



# **Investigation into Perceptual-Aware Optimization for Single-Image Super-Resolution in Embedded Systems**

Khanh Hung Vu<sup>1,2,†</sup>, Duc Phuc Nguyen<sup>1,2,†</sup>, Duc Dung Nguyen<sup>1,2,\*</sup> and Hoang-Anh Pham<sup>1,2,\*</sup>

- <sup>1</sup> Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology (HCMUT), 268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City 72506, Vietnam
- <sup>2</sup> Vietnam National University Ho Chi Minh City (VNU-HCM), Thu Duc, Ho Chi Minh City 71308, Vietnam
- \* Correspondence: nddung@hcmut.edu.vn (D.D.N.); anhpham@hcmut.edu.vn (H.-A.P.)
- + These authors contributed equally to this work.

Abstract: Deep learning has been introduced to single-image super-resolution (SISR) in the last decade. These techniques have taken over the benchmarks of SISR tasks. Nevertheless, most architectural designs necessitate substantial computational resources, leading to a prolonged inference time on embedded systems or rendering them infeasible for deployment. This paper presents a comprehensive survey of plausible solutions and optimization methods to address this problem. Then, we propose a pipeline that aggregates the latter in order to enhance the inference time without significantly compromising the perceptual quality. We investigate the effectiveness of the proposed method on a lightweight Generative Adversarial Network (GAN)-based perceptual-oriented model as a case study. The experimental results show that our proposed method leads to significant improvement in the inference time on both Desktop and Jetson Xavier NX, especially for higher resolution input sizes on the latter, thereby making it deployable in practice.

**Keywords:** single-image super-resolution (SISR); deep learning; quantization; network pruning; knowledge distillation



Citation: Vu, K.H.; Nguyen, D.P.; Nguyen, D.D.; Pham, H.-A. Investigation into Perceptual-Aware Optimization for Single-Image Super-Resolution in Embedded Systems. *Electronics* **2023**, *12*, 2544. https://doi.org/10.3390/ electronics12112544

Academic Editor: Sunggu Lee

Received: 17 May 2023 Revised: 1 June 2023 Accepted: 2 June 2023 Published: 5 June 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/).

# 1. Introduction

Super-resolution is a low-level vision task that estimates a high-resolution (HR) image from an input image with less information, called a low-resolution (LR) image. If we have an HR image, it is easy to create an LR image by using degradation operations such as noise and blur, but the reverse problem is challenging because we have too little information in LR images to make a rich-information image. Moreover, many natural images can downsample precisely to a given LR input, which leads to an inability to find an explicit interpolation function that converts an LR image to an HR one. Granted, super-resolution (SR) is a tough problem, but it is a crucial demand, since it addresses some applications which need this problem to be solved, including medical imaging [1–3], satellite imaging [4,5], and multimedia streaming [6,7].

- In medical imaging, intrinsic and extrinsic factors cause an image's resolution loss. The intrinsic limitations mainly originate from the physical limitations of imaging systems such as X-rays, MRIs, CTs, and ultrasounds. An imaging system's spatial resolution is limited by factors such as the size of the detector or sensor, the wavelength of the imaging radiation, and the optics used to focus the radiation onto the detector. Extrinsic resolution limitations result from various factors during the image acquisition process, such as motion artifacts, patient movement, and image noise. SR technique can overcome extrinsic drawbacks;
- In the satellite field, the image captured can be affected by some natural conditions, such as haze, fog, and cloud cover. This impacts the quality and resolution of satellite images. These conditions can cause distortion, blurring, or loss of contrast in the

images, affecting interpretation and analysis accuracy. Moreover, the system itself considerably contributes to the quality of the image. The size and design of the imaging sensor, the satellite's altitude, and the imaging sensor's angle are some factors that limit the ability to detect and identify small objects or features. SR helps recover important information, which can be used for image classification or image recognition of an area or geographical location;

 In the field of information transmission, because HR images require more data to represent, transmitting or storing HR images requires more bandwidth and storage space than lower-resolution ones. This can lead to network congestion and high latency, which negatively affect the user experience with phenomena such as video lag or prolonged loading. SR can address this issue. Firstly, the image is degraded in quality before being sent to the gateway. Secondly, the LR image is processed into a high-quality image before being sent to the end-user device.

Meanwhile, deep learning-based approaches have significantly improved image reconstruction quality in recent years. However, most methods stack deeper layers or introduce complex architecture, which result in enormous execution time and computational requirements even on desktops, let alone low-end devices such as mobiles or embedded systems. The main contributions of our work in this paper can be summarized as follows:

- Conduct a comprehensive survey of existing solutions for SISR problems and feasible
  optimization methods that can be applied to enhance the performance of SISR models;
- Revisit the optimization methods to align with our previous work F2SRGAN, which
  is a lightweight GAN-based perceptual-oriented SISR model [8], as a case study. The
  objective is to demonstrate that these optimization methods remain effective for SISR
  problems while considering perceptual quality as a key aspect;
- Propose a pipeline that incorporates the optimization techniques, followed by in-depth experiments to validate the effectiveness of the proposed optimization pipeline.

The rest of the paper is constructed as follows. Section 2 comprehensively presents a survey of existing approaches for SISR. Meanwhile, Section 3 presents three optimization techniques for efficient model deployment. Then, we propose a method to reduce the inference time of our previous work as a case study in Section 4. The implementation and experiments to demonstrate effectiveness of the proposed method are presented in Section 5. Finally, Section 6 provides the final concluding remarks and suggested future work.

#### 2. Existing Approaches for Single-Image Super-Resolution (SISR)

In this section, we present a comprehensive survey of existing approaches for SISR. We characterize these methods into eight categories, including Convolutional Neural Network (CNN)-based, Distillation, Attention-based, Feedback network-based, Recursive learning-based, GAN-based, Transformer-based, and Frequency-domain based methods.

# 2.1. CNN-Based Methods

The most primitive and pioneering method using CNN is the SRCNN, proposed by Dong et al. [9], which has proven to be superior to traditional non-deep learning methods in terms of reconstructed image quality. This study also shows that the normal sparse-codingbased image recovery model can be viewed as a depth model. However, the three-layer network is unsuitable for recovering compressed images, especially when dealing with blocking artifacts and smooth regions. When different artifacts are concatenated together, the features extracted by the first layer are noisy, which causes unexpected noise patterns during the reconstruction.

Because of the real-time advantage of the SRCNN model, and to overcome the computational disadvantage, Ahn et al. [10] proposed a model that can be implemented directly on the FPGA, named Optimal-FSRCNN. This model has used the Transforming DeConvolutional (TDC) layer within the Convolutional Layer method to convert the deconvolution layer to the equivalent convolution layer, to overcome the inherent overlapping sum problem, which causes increased latency, consumes a lot of power and other hardware resources, and maximizes real-time super-resolution image parallelization using lightweight deep learning.

Inheriting the improvement of the above disadvantages, another model was formed to contribute to their improvement, which is the LAPSRN model proposed by Lai et al. [11]. Deep supervision with the Charbonnier loss function improves performance through better handling of outliers. Therefore, the model has a great ability to learn complex mappings and is effective in minimizing undesirable visualizations. Furthermore, learning upsampling filters not only minimize the generation of reconstruction artifacts produced by bicubic interpolation but also contribute to minimizing computational complexity. Experimental results show that this model is capable of solving the story of time. However, the model size is still relatively large. To reduce the number of parameters, one can replace the deep convolution layers at each level with recursive layers. In terms of image quality, not only LAPSRN, but also most other parametric SR methods fail to recover fine structure.

#### 2.2. Distillation Methods

Although the CNN-based models achieve outstanding performance, the proposed networks still have disadvantages. To achieve better performance, one needs to design a deeper network. However, as a result, these methods are computationally expensive and consume large amounts of hardware resources, which are rarely applied in mobile and embedded applications.

To solve that problem, some researchers have proposed some models to meet the needs. First, Hui et al. [12] proposed the IDN model, which extracts more helpful information with fewer convolutional layers. Although IDN has reduced parameters compared to the previous method, this reduction is achieved at the cost of significant performance sacrifice. Then, they proposed an alternative model, IDMN [13], based on their previous work IDN, with a more lightweight structure and faster running. The IMDN model uses an information multi-distillation block (IMDB) to further improve performance in both PSNR and inference time; this model took first place in the AIM 2019 constrained image super-resolution challenge [14]. However, the number of parameters in the IMDN model is greater than most lightweight SR models, such as VDSR [15] and IDN. However, the IMDN model still has room for improvement to be more lightweight.

The main component of both IDN and IMDN is an *Information Distillation Mechanism* (IDM) that explicitly divides previously extracted features into two parts: retained and refined. On the other hand, IDM is not efficient enough and brings inflexibility to network design. It is not easy to combine identity connections with IDM. However, models using this approach can be geared towards real-time systems because such models are often very flexible in making the trade-off between PSNR and inference time, via a parameter called *Channel Modulation Coefficient*.

#### 2.3. Attention-Based Methods

The authors in [16] introduced SENet, which uses channel attention to exploit interdependencies between channels of a model, improving feature map efficiency. This CNN-based squeeze-and-excitation network enhances classification networks and is now widely used in neural network design for down-streaming computer vision tasks [17].

Channel attention mechanisms have been introduced to improve the performance of neural networks in the image super-resolution domain. Zhang et al. [18] developed a CNN model called RCAN that utilizes channel attention to address SISR problems. RCAN combines residual in residual (RIR) and channel attention (CA), in which RIR is used to propagate low-frequency information [19] from the input to the output, allowing the network to learn residual information at a coarse level. The deep architecture of RCAN, with over 400 layers, enables the network to learn deeply and achieve high performance.

