Image Super Resolution using GANs

Ashwanth Kumar Gundeti
Purdue University
Indiana, USA
agundeti@purdue.edu

Abstract

In this project, we investigate the performance of three different generative adversarial network (GAN) models for super-resolution image generation: SRGAN, ESRGAN, and SRFeat. Super-resolution is a critical task in image processing that aims to generate high-resolution images from low-resolution inputs. Our study focuses on comparing the quality of images produced by these three models and analyzing their strengths and weaknesses. We conduct experiments on a benchmark dataset and evaluate the results using various metrics, including peak signal-to-noise ratio (PSNR) and structural similarity index (SSIM). Our findings show that while all three models are effective at generating high-resolution images, ESRGAN performs slightly better than the other two models in terms of image quality and visual fidelity. Our study provides useful insights into the performance of GAN models for super-resolution image generation and may help guide future research in this area.

1 Introduction

Super-resolution image reconstruction is a challenging task that has been extensively studied in the field of computer vision. The goal is to recover high-resolution images from low-resolution inputs, which is a problem that arises in various applications such as medical imaging, remote sensing, and surveillance. Traditional approaches based on interpolation and filtering techniques have limited success in producing high-quality results, especially when dealing with complex and detailed images. In recent years, deep learning techniques have emerged as a powerful tool for solving this problem, particularly using Generative Adversarial Networks (GANs).

The use of GANs for super-resolution has gained significant attention due to their ability to generate high-quality images with realistic details and textures. GANs consist of a generator network that generates high-resolution images from low-resolution inputs and a discriminator network that tries to distinguish between the generated images and the ground truth. By training the generator network to fool the discriminator network, GANs can learn to generate high-quality images that are indistinguishable from the ground truth.

Despite the success of GANs in super-resolution, there are still challenges that need to be addressed. One of the main challenges is the ability to generate high-quality images with fine details and textures, while avoiding artifacts such as blurriness and checkerboard patterns. Additionally, training GANs for super-resolution is a computationally intensive task that requires large amounts of training data and high-performance computing resources. Nevertheless, recent advances in GAN architectures and training techniques have shown promising results in improving the performance and efficiency of GAN-based super-resolution.

In this paper, we look at three existing approaches for super-resolution using GANs, which incorporate modified residual block structure and perceptual loss function. We compare these three models and demonstrate their superior performance in terms of visual quality and quantitative metrics. All the

Figure 1: Example of an image pair in the training dataset. On the left is the low-resolution image, and on the right is it's corresponding high-resolution pair. [2× upscaling]

models perform an up-scaling of factor 2×, i.e, a low resolution image of size (h,w) will be used to generate high resolution image of size (2×h, 2×w).

In the following sections, we will describe our methodology for comparing the models, present our experimental results and analysis, and discuss the implications and future directions of this research, highlighting the potential for further improvements using updated learning techniques and other GAN architectures.

2 Related Work

Image super-resolution is the process of generating a high-resolution (HR) image from a low-resolution (LR) input. This is a challenging task as the LR image contains less information than the HR image, making it difficult to accurately recover the details lost during down-sampling. Prediction-based methods were among the first approaches used to tackle this problem. These methods are typically based on interpolation and filtering techniques such as linear, bicubic, or Lanczos filtering. However, these methods often result in blurry and unrealistic HR images. With the advent of deep learning, researchers have started to explore the use of neural networks for image super resolution.

Deep learning-based super resolution algorithms work by training a neural network to learn the mapping between the LR and HR images. The network is typically trained on a large dataset of LR and HR image pairs, with the goal of minimizing the difference between the network's output and the ground truth HR image. Over the years, researchers have proposed various neural network architectures and training strategies to improve the performance of image SR algorithms. They have developed learning-based approaches that use the example-pairs of LR and HR images to learn a complex mapping between the two. These approaches typically rely on training data to establish this mapping. One popular approach is to use deep learning techniques, such as convolutional neural networks (CNNs), which have shown excellent performance in recent years. By learning the mapping between LR and HR images, these methods can generate much more realistic and detailed HR images.

