CV Super Resolution

Om Bhojraj
bhojraj@usc.edu

Shyam Sunder Rajasekaran
srajasek@usc.edu

***Abstract*—The Super-Resolution Generative Adversarial Network (SRGAN) and the Super-Resolution Convolutional Neural Network (SRCNN) are two modern super-resolution (SR) algorithms that are compared in this study. SRCNN emphasizes architectural decisions by mapping low-resolution to high-resolution pictures using deep convolutional neural networks. Through adversarial training, SRGAN generates realistic high-resolution images using generative adversarial networks (GANs).**

We present a thorough comparative study between SRCNN and SRGAN, including both qualitative and quantitative measures (PSNR, SSIM). We examine computational requirements, benchmark performance, and architectural features. By shedding light on the advantages and disadvantages of various SR techniques, the research seeks to assist practitioners in choosing the best one for their particular set of circumstances.

***Index Terms*—SRCNN, SRGAN, PSNR, SSIM, neural networks**

I. Introduction

The pursuit of higher visual clarity and finer detail in photographs has prompted research into image upsampling, which is a crucial procedure for artificially increasing an image's spatial resolution or the number of pixels that contain the image's visual content. Neural super-resolution techniques are more sophisticated than typical upsampling methods, which are widely used when zooming into images on digital devices or watching content with various qualities. This explores these sophisticated techniques, emphasizing how they can produce high-fidelity approximations of images at increasing resolutions.

Neural super-resolution approaches are different from traditional upsampling methods in that they strive for something more than just pixel augmentation. Conventional upsampling methods primarily serve to accommodate images for display on screens or in print.

Fig. 1. Image Upscaling

When seeing images at resolutions higher than their native encoding, the limits of the usual approaches become evident, even though they might be adequate for common visualization purposes. The final photos frequently include blockiness or blurriness, which is a sign of a basic upscaling procedure done without a thorough comprehension of the underlying content as shown in Figure 1.

This neural super-resolution methodology explores the nuances of models such as the Super-Resolution Generative Adversarial Network (SRGAN) and the Super-Resolution Convolutional Neural Network (SRCNN). These sophisticated models, in contrast to popular upsampling techniques, are made to produce precise, thorough estimations of images at higher resolutions. The investigation includes a qualitative assessment by visually examining super-resolved images in addition to a quantitative assessment using metrics like peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM).

We compare the limits of traditional upsampling methods, which frequently produce low-resolution, low-detail images when intentionally viewed at higher resolutions, to highlight the relevance of these developments. The goal is to give practitioners a thorough grasp of brain super-resolution techniques, highlighting some of their possible uses and assisting them in applying these approaches to a variety of imaging circumstances. This study aims to provide important insights into the changing field of picture enhancement technology as we examine the intricacies of image upsampling.

II. Mini Literature Survey

In this survey, we delve into the realm of super-resolution (SR), a transformative process in computer vision that estimates high-resolution images from their low-resolution counterparts. The evolution of SR techniques, from traditional methods to advanced computational algorithms, highlights its significance in diverse applications like medical imaging and satellite imagery. A critical analysis of the peak signal-to-noise ratio (PSNR) in evaluating SR algorithms is presented. These metrics, though standard, fall short in capturing perceptually relevant details such as texture and sharpness, a gap that becomes evident through comparative visual examples. The survey further explores a range of previous SR approaches, including compressed sensing and convolutional sparse coding, providing insights into their methodologies and contextual relevance. A special focus is given to SRGAN, a state-of-the-art method introduced in [1] "Photo-realistic Single Image Super-Resolution Using a Generative Adversarial Network". This novel approach integrates content loss with adversarial

loss in a generative adversarial network framework, achieving more photo-realistic reconstructions for large upscaling factors. The survey also discusses the potential of shallower network architectures and the impact of ResNet designs on deeper networks. In conclusion, the survey not only provides a comprehensive overview of SR's challenges and advancements but also underscores the growing importance of perceptual quality in SR algorithms, suggesting promising directions for future research in this rapidly evolving field.