Super-resolution algorithms aim to restore mid-level and high-level frequencies because the low-level frequencies can be obtained from the input LR image without the need for highly complex computations. The RCAN model models the features equally or on a limited scale, ignoring the abundantly rich frequency representation at other scales. As such, these lack discriminative learning capability and capacity across channels, limiting convolutional neural network capabilities. To overcome this limitation, Anwar et al. [20] proposed the DRLN network, which uses dense connections between RBs to utilize previously computed features. The model also uses the Laplacian pyramid attention to weigh the features at multiple scales and according to their importance. The DRLN network has fewer convolutional layers than the RCAN model but more parameters. Nevertheless, it is computationally efficient due to the multiplexing of the channels, contrarily to RCAN, which uses a more expensive operation that involves more channels.

Channel attention-based approaches in image super-resolution have limitations in preserving texture and natural details due to the processing of feature maps at different layers, which can result in the loss of details in the reconstructed image. To address this issue, Niu et al. [21] proposed the HAN network, which can discover correlations between hierarchical layers, channels within each layer, and the positions of each channel, thereby activating the representative power of CNN. The HAN model also proposes a LAM model to demonstrate the relationship between features at hierarchical levels to promote CNN's performance. Additionally, the CSAM module improves the discriminative learning of the network. However, LAM only assigns a single importance weight to all features in the same class and does not consider the difference in the spatial positions of these features. While discovering relationships between features at different layers can benefit representational learning in neural networks, it is computationally expensive due to the quadratic complexity of dot-product attention, increasing the complexity from  $(HW)^2L$  to  $(HWL)^2$ , where *H* and *W* are the sizes of the feature mapping and *L* is the number of layers.

#### 2.4. Feedback Network-Based Methods

The feedback mechanism differs from conventional input-to-target mapping, incorporating a self-correcting phase during the model's learning process. In computer vision, feedback mechanisms have become increasingly popular in recent years. In particular, feedback mechanisms are commonly used in SR models because they can transfer in-depth information to the front end of the network to help process shallow information more effectively. This aids in the reconstruction of HR images from LR images.

Haris et al. [22] proposed a method called DBPN to capture the interdependencies between LR and HR image pairs. The method uses iterative back projection to calculate reconstruction error and extract high-frequency features, which are then merged to improve the accuracy of the HR image. DBPN alternates between the upsampling and downsampling layers and improves the performance through dense connections, especially for magnification, which improves by a factor of eight. However, this method is computationally expensive and increases network complexity and inference time.

Zhen et al. [23] developed the SRFBN network, which employs the negative feedback mechanism of human vision to enhance low-level representations with high-level information. The intermediate states in a constrained RNN are used to implement this feedback mode. The feedback blocks are designed to handle the feedback wiring system and generate high-level information more efficiently. The SRFBN network improves the reconstruction performance using few parameters to reduce the likelihood of overfitting due to the feedback mechanism. Still, this approach leads to an increase in computational costs. However, networks such as DBPN and SRFBN cannot learn feature mapping at multiple context scales.

The feedback mechanism used in SRFBN only propagates the highest-level feature to a shallow layer, leaving out other high-level information captured in different receptive fields. As a result, SRFBN does not make full use of high-level features or adequately refine low-level features. To address these drawbacks, Li et al. [24] introduced the GMFN network, which transfers refined features to the shallow layers. This model assumes that enough contextual information can refine the basic layers. The feature maps extracted at different layers contain complementary information for image reconstruction and are captured in different receptive fields. Then, the feedback connection optimizes the basic information with the help of the advanced counterpart.

Liu et al. [25] developed the HBPN network, taking inspiration from the DBPN model. It employs residual hourglass modules in a hierarchical structure to improve error estimation and achieve superior results. However, the ability of these models to generalize is restricted by the kernel set *k* and scaling factors *s*. Furthermore, the HBPN model requires both RGB and YUV images as input, resulting in considerable computational overhead.

The authors in [26] also introduced an ABPN network that follows the concept of the HBPN model and utilizes RBPB blocks to expand the receptive field of back-projection blocks. By leveraging the original information in the LR input, RBPB can enhance the SR performance by exploiting the interdependencies between the LR input and the SR output. However, ABPN has some limitations. Firstly, it fails to merge high-frequency features. Secondly, the standard convolutional layers and self-attention modules do not distinguish between different degrees of feedback errors, resulting in back-projection blocks being unable to focus on areas with significant errors and reducing the correction effect.

# 2.5. Recursive Learning-Based Methods

The feedback-based model utilizes self-correcting parameters, distinguishing it from the recursive learning-based model, where the parameters are shared among the modules.

The CARN network proposed by Ahn et al. [27] uses a lightweight model that replaces the standard RB block with an efficient RB version, which has fewer parameters and computational costs than RB but similar learning capabilities. The CARN network achieved super-resolution benchmark results among lightweight models with a parameter count of less than 1.5 million. Still, the performance is limited by the number of parameters, and the PSNR and SSIM metrics are reduced.

To reduce computational complexity and cost, Choi et al. [28] proposed the BSRN network, which includes an initial feature extractor, a recursive RB, and an upscaling part. The SRRFN model, on the other hand, was proposed by Li et al. [29] and achieved superior results with fewer parameters and less execution time than the RCAN model. SRRFN introduced a new fractal module (FM), which can create multiple topological structures based on a simple component to detect rich image features and increase the fault tolerance of the model.

The LP–KPN network proposed by Cai et al. [30] is based on the Laplacian pyramid to learn kernels per pixel for the decomposed image pyramid, to achieve high computational efficiency with large kernel sizes. The LP–KPN model outperforms the CRAN models trained on simulation data while having fewer convolutional layers. However, the LP– KPN model only reconstructs the LR image by collaborating with different pixel-local reconstructions, which does not fully use hierarchical features across different frequencies.

#### 2.6. GAN-Based Methods

GANs [31] uses a game theory approach that includes the generator and the discriminator trying to fool each other. The generator generates SR images that the discriminator cannot distinguish as real or artificial HR images. In this way, HR images with better perceptual quality are produced. The corresponding PSNR values are often attenuated, highlighting the problem that the quantitative measures common in the SR literature do not encapsulate the perceptual accuracy of the generated HR outputs.

GAN models overcome the weakness when using the loss function MSE as the criterion. Although minimizing MSE also maximizes PSNR and is a common metric used to evaluate and compare SR algorithms, the ability of MSE (and PSNR) to capture relevant differences in terms of perceptions, such as high texture details, is very limited because they are determined based on differences in the image pixels. The higher PSNR does not necessarily reflect a better perceptual outcome. Realizing the above, Ledig et al. [32] proposed the generative model related to GAN in the super-resolved problem as SRGAN. The SRGAN model uses a deep residual network (ResNet) with skip connections and diverges from the MSE as the sole optimization objective. The model also identifies a new perceptual loss function using VGG network high-level feature mappings combined with a discriminator that encourages solutions that are difficult to distinguish from HR images. Although SRGAN significantly improves the overall image quality of the reconstruction, its disadvantage is that the model is difficult to train, often producing artifacts in SR images.

To avoid the generation of SR images with artifacts, Xintao Wang et al. [33] proposed the ESRGAN network to make the reconstructed image more realistic. Firstly, this model removes the BN layer, reducing computational costs and memory. Moreover, it also contributes to reducing the artifacts in the SR image. Secondly, the use of pre-activation features results in a more accurate brightness distribution (i.e., closer to the actual brightness), producing sharper edges and richer textures. Thirdly, the deep learning model shows outstanding performance in easy training thanks to the RRDB block without the BN layer. Even so, the model has difficulty recreating the high-frequency edges. Furthermore, the regeneration effect greatly deteriorates when the ESRGAN model is applied to multiple degradations. One form of the SRGAN model with a slight variation that gives the most satisfactory results in terms of inference time and applies to low-end devices, such as embedded systems, is the SwiftSRGAN model, proposed by Koushik Sivarama Krishnan et al. [34], so that the model can run on a time-constrained system; this model changes the convolution block to a DSC block.

The generator structure of the SRGAN model uses a deep residual network called SRResNet. This type of architecture gives good results in terms of structural similarity and detail. However, in experiments, it was found to perform poorly in maintaining global information and high-level structure, sometimes distorting the overall characteristics of the image. Therefore, Mirchandani et al. [35] proposed the DPRSGAN model. To avoid the above problem, the model uses dilated convolution to capture global structures and fewer parameters. On the other hand, changing the discriminator to a Markovian discriminator (PatchGAN) speeds up model training and produces sharper details.

Additionally, in order to deal with the degraded images in the real world in general, Real-ESRGAN was formed, as proposed by Wang et al. [36]. Compared to ESRGAN models, Real-ESRGAN is trained entirely on synthetic data, which helps it to recover complex realworld images with better image performance. Another model used for the real-world image super-resolution problem is BSRGAN, proposed by Zhang et al. [37]. It has been proposed as a real-world degradation model to remove the disadvantages of synthetic data generation and build a robust model for different combinations of downsampling kernels, blur kernels, and noise. The method demonstrates outstanding results when dealing with real-world datasets where the degradation model is unknown. Nevertheless, since the models are trained using pairs of images generated by such real-world degradation models and considered for general scenarios, it can be confirmed that the denoising has surpassed the required level. Hence, the real-world degradation model is not appropriate for specific visual inspection tasks that require fixed HR and LR noise levels for noise reduction.

In addition, F2SRGAN [8] enhances the receptive field of the convolution operator by employing Fast Fourier Convolution. This technique enables the model to capture lowfrequency characteristics in the frequency domain, resulting in quicker coverage of highfrequency features compared to the traditional spatial domain of standard convolution.

#### 2.7. Transformer-Based Methods

Methods employing CNN or GAN are limited to using local image information, ignoring the global interaction between image components, resulting in low-quality recovery. Transformer is a novel deep learning model that employs a self-attention mechanism based on assigning varying weights to the significance of each portion of the input data. It has been a cutting-edge method in natural language processing (NLP) since its inception. Transformer has become increasingly prominent in computer vision tasks such as object detection, segmentation, classification, and image super-resolution due to its ability to solve long-term dependency issues. It can utilize local and global information from the input image to produce a more detailed output image. This innovation has attracted many researchers and introduced a new network architecture for image super-resolution.