One approach is based on compressed sensing, where the sparsity of the image is used to reconstruct the HR image. Other approaches use self-similarity, where redundancies across scales within the image are used to drive the super resolution. Some methods rely on neighborhood embedding, where similar LR training patches are found in a low-dimensional manifold and combined to reconstruct the HR image. Another recent trend in image super-resolution is to incorporate perceptual similarity into the loss function used during training. This approach aims to generate visually more convincing HR images by minimizing the difference between the generated HR image and the ground truth HR image, not just in terms of pixel values but also in terms of perceived quality.

Overall, image super-resolution is a rapidly advancing field with many promising approaches that can significantly improve the quality of low-resolution images. As computational resources continue to improve, we can expect these methods to become even more powerful and effective in the years to come.

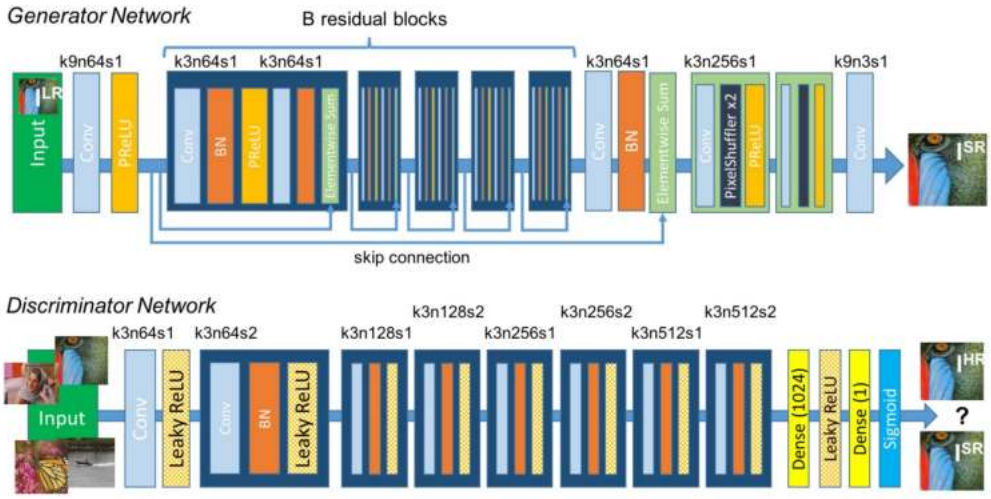


Figure 2: Image Credit [1]. Architecture of SRGAN Generator Network with corresponding kernel size (k), number of feature maps (n) and stride (s) indicated for each convolutional layer. Figure also shows the Discriminator Network architecture common to all the models.

3 Methodology

Our aim is to generate a high-resolution image $I^{'}_{HR}$ from an input low-resolution image I_{LR}. We also use the corresponding high-resolution mappings, I_{HR}, for the low-resolution inputs during the training phase of our models. For an image with C color channels, we describe I_{LR} by a real-valued tensor of size $W \times H \times C$ and I_{HR} by $2W \times 2H \times C$ respectively. For achieving this, the goal is to train a generating function G, which is a feed-forward convolutional neural network (CNN), to estimate the high-resolution image from the low-resolution input image.

The network generates a image $I^{'}_{HR}$ from I_{LR}. The image $I^{'}_{HR}$ has the same dimensions as I_{HR}. The network is first trained to reduce the pixel-wise difference between $I^{'}_{HR}$ and I_{HR}. Pixel-wise loss helps reproduce a good $I^{'}_{HR}$ in terms of PSNR, but generally results in a blurry image. To improve the visual quality of $I^{'}_{HR}$, we employ a perceptual loss and introduce additional GAN-based loss functions. These losses enable the network to generate more realistic images by approximating the distributions of I_{HR} images and their feature maps.

