Available Online: 3 September 2026 DOI: 10.54254/2977-3903/2026.36523

DADE: difficulty-aware distillation-enhanced network for single image super-resolution

Yuxuan Lin

College of Computer Science and Cybersecurity, Chengdu University of Technology, Chengdu, China

linyuxuan@stu.cdut.edu.cn

Abstract. Single Image Super-Resolution (SISR) has achieved remarkable progress with deep neural networks, but the ever-growing model depth brings a heavy computational burden that limits practical deployment. Reducing the inference cost of SR networks has therefore become a central concern. A promising direction is to exploit the fact that image regions differ in restoration difficulty and to allocate computation accordingly. Existing content-adaptive methods have shown the potential of this idea, yet they typically rely on maintaining several separate sub-networks, so that the overall computational and storage cost remains high. In this paper, we propose DADE, a Difficulty-Aware Distillation-Enhanced network for efficient super-resolution. A single backbone is equipped with multiple intermediate experts corresponding to shallow, medium and deep computational paths that share their parameters, and a lightweight difficulty-aware classifier is jointly trained with the backbone to route each image patch to a suitable expert. To strengthen the shallow paths, we further introduce a knowledge-distillation scheme in which the deepest expert acts as a teacher and supervises the earlier experts during training. Extensive experiments on four standard benchmarks show that the proposed method substantially reduces the computational cost of super-resolution while maintaining reconstruction quality, with only negligible degradation in Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM), and at the same time greatly lowers the number of stored parameters.

Keywords: single image super-resolution, lightweight, dynamic networks, knowledge distillation

1. Introduction

Single Image Super-Resolution (SISR) aims to reconstruct a High-Resolution (HR) image from its Low-Resolution (LR) counterpart, and is a fundamental low-level vision task with wide applications in photography, surveillance, medical imaging and remote sensing. Since the seminal SRCNN [1] demonstrated that a convolutional neural network can learn the LR-to-HR mapping end-to-end, deep learning has become the dominant paradigm for SISR. Subsequent works progressively increased network depth and representational power: Very Deep Super-Resolution (VDSR) [2] introduced very deep networks with residual learning, Enhanced Deep Super-Resolution (EDSR) [3] removed batch normalization and greatly enlarged the model, Residual Channel Attention Network (RCAN) [4] proposed a residual-in-residual structure with channel attention exceeding 400 layers, and Residual Dense Network (RDN) [5] exploited dense connections.

More recently, Transformer-based models such as SwinIR [6] and HAT [7] have set new state-of-the-art records by modeling long-range dependencies through self-attention.

Despite their impressive accuracy, these networks are computationally expensive. In practical usage, large images (2K-8K) are decomposed into small sub-images (patches) that are processed independently and then stitched back together. However, different patches carry very different amounts of information: smooth regions such as sky or flat walls are easy to super-resolve, whereas regions with rich textures are hard. An intuitive idea therefore arises: simple patches should be handled by a simple network and complex patches by a complex one. Therefore, processing every patch with the same network inevitably wastes computation on the easy ones.

This idea has motivated a line of content-adaptive or dynamic SR methods that allocate computation according to patch difficulty [8-10]. ClassSR [8] realizes it with a classification module that assigns each patch to one of several independent SR sub-networks of different sizes, trained jointly with a Class-Loss and an Average-Loss so that the three classes are used in balanced proportions, saving up to 50% FLOPs on large-image benchmarks. However, maintaining several separate networks multiplies the number of parameters and the storage cost. ARM [9] mitigates this by sharing weights among sub-networks through a supernet and an edge-to-PSNR lookup table, and PCSR [10] pushes the granularity to the pixel level. Nevertheless, most of these methods are designed and validated on a specific backbone, and their generality across the very different architectural families used in modern SR has not been systematically established.

To quantify this heterogeneity in our own setting, we conduct a simple experiment. We first build a progressive network with three experts of increasing depth (a shallow, a medium and a deep expert), and decompose the images of the standard benchmarks into a large number of patches. Each patch is super-resolved by all three experts, and for every patch we record how much its reconstruction quality (PSNR) improves as the network goes deeper. We then sort all patches by this quality gain and evenly divide them into three groups (simple, medium and hard) according to how much they benefit from additional depth.

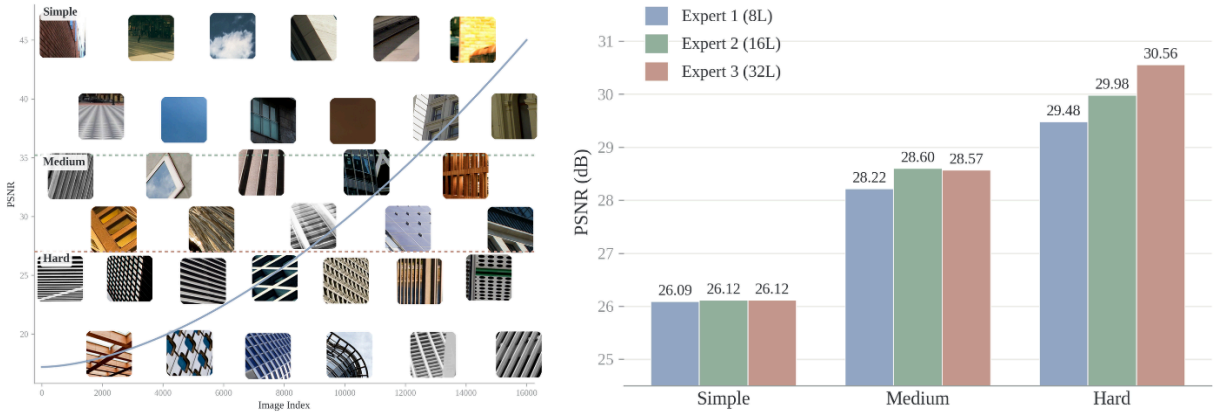

Figure 1. Restoration difficulty varies across patches. Left: three patch types of increasing texture comply. Right: average PSNR of the three difficulty groups at the three experts

As shown in Figure 1, the three groups respond very differently to network depth: for simple patches, the deep expert brings essentially no improvement over the shallow one, indicating that a shallow network already suffices; for medium patches, deeper experts yield a moderate gain; and for hard patches, the quality improves substantially as the network deepens, so that only the deepest expert can faithfully restore their fine details. Figure 3 further illustrates this at the patch level: the reconstructions of a simple patch are almost identical

across the three experts, whereas those of a hard patch become progressively sharper from the shallow to the deep expert. Moreover, as visualized in the spatial routing map of Figure 2, simple and hard regions are distributed across the whole image, so a large portion of patches can be safely handled by shallow experts. This large gap in how different patches respond to network depth is precisely what makes difficulty-aware computation allocation worthwhile, and it directly motivates the design of our framework.

