Adaptive Branch Selection for Accelerate Image Super-Resolution

Cheng Ding[a], **Zhong-Qiu Zhao**[a,b,c,*], **Hao Shen**[d]
[a]School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230009, China
[b]Intelligent Interconnected Systems Laboratory of Anhui Province (Hefei University of Technology), China
[c]Guangxi Academy of Sciences, China
[d]School of Public Security and Emergency Management, Anhui University of Science and Technology, Hefei, 231131, Anhui, China

Abstract. In practical applications, image resolution has already reached 4K. However, large images often contain many smooth regions that can achieve good performance using networks with fewer channels. Existing methods for accelerating SR models typically divide images into multiple patches and process them through separate branches. However, these approaches suffer from two limitations:(1) The lack of scalability to be developed on platforms with different capacities. (2) The lack of interaction between multiple branches results in limited acceleration performance. Based on this, we propose Adaptive Branch Selection (ABS) for accelerating image super-resolution. ABS utilizes an efficient regressor to predict the performance increment between branches, dynamically selecting different branches for each patch by adjusting thresholds. To further enhance the acceleration performance, we introduce Progressive Mutual Information Knowledge Distillation (PMID) to help improve the SR performance of branches with fewer channels. Experimental results on the Test8K dataset show that FSRCNN-ABS achieves performance comparable to the original model while using only 49% of the FLOPs.

Keywords: super-resolution, regressor, knowledge distillation.

*Zhongqiu Zhao, z.zhao@hfut.edu.cn

1 Introduction

Single Image Super-Resolution(SISR) is an advanced technique that aims to generate a high-resolution image from a low-resolution input. The evolution of convolutional neural networks (CNNs) has spurred the development of numerous effective methods aimed at addressing this inherently ill-posed problem.[1] Existing methods[2,3] primarily enhance performance by increasing the number of network blocks,[4] refining the attention mechanism,[5] introducing the transformer architecture, and other approaches.[5–8] However, these small improvements double the computation cost of the network, which hinders real-world applications. Particularly within fields like surveillance and video transmission, image resolutions have already reached ultra-high-definition (UHD), *e.g.,* 4K (3840×2160). Taking the large image as input and reconstructing the intermediate features in the UHD space significantly increases both memory consumption and computational costs.

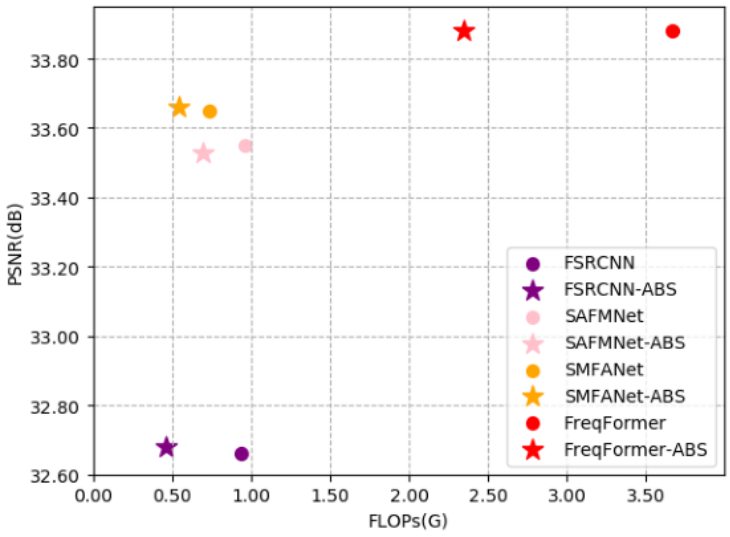


Fig. 1 PSNR results vs. the total FLOPs of different methods for single image SR(×4) on Test8K dataset.

In recent years, researchers have focused on proposing lightweight networks to enhance efficiency in image super-resolution. Some methods introduced spatially adaptive feature modulation mechanisms[9,10] and multi-scale feature cascading[7] to achieve efficient feature extraction. Other approaches concentrated on proposing new upsampling layers[11] to replace traditional deconvolution layers. In addition to special structural designs, some model compression strategies, such as knowledge distillation[12,13] and model quantization,[14] have also been applied in the field of image super-resolution. These methods offer effective strategies to balance performance and resource consumption, making them suitable for resource-limited devices. However, the aforementioned methods neglect to consider the sparsity in images. A significant portion of natural images consists of smooth regions that only require less processing to achieve good performance.

Following the introduction of RAISR,[15] researchers have started to employ various strategies[16] to handle different regions of images. Overall, the mainstream methods are primarily categorized into pixel-based methods and patch-based methods. The pixel-based method directly divides the

image into different pixel regions, which often have irregular shapes. However, since convolutional kernels are rectangular, they are not well-suited for convolution operations on such irregular regions, resulting in limited acceleration performance. The patch-based[17] method first divides the image into rectangular patches of the same size, and then uses different modules to process each patch separately. ClassSR[18] proposed a patch classification strategy to route different patches into different branches, where each branch had the same structure but with different channel numbers. Although this approach effectively improves the model's efficiency, the interaction between different branches is not considered. ARM[19] established a supernet that uses an edge-to-PSNR LUT (look-up table) to divide different patches into various subnets. However, LUT-based methods require a considerable amount of storage memory to store the look-up table. While the aforementioned methods reduce the model's FLOPs through various approaches, they are unscalable and fail to train one single network to adapt to devices with different resource constraints.

To address the above issues, we propose an Adaptive Branch Selection (ABS) for Accelerate Image Super-Resolution. From Fig. 1, we can observe that our ABS maintains comparable performance to the original model while reducing the model's FLOPs. Moreover, ABS achieves a greater reduction in FLOPs when applied to FreqFormer, which is a Transformer-based super-resolution method. Our ABS contains three branches, which have identical architectures but differ in the number of channels. Considering the interactions between each branch, we employ PMID (Progressive Mutual Information Knowledge Distillation) to improve the feature extraction capabilities of the branches with fewer channel numbers. After that, we train a lightweight regressor to predict the performance increments between different branches. During testing, the input LR image is first divided into multiple patches. By setting thresholds, various patches of a single image are distributed to different branches to save computational resources. Finally, all the patches output

from the branches will be merged into a single SR image.

1.1 Contributions

The main contributions of our paper are summarized as follows:

- We propose a novel SR accelerate framework that utilizes a lightweight regressor to distribute multiple patches into different branches for model efficiency.
- We adopt progressive mutual information knowledge distillation to enhance the performance of the branch with fewer channels in the ABS.
- We conduct quantitative and qualitative evaluations on multiple benchmark datasets, which demonstrate the superiority of our ABS.

2 Related Works

2.1 CNN-based Super Resolution Methods

With the development of deep learning, SRCNN[20] was the first to introduce a convolutional neural network for solving image super-resolution. Following this, VDSR[21] and DRCN[22] utilized residual connections to deepen the network, significantly improving performance by enabling the training of much deeper networks without suffering from vanishing gradient problems. By introducing attention mechanisms, RCAN[23] and SAN[24] effectively exploited the self-similarity within an image. In addition, many effective architectures have also been introduced into the field of image super-resolution. SRGAN[25] employed an adversarial loss to alternately train the GAN(Generative adversarial network), subsequently producing visually pleasing SR images. IPG[26] applied a GCNN(Graph Convolutional Neural Network) to dynamically aggregate similar regions, thereby enhancing the

interaction between different regions. However, these methods suffer from numerous parameters and high computational burden, which makes them less suitable for practical applications.