The network is formulated as an adversarial min-max problem in the below equation:

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I_{HR} \sim p_{train}(I_{HR})} \left[\log D_{\theta_D}(I_{HR})\right] + \mathbb{E}_{I_{LR} \sim p_G(I_{LR})} \left[\log(1 - D_{\theta_D}(G_{\theta_G}(I_{HR})))\right] \quad (1)$$

We will describe the network architectures of different models in the following subsections along with the training loss functions in detail.

3.1 SRGAN

Figure 2 shows the network architectures of generator and discriminator. The generator network, G, consists of 16 identical residual blocks, where each block has two convolutional layers with 3×3 kernels and 64 feature maps. These convolution layers are followed by batch-normalization and ParametricReLU activation layers in every residual block. The input resolution is increased using two trained sub-pixel convolution layers.

To discriminate I_{HR} from generated $I^{'}_{HR}$ samples we train a discriminator network. It contains 8 convolutional layers with an increasing number (by a factor of 2) of 3×3 filter kernels, increasing from 64 to 512 kernels. Each convolution layer is followed by batch-normalization and LeakyReLU ($\alpha = 0.2$) activation layers. Strided convolutions are used to reduce the image resolution each time

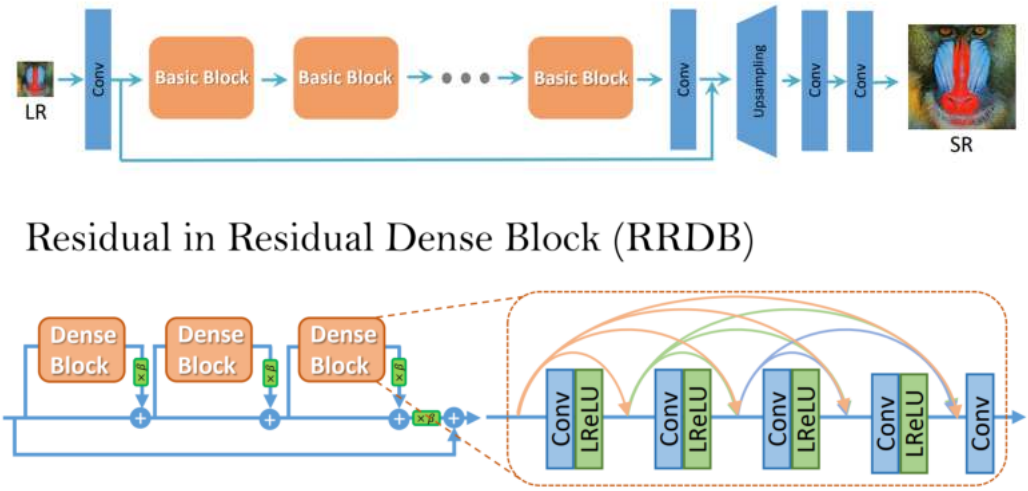


Figure 3: Image Credit [2]. Architecture of ESRGAN Generator Network. The basic block is replaced by the proposed Residual in Residual Dense Block.

the number of features is doubled. The resulting 512 feature maps are followed by two dense layers and a final sigmoid activation function to obtain a probability for sample classification. We use a similar discriminator network for the other two models as well.

3.2 ESRGAN

The author proposes two modifications to the structure of the generator network in the SRGAN architecture to improve the quality of the generated super-resolved images. The first modification is the removal of all batch-normalization layers from the generator network. These layers are commonly used to normalize features during training by computing the mean and variance within a batch of data. However, when the statistics of the training and testing datasets differ significantly, batch-normalization layers can introduce artifacts that limit the generalization ability and stability of the network. Therefore, removing batch-normalization layers can lead to a more stable and consistent performance, as well as reduced computational complexity and memory usage.

The second modification to the generator network is the replacement of the original basic block with a proposed Residual-in-Residual Dense Block (RRDB). The RRDB combines multi-level residual networks and dense connections to create a deeper and more complex structure than the original residual block in SRGAN. Residual learning is used in different levels within the RRDB, and the network capacity is increased with the use of 22 such RRDB blocks in series. This modification allows for more layers and connections, which can boost performance. The convolutional layers in these blocks use the same parameters as defined in SRGAN.

