



Article

Adaptive Dynamic Shuffle Convolutional Parallel Network for Image Super-Resolution

Yiting Long ¹, Haoyu Ruan ^{2,*} , Hui Zhao ^{2,*}, Yi Liu ¹, Lei Zhu ¹ , Chengyuan Zhang ² and Xinghui Zhu ¹

¹ College of Information and Intelligence, Hunan Agricultural University, Changsha 410128, China; lyt@hunau.edu.cn (Y.L.); yiliu@hunau.edu.cn (Y.L.); leizhu@hunau.edu.cn (L.Z.); zhuxh@hunau.edu.cn (X.Z.)

² College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China; cyzhangcse@hnu.edu.cn

* Correspondence: rhyy@hnu.edu.cn (H.R.); timecomet@hnu.edu.cn (H.Z.)

Abstract: Image super-resolution has experienced significant advancements with the emergence of deep learning technology. However, deploying highly complex super-resolution networks on resource-constrained devices poses a challenge due to their substantial computational requirements. This paper presents the Adaptive Dynamic Shuffle Convolutional Parallel Network (ADSCP), a novel lightweight super-resolution model designed to achieve an optimal balance between computational efficiency and image reconstruction quality. The ADSCP framework employs large-kernel parallel depthwise separable convolutions, dynamic convolutions, and an enhanced attention mechanism to optimize feature extraction and improve detail preservation. Extensive evaluations on standard benchmark datasets demonstrate that ADSCP achieves state-of-the-art performance while significantly reducing computational complexity, making it well-suited for practical applications on devices with limited computational resources.

Keywords: image super-resolution; lightweight model; large-kernel convolutions; dynamic convolution; attention mechanisms



Citation: Long, Y.; Ruan, H.; Zhao, H.; Liu, Y.; Zhu, L.; Zhang, C.; Zhu, X. Adaptive Dynamic Shuffle Convolutional Parallel Network for Image Super-Resolution. *Electronics* **2024**, *13*, 4613. <https://doi.org/10.3390/electronics13234613>

Academic Editor: Dah-Jye Lee

Received: 29 September 2024

Revised: 8 November 2024

Accepted: 11 November 2024

Published: 22 November 2024



Copyright: © 2024 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

Image super-resolution (SR) [1–3] refers to the process of reconstructing or enhancing a low-resolution image to obtain a higher-resolution version, with more detailed and finer visual features. It aims to increase the spatial resolution of an image, which typically involves predicting or estimating the missing high-frequency details from the available low-frequency data. SR is crucial for applications where high-quality images are necessary but only low-resolution versions are available, such as in medical imaging [4,5], satellite photography [6,7], image recognition [8–10] and video streaming [11–13]. Conventional SR techniques, such as interpolation or statistical model-based approaches, exhibit limitations in reconstructing high-frequency details of images. In contrast, deep learning technologies [14–18], particularly through models like convolutional neural networks (CNNs) [19–22], have significantly enhanced the performance of SR. By employing an end-to-end training framework, deep learning models are capable of automatically learning the complex mapping between low-resolution and high-resolution images from large-scale datasets, thereby circumventing the constraints associated with manually designed feature extraction.

Most current deep learning-based methods [1,23–25], such as FSRCNN [26], directly utilize low-resolution images as input, modifying the nonlinear mapping approach of SRCNN [27] by initially reducing the dimensionality and subsequently upscaling it, with convolutional layers employed for feature extraction in between. Following this, DRCN [28] introduced a structure comprising three components: an embedding network, an inference network with recursive layers that increase network depth without significantly increasing the number of parameters, and a reconstruction network. DRRN [29] builds upon these

algorithms by incorporating concepts from DRCN and utilizing local skip connections inspired by ResNet [19]. It further integrates global skip connections and recursive blocks to deepen the network while controlling the parameter count. EDSR [30], through an analysis of the residual blocks in SRResNet [31], observed that batch normalization layers tend to distort image color and contrast, negatively affecting image quality. Consequently, they removed the batch normalization layers, simplifying the residual blocks and stacking more of them to increase the model's capacity. In addition, ESRGCNN [23] introduced an enhanced group convolutional strategy to improve feature extraction, while DSRNet [32] utilized dynamic networks to adaptively adjust its internal structure based on input characteristics. Moreover, HDSRNet [33] proposed a heterogeneous dynamic convolutional network to balance computational cost and performance in SR tasks.

While these deep models substantially enhance the reconstruction performance of super-resolution images, they generally achieve this by increasing network depth through direct block stacking or recursive structures. This approach often results in challenging training processes, high computational complexity, slow processing speeds, and a large number of parameters. Furthermore, these networks do not adequately account for the effects of convolutional structures and components on super-resolution performance, thereby imposing a considerable computational burden. Notably, very deep networks such as VDSR [34] incur significant computational costs due to their depth and large parameter scale, rendering them impractical for mobile devices and real-time applications.

To address the aforementioned challenges, it is crucial to design lightweight SR models to balance model complexity, inference time, and high-resolution reconstruction quality. Recent research efforts have focused on various strategies to reduce the model scale and enhance their efficiency. For instance, Ahn et al. [35] reduced the depth of a new designed Cascading Residual Network (CARN), resulting in the CARN-M model, though with a slight performance trade-off. Similarly, IDN [36] is a lightweight information distillation model constructed by a channel-splitting strategy. Building on this, Information Multi-Distillation Network (IMDN) [37] was designed, which adopts a residual architecture, utilizing multi-distillation through channel-splitting, feature concatenation and contrast-based channel attention, thereby further reducing dimensionality. Liu et al. [38] developed the Residual Feature Distillation Network (RFDN) by replacing the channel-splitting operation of IMDN with multiple feature distillation connections, thereby learning more discriminative feature representations. Kong et al. [39] refined RFDN by replacing feature distillation connections with stacked convolutional layers and activation functions for feature extraction. They also refined the activation functions within the feature extractor, combined with contrastive loss and a warm-start learning rate decay strategy, leading to improved feature extraction capability and enhanced model compactness, while maintaining a balance between accuracy and inference speed. Recent architectures such as Transformers and MLPs have also brought new directions for lightweight SR research. Zhang et al. [40] introduced the concept of superpixel, clustering similar local pixels to form similar feature regions, and proposed a Superpixel Token Interaction Network with intra- and cross-attention blocks, significantly enhancing model interpretability, which resulted in a more compact model with faster inference speeds.

Motivation. Despite the advancements made by existing methods, the trade-off between super-resolution (SR) performance and model compactness remains a persistent challenge. To address this issue, a promising emerging approach involves utilizing large-kernel convolutions for image feature extraction. Notably, AlexNet [41], a pioneering model employing large-kernel convolutions, demonstrated remarkable performance, inspiring subsequent research in the image SR domain. Specifically, large-kernel strided convolutions are often employed in attention mechanisms to capture salient local features from images. While depthwise separable convolutions offer advantages in terms of reduced parameter count and computational complexity, they convolve input features channel-wise, limiting the model's ability to fully capture spatial information across different channels. ShuffleMixer [42], for instance, leverages large-kernel depthwise separable convolutions

to construct a lightweight SR model by incorporating channel projection and shuffling to activate features along the channel dimension, facilitating information exchange between grouped features. However, this approach fails to enhance spatial feature representation or improve feature interactions between neighboring regions in the image, thereby affecting the SR reconstruction quality. Consequently, how to effectively apply large-kernel depthwise separable convolutions to lightweight SR tasks is still a challenge for further research.

Our Method. To address the challenges outlined, this paper introduces a novel **Adaptive Dynamic Shuffle Convolutional Parallel Network (ADSCPN)**. First, the proposed model incorporates a *large-kernel parallel depthwise separable convolution technique*, which enhances the network's ability to capture local features from various orientations, thereby expanding the receptive field and improving the extraction of fine details. This approach effectively mitigates the risk of feature loss, leading to more accurate image reconstruction. Second, the method integrates a *dynamic convolution mechanism*, which leverages an attention mechanism to selectively emphasize key deep features, striking a balance between computational efficiency and robust feature extraction. Moreover, ADSCPN employs a *point-attention convolution group (PACG)*, which refines deep feature representations by applying point-wise attention, thereby minimizing the risk of feature distortion during training. Additionally, the proposed model introduces an *Efficient Enhanced Spatial Attention (EESA) block*, which streamlines the traditional spatial attention mechanism to preserve performance while reducing computational complexity. Collectively, these techniques enable the SR model to reconstruct high-quality images with lower computational costs, making it highly suitable for deployment on resource-constrained devices.

Contributions. The main contributions of this paper can be summarized as follows:

- We propose a novel lightweight network architecture, named ADSCPN, designed to enhance feature learning across both channel and spatial dimensions. This innovative approach markedly improves the performance of image SR reconstruction.
- We introduce an innovative feature processing module, incorporating multi-layer pointwise perceptrons and shuffling attention, which enhances feature extraction through grouped convolutions and channel shuffling. Additionally, the framework integrates large-kernel convolution groups with dynamic convolutions, effectively reducing model size while maintaining high computational efficiency. These advancements enable the model to capture deep semantic features with low complexity, leading to a significant improvement in the quality of reconstructed images.
- We conduct extensive experiments on multiple benchmark datasets to evaluate the performance of our method. The results demonstrate that our model achieves superior performance compared to several state-of-the-art approaches, while maintaining low computational complexity.

Roadmap. The remainder of this paper is organized as follows: Section 2 provides an overview of related work on lightweight image SR techniques. Section 3 details the proposed methodology, including problem formulation and technical implementation. Section 4 presents the experimental results, while Section 5 concludes the paper with a summary of the key findings and contributions.

2. Related Work

Image super-resolution technology has advanced significantly in recent years, largely driven by deep learning innovations [42–44]. Numerous methods have been proposed to enhance image resolution, offering diverse solutions to this challenge. Building upon these foundations, our study introduces several extensions and innovations. This section reviews related work, covering traditional super-resolution techniques, deep learning-based approaches, lightweight SR methods, channel and spatial attention mechanisms, and the application of large-kernel convolutions.

2.1. Traditional Super-Resolution Methods

Early image SR techniques predominantly relied on interpolation algorithms, such as bilinear and bicubic interpolation. These approaches generate high-resolution images by spatially interpolating pixel values, but they often fail to accurately recover fine details and edge structures [45,46]. In response, methods based on sparse representation and dictionary learning were introduced, which reconstruct SR images by learning sparse representations of low-resolution and high-resolution patches, thereby achieving improved preservation of image details [47]. Despite their advantages, these methods are computationally demanding and exhibit limited effectiveness when applied to complex scenes.

2.2. Deep Learning-Based Super-Resolution Methods

With the advent of deep learning, SR methods based on CNNs have garnered significant attention. SRCNN [27] was among the first CNN-based models, achieving notable performance improvements by learning a direct mapping from low-resolution to high-resolution images. Building on this, VDSR [34] introduced a deeper network architecture, further improving SR performance through residual learning. EDSR [30] enhanced reconstruction quality by eliminating redundant batch normalization layers. Although these deep learning-based approaches substantially improve image reconstruction, particularly in restoring high-frequency details, their high computational demands limit their feasibility on resource-constrained devices.

2.3. Lightweight Super-Resolution Methods

To balance reconstruction quality with computational complexity, researchers have developed various lightweight SR models. For instance, MobileNet [48] and ShuffleNet [49] significantly reduce model parameters and computational overhead by leveraging depth-wise separable convolutions and group convolutions, making them well-suited for resource-constrained environments such as mobile devices. Additionally, some approaches have incorporated channel attention mechanisms, such as SENet [50], to further enhance model efficiency and performance. While these lightweight models exhibit strong performance under resource limitations, they still face challenges in accurately reconstructing complex scenes.

2.4. Channel and Spatial Attention Mechanisms

The attention mechanism is instrumental in enhancing the feature extraction capabilities of SR networks. Channel attention mechanisms improve reconstruction quality by assigning importance to feature maps across channels, allowing the network to focus on critical features [50]. In contrast, spatial attention mechanisms emphasize distinct spatial regions of the image, thereby further enhancing the network's ability to capture localized features [51]. The integration of these mechanisms into lightweight SR models significantly enhances both their efficiency and performance.