An extensive investigation of important topics in image processing and computer vision is provided in the paper [2] "Image Super-Resolution Using Deep Convolutional Networks" The first section of the paper delves further into the assessment of image quality, stressing the significance of structural similarity, defect visibility, and the creation of sophisticated measures such as the Multiscale Structural Similarity Index. It covers a wide range of topics related to super-resolution, from research on self-similarities and benchmarking techniques to novel methods such as fast super-resolution via in-place example regression, coupled dictionary training, and using sparse representations for image scale-up. Techniques for denoising are also covered, in particular the training of individual networks for varying noise levels. The research does not, however, go into great detail about these networks' intricacies. Another focus area is object detection, where the study investigates the use of deep convolutional neural networks and the use of super-resolution methods for improved object detection. The paper also discusses a wide range of topics, such as deep learning applications in face representation, databases for image segmentation evaluation, non-blind image deconvolution techniques, the use of rectified linear units in machine learning, simplification of convolutional neural networks for accelerated learning, and advanced methods for fast super-resolution. A rich tapestry of current research and methodology in image processing and computer vision is provided through this survey in "Image Super-Resolution Using Deep Convolutional Networks" demonstrating the breadth of the discipline and the cutting-edge approaches being pursued.

In the publication [3] important issues about image super-resolution (SISR) are addressed. It criticizes the shortcomings of conventional metrics, such as MSE and PSNR, in evaluating visual quality in SISR and argues in favor of more precise measurements that take perceptual picture quality into account. It also emphasizes how current SISR models, which are primarily trained on synthetic data, have limited generalization, pointing to the need for a move toward unsupervised learning for more practical use. To improve practicality in real-life circumstances, the research highlights the need for more efficient models with fewer parameters in response to the increased computational needs of deep SISR models. Additionally, it examines the important uses of SISR in medical and satellite imaging, showcasing how it may be used to solve problems with noise and low resolution. The article concludes by summarizing the difficulties encountered in SISR research, outlining a path forward for future developments in the area, and offering suggestions for improving optimization aims and model construction.

III. Dataset

The DIV2K dataset is split up as follows: train data: Using 800 high-definition, high-resolution photos as a starting point, we generate corresponding low resolution images, giving us high and low quality images for downscaling factors of 2, 3, and 4. Validation data: A set of 100 high-definition, high-resolution images is used to generate corresponding low-resolution images. The DIV2K dataset is organized as follows: 1000 photos in 2K resolution are split up into: There are 800 training photos, 100 validation images, and 100 testing images. With 1. bicubic or 2. unknown degrading operators for each challenge track, we have: the photos in high definition: 1000.png, 0001.png, 0002.png,... the images that have been downscaled. The DIV2K folder structure looks like this: YYYYx2.png for downscaling factor x2; YYYY is the image ID YYYYx3.png for downscaling factor x3; YYYY is the image ID YYYYx4.png for downscaling factor x4; YYYY is the image in ID.

IV. Implementation

A. Data Processing

To get images ready for super-resolution tasks, the srcnn ImageTransforms class applies several data transformations. These conversions are essential for validating and training algorithms that aim to improve image resolution. Let's dissect each stage of the transformation Cropping Based on Splits: Firstly, the class determines whether the image belongs to the validation ('valid') or training ('train') dataset. A random crop of the original image is used for training and validation images. The crop-size option establishes the size of this crop. Creating a high-resolution (HR) image that the model may be trained on or validated against requires this step. To Produce Low-Resolution (LR) Images, to get the matching LR image, the cropped HR image is downscaled. The scaling factor argument controls the amount of downscaling. The bicubic interpolation method is used for the downscaling. When it comes to interpolation, bicubic interpolation is a more sophisticated method than bilinear or nearest-neighbor interpolation. The resulting smoother and more realistic LR image takes into account the surrounding pixels' intensity values and computes the pixel value via polynomial interpolation. Following resizing, the HR image's width and height equal the LR image's times its scaling factor. A sanity check is performed in this stage to ensure that the downscaling procedure was executed correctly and by the specified scaling factor. Format Conversion: The original format of the LR and HR photos—possibly PIL images, given that PIL's resize method is used—is converted to the destination formats that are specified. To accomplish this, use the convertimage function. When the ImageTransforms object is initialized, the target formats for LR and HR photos are set. To guarantee compatibility with later processing or neural network input requirements, format conversion is probably required.

