Meanwhile, the discriminator's task is to differentiate between the HR images and the artificially generated "fake" images produced by the generator.

Some previous work on super-resolution using GANs like ESRGAN(Enhanced Super-Resolution Generative Adversarial Networks) uses a GAN-based architecture. It introduces a novel residual-in-residual dense block and a perceptual loss function to enhance image quality, was state-of-the art before A-ESRGAN. Other work like RealSR, BSRGAN, Real-ESRGAN nearly utilizes the same architecture and were known to be state-of-the art before their successors for few datasets. Below we provide a figure for visual comparision, from the original paper[5], between these various kind of models that were prevalent.

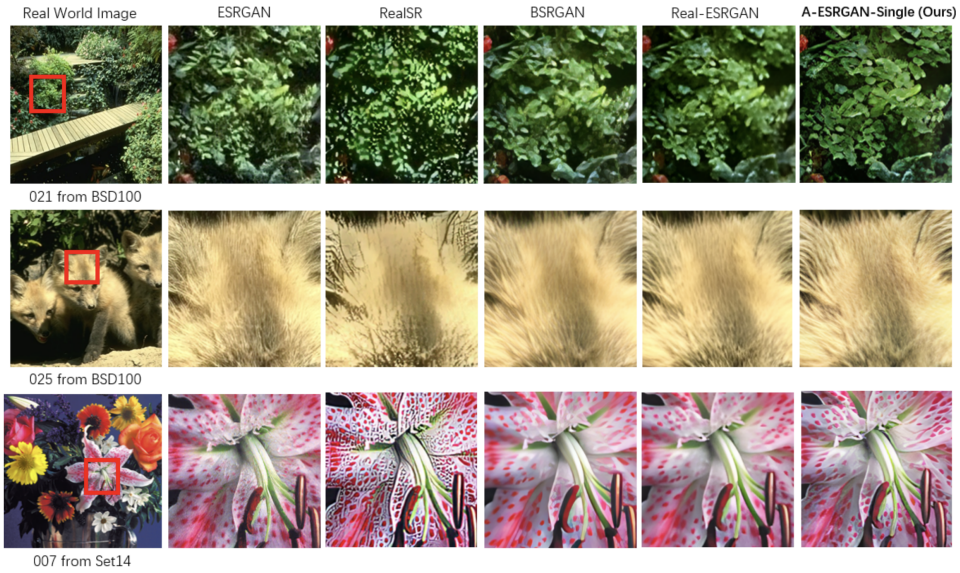

Figure 2: Visual comparison of A-ESRGAN with other ×4 super-resolution methods. Zoom in for the best view.

2 Details of the approach

To implement A-ESRGAN, we followed the architecture described in the original paper (Fig. 3). For the generator, we stacked 23 residual-in-residual dense blocks, as seen in the authors' implementation of the A-ESRGAN generator. For the discriminator, we used the single U-net encoder decoder structure from the original A-ESRGAN paper with added attention blocks (Fig. 4). We used the same loss functions—binary cross-entropy (BCE) loss, L1 loss, and perceptual loss—as in the A-ESRGAN implementation to more closely replicate the results.

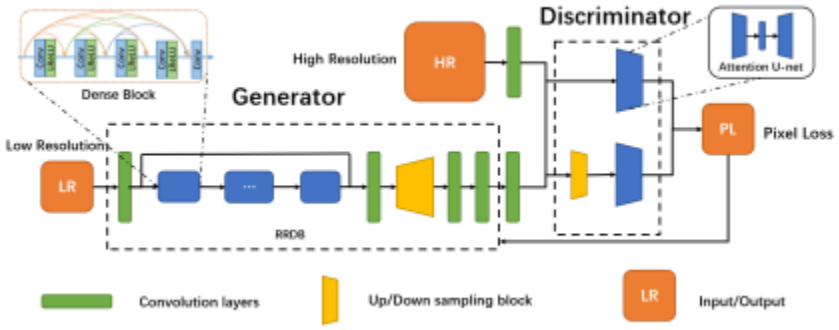

Figure 3: Overall A-ESRGAN architecture

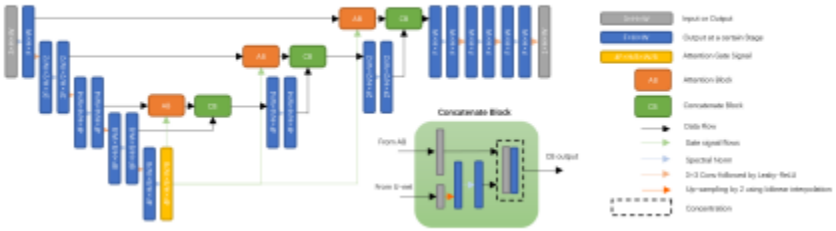

Figure 4: U-NET architecture of the A-ESRGAN discriminator with attention blocks

For the training of the model, we implemented our own training loop and PyTorch dataset to help with transforming the images. Our dataset iterated through a folder with our training data and loaded the path to each image. On image retrieval, we perform a random horizontal flip and crop/pad the data to be 400x400. We then perform the degradation process on the augmented image. We sharpen the image with USM sharpening to get out high resolution ground truth. We then blur the image with one of 6 randomly chosen kernels or a sinc kernel. The image is randomly resized, has random gaussian or poisson noise added to it, and is passed through a JPEG compressor. The process is repeated a second time but with a random chance to perform the blur. Finally, we pass the image through a final sinc filter, compress it and crop the image to be 256x256.

