Edge Enhanced Three-Stage Deep Network for Image Super-Resolution

Farhan Sadik[1]

Abstract

Single Image Super-Resolution (SISR) poses an ill-posed problem, involving the generation of high-resolution images from low-resolution input. This implies a scenario with a higher number of variables to address compared to the equations available for solving. Historically, SISR challenges have been addressed through interpolation techniques. However, the advent of deep learning has significantly enhanced super-resolution performance. Data-driven approaches leverage multiple convolution layers to acquire feature representations from diverse images, enabling more accurate interpolation of low-resolution data. However, in conventional deep learning methods, low-resolution images contain high-frequency information that is uniformly handled across all channels. As a result, these techniques do not fully take advantage of the high-frequency information available. In this paper, I reimplemented a deep learning-based architecture that exploits high-frequency information in a separate stage and incorporated that information as a loss constraint. The entire structure consists of a dual-stage design, with one stage dedicated to an end-to-end network for SISR and the other dedicated to retaining high-frequency information. Moreover, I incorporated a third stage, where the edge information of the image is extracted through the Sobel kernel and incorporated as a loss to boost the overall end-to-end performance. The proposed method achieved a PSNR (Peak Signal-to-Noise Ratio) of 30.39 dB on the DIV2K dataset. Moreover, Incorporating the Sobel loss also increased the PSNR to 30.87 dB.

[1]Purdue University, West Lafayette, Indiana, USA. Correspondence to: Farhan Sadik <fsadik@purdue.edu>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

1. Introduction

Reducing an image to its low resolution can greatly enhance data transfer efficiency. However, at the user end, it needs to be reconstructed to high resolution. Such techniques find applications in diverse fields. One such area is MRI acquisition which involves reconstructing high-resolution data from low-resolution noisy images (Yang et al., 2018). Another example can be smartphone cameras where hardware might have limitations in resolution, but post-processing super-resolution techniques can significantly enhance image quality (Chen et al., 2019).

In this paper, I reimplemented the full work of (Han et al., 2021), where a dual-stage deep neural network is incorporated for SISR task. A two-stage network is employed where the first stage is an end-to-end network that converts a low-resolution (LR) input to an LR output with high-frequency information. Specifically, residual channel attention blocks are integrated to perform this task, i.e., retaining high-frequency information. During the second stage, the LR image, containing high-frequency information, is transformed into a high-resolution (HR) image while preserving the high-frequency details. A dual-stage loss is employed to collectively acquire knowledge about high-frequency information across both stages. The authors did not supply the code, prompting me to re-implement it entirely. Additionally, I expanded on the work by introducing an extra stage that enhanced edge information through the application of a loss constraint based on the Sobel operator.

2. Review of paper to implement or extend

2.1. Storyline

High-level motivation/problem Single Image Super Resolution is an intriguing field, as it involves an ill-posed problem where you have more variables than equations to solve for (Yang et al., 2019). In this particular challenge, the task is to reconstruct a high-quality image, which involves a larger number of pixels (variables) from low-resolution images, where there are fewer pixels (equations) to guide the process. Due to limitations in hardware capabilities, most of the information is transmitted after compressing the data. This expedited data transfer comes at the cost of losing some data. Fast and accurate Single Image Super Resolution

(SISR) facilitates quicker transmission and ensures a precise portrayal of the image at the user's end.

Prior work on this problem Existing Single Image Super Resolution (SISR) methods can be classified into three main categories: interpolation-based approaches (Chang et al., 2004; Bevilacqua et al., 2012), reconstruction-based algorithms (Chakrabarti et al., 2007), and learning-based techniques (Glasner et al., 2009; Huang et al., 2015; Freedman & Fattal, 2011; Johnson et al., 2016). Both interpolation-based techniques and reconstruction-based algorithms experience reduced efficiency and increased sensitivity to scaling factors. Deep Learning (DL), as a subset of learning-based methods, grasps the nonlinear relationship between low-resolution input and high-resolution output using convolutional neural networks (CNNs). Don et al. (Dong et al., 2016) introduced the Super Resolution CNN (SR-CNN), which utilizes basic CNN operations for extraction, non-linear mapping, and reconstruction of images in alignment with sparse coding (Yang et al., 2010) to learn high-resolution pixels. More intricate networks such as LapSRN (Lai et al., 2017), DRRN (Tai et al., 2017), SR-ResNet (Ledig et al., 2017), EDSR (Lim et al., 2017) and RCAN (Zhang et al., 2018b). were employed to reconstruct superior-quality images, involving the utilization of large models with increased parameters to grasp complex non-linear mappings. In recent times, attention-based networks have been employed in super-resolution tasks such as non-local attention learning (Zhang et al., 2019) which helps to retain low-level features.

Research gap Nonetheless, blindly enhancing the complexity and depth of the network marginally improves performance, but the associated computational costs may hinder many potential applications (Han et al., 2021). Furthermore, a majority of CNN-based approaches overlook the significance of high-frequency features (Han et al., 2021). Consequently, many of the reconstructed images exhibit significant blurriness (Han et al., 2021).

Contributions In this paper (Han et al., 2021), a Two-Stage Network (TSN) is proposed to reconstruct high-resolution images from low-resolution images by exploiting the high-frequency information present in the high-resolution images. The proposed TSN in this paper captures high-frequency details from high-resolution images and then learns to reconstruct high-resolution images from low-resolution inputs.

2.2. Proposed solution

To manage the network's complexity while also extracting high-frequency details, this paper proposes the use of Residual Channel Attention Groups (RCAGs) (Zhang et al., 2018b) to obtain a high-frequency low-resolution image from a high-frequency high-resolution image. In the first stage of the network, the low-resolution image is transformed into a low-resolution image that retains high-frequency information. Subsequently, in the second stage, this low-resolution, high-frequency image is further transformed into a high-resolution image with enhanced high-frequency details. Additionally, a two-stage learning loss is introduced to guide the super-resolution process by simultaneously controlling the learning in both stages.

The first stage is divided into two simultaneous operations. Firstly, several RCAGs followed by a downscale module (collectively they named it H_{down}) are used to transform the high-resolution images (I_{HR}) to low-resolution images with high-frequency information ($I_{LR'}$) which will be later used for calculating the loss.

$$I_{LR'} = H_{down}(I_{HR}) \tag{1}$$

Secondly, the network is initiated by a convolution layer (H_{SF}) to extract shallow feature (F_o) from the input LR image (I_{LR}).

$$F_o = H_{SF}(I_{LR}) \tag{2}$$

This shallow feature, F_o is passed through four consecutive RCAGs (H_{RCAG4}) to obtain the intermediate low-resolution high-frequency image (F_{HF}).

$$F_{HF} = H_{RCAG4}(F_o) \tag{3}$$

In the second stage, the low-resolution high-frequency image (F_{HF}) is passed through four consecutive RCAGs to extract more deep features (F_{DF}). Finally, in the reconstruction layer, F_{DF} is upsampled to get the super-resolution image (I_{SR}).

$$F_{DF} = H_{RCAG8}(F_{HF}) \tag{4}$$

$$I_{SR} = H_{\uparrow}(F_{DF}) \tag{5}$$

$I_{LR'}$ from Equation (1) is used to calculate the joint loss given by

$$L_\Theta = \frac{1}{N}\sum_{n=1}^{N} \lambda \|I_{HR} - I_{SR}\|_1 + \|I_{HF} - I_{LR'}\|_1 \tag{6}$$

where, $\theta \, \epsilon \, \Theta$ are the network parameters and I_{HR} is the high resolution ground truth image.