To reduce resource consumption, many methods concentrate on striking a balance between efficiency and performance. ESPCN[11] proposed the sub-pixel convolutional layer applied to real-time image super-resolution as an alternative to the deconvolution layer. SAFMNet[9] introduced an efficient spatial adaptive feature modulation mechanism for aggregating non-local features. SMFANet[10] utilized lightweight depth-wise separable convolutions to achieve channel and spatial adaptive feature extraction. Beyond focusing on structural design, several model compression strategies have also been utilized in image super-resolution. DFKD[12] adopted a progressive knowledge distillation strategy to enhance the feature extraction capability of lightweight networks. MTKD[13] selected multiple deep networks with diverse architectures as the teacher branch, enabling the lightweight SR model to generate more robust features. These approaches demonstrate the effectiveness of knowledge distillation in improving the performance of lightweight models. QuantSR[14] introduced a redistribution-driven learnable quantizer to achieve an accurate and efficient SR model. RefQSR[27] leveraged the self-similarity within a single image and designed a reference-based quantization module to save computational costs. However, the aforementioned methods failed to consider the sparsity within a single image. A significant portion of natural images consists of smooth regions that only require less processing to achieve good performance.

2.2 Transformer-based Super Resolution Methods

CNN-based SR methods are limited by the kernel size of convolutional operations, resulting in poorer global perception capabilities. Transformers were originally developed to address sequence modeling challenges in natural language processing, and have recently been widely adopted in im-

age classification and object detection. IPT[28] designed a multi-task Transformer that can simultaneously handle multiple upscaling scales, making it flexible for various image super-resolution tasks with different scaling requirements. SwinIR[29] utilized local attention and sliding cross-window interaction to extract global features, thereby significantly reducing redundant computations. By aggregating features across spatial and channel dimensions, DAT[30] effectively enhances SR performance. This method innovatively integrated contextual information from different parts of the image, leading to more accurate reconstruction and detail enhancement. HAT[31] combines both channel attention and self-attention to achieve further performance improvements. For practical applications, ESRT[32] proposed a lightweight transformer backbone to capture long-distance context dependence while reducing memory costs. SPIN[33] employs intra-superpixel attention to achieve efficient local information interaction. Transformer-based methods have significantly improved the performance of image super-resolution by leveraging their inherent ability to model long-range dependencies and capture global contextual information. Despite some approaches focusing on enhancing the efficiency of Transformer networks, they still bring about substantial computational resource consumption compared to CNN-based methods.

2.3 Region-aware Super Resolution Methods

In recent years, most research has begun to focus on region-aware image super-resolution. RAISR[15] divided image regions into different clusters and then designed multiple dedicated networks for each cluster, which significantly reduces the computational complexity of the model. SFTGAN[34] leveraged a segmentation probability map to modulate features, enabling the model to adaptively generate different texture details based on the content of each region. FADN[35] utilized Fourier Transform (FFT) to convert image features into the frequency domain and proposed a masking

strategy to separate high-frequency texture from low-frequency features. These features are then processed independently through different branch networks. These methods divide images into irregular regions based on each pixel, which makes them unsuitable for convolution operations, especially leading to blurring at the edges of the regions. Besides these methods, another approach is to divide the image into multiple rectangular patches. LAU-Net[36] employs reinforcement learning to implement a dynamic upscaling network that allows different patches to use distinct upscaling factors. ClassSR[18] uses a classifier and designs multiple loss functions to categorize simple patches into branches with fewer channels, thereby saving computational resources. ARM[19] trained a supernetwork to dynamically assign each patch to branches with varying numbers of feature channels. Meanwhile, it further reduces FLOPs by leveraging a look-up table. CAMixer[37] constructs a dual-branch network incorporating channel attention and self-attention, while balancing efficiency and performance by controlling the proportion of patches processed by each branch. Region-aware methods have been proven to effectively improve performance and reduce the FLOPs of the SR model. In our work, we propose a lightweight regressor that dynamically routes image patches into different branches of the SR-module to achieve region-aware super-resolution.

3 Proposed Method

In this section, we introduce our proposed ABS in detail. As shown in Fig. 2, our ABS consists of two parts: Regressor-Module and SR-Module. A large LR image is first divided into multiple rectangle patches, which are then fed into the regressor. The regressor outputs a two-dimensional vector based on the content of each patch. The SR-Module contains three branches with the same architecture but different numbers of channels. By using a manually set threshold, the patches are routed to the appropriate branch for processing. Finally, the outputs from different branches are

