Proceedings of CONF-MLA 2024 Workshop: Semantic Communication Based Complexity Scalable Image Transmission System for Resource Constrained Devices
DOI: 10.54254/2755-2721/83/2024GLG0065

Exploring SRGAN-Based Model for Image Super-Resolution

Runqi Yang

School of Electrical Engineering and Computer Science, Pennsylvania State University, State College, Pennsylvania, 16801, United States

rky5094@psu.edu

Abstract. Transforming low-resolution (LR) images into their high-resolution (HR) counterparts is a challenging endeavor, referred to as image super-resolution (SR). The emphasis of this research is on the Super-Resolution Generative Adversarial Network (SRGAN), a sophisticated deep learning approach that can generate aesthetically pleasing high-resolution images. Initially, a backbone model is trained with mean squared error (MSE) loss function to enhance image details. Then, the author took this pretrained the backbone and used it to initialize the generator part of SRGAN, after which the author performed adversarial training where various content accuracy - visual realism tradeoffs were considered during optimization process. This work tested models on standard benchmarks like Set5, Set14, and BSD100 with Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). The best results were achieved when using intermediate learning rates combined with 16 batch size – those settings resulted in highest PSNR and SSIM values across all datasets that have been tested. To sum up, SRGAN has proven itself effective at solving problems related to image super resolution but there is still room left for improvement by adjusting hyperparameters or modifying architecture design based on findings from this paper.

Keywords: Super-resolution, Super-resolution generative adversarial network, Deep learning.

1. Introduction

Super-resolution (SR) denotes the complex process of inferring a high-resolution (HR) image from its corresponding low-resolution (LR) version [1,2]. This technique holds significant value across a spectrum of practical uses, such as medical diagnostics, aerial photography from satellites, security monitoring, and everyday gadgets. The enhanced detail in high-resolution images is crucial for precise examination and informed decision-making within these industries.

Recent years have seen significant advancements in image super-resolution, largely attributed to the power of deep learning [3]. Nonetheless, traditional interpolation techniques often fail to keep high frequency details resulting in blurry images or images with no clarity, and this is a big blow in the image super-resolution field. Moreover, Approaches grounded in deep learning are capable of generating superb, high-fidelity, high-resolution images by mastering the intricate mapping functions between LR and HR imagery.

Super-Resolution Generative Adversarial Network (SRGAN) has become a groundbreaking method among these deep learning approaches. SRGAN is based on generative adversarial networks (GANs) to generate highly realistic HR images [4]. This framework is composed of two principal elements: the generator network, responsible for upgrading the resolution of images from low to high, and the

DOI: 10.54254/2755-2721/83/2024GLG0065

discriminator network, tasked with identifying authentic high-resolution images against those synthesized by the generator. Via this competitive training process, the generator is trained to craft high-resolution images that are both realistic and rich in detail.

There has been rapid progress in deep learning-based SR. SR techniques have been proposed and models have been developed to improve quality and efficiency. For example, the SRResNet model improves image details by adding residual blocks, while SRGAN extends that by combining it with adversarial training to further boost visual quality [5]. However, there is a need for better optimization of hyperparameters to balance image quality against speed of training. Additionally, exploring the impact of different network structures on SRGAN's performance can lead to further improvements in the model's effectiveness [6].

The paper focuses on the performance of the SRGAN network under various hyperparameter settings and different structures. This work utilizes well-known datasets such as Set5, Set14, and BSD100 for testing, and the COCO2014 dataset for training. This work systematically varies the learning rate and batch size for determining their influence on Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM). Experiments show that an intermediate learning rate together with a batch size of 16 are most effective in terms of achieving the highest PSNR and SSIM values respectively. Furthermore, the author considers how these hyperparameters might be used to achieve faster training or better performance in future research.

2. Method

2.1. Dataset and preprocessing

To train SRGAN model, several datasets are leveraged. COCO2014 was used as the training dataset [7]. Set5, Set14 and BSD100 were used for testing. The reason why this work chose COCO2014 is that it covers a wide range of real-life images with different scenes and objects which helps in improving the generalization ability of the model. At the same time, Set5, Set14 and BSD100 are well-known benchmarks in image super-resolution research that allow people to evaluate models against them fairly [8,9].