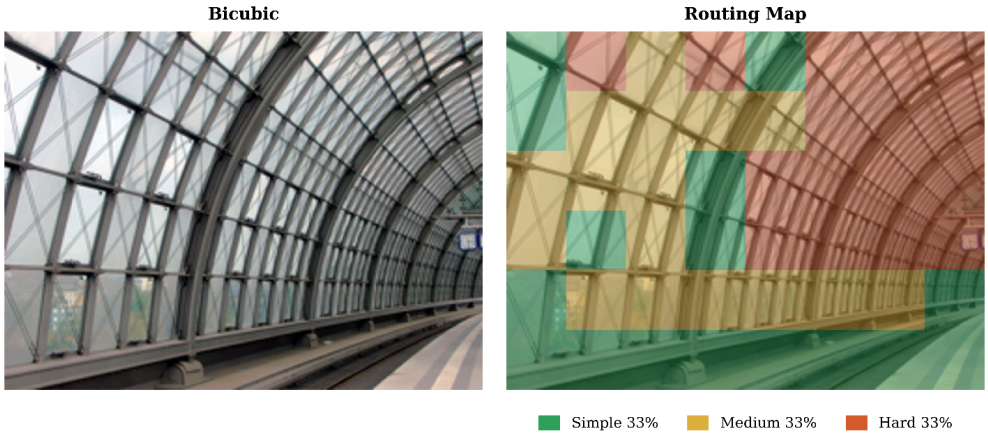

Figure 2. Spatial routing map on a sample Urban100 image

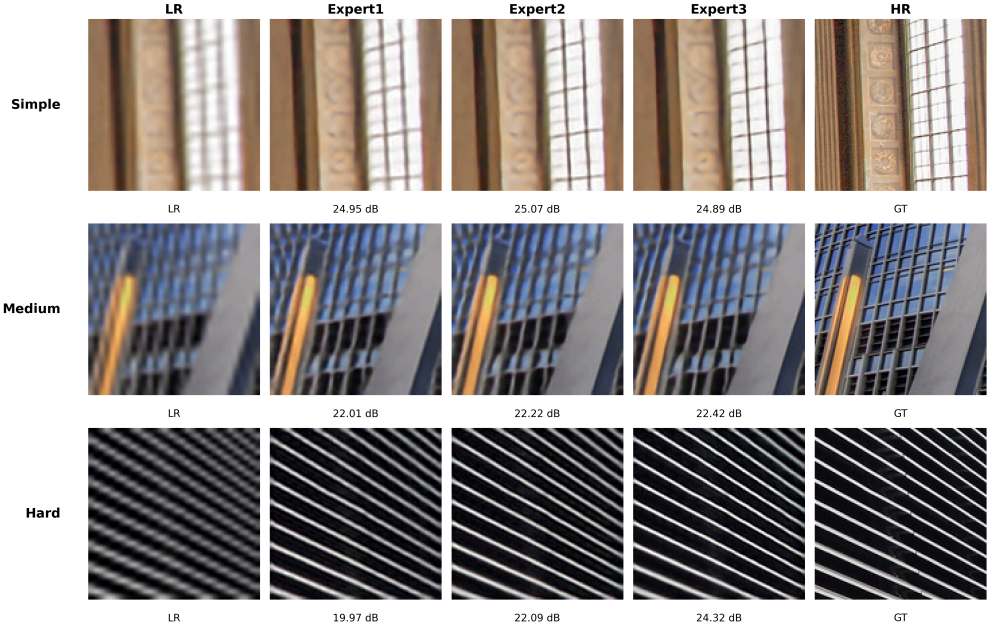

Figure 3. Reconstruction of three representative patches (simple, medium, hard) at the three experts (Expert 1/2/3) and the HR ground truth

In this paper we revisit the difficulty-aware acceleration paradigm from the viewpoint of parameter efficiency and generality. Instead of storing several independent networks, we equip a single backbone with multiple intermediate experts placed at increasing depths, so that a shallow, a medium and a deep computational path share the same parameters. A lightweight difficulty-aware classifier routes each patch to one expert. To compensate for the limited capacity of the early experts, the deepest expert acts as a teacher and distills its knowledge into the shallower experts during training. Finally, the classifier and the backbone are

jointly fine-tuned so that the network adapts itself to the routing decisions. The whole framework is deliberately architecture-agnostic.

The main contributions of this work are summarized as follows:

(1) We introduce a knowledge-distillation scheme in which the deepest expert supervises the shallower experts, and show that its benefit is proportional to the performance gap between experts.

(2) We design a lightweight difficulty-aware classifier and a joint training procedure with an average-distribution constraint and a difficulty-supervised cross-entropy loss, together with an inference-time balanced routing strategy.

(3) We replace the multiple independent sub-networks of existing content-adaptive methods with a single backbone equipped with multiple intermediate experts that share their parameters, which substantially reduces the number of stored parameters while retaining difficulty-aware acceleration.

(4) We show that the proposed framework is generic: it can be instantiated on representative backbones spanning different architectural families, and consistently reduces computation with negligible quality loss, achieving a better efficiency-quality trade-off than most existing lightweight SR networks.

2. Related work

2.1. Single image super-resolution

Deep-learning-based SISR began with SRCNN [1], a three-layer CNN that learns the LR-to-HR mapping. FSRCNN [11] accelerated it by operating in the LR space and using a deconvolution layer for upsampling. VDSR [2] showed that increasing depth with global residual learning substantially improves accuracy. EDSR [3] simplified the residual block by removing batch normalization and scaled the model up, winning the NTIRE 2017 challenge [12]. RDN [5] combined residual and dense connections to fully exploit hierarchical features, while RCAN [4] proposed a Residual-in-Residual (RIR) structure with channel attention that adaptively rescales channel-wise features, enabling networks with more than 400 layers. Beyond PSNR-oriented training, SRGAN [13] and ESRGAN [14] introduced adversarial and perceptual objectives to produce photo-realistic textures. More recently, Transformer-based approaches have pushed the state of the art: SwinIR [6] adapts the shifted-window self-attention of the Swin Transformer [15] to image restoration through Residual Swin Transformer Blocks (RSTB); HAT [7] activates more pixels by combining channel attention with window-based and overlapping cross-attention; and Restormer [16] and Uformer [17] design efficient Transformer architectures for high-resolution restoration. Comprehensive surveys of the field can be found in [18-20]. While these networks steadily improve reconstruction quality, they do so at the cost of ever-growing computation, which motivates research on efficient and lightweight SR.