Table 1. Effects of different modules. We report the PSNR on Set5 datasets in the 200 epoch.

Method	R_a	R_b	R_c	R_d	R_e
Two-stage		✓	✓	✓	✓
RCAG			✓	✓	✓
λ=1		✓	✓		
λ=5				✓	
λ=10					✓
PSNR	32.23	32.45	33.16	33.23	33.21

Figure 1. Table 1 is taken from page 5, section 4.2 of the original paper (Han et al., 2021) as a screenshot.

2.3. Claims-Evidence

Claim 1 The introduction of RCAG to transform the high-resolution space into a low-resolution space led to a significant improvement in the network's performance.

Evidence 1 Table 1 from the paper shows that after incorporating the loss where RCAG is used to calculate $I_{LR'}$, the performance increased by 1 dB PSNR (R_c in Figure 1).

Claim 2 The paper criticized existing methods because of higher complexity with marginal performance and proposed a model with low parameters but high performance.

Evidence 2 Table 3 (Figure 2) from the paper represents that TSN has significantly lower parameters but its performance is better than the existing state-of-the-art methods.

Table 3. Computational and parameter comparison (2X) Set5.

	EDSR	MemNet	MSRN	SeaNet	TSN
Para.	43M	677k	6M	8M	9.6M
PSNR	38.11	37.78	38.07	38.08	38.19

Figure 2. Table 3 is taken from page 5, section 4.4 of the original paper (Han et al., 2021) as a screenshot.

Claim 3 Methods described in existing literature tend to overlook high-frequency details, leading to blurry artifacts in the reconstructed images. The method presented integrates high-frequency information into both the network architecture and the loss function, resulting in higher-quality images.

Evidence 3 As illustrated in Figure 2 from the paper (Figure 3), the proposed TSN exhibits notably superior sharpness compared to other methods at a 4x scaling factor. The zoomed-in image of the selected section distinctly highlights the inability of most of the super-resolution methods to capture sharp details.

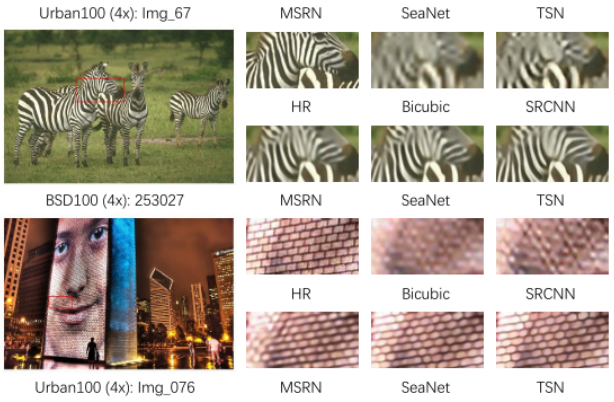


Figure 2. Visual comparison for 4x SR with BI model.

Figure 3. Figure 2 is taken from page 6, section 4.3 of the original paper (Han et al., 2021) as a screenshot.

2.4. Critique and Discussion

The paper is skillfully written, with a particularly impactful storytelling element that captivates its readers. The most enlightening aspect was their assertion regarding the effectiveness of employing attention blocks, substantiated through experimental evidence. Moreover, the claims made throughout the papers are justified through proper tables or figures.

However, the paper suffers from several issues. First of all, the implementation details of this paper are vague. The authors did not mention the number of epochs to train, and the training time, so that the reader could expect the training time while re-implementing their work. Nonetheless, the paper also failed to provide a proper justification for some elements. For example, the authors used 15 Residual Channel Attention Blocks without offering any rationale for their choice of this particular quantity.

Apart from that, L_1 loss is used for optimization instead of L_2, which is more common in vision applications. They could provide experimental results showing why L_1 loss is better than L_2 loss in this case.

3. Review of 2^{nd} paper

3.1. Storyline

High-level motivation/problem Single Image Super Resolution (SISR) refers to the process of generating a high-resolution image from a low-resolution input. The problem is ill-posed as one low-resolution image can have multiple high-resolution solutions. Nonetheless, the majority of Single Image Super Resolution (SISR) techniques demand significant memory resources, leading to subpar performance which hinders their potential for commercialization. This paper aims to work on putting emphasis on second-order

channel attention which is different from all the previous works as well as focuses on improving the visual quality of the images.

Prior work on this problem The strong capability of Convolutional Neural Networks (CNNs) in representing features has made them widely utilized in single-image super-resolution tasks. CNNs function as a method to map features from low-resolution input images to high-resolution output images. While CNNs (Zhang et al., 2018a), (Zhang et al., 2018b), (Zhang et al., 2018c) have attained cutting-edge performance in Single-Image Super-Resolution (SISR), they encounter issues such as generating redundant information or hallucinations. Nonetheless, the reconstruction performance of the CNNs is a great improvement over the previous interpolation methods such as non-local similarity prior (Zhang et al., 2012) and sparsity prior (Dong et al., 2011).

Research gap Most of the CNN-based SISR methods incorporate deeper networks for unrolling the images but that in turn avoids feature relation between intermediate layers. Therefore, these methods do not fully exploit the information that lies within the low-resolution images which leads to poor performance. Moreover, deepening the layers only tends to learn the high-level features hindering the representational power of CNNs. Nevertheless, transitioning away from CNNs isn't practical, as interpolation-based methods also require extensive time for optimization. Additionally, these techniques quickly become ineffective when the model processes input data with different statistics than the prior dataset.

Contributions In this work (Dai et al., 2019), the author introduces a second-order attention network (SAN) for more robust feature expression and feature correlation between intermediate layers. Specifically, second-order channel attention is proposed to rescale feature statistics and enhance discriminative learning power. To build a deeper network, a non-locally enhanced residual group (NLRG) is utilized that shares contextual information. Additionally, skip connections are utilized to transfer learned representations across numerous layers within the neural network. These connections enable the network to effectively utilize the bypassed skip connections, facilitating the flow of information from low-resolution input images to the subsequent layers and simplifying the training process.

3.2. Proposed solution

The proposed SAN network can be divided into four parts. 1) shallow feature extraction, 2) non-locally enhanced residual group (NLRG) based deep feature extraction, 3) up-scale module, and 4) reconstruction part. Let, I_{LR} and I_{SR} are the input and output of the network. At first, the network extracts shallow feature F_0 from the LR input

$$F_0 = H_{SF}(I_{LR}) \tag{7}$$

where H_{SF} (.) corresponds to the convolution operation. Then the shallow features are fed into NLRG-based deep feature extraction and the deep features F_{DF} are found.

$$F_{DF} = H_{NLRG}(F_0) \tag{8}$$

Here, H_{NLRG} enlarges feature learning by sequentially passing the input through several non-local modules. The output of NLRG is then upscaled via upscale module $F_{\uparrow}$.

$$F_{\uparrow} = H_{\uparrow}(F_{DF}) \tag{9}$$

In the reconstruction part, the upscaled features are converted to the super-resolution output via one convolution block H_R.

$$I_{SR} = H_R(F_{\uparrow}) = H_{SAN}(I_{LR}) \tag{10}$$

Here, H_{SAN} is the overall SAN network.