In addition to these modifications, the author also employs several techniques to facilitate training a very deep network, including residual scaling and smaller initialization. Residual scaling involves scaling down the residuals by multiplying a constant between 0 and 1 (which we set to 0.3) before adding them to the main path, which helps to prevent instability. Smaller initialization is used because the findings suggest that residual architectures are easier to train when the initial parameter variance is smaller.

The architecture of the modified generator network is shown in the Figure 3.

3.3 SRFeat

SRFeat uses a Deep Convolutional Neural Network (DCNN) generator that is designed to enhance the resolution of low-resolution images. The network is composed of multiple residual blocks and long-range skip connections.

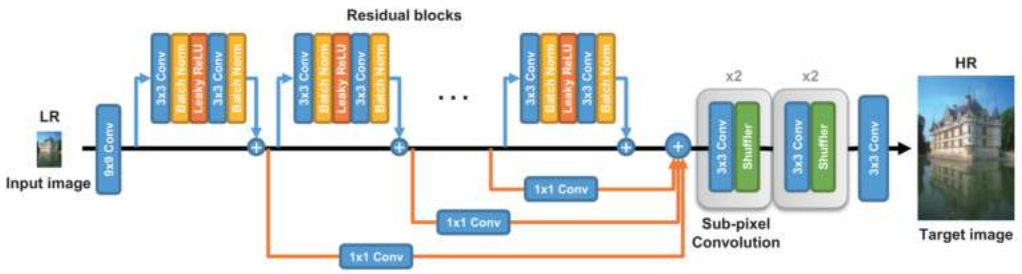


Figure 4: Image Credit [3]. Architecture of SRFeat Generator Network with short and long range skip connections

The network uses multiple residual blocks to learn higher-level features with more nonlinearities and larger receptive fields. Each residual block has a short-range skip connection as an identity mapping that preserves the signal from the previous layer and allows the network to learn residuals only, while back-propagating gradients through the skip-connection path.

The residual blocks consist of successive layers of 3×3 convolution, batch-normalization, LeakyReLU $(\alpha = 0.2)$, 3×3 convolution, and batch-normalization layers similar to SRGAN. The network uses 16 residual blocks to extract deep features, and all the residual blocks are applied to the features of the low-resolution spatial dimensions for efficient memory usage and fast inference.

The network utilizes additional long-range skip connections to aggregate features from different residual blocks. The output of each residual block is connected to the end of the residual blocks with one 1×1 convolution layer to encourage back-propagation of gradients and re-use intermediate features to improve the final feature. The outputs of different residual blocks correspond to different levels of abstraction of image features, and a 1×1 convolution is applied to each long-range skip connection to adjust the outputs and balance them.

To upsample the feature map obtained by the residual blocks to the target resolution, the network uses sub-pixel convolution layers, again similar to SRGAN. A sub-pixel convolution layer enlarges an input feature map by the scale factor in each spatial dimension. The upsampled feature map goes into a 3×3 convolution layer with 3 filters to obtain a 3-channel color image.

Overall, the DCNN generator architecture makes use of long-range skip connections to produce high-quality and realistic images.

The architecture of the modified generator network is shown in the Figure 4.

3.4 Loss functions

The definition of loss function is critical for the performance of generator and discriminator networks, especially generator. A generator should have the capability to produce realistic and high-quality images to beat a well trained discriminator. Hence, the definition is key for it's functionality. We formulate the generator loss (l_G) as a weighted sum of content loss (l_{CL}) and adversarial loss (l_{AL}):

$$l_G = l_{CL} + 10^{-4} l_{AL} \tag{2}$$

We describe the content loss and adversarial loss in the upcoming subsections.