2.5. Application of Large-Kernel Convolutions

Large-kernel convolutions have gained increasing attention in computer vision tasks in recent years. By expanding the receptive field, they can more effectively capture global information of images. For lightweight super-resolution tasks, researchers have proposed decomposing large-kernel convolutions into multiple small-kernel convolutions, combining them with dynamic convolutions and group convolutions to achieve efficient feature extraction [52,53]. These methods significantly improve the recovery of image details while maintaining low computational complexity, but further optimization of the application of large-kernel convolutions in super-resolution tasks remains an open question.

2.6. Summary

Although considerable advancements have been made in enhancing image super-resolution quality and reducing computational complexity, further performance improve-

ments are needed for more complex scenarios. The ADSCP model introduced in this paper establishes a novel balance between image reconstruction quality and computational efficiency by integrating large-kernel parallel convolutions, dynamic convolutions, and both channel and spatial attention mechanisms. Compared to existing approaches, ADSCP excels across multiple benchmark tests, showcasing its substantial potential for application, particularly in resource-constrained environments.

3. Methodology

This section provides an in-depth examination of the proposed ADSCP model. We commence by defining the relevant notations and problem. Subsequently, we present a detailed description of each model component. The section concludes with a discussion of the model optimization algorithm.

3.1. Notations and Problem Definition

Notations. Without loss of generality, the lightweight image SR is aiming to reconstruct a high-resolution (HR) image \mathbf{I}_{HR} from its low-resolution (LR) counterpart \mathbf{I}_{LR} . We denote the input low-resolution image as $\mathbf{I}_{LR} \in \mathbb{R}^{H \times W \times C}$, where H , W , and C represent the height, width, and number of channels of the image, respectively. The high-resolution image is denoted as $\mathbf{I}_{HR} \in \mathbb{R}^{rH \times rW \times C}$, where r is the scaling factor. Let \mathbf{F}_{enc} be the encoded shallow features extracted from the input LR image \mathbf{I}_{LR} using a shallow feature extraction function $f_{enc}(\cdot)$, and \mathbf{F}_K represents the deep features processed through K feature processing blocks. The output SR image \mathbf{I}_{SR} is obtained after passing through a series of upsampling and reconstruction layers. Additionally, \mathbf{F}_{proj} and \mathbf{F}_{spat} represent the features processed by the channel projection block and the spatial attention block, respectively. The attention weights computed for the spatial features are denoted by \mathbf{A}_{spat} . For readability, the notations frequently used in this paper are summarized in Table 1.

Table 1. The summary of frequently used notations.

Notation	Definition
\mathbf{I}_{LR}	Low-resolution input image
\mathbf{I}_{HR}	High-resolution target image
\mathbf{I}_{SR}	Super-resolution output image
r	Scaling factor
H, W, C	Height, width, and channels of the image
\mathbf{F}_{enc}	Shallow feature maps extracted from \mathbf{I}_{LR}
\mathbf{F}_K	Deep features after processing through K blocks
\mathbf{F}_{proj}	Features after channel projection
\mathbf{F}_{spat}	Features after spatial attention block
\mathbf{A}_{spat}	Attention weights for spatial features

Problem Definition. Given a training dataset $\mathcal{D} = \{(\mathbf{I}_{LR}^{(i)}, \mathbf{I}_{HR}^{(i)})\}_{i=1}^N$, where N is the total number of training samples, our goal is to learn a mapping $f_{\theta}(\cdot; \theta)$ that transforms the low-resolution image \mathbf{I}_{LR} into a high-resolution image \mathbf{I}_{SR} such that the difference between \mathbf{I}_{SR} and the ground-truth high-resolution image \mathbf{I}_{HR} is minimized. The objective function is the average loss over \mathcal{D} :

$$\min_{\theta} \frac{1}{N} \sum_{i=1}^N \mathcal{L}(\mathbf{I}_{HR}^{(i)}, f_{\theta}(\mathbf{I}_{LR}^{(i)}; \theta)). \quad (1)$$

This optimization process ensures that the learned model $f_{\theta}(\cdot; \theta)$ is capable of generating high-quality super-resolution images with low computational complexity, making it suitable for deployment on resource-constrained devices.

3.2. Architecture of ADSCP

The architecture of ADSCP is illustrated in Figure 1, comprising three primary components. Given an input low-resolution image I_{LR} , the model first processes it through a shallow feature encoder, f_{enc} , which consists of a single 3×3 convolutional layer. The resulting feature map, denoted as $X_{enc} = f_{enc}(I_{LR})$ is subsequently passed through K Feature Processing Blocks, denoted as f_{FE} , which incorporate an innovative combination of large-kernel parallel depthwise separable convolutions, attention mechanisms, and grouped convolutions. Following this, the features are processed by a single 3×3 convolutional layer, $f_{3 \times 3}$, integrated with a residual connection. The final stage involves the upsampling module, f_{UP} , which is consistent with the structure outlined in Section 3. This module utilizes pixel shuffle operations and 3×3 convolutional layers to reconstruct the super-resolution image I_{SR} with the desired scaling factor. The entire process is mathematically formalized as follows:

$$I_{SR} = f_{3 \times 3} \left(f_{UP} \left(X_{enc} + f_{3 \times 3} \left(f_{FE}^K \left(f_{FE}^{K-1} \left(\dots f_{FE}^1 (X_{enc}) \right) \right) \right) \right) \right). \tag{2}$$

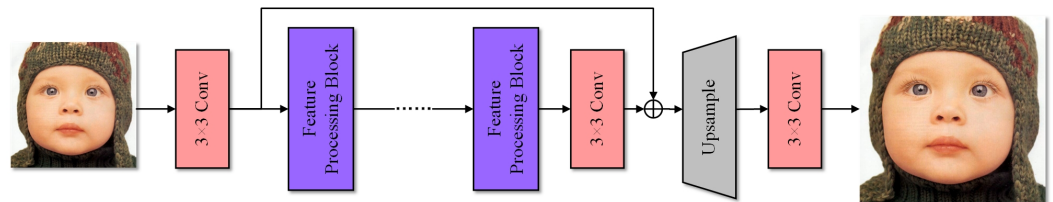


Figure 1. The architecture of ADSCP.

3.3. Feature Processing Block

The Feature Processing Block, depicted in Figure 2, is composed of several novel modules: the Channel Processing Block (CPB), Feature Extraction Block (FEB), Point-Attention Convolution Group (PACG), and Efficiently Enhanced Spatial Attention (EESA) module. These modules collaboratively perform channel projection and feature shuffle attention learning, effectively enhancing feature extraction, calibration, and spatial augmentation. A residual structure is integrated before and after the Feature Extraction Block to stabilize the training process and facilitate the recovery of low-frequency textures.

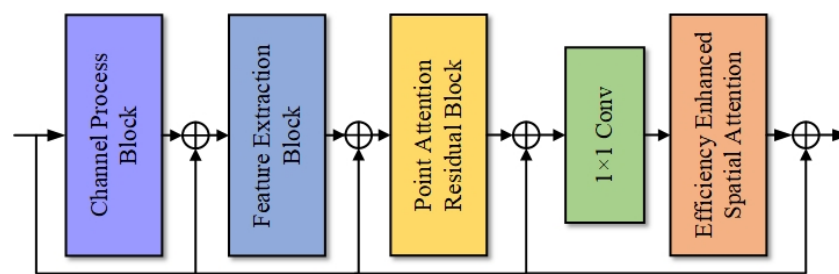


Figure 2. The structure of the Feature Process Block (FPB).

The input to the i -th Feature Processing Block is denoted as X_i . The channel processing block (CPB) is designed to project and shuffle spatial features, thereby enhancing deep feature extraction while maintaining computational efficiency. The input is subsequently passed through a 3×3 convolutional layer and a residual structure, further refining deep feature representations and supporting shallow feature processing. The output is then fused with the original input, followed by parallel operations involving a 1×1 convolution and the Efficiently Enhanced Spatial Attention (EESA) module, which calibrates the spatial features by focusing on the most salient information. The computational flow can be formalized as follows:

$$X_{i+1} = X_i + f_{EESA} \left(f_{1 \times 1} \left(f_{CPB} (X_i) + f_R \left(f_{FE} (X_i + f_{CP} (X_i)) \right) \right) \right). \tag{3}$$

3.3.1. Channel Processing Block

The channel processing block (CPB) is the first component of the feature processing block. It is composed of a point-wise multi-layer perceptron (PW-MLP), a 3×3 convolution, and Shuffle Attention, focusing on extracting information from feature channels and spatial dimensions, as shown in Figure 3. The multi-layer point-wise perceptron projects and activates important low-frequency feature information from the input channels through expansion and contraction operations, thereby enhancing the channel features. The 3×3 convolution further activates the channel features and extracts deep local spatial information. The learned features are then fed into the Shuffle Attention module, where grouped channel features are processed. The module calculates attention within and across channels and spatial dimensions, enhancing the efficiency of the attention computation while promoting information fusion and interaction among grouped features. Since the Shuffle Attention block can easily lose part of the feature information, residual connections are added around the channel processing block to supplement shallow features.

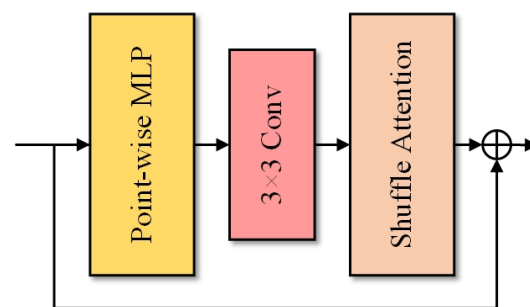


Figure 3. The structure of the channel processing block (CPB).

As the first component of the channel processing block, the PW-MLP facilitates the encoding of spatial information within channels, activates channel features, and lays the foundation for subsequent local feature extraction and channel-wise grouped shuffle attention learning. As shown in Figure 4, it is composed of a 1×1 convolution, a GeLU activation function, and another 1×1 convolution in sequence. Given an output feature map of size $H \times W \times C$, the channel expansion factor is denoted as ϕ . The channel dimension undergoes expansion and contraction processing to activate the channel features. After processing by the first 1×1 convolution, the feature map size is expanded as $H \times W \times C \times \phi$, which is then reduced by the subsequent 1×1 convolution to restore the original size. In our solution, the expansion factor ϕ is set to 2.

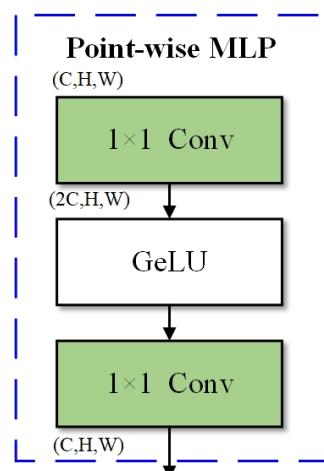


Figure 4. The structure of point-wise multi-layer perceptron (PW-MLP).

3.3.2. Feature Extraction Block

The structure of the newly designed feature extraction block is shown in Figure 5. The first component is a large-kernel parallel convolution processing unit, consisting of a 1×7 depthwise separable convolution, a 7×1 depthwise separable convolution, and a 1×1 group convolution. The input to this block, \mathbf{X}' , is the low-frequency feature map generated by the CPB module. The 1×7 and 7×1 convolutions process the same input to capture local features in both horizontal and vertical directions, enhancing the extracted image features while filling in nearby spatial details. This design improves the network's ability to capture local features and strengthens relationships between surrounding pixels. Compared to a single 7×7 depthwise separable convolution, this unit delivers a more robust performance.

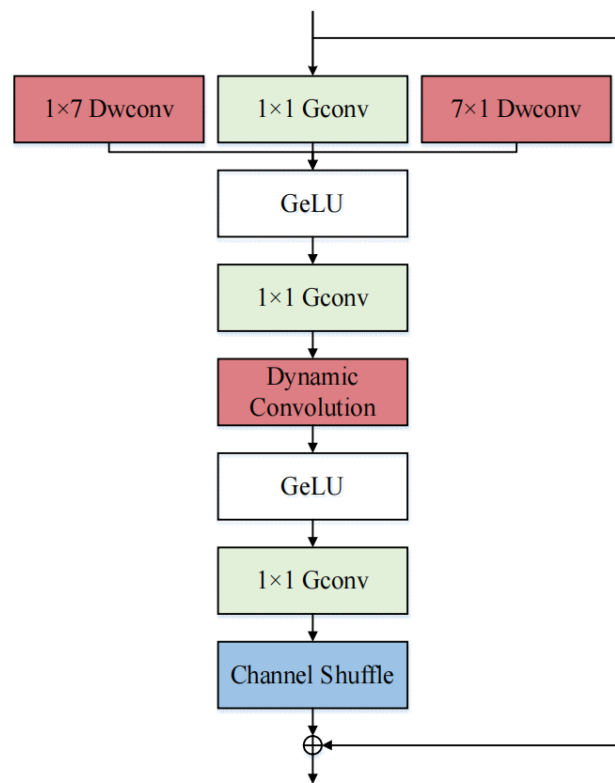


Figure 5. The structure of the feature extraction block (FEB).