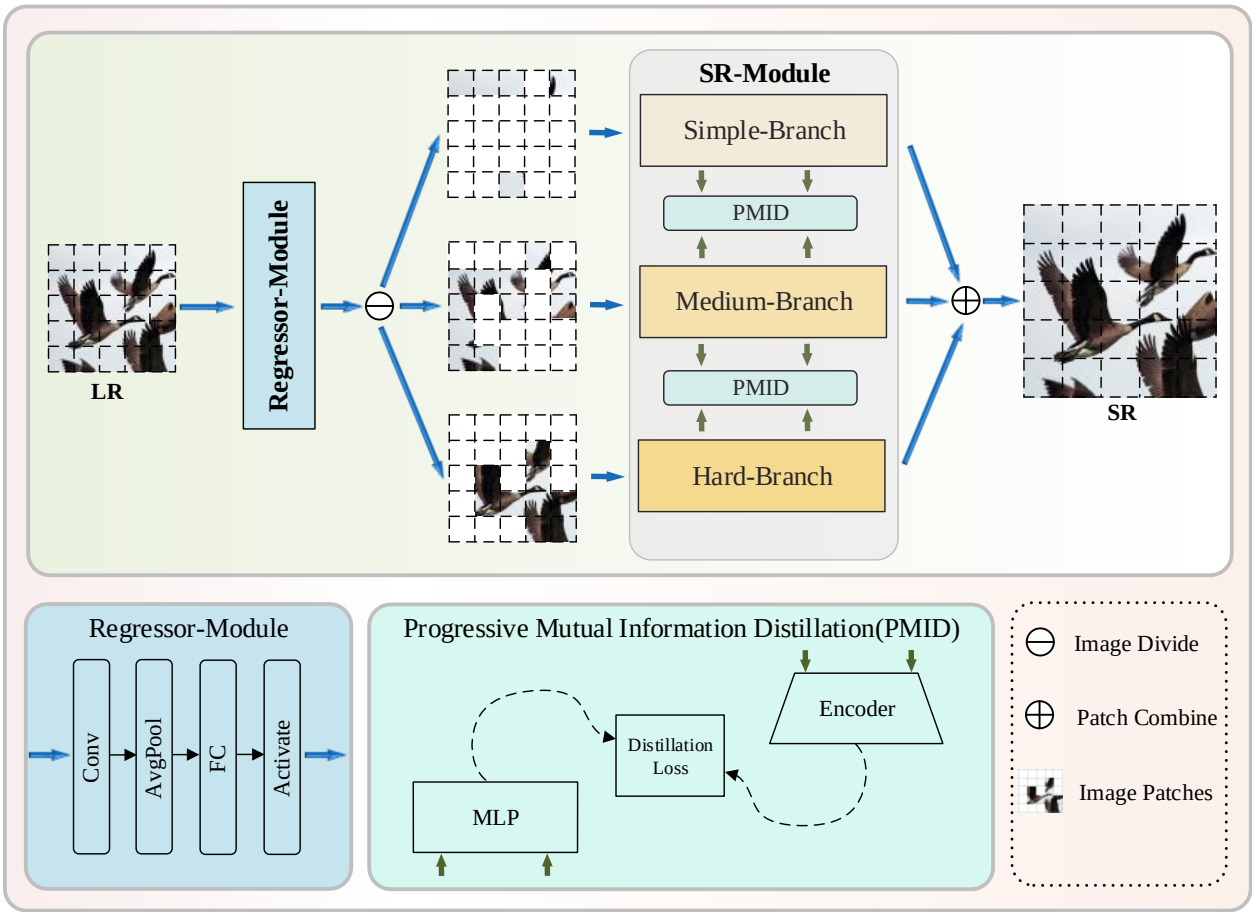


Fig. 2 The overview of our proposed Adaptive Branch Selection(ABS). Regressor-Module: aiming to generate the performance increment between multiple branches; SR-Module: aiming to deal with the corresponding patches.

merged to generate the SR image.

3.1 Regressor-Module

The performance increment is defined as the difference in PSNR values across branches, representing the performance gap between them. The goal of the Regressor-Module is to estimate the performance increment between different branches. To avoid adding excessive FLOPs, we designed a lightweight Regressor-Module. As shown in Fig. 2, the Regressor-Module is a network composed of a convolutional layer, an average pooling, and a fully connected layer. For a 64×64 input patch, the network has FLOPs of 0.7M, accounting for only 0.07% of FSRCNN. Therefore,

it introduces a very little additional computational cost. The Regressor-Module is formulated as:

$$(\hat{p_1}, \hat{p_2}) = Reg(x_i) \tag{1}$$

where x_i denotes the LR image, $(\hat{p_1}, \hat{p_2})$ denotes a 2-dimensional vector. During testing, we control the assignment of each patch to different branches by setting thresholds λ_1 and λ_2. Specifically, patches where p_1 is less than λ_1 are routed to the Simple-Branch. If p_1 is greater than λ_1 but p_2 is less than λ_2, these patches are directed to Medium-Branch. Finally, patches with p_1 and p_2 greater than λ_1 and λ_2 respectively, are sent to a Hard-Branch. By setting different thresholds, we can achieve dynamic branch selection, enabling the SR model to operate with varying FLOPs.

3.2 SR-Module

The SR-Module consists of three branches with identical structures but different channel numbers. As a general acceleration strategy, each branch can be replaced by any other SR network. We use an existing SR model as the *Hard-Branch* and construct the other two branches by reducing the number of channels. As shown in Fig. 2, taking FSRCNN as an example, the Hard-Branch, Medium-Branch, and *Simple-Branch* have channel numbers of 56, 36, and 16, respectively. To better compare and validate the effectiveness of ABS, we maintain the same number of branches as ClassSR. This consistency allows for a fair evaluation between the two methods. The output of each branch is represented as follows:

$$y_i = f_{SR}^k(x_i) \tag{2}$$

where y_i and x_i denotes output and input of the k-th branch in SR-Module.

Inspired by knowledge distillation, we introduce a Progressive Mutual Information Knowledge Distillation (PMID) to encourage branches with fewer channels to learn from those with more channels. At the same time, if the performance gap between the two branches in the distillation process is too large, it may lead to a performance drop. Therefore, we implemented PMID between f_{SR}^1,f_{SR}^2 and f_{SR}^2,f_{SR}^3 rather than between f_{SR}^1,f_{SR}^2 and f_{SR}^1,f_{SR}^3. Due to the differences in feature channel numbers among different branches, directly computing the MAE (Mean Square Error) loss might be inappropriate and could degrade model performance. To address this issue, we proposed PMID to better accommodate these structural differences and enhance overall performance.

PMID maximizes the mutual information between branches instead of directly computing the differences between feature maps. To illustrate PMID, we use the branches f_{SR}^2,f_{SR}^3 as an example. We utilize an encoder to map the features output by f_{SR}^2 into two representations that have the same dimension as the features output from f_{SR}^3, denoted as follows:

$$\mu_i, b_i = E(F_i) \tag{3}$$

where E denotes the encoder, the output of the encoder is represented as μ_i and b_i. F_i is the ouput feature of the branch f_{SR}^2. The encoder consists of two 1×1 convolutional layers followed by activation layers. The representation μ_i contains rich information derived from the branch f_{SR}^2, while b_i is used to control the distillation process. Specifically, when the value of b_i is larger, more knowledge is transferred from f_{SR}^3 to f_{SR}^2, facilitating a stronger learning effect. To avoid the negative impact on model performance caused by directly aligning feature parameters, we use an MLP (Multilayer Perceptron) to transform the output features of f_{SR}^3 before aligning them with the output features of f_{SR}^2. We perform the same operations between branches f_{SR}^2 and f_{SR}^1.

3.3 Loss Functions

The overall loss function of ABS is composed of three key components: the reconstruction loss L_r, the performance incrementation loss L_p, and the distillation loss L_d. Each of these components plays a distinct role in constructing ABS. The L_r is used to ensure the quality of reconstructed high-resolution images, L_d improves the performance of branches with fewer channels and the L_p guarantees the accuracy of the predictions of the regressor. The loss function is defined as:

$$L = w_1 L_r + w_2 L_d + w_3 L_p \tag{4}$$

where w_1, w_2, and w_3 are the weights to balance different loss terms. The above-mentioned loss function and the setting of weights will be detailed below.

Reconstruction Loss. The reconstruction loss uses the $L1$ loss and is defined as follows:

$$L_r = \sum_{i=1}^{N} |\hat{y}_i - y_i| \tag{5}$$

where $\hat{y}_i$ denotes SR results from ABS, and y_i denotes the ground truth. N is the number of batches. The $L1$ loss is widely used in low-level tasks such as image super-resolution, effectively minimizing the gap between the SR image and the HR image.

Distillation Loss. The three branches in ABS have identical structures and are trained on the same standard dataset, which makes them more conducive to benefiting from the knowledge distillation strategy. By employing progressive mutual information knowledge distillation, we avoid directly minimizing the features themselves, instead focusing on the similarity between feature

distributions. This approach further enhances the performance of the branch with fewer channel numbers. The SR model typically consists of multiple stacked blocks with identical structures, we first compute the loss for the output of each block. It is formulated as:

$$L_j = \sum_{i=0}^{N} \frac{|F_i^j - \mu_i|}{b_i} \tag{6}$$

where F_i^j denotes the feature output of the i-th input at the j-th block. Since the deeper layers of the SR model contain more useful information, we adopt a progressive distillation approach. We sum these losses from different blocks and apply different weights to them. The Distillation Loss is defined as follows:

$$L_d = \sum_{j=0}^{M} (\frac{j}{M} * L_j) \tag{7}$$

where M represents the number of blocks in the SR model.