2.2. Lightweight super-resolution

A large body of work aims to reduce the cost of SR networks. One line designs compact architectures: CARN [21] introduces cascading connections over a residual network and a mobile variant CARN-M using group convolution; IMDN [22] proposes information multi-distillation blocks that progressively split and fuse features; RLFN [23], the winner of the NTIRE 2022 efficiency track, further streamlines the distillation blocks; and FDIWN [24] designs feature-distillation interaction-weighting blocks for lightweight SR.A second line compresses existing networks through knowledge distillation [25], where a compact student learns from a larger teacher; FAKD [26] transfers feature affinity matrices, and the privileged-information framework of [27] and joint-distillation methods improve the student without extra inference cost. Other directions include network quantization, e.g. the content-aware dynamic quantization CADyQ [28], and iterative pruning

combined with distillation as in DIPNet [29]. Our method also uses distillation, but in an intra-network manner: rather than distilling from a separate large teacher into a separate small student, the deepest expert of a single backbone distills into its own shallower experts, so that no additional teacher network needs to be stored.

ClassSR [8] classifies each patch into simple/medium/hard and processes it with one of three independent sub-networks, trained jointly with a Class-Loss and an Average-Loss; it saves up to 50% FLOPs on large images but triples the parameter count because the sub-networks are independent. ARM [9] removes the extra parameters by letting the sub-networks share weights within a supernet and by building an edge-to-PSNR lookup table to estimate the best subnet for each patch. PCSR [10] performs the distribution at the pixel level rather than the patch level and offers tunable performance-cost trade-offs at inference. Our framework belongs to the same family but differs in two respects: (i) we use a single progressive backbone with multiple experts, so that the different computational paths share all parameters, and (ii) we combine intra-network distillation with a joint training that explicitly balances the three routing decisions, and we validate the design across CNN and Transformer backbones rather than a single architecture.

3. Method

Here, we first give an overview of the proposed framework and then describe its two key components: the distillation-enhanced progressive backbone (Section 3.2) and the difficulty-aware dynamic routing with joint training (Section 3.3).

3.1. Overview

Single image super-resolution aims to recover a high-resolution image I^{HR} from a low-resolution observation I^{LR}, which is modeled as a degradation of I^{LR}.The goal is to learn a mapping f such that the super-resolved image $I^{LR} = f_{IR(I^{HR})}$ approximates I^{HR}. Given an LR image, we follow the common practice of decomposing it into overlapping patches. Each patch is fed to a single progressive backbone that contains three experts located at increasing depths, corresponding to a shallow, a medium and a deep computational path that share all preceding parameters. A lightweight classifier predicts the restoration difficulty of the patch and routes it to the corresponding expert; easy patches leave the network early, saving computation, while hard patches traverse the full depth. The super-resolved patches are then stitched back into the output image. Training proceeds in two stages: (1) the backbone is trained so that all three experts can reconstruct the HR patch, with the deepest expert distilling knowledge into the shallower ones; (2) the classifier is jointly trained with the backbone under an average-distribution constraint and a difficulty-supervised loss so that the three experts are used in balanced proportions and each patch is routed to a suitable path. Let the LR patch be x and the ground-truth HR patch be y; the three experts Single image super-resolution aims to recover a high-resolution image I^{HR} from its low-resolution observation I^{LR}, which is a degraded version of I^{HR}. The goal is to learn a mapping f such that the super-resolved image $I^{LR} = f_{IR(I^{HR})}$ approximates the ground truth I^{HR}.

Given an LR image, we follow the common practice of decomposing it into overlapping patches, each of which is processed by our DADE network. The core of DADE is a single progressive backbone containing three experts loc+le hard patches traverse the full depth. The super-resolved patches are then stitched back into the output image.

The framework is optimized in two stages. In the first stage, the backbone is trained so that all three experts can reconstruct the HR patch, with the deepest expert distilling its knowledge into the shallower ones to strengthen the early experts. In the second stage, the classifier is jointly trained with the backbone under an

average-distribution constraint and a difficulty-supervised loss, so that the three experts are used in balanced proportions and each patch is routed to a suitable path. Let the LR patch be x and the ground-truth HR patch be y ; the three experts produce reconstructions $S_1(x)$, $S_2(x)$, $S_3(x)$ of increasing capacity. The overall framework is illustrated in Figure 4.

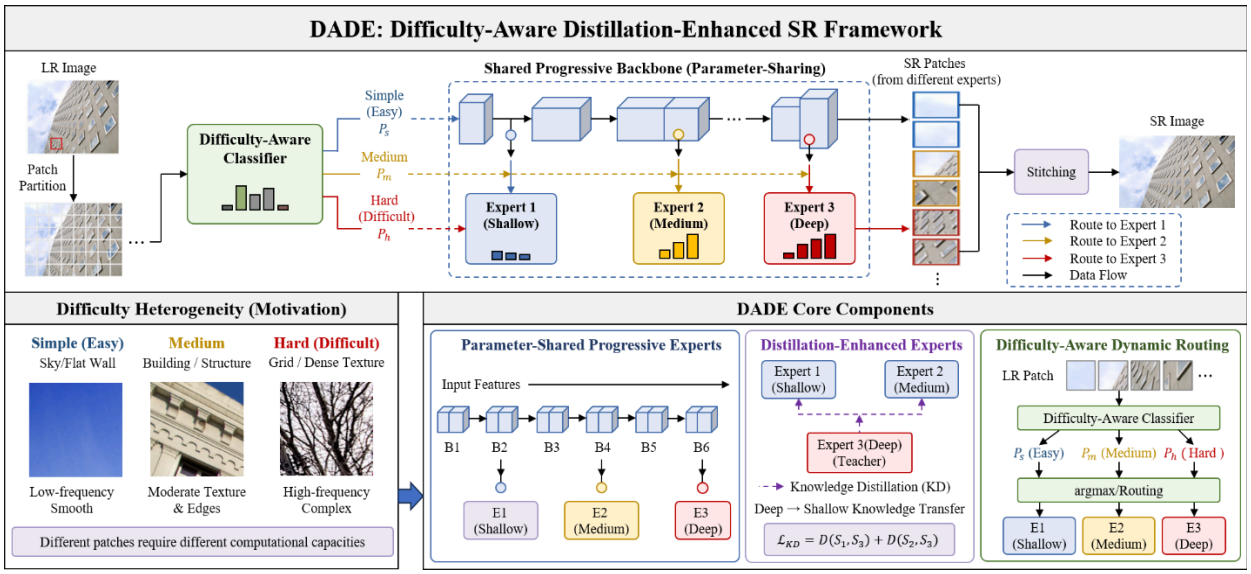

Figure 4. Overview of the DADE-difficulty-aware distillation-enhanced progressive network