Following the large-kernel parallel convolution processing, the two branches are fused into a single output, denoted as \mathbf{F}_{BP} . This output is activated by the GeLU function and passed through a 1×1 group convolution to further enhance feature aggregation and learning efficiency. The processed feature map, \mathbf{F}_{BP} , is then fed into a dynamic convolution with four adjustable convolutional kernels. By modulating the convolutional kernels and adaptively adjusting parameters, this dynamic convolution balances computational cost while significantly improving representational capacity. The output, denoted as \mathbf{X}_{FE} , is subsequently refined through a 1×1 group convolution, activated by GeLU, and followed by a channel shuffle for final feature aggregation. This output is then fused with the original input and processed by $f_{shuffle}$ to facilitate grouped feature interaction and aggregation. This process effectively refines the image's high-frequency features, promoting stable gradient flow and enhancing overall model performance. The entire process of the feature extraction block (FEB) can be expressed as follows:

$$\mathbf{F}_{BP} = f_{1 \times 7, Dwconv}(\mathbf{X}') + f_{1 \times 1, Gconv}(\mathbf{X}') + f_{7 \times 1, Dwconv}(\mathbf{X}'), \quad (4)$$

$$\mathbf{X}_{BP} = f_{1 \times 1, Gconv}(GeLU(\mathbf{F}_{BP})), \quad (5)$$

$$\mathbf{X}_{FE} = \mathbf{X}' + f_{shuffl}(f_{1 \times 1, Conv}(GeLU(f_{Dynamic}(\mathbf{X}_{BP}))))). \tag{6}$$

3.3.3. Point-Attention Convolution Group

Similar to EDSR [30], where the model is primarily composed of residual blocks interconnected by 3×3 convolutions and activation functions, we introduce a novel point-attention convolution group placed at the end of the feature processing block to further deepen the network. This component performs weighted learning on the deep features processed by the convolution group, enhancing and refining them after convolutional processing. To mitigate potential feature loss during deep model training, these refined features are merged with the output of the convolution group. The convolution group comprises two 3×3 convolutional layers connected in series, as depicted in Figure 6. The original input \mathbf{X}_{FE} passes through the convolution group, producing an intermediate output \mathbf{X}_{RB} . This is followed by a point-attention block, consisting of a 1×1 convolution and a Sigmoid activation function, which learns attention weights for the deep features, extracting the most significant ones. The attention-weighted deep features, denoted as \mathbf{X}_i , are multiplied with \mathbf{X}_{RB} to calibrate the convolutional output and minimize feature loss, resulting in the enhanced deep semantic features \mathbf{X}'_{RB} . Finally, the enhanced features are fused with the original \mathbf{X}_{RB} to produce the final output, \mathbf{X}''_{RB} . The entire process is mathematically expressed as follows:

$$\mathbf{X}_{RB} = f_{3 \times 3}(f_{ReLU}(f_{3 \times 3}(\mathbf{X}_{FE}))), \tag{7}$$

$$\mathbf{X}_i = f_{Sigmoid}(f_{1 \times 1}(\mathbf{X}_{RB})), \tag{8}$$

$$\mathbf{X}'_{RB} = \mathbf{X}_{RB} \times \mathbf{X}_i, \tag{9}$$

$$\mathbf{X}''_{RB} = \mathbf{X}_{RB} + \mathbf{X}'_{RB}. \tag{10}$$

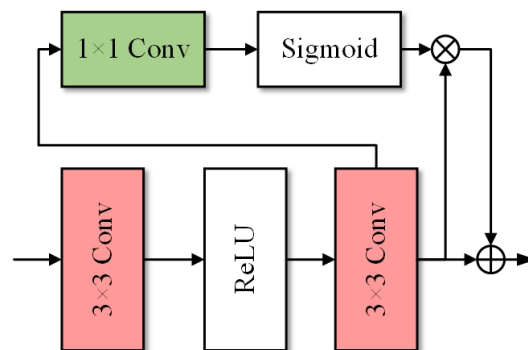


Figure 6. The structure of the point-attention convolution group (PACG).

3.3.4. Efficiently Enhanced Spatial Attention

The lightweight Enhanced Spatial Attention (ESA) module [38], frequently employed in various lightweight super-resolution models, enhances feature representation capabilities while maintaining efficiency and ease of integration. As depicted in Figure 7a, the ESA architecture includes a 1×1 convolution for channel dimension reduction, followed by a 3×3 convolution and a max pooling layer with a large receptive field, before upsampling back to the original resolution. Subsequently, a 1×1 convolution is applied to restore the original channel dimensions, and the output is processed through a Sigmoid activation function to generate the spatial attention map. The ESA has been further optimized into the Efficiently Enhanced Spatial Attention (EESA) module [54], as shown in Figure 7b. In this modified version, the 3×3 convolution and upsampling operations are replaced with a single 3×3 convolution, effectively reducing and recovering the channel dimensions without the need for additional convolutions in the residual connection. This adjustment reduces computational complexity while preserving essential feature details. Additionally,

a pooling layer with a 7×7 window size and a stride equivalent to the original max pooling layer is introduced to further expand the receptive field. Despite its simplified design, this lightweight version maintains a comparable performance level to the original ESA module with minimal impact on accuracy.

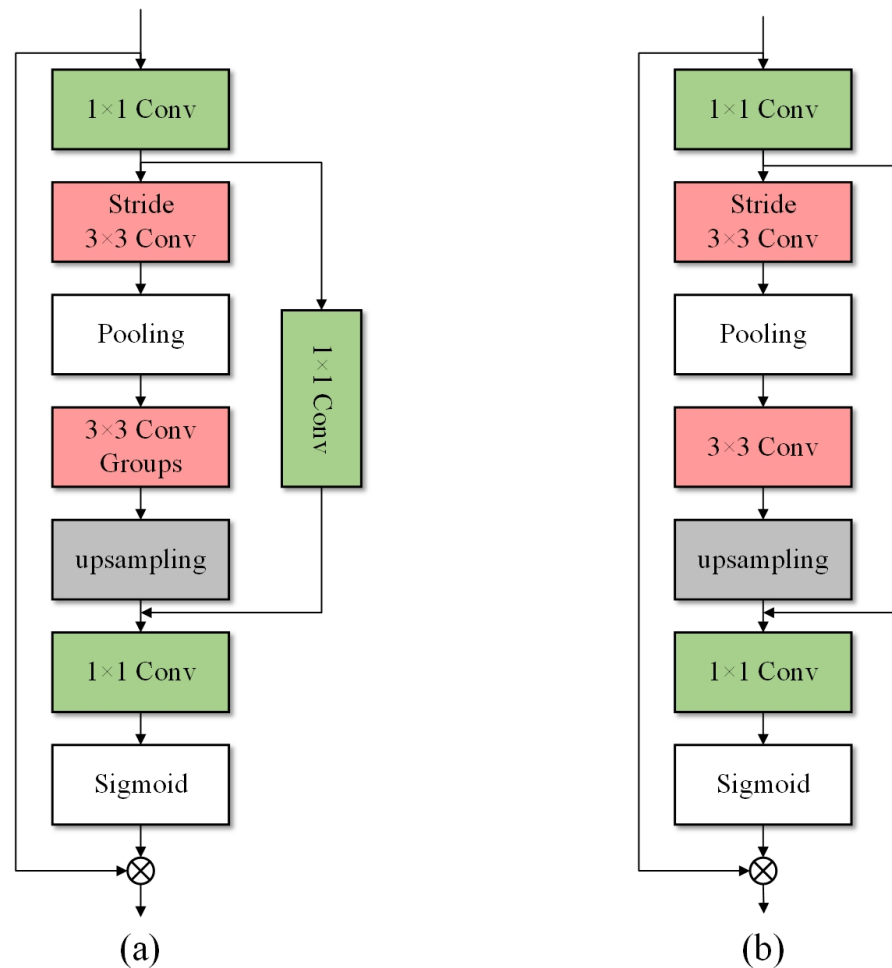


Figure 7. ESA and EESA module structure comparison. (a) Original ESA block. (b) EESA block.

3.4. Optimization

The optimization procedure of the ADSCPN is outlined in Algorithm 1. Initially, the training dataset undergoes preprocessing, followed by data augmentation to enhance the training scale. Key hyperparameters, including training epochs, batch size, number of iterations, and learning rate, are configured, along with a reduction in the total number of epochs. Subsequently, the model parameters are initialized. As training progresses, learning rate decay is applied based on the current epoch number and iteration count.

For each batch, a fixed number of high-resolution and low-resolution image pairs are input into the model for feature extraction. During this phase, shallow features are encoded, and deep semantic features are extracted. The features are then fused to generate comprehensive semantic representations of the images. The upsampling module reconstructs the corresponding super-resolution images. The $L1$ loss is computed between the reconstructed super-resolution images and their corresponding high-resolution ground truth. Model parameters are iteratively updated using the Adam optimizer through back-propagation. This process continues until the optimal model is achieved and is validated through performance metrics.

Algorithm 1 The pseudo-code of ADSCN Algorithm

Input: Training dataset \mathcal{D} ; training period T ; batch size N ; image crop size P ; number of iterations I ; learning rate η ; model channel C ; number of groups G ; number of feature processing blocks N_f ; batch size N , with each batch size as $[N, \text{channels}, \text{height}, \text{width}]$;

Output: Super-resolution image \mathbf{I}_{SR} ; model parameters θ

- 1: Initialize the training environment, crop high-resolution images from the training dataset into low-resolution images based on the crop size P , and adjust the channels.
- 2: Calculate the number of image blocks M required according to the iteration I and batch size N .
- 3: Construct a list of corresponding high-resolution and low-resolution image blocks $[[\mathbf{I}_{LR}^1, \mathbf{I}_{SR}^1], \dots, [\mathbf{I}_{LR}^m, \mathbf{I}_{SR}^m]]$.
- 4: Build the ADSCPN model, set the convolution kernel size, input channels as C , build the upsampling module based on the scale factor, stack N_f feature processing blocks, and initialize the model parameters θ .
- 5: **for** each $t = 1$ to I **do**
- 6: Adjust the learning rate;
- 7: **for** each pair of high-resolution and low-resolution image blocks $t = 1$ to N **do**
- 8: Feed the low-resolution image \mathbf{I}_{LR} into ADSCPN, perform feature extraction and multiple feature processing blocks, and obtain the corresponding super-resolution image \mathbf{I}_{SR} ;
- 9: Calculate the loss function \mathcal{L} between the generated super-resolution image \mathbf{I}_{SR} and the corresponding high-resolution image \mathbf{I}_{HR} ;
- 10: **end for**
- 11: Update model parameters θ using the Adam optimizer and backpropagation;
- 12: **if** convergence is reached **then**
- 13: **break**;
- 14: **end if**
- 15: **end for**

4. Experiment

This section outlines a comprehensive experimental evaluation conducted on several standard benchmark datasets. We begin by detailing the experimental setup, encompassing the datasets utilized, evaluation metrics, baseline models, and implementation specifics. Following this, we present a performance analysis derived from multiple experimental groups.