Performance Incrementation Loss. To effectively train our regressor and accurately capture the performance differences among different branches, we designed a novel loss function called Performance Incrementation Loss. This loss aims to minimize the gap between predicted $\hat{p_1}$, $\hat{p_2}$ and actual performance increments p_1, p_2. The performance incrementation loss is formulated as:

$$L_p = ||p_1 - \hat{p_1}||_2^2 + ||p_2 - \hat{p_2}||_2^2 \tag{8}$$

where the p_1 and p_2 represent the performance differences between f_1,f_2 and f_2,f_3. PSNR is widely used in the field of super resolution for quantifying the quality of SR images. It provides a measure of the difference between the pixels of the predicted high-resolution image and the

ground truth image, with higher PSNR values indicating better quality. To more accurately assess the performance increments between branches, we use PSNR values as the evaluation metric. The metric is defined as follows:

$$p_1 = P(f_2(x_i)) - P(f_1(x_i)) \tag{9}$$

$$p_2 = P(f_3(x_i)) - P(f_2(x_i)) \tag{10}$$

where P denotes the computation of PSNR.

3.4 Training strategy

The training strategy of ABS includes two stages. The first stage trains the single SR branch. In the second stage, we fix the parameters of the SR-Module to train the Regressor. After that, by setting thresholds, the input patches are adaptively assigned to different branches for processing, enabling the model to adaptively balance performance and efficiency.

In the first stage, we use L_r to train a basic SR model, which serves as the Hard-Branch f_{SR}^3. Next, we use PMID to train f_{SR}^2, allowing f_{SR}^2 to learn from both the ground truth and branch f_{SR}^3. Finally, due to the large gap between f_{SR}^1 and f_{SR}^3, we use f_{SR}^2 as the teacher network to train f_{SR}^1. We utilize both L_r and L_d to training the branches f_{SR}^2 and f_{SR}^1. The weights w_1 and w_2 are set as 1 and 0.1, respectively. This approach progressively transfers knowledge from the f_{SR}^3 branch to the f_{SR}^1 branch, thereby further improving the performance of the f_{SR}^1 branch.

In the second stage, we train the regressor to accurately estimate the performance increment between branches f_{SR}^1, f_{SR}^2 and f_{SR}^2, f_{SR}^3. Changes in the SR model's parameters can lead to variability in the performance increments, which makes it difficult for the regressor to converge. Therefore, we fix the parameters of the SR-Module to ensure stability in the performance increment

during the second stage. This stabilization is essential for the regressor to learn the performance increments between different branches effectively.

4 Experiments

In this section, we applied the proposed acceleration strategy to FSRCNN,[6] SAFMNet,[9] SMFANet,[10] and FreqFormer,[38] significantly reducing the FLOPs of these models. Meanwhile, we compared our ABS with other existing acceleration strategies. To further validate the effectiveness of our approach, we conducted several ablation studies to analyze the impact of different components and key hyperparameters. Additionally, we performed visual comparisons to illustrate the performance of ABS intuitively.

4.1 Implementation Details

We train the ABS with scaling factors $\times 4$. The batch size and HR image size were set to 32 and 256, respectively. During training, an initial learning rate is set as $1e-3$, with updates to the learning rate following a cosine annealing scheme. In the first stage, the total number of iterations for each branch is 500K. In the second stage, we use 200K iterations to train the regressor. Throughout the training period, we employed horizontal and vertical flipping as data augmentation to increase the diversity of the training data and improve the model's performance. The channel configurations of the three branches are (16, 36, 56) for FSRCNN, (16, 28, 36) for SAFMNet, (16, 28, 36) for SMFANet, and (16, 48, 60) for Freqformer. All PSNR and FLOPs are evaluated on 3080Ti GPUs.

Table 1 PSNR values on DIV2K and Test2K.

Model	Param.	DIV2K	FLOPs	Test2K	FLOPs
FSRCNN[6]	25K	27.82dB	936M(100%)	25.61dB	936M(100%)
FSRCNN-ABS	55K	27.82dB	486M(52%)	25.61dB	515M(55%)
SAFMNet[9]	239K	28.95dB	963M(100%)	26.16dB	963M(100%)
SAFMNet-ABS	440K	28.94dB	790M(82%)	26.16dB	799M(83%)
SMFANet[10]	197K	29.05dB	737M(100%)	26.21dB	737M(100%)
SMFANet-ABS	368K	29.05dB	612M(83%)	26.21dB	619M(84%)
FreqFormer[38]	889K	29.28dB	3.67G(100%)	26.42dB	3.67G(100%)
FreqFormer-ABS	1589K	29.29dB	2.79G(76%)	26.42dB	2.71G(74%)

4.2 Datasets

We train our proposed ABS using the DIV2K[39](index 001-800) dataset. DIV2K is a high-quality image dataset that includes 800 training images and 100 validation images, all with a resolution of 2K. During the testing phase, we first divide the images into multiple overlapping patches of size 64×64 with stride 62. The size of the images in widely used datasets like Set5,[40] Set14,[41] B100,[42] Urban100,[43] and Manga109[44] are generally too small. Therefore, we finally select the DIV2K validation set (index 801-900) and additionally choose three hundred images(index 1201-1500) from the DIV8K[45] datasets as our test dataset to validate the effectiveness of ABS. Some of the images are downsampled to 2K and 4K resolutions to construct the datasets Test2K (index 1201-1300) and Test4K (index 1301-1400), respectively. The remaining images are left unchanged, forming the Test8K (index 1401-1500) dataset. Unless otherwise specified, the following PSNR and FLOPs results are tested on all datasets with a scale factor of ×4.

Table 2 PSNR values on Test4K and Test8K.

