Optimizing Single Image Super-resolution and upscaling for resource-constrained computing environments

Tanveer Patil [1], Tanuja Khatavkar[1], Vaishali Gongane[1]

[1]Department of Electronics and Telecommunication, Pune Vidyarthi Griha's College of Engineering and Technology & G K Pate (Wani) Institute of Management, (ENTC), Pune, Maharashtra, India, tanveersantosh30@gmail.com, tsk_entc@pvgcoet.ac.in, vug_entc@pvgcoet.ac.in

Abstract

This research tackles the challenge of democratizing deep learning, focusing specifically on Single Image Super Resolution (SISR). In response to the inherent exclusivity stemming from high computational demands, this study introduces an optimized approach that harnesses the power of sub-pixel convolution networks. Specifically designed for everyday desktop GPUs, our methodology emphasizes computational efficiency and resource optimization, enabling the implementation of SISR without the need for specialized hardware. This work contributes to the broader mission of democratizing AI by optimizing the Efficient Sub-Pixel Convolutional Network (ESPCN) model for a standard PC with modest specs. The goal was to achieve a PSNR between 24-30 dB, surpassing traditional interpolation methods. The model was scaled to attain a PSNR of 27.98 for 1000 images and maintained a high value of 27.88 for 2500 images. This demonstrates the model's superior performance on resource-constrained PCs, bridging the gap between advanced AI and everyday computing.

1. Introduction

DEEP learning (DL) [1] is a branch of machine learning algorithms that aims at learning the hierarchical representations of data. Deep learning has shown prominent superiority over other machine learning algorithms in many artificial intelligence domains, such as computer vision [2], speech recognition [3], and natural language processing [4]. Super-resolution (SR) refers to the task of restoring high resolution images from one or more low-resolution observations of the same scene. According to the number of LR images, the SR can be classified into single image super resolution (SISR) and multi-image super-resolution (MISR) [5].

Image processing is the application of procedures to an image in order to enhance it or derive useful information from it. It is a type of signal processing where an image is supplied; and the output is either the picture or its characteristics/features [6]. One of the most well-known issues in the field of computers is Single Image Super-Resolution. It is particularly challenging to obtain a high-resolution image from its low-resolution equivalent when there is little to no information available. For this reason, deep learning models are mostly trained on big data sets and high-end computers. The ability of deep learning to absorb synthetic data and perform well during reconstruction has long been demonstrated. In this study, the primary objective centered on democratizing access to Image Super Resolution (ISR) by fine-tuning a Deep Learning Image Upscaling AI Model to operate efficiently on low to mid-specification personal computers. In this study, we present a Deep Learning model for Single-Image Super-Resolution that can up sample an input image with low resolution by a factor of three. In order to accomplish this, we will be utilizing a relatively standard personal computer system with an NVIDIA GTX 1650 and an Intel Core i5-9300H. In this research, we use the Deep Learning method of Efficient Sub-Pixel Convolutional Neural Networks (ESPCN) to increase the resolution of a given Low Resolution (LR) Image by up to 3 times (a satisfactory PSNR value in the range of 24dB to 30dB for 2,048 x 1,080-pixel images) and achieve a High Resolution (HR) Image while attaining better performance as compared to the classical Interpolation approach.

2. Related Work

The project draws inspiration from three websites (https://waifu2x.udp.jp/, https://letsenhance.io/ and https://github.com/bloc97/Anime4K), each reflecting the growing popularity of anime as a form of entertainment.

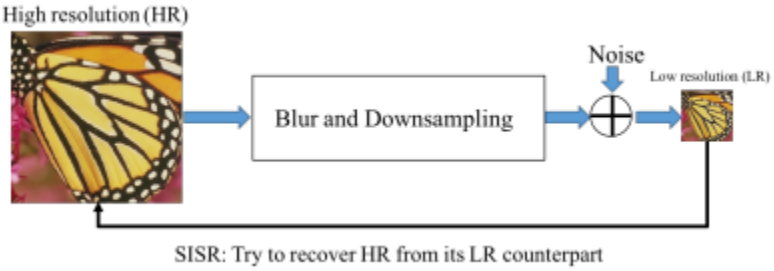


Fig. 1 Sketch of the overall framework of SISR [5]