To form the training pairs for the model, the author downsampled all HR images into LR images. Then all of them were normalized within the range [-1, 1] for normalization. Moreover, the author performed a few augmentation techniques on the training data to increase diversity and prevent overfitting such as random cropping or horizontal flipping.

2.2. Model

The SRGAN model training process involves two primary steps. First, this work trains the SRResNet model, which serves as the backbone for the SRGAN generator. Using a mean squared error (MSE) loss function, the author trains SRResNet to generate HR images from low-resolution (LR) inputs. Once SRResNet is trained, it is used to initialize the generator network of the SRGAN model. In the second stage, the author proceeds with training the SRGAN model through adversarial training. This includes further refining the HR images generated by the pre-trained SRResNet generator using the MSE loss function. Concurrently, the discriminator is trained with a binary cross-entropy loss function to differentiate between real HR images and those generated by SRGAN. Finally, by combining the content loss from the generator and the adversarial loss from the discriminator, the SRGAN model is optimized to produce HR images that are both accurate and visually convincing.

2.3. Evaluation metrics

In this study, the performance of the SRGAN model was thoroughly assessed using the following evaluation criteria: PSNR, which quantifies the discrepancy between the synthesized and actual images, with higher scores signifying superior image clarity. SSIM, which evaluates the structural congruence between the rendered and genuine images, with elevated scores denoting greater retention of structural integrity [10].

DOI: 10.54254/2755-2721/83/2024GLG0065

3. Results

3.1. Training details

To ensure reproducibility and accuracy, the author conducted experiments using a set of well-defined hyperparameters in a controlled software environment. For SRResNet and SRGAN models, the initial learning rate was 10^{-4}, which was determined to be a suitable value for convergence speed and stability after some trial runs. The batch size of 16 was chosen as it strikes a good balance between model performance and training speed when evaluated with PSNR and SSIM metrics. The SRResNet model was learned for 30 epochs. The SRGAN model was trained for an additional 50 epochs. These epoch numbers were determined based on early stopping criteria observed in preliminary trials, where the model's performance plateaued.

To sum up, the default setup is as following. The default setup for SRResNet includes: epochs 30, batch size 16, learning rate 10^{-4}. The default setup for SRGAN includes: epochs 50, batch size 16, learning rate 10^{-4}.

The loss functions employed during training were carefully selected to suit each phase of the model's development. For the SRResNet training, the author used the MSE loss function to minimize the pixel-level differences between the generated HR images and ground truths. When transitioning to the SRGAN model, the training process incorporated both content preservation and image quality enhancement. This was achieved by alternating between minimizing the MSE loss for content preservation and optimizing the adversarial loss for realism. Thus, it was ensured that the generated images closely mirrored the essence of the original images while also presenting a visually persuasive appearance.

Experiments were done using Python due to its rich library supportiveness combined with deep learning frameworks compatibility. PyTorch was chosen as the deep-learning framework because it is flexible when implementing and training the SRGAN model. The GPU is the NVIDIA A100, which has enough computational power required for massive data handling capacity coupled with complex computation needs during the training phase. The use of the A100 GPU significantly reduced training time, enabling extensive experimentation with various hyperparameters and model architectures. This setup was chosen to create a strong basis for the model development under test conditions which may be repeated in future iterations.

3.2. Quantitative performance

3.2.1. Impact of training epochs in SRResNet. As demonstrated in Table 1, upon elevating the training period to 60 epochs, no significant enhancement in the model's performance is observed beyond the results achieved with 30 epochs of training. Conversely, training for only 15 epochs results in a less effective model than training for 30 epochs. This suggests that the model may have overfitted during training beyond 30 epochs.

Table 1. Performance of different epochs using SRResNet, with learning rate 1e-4 and batch size 16.

Epoch	Dataset	PSNR	SSIM
15	Set5	25.079	0.768
	Set14	25.307	0.709
	BSD100	24.707	0.651
30	Set5	28.294	0.829
	Set14	25.941	0.714
	BSD100	25.048	0.661
60	Set5	23.350	0.791
	Set14	25.941	0.714
	BSD100	25.048	0.661

DOI: 10.54254/2755-2721/83/2024GLG0065

3.2.2. Impact of learning rate in SRResNet. As demonstrated in Table 2, the model performs the best when the learning rate is 1e-4.