Model	Param.	Test4K	FLOPs	Test8K	FLOPs
FSRCNN[6]	25K	26.90dB	936M(100%)	32.66dB	936M(100%)
FSRCNN-ABS	55K	26.89dB	496M(53%)	32.72dB	458M(49%)
SAFMNet[9]	239K	27.62dB	963M(100%)	33.55dB	963M(100%)
SAFMNet-ABS	440K	27.62dB	722M(75%)	33.53dB	693M(72%)
SMFANet[10]	197K	27.70dB	737M(100%)	33.65dB	737M(100%)
SMFANet-ABS	368K	27.70dB	567M(77%)	33.66dB	538M(73%)
FreqFormer[38]	889K	27.91dB	3.67G(100%)	33.88dB	3.67G(100%)
FreqFormer-ABS	1589K	27.91dB	2.53G(69%)	33.88dB	2.35G(64%)

4.3 Main Results

4.3.1 Quantitative Results

We applied our proposed ABS to Transformer-based models like FreqFormer and CNN-based models such as FSRCNN, SAFMNet, and SMFANet. On the Test8K dataset, these models integrated with ABS achieved an average of only 64% of the FLOPs required by the original models. This demonstrates that ABS can be effectively utilized in different network architectures. ABS controls each patch entering different branches by setting thresholds. In order to maintain the same performance as the original models, we set the thresholds λ_1 and λ_2 for the four SR models as follows: (0.83, 0.93) for FSRCNN, (0.78, 0.88) for SAFMNet, (0.75, 0.90) for SMFANet, and (0.72, 0.88) for FreqFormer. As shown in Table1 and Table 2, our ABS maintains SR performance comparable to the original models while reducing FLOPs. Specifically, on the Test8K dataset, FSRCNN-ABS, SAFMNet-ABS, SMFANet-ABS, and FreqFormer-ABS achieved only 49%, 72%, 73%, 64% of the computational cost compared to FSRCNN, SAFMNet, SMFANet, FreqFormer, respectively. Relative to DIV2K, Test2K, and Test4K, ABS achieves greater FLOPs reduction on the Test8K dataset. This is due to the fact that the images in Test8K have a higher

resolution and contain more smooth regions, which are directed to branches with fewer channels by the Regressor-Module. As a result, these smoother areas benefit more from the efficient processing provided by ABS, leading to a more significant reduction in FLOPs. Since we employed three branches with different numbers of channels, the parameters in ABS are approximately twice those of the original models. However, in practical applications, the cost of increasing memory is relatively low, so it is acceptable to trade memory for an improvement in efficiency.

4.3.2 Comparison with Other Accelerate Strategy

We also compared our ABS with other methods for accelerating SR. From Table 3, we can see that ABS achieves the highest reduction in FLOPs while maintaining performance comparable to the original model. The classifier used in FSRCNN-ClassSR is relatively complex, leading to it having five times as many parameters as FSRCNN. Although FSRCNN-ARM and FSRCNN-MGA introduce fewer additional parameters compared to FSRCNN-ABS, their reductions in FLOPs are 13% and 4% less than our method, respectively. FSRCNN-FSR transforms images into the frequency domain and divides them into multiple branches based on frequency levels. The complex image transformation operations, along with the configuration of multiple branches, result in a significantly larger number of additional parameters compared to our method. This proves that our method is superior to existing methods for accelerating SR models.

4.3.3 Performance Efficiency Trade-off Results

By adjusting the thresholds, the ABS is capable of adaptively generating SR models with varying levels of FLOPs to meet different performance and efficiency requirements. We applied ABS to four different models and obtained multiple SR models with varying FLOPs. As shown in Fig. 3,

Table 3 Comparison with existing accelerate strategy.

Model	Param.	Test8K	FLOPs
FSRCNN[6]	25K	32.66dB	936M(100%)
FSRCNN-ClassSR[18]	113K	32.73dB	496M(53%)
FSRCNN-ARM[19]	25K	32.73dB	580M(62%)
FSRCNN-MGA[46]	43K	32.69dB	498M(53%)
FSRCNN-FSR[47]	154K	32.73dB	568M(61%)
FSRCNN-ABS (Ours)	55K	32.72dB	458M(49%)

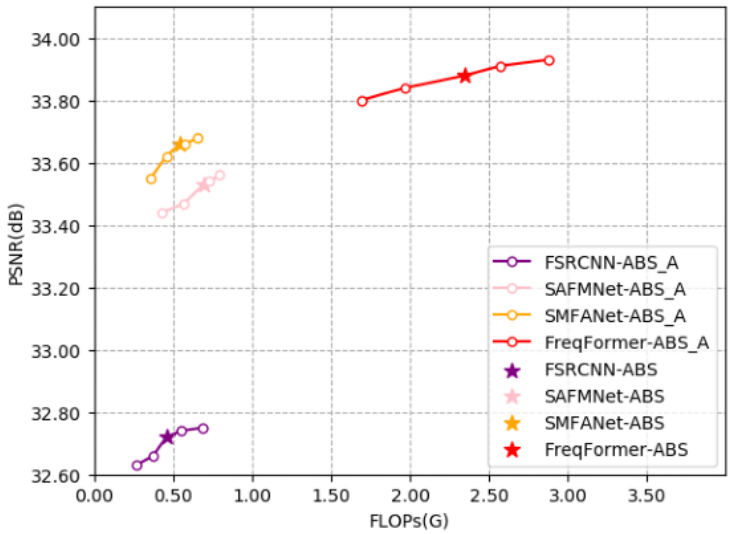


Fig. 3 The performance-efficiency trade-off results tested on the Test8K dataset.

higher FLOPs result in higher PSNR, while lower FLOPs lead to lower PSNR. This demonstrates that our ABS can dynamically adapt to different FLOPs requirements. Specifically, for limited computational resources, lower FLOPs can be adopted, whereas for abundant computational resources, higher FLOPs can be utilized to achieve good SR performance.

4.3.4 Ablation Study on MB(Multi branch) and PMID

ClassSR splits the images in the DIV2K dataset into patches and then divides these patches into three groups based on their PSNR values, ensuring that each group contains an equal number of

Table 4 Ablation Study on MB and MID.

Model	MB	PMID	Test8K	FLOPs
FSRCNN	✗	✗	32.66dB	936M(100%)
FSRCNN	✓	✗	32.68dB	702M(75%)
FSRCNN-ClassSR	✓	✗	32.69dB	618M(66%)
FSRCNN-ABS (Ours)	✓	✓	32.72dB	458M(49%)

patches. The three groups are then used to train three separate branches with different numbers of channels, respectively. Different from ClassSR, we train all three branches using the complete DIV2K dataset. Since all branches utilize the same input, we employ progressive mutual information distillation to enhance the performance of branches with fewer channels. Table 4 shows the ablation experiments on the MB (Multi-Branch) and MID (Progressive Mutual Information Distillation). Note that using only the MB without PMID results in less FLOPs reduction compared to ClassSR, as ClassSR requires branches with fewer channels to process only smooth regions, whereas in ABS, all patches must be processed. This increased complexity makes it more challenging for branches with fewer channels in ABS to achieve comparable performance. Without PMID, our ABS failed to effectively transfer knowledge from branches with more channels to those with fewer channels. After implementing PMID to enhance interaction between multiple branches, our ABS achieves a 9% greater reduction in FLOPs compared to ClassSR.