Despite this surge in interest, existing platforms often fall short in meeting widespread demand, leading to compromises in picture quality. A notable solution to this challenge surfaced through our discovery of Anime4k, a real-time quality enhancer. Recognizing the broader implications of image and video quality limitations, not only in entertainment but across various sectors, we identified a significant gap in the medical field. In a realm where image processing remains at a nascent stage, particularly in critical areas like healthcare, we believe that the quality of visual data should not be a hindrance to delivering world-class treatment. Consequently, our research focus shifted to Single Image Super-Resolution (SISR), aiming to address these pivotal concerns. The framework of SISR is shown in the Fig. 1.

ESPCN, or Efficient Sub-Pixel Convolutional Network, is a method employed in super-resolution tasks, particularly in the realm of image processing. This technique is characterized by its post-up sampling approach, where feature extraction is conducted in a lower resolution space. The distinctive feature of ESPCN lies in its utilization of sub-pixel convolution, a method that replaces traditional deconvolutional layers. This substitution optimizes the up-sampling process, enhancing computational efficiency and contributing to the overall effectiveness of the super-resolution model. Several models underwent careful consideration before the selection of the ESPCN model. The decision-making process involved a thorough evaluation of various contenders, each scrutinized for its strengths and limitations. The intricate comparison of these models played a crucial role in identifying the ESPCN model as the optimal choice for our specific requirements.

The enhancement of the quality of an image can be at times very subjective depending on the viewer's perspective. Which method provides the best results when it comes to image enhancement can vary from person to person as an opinion. Therefore, it is quintessential to establish an empirical measure to compare the effects of enhancement algorithms on various images and the quality of the image. In this project, we propose a similar quantitative measure- Peak signal-to-noise ratio (PSNR).

The term peak signal-to-noise ratio (PSNR) [5] is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. Because many signals have a very wide dynamic range, (ratio between the largest and smallest possible values of a changeable quantity) the PSNR is usually expressed in terms of the logarithmic decibel scale. PSNR is one of the ideal ways to quantify the reconstruction quality of images subject to irreversible compression from data encoding. Typical values of PSNR are between 30-50dB, for a bit depth of 8-bits.

The formula for PSNR is as follows:

$$MSE = \sum_{M,N} \frac{\{I_1(M,N) - I_2(M,N)\}^2}{M \times N} \quad (1)$$

M and *N* are the number of rows and columns in the input images.

To calculate the PSNR:

$$PSNR = 10 \log_{10} \left(\frac{R^2}{MSE} \right) \quad (2)$$

In equation (2), *R* is the maximum fluctuation in the input image data type. For example, if the input image has a

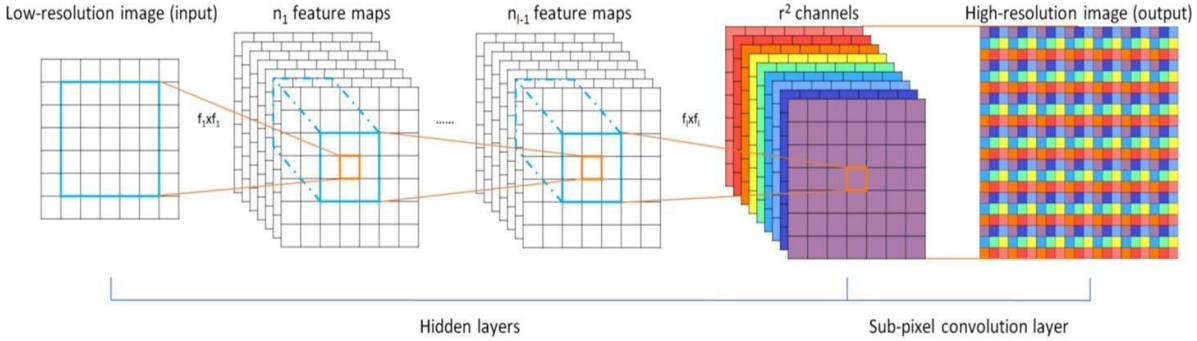


Fig. 2 Sub Pixel Convolution [5]

