The Ninth NTIRE 2024 Efficient Super-Resolution Challenge Report

Bin Ren* Yawei Li* Nancy Mehta* Radu Timofte* Hongyuan Yu Cheng Wan
Yuxin Hong Bingnan Han Zhuoyuan Wu Yajun Zou Yuqing Liu Jizhe Li
Keji He Chao Fan Heng Zhang Xiaolin Zhang Xuanwu Yin Kunlong Zuo
Bohao Liao Peizhe Xia Long Peng Zhibo Du Xin Di Wangkai Li
Yang Wang Wei Zhai Renjing Pei Jiaming Guo Songcen Xu Yang Cao
Zhengjun Zha Yan Wang Yi Liu Qing Wang Gang Zhang Liou Zhang
Shijie Zhao Long Sun Jinshan Pan Jiangxin Dong Jinhui Tang Xin Liu
Min Yan Qian Wang Menghan Zhou Yiqiang Yan Yixuan Liu Wensong Chan
Dehua Tang Dong Zhou Li Wang Lu Tian Barsoum Emad Bohan Jia
Junbo Qiao Yunshuai Zhou Yun Zhang Wei Li Shaohui Lin Shenglong Zhou
Binbin Chen Jincheng Liao Suiyi Zhao Zhao Zhang Bo Wang Yan Luo
Yanyan Wei Feng Li Mingshen Wang Yawei Li Jinhan Guan Dehua Hu
Jiawei Yu Qisheng Xu Tao Sun Long Lan Kele Xu Xin Lin Jingtong Yue
Lehan Yang Shiyi Du Lu Qi Chao Ren Zeyu Han Yuhan Wang
Chaolin Chen Haobo Li Mingjun Zheng Zhongbao Yang Lianhong Song
Xingzhuo Yan Minghan Fu Jingyi Zhang Baiang Li Qi Zhu Xiaogang Xu
Dan Guo Chunle Guo Jiadi Chen Huanhuan Long Chunjiang Duanmu
Xiaoyan Lei Jie Liu Weilin Jia Weifeng Cao Wenlong Zhang Yanyu Mao
Ruilong Guo Nihao Zhang Qian Wang Manoj Pandey Maksym Chernozhukov
Giang Le Shuli Cheng Hongyuan Wang Ziyan Wei Qingting Tang
Liejun Wang Yongming Li Yanhui Guo Hao Xu Akram Khatami-Rizi
Ahmad Mahmoudi-Aznaveh Chih-Chung Hsu Chia-Ming Lee Yi-Shiuan Chou
Amogh Joshi Nikhil Akalwadi Sampada Malagi Palani Yashaswini Chaitra Desai
Ramesh Ashok Tabib Ujwala Patil Uma Mudenagudi

Abstract

*This paper provides a comprehensive review of the NTIRE 2024 challenge, focusing on efficient single-image super-resolution (ESR) solutions and their outcomes. The task of this challenge is to super-resolve an input image with a magnification factor of $\times 4$ based on pairs of low and corresponding high-resolution images. The primary objective is to develop networks that optimize various aspects such as runtime, parameters, and FLOPs, while still maintaining a peak signal-to-noise ratio (PSNR) of approximately 26.90 dB on the DIV2K_LSDIR_valid dataset and 26.99 dB on the DIV2K_LSDIR_test dataset. In addition, this challenge has 4 tracks including the main track (overall performance), sub-track 1 (runtime), sub-track 2 (FLOPs), and sub-track 3 (parameters). In the main track, all three metrics (*i.e.*, runtime, FLOPs, and parameter count) were considered. The ranking of the main track is calculated based on a weighted sum-up of the scores of all other sub-tracks. In sub-track 1, the practical runtime performance of the submissions was evaluated, and the corresponding score was used to determine the ranking. In sub-track 2, the number of FLOPs was considered. The score calculated based on the corresponding FLOPs was used to determine the ranking. In sub-track 3, the number of parameters was considered. The score calculated based on the corresponding param-*

* B Ren (bin.ren@unitn.it, University of Pisa & University of Trento, Italy), Y. Li (yawei.li@vision.ee.ethz.ch, ETH Zürich, Switzerland), N. Mehta (nancy.mehta@uni-wuerzburg.de, University of Würzburg, Germany), and R. Timofte (Radu.Timofte@uni-wuerzburg.de, University of Würzburg, Germany) were the challenge organizers, while the other authors participated in the challenge.
Appendix A contains the authors' teams and affiliations.
NTIRE 2024 webpage: https://cvslai.net/ntire/2024/.
Code: https://github.com/Amazingren/NTIRE2024_ESR/.

eters was used to determine the ranking. RLFN is set as the baseline for efficiency measurement. The challenge had 262 registered participants, and 34 teams made valid submissions. They gauge the state-of-the-art in efficient single-image super-resolution. To facilitate the reproducibility of the challenge and enable other researchers to build upon these findings, the code and the pre-trained model of validated solutions are made publicly available at https://github.com/Amazingren/NTIRE2024_ESR/.