A typical model for applying Transformer to this super-resolution problem is the ESRT proposed by Zhisheng et al. [38]. The model consists of an LCB block, which uses HPB blocks to automatically adjust the feature map size to extract intensive feature mappings with low computational cost. The model also has an LTB block to capture long-term dependencies between similar patches in an image with the help of specially designed Efficient Transformer (ET) and Efficient Multi-Head Attention (EMHA) mechanisms. This model is proposed to effectively enhance the feature representation and long-term dependence of similar patches in an image, to achieve better performance with low computational cost. Although Transformer is a powerful model, there is still the problem that Transformer-based models are heavy models, i.e., the number of parameters and the amount of data to train are still large.

# 2.8. Frequency-Domain Based Methods

Super-resolution (SR) algorithms that use a frequency domain-based method transform low-resolution (LR) input images into the frequency domain to estimate an HR image. The reconstructed high-resolution (HR) image is then transformed back into the spatial domain. Fourier and wavelet transform-based methods are two algorithms that depend on the transformation used to convert images to the frequency domain. In [39], the authors converted LR satellite images to the frequency domain by using the discrete Fourier transform (DFT). Then, the relationship between the aliased DFT coefficients of the LR images and those of the unknown HR image was combined. In addition to enhancing the high-frequency information of images, frequency domain-based SR techniques have low computational complexity. However, these methods have some drawbacks, including being insufficient to handle real-world applications and having difficulty expressing prior knowledge used to regularize the SR problem. Many frequency domain approaches rely on Fourier transform properties such as the shifting and sampling theorems, making them easy to understand and apply. Some frequency domain methods make assumptions that enable the use of efficient procedures for computing restoration, such as the Fast Fourier Transform (FFT).

Implicit neural functions, parameterized by multilayer perceptrons (MLP), have shown great success in representing continuous domain signals such as images, shapes, and signals. However, one drawback of using a standalone MLP is that it tends to focus on low-frequency components [40] and may not capture fine details [41]. To address this limitation, Lee et al. [42] introduced a dominant frequency estimator called the LTE tool, which improves the input features of the MLP. The LTE model includes three additional trainable layers that process the encoder output and correspond to sine and cosine waves' amplitude, frequency, and phase. This output is then used as input for an MLP that has four fully connected layers. To further enhance the results, a global skip connection with a bilinear upscaled version of the input is added to the entire model, allowing the deep model to focus on the residual between the closed-form approximation and the final result. While LTE has achieved high-quality arbitrary-scale rectangular super-resolution with high-frequency details, its spatially varying nature prevents it from evaluating a frequency response for image warping.

Zhang et al. [43] developed a model called SwinFIR that combines the SwinIR model proposed by Liang et al. [44] with the FFC block [45]. This model uses a frequency domain approach to capture global information better and restore high-frequency details in images. The SwinFIR model performs well in restoring images with periodic transformations and challenging samples. However, one limitation of this model is that it is slow for large-scale images, as measuring importance at a global spatial scale requires vector multiplications along rows and columns. Recently, Nguyen et al. [8] proposed an enhanced

model F2SRGAN that further improves the FFC block by performing the convolution operator directly in the frequency domain, rather than splitting the real and imaginary parts and implementing them separately.

# 2.9. Summary

SISR is a challenging problem, since a single low-resolution (LR) image can correspond to multiple high-resolution (HR) images. Researchers have devised various solutions to tackle this ill-posed problem. We summarized existing works according to eight categories, as shown in Table 1. Initially, CNN-based methods were introduced, laying the groundwork for subsequent advancements. Feedback and recursive mechanisms leverage self-correction techniques, while attention mechanisms, particularly transformer-based methods, emphasize important information while disregarding irrelevant details. The latter became the tendency in recent years. These methods can give better reconstruction results, but they have the trade off of longer execution time due to complex model architectures.

Categories	Related Works
	[9]
CNN-based methods	[10]
	[11]
Distillation methods	[12]
	[13]
	[16]
Attention-based methods	[18]
	[20]
	[21]
	[22]
	[23]
Feedback network-based methods	[24]
	[25]
	[26]
	[27]
Recursive learning-based methods	[28]
8	[29]
	[30]
	[8]
	[32]
GAN-based methods	[33]
	[34]
	[33]
	[57]
	[38]
Transformer-based methods	[43]
	[44]
	[8]
	[39]
Frequency domain-based methods	[42]
	[43]
	[44]

Table 1. Existing approaches for single-image super-resolution (SISR).

To address the issue of increased execution time, several approaches have been proposed. Distillation methods reduce the number of channels through successive distillation, thus making it possible to produce a lightweight model. Similarly, GAN-based methods generate perceptually pleasing images that appear more natural than earlier approaches while requiring fewer parameters. Additionally, frequency domain-based methods employ features learned in the frequency domain to capture low-frequency information faster, resulting in lightweight models with fewer layers.

# 3. Optimization Techniques for Deploying the Model to the Embedded System

Recent benchmark deep learning models require substantial computational resources. Therefore, optimization becomes crucial for efficient real-world deployment, particularly on low-end devices. In this section, we present a survey of three optimization techniques, including quantization, network pruning, and knowledge distillation, which can be utilized for SISR.

# 3.1. Quantization

Quantization reduces the bit-width of computations and tensor storage compared to floating-point precision. This leads to more compact model representation, faster operations, and lower computational costs during runtime while maintaining reasonable accuracy.

Quantization involves two main operators: Quantize (Q) and Dequantize (DQ). For  $x \in [\beta, \alpha]$  as the range of real values and *b* as the bit-width of the lower precision format,  $\beta$  and  $\alpha$  are the smallest and largest values in the range of real floating-point values. A *b*-bit signed integer precision format can represent  $2^b$  possible integer values, with a value range of  $[-2^{b-1}, 2^{b-1} - 1]$ . In this super-resolution problem, we consider the real value range as a single-precision floating-point value range, which is a floating-point number with 32-bit precision. The quantization operation maps a real value  $x \in [\beta, \alpha]$  to a value within the value range of *b* bits with low precision  $[-2^{b-1}, 2^{b-1} - 1]$ .

The quantization operator includes two processes: real-value transformation and clipping process. The process of transforming a real value x into a quantized value  $x_q$  can be defined as shown in Equation (1):

λ

$$c_q = \left\lfloor \frac{x}{s} - z \right\rfloor,\tag{1}$$

where *z* is an immutable parameter (zero point) of the same type as the quantized value. It represents the quantization value  $x_q$  corresponding to the real value of zero (x = 0). *s* is a scaling factor that divides the real value range into a number of parts. In asymmetric quantization ( $-\alpha \neq \beta$ ), we can calculate the scaling factor *s* and zero point as shown in Equation (2):

$$\begin{cases} s = \frac{\alpha - \beta}{2^b - 1} \\ z = \left\lfloor \frac{\beta(2^b - 1)}{\alpha - \beta} + 2^{b - 1} \right\rfloor$$

$$(2)$$

However, if the output  $x_q$  falls outside the precision range of b bits, meaning it falls outside the interval  $[-2^{b-1}, 2^{b-1} - 1]$ , then we clip the range so that it does not fall outside that range. This means that if  $x_q < -2^{b-1}$ , we adjust  $x_q = -2^{b-1}$ , and if  $x_q > 2^{b-1} - 1$ , we adjust  $x_q = 2^{b-1} - 1$ . Mathematically, we can define the quantized value Q(x, b), with x as the real value and b as the low-precision bit width, using Equation (3):

$$Q(x,b) = \min\left(\max\left(x_{q}, -2^{b-1}\right), 2^{b-1} - 1\right)$$
(3)

The approximate quantization of a real value by a quantized operation is defined by Equation (4):

$$D(q) = s(q+z) \tag{4}$$

In general, the coefficients *s* and *z* depend heavily on the real interval  $[\beta, \alpha]$ . This is called *clipping range learning*. There are two approaches to determine this clipping range:

determining the range without re-training the model and determining the range by retraining the model. Two typical methods that reflect these approaches are Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT), respectively.

The approach that determines the clipping range without re-training model (such as PTQ) forms the clipping range by feeding representative samples into the model. Then, the model is quantized based on the results obtained after having some representative samples. This method is fast but produces lower output accuracy compared to learning the clipping range through re-training model.

The approach of clipping range through re-training model (such as QAT) involves more steps and takes longer to tune the model, but provides better output accuracy than the PTQ approach. Regarding the QAT method, the quantization nodes (Q Nodes) and de-quantization nodes (DQ Nodes) are inserted into the network according to a specific set of rules [46]. Then, the network is trained again in several batches in a process called fine-tuning. The Q/DQ nodes simulate quantization loss and add it to the training loss during fine-tuning, making the network more flexible in terms of quantization. However, a problem arises during the back-propagation stage of model tuning, where there exists a rounding function that is non-differentiable at some points and equal to 0 at almost all points.

$$\frac{d}{dx}\lfloor x \rfloor = \begin{cases} \text{Undefined if } x \in \mathbb{N} \\ 0 \text{ if } x \notin \mathbb{N} \end{cases}$$
(5)

One way to overcome this problem is to use Straight-Through Estimator (STE) [47] to estimate gradients for thresholding in neural networks. The concept of STE is to set the input gradients to a thresholding function equal to its output gradients, ignoring the derivative of the thresholding function. Specifically, STE allows us to approximate the derivative of the floor function as 1.

$$\frac{d}{dx}\lfloor x \rfloor \approx 1 \tag{6}$$

This means that we are allowed to "skip" the backpropagation calculation when passing through Q and DQ blocks during network training. As demonstrated in [48], a QAT process consists of the following four stages:

- **Stage 1**: Train the network with floating-point operations;
- **Stage 2**: Insert fake quantization layers *Q*<sub>fake</sub> into the trained network. The *Q*<sub>fake</sub> layer is used to simulate the process of integer quantization using floating-point computations. It is usually performed by a quantization step (Q) followed by a dequantization step (DQ);
- **Stage 3**: Perform fine-tuning of the model. Note that in this process, the gradient is still used in floating-point;
- Stage 4: Perform execution by removing Q<sub>fake</sub> and loading the quantized weights.