Optimization is one important aspect of neural networks and this work proposes a n_1-norm loss function between the output super-resolution image and the ground truth for this purpose.

$$L_{\Theta} = \frac{1}{N}\sum_{i=1}^{N} \lambda \| I^i_{HR} - H_{SAN(}I^i_{LR})\|_1 \tag{11}$$

where, $\theta \, \epsilon \, \Theta$ are the network parameters and I^i_{HR} is the high resolution ground truth image.

3.3. Claims-Evidence

Claim 1 According to the author, the proposed second-order channel attention (SOCA) adaptively learns features using second-order feature statistics instead of first-order ones which in turn puts emphasis on informative feature and improve discriminative learning ability rather than the other methods that incorporate first-order statistics.

Evidence 1 Figure 4 demonstrates that incorporating second-order statistics (R_e = 32.16 dB) indeed enhances image quality and captures more intricate details compared to first-order (R_d = 32.12 dB).

Table 1. Effects of different modules. We report the best PSNR (dB) values on Set5 ($4\times$) in 5.6×10^5 iterations.

	Base	R_a	R_b	R_c	R_d	R_e	R_f	R_g	R_h	R_i
RL-NL(before SSRG)		✓					✓	✓	✓	✓
RL-NL(after SSRG)			✓				✓	✓	✓	✓
share-source skip connection (SSC)				✓				✓	✓	✓
First-order channel attention (FOCA)					✓				✓	
Second-order channel attention (SOCA)						✓				✓
	32.00	32.04	32.06	32.07	32.12	32.16	32.08	32.10	32.14	32.20

Figure 4. Table 1 is taken from page 7, section 4.2 of the original paper (Dai et al., 2019) as a screenshot.

Claim 2 The authors claimed that the proposed SAN method performs better than RCAN (Dong et al., 2016) on datasets containing abundant texture information, such as Set5, Set14, and BSD100. However, it performs slightly worse on datasets like Urban100 and Manga109, which are characterized by extensive repeated edge patterns.

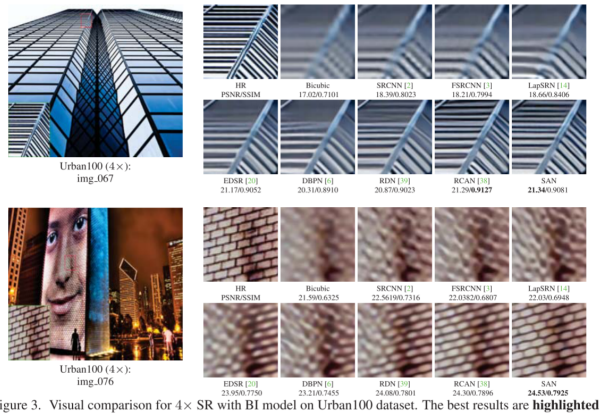


Figure 3. Visual comparison for 4× SR with BI model on Urban100 dataset. The best results are **highlighted**

Figure 5. Figure 3 is taken from page 7, section 4.3 of the original paper (Dai et al., 2019) as a screenshot.

Evidence 2 Figure 5 demonstrates that for the Urban100 dataset, the proposed method does not perform consistently better than RCAN. For example, for image "067", the SSIM of RCAN is better than the proposed SAN method. On the other hand, for other datasets, the proposed method outperforms other works and shows slightly better performance than RCAN.

Claim 3 The authors stated that the proposed SAN possesses a smaller parameter count than RCAN but a larger one compared to other techniques like MemNet and NLRG.

Evidence 3 We can see from Figure 6 that the proposed SAN method has 15.7M parameters which is less than RCAN (16M) but much higher than MemNet (677K) and NLRG (330K).

Table 4. Computational and parameter comparison (2× Set5).

	EDSR	MemNet	NLRG	DBPN	RDN	RCAN	SAN
Para.	43M	677k	330k	10M	22.3M	16M	15.7M
PSNR	38.11	37.78	38.00	38.09	38.24	38.27	**38.31**

Figure 6. Table 4 is taken from page 8, section 4.5 of the original paper (Dai et al., 2019) as a screenshot.

3.4. Critique and Discussion

This paper provides an overview of deep learning methods found in the literature review, integrating them seamlessly into the narrative, leaving a strong impression on the reader. The crucial aspect of this literature lies in the detailed theoretical exploration of the proposed network, which will undoubtedly be immensely valuable for anyone attempting to replicate their research. Furthermore, they incorporated numerous blocks within the network and conducted various experiments to validate each block. They condensed their findings into a table, making it easily comprehensible for readers.

The implementation details did not specify the number of epochs or the required hardware for training. Including this information would be beneficial for future researchers, allowing them to anticipate the necessary resources and expectations when implementing the study. On a positive note, they offered the code, which is highly valued and appreciated.

In the section titled "Model Size Analysis," a spelling error was found; the authors wrote 'lless' instead of 'less'. It is advisable for the authors to exercise caution and avoid such mistakes. In summary, the paper is concisely written, even for common audiences with math backgrounds who can easily comprehend what the authors are trying to establish.

4. Review of 3^{rd} paper

4.1. Storyline

High-level motivation/problem Certain images contain multiple patches that correspond to the same pixels as depicted in Figure 7. Therefore, some details can be resolved using multiple reference images.

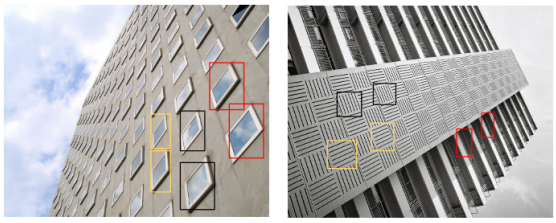


Figure 1. Examples of similar patches in images. These similar patches can help restore details from each other.

Figure 7. Figure 1 is taken from page 1, section 1 of the original paper (Lu et al., 2022) as a screenshot.

This worked as a motivation to incorporate a vision transformer for single-image super-resolution. Vision transformers utilize the inherent patches within images, and they process image input in the form of image patches. This phenomenon served as an inspiration to employ a transformer for image super-resolution, enabling the extraction of long-

term dependencies within the images. Despite the superior performance of vision transformers in single-image super-resolution compared to methods based on CNNs, their high memory consumption has limited their popularity.

Prior work on this problem Although deep learning networks have been used in single-image super-resolution, the concept of vision transformers is comparatively new. Some Recurrent Neural Network (RNN) based architectures were also deployed for this task but the reduced network parameters led them to perform poorly such as DRCN (Kim et al., 2016), SRRFN (Li et al., 2019), and IMDN (Hui et al., 2019). The transformer network came from the NLP literature, especially the element "self-attention" in the transformer which captures the long-term dependencies in sequence. However, in vision, it is represented as long-term dependencies among image patches (Dosovitskiy et al., 2020). A significant advancement occurred with SwinIR (Liu et al., 2021), employing Swin Transformers for single-image super-resolution with remarkable precision, surpassing the performance of state-of-the-art CNN techniques.