We explore the use of deep learning for super-resolution images in our research report, with particular attention to the Super-Resolution Generative Adversarial Network (SRGAN) and the Super-Resolution Convolutional Neural Network (SR-CNN) as two sophisticated models. These models are state-of-the-art in the field and demonstrate how deep-learning approaches can improve image resolution.

B. Model Architecture

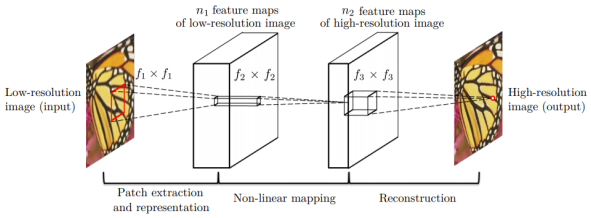
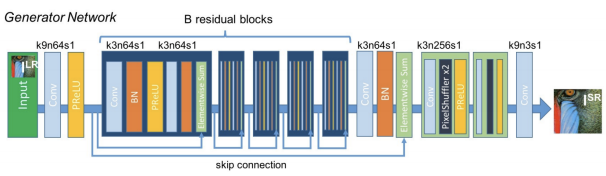


Fig. 2. SRCNN block diagram

Implementation of the SRCNN Model: To enable effective training and processing, the implementation starts with configuring the computational environment and optimizing for GPU usage. A deep convolutional neural network that can learn end-to-end mapping from low-resolution to high-resolution images is the SRCNN model depicted in figure 1. The super-resolution method depends on this mapping. The preparation and transformation of data is an essential part of our implementation process. For this reason, the ImageTransforms class was created, which handles cropping high-resolution photos and then downscaling them to create their matching low-resolution equivalents. Our training dataset is built around these altered photos. Furthermore, during the model training and validation stages, these image pairings are handled and loaded efficiently using proprietary data loaders. SRCNN's architecture is an enhanced version of the conventional model and is contained within the SRResNet class. Its sub-pixel convolutional layers and numerous residual blocks are essential for upscaling low-resolution photos. The dataset for the model's rigorous training consists of image pairs with different resolutions. To maximize the training process, methods like adaptive learning rate modifications and gradient clipping are applied. A validation set is used to assess the model's performance, and measures such as average loss are tracked to determine how well the training was done.



Implementation of the SRGAN Model: The model is based on [4]. Two essential parts make up the Generative

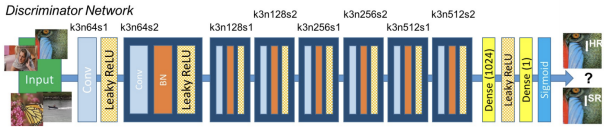


Fig. 3. SRGAN block diagram

Adversarial Network (GAN) operating structure for SRGAN: the Generator and the Discriminator as shown in Figure 3. The Generator's primary function is upscaling low-resolution photos. It does this by gradually improving the quality of the photographs by introducing features like residual blocks and pixel shuffler layers. The Discriminator distinguishes between the original high-resolution photos and the super-resolved images produced by the Generator and picks one of them to be the real one. The adversarial nature of SRGAN training is one of its distinguishing features. Training is done alternately on the Discriminator and Generator. With the goal of "fooling" the Discriminator, the Generator attempts to create images that are identical to actual high-resolution photos. On the other hand, the Discriminator gains proficiency in differentiating between artificially generated and actual images.

Here is the equation for Perceptual Loss:

$$L^{SR} = L_X^{SR} + 10^{-3} L_{Gen}^{SR} \tag{1}$$

where L_X^{SR} is the content loss and L_{Gen}^{SR} is the adversarial loss. The term L_X^{SR} represents the perceptual loss (for VGG-based content losses).

The foundation of the SRGAN is Perceptual Loss, a combination of Adversarial Loss and Content Loss. The adversarial loss is the generative component of the model. Content Loss, on the other hand, plays a very interesting role. The main function of the Perceptual Loss is to play the role of multiple humans evaluating these images [5]. The Content Loss facilitates this by extracting and comparing feature maps of the generated and original high-resolution images using a pre-trained network trained on millions of images (VGG19 in our case).