# 3.2. Network Pruning

Network pruning is an important technique for both memory size and bandwidth reduction. This allows neural networks to be deployed in constrained environments such as embedded systems.

To effectively prune a pre-trained model, two aspects need exploration: pruned architecture and candidate selection. Pruned architecture can be divided into two types: human-defined and algorithmic. Human-defined pruning involves determining a fixed ratio of pruned channels in each layer, while automatic pruning determines the target architecture through algorithms based on global comparisons of structure importance across layers. Unstructured pruning is a form of automatic pruning, where the positions of pruned weights are determined during training and the positions of zeros cannot be predetermined.

For automatic network pruning, the most commonly used approach is to remove redundant weights that provide little information to the pre-trained model. The brute-force method was the earliest approach used, which involves setting each weight to 0 and checking the loss function's change. However, due to the large search space, this method is not efficient. Therefore, other methods have been developed, broadly categorized into two types: magnitude-based pruning and penalty-based pruning. Both approaches generate values close to 0 for weights, effectively overcoming the drawback of the brute-force method.

Magnitude-based pruning method prunes weights based on the idea that those trained with larger values are more important. The most basic methods are to prune weights with a value of 0 or all weights within a given threshold. One method based on the Hessian matrix is LeCun et al.'s Optimal Brain Damage (OBD) [49], which uses a second-order Taylor expansion approximation to minimize the difference between the loss value of the pruned model weights and the loss value of the model weights before pruning. This method achieves high accuracy but requires significant computation, particularly for the inverse matrix in the optimization solution.

The penalty-based pruning method aims to reduce the overfitting of the model using regularization. This involves adding an extra term to the loss function to evaluate the complexity of the model. This new loss function is called the regularization loss function, which is usually defined as shown in Equation (7):

$$\mathcal{L}_{\text{reg}}(\mathbf{w}) = \mathcal{L}(\mathbf{w}) + \lambda \mathcal{R}(\mathbf{w})$$
(7)

where  $\mathcal{L}(\mathbf{w})$  is the original loss function of the model, and  $\mathcal{R}(\mathbf{w})$  is the regularization term depending only on the weights of the model. The constant  $\lambda$  is usually a small positive number, also known as the regularization parameter. The regularization parameter is often chosen to be small to ensure that the solution of the optimization problem for  $\mathcal{L}_{reg}(\mathbf{w})$  is not far from the solution of the optimization problem for  $\mathcal{L}(\mathbf{w})$ . The regularization term often used in the regularization technique is the  $L_p$  regularization function, which is defined by Equation (8):

$$\left\|\mathbf{w}\right\|_{p} = \left(\sum_{i=1}^{n} \left|\mathbf{w}_{i}\right|^{p}\right)^{\frac{1}{p}}$$

$$(8)$$

where *n* is the number of elements in the vector **w**. For p = 1, this is called LASSO pruning, and for p = 2, this is called weight decay pruning. The use of LASSO pruning has been shown to be more effective in weight selection than using weight decay pruning, because the use of the  $L_2$  regularization function (also known as the ridge loss function) only generates weights that approach 0 rather than being equal to 0, while using the  $L_1$  regularization function generates better 0 weights. However, using the  $L_1$  regularization function raises a trade-off problem between the sparsity and performance of the model [50].

The mentioned pruning algorithms only prune at the weight-level, but higher-level pruning requires pruning methods based on structures such as groups or networks. The regularization technique can be extended to grouped regularization, expressed as the grouped regularization loss function:

$$\mathcal{L}_{\text{reg}}(\mathbf{W}) = \mathcal{L}(\mathbf{W}) + \lambda \sum_{k=1}^{K} \mathcal{R}(\mathbf{W}^k)$$
(9)

where  $\mathcal{L}(\mathbf{W})$  is the original loss function of the model, **W** is the set of weights that can be trained for all *K* layers in the model, and  $\mathcal{R}(\mathbf{W}^k)$  is the regularization operator in layer *k* to prune the set of weights,  $\{\mathbf{W}^k\}$ . The parameter  $\lambda$  is the regularization parameter, which is used to balance the loss function and the pruning criterion. Yuan and Lin [51] proposed the grouped LASSO regularization. To reduce subsets of weights such as filters or channels, the subsets need to be considered as groups in the regularization criterion.

The grouped LASSO restricts subsets of unnecessary parameters to simultaneously equal 0. The regularization term of the grouped LASSO is defined by Equation (10):

$$\mathcal{R}_{GL}(\mathbf{W}^k) = \sum_{g \in G} \left\| \mathbf{W}_g^k \right\|_2 \tag{10}$$

where  $g \in G$  is a group in the set of groups G,  $W_g^k$  is the weight matrix or weight vector of group g (i.e., a submatrix or subvector of  $W^k$ ). However, estimating the sum of regularization terms across different layers may not be cumulative and, therefore, may be meaningless due to differences in distribution and intensity. Realizing this, Gongfan Fang et al. [52] proposed a regularization term in which the importance level of regularization terms is normalized for each layer by adding a constant for each different layer, to ensure that removing groups becomes safer. Specifically, the regularization term that Gongfan Fang et al. [52] defined is represented in Equation (11):

$$\mathcal{R}_{DG}(\mathbf{W}^k) = \sum_{g \in G} \gamma_k \left\| \mathbf{W}_g^k \right\|_2 \tag{11}$$

where  $\mathcal{R}_{DG}(\mathbf{W}^k)$  represents the regularization term of layer *k* and  $\gamma_k$  is the regularization parameter for layer *k*.

# 3.3. Knowledge Distillation

Knowledge distillation (KD) refers to the process of transferring knowledge from one large complex model, or a set thereof, to a smaller, deployable model under real-world constraints. It was first demonstrated by Buciluă et al. [53] to compress a model and transfer information to train a smaller model without sacrificing accuracy. A knowledge distillation system consists of three main components: knowledge, distillation schemes, and distillation algorithms.

In a neural network, *knowledge* typically refers to the learned weights and biases. There are three main types of knowledge that can be distilled from a teacher model to a student model. The first is response-based knowledge, where the student model mimics the output of the teacher model. This is the most common type of knowledge system. The loss of information in such cases is related to the computation of the divergence between the logits of the teacher and student models. Kullback–Leibler divergence (KL divergence) is commonly used in feedback-based knowledge distillation methods [54,55]. In feature-based knowledge systems, the teacher model learns to recognize features in its intermediate layers, which can be used to train the student model. The loss function for knowledge distillation achieves this by minimizing the difference between the feature activations of the teacher and student models. Finally, in relation-based knowledge systems, the relationships between feature mappings are used to train the student model. These relationships can be modeled as correlations between feature maps, graphs, similarity matrices, feature embeddings, or probability distributions based on feature representations.

In terms of *distillation schemes*, there are three major techniques that are commonly used: Offline Distillation, Online Distillation, and Self-Distillation. Offline Distillation is the most popular technique, in which a pre-trained teacher model guides a student model. The pre-trained teacher model is a large deep neural network in the Offline Distillation process. In certain circumstances, the pre-trained model may not be accessible for Offline Distillation. In an end-to-end training procedure, Online Distillation can be used to circumvent this limitation by simultaneously updating the teacher and student models. Using parallel computing, Online Distillation is a highly effective process. Knowledge Distillation has two issues. The *first issue* is that the selection of the teacher model significantly impacts the accuracy of the student model. The *second issue* is that student models cannot consistently achieve the same high level of accuracy as teacher models [56], which may result in an unacceptable loss of precision during deployment. Self-Distillation addresses these concerns by utilizing the same network for both teacher and student models. Self-Distillation

begins by affixing shallow attention-based classifiers after the intermediate layers of the network at various depths. Then, during training, the deeper classifiers are regarded as teacher models that guide the training of the student models using a divergence-based loss function on the outputs and an  $L_2$  loss function [57] on the feature maps. In the deployment stage, all the additional shallow classifiers are removed.

Meanwhile, in terms of *distillation algorithm*, there are currently nine commonly used algorithms. The *first type* is adversarial learning, which enhances existing training sets to improve model performance or allow teacher-student models to learn better data distributions [58]. The second type is multi-teacher distillation, which uses different teacher structures that can provide different types of knowledge. When distilled into a student model, it can produce better predictions than individual models. The *third type* is cross-modal distillation, which transfers knowledge between different modalities. This occurs when there are insufficient data or labels for specific modalities during training or testing [59], so knowledge must be transferred between modalities. The *fourth type* is graphbased distillation, which captures internal data relationships using a graph. The graph is used in two ways:-as a means of transferring knowledge and to control the transfer of teacher knowledge. The *fifth type* is attention-based distillation, which transfers knowledge using attention mappings. The sixth type is a data-free distillation process, which synthesizes data from a pre-trained teacher model. The seventh type is quantization distillation, which transfers knowledge from a high-precision teacher network (e.g., 32-bit floating-point) to a low-precision student network (e.g., 8 bits). The *eighth type* is *lifelong distillation*, which is based on continuous learning mechanisms such as continual learning and meta-learning, where previously acquired knowledge is accumulated and transferred to future learning. The *ninth type* is NAS-based distillation, which is also utilized to identify appropriate student model architectures for optimizing learning from teacher models.