Research gap CNN-based methods incur substantial GPU costs, prompting researchers to explore RNN-based alternatives. Additionally, CNNs can leverage local features, but they do not comprehensively analyze global features. However, RNN methods perform notably worse than CNN-based approaches due to their shallower network depth. Although Swin Transformer significantly improves reconstruction performance it suffers from high memory consumption.

Contributions In this paper (Lu et al., 2022), the author proposed an efficient transformer network for image super-resolution known as ESRT (Efficient Super-Resolution Transformer). Specifically, the author proposed a hybrid model that consists of a lightweight CNN backbone (LCB) and a lightweight transformer backbone (LTB). LCB utilizes high-preserving blocks (HPBs) to adjust the feature maps with low-cost computation and extract texture details before feeding into the LTB. The LTB utilizes efficient multihead attention (EMHA) mechanisms to efficiently capture the dependencies among image patches with reduced cost.

4.2. Proposed Solution

Efficient Super-resolution transformer consists of four parts. They are shallow feature extraction, Lightweight CNN backbone (LCB), Lightweight transformer backbone (LTB), and reconstruction block. The image input is I_{LR} and the output I_{SR}, the shallow feature extractor is defined as a convolution operation (f_s).

$$F_0 = f_s(I_{LR}) \tag{12}$$

The extracted shallow feature, F_0 is passed into the LCB which consists of several High Preserving Blocks (HPBs) denoted as ζ.

$$F_n = \zeta^n(\zeta^{n-1}(...(\zeta^2(\zeta^1(F_0))))) \tag{13}$$

Here, ζ^n represents the n^{th} HPB block and F_n denotes the output of n^{th} HPB. The outputs are combined and sent to several efficient transformers (denoted as ϕ) to merge intermediate features.

$$F_d = \phi^n(\phi^{n-1}(...(\phi^2(\phi^1([F_1, F_2, ..., F_n])))))) \tag{14}$$

Here, F_d denotes the output of LTB. Finally, in the reconstruction module, the shallow feature F_0 and the output of the LTB, F_d are used jointly to reconstruct the high-resolution image.

$$I_{SR} = f(f_p(f(F_d))) + f(f_p(F_0)) \tag{15}$$

Here, f denotes the convolution layer, and f_p denotes the pixel shuffle layer.

4.3. Claims-Evidence

Claim 1 There are similar patches in a single image with the same type of pixels which means we can count them as ground truth. Therefore, for each patch, we can have multiple ground-truth images. The author claimed that similar patches can be used for ground truth. Therefore, certain texture details can be resolved using multiple reference patches. This operates in a manner akin to the way transformers function.

Evidence 1 Figure 8 demonstrates that without the transformer the PSNR drops from 32.18 dB to 31.96 dB. Therefore, the transformers are able to exploit the patches in the image and improve the reconstruction.

Case	PSNR(dB)	Parame.(K)	GPU Memory
w/o TR	31.96	554	1931M
Original TR [38]	32.14	971	16057M
1 ET	**32.18**	751	4191M
2 ET	32.25	949	6499M

Table 4. Study of Efficient Transformer (ET) on Set5 (×4). The GPU memory here refers to the cost of the model during training, which patch_size = 48*48 and batch_size=16.

Figure 8. Table 4 is taken from page 8, section 4.5.2 of the original paper (Lu et al., 2022) as a screenshot.

Claim 2 Because the Fourier Transform requires a significant amount of memory, the authors suggested using a lightweight, differentiable high-frequency filtering module (HFM) to extract high-frequency information from low-resolution input in a more efficient manner.

Evidence 2 In Figure 9, we can see that for cases 1, 2, and 3 the HFM module improves the performance with less amount of parameters.

Case Index	1	2	3	4
HFM		√	√	√
CA	√	√		√
ARFB	√	√	√	
RB				√
Parameters	658K	751K	724K	972K
PSNR	32.02dB	32.19dB	32.08dB	**32.20dB**

Table 3. Study of each component in HPB on Set5 (×4).

Figure 9. Table 3 is taken from page 7, section 4.5.1 of the original paper (Lu et al., 2022) as a screenshot.

Claim 3 While Swin Transformer (SwinIR) is effective for image super-resolution, the author asserted that it consumes more GPU memory. The proposed approach, however, presents a superior solution by enhancing image quality with fewer parameters, addressing the issue of high GPU memory consumption associated with SwinIR.

Evidence 3 As we can see from Figure 10, the number of parameters for SwinIR is higher than the proposed ESRT but the difference between PSNR is 0.01 dB. Therefore, the proposed ESRT can fully exploit the advantages of a transformer with fewer parameters.

Method	Parame.	GPU Memory	BSD100	Manga109
SwinIR	886K	6966M	29.20/0.8082	33.98/0.9478
ESRT	770K	4191M	29.15/0.8063	33.95/0.9455

Table 7. A detailed comparison of SwinIR and ESRT (×4).

Figure 10. Table 7 is taken from page 8, section 4.7 of the original paper (Lu et al., 2022) as a screenshot.

4.4. Critique and Discussion

The writing in this paper is undeniably exceptional, particularly in the introduction section. What fascinated me the most was the explanation of utilizing transformers, demonstrated in a figure that illustrates how multiple patches correspond to the same image. Therefore, the rationale underlying the transformer was evident and comprehensible.

In the related works section, the authors provided a brief overview of CNN-based research. It is recommended that the authors expand their coverage of prior studies to enhance the reader's comprehension of the field's historical context and development. Nevertheless, the sections on attention and transformers were presented in a general manner, allowing the reader to discern the underlying trend and the author's intended message.

While certain implementation specifics were missing, the paper provided a thorough examination of each network block utilized, rendering it comprehensive yet accessible to a broader audience.

5. Implementation

5.1. Implementation motivation

The authors of the paper did not share any code, prompting me to undertake the project and reimplement their work using PyTorch. The paper lacks crucial details, such as the difference image between input and output, and the rationale behind the use of l_1 loss instead of the more commonly used l_2 loss. Additionally, there is a lack of explanation regarding the utilization of RCAG. It remains unclear how attention blocks collectively work to preserve high-frequency information. The paper vaguely states the use of 15 Residual Channel Attention Blocks (RCABs) in each RCAG. The selection of this specific number of RCABs should be substantiated with several experiments to validate its effectiveness. Through the reimplementation, I will seek to answer these questions.

Additionally, I have one particular idea to implement inside the network - a sharpness enhancer feature block. This block would utilize phase images (which inherently contain edges) in the reconstruction process, employing RCAGs to incorporate them by addition or multiplication before the final concatenation. Given that many networks face issues with blurriness, I believe incorporating a sharpness enhancer could mitigate blurriness and enhance sharp edges.