3.2. Distillation-enhanced progressive backbone

The backbone produces progressive outputs at three experts of increasing depth: a shallow expert S_1, a medium expert S_2 and a deep expert S_3, each attached after an increasing number of feature-extraction blocks. Each expert has its own tail convolution, upsampler and reconstruction head, while the shallow experts reuse all preceding parameters of the deep one. Consequently, the three computational paths are contained in a single network, and the total parameter count is close to that of one network plus three small heads, in sharp contrast to storing several complete independent networks.

The backbone is trained with an L1 reconstruction loss applied to all three experts simultaneously:

$$L_{rec} = |S_{1(x)} - y| + |S_{2(x)} - y| + |S_{3(x)} - y| \tag{1}$$

To improve the weaker early experts, we introduce an intra-network knowledge-distillation scheme in which the deepest expert S_3 serves as the teacher network and the shallow experts S_1 and S_2 serve as student networks. The distillation loss encourages the student outputs to approach the teacher output:

$$L_{distill} = 0.4 \cdot |S_{1(x)} - sg(S_{3(x)})| + 0.6 \cdot |S_{2(x)} - sg(S_{3(x)})| \tag{2}$$

where $sg(\cdot)$ denotes stop-gradient so that the teacher is not degraded. The total backbone loss is $L_{backbone} = L_{rec} + \lambda_d \cdot L_{distill}$, with $\lambda_d = 0.5$. In this way, the deep expert transfers its representational knowledge to the shallow experts, raising the quality of the early outputs without introducing any additional network. The effectiveness of this distillation is further analyzed in the ablation study.

3.3. Difficulty-aware dynamic routing

Classifier. The routing decision is made by a lightweight classifier that combines a small convolutional branch operating on the LR patch with a multi-layer-perceptron branch operating on eight hand-crafted features

(gradient magnitude mean and standard deviation, Laplacian edge density, local variance, high-frequency energy, mean intensity, edge-pixel ratio and colour variance). The two branches are concatenated and mapped to three logits. To guarantee that the training and test statistics are identical, the mean and standard deviation used to normalize the hand-crafted features are computed once on the training set and stored together with the classifier weights, and the same statistics are applied at test time.

Joint training losses. Unlike a separately trained router attached to a fixed backbone, we optimize the classifier and the backbone together: the classifier is trained with a normal learning rate while the backbone is fine-tuned with a small learning rate, so that the backbone gradually adapts its experts to the routing decisions. During joint training the classifier produces a probability vector $p = (p_0, p_1, p_2)$ over the three experts via a softmax, and three losses are used. First, a soft-routing reconstruction loss weights the three expert reconstruction losses by their routing probabilities:

$$L_{route} = mean(p_0 \cdot |S_{1(x)} - y| + p_1 \cdot |S_{2(x)} - y| + p_2 \cdot |S_{3(x)} - y|) \tag{3}$$

Second, following ClassSR, an average-distribution loss encourages the three experts to be selected in equal proportions:

$$L_{avg} = |mean(p_0 - \tfrac{1}{3})| + |mean(p_1 - \tfrac{1}{3})| + |mean(p_2 - \tfrac{1}{3})| \tag{4}$$

with a sufficiently large weight ($\lambda_{avg} = 2.0$) so that it is not overwhelmed by the reconstruction term.

Third, a difficulty-supervised cross-entropy loss uses pseudo-labels derived from the true per-patch quality gains. For each patch, we measure how much its reconstruction improves as the network goes deeper. Let L_1, L_2 and L_3 denote the reconstruction losses at the shallow, medium and deep experts, respectively; we define the quality gains as:

$$d_{21} = L_1 - L_2, \quad d_{32} = L_2 - L_3 \tag{5}$$

A patch is assigned to the medium class if d_{21} exceeds the batch median, to the hard class if d_{32} exceeds the batch median, and to the simple class otherwise. This cross-entropy term prevents the classifier from collapsing to a trivial all-one-class solution, a failure mode we observed when only the soft average-distribution constraint was used. The overall objective of the joint stage is the sum of these three terms.

4. Experiments

4.1. Datasets

DIV2K [30] is a widely-used high-quality dataset for single image super-resolution, consisting of 1,000 2K-resolution images with diverse contents, of which 800 are provided for training. Following common practice, we use its 800 training images to train our models. The evaluation is conducted on four widely-used benchmark datasets: Set5 [31], Set14 [32], BSD100 [33] and Urban100 [34], which cover a wide range of contents, from a few images with smooth structures (Set5, Set14) to natural scenes (BSD100) and challenging urban images with rich repetitive textures (Urban100). All experiments are conducted for ×4 super-resolution, and the LR images are obtained by bicubic downsampling of the HR images.

4.2. Baselines

To demonstrate that the proposed framework is architecture-agnostic, we instantiate it on three representative super-resolution backbones spanning the main families of SR models:

-EDSR [3], a residual-block-stacked convolutional network;

-RCAN [4], a deep convolutional network with residual-in-residual structure and channel attention;

-SwinIR [6], a Transformer-based network built on residual Swin Transformer blocks.

These three backbones represent plain residual CNNs, attention-based CNNs and Transformers, respectively, allowing us to verify the generality of our method across fundamentally different architectures.

4.3. Implementation details

Network configuration. For each backbone we insert three experts at increasing depths. EDSR uses 64 feature channels with experts after 8, 16 and 32 residual layers. RCAN uses 64 channels with 10 residual channel-attention blocks per residual group, and experts after 3, 6 and 10 residual groups. SwinIR uses an embedding dimension of 60 with experts after 2, 4 and 6 residual Swin Transformer blocks. The lightweight classifier has a base width of 32 channels and 585,139 parameters (2.2 MB), and is shared in design across all three backbones.