#### 3.4. Summary

We summarize existing works related to three optimization techniques in Table 2. Knowledge distillation utilizes the knowledge transfer technique, which guides a simpler compressed student model to perform as well as the original model. On the other hand, network pruning focuses on the significance of weights by removing or zeroing out unimportant parameters, resulting in a more compact network. Regarding the input datatype, quantization methods reduce the bit width of input for each layer. This process significantly enhances model performance as operations are executed using smaller numerical representations.

<b>Optimization Techniques</b>	<b>Related Works</b>				
Quantization	[46] [48]				
Network pruning	[49] [50] [52]				
Knowledge distillation	[53] [54] [56] [58] [59]				

 Table 2. Optimization techniques for deploying the model to the embedded system.

# 4. Case Study

In this section, we have opted to employ three optimization methodologies in order to illustrate their influence on the inference time of a specific SISR model, namely, F2SRGAN [8]. The optimization techniques employed include quantization, network pruning, and knowledge distillation; each is then revisited to fit the F2SRGAN model. The reason why we chose F2SRGAN is that, first, it is a lightweight GAN-based model which is feasible to deploy on embedded systems; second, F2SRGAN is a perceptual-oriented model which produces SR images that look more natural in comparison to the common PSNR-oriented model; and third, F2SRGAN learns features in the frequency domain to extract global information, and the Fast Fourier Transform (FFT) may pose a bottleneck to the model, so optimization is needed to address this problem.

#### 4.1. Quantization

Calculating a two-dimensional Fast Fourier Transform (2D-FFT) in F2SRGAN model when using only a 16-bit floating-point range is a drawback. Consider a 2D matrix **f** with all elements in the range (0, 1). We use the Fourier transform of the matrix **f** with size  $N \times M$  to obtain the matrix **F** with the element at the position (u, v) of the matrix **F**, defined by the Equation (12):

$$\mathbf{F}(u,v) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \mathbf{f}(x,y) e^{-j2\pi \left(\frac{ux}{N} + \frac{vy}{M}\right)}$$
(12)

Putting u = 0 and v = 0 into the Equation (12) gives the following:

$$\mathbf{F}(0,0) = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \mathbf{f}(x,y) e^{-j2\pi \left(\frac{0 \times x}{N} + \frac{0 \times y}{M}\right)} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \mathbf{f}(x,y)$$
(13)

Equation (13) proves that the value of the element at position (0,0) of the Fourier matrix **F** is equal to the sum of all elements of the input matrix **f**. If we consider the case of an 8K image, i.e., with a resolution of  $1080 \times 1080$ , and whose elements are all equal to 0.5, the value of **F**(0,0) is as follows:

$$1080 \times 1080 \times 0.5 = 583200$$

The number 583200 is not in the 16-bit floating-point range (from -65,504 to 65,504) but is in the 32-bit floating-point range (from  $-3.4 \times 10^{38}$  to  $3.4 \times 10^{38}$ ). This is enough to prove that it is impossible to calculate the FFT2D of all matrices using only a 16-bit floating-point range with image resolutions up to  $1080 \times 1080$ .

Through the analysis of the above drawback, we decided to use mixed precision. We use a 16-bit floating point to reduce the calculation time and a 32-bit floating point to make it possible to calculate FFT and avoid unexpected results due to being outside the range of the 16-bit floating point. The F2SRGAN model is quantized to ensure that the accuracy reduction is not significant, but the execution time is shortened using the QAT method. Specifically, the discriminator model is preserved here, and we perform quantization at the generator. More specifically, we insert fake quantization, including Q (quantization) and DQ (dequantization) layers, into basic convolution-related layers, including DSC and activation layers such as ReLU and PreLU.

During inference, the input to the model is a matrix of elements with a precision of 16bit floating-point instead of 32-bit floating-point. For the CFU block, we compute it entirely with 32-bit floating-point precision. Details of the 32-bit floating-point calculation of the CFU block are shown in Figure 1. More specifically, during inference, a 16-bit floating-point number input ise expanded to a 32-bit floating-point. Moreover, the weight to calculate the Complex Convolution is extended from the real and complex parts of the weight to a 32-bit floating-point so that the weight of the complex number has the precision of a 64-bit floating point, so that the CFU block can be calculated within a 32-bit floating-point without problems related to output values that do not fall within the range allowed by a 16-bit floating-point. After computing in the CFU block with 32-bit floating-point precision, the output will be converted to a value with 16-bit floating-point precision.



To show more clearly the advantages when using the QAT method, we also propose a method to use the F2SRGAN model directly with the input of a 16-bit floating-point precision without using the QAT method (W/O QAT).

**Figure 1.** Implement the calculation on the 32-bit floating-point precision of the CFU block, where fp32 is a floating-point number with 32-bit precision and complex64 is a 64-bit floating-point complex number (the real part has 32-bit floating-point precision, and the imaginary part also has 32-bit floating-point precision).

# 4.2. Network Pruning

In network pruning, we apply the newest method presented in [52], where the authors proposed the automatic pruning process via a dependency graph to represent the dependency between layers. Moreover, the proposed method does not rely on a specific problem or architecture. Therefore, even with simple pruning criteria, it gives better results compared to previous ones.

Firstly, we load the pre-trained model to optimize the weight of the particular model. This step varies depending on a specific task or problem. Secondly, we build the dependency graph based on that model. The dependency graph represents the dependency between layers via a symmetric Boolean matrix. The authors define two types of dependency: interlayer dependency and intra-layer dependency. The former is known when the structure of the network is pre-defined. We only have to check whether a connection exists from one layer's output to another's input, while the latter inspects if they have the same pruning schemes. The Batch Norm layer does share pruning schemes, whereas Convolution does not work in most cases. Let *F* be the network, with *L* layers on which we want to build the dependency graph. Dependency graph *G* has the size of  $2L \times 2L$ , as there are *L* layers overall. For layer *i*, we have  $F_i^-$  and  $F_i^+$  as the input and output, respectively. If i = j, it is an intra-layer dependency, and they share the same pruning schemes, denoted as  $\operatorname{sch}(F_i^-) = \operatorname{sch}(F_i^+)$ , while, if  $i \neq j$ , it is an inter-layer dependency. Overall, it can be formulated by Equation (14):

$$G\left(F_{i}^{-},F_{j}^{+}\right) = \left(i \neq j \land \text{CONNECT}\left(F_{i}^{-},F_{j}^{+}\right)\right) \lor \left(i = j \land \text{sch}(F_{i}^{-}) = \text{sch}(F_{j}^{+})\right), \quad (14)$$

where  $\wedge$  and  $\vee$  are logic operators AND and OR, respectively, and function CONNECT  $(F_i^-, F_j^+)$  represents whether there is a connection between  $F_i^-$  and  $F_j^+$ . After that, layers related to each other through the dependency graph are grouped together, which creates a set of grouped layers.

The next step is to prune the network based on the grouped layers and their importance score. To simplify the process, we chose to experiment on three different importance scores as follows:

- **Random**: Randomize the importance score of parameters within each group;
- **Mean Absolute Error** (L<sub>1</sub>): Li et al. [60] proposed to prune a proportion of filter that has the lowest sum of its absolute kernel weight. If we let filter *j* be  $\mathbf{W}_{j,i} \in \mathbb{R}^{k \times k}$  and *m* be the number of filters, the importance score of filter *j* is defined by Equation (15) [60]:

$$s_j = \sum_{i=1}^m \left\| \mathbf{W}_{\mathbf{j}, \mathbf{i}} \right\|_1; \tag{15}$$

• LAMP: Conceptually, LAMP measures the importance of a connection with the unpruned ones. LAMP considers network layers as operators, following the logic of lookahead pruning [61]. Based on minimizing model-level  $L_2$  distortion, Lee et al. [62] proposed a score function. First, the authors sort the weight tensor  $W_1, \ldots, W_L$  in ascending order so that, for any pair of u and v with u < v, the connected weights maintain at least the same inequality  $|W[u]| \le |W[v]|$ . The LAMP score is calculated using Equation (16) [62]:

LAMP
$$(u, W) = \frac{(W[u])^2}{\sum_{u \ge v} (W[v])^2}$$
 (16)

where *W* is weight tensor and *u* is the index of calculated weight. The score function guarantees that only one weight in each layer has a score of 1, which is the highest magnitude.

In addition, the authors in [52] proposed a different way to calculate the importance score for a whole set of groups. Finally, the model is much lighter after pruning all components with low importance scores. However, the accuracy decreases as a result. Therefore, we have to fine-tune the model at the end to address this issue.

# 4.3. Knowledge Distillation

In the F2SRGAN model, the transfer process between the frequency domain and the spatial domain may prolong the overall inference time of the model. To address this problem, we design a student network without the Fourier Residual Block, which means that it now has 8 Residual Blocks without BN, instead of the 16 in the teacher model. The overall student network architecture is represented in Figure 2.





Figure 2. F2SRGAN student architecture.