4.3.5 Ablation Study on Patch Size

Since ABS first splits large images into multiple patches, different patch sizes and strides have an impact on the performance of our RegSR. The ablation study on patch size and stride is shown in the Table 5. We observe that PSNR decreases with smaller patch sizes due to the limited amount of information contained within each patch, which restricts the model performance. When the

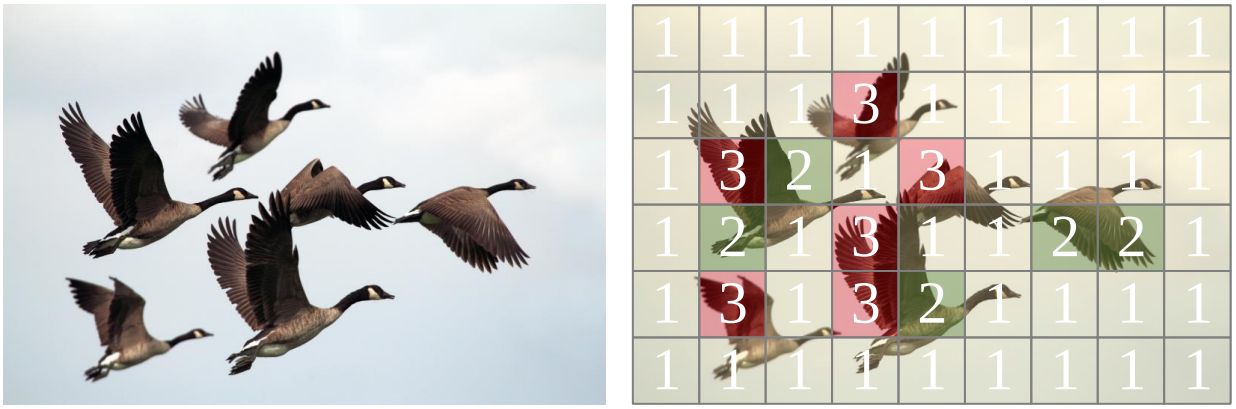
Table 5 Ablation on Patch Size and Stride.

Model	Patch	Stride	Test8K	FLOPs
FSRCNN-ABS	32	30	32.62dB	936M(54%)
FSRCNN-ABS	40	38	32.67dB	524M(56%)
FSRCNN-ABS	48	46	32.69dB	515M(55%)
FSRCNN-ABS	64	62	32.72dB	458M(49%)
FSRCNN-ABS	72	70	32.72dB	466M(50%)

Table 6 Ablation on the Number of Branches.

Model	Test8K	FLOPs
FSRCNN	32.66dB	936M(100%)
FSRCNN-ABS(2)	32.70dB	442M(47%)
FSRCNN-ABS(3)	32.72dB	458M(49%)
FSRCNN-ABS(4)	32.73dB	475M(51%)
FSRCNN-ABS(5)	32.73dB	482M(51%)

patch size exceeds 64, the improvement in SR performance becomes negligible, while the FLOPs increase. Therefore, we chose 64 and 62 as the patch size and stride for ABS, respectively.



(a) Original Image (b) ABS Patches

Fig. 4 Visualization of ABS. The number in the patch represents the branch index of each patch.

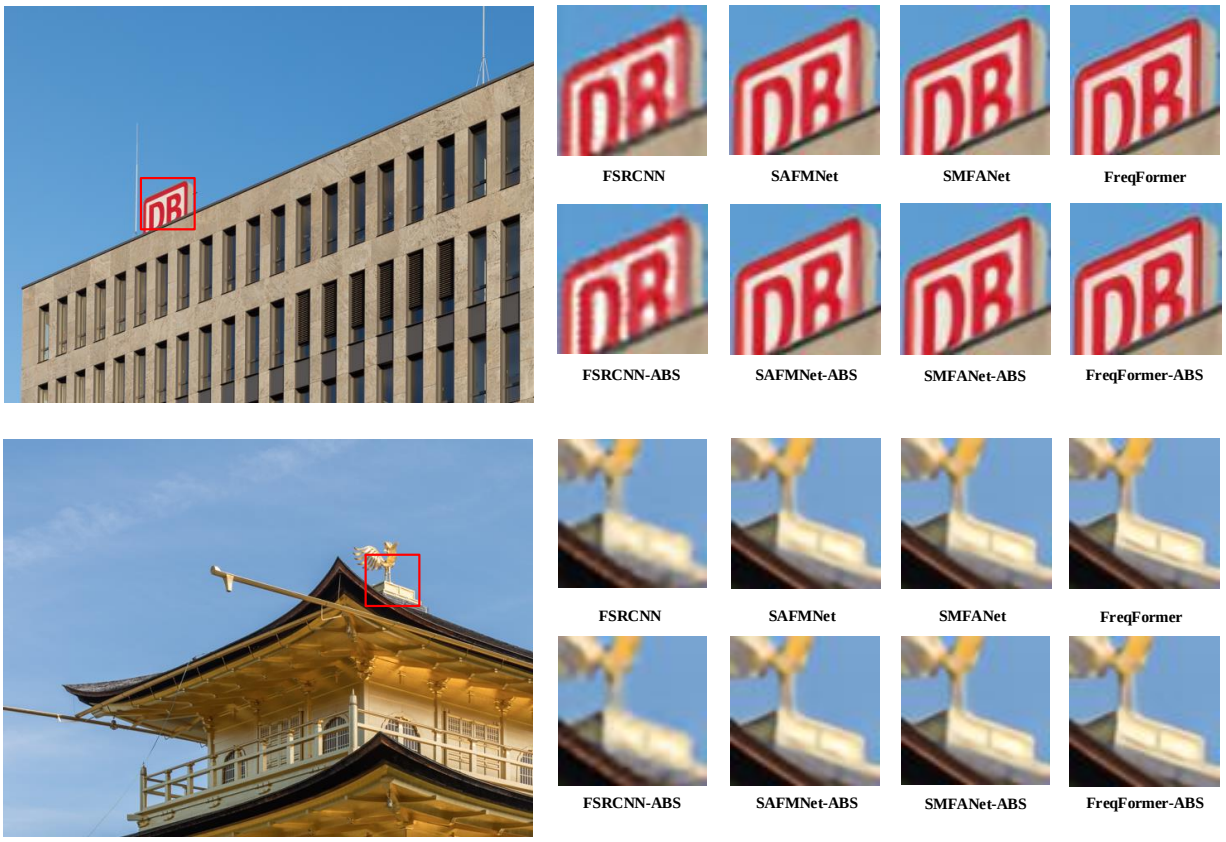


Fig. 5 Visualization results on Test2K. We selected Img No.1238 and No.1300 from the Test2K dataset to demonstrate that ABS achieves comparable SR performance with original models.

4.3.6 Ablation Study on the Number of Branches

ABS adopts three branches with different numbers of channels to accelerate the SR model. As shown in Table 6, we explored the impact of the number of branches. Our channel configurations are set as follows: for 2 branches, we use (16, 56), for 3 branches, we use (16, 36, 56), for 4 branches, we use (16, 28, 36, 56), and for 5 branches, we use (16, 28, 36, 48, 56). It can be seen that the number of branches has a minor impact on PSNR and FLOPs. Although increasing the number of branches slightly improves PSNR, it also leads to a higher number of FLOPs. Therefore, the number of branches can be chosen based on the specific requirements of the practical application.