3.4.1 Content loss

The pixel-wise MSE loss is given by:

$$l_{MSE} = \frac{1}{4WH} \sum_{i=1}^{2W} \sum_{j=1}^{2H} \left(I_{HR/(i,j)} - I^{'}_{HR/(i,j)} \right)^2 \tag{3}$$

This is the most commonly used optimization target for image super resolution tasks. This helps in achieving high PSNR values. However, these models lack high frequency content resulting in

visually unsatisfactory images with overly smooth textures. Hence, we also introduce an additional loss function, that improves the perceptuality, known as the perceptual similarity loss. It measures the difference between two images in feature domain rather than the pixel domain. This loss function is defined in the following manner:

$$l_P = \frac{1}{W_m H_m C_m} \sum_{i=1}^{W_m} \sum_{j=1}^{H_m} \sum_{k=1}^{C_m} \left(\phi^m_{i,j,k}(I_{HR}) - \phi^m_{i,j,k}(I^{'}_{HR}) \right)^2 \tag{4}$$

where W_m, H_m, C_m denote the dimensions of the images I_{HR} and $I^{'}_{HR}$ at the m^{th} feature map, ϕ^m, when fed as input to a pre-trained recognition network. We use a 19 layer VGG network as the recognition network for this task. ϕ^m denotes the output of the activation layer after convolution before the m^{th} max-pooling layer. The content loss is formulated as:

$$l_{CL} = l_{MSE} + 10^{-1} l_P \tag{5}$$

3.4.2 Adversarial loss

In addition to the content loss, we also penalize the generator if it fails to fool the discriminator network. The job of the discriminator is to discriminate fake generated images $I^{'}_{HR}$ from ground truth high-resolution images I_{HR}. Thus, the discriminator loss is given as:

$$l_D = -0.5 \log \left(D_{\theta_D}(I_{HR}) \right) - 0.5 \log(1 - D_{\theta_D}(I^{'}_{HR})) \tag{6}$$

We build on top of this to define an adversarial loss function for the generator network as:

$$l_{AL} = -\log(D_{\theta_D}(I^{'}_{HR})) \tag{7}$$

Here $D_{\theta_D}(I)$ is the probability that the image I is the ground truth high-resolution image.

4 Experiments

4.1 Dataset

The DIV2K dataset is a widely-used benchmark dataset for image super resolution tasks. It consists of 900 high-resolution images with varying resolutions. Each high-resolution image also has a corresponding low-resolution image that is generated using some unknown algorithm. The images cover a wide range of scenes, including natural landscapes, urban scenes, and indoor environments.

The dataset is publicly available and can be downloaded for free. With 900 high-resolution images, the dataset provides a large and diverse set of images for training and testing. This makes it possible to evaluate the performance of image super resolution algorithms across a wide range of scenes and image types. The dataset uses various pre-processing techniques to provide corresponding low-resolution images, which can be used as input to image super resolution algorithms. We have used 800 image pairs for training and the remaining 100 for testing.

4.2 Training details

We trained all the networks on Google Colab using GPU as hardware accelerator. We randomly cropped the images to 128×128 for low-resolution images and 256×256 for high-resolution images. We also normalized the range of images to $[0, 1]$.

We pre-train the generator for 5 epochs just with pixel-wise loss to obtain reasonable images so that discriminator receives relatively good images instead of extreme fake outputs in the initial epochs from the generator. This also enables the discriminator to focus more on texture details from the get go.

For both pre-training and adversarial training, we use Adam Optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.99$ with a learning rate of 10^{-4}. We update the discriminator every 3 epochs in all the

Figure 5: From left to right: Quantitative comparison of Bicubic, Nearest, SRGAN, ESRGAN, SRFeat, Ground Truth. Corresponding SSIM and PSNR values are shown in the brackets.

approaches. We define a stopping condition for training when the loss of discriminator is stuck between 0.5 and 0.7 and the generator loss hasn't decreased enough. It took 120 epochs for SRGAN, 100 epochs for ESRGAN and 250 epochs for SRFeat to reach this stopping criteria. The batch size is set to 4 for SRGAN and SRFeat, but it is set to 2 for ESRGAN due it's large network. We disable the batch-normalization layers (if any) during the testing phase.