Table 2. Performance of different learning rate using SRResNet, with epoch 30 and batch size 16.

Learning rate	Dataset	PSNR	SSIM
1e-5	Set5	27.846	0.833
	Set14	25.955	0.726
	BSD100	24.512	0.629
1e-4	Set5	28.294	0.829
	Set14	25.941	0.714
	BSD100	25.048	0.661
1e-3	Set5	20.265	0.743
	Set14	24.723	0.727
	BSD100	24.803	0.685

3.2.3. Impact of batch size in SRResNet. As displayed in Table 3, the model achieves the best results with a batch size of 16. From this, it can be concluded that a batch size that is too large may speed up the training process but does not lead to the optimal solution. Conversely, a batch size that is too small can negatively impact the training of the model and may lead to overfitting.

Table 3. Performance of different batch size using SRResNet, with learning rate 1e-4 and epoch 30.

Batch size	Dataset	PSNR	SSIM
8	Set5	25.079	0.851
	Set14	25.307	0.722
	BSD100	24.707	0.665
16	Set5	28.294	0.829
	Set14	25.941	0.714
	BSD100	25.048	0.661
32	Set5	27.643	0.828
	Set14	24.442	0.675
	BSD100	23.831	0.620

3.2.4. Impact of training epochs in SRGAN. Performance in Table 4 demonstrates that the SRGAN achieves the best performance at 50 epochs. When the number of epochs is increased to 100, both PSNR and SSIM metrics across all datasets (Set5, Set14, BSD100) declines. Similarly, reducing the number of epochs to 25 also results in lower performance metrics compared to 50 epochs. This suggests that learning the SRGAN model for 50 epochs provides an optimal balance between training duration and model performance. Excessive training epochs could result in overfitting, a scenario where the model excels on the training set but falters on the test set. However, an insufficient number of training epochs might deprive the model of the adequate exposure needed to learn and internalize critical patterns.

Table 4. Performance of different epochs using SRGAN, with learning rate 1e-4 and batch size 16.

Epoch	Dataset	PSNR	SSIM
25	Set5	27.774	0.822
	Set14	24.122	0.660
	BSD100	23.360	0.600

DOI: 10.54254/2755-2721/83/2024GLG0065

Table 4. (continued)

50	Set5	28.294	0.829
	Set14	25.941	0.714
	BSD100	25.048	0.661
100	Set5	27.366	0.795
	Set14	25.806	0.699
	BSD100	24.816	0.634

3.2.5. Impact of learning rate in SRGAN. As shown in Table 5. Learning rate with 1e-5 performs the best in all explorations.

Table 5. Performance of different learning rate using SRGAN, with epoch 50 and batch size 16.

Learning rate	Dataset	PSNR	SSIM
1e-5	Set5	29.770	0.860
	Set14	27.327	0.765
	BSD100	26.582	0.722
1e-4	Set5	28.294	0.829
	Set14	25.941	0.714
	BSD100	25.048	0.661
1e-3	Set5	27.763	0.836
	Set14	25.238	0.699
	BSD100	24.755	0.669

3.2.6. Impact of batch size in SRGAN. As shown in Table 6, an excessively small batch size may cause the model to converge prematurely at a local optimum, hindering it from attaining optimal performance. In contrast, an overly large batch size can lead to a plateau in performance gains, and in some cases, it might even result in a decline in the model's effectiveness. Experiments showed that a batch size of 16 provided the best balance, yielding the highest PSNR and SSIM across the tesets. In future experiments, the author can try adjusting to a higher batch size to potentially improve the training speed of the model while ensuring it maintains high performance.

Table 6. Performance of different batch size using SRGAN, with learning rate 1e-4 and epoch 50.