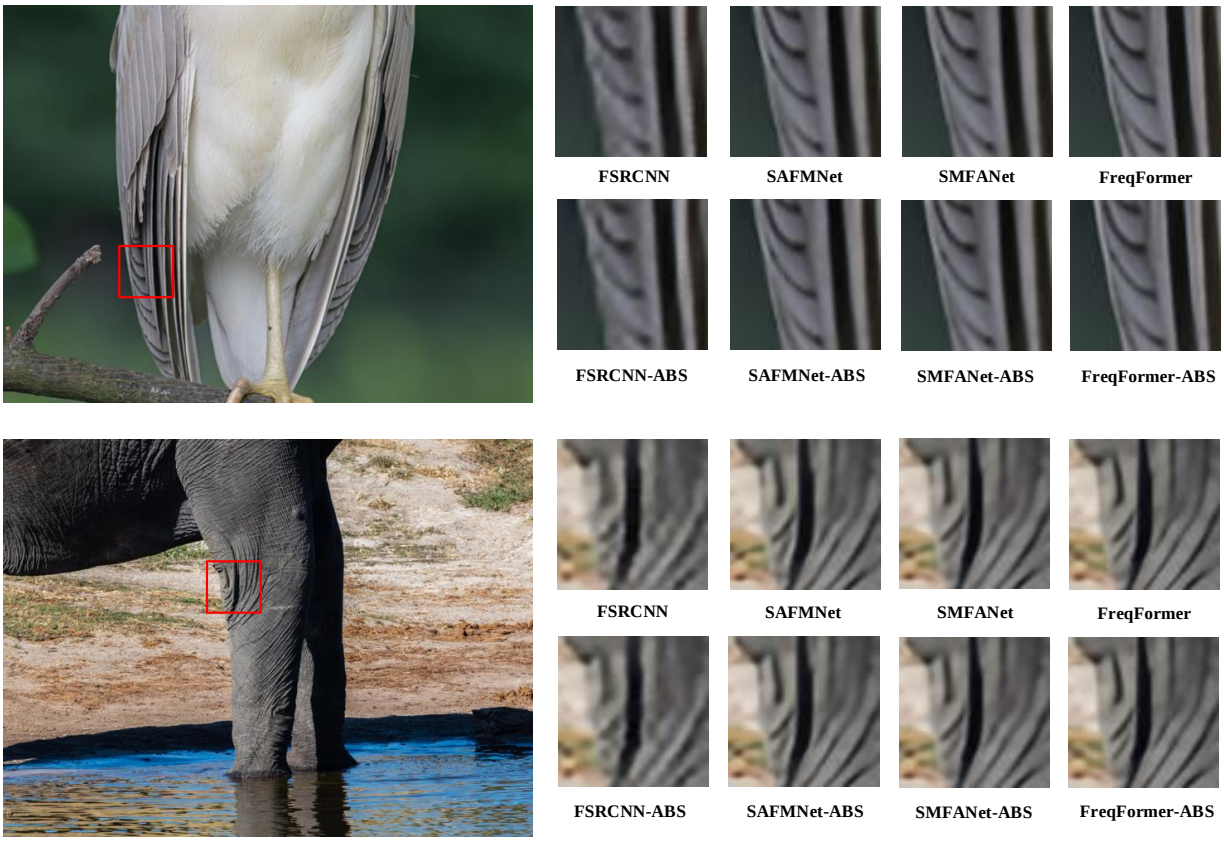


Fig. 6 Visualization results on Test2K. We selected Img No.1224 and No.1277 from the Test2K dataset to demonstrate that ABS achieves comparable SR performance with original models.

4.3.7 Visual Results

Fig. 4 illustrates the assignment of different image patches to their respective branches. The original image is divided into ABS patches, where the number in the ABS patches indicates the patch belongs to i-th branch. Specifically, 1, 2, and 3 represent the Simple-Branch, Medium-Branch, and Hard-Branch, respectively. As can be seen, patches with more textures are routed to the branch with the highest channels (Hard-Branch), smoother regions are processed by the branch with fewer channels (Simple-Branch), and patches with intermediate complexity are handled by Medium-Branch. This demonstrates that our ABS can dynamically and efficiently process image patches based on their texture complexity, thereby reducing computational resources cost and

achieving comparable SR performance.

Fig. 5 and Fig. 6 show the visualization results of our ABS. We present the visual results of two groups of images (No. 1238, No. 1300 and No.1224, No.1277) from the Test2K dataset. To better demonstrate that our ABS achieves visual results comparable to the original model, we have magnified specific regions of these images for detailed comparison.

5 Conclusion

In this paper, we propose ABS for Accelerating Image Super-Resolution. ABS leverages the Regressor-Module to predict the performance increment between branches. During testing, dynamic branch selection is achieved by setting thresholds, which effectively reduces FLOPs. In the meantime, we proposed PMID to further enhance the performance of branches with fewer channels. Extensive experiments demonstrate that our ABS effectively reduces the model's FLOPs while maintaining SR performance.

Disclosures

The article has no conflicts of interest.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 61976079), the Anhui Key Research and Development Program (No.202004a05020039), the Anhui High-level Talents Program(No.T000642),the National Key R&D Program of China(No.2018AAA0100100), Science and Technology Innovation 2030 Major Projects of China (No. 2021Z D0201904), the Key Project of Science and Technology of Guangxi (No.AA22068057 and 2021AB20147), the Natural Science Foundation of Guangxi (No.2021JJA170204 and 2021JJA170199), and the Guangxi Science and Technology Base and Talents Special Project (No. 2021AC19354 and 2021AC19394).

References

1 H. Li, Z. Liu, Y. Liu, *et al.*, "Lightweight image super-resolution network using 3d convolutional neural networks," *Journal of Electronic Imaging* **33**(1), 013016–013016 (2024).

2 H. Shen, H. Ding, Y. Zhang, *et al.*, "Spatial-frequency adaptive remote sensing image dehazing with mixture of experts," *IEEE Transactions on Geoscience and Remote Sensing* **62**, 1–14 (2024).

3 L. Chen, J. Zuo, K. Du, *et al.*, "Image super-resolution using dilated neighborhood attention transformer," *Journal of Electronic Imaging* **33**(2), 023003–023003 (2024).

4 H. Shen, Z.-Q. Zhao, Y. Zhang, *et al.*, "Mutual information-driven triple interaction network for efficient image dehazing," in *Proceedings of the 31st ACM international conference on multimedia*, 7–16 (2023).

5 H. Shen, Z.-Q. Zhao, W. Liao, *et al.*, "Joint operation and attention block search for lightweight image restoration," *Pattern Recognition* **132**, 108909 (2022).

6 C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, 391–407, Springer (2016).

7 N. Ahn, B. Kang, and K.-A. Sohn, "Fast, accurate, and lightweight super-resolution with cascading residual network," in *ECCV*, 252–268 (2018).

8 H. Shen, Z.-Q. Zhao, and W. Zhang, "Adaptive dynamic filtering network for image denoising," in *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**(2), 2227–2235 (2023).

9 L. Sun, J. Dong, J. Tang, *et al.*, "Spatially-adaptive feature modulation for efficient image super-resolution," in *Proceedings of the IEEE/CVF international conference on computer vision*, 13190–13199 (2023).

10 M. Zheng, L. Sun, J. Dong, *et al.*, "Smfanet: A lightweight self-modulation feature aggregation network for efficient image super-resolution," in *European Conference on Computer Vision*, 359–375, Springer (2024).