In our training loop, we used the Adam optimizer at a learning rate of 10^{-4} with default parameters and a batch size of 4. We also used the same weights for the L1, perceptual and BCE loss (1, 1,0.1). The original paper trained the model for $400K$ iterations on 4 GPUS (one NVIDIA A100 and three NVIDIA A40s) which we did not have the resources to replicate. As such, we trained our model for 25 and 120 iterations on a single GPU through Google Colab. We used the DIV2k data set for training, which contains 800 images for training and 100 for validation. To test performance, we also evaluated our trained model on the Set14 and Urban100 datasets which contain 14 and 100 images respectively.

Figure 5: Example images of degradation process. Top is the untouched image, left is the sharpened patched used as the ground truth for training, and right is the patch after going through the degradation process

3 Results

To evaluate the model's performance, we calculated the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) between the super resolution image and ground truth and averaged it over the entire set of validation images.

Table 1: Comparison of Different Models

NIQE	Bicubic	ESRGAN	BSRGAN	RealESRGAN	RealSR	A-ESRGAN	A-ESRGAN-Ours
Set5	7.8524	5.6712	4.5806	4.8629	3.5064	3.9125	-
Set14	7.5593	5.0363	4.4096	4.4978	3.5413	3.4983	3.7691
BSD100	7.3413	3.1544	3.8172	3.9826	3.6916	3.2948	-
Sun-Hays80	7.6496	3.6639	3.5609	2.9540	3.3109	2.6664	-
Urban100	7.1089	3.1074	4.1996	4.0950	3.929	3.4728	3.3793