double-precision floating-point data type, then *R* is 1. If it has an 8-bit unsigned integer data type, *R* is 255. Different approaches exist for computing the PSNR of a color image. Because the human eye is most sensitive to luma information, you can compute the PSNR for color images by converting the image to a color space that separates the intensity (luma) channel, such as YCbCr. The Y (luma), in YCbCr represents a weighted average of R, G, and B. G is given themost weight, again because the human eye perceives it most easily. Compute the PSNR only on the luma channel. Both MSE (Mean Square Error) and PSNR are used to compare the quality of reconstructed images. MSE represents the cumulative squared error between the reconstructed image and the original image. The lower value of MSE indicates that the error is low. Yhang et al. [5] succinctly reviews recent deep learning advances in Single Image Super-Resolution (SISR). It categorizes works into architecture simulation and optimization objectives, highlighting limitations and presenting representative solutions. The paper concludes with insights into current challenges and future trends in SISR, emphasizing the superior precision of deep learning over traditional methods. Sub-pixel convolution works by converting depth to space, as seen in Fig. 2. Pixels from multiple channels in a low-resolution image are rearranged to a single channel in a high-resolution image. To give an example, an input image of size 5×5×4 can rearrange the pixels in the final four channels to a single channel, resulting in a 10×10 HR image. [5] A similar process of sub-pixel convolution takes place with the input PNG image where the image is upsampled by a factor of 3. The processed image is the plot into a graph for further comparison with the original image as well as with a similar image obtained by performing Bicubic Interpolation. Table 1

Yogeshvari et al. [10] explores the efficient Super-Resolution Convolutional Neural Network (SRCNN) and extends to various networks, including Generative Adversarial Networks (GAN). It emphasizes the balance between quality and speed but notes the impractical training demands of SRCNN and SRGAN due to high computational needs.

Table 1 compares the PSNR values in dB of the models on various datasets. This work in [13] focuses on a Generative Adversarial Network (GAN) Deep Learning Model with a sophisticated perceptual loss function. The model, capable of inferring photo-realistic images for 4x upscaling, emphasizes the preservation of fine textures.

3. Experimentation details

ESPCN (Efficient Sub-pixel Convolutional Network) which is a post-up sampling Super-resolution method is proposed. In post-up sampling, feature extraction is done in the lower resolution space. This reduces the computation

significantly as up sampling is only done at the end. Thus, the ESPCN model implements a sub-pixel convolution method which is used to replace the deconvolutional layers. During the evaluation, a publicly available benchmark datasets including the Timofte dataset widely used by SISR papers [7, 8, 9] which provides source code for multiple methods, 91 training images and two test datasets Set5 and Set14 which provides 5 and 14 images is used.
The Berkeley segmentation dataset BSD300 and BSD500 which provides 100 and 200 images for testing and the super texture dataset which provides 136 texture images. For our final models, we use 50,000 randomly selected images from ImageNet for the training. Following previous works, we only consider the luminance channel in YCbCr colour space in this section because humans are more sensitive to luminance changes. For each upscaling factor, we train a specific network [5]. The specifications of our system for implementation of ESPCN model is shown in Table 2. Table 2 provides details on the ASUS ROG Strix G531GT laptop, featuring an Intel Core i5-9300H processor (4 cores, 8 threads, 2.40GHz base frequency, and benchmark scores of 949/3307 in Geekbench 5). Additionally, it includes information on the NVIDIA GPU with 896 CUDA cores, a boost clock of 1545MHz, and 4GB GDDR5 VRAM.