11 W. Shi, J. Caballero, F. Huszár, *et al.*, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1874–1883 (2016).

12 Y. Zhang, H. Chen, X. Chen, *et al.*, "Data-free knowledge distillation for image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7852–7861 (2021).

13 Y. Jiang, C. Feng, F. Zhang, *et al.*, "Mtkd: Multi-teacher knowledge distillation for image super-resolution," in *European Conference on Computer Vision*, 364–382, Springer (2024).

14 H. Qin, Y. Zhang, Y. Ding, *et al.*, "Quantsr: accurate low-bit quantization for efficient image super-resolution," *Advances in Neural Information Processing Systems* **36**, 56838–56848 (2023).

15 Y. Romano, J. Isidoro, and P. Milanfar, "Raisr: rapid and accurate image super resolution," *IEEE Transactions on Computational Imaging* **3**(1), 110–125 (2016).

16 S. Wang, J. Liu, K. Chen, *et al.*, "Adaptive patch exiting for scalable single image super-resolution," in *European Conference on Computer Vision*, 292–307, Springer (2022).

17 C. Ding, Z. Zhao, and Y. Zhao, "Pssr: Hybrid path selection mechanism for efficient image super-resolution," in *ICASSP 2025-2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1–5, IEEE (2025).

18 X. Kong, H. Zhao, Y. Qiao, *et al.*, "Classsr: A general framework to accelerate super-resolution networks by data characteristic," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12016–12025 (2021).

19 B. Chen, M. Lin, K. Sheng, *et al.*, "Arm: Any-time super-resolution method," in *European Conference on Computer Vision*, 254–270, Springer (2022).

20 C. Dong, C. C. Loy, K. He, *et al.*, "Learning a deep convolutional network for image super-resolution," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part IV 13*, 184–199, Springer (2014).

21 J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1646–1654 (2016).

22 J. Kim, J. K. Lee, and K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1637–1645 (2016).

23 Y. Zhang, K. Li, K. Li, *et al.*, "Image super-resolution using very deep residual channel attention networks," in *Proceedings of the European conference on computer vision (ECCV)*, 286–301 (2018).

24 T. Dai, J. Cai, Y. Zhang, *et al.*, "Second-order attention network for single image super-

resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 11065–11074 (2019).

25 C. Ledig, L. Theis, F. Huszár, *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*, 4681–4690 (2017).

26 Y. Tian, H. Chen, C. Xu, *et al.*, "Image processing gnn: Breaking rigidity in super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 24108–24117 (2024).

27 H. G. Lee, J.-S. Yoo, and S.-W. Jung, "Refqsr: Reference-based quantization for image super-resolution networks," *IEEE Transactions on Image Processing* **33**, 2823–2834 (2024).

28 H. Chen, Y. Wang, T. Guo, *et al.*, "Pre-trained image processing transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 12299–12310 (2021).

29 J. Liang, J. Cao, G. Sun, *et al.*, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 1833–1844 (2021).

30 Z. Chen, Y. Zhang, J. Gu, *et al.*, "Dual aggregation transformer for image super-resolution," in *Proceedings of the IEEE/CVF international conference on computer vision*, 12312–12321 (2023).

31 X. Chen, X. Wang, J. Zhou, *et al.*, "Activating more pixels in image super-resolution transformer," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 22367–22377 (2023).

32 Z. Lu, J. Li, H. Liu, *et al.*, "Transformer for single image super-resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 457–466 (2022).

33 A. Zhang, W. Ren, Y. Liu, *et al.*, "Lightweight image super-resolution with superpixel token interaction," in *Proceedings of the IEEE/CVF international conference on computer vision*, 12728–12737 (2023).

34 Y. Zhang, X. Li, and J. Zhou, "Sftgan: a generative adversarial network for pan-sharpening equipped with spatial feature transform layers," *Journal of Applied Remote Sensing* **13**(2), 026507–026507 (2019).

35 W. Xie, D. Song, C. Xu, *et al.*, "Learning frequency-aware dynamic network for efficient super-resolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4308–4317 (2021).

36 X. Deng, H. Wang, M. Xu, *et al.*, "Lau-net: Latitude adaptive upscaling network for omnidirectional image super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9189–9198 (2021).

37 Y. Wang, Y. Liu, S. Zhao, *et al.*, "Camixersr: Only details need more" attention"," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 25837–25846 (2024).

38 T. Dai, J. Wang, H. Guo, *et al.*, "Freqformer: frequency-aware transformer for lightweight image super-resolution," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 731–739 (2024).

39 R. Timofte, E. Agustsson, L. Van Gool, *et al.*, "Ntire 2017 challenge on single image super-

resolution: Methods and results," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 114–125 (2017).

40 M. Bevilacqua, A. Roumy, C. Guillemot, *et al.*, "Low-complexity single-image super-resolution based on nonnegative neighbor embedding," in *British Machine Vision Conference*, 1–10 (2012).

41 R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces: 7th International Conference, Avignon, France, June 24-30, 2010, Revised Selected Papers 7*, 711–730, Springer (2012).

42 D. Martin, C. Fowlkes, D. Tal, *et al.*, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings eighth IEEE international conference on computer vision. ICCV 2001*, **2**, 416–423, IEEE (2001).

43 J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5197–5206 (2015).

44 Y. Matsui, K. Ito, Y. Aramaki, *et al.*, "Sketch-based manga retrieval using manga109 dataset," *Multimedia tools and applications* **76**, 21811–21838 (2017).

45 S. Gu, A. Lugmayr, M. Danelljan, *et al.*, "Div8k: Diverse 8k resolution image dataset," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 3512–3516, IEEE (2019).

46 X. Hu, J. Xu, S. Gu, *et al.*, "Restore globally, refine locally: A mask-guided scheme to

accelerate super-resolution networks," in *European Conference on Computer Vision*, 74–91, Springer (2022).

47 J. Li, T. Dai, M. Zhu, *et al.*, "Fsr: A general frequency-oriented framework to accelerate image super-resolution networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, **37**(1), 1343–1350 (2023).

Cheng Ding is currently pursuing the Ph.D. degree. His current research interests include computer vision, image restoration, and image super-resolution.

Zhong-Qiu Zhao received the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2007. From 2008 to 2009, he held a post-doctoral position in image processing with the CNRS UMR6168 Lab Sciences de Information et des Systé mes, La Garde, France. From 2013 to 2014, he was a Research Fellow in image processing with the Department of Computer Science, Hong Kong Baptist University, Hong Kong. He is currently a Professor with the Hefei University of Technology, Hefei. His current research interests include pattern recognition, image processing, and computer vision.

Hao Shen received the Ph.D. degree from the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei, China, in 2024, where he is currently a Lecturer with the School of Public Security and Emergency Management, Anhui University of Science and Technology, Hefei, China. He has published more than 10 papers in conferences such as AAAI, ACM MM, CVPR, ECAI, and ICME, and journals such as IEEE TGRS and PR. His main research interests include image restoration and deep learning.

List of Figures

List of Tables

6 Ablation on the Number of Branches.