Training settings. All models are trained on the 800 DIV2K training images and optimized with the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$). The backbone is first trained for 200 epochs with a cosine-annealed learning rate starting at 1×10^{-4} and a distillation weight $\lambda_d = 0.5$. The classifier is then jointly trained with the backbone for 100 epochs, using a learning rate of 1×10^{-4} for the classifier and 1×10^{-5} for the backbone, with the average-distribution weight $\lambda_{avg} = 2.0$. LR patches of size 32×32 are used during training, and standard data augmentation (random flipping) is applied. No weight decay is used. All experiments are conducted on a single NVIDIA RTX 5090 D GPU (32 GB) with PyTorch 2.8 and CUDA 12.8.

4.4. Loss functions and evaluation metrics

The training objective combines the four loss terms described in Section 3: the L1 reconstruction loss (L_{rec}), the intra-network distillation loss ($L_{distill}$), the average-distribution loss (L_{avg}) and the difficulty-supervised cross-entropy loss (L_{ce}). For evaluation, we report reconstruction quality using PSNR and SSIM [35] computed on the Y channel of the YCbCr colour space, and efficiency using the average routed depth, the number of Multiply-Accumulate Operations (MACs) and the number of parameters.

5. Results

5.1. Performance and efficiency gains

Tables 1–3 report, for each backbone, the PSNR/SSIM of the full-depth model (Expert3 for every patch) and of our DADE model with difficulty-aware routing, together with the corresponding computation saving. Across all three backbones and all four datasets, the routed model matches the full-depth model almost exactly: the average PSNR drop is only 0.071 dB for EDSR, 0.025 dB for RCAN and 0.120 dB for SwinIR, and the average SSIM drop never exceeds 0.001. On BSD100, the routed model even slightly surpasses the full-depth model for RCAN and SwinIR, since the balanced routing assigns some patches to a path better matched to their difficulty. Meanwhile, the routed model traverses on average only about 18.7 of the 32 equivalent layers, i.e. roughly 40% fewer, substantially reducing the computational cost with negligible loss in reconstruction quality. The corresponding qualitative comparisons for EDSR, RCAN, and SwinIR are shown in Figures 5, 6, and 7, respectively.

Table 1. Routing results on EDSR (x4). PSNR/SSIM of the full model vs. our model, and computation saving

Dataset	EDSR	EDSR-Ours	ΔPSNR	ΔSSIM	Comp. saving
Set5	31.88	31.81	-0.07	-0.0001	38.8%
Set14	28.41	28.35	-0.06	-0.0003	41.6%
BSD100	26.24	26.24	+0.00	+0.0000	41.7%
Urban100	25.71	25.56	-0.16	-0.0009	41.3%

Table 2. Routing results on RCAN (x4). PSNR/SSIM of the full model vs. our model, and computation saving

Dataset	RCAN	RCAN-Ours	ΔPSNR	ΔSSIM	Comp. saving
Set5	32.18	32.17	-0.01	-0.0000	37.7%
Set14	28.59	28.57	-0.02	-0.0001	42.8%
BSD100	26.28	26.29	+0.01	+0.0001	41.6%
Urban100	26.13	26.05	-0.08	-0.0003	41.0%

Table 3. Routing results on SwinIR (x4). PSNR/SSIM of the full model vs. our model, and computation saving

Dataset	SwinIR	SwinIR-Ours	ΔPSNR	ΔSSIM	Comp. saving
Set5	31.98	31.83	-0.14	-0.0003	38.5%
Set14	28.45	28.36	-0.10	-0.0005	41.8%
BSD100	26.26	26.28	+0.02	-0.0001	41.7%
Urban100	25.78	25.51	-0.26	-0.0020	41.3%

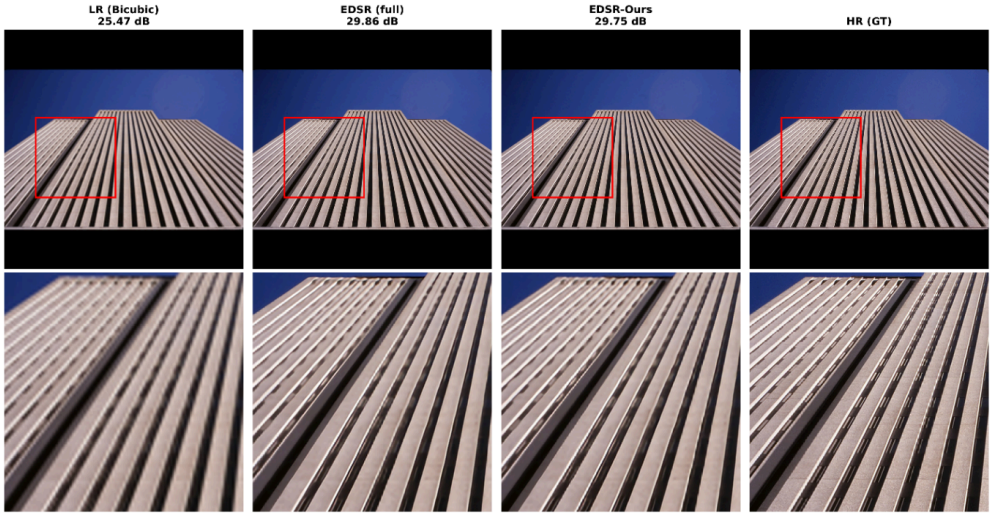

Figure 5. Qualitative comparison of EDSR on Urban100 (×4)

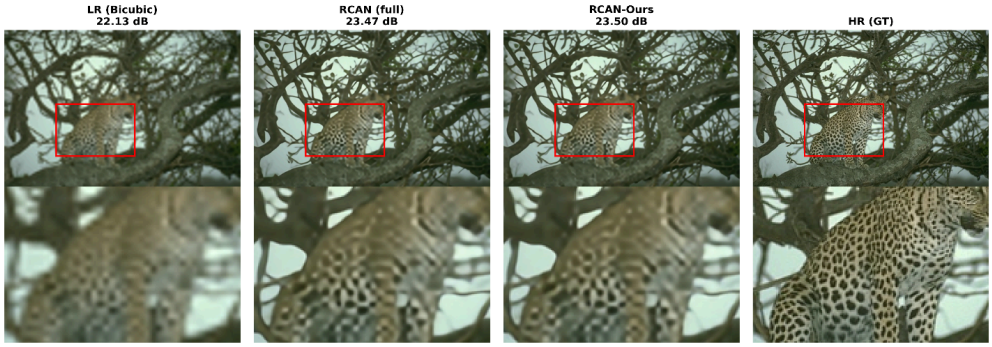

Figure 6. Qualitative comparison of RCAN on BSD100 (×4)

Figure 7. Qualitative comparison of SwinIR on Set14 (×4)

5.2. Comparison with existing methods

5.2.1. Comparison with lightweight SR methods

Table 4 compares our DADE model with several representative classical and lightweight SR methods, whose results are quoted from their original papers. It should be noted that these methods are trained until full convergence, whereas, due to our limited computational budget, our backbone is trained for far fewer epochs; the absolute PSNR is therefore reported here mainly for reference.