Batch size	Dataset	PSNR	SSIM
8	Set5	23.476	0.739
	Set14	24.935	0.688
	BSD100	24.110	0.614
16	Set5	28.294	0.829
	Set14	25.941	0.714
	BSD100	25.048	0.661
32	Set5	28.441	0.827
	Set14	25.604	0.693
	BSD100	24.415	0.632

DOI: 10.54254/2755-2721/83/2024GLG0065

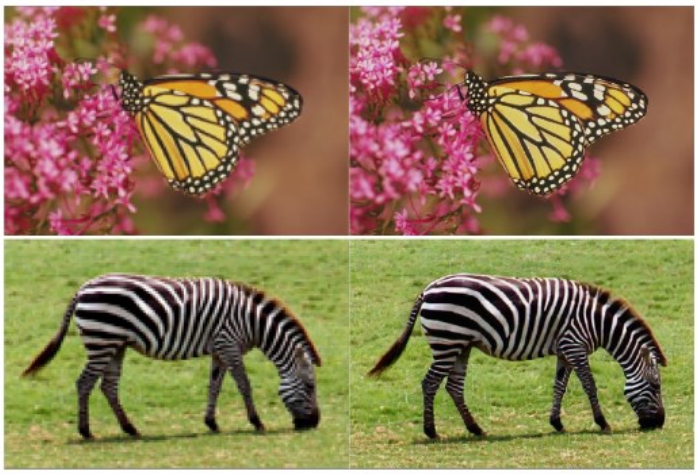

Figure 1. Representative examples of low and high resolution images (Figure Credits: Original).

3.3. Visualization

In this section, the author presents a visual comparison between LR and the HR images generated by pretrained SRGAN model, as demonstrated in Figure 1. The down-sampled LR images lose much of the detail and clarity that is present in the originals. These pictures are then upscaled by a factor of 4 using SRGAN, which brings back lots of the initial detail and improves their visual quality.

4. Conclusion

This work investigated the use of the SRGAN model in this project for image super-resolution, concentrating on creating high-quality images from low-quality inputs. The author laid a strong basis for the SRGAN generator by training the SRResNet model first. Both the content fidelity and visual plausibility of produced pictures were improved during subsequent adversarial training. According to tests in this work, hyperparameters are changed including the number of training epochs, learning rate, and batch size systematically to find out their influence on the model performance measured by PSNR and SSIM metrics. It can be seen from the findings that SRGAN is good at balancing content fidelity and perceptual quality especially when it is trained with an optimal set of hyperparameters. The best results were observed with a learning rate of 1e-4, a batch size of 16, and 50 training epochs, which provided the highest PSNR and SSIM values across the datasets tested. Also, visual comparisons have shown that fine details can be recovered by SRGAN while producing more visually satisfying images than traditional interpolation methods do. However, there is still much to be desired. Future studies may concentrate on further adjustment of hyperparameters, exploring different network architectures as well as applying SRGAN onto wider range of diverse difficult datasets so that it could push beyond current limits in area of image super-resolution. In conclusion, this study explores the potency of SRGAN in addressing challenges associated with image super-resolution and establishes a solid groundwork for ongoing advancements in this domain.

References

[1] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., et, al. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4681-4690.

[2] Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J. H., & Liao, Q. (2019). Deep learning for single image super-resolution: A brief review. IEEE Transactions on Multimedia, 21(12), 3106-3121.

[3] Anwar, S., Khan, S., & Barnes, N. (2020). A deep journey into super-resolution: A survey. ACM Computing Surveys, 53(3), 1-34.

DOI: 10.54254/2755-2721/83/2024GLG0065

[4] Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., & Bharath, A. A. (2018). Generative adversarial networks: An overview. IEEE signal processing magazine, 35(1), 53-65.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 770-778.

[6] Chen, H., He, X., Qing, L., Wu, Y., Ren, C., Sheriff, R. E., & Zhu, C. (2022). Real-world single image super-resolution: A brief review. Information Fusion, 79, 124-145.

[7] Lin, T. Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. In European Conference of Computer Vision, 740-755.

[8] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings eighth IEEE international conference on computer vision. 2, 416-423.

[9] Huang, J. B., Singh, A., & Ahuja, N. (2015). Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE conference on computer vision and pattern recognition, 5197-5206.

[10] Hore, A., & Ziou, D. (2010). Image quality metrics: PSNR vs. SSIM. In international conference on pattern recognition, 2366-2369.