Here is the equation for Content Loss:

$$MSE^{SR} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} \left(I_{x,y}^{HR} - G_{\theta_G}(I_{x,y}^{LR}) \right)^2 \tag{2}$$

//

Here is the equation for Adversarial Loss:

$$L_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta_D} \left(G_{\theta_G} \left(I^{LR} \right) \right) \tag{3}$$

The generator consists of 16 (experimented with different numbers) ResNet blocks that capture the spatial information

required for upsampling later on in the network. A skip connection is also used from the outputs of the initial convolution layer to the upsampling layer to retain information about the low-resolution image.

The discriminator architecture is quite similar to conventional Convolutional Neural Networks. It has a network of convolutional layers that are eventually flattened and passed through a Sigmoid function for binary classification (real or fake).

C. Training

We used the processing capability of NVIDIA's GEFORCE RTX 2060 and Kaggle's P100 GPU for our SRCNN model training. The training schedule was modified with a batch size of 32 on the P100 to accommodate the GPU memory limitations. To make the best use of the computational resources at our disposal, this change was required. A learning rate of 1e-4 was shown to be ideal after multiple iterations, striking a balance between the training process' stability and the requirement for quick convergence. We used an exponentially decaying learning rate scheduler to make sure the learning rate stayed effective during training. The Adam optimizer was the optimizer of choice. Using frequent checkpoints was a crucial part of our training approach. We were able to reduce the possibility of data loss from possible disruptions by often storing the model's state with weights at intervals of ten epochs. By allowing us to resume training from particular epochs and evaluate the effects of different hyperparameters on the model's learning trajectory, this approach also made it easier to evaluate the model's performance over time.

SRGAN training: We used Kaggle's P100 GPU and NVIDIA's GEFORCE RTX 2060 to train our models. The batch size was 8 on Kaggle's GPU and 4 on the RTX 2060, owing to limited GPU access. We experimented with different learning rates and found 1e-4 to be the most stable. We used Adam optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.9$. Additionally, we used an exponentially decaying learning rate scheduler with a factor of 0.1 to modify the learning rate as the training progresses. Training checkpoints and the frequent saving of model states provided further support for the training process [6]. This made it possible to resume training at different points and made it easier to assess the models at different levels of training.

V. Results

After training the SRCNN model a random image from the test dataset is passed through and the output is predicted. Along with this the loss curve of the SRCNN is shown in figure 4.

The result of SRCNN [7] is shown in Figure 5 and Figure 6, a random picture that was run through the model from the dataset which includes the predicted picture and low-resolution and high-resolution images.

The output of the image passed through the SRGAN model is shown in figure 7 and Figure 8.