4.1. Dataset

To comprehensively evaluate the performance of the proposed super-resolution model, a series of experiments were conducted using several widely-recognized datasets. The **DIV2K** dataset [55] was employed for model training, while evaluations were performed on standard benchmark datasets, including **Set5** [56], **Set14** [57], **B100** [58], and **Urban100** [59]. Furthermore, to assess the model's robustness under diverse real-world conditions, we incorporated blind super-resolution datasets such as **DIV2KRRK** [60], **DRealSR** [61], and **RealSR** [62]. These datasets provide varying levels of real-world complexity, with **DRealSR** and **RealSR** capturing real-world noise and degradation processes, making them more challenging than the original benchmark datasets like **Set5** [56] and **B100** [58], which primarily consist of controlled synthetic degradations. We provide a brief introduction to these datasets below:

- **DIV2K.** The **DIV2K** dataset comprises 1000 high-resolution images, characterized by a rich variety of features and diverse scene categories, including environments, humans, and other objects. For experimental purposes, the first 800 images are designated for training, images 801–900 are allocated to the validation set, and the remaining 100 images are reserved for performance evaluation. Each image in the dataset has a native 2 K resolution, accompanied by corresponding low-resolution images generated through the application of various degradation kernels, such as Gaussian white noise.

This dataset is widely utilized for training and benchmarking super-resolution models due to its diversity and high-quality annotations.

- **Set5 and Set14.** The Set5 and Set14 datasets are among the earliest small-scale benchmarks introduced in the field of image processing. Set5 contains 5 images, while Set14 consists of 14 images, both encompassing a mix of human subjects and natural scenes. Despite their limited size, these datasets remain widely used for testing and validation purposes, offering a convenient and efficient means of assessing model performance, particularly in super-resolution tasks.
- **B100.** The B100 dataset comprises 100 images depicting a variety of subjects, including animals, plants, and real-world scenes. While offering a diverse range of content, the images in this dataset are relatively small in resolution and exhibit less detailed textures compared to other high-resolution benchmarks, making it a useful resource for evaluating super-resolution models under more constrained conditions.
- **Urban100.** The Urban100 dataset contains 100 high-resolution images, specifically focused on capturing intricate details of urban architecture. These images provide complex structural patterns and fine-grained textures, making the dataset particularly challenging and suitable for evaluating the performance of super-resolution models on architectural and man-made scenes.
- **DRealSR and RealSR.** The RealSR and DRealSR datasets are created using DSLR cameras with multiple zoom levels, varying aperture settings, and different lens focal lengths to simulate signal noise introduced during the degradation process. An image registration algorithm is employed to precisely align high- and low-resolution image pairs. These datasets encompass a wide range of scenes, including natural landscapes, architectural structures, as well as human and animal subjects. Both datasets are widely used for real-world image super-resolution reconstruction tasks, providing realistic and challenging conditions for model evaluation.

4.2. Evaluation Metrics

Quantitative evaluation of the model's performance was carried out using the Peak Signal-to-Noise Ratio (**PSNR**) and Structural Similarity Index (**SSIM**) metrics [63], ensuring a thorough comparison of the model's accuracy and perceptual quality across different datasets.

PSNR is a widely used metric for evaluating the quality of reconstructed images at the pixel level by measuring the pixel-wise differences between two images. A higher PSNR value indicates that the reconstructed super-resolution image is closer to the original high-resolution image, suggesting improved reconstruction accuracy. However, PSNR does not account for structural differences or human visual perception, which can lead to high PSNR values even when the reconstructed image visually deviates from the original, particularly in local regions. To compute PSNR, the Mean Squared Error (MSE) between two images, I_1 and I_2 , must first be calculated, where W and H represent the width and height of the image, respectively. The mathematical expression for MSE is given by:

$$\text{MSE} = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (I_1(i, j) - I_2(i, j))^2. \quad (11)$$

The PSNR metric is then calculated as follows:

$$\text{PSNR} = 10 \log_{10} \left(\frac{(2^n - 1)^2}{\text{MSE}} \right), \quad (12)$$

where $2^n - 1$ represents the maximum possible pixel value in the image, with $n = 8$ for typical grayscale images. For RGB images, the calculation is performed only on the Y channel after converting the image to the YCbCr color space.

SSIM evaluates the overall structural similarity between the reconstructed super-resolution image I_{SR} and the original high-resolution image I_{HR} . SSIM is computed by

considering three key components: luminance similarity, contrast similarity, and structural similarity. The SSIM value ranges from 0 to 1, with values closer to 1 indicating that the reconstructed image exhibits greater structural similarity to the original high-resolution image. Thus, a higher SSIM value reflects a more accurate representation of the original image's structural content. Assume that both \mathbf{I}_{SR} and \mathbf{I}_{HR} images contain N pixels. First, the average luminance μ and standard deviation σ of each image \mathbf{I} are calculated to quantify the luminance and contrast. Then, the normalized image \mathbf{I}' , which quantifies the structural information, is computed using the following formulas:

$$\mu = \frac{1}{N} \sum_{i=1}^N \mathbf{I}(i), \quad (13)$$

$$\sigma = \sqrt{\frac{1}{N-1} \sum_{i=1}^N (\mathbf{I}(i) - \mu)^2}, \quad (14)$$

$$\mathbf{I}'(i) = \frac{\mathbf{I}(i) - \mu}{\sigma}, i \in [1, \dots, N]. \quad (15)$$

Next, the luminance similarity l , contrast similarity c , and structural similarity s are calculated. These are expressed as follows:

$$l = \frac{2\mu_{HR}\mu_{SR} + c_1}{\mu_{HR}^2 + \mu_{SR}^2 + c_1}, \quad (16)$$

$$c = \frac{2\sigma_{HR}\sigma_{SR} + c_2}{\sigma_{HR}^2 + \sigma_{SR}^2 + c_2}, \quad (17)$$

$$s = \frac{\sigma_{HR,SR} + c_3}{\sigma_{HR}\sigma_{SR} + c_3}, \quad (18)$$

where μ_{HR} and μ_{SR} represent the mean luminance of the original high-resolution image and the super-resolution result, respectively. σ_{HR} and σ_{SR} represent the standard deviations, while $\sigma_{HR,SR}$ represents the covariance between the original high-resolution image and the super-resolution result. c_1 , c_2 , and c_3 are constants introduced to avoid division by zero. Finally, the SSIM value is calculated as:

$$\text{SSIM} = l^\alpha c^\beta s^\gamma, \quad (19)$$

where α , β , and γ are parameters that adjust the importance of luminance, contrast, and structural details in the calculation. These parameters are typically set to 1.

4.3. Baselines and Implementation Details

Baseline. We compared ADSCPN with several baselines, including Bicubic [46], SRCNN [27], FSRCNN [26], VDSR [34], DRCN [28], DRRN [29], CARN-M [35], CARN [35], MemNet [64], LapSRN [65], IDN [36], MADNet [66], ECBSR (Edge-oriented Convolution Block for Real-time Super-Resolution) [67], and FALSAR-A (Fast, Accurate, and Lightweight Super-Resolution) [68]. These methods were chosen as they represent state-of-the-art lightweight and efficient super-resolution techniques, covering a diverse range of architectures from early CNN-based methods (e.g., SRCNN [27], VDSR [34]) to more recent lightweight models (e.g., CARN [35], MADNet [66]), providing a comprehensive benchmark for comparison.

Implementation Details. During the training process, the Adam optimizer was used to update the network parameters, with hyperparameters configured as follows: $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and the initial learning rate of 5×10^{-4} . The batch size was set to 16, and the L1 loss function was employed as the model training loss criterion. All experiments were conducted on a workstation equipped with an Intel(R) Core i9-12900K 3.9 GHz CPU, 128 GB of RAM, 1 TB SSD storage, 2 TB HDD storage, and dual NVIDIA

GeForce RTX 3090 GPUs, operating under Ubuntu 18.04.1. The implementation of all techniques was carried out using Python 3.6 and PyTorch 1.8.1.

4.4. Performance Evaluation

4.4.1. Model Parameter Analysis

This section examines the influence of hyperparameter configurations, such as learning rate, number of layers, and batch size, on the performance of the ADSCP model. We conducted extensive experiments to understand how these hyperparameters impact the convergence speed, stability, and final quality of the reconstructed images.

The learning rate plays a critical role in training deep models effectively. In our approach, we employed a staged-halving learning rate schedule to guide the training process. During the initial stages, a higher learning rate was used to accelerate the convergence, enabling the model to quickly capture general patterns in the data. As training proceeded, the learning rate was halved at specific intervals to gradually slow down the learning process. This staged reduction helped stabilize the training, allowing the model to refine its parameters and converge to better local optima. This strategy proved effective, as it balanced fast convergence at the beginning with careful parameter tuning towards the end, leading to consistent improvements in PSNR and SSIM values across all test datasets.

We conducted an ablation study to determine the optimal number of Feature Process Blocks (also referred to as “layers”) in the model. As illustrated in Figure 8a, increasing the number of Feature Process Blocks initially led to significant improvements in PSNR on the B100 test set, establishing a positive correlation between model depth and reconstruction quality. However, when the number of blocks exceeded 6, the performance gains began to plateau, indicating diminishing returns. Based on these findings, we selected 6 blocks as the optimal configuration, which strikes a balance between model complexity and performance enhancement. Increasing the number of blocks beyond 6 resulted in increased computational costs without notable improvements, thus establishing 6 blocks as the best trade-off.

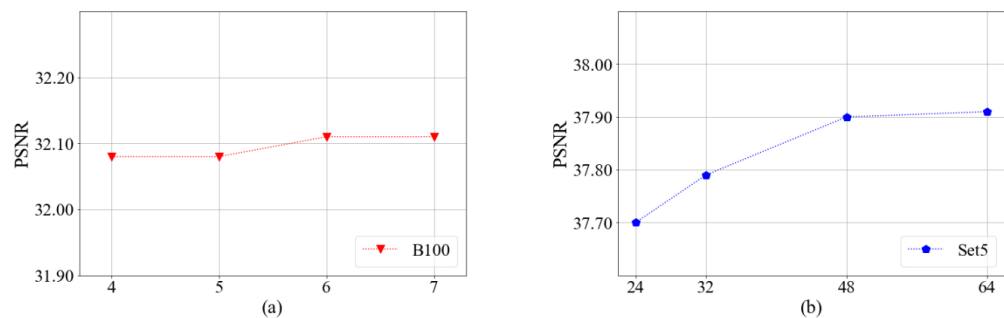


Figure 8. The impact of different settings on quantitative PSNR performance. (a) The number of Feature Process Blocks. (b) The number of model channels.

Figure 8b presents PSNR results for the Set5 test set, which demonstrates that increasing the number of channels improves reconstruction quality. However, as the number of channels surpasses 64, the performance gains become less pronounced, indicating a performance plateau. Given the increased complexity and computational burden of using more channels, we determined that 48 channels provide an optimal balance between reconstruction quality and model efficiency, as summarized in Table 2. This selection reduces computational requirements while maintaining high-quality output.

Table 2. Quantitative comparison of ADSCPN with different group numbers.

Group Number	Parameters (K)	MACs (G)	Set5	Set14	B100
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
2	762	651	37.92/0.9603	33.45/0.9165	32.10/0.8987
4	703	597	37.88/0.9602	33.44/0.9170	32.10/0.8989
6	684	579	37.90/0.9602	33.44/0.9169	32.10/0.8988
8	674	570	37.90/0.9602	33.51/0.9174	32.11/0.8990

The batch size used during training affects both the stability and efficiency of the optimization process. Inspired by hyperparameter settings from well-established super-resolution models like EDSR and RCAN, we experimented with different batch sizes and concluded that a batch size of 16 was ideal for ADSCPN. This value provided a good trade-off between convergence stability and GPU memory utilization, allowing efficient training without compromising on model quality. While larger batch sizes provided more stable gradient estimates, the performance gains diminished, and smaller batch sizes resulted in higher variance in gradients, which led to less stable training dynamics.

Furthermore, we analyzed the effect of the number of groups within the architecture through experiments involving the division of convolutional groups within the ADSCPN network. As reported, the model achieves optimal performance on the Set5 test set when configured with 8 groups. Increasing the number of groups further leads to reduced model parameters and a smaller memory footprint but with diminishing performance gains. Thus, we established 8 groups as the optimal configuration to balance between model complexity and overall performance.

These findings provide clear guidance for the optimal hyperparameter configuration of the ADSCPN model. Specifically, the staged-halving learning rate schedule allows for both rapid initial convergence and stable parameter refinement, while the chosen number of Feature Process Blocks, channels, and groups ensures that the model remains computationally efficient without sacrificing performance. A batch size of 16 further guarantees a smooth training process, optimizing GPU utilization and stability. These hyperparameter choices effectively balance learning efficiency, generalization, and practical deployment feasibility, making ADSCPN suitable for high-quality image reconstruction tasks.