4.3 Performance of the networks

We compare the performance of SRGAN, ESRGAN, SRFeat along with Nearest Neighbor and Bicubic interpolation on the test dataset. Quantitative results can be observed in Figure 5 and are summarized in Table 2 and confirm that ESRGAN (in terms of PSNR/SSIM) is the superior based on these results. ESRGAN is capable of generating more detailed textures while other methods either fail to produce enough details or add undesired textures. We can also look at the qualitative results in Figure 6.

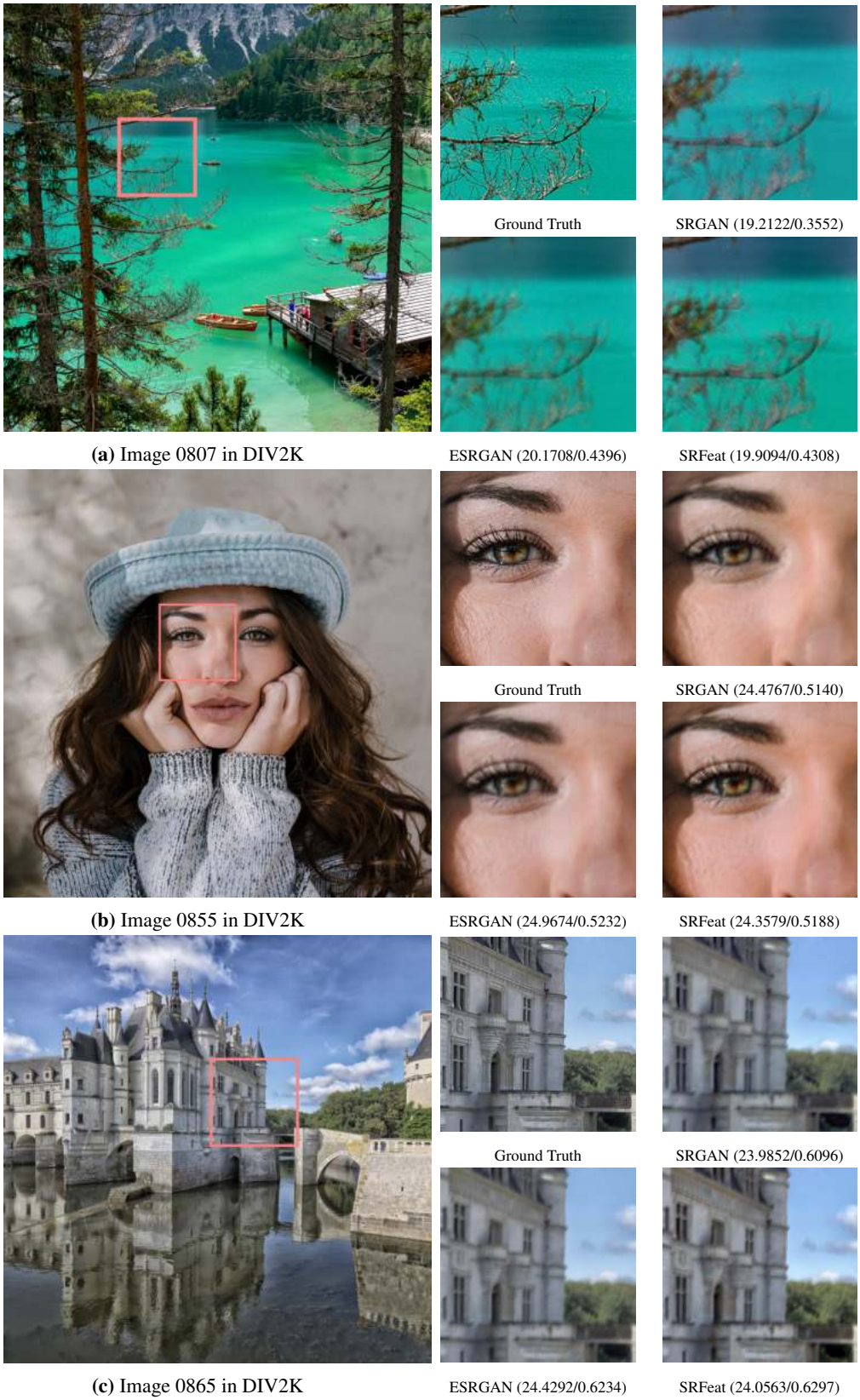
DIV2K	Bicubic	NN	SRGAN	ESRGAN	SRFeat	HR
RMSE	0.0721	0.0717	0.0740	0.0601	0.0712	0
PSNR	22.8412	22.8896	23.2326	25.2020	23.4870	∞
SSIM	0.6594	0.6557	0.6679	0.7144	0.6795	1