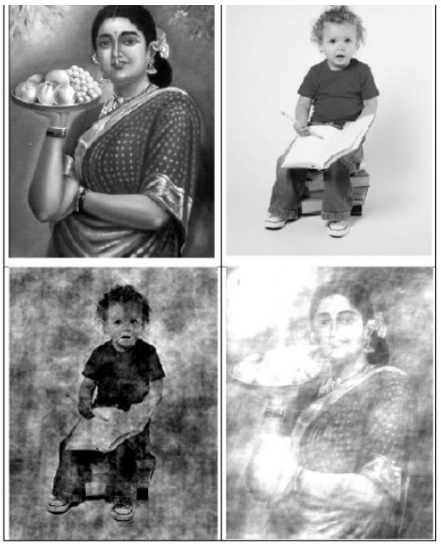
If it does not work, I will try to enhance sharpness by additional constraints within the model loss as shown below:

$$L_{\Theta'} = L_{\Theta} + \|edge(I_{HR}) - edge(I_{SR})\|_1 \quad (16)$$

The edge can be extracted whether by reconstructing with phase only or through traditional algorithms e.g., Canny edge detection. If the aforementioned methods prove ineffective, I will attempt to integrate the 'Sobel' kernel for edge extraction, which is differentiable. The image [1] in Figure 11 somewhat explains the intuition behind using phase reconstruction as a sharpness enhancer.

Among phase and magnitude, phase determines the maximum information of an image, including the edges. Therefore, I am expecting to see improvements in sharpness after incorporating the sharpness feature enhancer.

[1] https://math.stackexchange.com/questions/849382/image-reconstructionphase-vs-magnitude



a	b
c	d

(a) test image1 (b) test image2 (c) Image Reconstruction from Magnitude of Image1 and Phase of Image2 (d) Image Reconstruction from Magnitude of Image2 and Phase of Image1

Figure 11. Both the image and the caption are taken from the website mentioned in the footnote.

5.2. Implementation plan and setup

In the process of reimplementation, I plan to utilize the DIV2K dataset (Agustsson & Timofte, 2017; Timofte et al., 2017) and PyTorch framework for developing the network. First, the DIV2K training data set (consisting of 800 images) and the test set (consisting of 100 images) are used to create the baseline of the project. The setup is implemented on an 8 GB RTX 4060 GPU. In the initial phase of my implementation, I will preprocess the dataset and execute a basic super-resolution using a simple CNN model along with MSE loss between the reconstructed image and the ground truth image. This step will help me to familiarize myself with the dataset and the super-resolution process. After that, I will try to implement a super-resolution network only with the Residual Channel Attention Block (RCAB) to fully understand the necessity of residual channel attention blocks for super-resolution. Afterward, I will undertake the task of implementing the proposed TSN, omitting the second stage, which is responsible for transforming the high-resolution image into a low-resolution version with high frequency (the second component of the loss function). The authors of the paper also mentioned it in their "Ablation Study" and will check whether my results are consistent with the paper. From that, I will try to implement the original TSN network with consistent parameters. However, parameter study will not be a priority for this project rather I will focus on implementing the network architecture. The evaluation metric considered for this project is PSNR (Peak Signal-to-Noise Ratio) (Horé & Ziou, 2010).

To search for concerns raised in Section 5.1, I will include a difference image between I_{SR} and I_{HR} to fully visualize the missing pixels in the reconstruction process. Following that, I will conduct multiple experiments, varying the number of RCABs (1, 15, 20) to comprehensively assess the impact of attention blocks in the context of super-resolution. Additionally, I will incorporate the l_2 loss instead of the l_1 loss to provide justification for using the first one.

To validate my hypothesis of the edge enhancer block, I will attempt to incorporate a sharpness enhancer block and assess whether it enhances overall performance.

5.3. Implementation details

The original network's code is not accessible to the public, necessitating the creation of the code from scratch. However, I did take motivation from some of the previous works. The channel attention block is first introduced by Zhang et. al.,(Zhang et al., 2018b). In their work, they stacked residual channel attention blocks to increase the super-resolution performance. I took the channel attention layer code directly from them.

The data loader was written by me but took some inspiration from the book, "Deep Learning with Pytorch" (Stevens et al., 2020). I initially trained the network in a dual-loop setting like the other papers did. But later I found out that the whole dataset is small enough to fit within the GPU along with the model. So, I came up with a different approach. At first, I loaded the GPU with all the possible batches of data at once. During training, a random number is called in each epoch that takes one random batch of images to calculate the gradient and backpropagate. It saved more time than expected. Initially, in a dual-loop setting, it took 15 hours to train for 500 epochs. Later, in a single-loop setting, I was able to train the model for 4000 epochs and it took 2 hours only.

At first, I attempted to utilize the Mean Squared Error (MSE) loss, but subsequently transitioned to the l_1 loss, which aligns more coherently with both theoretical principles and experimental observations. In a super-resolution task, the goal is to match as much as pixels possible. The l_1 loss tends to sparsify the difference, implying that the gap between the low-resolution image and the predicted high-resolution image should approach zero. That's why it is better to use l_1 loss instead of l_2 in super-resolution tasks. The PSNR was calculated in a different way. Traditionally, the PSNR is expected to be based on the calculation of the MSE error from the RGB images. However, in Python, the values are significantly lower than Matlab values. Nonetheless, it is a common practice to convert the RGB images to YCbCr

[2] and take the Y plane only to measure the PSNR. In my implementation, I have also followed the same convention to match the PSNR values with the paper.

5.4. Results and interpretation

Initial experiments focused more on setting up a fundamental Convolutional Neural Network (CNN) architecture for super-resolution which includes the construction of the workflow and fitting the data into the network. The first experiment is summarized below:

- a simple super-resolution network was built with convolutional neural networks only.
- For this experiment, a simple MSE loss was used. For further experiments, MSE loss was used unless specified.
- The parameters (batch size, epochs) do not exactly follow the original paper, as the parameter study is not the priority.

A basic CNN should theoretically be capable of reconstructing high-resolution images due to its inherent averaging property in convolution, where neighboring pixels are summed together. However, there should be a significant blurring. We can observe this blurring in Figure 12.

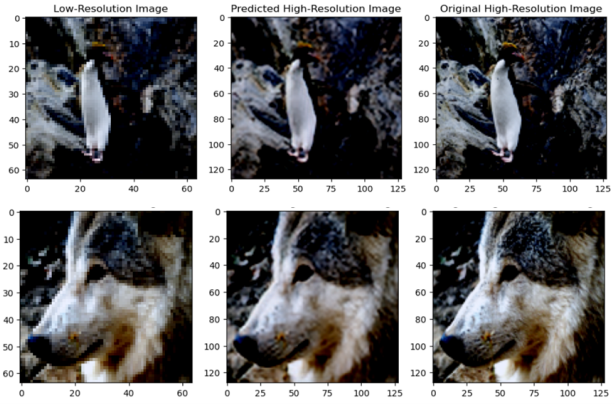


Figure 12. The input image is scaled by 0.5. All the images are normalized between values 0 to 1. This simple CNN model was trained for 100 epochs.

To fully understand the necessity of residual channel attention blocks a network is trained only using Residual Channel Attention Blocks with some basic convolutions. In this case, the MSE loss is used and the training persisted for 100 epochs. The predicted images are shown in Figure 13.

[2] https://github.com/krasserm/super-resolution/issues/19

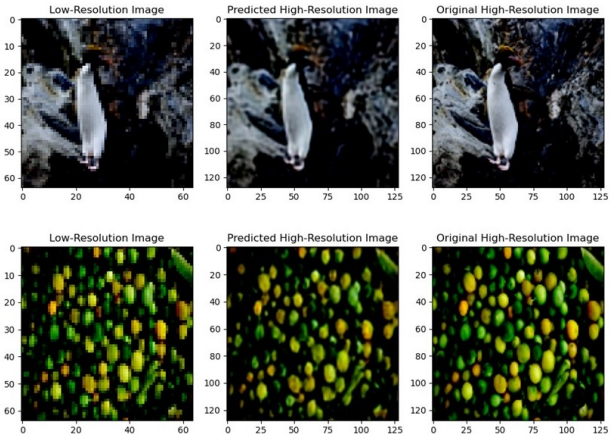


Figure 13. The input image is scaled by 0.5. All the images are normalized between values 0 to 1. In this model, a single RCAB block is utilized along with some convolution operations before and after the RCAB block.