4.4.2. Quantitative Experimental Results Comparison

In this section, we compare the performance of the proposed model with several popular and effective methods on standard single-image super-resolution benchmark datasets across various scaling factors. The specific results are summarized in Tables 3–5, where the best and second-best average PSNR and SSIM values are highlighted in **bold** and underlined, respectively. The comparative methods include Bicubic [46], SRCNN [27], FSRCNN [26], VDSR [34], LapSRN [65], DRCN [28], DRRN [29], CARN-M [35], CARN [35], MemNet [64], IDN [36], MADNet [66], ECB-SR [67], LINF [69], FPL [70] and FALSRA [68]. The results are averaged from the data reported in the respective studies.

When the scaling factor is set to 2, the ADSCPN-plus model consistently achieves either the best or second-best performance across all datasets. In particular, for scaling factors of $3\times$ and $4\times$, the ADSCPN model surpasses the ECB-SR and FALSRA methods, especially on the Urban100 dataset, which features more complex scenes, where it attains the highest performance. Notably, the PSNR and SSIM values for ADSCPN-plus are significantly higher than those of competing methods at scaling factors of $3\times$ and $4\times$, with an average PSNR increase exceeding 0.02 dB for ADSCPN-plus at a scaling factor of 3. Furthermore, both ADSCPN and ADSCPN-plus outperform other methods on the Urban100 and Set14 datasets in terms of PSNR at a scaling factor of $3\times$. It is important to highlight that the model is updated with only 1600 batches per epoch, resulting in a remarkably short training time while achieving these performance metrics. This demonstrates the effectiveness,

as well as the theoretical and practical value of the model as a lightweight solution for super-resolution tasks.

Table 3. Quantitative comparison of methods with scale factor 2.

Method	Scale Factor	Set5	Set14	B100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×2	33.66/0.9299	30.24/0.8688	29.56/0.8431	26.88/0.8403
SRCNN	×2	36.66/0.9542	32.42/0.9063	31.36/0.8964	29.50/0.8946
FSRCNN	×2	37.00/0.9558	32.63/0.9088	31.53/0.8920	29.88/0.9009
VDSR	×2	37.53/0.9590	33.05/0.9130	31.90/0.8960	30.77/0.9140
LapSRN	×2	37.52/0.9591	32.99/0.9134	31.80/0.8952	30.41/0.9100
DRCN	×2	37.63/0.9588	32.98/0.9130	31.85/0.8942	30.75/0.9133
DRRN	×2	37.74/0.9591	33.23/0.9145	32.05/0.8973	31.23/0.9188
CARN	×2	37.76/0.9590	33.52/0.9166	32.09/0.8978	31.92/0.9256
MemNet	×2	37.78/0.9597	33.28/0.9142	32.08/0.8978	31.31/0.9195
IDN	×2	37.83/0.9600	33.30/0.9148	32.08/0.8985	31.27/0.9196
FALSR-A	×2	37.82/0.9595	33.55/0.9168	32.12/0.8987	31.93/0.9256
LINF	×2	37.49/—	33.38/—	32.16/—	31.22/—
FPL	×2	35.73/0.9509	31.59/0.9059	30.75/0.9022	30.46/0.9257
ADSCPN	×2	37.84/0.9600	33.42/0.9165	32.06/0.8983	31.67/0.9238
ADSCPN-plus	×2	37.90/0.9602	33.51/0.9174	32.11/0.8990	31.93/0.9262

Table 4. Quantitative comparison of methods with scale factor 3.

Method	Scale Factor	Set5	Set14	B100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×3	30.39/0.8682	27.55/0.7742	27.21/0.7385	24.46/0.7349
SRCNN	×3	32.75/0.9090	29.30/0.8203	28.41/0.7863	26.24/0.8090
FSRCNN	×3	33.06/0.9140	29.43/0.8242	28.53/0.7910	26.43/0.8080
VDSR	×3	33.66/0.9213	29.77/0.8314	28.82/0.7976	27.14/0.8279
DRCN	×3	33.82/0.9226	29.76/0.8311	28.80/0.7963	27.15/0.8276
DRRN	×3	34.03/0.9244	29.96/0.8349	28.95/0.8004	27.53/0.8378
IDN	×3	34.11/0.9253	30.13/0.8360	29.01/0.8013	27.40/0.8359
CARN-M	×3	33.99/0.9245	30.08/0.8352	28.91/0.8000	27.47/0.8371
MemNet	×3	34.09/0.9248	30.00/0.8350	28.96/0.8001	27.56/0.8376
MADNet	×3	34.16/0.9253	30.21/0.8398	29.08/0.8023	27.82/0.8423
LINF	×3	33.94/—	29.84/—	28.55/—	28.39/—
ADSCPN	×3	34.21/0.9252	30.22/0.8398	28.99/0.8024	27.81/0.8444
ADSCPN-plus	×3	34.23/0.9258	30.24/0.8399	29.10/0.8033	28.78/0.8463

Table 6 presents the quantitative evaluation results for scaling factors of 2× and 4× on the DIV2K dataset [60]. The best performances are highlighted for clarity. Notably, despite the lightweight complexity of the proposed model, it achieves super-resolution performance comparable to that of more complex architectures, particularly at a scaling factor of 4×. The PSNR and SSIM values for ADSCPN-plus with a patch size of 192 are 25.65 dB and 0.6940, respectively, which are slightly superior to the results of ADRBN under the same training conditions, showing a difference of 0.01 dB and 0.0004. Additionally, Table 7 provides the quantitative results for the model trained on the DIV2K dataset [71] and tested across various scaling factors on the DRealSR [61] real-world scenario test set. Furthermore, Table 8 displays the quantitative evaluation results across different scaling factors following retraining and testing on the RealSR dataset [62].

Table 5. Quantitative comparison of methods with scale factor 4.

Method	Scale Factor	Set5	Set14	B100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	×4	28.42/0.8104	26.00/0.7027	25.96/0.6675	23.14/0.6577
SRCNN	×4	30.48/0.8628	27.50/0.7513	26.90/0.7101	24.52/0.7221
FSRCNN	×4	30.72/0.8660	27.61/0.7550	26.98/0.7150	24.62/0.7280
VDSR	×4	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524
LapSRN	×4	31.54/0.8852	28.09/0.7700	27.32/0.7275	25.21/0.7562
DRCN	×4	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510
DRRN	×4	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
IDN	×4	31.82/0.8903	28.25/0.7730	27.41/0.7297	25.41/0.7632
CARN-M	×4	31.92/0.8903	28.42/0.7762	27.44/0.7304	25.62/0.7694
MemNet	×4	31.74/0.8893	28.26/0.7723	27.40/0.7281	25.50/0.7630
MADNet	×4	31.93/0.8917	28.44/0.7780	27.47/0.7327	25.76/0.7746
ECBSR	×4	31.92/0.8946	28.34/0.7817	27.48/0.7393	25.53/0.7773
LINF	×4	31.70/—	27.54/—	26.62/—	25.15/—
FPL	×4	30.39/0.8805	26.91/0.7688	26.29/0.7393	24.78/0.7880
ADSCPN	×4	32.04/0.8929	28.50/0.7793	27.52/0.7339	25.95/0.7812
ADSCPN-plus	×4	32.12/0.8940	28.49/0.7794	27.53/0.7345	25.98/0.7823

Table 6. Quantitative comparison of methods on the DIV2K test set.

Method	×2	×4
	PSNR/SSIM	PSNR/SSIM
Bicubic	28.69/0.8058	25.38/0.6822
ADSCPN-plus	29.19/0.8227	25.65/0.6940
EDSR(192)	29.21/0.8234	25.66/0.6945
ADRBN-plus	29.21/0.8233	25.66/0.6944

Table 7. Quantitative results on the DRealSR test set.

Method	×2	×3	×4
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	32.67/0.9060	31.51/0.8700	30.56/0.8595
ADSCPN-plus	32.82/0.9089	31.60/0.8727	30.62/0.8611

Table 8. Quantitative comparison of methods on the RealSR test set.

Method	×2	×3	×4
	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Bicubic	31.67/0.8865	28.63/0.8085	27.23/0.7637
ADSCPN-plus	34.01/0.9225	30.88/0.8465	29.27/0.8255

4.4.3. Qualitative Results Comparison

To evaluate the visual quality of reconstructed images generated by different models, localized regions from the test set results were enlarged to facilitate direct visual inspection of the image detail reconstruction. The comparison methods include Bicubic [46], LapSRN [65], CARN-M [35], IDN [36], and the proposed ADSCPN-plus model, with visual results sourced from official publications for each method.

As illustrated in Figure 9, for the 2× scaling factor, the region highlighted within the red box of the high-resolution image from the Urban100 test set (“image092”) reveals processed horizontal lines. In contrast, the areas reconstructed by the three interpolation methods exhibit significant blurring, with LapSRN [65] displaying the most distorted slanted lines. In comparison, the proposed model achieves superior visual and quantitative

results. For the $3\times$ scaling factor, an enlarged region from the high-resolution image in the B100 test set (“img063”) demonstrates that the model effectively retains the brightness in the eye region, exhibiting sharper edges compared to the high-resolution images. At a $4\times$ scaling factor, the enlarged image region in the Urban100 test set (“image044”) reveals that the building edges are processed more accurately, resembling high-resolution images, while other methods produce blurred edges. The proposed model successfully captures texture patterns closer to those of the high-resolution images, underscoring its effectiveness in handling complex scenes. Figure 10 presents the results from the DIV2K/RRK, DRealSR, and RealSR datasets, further demonstrating the model’s ability to generalize across different datasets and maintain high-quality reconstructions.

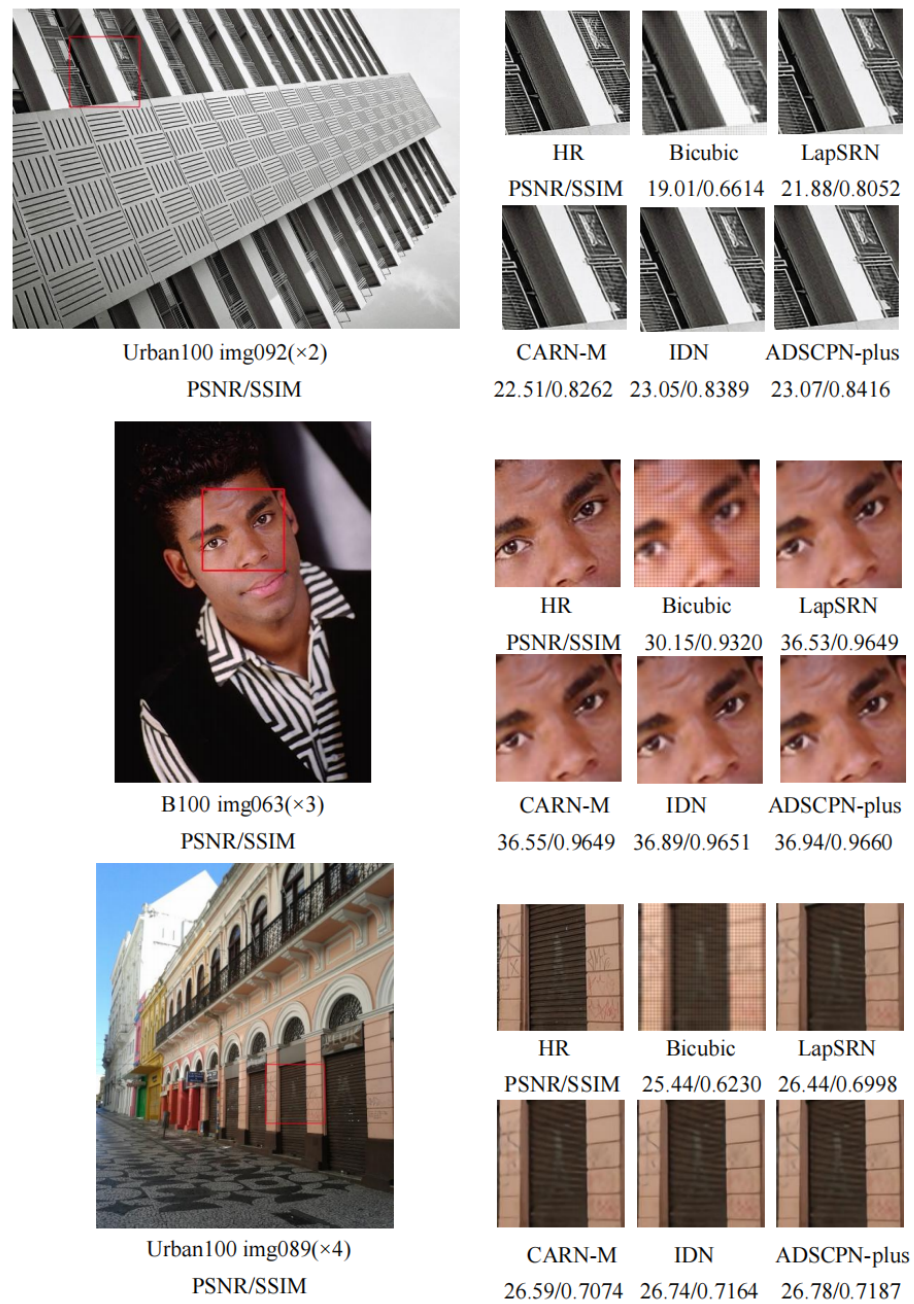


Figure 9. Visual comparison of the results for different methods.