As F2SRGAN prioritizes learning low-frequency features to cover faster high-frequency ones, we want the student network to behave similarly via feature-based knowledge distillation. Specifically, let  $\mathbf{h}^{T} = \{h_{1}^{T}, \dots, h_{T}^{T}\}$  and  $\mathbf{h}^{S} = \{h_{1}^{S}, \dots, h_{S}^{S}\}$  be feature candidates from teacher and student with *S* and *T* as the number of feature candidates of student and teacher. We, respectively, include two loss components as follows:

• Activation-Based and Gradient-Based Attention Transfer: Attention Transfer (AT) distillation was proposed by Sergey Zagoruyko et al. [63], with the use of attention maps, which can be formulated by Equation (17):

$$\mathcal{L}_{\mathrm{AT}} = \frac{1}{2} \sum_{j \in \mathcal{J}} \left\| \tilde{\phi}^{\mathrm{C}}(h_{j}^{\mathrm{T}}) - \tilde{\phi}^{\mathrm{C}}(h_{j}^{\mathrm{S}}) \right\|_{p}$$
(17)

where  $\mathcal{J}$  is the indices of all student–teacher attention map pairs, and  $\tilde{\phi}^{C}$  represents the aggregated feature map in space with  $L_2$  normalization, which means that we replace each vectorized attention map  $\mathbf{Q}$  with  $\frac{\mathbf{Q}}{\|\mathbf{Q}\|_2}$ ;

Attention-Based Feature Distillation: Attention-Based Feature Distillation was proposed by Mingi Ji et al. [64] to define hint position and weights for hints. The authors evaluate the pair of features between student and teacher through *attention values* α<sub>t,s</sub>. The loss component can be presented by Equation (18):

$$\mathcal{L}_{\text{AFD}} = \frac{1}{2} \sum_{t} \sum_{s} \alpha_{t,s} \left\| \tilde{\phi}^{\text{C}}(h_t^{\text{T}}) - \tilde{\phi}^{\text{C}}(\hat{h}_s^{\text{S}}) \right\|_p$$
(18)

where p = 2 (according to the authors), and  $\tilde{\phi}^C$  represents the normalized channel-wise average pooling function with  $L_2$  normalization.  $\hat{h}_s^S$  is the upsampled or downsampled version of  $h_s^S$  to match the feature map size of the teacher model.

In addition, the authors in [63,64] used the Kullback–Leibler loss in their studies. Let a probability vector soften, with a hyper-parameter  $\tau$  for network f in  $\mathbf{p}^{f}(\tau)$ . The value k of the soften probability vector  $\mathbf{p}^{f}(\tau)$  is calculated as shown in Equation (19) [64]:

$$\mathbf{p}_{k}^{f}(\tau) = \frac{\exp\left(\frac{\mathbf{z}_{k}^{f}}{\tau}\right)}{\sum_{j=1}^{K} \exp\left(\frac{\mathbf{z}_{j}^{f}}{\tau}\right)}$$
(19)

where  $\mathbf{z}_k^f$  is the *k* value of vector logit  $\mathbf{z}^f$ , *K* is the number of classes, and  $\exp(x) = e^x$  is the natural exponential. The Kullback–Leibler loss is defined by Equation (20) [64]:

$$\mathcal{L}_{\mathrm{KL}}(\mathbf{p}^{s}(\tau), \mathbf{p}^{t}(\tau)) = \tau^{2} \sum_{j} \mathbf{p}_{j}^{t}(\tau) \log \frac{\mathbf{p}_{j}^{r}(\tau)}{\mathbf{p}_{j}^{s}(\tau)}.$$
(20)

The overall loss function for student is defined as Equation (21).

$$\mathcal{L}_{\text{Student}} = \mathcal{L}_{G} + \gamma \mathcal{L}_{\text{KD}} + \delta \mathcal{L}_{\text{KL}},\tag{21}$$

where  $\mathcal{L}_{\text{KD}}$  is the knowledge distillation loss, which is  $\mathcal{L}_{\text{AT}}$  or  $\mathcal{L}_{\text{AFD}}$  in this case,  $\gamma$  and  $\delta$  are hyperparameters, and  $\mathcal{L}_{G}$  is the loss function proposed for F2SRGAN Generator. However, including the  $\mathcal{L}_{\text{KL}}$  increases the overall PI and hurts the perceptual quality.

# 4.4. Lightweight Optimization Pipeline

In this section, we propose a comprehensive pipeline that integrates knowledge distillation, network pruning, and quantization techniques into a unified solution. Figure 3

provides a step-by-step representation of the pipeline, which can be applied to any deep learning model.

To begin, we define a student model based on the target model we aim to optimize. For instance, in the case of F2SRGAN, we exclude all Fourier Residual blocks and only retain the Residual block, as the Fourier operator may introduce bottlenecks in the model. Next, we re-train the student model using the knowledge distillation scheme, leveraging the pre-trained teacher model. Following knowledge distillation, we proceed with network pruning to further optimize the model. The pruning process may lead to a loss in accuracy, so we fine-tune the pruned model to mitigate this impact. Lastly, we apply quantization to the pruned model, reducing the precision of the model's parameters. After quantization, we conduct another round of fine-tuning to refine the performance.

It is important to note that the specific training techniques employed during the fine-tuning step may vary depending on different architectures and problem settings.



Figure 3. Lightweight optimization pipeline.

# 5. Experiments

#### 5.1. Datasets

We followed the methodology described in F2SRGAN [8] to train our model. For training, we utilized the HR image dataset from DIV2K [65], which consisted of 800 training images and 100 test images. Additionally, we incorporated the Flicker2K [66] dataset, which included 2650 images. During training, we adopted the following process. First, we randomly cropped patches from the HR images, with each patch having a size of 48. These patches were then downscaled to create the corresponding LR training images. Furthermore, we applied horizontal flips, vertical flips, and rotations to augment the dataset. The DIV2K test images were utilized as the validation dataset, allowing us to assess the performance of the model. Finally, we evaluated the results of our model on various benchmark datasets for Image Super-Resolution, including Set5, Set14, BSD100, and Urban100.

# 5.2. Implementation Details

We employed AdamW [67] with the default settings of  $\beta_1$  value of 0.9,  $\beta_2$  value of 0.999, and weight decay value of 0. In F2SRGAN [8], the authors presented the training process as consisting of two phases, which are  $L_1$  and perceptual phase. However, the implementation details vary among our methods.

In quantization, we first loaded the pre-trained F2SRGAN model, then we fine-tuned the model by the method shown in Section 4.1 with the same training method as training the F2SRGAN model, which includes the loss function used to update weight after an iteration, numbers of iterations to train, and the optimizer policy. Finally, we removed all fake quantization layers and loaded the weight after fine-tuning for inference with the input tensor as a 16-bit floating-point precision.

In the context of network pruning, we adopted the methodology proposed by Gongfan Fang et al. [52]. Specifically, we first built a dependency graph to demonstrate the relationship between layers. Subsequently, we determined the importance score for each layer in three different manners, as described in Section 4.2, using the layer groupings from the dependency graph. To mitigate the potential accuracy degradation caused by the pruning process, we conducted a fine-tuning stage, similar to the perceptual phase, on the model, which helps to address any loss in accuracy that may arise.

Regarding the knowledge distillation method, we began by training the student network, as illustrated in Section 4.3, using the  $L_1$  phase. Then, in the perceptual phase, we incorporate da pre-trained F2SRGAN model, acting as a teacher model, to guide the training process.

The training process of Light F2SRGAN incorporates each method's implementation details in a step-by-step manner. Our implementation was based on PyTorch and ran on an Intel Xeon CPU and Tesla P100 GPU.

# 5.3. Evaluation Metrics

Image reconstruction accuracy, such as PSNR (Peak Signal-to-Noise Ratio), is the commonly used metric in super-resolution tasks. Additionally, we also evaluated the model with Perceptual Index (PI) [68], due to the limitations of PSNR. The best results were selected based on the latter, as better PI demonstrates better perceptual quality.

Regarding the inference time, we evaluated the model on both Desktop and Jetson Xavier NX. The Desktop settings included CPU Intel Xeon and GPU Tesla P100. We averaged out 50 inference times on various sizes of square images to obtain the best result.

# 5.4. Experimental Results

Table 3 summarizes the quantitative perceptual of different methods. Specifically, applying quantization for F2SRGAN further improves the PI, surpassing the F2SRGAN baseline's quantitative result, while reducing the average inference time. Regarding the network pruning method, calculating the importance score using LAMP yields the best PI among the three investigated scores. The knowledge distillation method, with the AT loss component, significantly improves the PI compared to the baseline, which excludes the teacher model and trains on a normal scheme, while the AFD loss component slightly enhances the PI metric.

**Table 3.** Quantitative perceptual (average PI/PSNR) comparison between optimization methods on benchmark datasets on the Y channel from the YCbCr space. Lines separate each method.

Method	Scale	Set5		Set14		BSDS100		Urban100	
		PI	PSNR	PI	PSNR	PI	PSNR	PI	PSNR
F2SRGAN Baseline [8]	$\times 4$	4.39	27.99	4.05	25.93	4.16	26.21	4.43	23.21
W/O Quantization Aware Training (QAT) Quantization Aware Training (QAT)	${ imes 4} { imes 4}$	4.38 3.97	28.00 27.99	4.05 3.85	25.94 25.91	4.17 3.79	26.21 26.11	4.43 4.20	23.21 23.10
Network Pruning w/Random Network Pruning w/MAE [60] Network Pruning w/LAMP [62]	${ imes 4} \\ { imes 4} \\ { imes 4}$	4.85 4.73 4.72	27.44 27.69 27.83	4.33 4.26 4.22	25.53 25.75 25.87	4.25 4.44 4.27	25.83 26.02 26.09	4.47 4.55 4.42	22.90 23.02 23.08
Knowledge Distillation (KD) Baseline Knowledge Distillation w/AFD [64] Knowledge Distillation w/AT [63]	${ imes 4} \\ { imes 4} \\ { imes 4}$	5.29 5.18 4.62	27.79 27.88 27.50	4.66 4.84 4.27	25.84 25.91 25.59	4.80 5.04 4.35	26.31 26.35 26.20	4.70 4.81 4.58	23.12 23.13 23.04
Light F2SRGAN	$\times 4$	4.61	26.63	4.44	24.77	4.34	25.42	4.44	22.41

Specifically, in the quantization method, even without employing the QAT method, the results remain the same or even better than the baseline when using 16-bit inputs without any additional pre-training step. This outcome highlights the substantial contribution of the FFT block and its further advantages when fine-tuned to enhance the PI parameter.