As we can see, the images are much sharper compared to using CNNs alone. It is evident that the residual channel attention block plays a crucial role in enhancing image sharpness and removing blurriness.

The paper (Han et al., 2021) mentioned that they used 15 Residual Channel Attention Blocks (RCABs) per Residual Channel Attention Group (RCAG). We conducted an experiment where one RCAG (consisting of 15 RCABs) was utilized. The predicted images are shown in Figure 14.

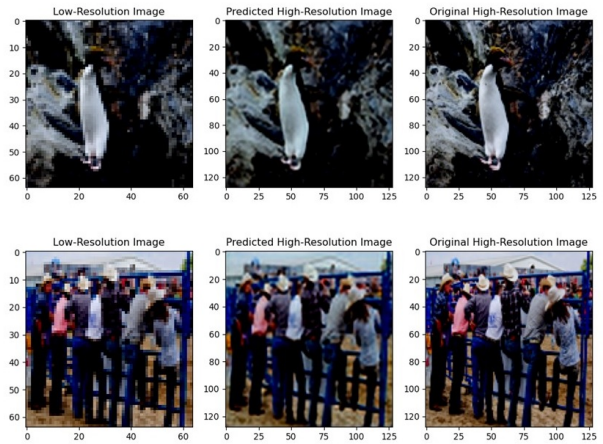


Figure 14. The input image is scaled by 0.5. All the images are normalized between values 0 to 1. In this model, a single RCAG block is utilized along with some convolution operations before and after the RCAG block.

Finally, I implemented the original network without the second stage (the second term in the loss function in Equation (6)). In this experiment, 8 RCAG blocks were used where each RCAG consists of 15 RCAB blocks. The train-

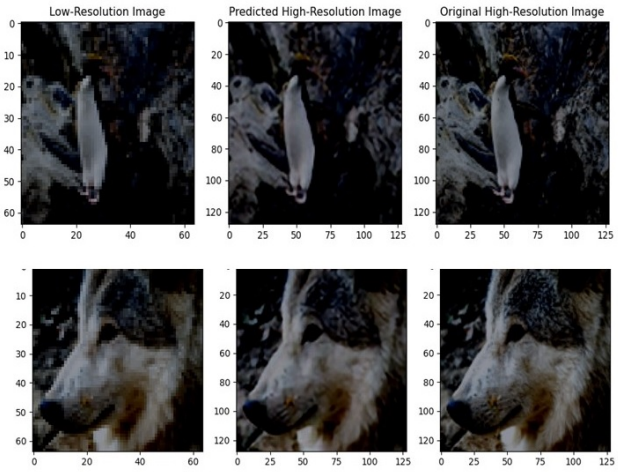


Figure 15. The input image is scaled by 0.5. All the images are normalized between values 0 to 1. In this model, eight consecutive RCAGs were used to reconstruct the image.

ing was stopped at 135 epochs because it required significantly more time compared to previous experiments. This time I used the l_1 loss, keeping consistency with the original paper. The output images as well as the difference images are shown in Figure 15, and Figure 16.

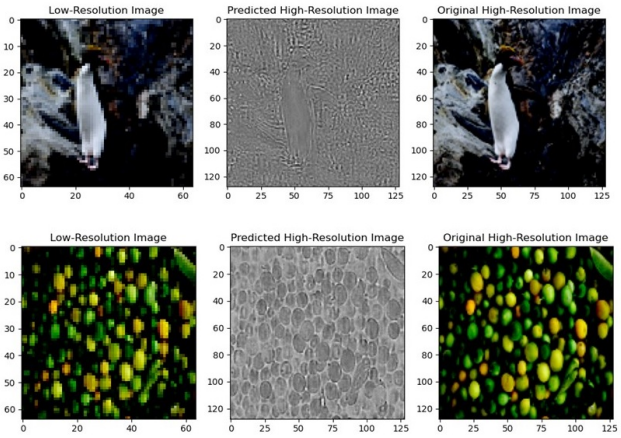


Figure 16. To generate the difference image, both the ground truth and the predicted image were transformed into grayscale. In this process, subtraction was applied between the two images to calculate the difference.

A difference image can visualize the performance of good reconstruction. Therefore, from Figure 16, it is evident that the reconstruction preserved most of the information, although it did miss certain pixel values.

It is expected that incorporating more RCAB blocks can lead to good reconstruction performance. From Table 1, it is evident that using 15 RCAB instead of one improves the performance. However, as we can see after implementing

Table 1. PSNR values on the test set for different methods.

DATA SET	MODEL	AVERAGE PSNR
DIV2K	1 RCAB	26.74 DB
DIV2K	1 RCAG (EACH INCLUDES 15 RCAB)	27.23 DB
DIV2K	TSN (8 RCAG)	26.07 DB

the TSN network the performance drops. This is due to the fact that the training was stopped at epoch 135. Moreover, the PSNR values slightly differ from the values mentioned in the paper. This is a common issue faced by the Python community. I used Python to calculate PSNR. Python "imresize" function is different from the Matlab "imresize" and this leads to some errors while calculating the PSNR. Therefore, I utilized the YCbCr planes to calculate PSNR (NOD, 2007) in the next experiments as mentioned in the "Implementation details" section.

In the first checkpoint, the first stage of the two-stage network was implemented. In particular, I implemented a comprehensive deep network, but the second-stage network, which served as an additional loss component, was not implemented. In the second checkpoint, I implemented the second network, through which the high-resolution input goes through multiple residual channel attention groups and is then compared with a corresponding layer from the initial network. The resulting loss is combined with the original reconstruction loss. If I recall from the previous checkpoint, I implemented the second portion of the loss shown in Equation 17.

$$L_\Theta = \frac{1}{N}\sum_{n=1}^{N} \lambda \|I_{HR} - I_{SR}\|_1 + \|I_{HF} - I_{LR'}\|_1 \quad (17)$$

Furthermore, I initially intended to integrate a phase loss to emphasize edge enhancement within the network. Unfortunately, this plan had to be abandoned because the Fourier Transform method required significantly more computa-

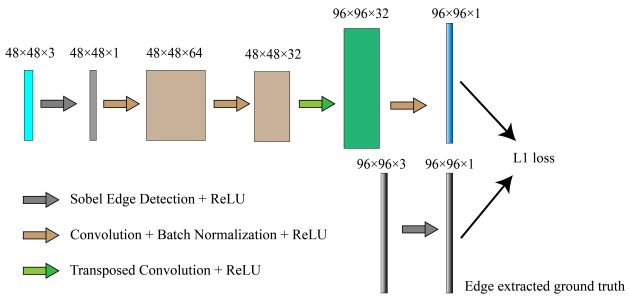


Figure 17. In the third stage network, a CNN initialized with a "Sobel" kernel is integrated to compute the edge loss.