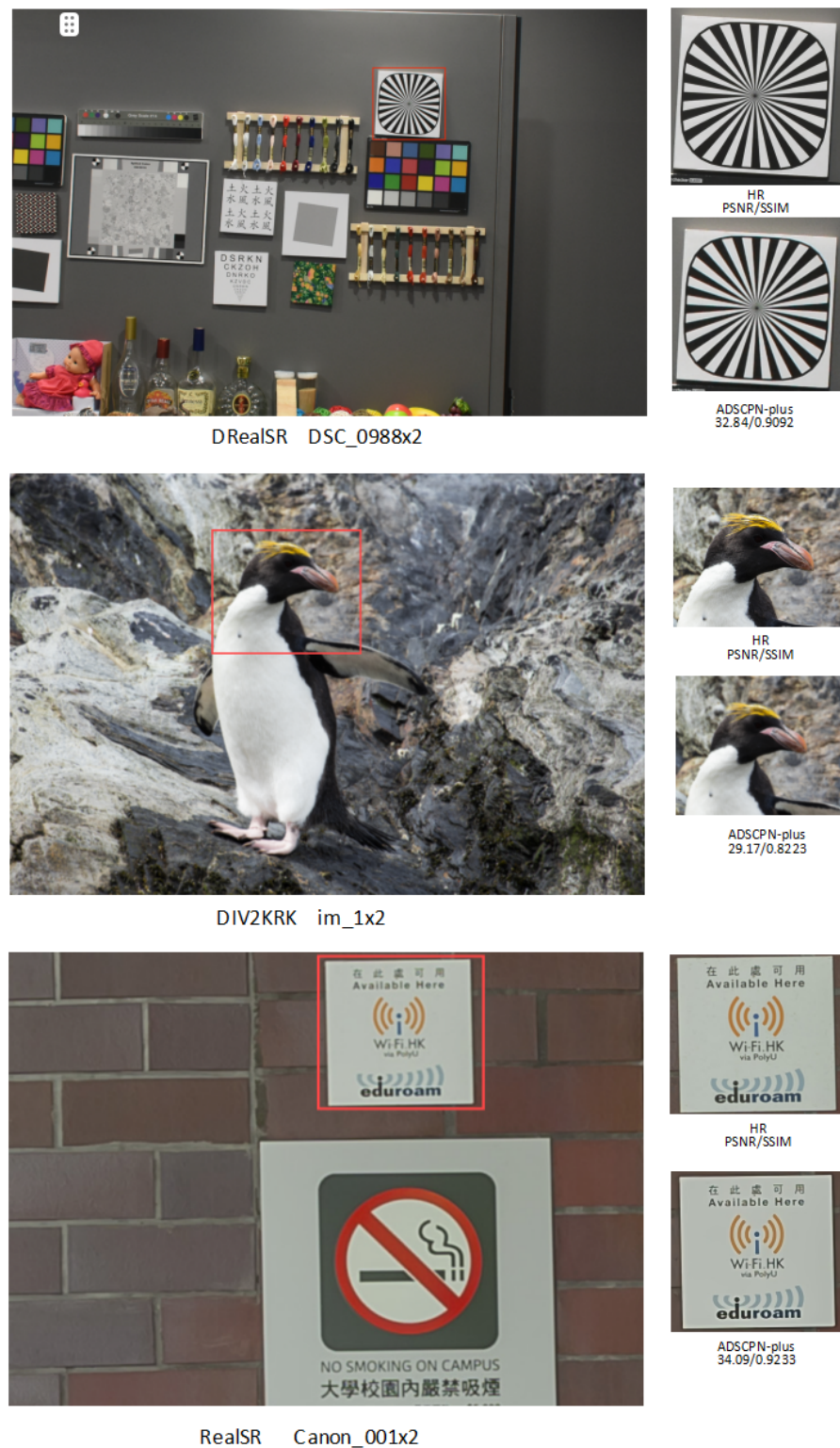


Figure 10. Visual comparison of different super-resolution methods on the DRealSR, DIV2KRR, and RealSR datasets.

4.4.4. Model Complexity

In this section, we analyze the model complexity of ADSCPN in terms of FLOPs (Floating Point Operations) and MACs (Multiply-Accumulate Operations). FLOPs represent the total number of floating-point calculations needed by the model, which directly reflects the

computational power required. MACs are a measure of the model’s efficiency, representing the number of times two numbers are multiplied and added, a critical metric for evaluating computational load in deep learning models. These metrics are particularly important for understanding the model’s suitability for resource-constrained environments.

When the scaling factor is set to 2 and the input image size is $3 \times 1280 \times 720$, the model’s complexity was calculated using the ‘thop’ library and compared against several established models, including SRCNN [27], VDSR [34], LapSRN [65], DRCN [28], DRRN [29], MemNet [64], CARN [35], MADNet [66], and FALSR-A [68]. The comparison focuses on model parameters (Flops) and multiply-accumulate operations (MACs), which estimate the memory requirements and computational costs of the models. As demonstrated in Table 9, the proposed model exhibits significantly fewer parameters compared to MADNet, FALSR-A, and CARN, while maintaining a comparable level of performance. Although the model prioritizes convolution operations and consequently incurs a higher number of multiply-accumulate operations, the overall computational load remains relatively manageable for hardware devices. This characteristic renders the model well-suited for real-time applications on embedded mobile devices.

Table 9. Comparison of model parameters and MACs.

Method	Parameters (K)	MACs (G)
SRCNN	57	53
VDSR	665	613
LapSRN	813	30
DRCN	1774	17,974
DRRN	297	6797
MemNet	677	2662
CARN	1592	223
MADNet	878	187
FALSR-A	1021	235
Ours	674	570

4.5. Ablation Analysis

To substantiate the rationale behind the design of each component within the feature processing block and to assess the effectiveness of these components in achieving high-quality reconstruction, we will conduct ablation experiments. These experiments will validate each component individually and investigate the impact of their combined presence on quantitative performance.

4.5.1. Effectiveness of Large-Kernel Parallel Convolution Groups

The large-kernel parallel depthwise separable convolution group plays a crucial role in enhancing the receptive field without significantly increasing the computational complexity. It is composed of a 1×7 depthwise separable convolution, a 1×1 group convolution, and a 7×1 depthwise separable convolution, which together enable efficient spatial feature learning. Specifically, the 1×7 and 7×1 convolutions separately learn features along the horizontal and vertical axes, which enhances the 1×1 convolution’s ability to integrate these directional features effectively.

Compared to traditional 3×3 convolutions often employed in lightweight models, the large-kernel parallel structure significantly expands the receptive field, enabling the model to capture more global contextual information while maintaining a lightweight architecture. This structure provides a richer and more detailed spatial feature representation, leading to improved feature extraction capability.

Unlike a single 7×7 depthwise separable convolution, which may introduce more computational burden and inefficiencies, the large-kernel parallel group processes the input using separate directional convolutions. This parallel approach not only maintains computational efficiency but also improves the model’s ability to capture intricate details, which is reflected in the enhanced PSNR and SSIM values seen across multiple test datasets.

The experimental results, summarized in Table 10, demonstrate that the large-kernel parallel convolution group offers superior performance in comparison to using a single 7×7 convolution, especially for the Set5 dataset. This effectiveness can be attributed to the improved ability to capture directional features and integrate them efficiently. Table 11 further highlights that this convolution strategy achieves a better trade-off between model complexity and computational cost, which is crucial for achieving high-quality super-resolution while ensuring the model remains practical for deployment.

Table 10. Comparison of replacing large kernel parallel depthwise convolution with 7×7 depthwise separable convolution.

Convolution Structure	Scale Factor	Set5	Set14	B100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Large Kernel Parallel Depthwise Separable Conv Group	$\times 2$	37.90/0.9602	33.51/0.9174	32.11/0.8990	31.93/0.9262
7×7 Depthwise Separable Conv	$\times 2$	37.92/0.9603	33.43/0.9167	31.10/0.8988	31.86/0.9257

Table 11. Comparison of model complexity between large-kernel parallel depthwise separable convolution groups and a single 7×7 depthwise separable convolution.

Convolution Structure	Parameters (K)	MACs (G)
Large-Kernel Parallel Depthwise Separable Conv. Group	674	570.07
7×7 Depthwise Separable Convolution	732	624.22

4.5.2. Effectiveness of Dynamic Convolution

After applying the large-kernel parallel depthwise separable convolution group, dynamic convolution is used to further enhance the model's ability to extract meaningful features. Dynamic convolution introduces an adaptive mechanism where the convolutional filters are adjusted based on the importance of the features, as determined by an embedded attention mechanism. This attention-driven adaptation ensures that critical regions of the input receive more focus during processing, thereby significantly enhancing the model's capacity to represent and learn from complex image features.

In contrast to standard convolution, which applies static filters across all regions of the input, dynamic convolution provides a context-dependent filter adaptation that aligns with the content's specific characteristics. This capability not only enhances feature extraction but also allows the model to better capture the nuances in more challenging datasets, such as Urban100, which contain complex structures and textures.

For this study, the dynamic convolution used a kernel size of 3×3 . We conducted an ablation experiment by replacing dynamic convolution with standard 3×3 convolution to evaluate its impact. As shown in Table 12, dynamic convolution consistently outperformed the standard convolution, particularly for a scale factor of 2, across all test datasets. Notably, the improvement was most significant on the Urban100 dataset, indicating the model's enhanced ability to generalize to complex, real-world scenes.

Table 12. Quantitative comparison of replacing dynamic convolution with 3×3 standard convolution.

Convolution Structure	Scale Factor	Set5	Set14	B100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Dynamic Convolution	$\times 2$	37.90/0.9602	33.51/0.9174	32.11/0.8990	31.93/0.9262
3×3 Conv	$\times 2$	37.88/0.9601	33.46/0.9167	32.10/0.8988	31.82/0.9253

The embedded attention mechanism within dynamic convolution plays a key role in this performance gain by dynamically highlighting the most informative features while suppressing less important ones. This selective focus directly contributes to improved PSNR and SSIM values, thereby enhancing overall reconstruction quality. The adaptive nature of dynamic convolution, driven by attention, makes it particularly effective for scenarios where image content varies widely in terms of texture and detail complexity. This explains its superior performance compared to standard convolution, as evidenced by the experimental results.

4.5.3. Effectiveness of the Point Attention Convolution Group

The point attention convolution group introduces an attention mechanism aimed at enhancing the model's ability to focus on the most critical features of an image, which significantly impacts overall performance. It improves upon the traditional convolution group—typically consisting of two 3×3 convolution layers with ReLU activation—by adding an additional branch consisting of a 1×1 convolution followed by a Sigmoid activation. This branch computes attention coefficients for each feature channel, which are then used to recalibrate the input features dynamically.

The recalibration process ensures that more informative features are amplified while less significant features are suppressed, allowing the model to allocate its resources more effectively toward reconstructing key image details. This type of attention mechanism enables better feature calibration, particularly in deeper layers where important image features might otherwise become diluted or distorted.

In our experiments, replacing the point attention convolution group with a traditional convolution group resulted in a noticeable decrease in both PSNR and SSIM values across all test sets, as shown in Table 13. The most significant performance gains were observed in datasets with high structural complexity, such as Urban100, demonstrating the point attention group's effectiveness in capturing intricate textures and structures.