Table 1 Model Comparison Chart [5]

Dataset	Scale	Bicubic	SRCNN	TNRD	ESPCN
Set5	3	30.39	32.75	33.17	33.13
Set14	3	27.54	29.30	29.46	33.13
BSD300	3	27.21	28.41	29.46	28.54
BSD500	3	27.26	28.48	29.46	28.64
Super Texture	3	25.40	26.60	26.66	26.70

Table 2 Specifications of the system

Name	Vendor	Specification
System Notebook Type: Laptop	ASUSTek Computer Inc	Asus ROG Strix G531GT(8Gb)
CPU (Cores, Threads, MaxFreq, Benchmark)	Intel	Intel Core i5-9300H, 2.40 GHz, 4 Core 8 Thread, 4100Mhz (Single Core / Multi-Core): 949/3307 [Geekbench 5]
GPU (Cuda Cores, BoostClock, VRAM)	NVIDIA	Cuda Cores: 896, Boost Clock: 1545 Mhz, VRAM: 4Gb GDDR5

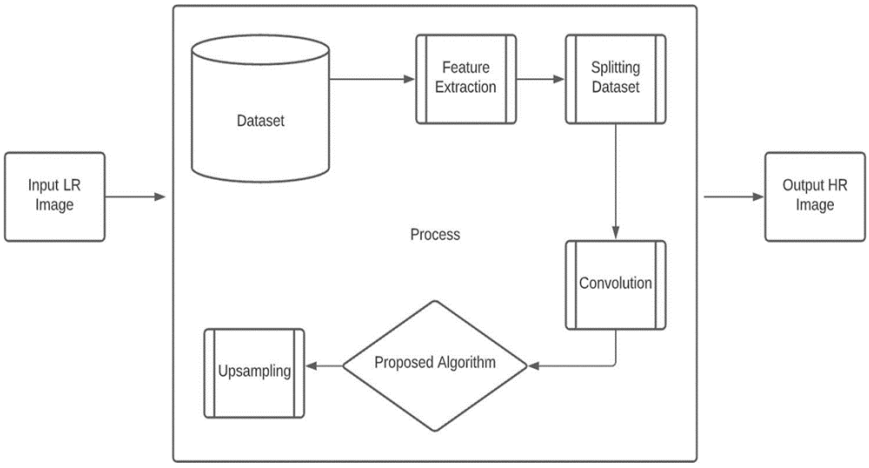


Fig. 3 Block diagram of the proposed system

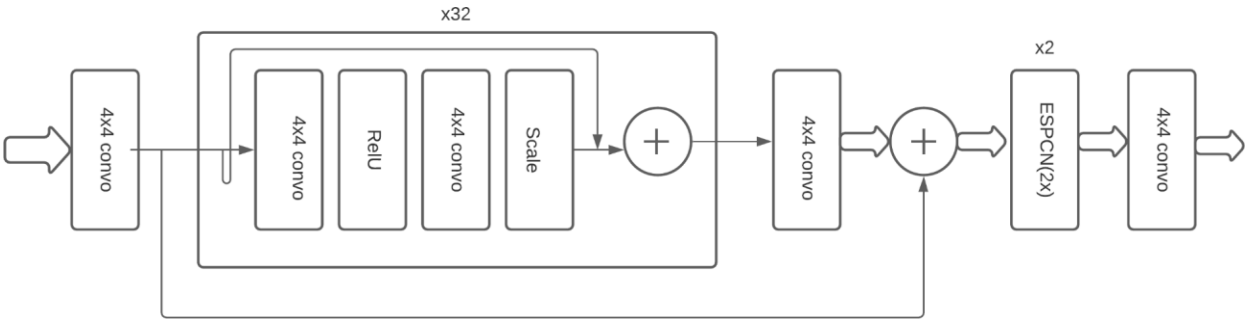


Figure 4: Block Diagram of the Algorithm

As shown in Fig. 3, the first block is the input block where the user will upload a Low-Resolution image for upscaling. The format of this image is in .png format for further processing. The second block of the image is the most crucial part, and this is where all the processing takes place. In this block, the images are stored in the database for further processing. This is then fed to the Deep Learning algorithm which then extracts the features of the image and further processing takes place. The proposed algorithm is the ESPCN algorithm which uses post-up-sampling for computing. In the next block, we split the given dataset for training and validation. As the concerned dataset contains 1000 images, we have split 800 images for training and 200 images for validation. The given image is then processed and up sampled. The output block gives the user a PNG image that has been upscaled by a factor of 3 and can be downloaded.