Table 2. PSNR values on the test set for different methods.

DATA SET	MODEL	AVERAGE PSNR
DIV2K	1 RCAB	26.74 DB
DIV2K	1 RCAG (EACH INCLUDES 15 RCAB)	27.23 DB
DIV2K	TSN (8 RCAG)	26.07 DB
DIV2K	TSN	30.39 DB
DIV2K	TSN + SOBEL LOSS	30.87 DB

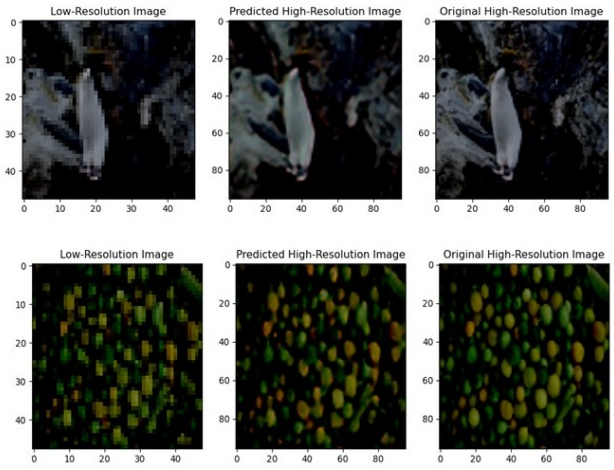


Figure 18. Predicted high-resolution images for the two-stage network (TSN).

tional power than my current setup could handle. Consequently, I opted for an alternative approach. I incorporated a shallow convolution neural network using a "Sobel" kernel at the network's input and included it as another loss term. The results of this approach, as well as the previous one (two-stage network), are shown in Table 2. The third stage that I added is shown in Figure 17. For the TSN method, the predicted high resolution images as well as the difference images are shown in Figures 18, and 19.

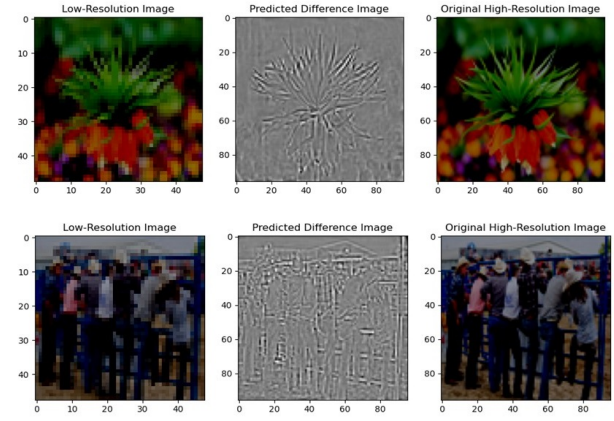


Figure 19. Difference images for the two-stage network (TSN).

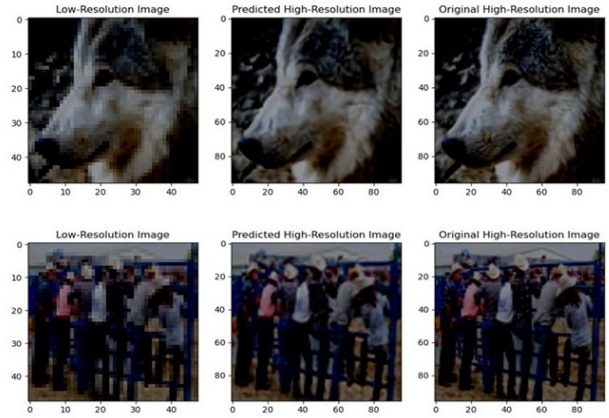


Figure 20. Predicted high-resolution images for the network where Sobel loss is incorporated as an additional loss with the TSN network.

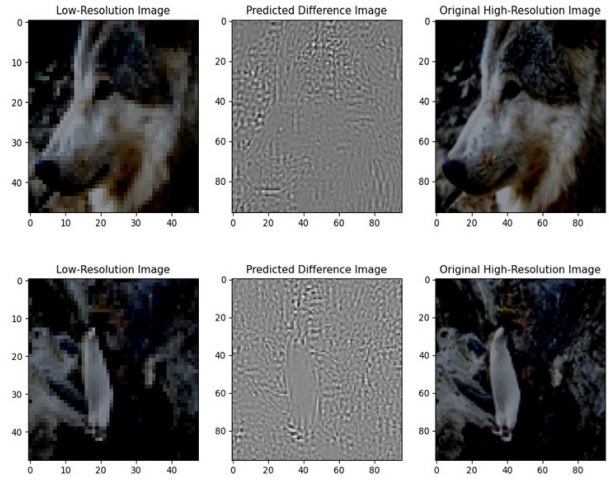


Figure 21. Difference images for the network where Sobel loss is incorporated as an additional loss with the TSN network.

As we can see from the difference images in Figure 19, the edges are not fully resolved. However, after incorporating Sobel loss this edge differences are slightly imrpoved as shown in Figures 20, and 21. Nevertheless, the discrepancy is not substantial, but I attribute this largely to employing a relatively shallow network for enhancing edges. For future studies, I will try to pass the edge extracted image (by "Sobel" kernel) through residual channel attention blocks so that the network can pull more low-level details from the edges.

6. Conclusion and Discussion

The proposed approach integrates residual channel attention blocks to preserve high-frequency information, consequently enhancing the performance of the super-resolution

network. The dual-stage loss functions facilitated the collaborative optimization of the network, guiding it to preserve high-frequency details. While implementing the network, I discovered various details that were not explicitly outlined in the original paper.

Initially, the original paper does not clearly identify the specific block responsible for preserving high-frequency information. However, based on my implementation, it becomes apparent that the residual channel attention block is the layer responsible for retaining high-frequency information. In each channel attention block, the input is passed through several layers and the final output is multiplied with the input. In my understanding, the elementwise multiplication boosts the pixels, thereby preserving high-frequency information. Therefore, in theory, increasing the attention blocks should lead to improved performance. From the data presented in Table 2, it is observed that integrating several RCABs within a single RCAG resulted in an approximately 1 dB increase in PSNR.

Secondly, the paper lacks clarity on the reasoning behind the number of RCAGs. Therefore, I experimented with 4, 8, and 15 RCABs within a single RCAG. It is confirmed that augmenting the number of RCABs marginally boosts the PSNR. Unfortunately, I couldn't explore further due to hardware limitations.

Thirdly, the idea behind incorporating an edge-enhanced layer was to maintain the edges of the low-resolution image. As evident in the difference image shown in Figure 21, the edges are preserved more effectively compared to previous methods. However, it is noteworthy that the edge-enhance layer was relatively shallow. While deepening the layer might potentially enhance performance, I did not explore this aspect in the current project, reserving it for future investigation.

During the implementation of the paper, I discovered a method that enabled me to train my network for 1000 epochs in just 20 minutes. I loaded all my data onto the GPU in a single loop. Subsequently, for each epoch, I randomly accessed the stored data, computed gradients, and updated them. This approach significantly saved time, considering that in the traditional double-loop format, data loading consumes a considerable amount of time.

I am presently engaged in an MRI project, wherein the task involves reconstructing a high-resolution image from an aliased image. In this context, I can seamlessly integrate the method discussed. The input needs to be adjusted within the MRI reconstruction pipeline for effective implementation.