Regarding the knowledge distillation method, adopting AT loss shows superior results compared to AFD loss. The reason behind this is, while AFD considers all feature pairs between the teacher and student, providing a more comprehensive perspective than AT, its effectiveness is highly dependent on the similarity of features between the teacher and student. In the case of F2SRGAN, the teacher model learns features in the frequency domain through FFC blocks, whereas the student's features are learned in the spatial domain. This mismatch between the features of the student and teacher leads to a performance decrease in AFD loss. On the other hand, AT loss component provides more effective spatial guidance to the student model by aligning attention maps, thus potentially compensating for the loss of frequency-based information.

Figure 4 depicts the average inference time of optimization methods on both Desktop and Jetson Xavier NX. The left chart shows the improvement in inference time of all forms; especially, our proposed Light F2SRGAN reduces the inference time from over 800 ms to about 300 ms on 1080 squared input image. At the same time, the right chart illustrates the run time on Jetson Xavier NX. Due to the limited computation strength, we can only run models with a resolution from 144 to 480 squared input images. However, the quantization method makes it possible to deploy with 720-pixel resolution. In addition, our proposed Light F2SRGAN continues to show large advancement in model inference time on Jetson Xavier NX.



**Figure 4.** Average inference time of optimization methods ( $\times$ 4 scale) with different resolutions. The left chart demonstrates Desktop inference time, while the right chart is Jetson Xavier NX inference time.

In general, the network pruning, knowledge distillation, and Light F2SRGAN focus more on inference time, so that they may introduce artifacts and distortion in the visual results. However, the quantization method gains better enhancement in color intensity and light effect, especially in the visual result of image 167062 in the BSDS100 dataset, the second visual result in Figure 5.



Figure 5. Visual results (×4) of optimization methods on benchmark datasets.

# 5.5. Discussion

Our investigation involves experimenting with three optimization techniques on a specific model. All three techniques demonstrate promising results, showing relatively close PI and improved inference time compared to the original model. Knowledge distillation proves advantageous as it resolves time bottlenecks caused by the global learning of the FFT block, resulting in a significant improvement in inference time.

Additionally, quantization with the QAT method achieves the highest PI, surpassing the baseline of F2SRGAN. We observe that the QAT method aims to preserve the model's characteristic learning properties to a greater extent, optimizing it by reducing computation and memory through lower-bit computations. Therefore, this approach is more suitable for low-end devices. While the network pruning method also achieves competitive results, it is essential to note that this approach is more complex than the two mentioned methods. More attention should be given to pruning in deeper layers, as they may contain globally aggregated features from previous layers and some connections, such as residual connections in SISR.

To minimize inference time and leverage the benefits of each optimization method, we propose a straightforward yet impactful pipeline for compressing the model without compromising quality. However, it is worth noting that our pipeline requires considerable training time due to the fine-tuning step necessary for quality recovery. We suggest more work should focus on efficiently incorporating existing techniques. Additionally, the 1080-pixel resolution is not feasible on Jetson Xavier NX due to the limited hardware resources, even for a lightweight model with only a few hundred thousand parameters. This issue should be thoroughly solved in the future.

# 6. Conclusions

This paper presents a comprehensive survey of existing solutions for single-image super-resolution (SISR) problems and optimizations aimed at efficient model deployment on common embedded systems. Additionally, we propose an optimization pipeline specifically designed to enhance inference time performance. Experimental results obtained using the F2SRGAN model for an SISR problem demonstrate that each optimization technique and the proposed pipeline, which are applicable to various problem domains and solutions, also yield significant improvements in the SISR context. Furthermore, our proposed approach achieves substantial advancements in inference time on both Desktop and Jetson Xavier NX platforms. Notably, the quantization method enables the deployment of our solution for 720-pixel squared input images.

While our study has achieved promising results by incorporating three optimization techniques for the SISR problem, we acknowledge that there is still potential for further improvement. Specifically, exploring group-level pruning with sparse training could enhance the calculation of importance scores in network pruning. In the case of knowledge distillation, numerous techniques warrant exploration. Additionally, we recognize that our proposed pipeline entails significant time requirements, particularly during the fine-tuning stage. As part of our future work, we aim to address these issues and devise solutions accordingly.

Author Contributions: Conceptualization, K.H.V., D.P.N. and H.-A.P.; methodology, K.H.V., D.P.N. and D.D.N.; validation, K.H.V., D.P.N., D.D.N. and H.-A.P.; investigation, K.H.V. and D.P.N.; writing—original draft preparation, K.H.V. and D.P.N.; writing—review and editing, D.D.N. and H.-A.P.; supervision, D.D.N. and H.-A.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

Acknowledgments: The authors acknowledge Ho Chi Minh City University of Technology (HC-MUT), VNU-HCM, for supporting this study.

**Conflicts of Interest:** The authors declare no conflict of interest.

# References

- 1. Park, J.; Hwang, D.; Kim, K.Y.; Kang, S.K.; Kim, Y.K.; Lee, J.S. Computed tomography super-resolution using deep convolutional neural network. *Phys. Med. Biol.* 2018, 63, 145011. [CrossRef] [PubMed]
- You, C.; Li, G.; Zhang, Y.; Zhang, X.; Shan, H.; Li, M.; Ju, S.; Zhao, Z.; Zhang, Z.; Cong, W.; et al. CT Super-resolution GAN Constrained by the Identical, Residual, and Cycle Learning Ensemble (GAN-CIRCLE). *IEEE Trans. Med. Imaging* 2019, *39*, 188–203. [CrossRef] [PubMed]
- Chen, Y.; Xie, Y.; Zhou, Z.; Shi, F.; Christodoulou, A.G.; Li, D. Brain MRI super resolution using 3D deep densely connected neural networks. In Proceedings of the 15th International Symposium on Biomedical Imaging (ISBI 2018), Washington, DC, USA, 4–7 April 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 739–742. [CrossRef]
- 4. Müller, M.U.; Ekhtiari, N.; Almeida, R.M.; Rieke, C. Super-resolution of multispectral satellite images using convolutional neural networks. *arXiv* 2020, arXiv:2002.00580.
- Shermeyer, J.; Van Etten, A. The Effects of Super-Resolution on Object Detection Performance in Satellite Imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–20 June 2019; pp. 1432–1441. [CrossRef]
- 6. Kwon, I.; Li, J.; Prasad, M. Lightweight Video Super-Resolution for Compressed Video. Electronics 2023, 12, 660. [CrossRef]
- Khani, M.; Sivaraman, V.; Alizadeh, M. Efficient Video Compression via Content-Adaptive Super-Resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Virtual, 11–14 October 2021; pp. 4521–4530. [CrossRef]
- 8. Nguyen, D.P.; Vu, K.H.; Nguyen, D.D.; Pham, H.A. F2SRGAN: A Lightweight Approach Boosting Perceptual Quality in Single Image Super-Resolution via a Revised Fast Fourier Convolution. *IEEE Access* **2023**, *11*, 29062–29073. [CrossRef]
- 9. Dong, C.; Loy, C.C.; He, K.; Tang, X. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 295–307. [CrossRef]
- Ahn, S.; Kang, S.J. Deep Learning-based Real-Time Super-Resolution Architecture Design. J. Broadcast Eng. 2021, 26, 167–174. [CrossRef]
- 11. Lai, W.; Huang, J.; Ahuja, N.; Yang, M. Deep Laplacian Pyramid Networks for Fast and Accurate Super-Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017. [CrossRef]
- 12. Hui, Z.; Wang, X.; Gao, X. Fast and Accurate Single Image Super-Resolution via Information Distillation Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018. [CrossRef]
- Hui, Z.; Gao, X.; Yang, Y.; Wang, X. Lightweight Image Super-Resolution with Information Multi-distillation Network. In Proceedings of the 27th ACM International Conference on Multimedia, Nice France, 21–25 October 2019; ACM: New York, NY, USA, 2019; pp. 2024–2032. [CrossRef]
- Zhang, K.; Gu, S.; Timofte, R.; Hui, Z.; Wang, X.; Gao, X.; Xiong, D.; Liu, S.; Gang, R.; Nan, N.; et al. AIM 2019 Challenge on Constrained Super-Resolution: Methods and Results. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019. [CrossRef]
- 15. Kim, J.; Lee, J.K.; Lee, K.M. Accurate Image Super-Resolution Using Very Deep Convolutional Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016.
- 16. Hu, J.; Shen, L.; Sun, G. Squeeze-and-Excitation Networks. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141. [CrossRef]
- 17. Xie, C.; Zeng, W.; Jiang, S.; Lu, X. Bidirectionally aligned sparse representation for single image super-resolution. *Multimedia Tools Appl.* **2018**, *77*, 7883–7907. [CrossRef]
- Zhang, Y.; Li, K.; Li, K.; Wang, L.; Zhong, B.; Fu, Y. Image Super-Resolution Using Very Deep Residual Channel Attention Networks. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018. [CrossRef]
- 19. Yang, C.; Lu, G. Deeply Recursive Low- and High-Frequency Fusing Networks for Single Image Super-Resolution. *Sensors* 2020, 20, 7268. [CrossRef]
- Anwar, S.; Barnes, N. Densely Residual Laplacian Super-Resolution. *IEEE Trans. Pattern Anal. Mach. Intell.* 2019, 44, 1192–1204. [CrossRef]
- Niu, B.; Wen, W.; Ren, W.; Zhang, X.; Yang, L.; Wang, S.; Zhang, K.; Cao, X.; Shen, H. Single Image Super-Resolution via a Holistic Attention Network. In Proceedings of the 16th European Conference, Glasgow, UK, 23–28 August 2020. [CrossRef]
- 22. Haris, M.; Shakhnarovich, G.; Ukita, N. Deep Back-Projection Networks For Super-Resolution. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018. [CrossRef]
- Li, Z.; Yang, J.; Liu, Z.; Yang, X.; Jeon, G.; Wu, W. Feedback Network for Image Super-Resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–20 June 2019. [CrossRef]
- 24. Li, Q.; Li, Z.; Lu, L.; Jeon, G.; Liu, K.; Yang, X. Gated Multiple Feedback Network for Image Super-Resolution. *arXiv* 2019, arXiv:1907.04253. [CrossRef]
- Liu, Z.S.; Wang, L.W.; Li, C.T.; Siu, W.C. Hierarchical Back-Projection Network for Image Super-Resolution. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Long Beach, CA, USA, 16–20 June 2019; pp. 2041–2050. [CrossRef]