Table 1: Comparison of Bicubic, NN, SRGAN, ESRGAN, SRFeat and the original HR on test dataset. [$2\times$ upscaling]

5 Conclusion

In this project, we compared three different Image Super Resolution (ISR) models, namely SRGAN, ESRGAN, and SRFeat that can produce perceptually pleasing images by employing different Generative Adversarial Networks.. These models were chosen because they represent the state-of-the-art models in the field of ISR and have achieved significant improvements in terms of image quality compared to traditional ISR methods. First, we provided a brief overview of the ISR problem and discussed the different evaluation metrics used to measure the performance of ISR models. Then, we introduced each of the three models, explaining their unique features, network architectures, and training procedures. Although the difference lies in the generator Network, the discriminator encourages the generator to make more structural high-frequency details rather than noisy artifacts. We also discussed the advantages and limitations of each model.

Next, we performed a comparative study of the three models by evaluating their performance on a benchmark dataset. The results showed that ESRGAN performed better than the other two models in terms of objective metrics such as PSNR and SSIM, we can also consider subjective metrics such as Mean Opinion Score (MOS) as described in [1]. Overall, our study highlights the importance of selecting appropriate evaluation metrics when comparing ISR models. While objective metrics provide a quantitative measure of performance, they may not always correspond to perceived image



(a) Image 0807 in DIV2K

(b) Image 0855 in DIV2K

(c) Image 0865 in DIV2K

Figure 6: Qualitative results of all the models over 3 images from the testing dataset. ESRGAN produces relatively more natural textures.

quality. Thus, it is important to consider both objective and subjective evaluations when comparing ISR models. In conclusion, the three models we compared represent significant advancements in the field of ISR and have demonstrated impressive results in enhancing the resolution of low-resolution images. Our study provides valuable insights into the strengths and weaknesses of each model, which can be useful for researchers and practitioners in choosing the appropriate model for their specific applications.

Based on our conclusion that ESRGAN performed better than SRGAN and SRFeat in terms of PSNR and SSIM, one potential improvement could be to further optimize the training process (changing the stopping criteria) of ESRGAN to achieve even better results. This could involve experimenting with different hyperparameters, such as learning rates, batch sizes, and number of epochs, as well as incorporating other techniques like data augmentation, regularization, and advanced loss functions.

Another improvement could be to test these models on a wider range of images and scenarios to evaluate their robustness and generalizability. It could also be interesting to compare the computational efficiency of these models and their ability to handle different image sizes and resolutions. The models also generate visually realistic images for $4\times$ upscaling task. We can add additional convolution layers to achieve this.

Furthermore, exploring new architectures or combining the strengths of different models could potentially lead to even better results. For example, recent studies have proposed hybrid models that combine the advantages of generative and discriminative models, as well as models that incorporate attention mechanisms and multi-scale processing.

References

[1] Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J. and Wang, Z. (2017) *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.*

[2] Xintao Wang, Ke Yu, Shixiang Wu, Jinjin Gu, Yihao Liu, Chao Dong, Chen Change Loy, Yu Qiao and Xiaoou Tang (2018) *ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks.*

[3] Seong-Jin Park, Hyeongseok Son, Sunghyun Cho, Ki-Sang Hong and Seungyong Lee (2018) *SRFeat: Single Image Super-Resolution with Feature Discrimination.*

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun (2015) *Deep Residual Learning for Image Recognition*