Table 13. Comparison between point attention convolution group and standard convolution group.

Convolution Structure	Scale Factor	Set5	Set14	B100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
Point Attention Convolution Group	$\times 2$	37.90/0.9602	33.51/0.9174	32.11/0.8990	31.93/0.9262
Standard Convolution Group	$\times 2$	37.90/0.9601	33.46/0.9169	32.09/0.8986	31.86/0.9256

By dynamically computing and applying attention coefficients, the point attention convolution group helps the model selectively focus on salient regions of the image, thereby improving the quality of feature representation. This selective focus, facilitated by the embedded attention mechanism, directly leads to better feature extraction, less feature distortion, and improved reconstruction quality. The attention-driven recalibration of features also ensures that the network learns to prioritize the most critical information, which is crucial for achieving higher PSNR and SSIM metrics.

4.5.4. Effectiveness of the Efficient Enhanced Spatial Attention Block

The Efficient Enhanced Spatial Attention (EESA) block is a crucial component designed to improve the model's ability to focus on spatially important regions while maintaining computational efficiency. The EESA block is an optimized version of the Enhanced Spatial Attention (ESA) block, achieved by removing the external 1×1 convolution and simplifying the convolution group to a single 3×3 convolution. This optimization reduces both model complexity and computational burden, making it more suitable for lightweight deployment without sacrificing performance.

The core mechanism of the EESA block is its ability to dynamically highlight spatial regions that are most important for the reconstruction task. By learning spatial attention maps that emphasize high-frequency details such as edges and textures, the EESA block directs the model's focus to areas with significant structural information. This selective

focus is essential for achieving high-quality reconstruction, particularly in regions with complex textures or fine details.

In our ablation studies, we compared the EESA block to the original ESA block to evaluate its effectiveness. As shown in Table 14, the EESA block not only reduces the number of parameters and multiply-accumulate (MAC) operations but also maintains comparable or better performance in terms of PSNR and SSIM across multiple test datasets. This improvement is largely due to the effective attention mechanism that allows the model to prioritize spatial regions that contribute the most to image quality.

Table 14. Comparison of model complexity between enhanced spatial attention block and efficient enhanced spatial attention block.

Spatial Attention Block	Parameters (K)	MACs (G)
Enhanced Spatial Attention Block	740	574.80
Efficient Enhanced Spatial Attention Block	674	570.07

The simplified architecture of the EESA block enhances computational efficiency while ensuring that the model retains the ability to focus on crucial spatial features. This efficiency is particularly beneficial for real-world applications where computational resources are limited. The attention mechanism within the EESA block plays a key role in guiding the model to learn meaningful spatial relationships, thereby enhancing the quality of feature extraction and overall image reconstruction.

4.5.5. Effectiveness of Combined Components

To assess the impact of the combined components within the feature processing block on the model's quantitative performance and to validate the rationale and superiority of the overall structure presented in this section, we conducted an ablation study focusing on the feature extraction block and the point attention convolution group. Following the previous ablation experiments that modified individual components, we replaced the large-kernel parallel depthwise separable convolution group in the feature extraction block with a single 7×7 large-kernel depthwise separable convolution, substituted dynamic convolution with a standard 3×3 convolution, and replaced the point attention convolution group with a standard convolution group.

The specific combinations of components are detailed in Table 15, where “✓” indicates the inclusion of the component, and “×” signifies replacement with a standard component. Optimal performance is highlighted in bold. As shown, except for the Set5 test set, the model proposed in this section achieves the best quantitative performance overall, with PSNR and SSIM values on Set14 and Urban100 improving by 0.05 dB and 0.0005, respectively, compared to the second-best performance, and by 0.01 dB and 0.0001 on B100. Notably, when all components were replaced with standard components, the performance on the Urban100 dataset was superior to configurations where only one or two components were replaced with standard components. This finding demonstrates that the feature processing block can effectively learn image features at multiple levels and that the designed combination of the three components is both indispensable and complementary, resulting in robust reconstruction performance with reduced model complexity.

Table 15. Quantitative ablation study of different model component combinations.

Large Kernel Parallel Depthwise Separable Conv Group	Dynamic Convolution	Point Attention Convolution Group	Set5	Set14	B100	Urban100
			PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
×	×	×	37.93/0.9602	33.44/0.9166	32.11/0.8989	31.88/0.9257
×	×	✓	37.88/0.9601	33.43/0.9159	32.10/0.8988	31.84/0.9257
×	✓	×	37.88/0.9602	33.46/0.9168	32.10/0.8986	31.82/0.9254
✓	×	×	37.88/0.9601	33.46/0.9167	32.09/0.8988	31.80/0.9253
×	✓	✓	37.92/ 0.9603	33.43/0.9167	31.86/0.9257	31.88/0.9258
✓	×	✓	37.87/0.9601	33.39/0.9161	32.10/0.8986	31.82/0.9252
✓	✓	×	37.90/0.9601	33.46/0.9169	32.11/0.8986	31.83/0.9256
✓	✓	✓	37.90/0.9602	33.51/0.9174	32.11/0.8990	31.93/0.9262

4.5.6. Effectiveness of Activation Functions

Incorporating appropriate activation functions can significantly enhance the nonlinear feature learning capabilities of the model, thereby improving the fidelity of image reconstruction. This section evaluates the performance of three activation functions: SiLU, GELU, and ReLU. With the exception of the Point Attention Convolution Group and the Efficient Enhanced Spatial Attention Block, all other components of the ADSCPN were subjected to an ablation study to compare the effects of these different activation functions. The quantitative results are summarized in Table 16, which identifies the optimal activation function. Notably, when the scaling factor is set to 2, the GELU activation function substantially outperforms the other options on the Set14 dataset, while exhibiting comparable performance across the remaining datasets. Consequently, GELU is selected as the activation function for the entire model in this study.

Table 16. Ablation study on activation functions.

Activation Function	Scaling Factor	Set5	Set14	B100	Urban100
		PSNR/SSIM	PSNR/SSIM	PSNR/SSIM	PSNR/SSIM
SiLU	×2	37.91/0.9602	33.45/0.9168	32.10/0.8989	31.88/0.9259
GELU	×2	37.90/0.9602	33.51/0.9174	32.11/0.8990	31.93/0.9262
ReLU	×2	37.91/0.9602	33.42/0.9167	32.12/0.8991	31.94/0.9263

5. Conclusions

Through comprehensive research and analysis in the realm of lightweight image super-resolution, we propose a novel lightweight super-resolution network founded on a large-kernel parallel convolution group. The network architecture employs a multi-layer point-wise perceptron for channel projection and utilizes shuffle attention for effective channel grouping, thereby enhancing the extraction of deep channel and spatial information while improving feature learning efficiency. To facilitate deeper feature learning, we introduce a large-kernel parallel depthwise separable convolution group alongside grouped convolution. Additionally, the point attention convolution group and efficient enhanced spatial attention block are integrated to optimize spatial feature learning, supplement low-frequency information, and enhance deep feature extraction. This design approach mitigates the distortion of deep features often encountered in deeper networks, significantly improving the quality of super-resolution images while maintaining low model complexity. Both quantitative and qualitative experiments, in comparison with existing methods, demonstrate the superior performance of our proposed approach, while extensive ablation studies further validate the rationale behind the component configurations.

Despite its promising performance, the ADSCPN model has some limitations, including sensitivity to noisy and low-quality inputs, high computational complexity, and limited generalization in diverse scenarios. These limitations may hinder its deployment in resource-constrained environments and challenging real-world applications. Future work will focus on enhancing robustness to noise, optimizing computational efficiency for edge devices, and expanding training with more diverse datasets to improve generalization. Such improvements will make ADSCPN more viable for practical applications like mobile photography, medical imaging, surveillance, and specialized domains.

Author Contributions: Y.L. (Yiting Long) and H.Z. designed the methodology, wrote the original draft, and designed and prepared all figures. H.R. contributed to conceptualization and project administration. C.Z. and L.Z. acquired funding and reviewed and edited the manuscript. Y.L. (Yiting Long) and H.R. conceived the experiments. Y.L. (Yiting Long), H.R. and H.Z. conducted the experiments and acquired the experimental results. Y.L. (Yi Liu), L.Z., and X.Z. analyzed the experimental results. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 62202163, 62072166 and 62372150, the Natural Science Foundation of Hunan Province

under Grant 2022JJ40190, 2022JJ30231 and 2023JJ30169, the Scientific Research Project of Hunan Provincial Department of Education under Grant 22A0145 and 22B0559.

Data Availability Statement: The public datasets used in this paper can be accessed through the following links: DIV2K (<https://data.vision.ee.ethz.ch/cv1/DIV2K/>), Set5 (<https://uofi.box.com/shared/static/kfahv87nfe8ax910l85dksyl2q212voc.zip>), Set14 (<https://uofi.box.com/shared/static/igsnfieh4lz68l926l8xbklwsnkn8we9.zip>), B100 (<https://uofi.box.com/shared/static/qgctsplb8txrksm9to9x01zfa4m61ngq.zip>), Urban100 (<https://uofi.box.com/shared/static/>), DIV2KRRK (https://www.wisdom.weizmann.ac.il/~vision/kernelgan/DIV2KRRK_public.zip), DRealSR (https://drive.google.com/drive/folders/1tP5m4k1_shFT6Dcw31XV8cWHtblGmbOk?usp=sharing), and RealSR (https://drive.google.com/open?id=1gKnm9BdgyqISCTDAbGbpVit-QII_unw), all accessed on 18 September 2024.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

SR	Super-resolution
HR	High-resolution
LR	Low-resolution
ADSCPN	Adaptive dynamic shuffle convolutional parallel network
CNNs	Convolutional neural networks
LD	Feature processing block
CPB	Channel processing block
FEB	Feature extraction block
PW-MLP	Point-wise multi-layer perceptron
PACG	Point-attention convolution group
EESA	Efficiently enhanced spatial attention
ESA	Enhanced spatial attention
PSNR	Peak signal-to-noise ratio
SSIM	Structural similarity index

References

- Chen, Z.; Zhang, Y.; Gu, J.; Kong, L.; Yang, X.; Yu, F. Dual aggregation transformer for image super-resolution. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 12312–12321.
- Chen, H.; He, X.; Qing, L.; Wu, Y.; Ren, C.; Sheriff, R.E.; Zhu, C. Real-world single image super-resolution: A brief review. *Inf. Fusion* **2022**, *79*, 124–145. [\[CrossRef\]](#)
- Zhang, Q.; Xiao, J.; Tian, C.; Chun-Wei Lin, J.; Zhang, S. A robust deformed convolutional neural network (CNN) for image denoising. *CAAI Trans. Intell. Technol.* **2023**, *8*, 331–342. [\[CrossRef\]](#)
- Wang, C.; Lv, X.; Shao, M.; Qian, Y.; Zhang, Y. A novel fuzzy hierarchical fusion attention convolution neural network for medical image super-resolution reconstruction. *Inf. Sci.* **2023**, *622*, 424–436. [\[CrossRef\]](#)
- Georgescu, M.I.; Ionescu, R.T.; Miron, A.I.; Savencu, O.; Ristea, N.C.; Verga, N.; Khan, F.S. Multimodal multi-head convolutional attention with various kernel sizes for medical image super-resolution. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 3–7 January 2023; pp. 2195–2205.
- Cornebise, J.; Oršolić, I.; Kalaitzis, F. Open high-resolution satellite imagery: The worldstrat dataset—with application to super-resolution. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 25979–25991.
- Arefin, M.R.; Michalski, V.; St-Charles, P.L.; Kalaitzis, A.; Kim, S.; Kahou, S.E.; Bengio, Y. Multi-image super-resolution for remote sensing using deep recurrent networks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Seattle, WA, USA, 14–19 June 2020; pp. 206–207.
- Zhu, L.; Yu, W.; Zhu, X.; Zhang, C.; Li, Y.; Zhang, S. MvHAAN: Multi-view hierarchical attention adversarial network for person re-identification. *World Wide Web* **2024**, *27*, 59. [\[CrossRef\]](#)
- Zhang, C.; Xie, F.; Yu, H.; Zhang, J.; Zhu, L.; Li, Y. PPIS-JOIN: A novel privacy-preserving image similarity join method. *Neural Process. Lett.* **2022**, *54*, 2783–2801. [\[CrossRef\]](#)
- Zheng, M.; Xu, J.; Shen, Y.; Tian, C.; Li, J.; Fei, L.; Zong, M.; Liu, X. Attention-based CNNs for image classification: A survey. In *Proceedings of the Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2022; Volume 2171, p. 012068.
- Li, Q.; Chen, Y.; Zhang, A.; Jiang, Y.; Zou, L.; Xu, Z.; Muntean, G.M. A super-resolution flexible video coding solution for improving live streaming quality. *IEEE Trans. Multimed.* **2022**, *25*, 6341–6355. [\[CrossRef\]](#)