The proposed algorithm is depicted in Fig. 4 which is an ESPCN algorithm. In this algorithm, we have used 3 Convolutional layers for sub-pixel convolution and instead of adding deconvolution, we have replaced that layer with the layer of up sampling. This happens thrice and hence the image is upscaled by a factor of 3. The up-sampling process in the algorithm exclusively enlarges the image at its final stage, resulting in a high-resolution output. This strategic approach, performing up-sampling only once in the concluding phase, significantly reduces computational demands by employing small-size feature maps throughout the network. The ESPCN Model, trained on the D2FK Dataset, utilizes 3450 high-resolution images with a resolution of 2,048 x 1,080 pixels (termed "2K"). This dataset is meticulously divided into training (2760 images), testing (345 images), and validation (345 images) subsets. Specifically designed to foster research in realistic image super-resolution challenges, the D2FK Dataset introduces scenarios of degradation. Moreover, the ESPCN model enhances computational efficiency by incorporating a sub-pixel convolution method, effectively replacing deconvolutional layers for up-sampling. This approach not only

optimizes efficiency but also mitigates the checkerboard issue associated with conventional deconvolution methods. Table 3 shows the parameters of the image considered for experimentation.

Table 3 Image Parameters

Parameter	Specification
Input Image type	PNG (image/png)
Output Image	PNG (image/png)
Training Image Dataset	DIV2k Image Dataset
Dataset Size	10 Gb
Dataset Division	For Training: 800 Images. For Testing: 100 Images. For Validation: 100 Images.
Training Dataset Image Resolution	2048 x 1080 px
Upscale Factor	3

4. Results

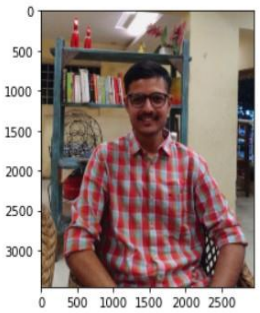
During the duration of Phase-I, we were successfully able to train the ESPCN Deep Learning Model. The model was trained, tested, and the validation, all was done and completed using the DIV2K image dataset. The PSNR of the generated image, when done in Google Colab, was found out to be 27.881190. We trained this model on 17x17 patches of the HR images. The original resolution of these images was 2048x1080px. The training loss at the first epoch was found out to be 0.0043 but later dropped to 0.0028 on the final epoch (Epoch-50). A total of 332 minutes was required to train the entire ESPCN Model on the Google Colab Notebook along with the use of the GPU provided. The PSNR of the generated image, when done on the Native machine, was found out to be 27.981799. We trained this model on 28x28 patches of the HR images. The original resolution of these images was 2048x1080px. The reason for the increase in the size of the patch of the image was the availability of greater computational power compared to Google Colab Notebook. Due to this, we were able to exploit the on-system GPU and increase the number of epochs to 60. The training loss at the first epoch was found out to be 0.0051, which was considerably high compared to the Colab Notebook, but later dropped to 0.0028 in the final epoch (Epoch-60). A total of 212 minutes was required to train the entire ESPCN Model on the native system. JupyterLab was also interfaced with the on-system GPU using the NVIDIA CUDA TOOLKIT and the CUDNN Tool. Table 4 compares the performance of ESPCN on three different hardware configurations. Fig. 5 shows the improved PSNR of test image. Table 5 compares the various parameters for three configuration of hardware.

Table 4 Results

Sr.	Parameters	Google Colab	Native System	Native System-1
1	PSNR of Test Image [HR](dB)	27.881190	27.981799	27.88123
2	PSNR of Test Image (Bicubic Interpolated) (dB)	27.474722	27.475621	27.457621
3	Image Resolution(px)	2048x1080	2048x1080	2048x1080
4	Training done on Image Patches (pixels)	17x17	28x28	32x32
5	Loss in the first Epoch	0.0043	0.0051	0.0044
6	Loss in the final Epoch	0.0028	0.0028	0.0030
7	Time taken to train the model(min.)	332	212	1042

```
In [4]:
1 from matplotlib import pyplot as plt
2 plt.imshow(cv2.cvtColor(HR_image, cv2.COLOR_BGR2RGB))

Out[4]: <matplotlib.image.AxesImage at 0x213d867cb20>
```




```
In [5]:
1 print("PSNR of ESPCN generated image: ", PSNR(cropped, HR_image))

PSNR of ESPCN generated image:  28.15214390009856
```