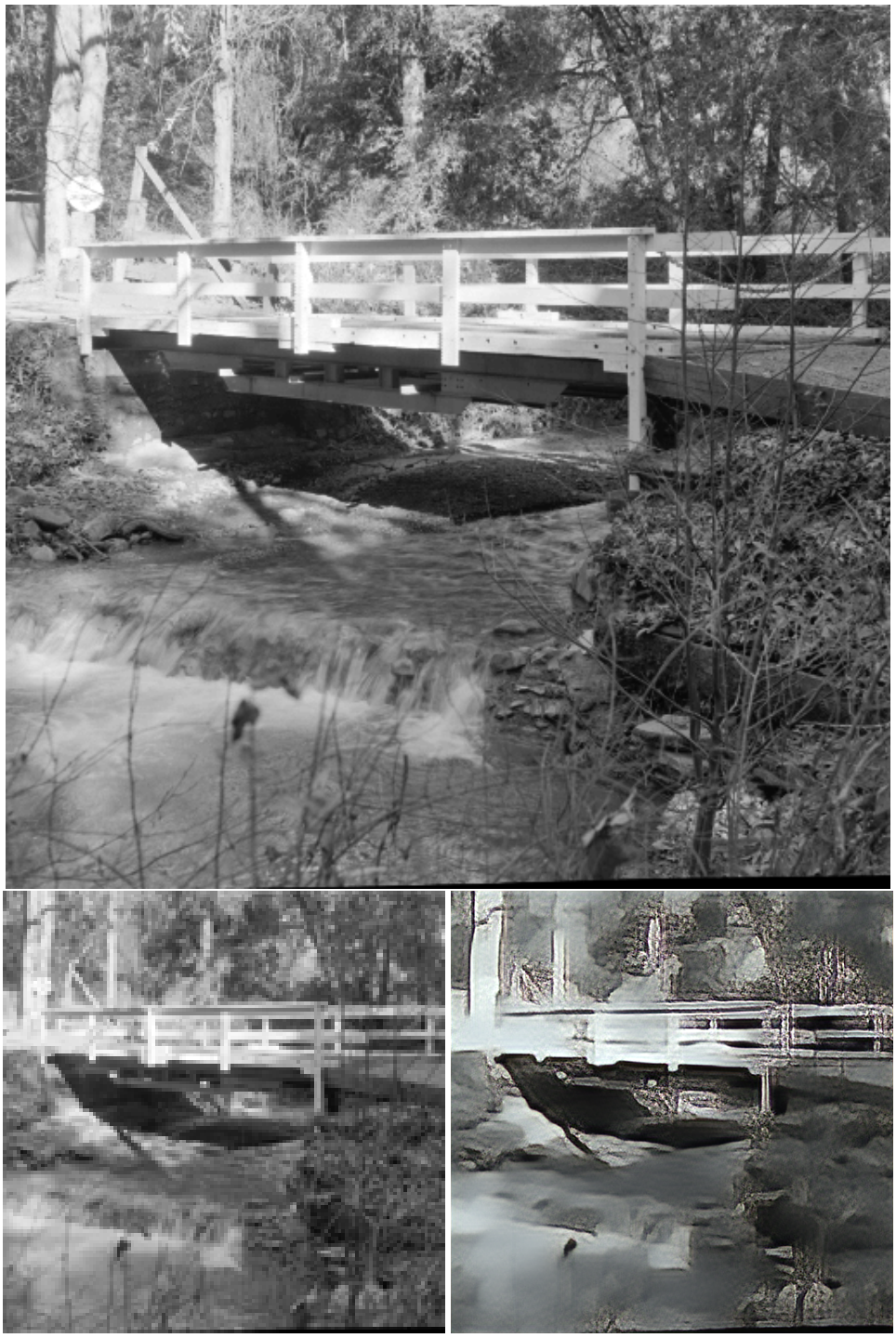

Figure 6: Result of our trained model on the Set14 dataset. The image on top is the ground truth, the image on the left is the lower resolution image, and the image on the right is the super resolution image

4 Discussion

Speaking of the image quality, the generated image do not show a high quality representation of low quality images. We see loss of detailed features, specifically near the edge lines. The results look

Figure 7: Result of our trained model on the Urban100 dataset. The image on top is the ground truth, the image on the left is the lower resolution image, and the image on the right is the super resolution image

like they are higher quality images from a distance, perceptually. but when taking a closer look, they do not contain as much information as the ground truth high quality image. This is slightly lower performance than what the original authors of the paper had presented. we believe that the reason for this low performance is -

- Insufficient training: we train our model for 120 epochs in total. This is less than what it would have required to achieve an optimal quality on the images. However, our models stopped giving any better accuracy after about 25 epochs, and we thought it best to stop training.
- Better initialization: we could have better initialized our model rather than random initialization, because it is a huge model and it can diverge very quickly to local optimums. The original paper does not have details of initializations. In retrospect we could have searched for a proper initialization technique for our application.[6]

Speaking of NIQE scores, I believe that we got almost the same score as the original papers. However, since we know that the quality of the images were not capturing fine details, we feel that there is a need to explain this. A few observations and explanations for this Behavior are as follows -

- Pitfalls of NIQE scores - There are some pitfalls of NIQE scores, Namely data set bias and limited generalization . NIQE scores also do not accurately assess image quality in presence of artifacts because they have been trained on natural images. These scores are highly dependent upon the data set that have been used to train and develop the model and

maybe Set14 and Urban100 do not have the same characteristics as the training data of NIQE scores. The solution to this problem is that we should not focus much on NIQE scores and rather qualitatively just the outputs of the model.[2]

5 Conclusion

In conclusion, we were able to train a super resolution model, based on the state of the art A-ESRGAN model. While our image quality was not up to the standard, our NIQE scores were still high. We achieved better scores than any other non state-of-the-art technique (excluding A-ESRGAN). We discussed the two anomalies and their potential solutions in the obtained results in the discussion section.

6 Statement of individual contribution:

Below are the contribution of individual team members in the project. Though these are the highlighted one, the team worked together and helped each other when needed to successfully accomplish the project goals.

Chris Cai

- Wrote PyTorch dataset for loading/transforming DIV2k dataset
- Wrote training/evaluation loop for A-ESRGAN, Trained the model, Tuned the model

Pratikshit Singh

- Wrote the Discriminator model and loading script.
- Developed the loss function needed to train generator and discriminator with.
- Project setup on Github and report setup.

Dushyant Singh Udawat

- Wrote the Generator model and loading script
- Setup the Github Repo structure, Trained the model.
- Conducted tests to find the NIQE scores of model over two datasets.

References

[1] Chao Dong, Chen Change Loy, and Kaiming He. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[2] Amir Husain and Michael J. Black. The pitfalls of niqe scores for image quality assessment. *IEEE transactions on image processing*, 22(11):4409–4422, 2013.

[3] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. *arXiv preprint arXiv:2107.10833*, 2021.

[4] Xudong Wang, Chao Liu, Evan Shelhamer, and Dong Yu. Esrgan: Enhanced super-resolution generative adversarial networks. *arXiv preprint arXiv:1809.00219*, 2018.

[5] Zihao Wei, Yidong Huang, Yuang Chen, Chenhao Zheng, and Jinnan Gao. A-esrgan: Training real-world blind super-resolution with attention u-net discriminators, 2021.

[6] Kilian Weinberger, Ilya Sutskever, and Ruslan Salakhutdinov. Initialization methods for deep learning. *arXiv preprint arXiv:1312.6184*, 2013.