References

Colorization in ycbcr color space and its application to jpeg images. *Pattern Recognition*, 40 (12):3714–3720, 2007. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2007.04.005. URL https://www.sciencedirect.com/science/article/pii/S0031320307001793.

Agustsson, E. and Timofte, R. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

Bevilacqua, M., Roumy, A., Guillemot, C., and Alberi Morel, M.-L. Neighbor embedding based single-image super-resolution using semi-nonnegative matrix factorization. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1289–1292, 2012. doi: 10.1109/ICASSP.2012.6288125.

Chakrabarti, A., Rajagopalan, A. N., and Chellappa, R. Super-resolution of face images using kernel pca-based prior. *IEEE Transactions on Multimedia*, 9(4):888–892, 2007. doi: 10.1109/TMM.2007.893346.

Chang, H., Yeung, D.-Y., and Xiong, Y. Super-resolution through neighbor embedding. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pp. I–I, 2004. doi: 10.1109/CVPR.2004.1315043.

Chen, C., Xiong, Z., Tian, X., Zha, Z.-J., and Wu, F. Camera lens super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, June 2019. doi: 10.1109/cvpr.2019.00175. URL https://doi.org/10.1109/cvpr.2019.00175.

Dai, T., Cai, J., Zhang, Y., Xia, S.-T., and Zhang, L. Second-order attention network for single image super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11057–11066, 2019. doi: 10.1109/CVPR.2019.01132.

Dong, C., Loy, C. C., He, K., and Tang, X. Image super-resolution using deep convolutional networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016. doi: 10.1109/TPAMI.2015.2439281.

Dong, W., Zhang, L., Shi, G., and Wu, X. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on Image Processing*, 20(7):1838–1857, 2011. doi: 10.1109/TIP.2011.2108306.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer,

M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An image is worth 16x16 words: Transformers for image recognition at scale, 2020. URL https://arxiv.org/abs/2010.11929.

Freedman, G. and Fattal, R. Image and video upscaling from local self-examples. *ACM Transactions on Graphics*, 30(2):1–11, April 2011. doi: 10.1145/1944846.1944852. URL https://doi.org/10.1145/1944846.1944852.

Glasner, D., Bagon, S., and Irani, M. Super-resolution from a single image. In *2009 IEEE 12th International Conference on Computer Vision*, pp. 349–356, 2009. doi: 10.1109/ICCV.2009.5459271.

Han, Y., Du, X., and Yang, Z. Two-stage network for single image super-resolution. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 880–887, 2021. doi: 10.1109/CVPRW53098.2021.00098.

Horé, A. and Ziou, D. Image quality metrics: Psnr vs. ssim. In *2010 20th International Conference on Pattern Recognition*, pp. 2366–2369, 2010. doi: 10.1109/ICPR.2010.579.

Huang, J.-B., Singh, A., and Ahuja, N. Single image super-resolution from transformed self-exemplars. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5197–5206, 2015. doi: 10.1109/CVPR.2015.7299156.

Hui, Z., Gao, X., Yang, Y., and Wang, X. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia*. ACM, October 2019. doi: 10.1145/3343031.3351084. URL https://doi.org/10.1145/3343031.3351084.

Johnson, J., Alahi, A., and Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision – ECCV 2016*, pp. 694–711. Springer International Publishing, 2016. doi: 10.1007/978-3-319-46475-6_43. URL https://doi.org/10.1007/978-3-319-46475-6_43.

Kim, J., Lee, J. K., and Lee, K. M. Deeply-recursive convolutional network for image super-resolution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1637–1645, 2016. doi: 10.1109/CVPR.2016.181.

Lai, W.-S., Huang, J.-B., Ahuja, N., and Yang, M.-H. Deep laplacian pyramid networks for fast and accurate super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5835–5843, 2017. doi: 10.1109/CVPR.2017.618.

Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. Photo-realistic single image super-resolution using a generative adversarial network. pp. 105–114, 07 2017. doi: 10.1109/CVPR.2017.19.

Li, J., Yuan, Y., Mei, K., and Fang, F. Lightweight and accurate recursive fractal network for image super-resolution. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 3814–3823, 2019. doi: 10.1109/ICCVW.2019.00474.

Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1132–1140, 2017. doi: 10.1109/CVPRW.2017.151.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9992–10002, 2021. doi: 10.1109/ICCV48922.2021.00986.

Lu, Z., Li, J., Liu, H., Huang, C., Zhang, L., and Zeng, T. Transformer for single image super-resolution. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 456–465, Los Alamitos, CA, USA, jun 2022. IEEE Computer Society. doi: 10.1109/CVPRW56347.2022.00061. URL https://doi.ieeecomputersociety.org/10.1109/CVPRW56347.2022.00061.

Stevens, E., Antiga, L., and Viehmann, T. *Deep learning with PyTorch*. Manning Publications, 2020.

Tai, Y., Yang, J., and Liu, X. Image super-resolution via deep recursive residual network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2790–2798, 2017. doi: 10.1109/CVPR.2017.298.

Timofte, R., Agustsson, E., Van Gool, L., Yang, M.-H., Zhang, L., Lim, B., et al. Ntire 2017 challenge on single image super-resolution: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

Yang, G., Yu, S., Dong, H., Slabaugh, G., Dragotti, P. L., Ye, X., Liu, F., Arridge, S., Keegan, J., Guo, Y., and Firmin, D. Dagan: Deep de-aliasing generative adversarial networks for fast compressed sensing mri reconstruction. *IEEE Transactions on Medical Imaging*, 37(6):1310–1321, 2018. doi: 10.1109/TMI.2017.2785879.

Yang, J., Wright, J., Huang, T. S., and Ma, Y. Image super-resolution via sparse representation. *IEEE Transactions on Image Processing*, 19(11):2861–2873, 2010. doi: 10.1109/TIP.2010.2050625.

Yang, W., Zhang, X., Tian, Y., Wang, W., Xue, J.-H., and Liao, Q. Deep learning for single image super-resolution: A brief review. *IEEE Transactions on Multimedia*, 21(12): 3106–3121, 2019. doi: 10.1109/TMM.2019.2919431.

Zhang, K., Gao, X., Tao, D., and Li, X. Single image super-resolution with non-local means and steering kernel regression. *IEEE Transactions on Image Processing*, 21 (11):4544–4556, 2012. doi: 10.1109/TIP.2012.2208977.

Zhang, K., Zuo, W., and Zhang, L. Learning a single convolutional super-resolution network for multiple degradations. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3262–3271, 2018a. doi: 10.1109/CVPR.2018.00344.

Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., and Fu, Y. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018b.

Zhang, Y., Tian, Y., Kong, Y., Zhong, B., and Fu, Y. Residual dense network for image super-resolution. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2472–2481, 2018c. doi: 10.1109/CVPR.2018.00262.

Zhang, Y., Li, K., Li, K., Zhong, B., and Fu, Y. Residual non-local attention networks for image restoration, 2019. URL https://arxiv.org/abs/1903.10082.

A. Appendix

The paper used ChatGPT as a paraphrasing tool. It is important to note that all content was originally written by me and later applied chatGPT to paraphrase them, ensuring that no sentences were directly copied from the original paper or any other source.