Fig. 5 Generated HR Image With a PSNR of 28.1521430000

Table 5 Value Look-up table for Software testing

Parameters	**(Google Colab)**	Native System-1	Native System-2
Epoch-1 Time	397s	288s	1036s
Epoch-1 Loss	0.0043	0.0051	0.0044
Epoch-1 PSNR	26.9361	26.0036	26.7721
Epoch-60 Time	398s	211s	972s
Epoch-60 Loss	0.0028	0.0028	0.0030
Epoch-60 PSNR	32.0210	31.5154	30.3192
PSNR of Test Image (HR)	27.8811	27.981799	28.1521439
PSNR of Test Image (Bicubic Interpolated)	27.4747	27.475621	26.875621

5. Conclusion:

In summary, our project successfully implemented a Deep Learning Model for Image Enhancement and Super Resolution, achieving the targeted upscale factor of 3. Our initial phase involved a comparative analysis between the ESPCN Model and a traditional Interpolation method. Notably, our experimentation yielded a notable Peak Signal-to-Noise Ratio (PSNR) of 27.981799 across a batch of 1000 images, each sized at 2048x1080 pixels. The project involved acquiring both theoretical and practical insights, identifying the necessary infrastructure, and seamlessly translating theoretical knowledge into practical application. Notably, we observed a substantial improvement in PSNR by 0.100609, indicating enhanced image quality, and a significant reduction in the model's training time by 120 minutes. As a pivotal advancement, our subsequent optimization endeavors enabled the seamless upscale to 2500 images of identical dimensions (2048x1080 pixels), all while consistently maintaining a commendable PSNR performance of 27.88123. This accomplishment was achieved while rigorously adhering to the specified computational constraints inherent in the designated PC hardware specifications. These outcomes underscore the technical success and efficiency gains achieved through the project's development and implementation phases. The future of single-image super-resolution involves advancements in deep learning architectures, attention mechanisms, real-time applications, and domain-specific approaches. Adversarial training, transfer learning, and multimodal integration are key focuses. Improved evaluation metrics, ethical considerations, and user interaction are also important aspects shaping the field's evolution. Ongoing developments will continue to impact these trends.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," nature, vol. 521, no. 7553, p. 436, 2015.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1097– 1105.

[3] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," IEEE Signal Processing Magazine, vol. 29, no. 6, pp. 82–97, 2012.

[4] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in Proceedings of the International Conference on Machine Learning, 2008, pp. 160–167

[5] W. Yang, X. Zhang, Y. Tian, W. Wang, J. Xue, and Q. Liao, "Deep Learning for Single Image Super-Resolution: A Brief Review," in IEEE Transactions on Multimedia, vol. 21, no. 12, pp. 3106-3121, Dec. 2019, DOI: 10.1109/TMM.2019.2919431.

[6]Sehgal, R., Gupta, N., Tomar, A., Sharma, M. D., & Kumaran, V. (2022). Smart Electrical and Mechanical Systems. Academic Press.

[7] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. arXiv preprint arXiv:1508.02848, 2015

[8] 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 (TPAMI), 2015

[9] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deeply improved sparse coding for image super-resolution. arXiv preprint arXiv:1507.08905, 2015.

[10] Makwana Yogeshvari & Patel, Pranay & Swadas, Prashant. (2020). Single Image Super- Resolution using Deep Learning: A Survey. 7. 22-27. 10.21090/ijaerd.83398.

[11] X. Jia, "Image recognition method based on deep learning," 2017 29th Chinese Control and Decision Conference (CCDC), 2017, pp. 4730-4735 doi:10.1109/CCDC.2017.7979332.

[12] Zhang, Yulun & Tian, Yapeng & Kong, Yu & Zhong, Bineng & Fu, Yun. (2018). Residual Dense Network for Image Super-Resolution. 2472-2481. 10.1109/CVPR.2018.00262.

[13] C. Ledig et al., "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 105-114, DOI: 10.1109/CVPR.2017.19.