Even under this disadvantaged setting, our DADE model achieves competitive or superior performance. On Set5, Set14 and Urban100, it attains PSNR comparable to or higher than these methods, while requiring substantially fewer multiply-accumulate operations than deeper recursive networks such as DRCN [35] and DRRN [36]. More importantly, unlike the fixed-capacity networks in Table 4, our method offers two distinctive advantages: it stores far fewer parameters than the multi-model baseline (Section 5.2), and it allows the inference cost to be reduced on-the-fly by routing easy patches to shallow experts. This combination of parameter efficiency and adjustable computation, rather than absolute PSNR alone, is the key strength of the proposed framework.

Table 4. Quantitative comparison (×4) with existing SR methods

Method	Multi-Adds	Set5 (PSNR/SSIM)	Set14	BSD100	Urban100
SRCNN [1]	52.7G	30.48/0.8628	27.49/0.7503	26.90/0.7101	24.52/0.7221
VDSR [2]	612.6G	31.35/0.8838	28.01/0.7674	27.29/0.7251	25.18/0.7524
DRCN [35]	9788.7G	31.53/0.8854	28.02/0.7670	27.23/0.7233	25.14/0.7510

Table 4. Continued

LapSRN [37]	149.4G	31.54/0.8850	28.19/0.7720	27.32/0.7280	25.21/0.7560
DRRN [36]	6796.9G	31.68/0.8888	28.21/0.7720	27.38/0.7284	25.44/0.7638
EDSR-Ours†	124.1G	31.70/0.8885	28.35/0.7754	26.25/0.7060	25.55/0.7672

5.2.2. Comparison with ClassSR

The core advantage of our DADE design over ClassSR-style methods lies in parameter efficiency. ClassSR stores several independent sub-networks of different sizes, one for each difficulty level, so that its storage cost grows with the number of sub-networks. In contrast, DADE packs the shallow, medium and deep experts into a single shared backbone. To quantify this, Table 5 compares the total stored parameters of our framework (a single backbone plus the lightweight classifier) with a ClassSR-baseline, which estimates the storage cost of the multi-model strategy by maintaining three independent complete networks. Our design reduces the stored parameters by 60.7% for EDSR, 64.5% for RCAN and 57.7% for SwinIR, while achieving essentially the same reconstruction quality and computation saving as reported in Section 5.1. This confirms that replacing multiple independent networks with a single progressive backbone is an effective way to retain difficulty-aware acceleration at a much lower storage cost.

Table 5. Parameter comparison with the ClassSR-baseline

Backbone	ClassSR-Baseline	Ours	Param. saving
EDSR	9,880,803 (37.7 MB)	3,878,740 (14.8 MB)	60.7%
RCAN	26,450,763 (100.9 MB)	9,402,060 (35.9 MB)	64.5%
SwinIR	6,561,027 (25.0 MB)	2,772,148 (10.6 MB)	57.7%

5.3. Ablation study

5.3.1. Effectiveness of the proposed distillation framework

To validate the intra-network distillation, we compare the per-expert PSNR of each backbone trained with and without the distillation loss (i.e., $\lambda_d = 0.5$ vs. $\lambda_d = 0$), on Set5. Tables 6-8 report the results for EDSR, RCAN and SwinIR respectively.The benefit of distillation is closely related to the performance gap between the experts. For EDSR (Table 6), whose three experts differ the most (a gap of 0.72 dB between Expert 1 and Expert 3), distillation clearly improves all experts, and especially the shallow ones (+0.31, +0.25 and +0.14 dB for Expert 1, Expert 2 and Expert 3). For SwinIR (Table 8), the improvement is moderate (+0.14, +0.06 and −0.02 dB). For RCAN (Table 7), whose experts are already very close before distillation (only a 0.19 dB gap between Expert 1 and Expert 3), distillation yields only a marginal gain on Expert 1 (+0.05 dB) and even slightly lowers the two deeper experts (−0.02 and −0.04 dB).These observations indicate that distillation is most effective when the teacher (the deepest expert) is substantially stronger than the students (the shallow experts); when all experts are already comparable, there is little knowledge to transfer, and forcing distillation may slightly sacrifice the deepest expert. We therefore report this as an applicability boundary of the distillation component rather than a universally positive effect.

Figure 8 provides a visual comparison of the shallow expert (Expert 1) before and after distillation on an Urban100 image. Without distillation, Expert 1 produces blurry textures; with the deepest expert distilling its knowledge into Expert 1, the recovered stripes become noticeably sharper. This visually confirms the quantitative improvement reported in Table 6, where distillation raises the shallow-expert quality the most.

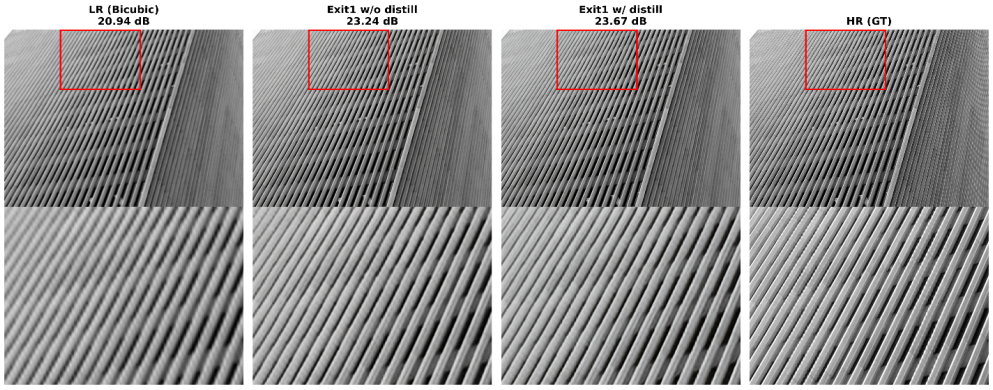

Figure 8. Effect of the proposed distillation on the shallow expert (EDSR, Urban100, ×4)

Table 6. Distillation ablation on EDSR (Set5, ×4)

Method	Expert1	Expert2	Expert3
w/o distillation	31.08	31.45	31.80
w/ distillation	31.39	31.70	31.94
Gain	+0.31	+0.25	+0.14