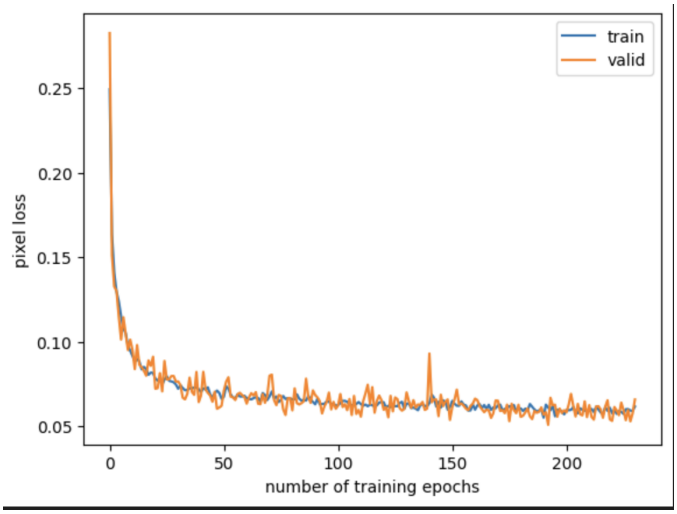


Fig. 4. SRCNN loss curve

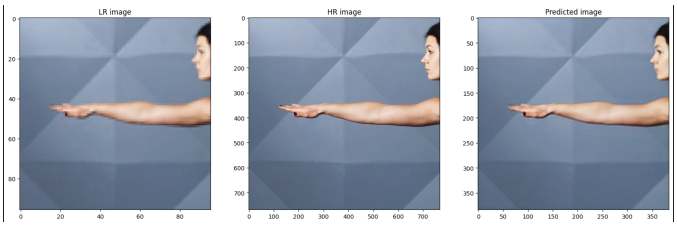


Fig. 5. SRCNN Output image 1

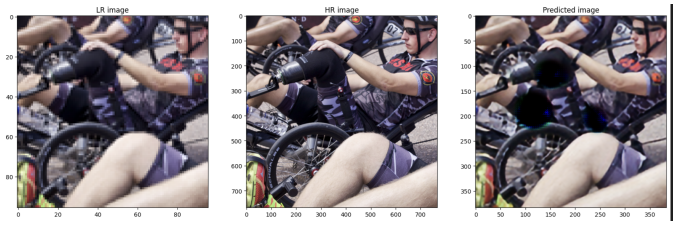


Fig. 6. SRGCNN Output image 2

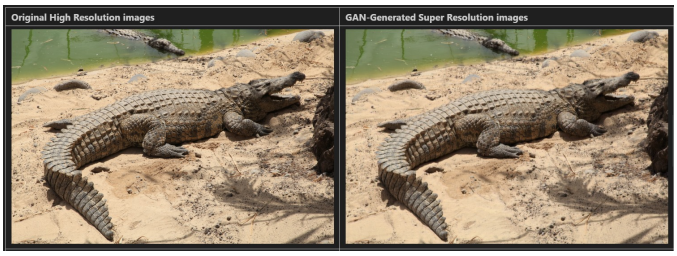


Fig. 7. SRGAN Output image 1

PSNR: A popular statistic in the field of picture super-resolution for assessing the caliber of reconstructed images is the peak signal-to-noise ratio (PSNR). It evaluates the degree of similarity between a super-resolved image generated by an algorithm like SRGAN (Super-Resolution Generative Adversarial Network) or SRCNN (Super-Resolution Convolutional Neural Network) and a high-resolution ground truth image.

Here is the equation for PSNR:

$$PSNR = 10 \times \log_{10} \left(\frac{MAX^2}{MSE} \right) \tag{4}$$

Fig. 8. SRGAN Output image 2

MSE: The average squared difference between the pixel values of the original high-resolution image and the super-resolved image generated by the super-resolution model is a simple and understandable quantitative metric. MSE has limitations, but it's easy to calculate and gives a clear indicator of how accurate the rebuilt image is. From the inference from [8] the main drawback of MSE is that it does not always correspond well with the perceived visual quality; in other words, an image with a lower MSE is not always regarded by humans to be of higher quality.

The Mean Squared Error (MSE) is given by the formula:

$$MSE = \frac{1}{N}\sum_{n=1}^{N}(I^n - P^n)^2 \tag{5}$$

SSIM: Conversely, is a more intricate metric that takes structural information, brightness, and contrast variations into account. A metric that is more in line with the experience of the human visual system is what SSIM attempts to give by comparing local patterns of pixel intensities that have been adjusted for brightness and contrast. When compared to the reference high-resolution image, the super-resolved image has a greater quality when its SSIM value is closer to 1. The Structural Similarity Index (SSIM) between two images X and Y is given by the formula:

$$SSIM(X,Y) = \frac{(2\mu_X\mu_Y + C_1)(2\sigma_{XY} + C_2)}{(\mu_X^2 + \mu_Y^2 + C_1)(\sigma_X^2 + \sigma_Y^2 + C_2)} \tag{6}$$

These metrics are used to compare the image quality with the predicted output as we referred from [9] where a series of tests realized on images extracted from the Kodak database gives a better understanding of the similarity and difference between the SSIM and the PSNR.

Extensions: Adding a multi-scale processing architecture would be a concrete way to expand the SRCNN model. In contrast to merely broadening the network's depth or breadth, multi-scale processing would allow the SRCNN to handle varying image resolutions and scales in a more sophisticated manner. This might be accomplished by incorporating a structure akin to a pyramid, in which images are analyzed at many resolutions concurrently, enabling the network to efficiently learn both fine and coarse information. The model's capacity to reconstruct high-frequency details while preserving global coherence in the super-resolved images could be greatly enhanced by such an architecture. Answered questions:The project offers a definitive response to the query of how well SRCNN can use deep learning techniques to improve image resolution. Quantitative measurements such as PSNR and SSIM demonstrate that SRCNN can really learn to upsample low-resolution pictures to higher resolutions with better pixel accuracy through methodical training, hyperparameter adjustment, and thorough evaluation. But finally the SRGAN model is lighter and more efficient when compared ot the SRCNN.