12. Huang, K.; Tian, C.; Xu, Z.; Li, N.; Lin, J.C.W. Motion Context guided Edge-preserving network for video salient object detection. *Expert Syst. Appl.* **2023**, *233*, 120739. [[CrossRef](#)]
13. Zhang, Y.; Zhang, Y.; Wu, Y.; Tao, Y.; Bian, K.; Zhou, P.; Song, L.; Tuo, H. Improving quality of experience by adaptive video streaming with super-resolution. In Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications, Virtual, 6–9 July 2020; pp. 1957–1966.
14. Qiao, C.; Li, D.; Guo, Y.; Liu, C.; Jiang, T.; Dai, Q.; Li, D. Evaluation and development of deep neural networks for image super-resolution in optical microscopy. *Nat. Methods* **2021**, *18*, 194–202. [[CrossRef](#)]
15. Gendy, G.; He, G.; Sabor, N. Lightweight image super-resolution based on deep learning: State-of-the-art and future directions. *Inf. Fusion* **2023**, *94*, 284–310. [[CrossRef](#)]
16. Tian, C.; Fei, L.; Zheng, W.; Xu, Y.; Zuo, W.; Lin, C.W. Deep learning on image denoising: An overview. *Neural Netw.* **2020**, *131*, 251–275. [[CrossRef](#)] [[PubMed](#)]
17. Tian, C.; Xu, Y.; Fei, L.; Yan, K. Deep learning for image denoising: A survey. In Proceedings of the Genetic and Evolutionary Computing: Proceedings of the Twelfth International Conference on Genetic and Evolutionary Computing, Changzhou, China, 14–17 December 2018; Springer: Berlin/Heidelberg, Germany, 2019; pp. 563–572.
18. Zhu, L.; Zhang, C.; Song, J.; Liu, L.; Zhang, S.; Li, Y. Multi-graph based hierarchical semantic fusion for cross-modal representation. In Proceedings of the 2021 IEEE International Conference on Multimedia and Expo (ICME), Virtual, 5–9 July 2021; pp. 1–6.
19. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
20. Tian, C.; Xu, Y.; Li, Z.; Zuo, W.; Fei, L.; Liu, H. Attention-guided CNN for image denoising. *Neural Netw.* **2020**, *124*, 117–129. [[CrossRef](#)] [[PubMed](#)]
21. Zhu, L.; Song, J.; Wei, X.; Yu, H.; Long, J. CAESAR: Concept augmentation based semantic representation for cross-modal retrieval. *Multimed. Tools Appl.* **2022**, *81*, 34213–34243. [[CrossRef](#)]
22. Tian, C.; Xu, Y.; Zuo, W. Image denoising using deep CNN with batch renormalization. *Neural Netw.* **2020**, *121*, 461–473. [[CrossRef](#)]
23. Tian, C.; Yuan, Y.; Zhang, S.; Lin, C.W.; Zuo, W.; Zhang, D. Image super-resolution with an enhanced group convolutional neural network. *Neural Netw.* **2022**, *153*, 373–385. [[CrossRef](#)]
24. Wang, Z.; Chen, J.; Hoi, S.C. Deep learning for image super-resolution: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3365–3387. [[CrossRef](#)] [[PubMed](#)]
25. Tian, C.; Zhang, X.; Lin, J.C.W.; Zuo, W.; Zhang, Y.; Lin, C.W. Generative adversarial networks for image super-resolution: A survey. *arXiv* **2022**, arXiv:2204.13620.
26. Dong, C.; Loy, C.C.; Tang, X. Accelerating the super-resolution convolutional neural network. In Proceedings of the Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, 11–14 October 2016; Proceedings, Part II 14; Springer: Berlin/Heidelberg, Germany, 2016; pp. 391–407.
27. 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](#)]
28. Kim, J.; Lee, J.K.; Lee, K.M. Deeply-recursive convolutional network for image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1637–1645.
29. Tai, Y.; Yang, J.; Liu, X. Image super-resolution via deep recursive residual network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3147–3155.
30. Lim, B.; Son, S.; Kim, H.; Nah, S.; Mu Lee, K. Enhanced deep residual networks for single image super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2017; pp. 136–144.
31. 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, Honolulu, HI, USA, 21–26 July 2017; Volume 2017, pp. 4681–4690.
32. Tian, C.; Zhang, X.; Zhang, Q.; Yang, M.; Ju, Z. Image super-resolution via dynamic network. *CAAI Trans. Intell. Technol.* **2024**, *9*, 837–849. [[CrossRef](#)]
33. Tian, C.; Zhang, X.; Ren, J.; Zuo, W.; Zhang, Y.; Lin, C.W. A Heterogeneous Dynamic Convolutional Neural Network for Image Super-resolution. *arXiv* **2024**, arXiv:2402.15704.
34. 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, Las Vegas, NV, USA, 27–30 June 2016; pp. 1646–1654.
35. Ahn, N.; Kang, B.; Sohn, K.A. 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; pp. 252–268.
36. 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; pp. 723–731.
37. 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; pp. 2024–2032.

38. Liu, J.; Tang, J.; Wu, G. Residual feature distillation network for lightweight image super-resolution. In Proceedings of the Computer Vision–ECCV 2020 Workshops, Glasgow, UK, 23–28 August 2020; Proceedings, Part III 16; Springer: Berlin/Heidelberg, Germany, 2020; pp. 41–55.
39. Kong, F.; Li, M.; Liu, S.; Liu, D.; He, J.; Bai, Y.; Chen, F.; Fu, L. Residual local feature network for efficient super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 19–24 June 2022; pp. 766–776.
40. Zhang, A.; Ren, W.; Liu, Y.; Cao, X. Lightweight image super-resolution with superpixel token interaction. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–6 October 2023; pp. 12728–12737.
41. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [[CrossRef](#)]
42. Sun, L.; Pan, J.; Tang, J. Shufflemixer: An efficient convnet for image super-resolution. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 17314–17326.
43. Tian, C.; Zhuge, R.; Wu, Z.; Xu, Y.; Zuo, W.; Chen, C.; Lin, C.W. Lightweight image super-resolution with enhanced CNN. *Knowl. Based Syst.* **2020**, *205*, 106235. [[CrossRef](#)]
44. Conde, M.V.; Choi, U.J.; Burchi, M.; Timofte, R. Swin2sr: Swinv2 transformer for compressed image super-resolution and restoration. In Proceedings of the European Conference on Computer Vision (ECCV), Tel Aviv, Israel, 23–27 October 2022; pp. 669–687.
45. Hou, H.; Andrews, H. Cubic splines for image interpolation and digital filtering. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 508–517.
46. Keys, R. Cubic convolution interpolation for digital image processing. *IEEE Trans. Acoust. Speech Signal Process.* **1981**, *29*, 1153–1160. [[CrossRef](#)]
47. Li, J.; Wu, J.; Deng, H.; Liu, J. A self-learning image super-resolution method via sparse representation and non-local similarity. *Neurocomputing* **2016**, *184*, 196–206. [[CrossRef](#)]
48. Howard, A.G. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
49. Ma, N.; Zhang, X.; Zheng, H.T.; Sun, J. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 116–131.
50. Hu, J.; Shen, L.; Sun, G. Squeeze-and-excitation networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
51. Hu, Y.; Li, J.; Huang, Y.; Gao, X. Channel-wise and spatial feature modulation network for single image super-resolution. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 3911–3927. [[CrossRef](#)]
52. 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 (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3867–3876.
53. Tian, C.; Zhang, Y.; Zuo, W.; Lin, C.W.; Zhang, D.; Yuan, Y. A heterogeneous group CNN for image super-resolution. In Proceedings of the IEEE Transactions on Neural Networks and Learning Systems, Virtual Event, 23–28 May 2022.
54. Luo, Z.; Huang, H.; Yu, L.; Li, Y.; Fan, H.; Liu, S. Deep constrained least squares for blind image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 21–24 June 2022; pp. 17642–17652.
55. Ma, C.; Yang, C.Y.; Yang, X.; Yang, M.H. Learning a no-reference quality metric for single-image super-resolution. *Comput. Vis. Image Underst.* **2017**, *158*, 1–16. [[CrossRef](#)]
56. Bevilacqua, M.; Roumy, A.; Guillemot, C.; Alberi-Morel, M.L. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In Proceedings of the 23rd British Machine Vision Conference (BMVC), Surrey, UK, 3–7 September 2012; BMVA Press: Durham, UK, 2012; pp. 135.1–135.10, ISBN 1-901725-46-4. [[CrossRef](#)]
57. Zeyde, R.; Elad, M.; Protter, M. On single image scale-up using sparse-representations. In Proceedings of the Curves and Surfaces: 7th International Conference, Avignon, France, 24–30 June 2010; Revised Selected Papers 7; Springer: Berlin/Heidelberg, Germany, 2012; pp. 711–730.
58. Martin, D.; Fowlkes, C.; Tal, D.; Malik, J. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In Proceedings of the Eighth IEEE international Conference on Computer Vision (ICCV), Vancouver, BC, Canada, 7–14 July 2001; Volume 2, pp. 416–423.
59. Huang, J.B.; Singh, A.; Ahuja, N. Single image super-resolution from transformed self-exemplars. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 5197–5206.
60. Chen, X.; Wang, X.; Zhou, J.; Qiao, Y.; Dong, C. Activating more pixels in image super-resolution transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–23 June 2023; pp. 22367–22377.
61. Huang, Y.; Li, S.; Wang, L.; Tan, T.; Luo, Z. Unfolding the alternating optimization for blind super resolution. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 5632–5643.
62. Fujimoto, A.; Ogawa, T.; Yamamoto, K.; Matsui, Y.; Yamasaki, T.; Aizawa, K. Manga109 dataset and creation of metadata. In Proceedings of the 1st International Workshop on Comics Analysis, Processing and Understanding, Cancun, Mexico, 4–7 December 2016; pp. 1–5.

63. Wang, Z.; Bovik, A.C.; Sheikh, H.R.; Simoncelli, E.P. Image quality assessment: From error visibility to structural similarity. *IEEE Trans. Image Process.* **2004**, *13*, 600–612. [[CrossRef](#)] [[PubMed](#)]
64. Tai, Y.; Yang, J.; Liu, X.; Xu, C. Memnet: A persistent memory network for image restoration. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017. pp. 4539–4547.
65. Lai, W.S.; Huang, J.B.; Ahuja, N.; Yang, M.H. Deep laplacian pyramid networks for fast and accurate super-resolution. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 624–632.
66. Liu, Z.; Mao, H.; Wu, C.Y.; Feichtenhofer, C.; Darrell, T.; Xie, S. A convnet for the 2020s. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 11976–11986.
67. Ding, X.; Zhang, X.; Han, J.; Ding, G. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 11963–11975.
68. Chu, X.; Zhang, B.; Ma, H.; Xu, R.; Li, Q. Fast, accurate and lightweight super-resolution with neural architecture search. In Proceedings of the 2020 25th International Conference on Pattern Recognition (ICPR), Milan, Italy, 10–15 January 2021; pp. 59–64.
69. Gou, Y.; Hu, P.; Lv, J.; Zhu, H.; Peng, X. Rethinking image super resolution from long-tailed distribution learning perspective. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 14327–14336.
70. Yao, J.E.; Tsao, L.Y.; Lo, Y.C.; Tseng, R.; Chang, C.C.; Lee, C.Y. Local implicit normalizing flow for arbitrary-scale image super-resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 1776–1785.
71. Agustsson, E.; Timofte, R.N. Challenge on single image super-resolution: Dataset and study. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Honolulu, HI, USA, 21–26 July 2016; pp. 21–26.

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.