Table 7. Distillation ablation on RCAN (Set5, ×4)

Method	Expert1	Expert2	Expert3
w/o distillation	32.12	32.29	32.31
w/ distillation	32.17	32.27	32.27
Gain	+0.05	-0.02	-0.04

Table 8. Distillation ablation on SwinIR (Set5, ×4)

Method	Expert1	Expert2	Expert3
w/o distillation	31.35	31.82	31.92
w/ distillation	31.49	31.88	31.90
Gain	+0.14	+0.06	-0.02

5.3.2. Effectiveness of the proposed classifier and joint training

Classifier performance. The difficulty-aware classifier is trained on the 121,845 patches extracted from the 800 DIV2K training images, using pseudo-labels obtained by the three-way split on the per-patch quality gains $\Delta(2\text{-}1)$ and $\Delta(3\text{-}2)$ described in Section 3. It is lightweight, containing only 585,139 parameters (2.2 MB), which is negligible compared with the SR backbone. As summarized in Table 9, the classifier reaches a best validation accuracy of 75.92%, with per-class accuracies of 60%, 74% and 83% for the simple, medium and hard classes, respectively. The relatively low accuracy on the simple class is expected, as the boundary between simple and medium patches is intrinsically ambiguous. Importantly, this has little impact on the final quality: routing an easy patch to a slightly deeper expert only wastes a small amount of computation, so such errors are far less costly than mis-routing a hard patch to a shallow expert that cannot restore its details.

Table 9. Performance of the difficulty-aware classifier

Item	Value
Validation accuracy	75.92%
Per-class accuracy	Simple 60% / Medium 74% / Hard 83%

Effectiveness of joint training. To verify the benefit of jointly fine-tuning the backbone together with the classifier, we compare two settings that differ only in whether the backbone is updated:

• Separate: the classifier is trained on top of a fixed (frozen) backbone (CLS + fixed backbone);

• Joint (ours): the classifier and the backbone are optimized together (CLS + backbone + joint fine-tuning).

All other components—the classifier architecture, the losses and the training schedule—are kept identical, so that the only difference is whether the backbone is updated. Table 10 reports the results on EDSR, averaged over the four benchmark datasets. Joint training is superior on both axes: it achieves a smaller average PSNR drop (−0.071 vs. −0.093 dB) and a larger computation saving (40.8% vs. 36.0%). Furthermore, joint training yields a near-perfect balanced routing (about 33% per class), whereas the separately-trained classifier is biased towards the medium expert (46–54%) and rarely selects the simple expert (12–19%).

The underlying reason is that a frozen backbone forces the classifier to be conservative: since the shallow experts are not optimized for routing, sending patches to them would degrade quality, so the classifier prefers the safer medium expert. In contrast, joint fine-tuning allows the backbone to adapt its shallow experts to the routing decisions, so that more patches can be safely handled by the simple expert. As a result, joint training improves both reconstruction quality and computation saving simultaneously, which explains its consistent advantage in Table 10.

Table 10. Joint vs. separate training on EDSR (×4, averaged over four datasets)

Method	Set5	Set14	BSD100	Urban100	Comp. saving	Routing (S / M / H)
Separate (frozen backbone)	31.80	28.33	26.25	25.51	36.0%	12-19% / 46-54% / ~33%
Joint training (ours)	31.81	28.35	26.24	25.56	40.8%	~33% / ~33% / ~33%

6. Conclusion

We presented a distillation-enhanced progressive framework with difficulty-aware routing for efficient single image super-resolution. By replacing the multiple independent sub-networks of ClassSR-style methods with a single backbone that has multiple depth experts, our method retains difficulty-aware acceleration while dramatically reducing the stored parameters. An intra-network distillation scheme improves the early experts, and a joint training procedure with an average-distribution constraint and a difficulty-supervised loss produces a balanced, effective routing. Instantiated on three representative backbones - EDSR, RCAN and SwinIR - the framework consistently reduces computation by about 40% in routed depth (21-33% in MACs) with negligible PSNR/SSIM loss, and reduces stored parameters by 57.7-64.5% relative to the independent multi-model baseline. These results show that the proposed strategy is effective and, importantly, generalizes across CNN and Transformer architectures. Future work includes matched comparisons with lightweight baselines, qualitative studies, adaptive routing thresholds, and extension to other restoration tasks.

References

[1] Dong, C., Loy, C. C., He, K., & Tang, X. (2014). Learning a deep convolutional network for image super-resolution. In D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *Computer Vision – ECCV 2014 (Lecture Notes in Computer Science, Vol. 8692)*, 184–199. Springer. https://doi.org/10.1007/978-3-319-10593-2_13

[2] Kim, J., Lee, J. K., & Lee, K. M. (2016). Accurate image super-resolution using very deep convolutional networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1646–1654. IEEE. https://doi.org/10.1109/CVPR.2016.182

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

[4] Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., & Fu, Y. (2018). Image super-resolution using very deep residual channel attention networks. In *Computer Vision – ECCV 2018: 15th European Conference (Part VII)*, 294–310. Springer. https://doi.org/10.1007/978-3-030-01234-2_18

[5] Zhang, Y., Tian, Y., Kong, Y., Zhong, B., & Fu, Y. (2018). Residual dense network for image super-resolution. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2472–2481. IEEE. https://doi.org/10.1109/CVPR.2018.00262

[6] Liang, J., Cao, J., Sun, G., Zhang, K., Van Gool, L., & Timofte, R. (2021). *SwinIR: Image restoration using Swin Transformer*. arXiv. https://doi.org/10.48550/arXiv.2108.10257

[7] Chen, X., Wang, X., Zhou, J., Qiao, Y., & Dong, C. (2023). Activating more pixels in image super-resolution transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 22367–22377). IEEE. https://doi.org/10.1109/CVPR52729.2023.02142

[8] Kong, X., Zhao, H., Qiao, Y., & Dong, C. (2021). ClassSR: A general framework to accelerate super-resolution networks by data characteristic. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 12016–12025). IEEE. https://doi.org/10.1109/CVPR46437.2021.01184

[9] Chen, B., Lin, M., Sheng, K., Zhang, M., Chen, P., Li, K., Cao, L., & Ji, R. (2022). ARM: Any-time super-resolution method. In S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, & T. Hassner (Eds.), *Computer Vision – ECCV 2022* (pp. 254–270). Springer Nature Switzerland. https://doi.org/10.1007/978-3-031-19800-7_15