The performance metrics of the SRGAN are given : Mean PSNR: tensor(23.0440)
STD PSNR: tensor(4.2116)
Min PSNR: tensor(12.5447)
Max PSNR: tensor(35.0528)
Mean SSIM: tensor(0.9204)
STD SSIM: tensor(0.0738)
Min SSIM: tensor(0.5447)
Max SSIM: tensor(0.9958)

Although the PSNR value of the SRCNN is higher and the SSIM value of the SRCNN is closer to one when compared to the SRGAN, the model size is 10 times that of the SRGAN. Therefore, in general, the trade-off between model size and performance informs our decision to prefer the SRGAN over the SRCNN.

The SRGAN can be improved, and still not have a very large model size. Apart from this, the SRGAN is a revelation of how deep learning with the right losses can tremendously improve the model performance.

VI. Future Work

The Super-Resolution Convolutional Neural Network (SRCNN) model has demonstrated encouraging outcomes in terms of improving image resolution. Nonetheless, there is a great deal of room for more study and advancement in this field. Future developments on SRCNN could go in several directions: More architectural improvements to broaden and deepen the model are anticipated, including incorporating complex mechanisms such as residual learning and attention models for better feature learning. More sophisticated options, like perceptual and adversarial loss functions, that are more in line with human visual perception may replace the conventional mean squared error loss function. The model's adaptability can be strengthened by expanding the variety of image types included in the training datasets. This will enable the model to be used in more specialized fields, such as satellite data or medical imaging. Real-time super-resolution is still a long way off, and further work is needed to reduce processing demands—perhaps by using specialized hardware acceleration or model reduction techniques. Adding adversarial training components could potentially result in outputs with improved visual appeal and resolution as well. It will be imperative to make the model more resilient to different image degradations, particularly for applications that are susceptible to distortions

from noise and compression. Incorporating human interaction may also usher in a new era of super-resolution tailored to specific applications, allowing users to define focal points within images. Finally, combining SRCNN with other imaging modalities could result in composite systems that can produce panoramic and high-dynamic-range images at previously unheard-of resolutions. When taken as a whole, these potential directions highlight not only SRCNN's near-future potential but also its significance as a pillar in the ongoing story of image super-resolution research.

Regarding the SRGAN, the data preprocessing can be further extended to represent a wider distribution of the input and output spaces in the following ways. Instead of centrally cropping the images to a certain dimension, a random crop can be applied. Also, various augmentations like flipping, rotating, etc, could be implemented to achieve the same. Alongside a more sophisticated data loader, various other loss functions can be used (e.g., ResNet Loss instead of VGG Loss for Perceptual Loss, VGG22 or VGG54 instead of VGG19 for Perceptual Loss, Wasserstein Loss in place of the conventional Adversarial Loss, etc).

Another possible avenue for improvement is the hyperparameters, optimizers, and the model itself. Hyperparameter tuning for a GAN requires a lot of time and experimenting. With sufficient time, a different set of hyperparameters and optimizers can be used with a slightly restructured model.

REFERENCES

[1] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4681–4690.

[2] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[3] Z. Cao, X. Liu, and Z. Wang, "Single image super-resolution via deep learning," in *2022 3rd International Conference on Computer Vision, Image and Deep Learning International Conference on Computer Engineering and Applications (CVIDL ICCEA)*, 2022, pp. 425–430.

[4] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 2017.

[5] C. Nguyen., "Image super-resolution using a generative adversarial network," https://github.com/cuongvng/srgan-pytorch, 2023.

[6] A. Persson, "Srgan," https://github.com/aladdinpersson/Machine-Learning-Collection/tree/master/ML/Pytorch/GANs/SRGAN, 2021.

[7] "Srgan with srresnet," https://www.kaggle.com/code/alimohammedz/srresnet-srganMetrics–PSNR--SSIM, 2021.

[8] U. Sara, M. Akter, and M. S. Uddin, "Image quality assessment through fsim, ssim, mse and psnr—a comparative study," *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, 2019.

[9] A. Horé and D. Ziou, "Image quality metrics: Psnr vs. ssim," in *2010 20th International Conference on Pattern Recognition*, 2010, pp. 2366–2369.