- Liu, Z.S.; Wang, L.W.; Li, C.T.; Siu, W.C.; Chan, Y.L. Image Super-Resolution via Attention based Back-Projection Networks. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019.
- Ahn, N.; Kang, B.; Sohn, K. Fast, Accurate, and, Lightweight Super-Resolution with Cascading Residual Network. In Proceedings
  of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018. [CrossRef]
- Choi, J.; Kim, J.; Cheon, M.; Lee, J. Lightweight and Efficient Image Super-Resolution with Block State-based Recursive Network. arXiv 2019. [CrossRef]
- Li, J.; Yuan, Y.; Mei, K.; Fang, F. Lightweight and Accurate Recursive Fractal Network for Image Super-Resolution. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 3814–3823. [CrossRef]
- Cai, J.; Zeng, H.; Yong, H.; Cao, Z.; Zhang, L. Toward Real-World Single Image Super-Resolution: A New Benchmark and A New Model. arXiv 2019. [CrossRef]
- Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative Adversarial Networks. *Commun. ACM* 2020, 63, 139–144. [CrossRef]
- Ledig, C.; Theis, L.; Huszar, F.; Caballero, J.; Cunningham, A.; Acosta, A.; Aitken, A.; Tejani, A.; Totz, J.; Wang, Z.; et al. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; IEEE Computer Society: Piscataway, NJ, USA, 2017; pp. 105–114. [CrossRef]
- Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; Loy, C.C. ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks. In Proceedings of the European Conference on Computer Vision (ECCV) Workshop, Munich, Germany, 8–14 September 2018; Springer International Publishing: Cham, Switzerland, 2019; pp. 63–79. [CrossRef]
- Krishnan, K.S.; Krishnan, K.S. SwiftSRGAN—Rethinking Super-Resolution for Efficient and Real-time Inference. In Proceedings of the 2021 International Conference on Intelligent Cybernetics Technology & Applications (ICICyTA), Bandung, Indonesia, 1–2 December 2021; pp. 46–51. [CrossRef]
- Mirchandani, K.; Chordiya, K. DPSRGAN: Dilation Patch Super-Resolution Generative Adversarial Networks. In Proceedings of the 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2–4 April 2021; pp. 1–7. [CrossRef]
- Wang, X.; Xie, L.; Dong, C.; Shan, Y. Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Virtual, 11–14 October 2021; pp. 1905–1914. [CrossRef]
- Zhang, K.; Liang, J.; Van Gool, L.; Timofte, R. Designing a Practical Degradation Model for Deep Blind Image Super-Resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Virtual, 11–14 October 2021. [CrossRef]
- 38. Lu, Z.; Liu, H.; Li, J.; Zhang, L. Efficient Transformer for Single Image Super-Resolution. arXiv 2021, arXiv.2108.11084. [CrossRef]
- 39. Tsai, R. Multiframe Image Restoration and Registration. Adv. Comput. Vis. Image Process. 1984, 1, 317–339.
- Rahaman, N.; Baratin, A.; Arpit, D.; Draxler, F.; Lin, M.; Hamprecht, F.; Bengio, Y.; Courville, A. On the Spectral Bias of Neural Networks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 5301–5310.
- Tancik, M.; Srinivasan, P.P.; Mildenhall, B.; Fridovich-Keil, S.; Raghavan, N.; Singhal, U.; Ramamoorthi, R.; Barron, J.T.; Ng, R. Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains. *Adv. Neural Inf. Process. Syst.* 2020, 33, 7537–7547.
- 42. Lee, J.; Jin, K.H. Local Texture Estimator for Implicit Representation Function. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–20 June 2022; pp. 1919–1928. [CrossRef]
- 43. Zhang, D.; Huang, F.; Liu, S.; Wang, X.; Jin, Z. SwinFIR: Revisiting the SwinIR with Fast Fourier Convolution and Improved Training for Image Super-Resolution. *arXiv* 2022, arXiv.2208.11247. [CrossRef]
- Liang, J.; Cao, J.; Sun, G.; Zhang, K.; Van Gool, L.; Timofte, R. SwinIR: Image Restoration Using Swin Transformer. In Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW), Virtual, 11–14 October 2021; pp. 1833–1844. [CrossRef]
- Chi, L.; Jiang, B.; Mu, Y. Fast Fourier Convolution. In *Advances in Neural Information Processing Systems*; Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., Eds.; Curran Associates, Inc.: Red Hook, NY, USA , 2020; Volume 33, pp. 4479–4488.
- Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 2704–2713. [CrossRef]
- 47. Bengio, Y.; Léonard, N.; Courville, A.C. Estimating or Propagating Gradients Through Stochastic Neurons for Conditional Computation. *arXiv* **2013**, arXiv.1308.3432. [CrossRef]
- 48. Siddegowda, S.; Fournarakis, M.; Nagel, M.; Blankevoort, T.; Patel, C.; Khobare, A. Neural Network Quantization with AI Model Efficiency Toolkit (AIMET). *arXiv* 2022, arXiv.2201.08442. [CrossRef]
- LeCun, Y.; Denker, J.; Solla, S. Optimal Brain Damage. In Proceedings of the Advances in Neural Information Processing Systems, Denver, Colorado, USA, 27–30 November 1989; Volume 2; pp. 598–605.

- 50. Zhang, Y.; Wang, H.; Qin, C.; Fu, Y. Learning Efficient Image Super-Resolution Networks via Structure-Regularized Pruning. In Proceedings of the Tenth International Conference on Learning Representations (ICLR), Virtual, 25–29 April 2022.
- Yuan, M.; Lin, Y. Model Selection and Estimation in Regression with Grouped Variables. J. R. Stat. Soc. Ser. B (Stat. Methodol. 2006, 68, 49–67. [CrossRef]
- 52. Fang, G.; Ma, X.; Song, M.; Mi, M.B.; Wang, X. Depgraph: Towards Any Structural Pruning. arXiv 2023, arXiv:2301.12900.
- 53. Buciluundefined, C.; Caruana, R.; Niculescu-Mizil, A. Model Compression. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, 20–23 August 2006; ACM: New York, NY, USA, 2006; pp. 535–541. [CrossRef]
- Kim, T.; Oh, J.; Kim, N.Y.; Cho, S.; Yun, S.Y. Comparing Kullback-Leibler Divergence and Mean Squared Error Loss in Knowledge Distillation. In Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, (IJCAI), Virtual, 19–26 August 2021; pp. 2628–2635. [CrossRef]
- 55. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. arXiv 2015, arXiv:1503.02531.
- Mirzadeh, S.; Farajtabar, M.; Li, A.; Ghasemzadeh, H. Improved Knowledge Distillation via Teacher Assistant: Bridging the Gap Between Student and Teacher. In Proceedings of the The Thirty-Fourth AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
- 57. Pham, M.; Cho, M.; Joshi, A.; Hegde, C. Revisiting Self-Distillation. arXiv 2022, arXiv.2206.08491. [CrossRef]
- Wang, X.; Zhang, R.; Sun, Y.; Qi, J. KDGAN: Knowledge Distillation with Generative Adversarial Networks. In Advances in Neural Information Processing Systems; Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., Eds; Curran Associates, Inc.: Red Hook, NY, USA; 2018; Volume 31.
- Gupta, S.; Hoffman, J.; Malik, J. Cross Modal Distillation for Supervision Transfer. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 26 June–1 July 2016; IEEE Computer Society: Piscataway, NJ, USA, 2016; pp. 2827–2836. [CrossRef]
- Li, H.; Kadav, A.; Durdanovic, I.; Samet, H.; Graf, H.P. Pruning Filters for Efficient ConvNets. arXiv 2016, arXiv.1608.08710. [CrossRef]
- Park, S.; Lee, J.; Mo, S.; Shin, J. Lookahead: A Far-Sighted Alternative of Magnitude-based Pruning. *arXiv* 2020, arXiv.2002.04809.
   [CrossRef]
- 62. Lee, J.; Park, S.; Mo, S.; Ahn, S.; Shin, J. A Deeper Look at the Layerwise Sparsity of Magnitude-based Pruning. *arXiv* 2020, arXiv.2010.07611. [CrossRef]
- 63. Zagoruyko, S.; Komodakis, N. Paying More Attention to Attention: Improving the Performance of Convolutional Neural Networks via Attention Transfer. *arXiv* **2016**, arXiv.1612.03928. [CrossRef]
- Ji, M.; Heo, B.; Park, S. Show, attend and distill: Knowledge distillation via attention-based feature matching. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtual, 2–9 February 2021 ; Volume 35, pp. 7945–7952.
- Agustsson, E.; Timofte, R. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1122–1131. [CrossRef]
- Lim, B.; Son, S.; Kim, H.; Nah, S.; Lee, K.M. Enhanced Deep Residual Networks for Single Image Super-Resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 1132–1140. [CrossRef]
- 67. Loshchilov, I.; Hutter, F. Fixing Weight Decay Regularization in Adam. arXiv 2017, arXiv.1711.05101. [CrossRef]
- Blau, Y.; Mechrez, R.; Timofte, R.; Michaeli, T.; Zelnik-Manor, L. The 2018 PIRM Challenge on Perceptual Image Super-Resolution. In *European Conference on Computer Vision (ECCV) Workshops*; Leal-Taixé, L., Roth, S., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 334–355. [CrossRef]

**Disclaimer/Publisher's Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.