[10] Jeong, J., Kim, J., Jo, Y., & Kim, S. J. (2024). Accelerating image super-resolution networks with pixel-level classification. In *Computer Vision – ECCV 2024: 18th European Conference (Part III*, pp. 236–251). Springer. https://doi.org/10.1007/978-3-031-72646-0_14

[11] Dong, C., Loy, C. C., & Tang, X. (2016). Accelerating the super-resolution convolutional neural network. In B. Leibe, J. Matas, N. Sebe, & M. Welling (Eds.), *Computer Vision – ECCV 2016 (Lecture Notes in Computer Science, Vol. 9906*, pp. 391–407). Springer. https://doi.org/10.1007/978-3-319-46475-6_25

[12] Wang, Z., Chen, J., & Hoi, S. C. H. (2021). Deep learning for image super-resolution: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 43*(10), 3365–3387. https://doi.org/10.1109/TPAMI.2020.2982166

[13] Ledig, C., Theis, L., Huszár, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., & Shi, W. (2017). Photo-realistic single image super-resolution using a generative adversarial network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 105–114). IEEE. https://doi.org/10.1109/CVPR.2017.19

[14] Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., & Loy, C. C. (2019). ESRGAN: Enhanced super-resolution generative adversarial networks. In L. Leal-Taixé & S. Roth (Eds.), *Computer Vision – ECCV 2018 Workshops* (pp. 63–79). Springer International Publishing. https://doi.org/10.1007/978-3-030-11021-5_5

[15] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin Transformer: Hierarchical vision transformer using shifted windows. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (pp. 10012–10022). IEEE. https://doi.org/10.1109/ICCV48922.2021.00986

[16] Zamir, S. W., Arora, A., Khan, S., Hayat, M., Khan, F. S., & Yang, M.-H. (2022). Restormer: Efficient transformer for high-resolution image restoration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 5728–5739). IEEE. https://doi.org/10.1109/CVPR52688.2022.00565

[17] Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., & Li, H. (2022). Uformer: A general U-shaped transformer for image restoration. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (pp. 17662–17672). IEEE. https://doi.org/10.1109/CVPR52688.2022.01716

[18] Li, J., Pei, Z., Li, W., Gao, G., Wang, L., Wang, Y., & Zeng, T. (2024). A systematic survey of deep learning-based single-image super-resolution. *ACM Computing Surveys, 56*(10), Article 249. https://doi.org/10.1145/3659100

[19] Gendy, G., He, G., & Sabor, N. (2023). Lightweight image super-resolution based on deep learning: State-of-the-art and future directions. *Information Fusion, 94*, 284–310. https://doi.org/10.1016/j.inffus.2023.01.024

[20] Zhang, Z., Han, Y., Zhu, L., Xi, X., Li, L., Liu, M., Tan, S., & Yan, B. (2024). Network architecture for single image super-resolution: A comprehensive review and comparison. *IET Image Processing, 18*, 2215–2243. https://doi.org/10.1049/ipr2.13100

[21] Ahn, N., Kang, B., & Sohn, K. A. (2018). Fast, accurate, and lightweight super-resolution with cascading residual network. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), *Computer Vision – ECCV 2018 (Lecture Notes in Computer Science, Vol. 11214)*, 252–268. Springer. https://doi.org/10.1007/978-3-030-01249-6_16

[22] Hui, Z., Gao, X., Yang, Y., & Wang, X. (2019). Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia*, 2024–2032. Association for Computing Machinery. https://doi.org/10.1145/3343031.3351084

[23] Kong, F., Li, M., Liu, S., Liu, D., He, J., Bai, Y., Chen, F., & Fu, L. (2022). Residual local feature network for efficient super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 766–776. IEEE.

[24] Gao, G., Li, W., Li, J., Wu, F., Lu, H., & Yu, Y. (2022). Feature distillation interaction weighting network for lightweight image super-resolution. *Proceedings of the AAAI Conference on Artificial Intelligence, 36*(1), 661–669. https: //doi.org/10.1609/aaai.v36i1.19946

[25] Hinton, G., Vinyals, O., & Dean, J. (2015). *Distilling the knowledge in a neural network*. arXiv preprint arXiv:1503.02531. https://doi.org/10.48550/arXiv.1503.02531

[26] He, Z., Dai, T., Lu, J., Jiang, Y., & Xia, S.-T. (2020). FAKD: Feature-affinity based knowledge distillation for efficient image super-resolution. In *2020 IEEE International Conference on Image Processing (ICIP)* (pp. 518–522). IEEE. https://doi.org/10.1109/ICIP40778.2020.9190917

[27] Lee, W., Lee, J., Kim, D., & Ham, B. (2020). Learning with privileged information for efficient image super-resolution. In *Computer Vision – ECCV 2020* (Vol. 12373, pp. 465–482). Springer International Publishing.

[28] Hong, C., Baik, S., Kim, H., Nah, S., & Lee, K. M. (2022). CADyQ: Content-aware dynamic quantization for image super-resolution. In *Computer Vision – ECCV 2022* (Vol. 13691, pp. 367–383). Springer Nature Switzerland.

[29] Yu, L., Li, X., Li, Y., Jiang, T., Wu, Q., Fan, H., & Liu, S. (2023). DIPNet: Efficiency distillation and iterative pruning for image super-resolution. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)* (pp. 1692–1701). IEEE. https://doi.org/10.1109/CVPRW59228.2023.00170

[30] Agustsson, E., & Timofte, R. (2017). NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops* (pp. 126–135).

[31] Bevilacqua, M., Roumy, A., Guillemot, C., & Alberi-Morel, M.-L. (2012). Low-complexity single-image super-resolution based on nonnegative neighbor embedding. In *Proceedings of the British Machine Vision Conference (BMVC 2012)* (pp. 1–12). BMVA Press. https://doi.org/10.5244/C.26.135

[32] Zeyde, R., Elad, M., & Protter, M. (2012). On single image scale-up using sparse-representations. In J. D. Boissonnat, P. Chenin, A. Cohen, C. Gout, T. Lyche, M. L. Mazure, & L. Schumaker (Eds.), *Curves and surfaces (Lecture Notes in Computer Science, Vol. 6920)* (pp. 711–730). Springer. https://doi.org/10.1007/978-3-642-27413-8_47

[33] Martin, D., Fowlkes, C., Tal, D., & Malik, J. (2001). 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) (Vol. 2)* (pp. 416–423). IEEE. https://doi.org/10.1109/ICCV.2001.937655

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

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

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

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