1. Introduction

Single image super-resolution (SR) aims at enhancing the resolution of low-resolution (LR) images to generate high-resolution (HR) counterparts. Typically, LR images are acquired through a degradation process that involves blurring and down-sampling. Among the models used to simulate this degradation in classical image SR, bicubic down-sampling stands out as widely adopted [45, 49, 63, 64]. Its prevalence as a benchmark enables the evaluation of different SR methods and facilitates direct comparisons between them, thereby validating the efficacy of novel SR methods.

Cutting-edge deep neural networks for SR face significant challenges, including parameter overparameterization, resource-intensive computation, and substantial latency. These obstacles hinder their integration into mobile devices for real-time SR applications. However, ongoing innovation continues to address these challenges. Enter a diverse array of research efforts focused on improving the efficiency of various architectures, including Convolutional Neural Networks (CNNs), Multi-Layer Perceptrons (MLPs), Transformers, and Mamba [6, 21, 29, 37, 68, 72, 86]. From the meticulous approach of network pruning [43, 58] to the straightforward technique of low-rank filter decomposition, and from the systematic process of network quantization to the advanced methods of neural architecture search [44, 91, 92], a range of effective solutions have emerged. Among these, knowledge distillation stands out as particularly promising. These efforts in network compression mark a significant advancement for image SR, offering not only improved resolution but also enhanced efficiency and accessibility for all.

The efficiency of a deep neural network encompasses various dimensions, evaluated across a range of metrics such as runtime, parameter count, and computational complexity (measured in FLOPs). These metrics play a crucial role in determining the network's feasibility for deployment across diverse platforms. Among these, runtime emerges as particularly significant, providing a direct indication of a network's operational efficiency and often serving as the primary criterion for evaluation. Of utmost concern is the relationship between computational complexity and energy consumption, a pivotal axis that directly impacts the viability of mobile devices. Higher computational complexity correlates with increased energy consumption, posing a significant threat to the delicate balance of battery life. Additionally, the number of parameters exerts a substantial influence on AI chip design, determining chip area and manufacturing costs. An increase in parameter counts can lead to larger chip sizes and elevated production expenses, thereby shaping the landscape of the AI device market.

In partnership with the 2024 New Trends in Image Restoration and Enhancement (NTIRE 2024) workshop, we are proud to announce the inception of the Efficient Super-Resolution Challenge. The challenge's primary objective is to achieve super-resolution of a low-resolution (LR) image with a magnification factor of $\times 4$, employing a network that optimizes runtime, parameters, and FLOPs, building upon the baseline method laid by RLFN [36]. Participants are tasked with maintaining a minimum peak signal-to-noise ratio (PSNR) of 26.90 dB on the DIV2K_LSDIR_valid dataset and 26.99 dB on the DIV2K_LSDIR_test dataset. This challenge serves as a platform for exploring state-of-the-art solutions in efficient super-resolution. We aim to rigorously assess their effectiveness and identify key trends in the design of efficient SR networks. We welcome participants to contribute to this endeavor by pushing the boundaries of innovation and advancing streamlined and effective image restoration and enhancement techniques.

This challenge is one of the NTIRE 2024 Workshop associated challenges on: dense and non-homogeneous dehazing [3], night photography rendering [4], blind compressed image enhancement [83], shadow removal [73], efficient super resolution (this challenge), image super resolution ($\times 4$) [12], light field image super-resolution [80], stereo image super-resolution [76], HR depth from images of specular and transparent surfaces [85], bracketing image restoration and enhancement [89], portrait quality assessment [7], quality assessment for AI-generated content [56], restore any image model (RAIM) in the wild [50], RAW image super-resolution [17], short-form UGC video quality assessment [42], low light enhancement [57], and RAW burst alignment and ISP challenge.

2. NTIRE 2024 Efficient Super-Resolution Challenge

The objectives of this challenge are multifaceted: (1) To stimulate curiosity and exploration within the field of efficient super-resolution. (2) To create a platform where diverse methodologies can be directly compared in terms of efficiency. (3) And to act as a dynamic hub where academic and industrial leaders come together, exchanging ideas and laying the groundwork for potential collaborations. This section will elucidate the intricate details of the challenge,

https://cvlai.net/ntire/2024/

providing participants with clear guidance through its complex structure and objectives.

2.1. Dataset

The DIV2K [1] dataset and the LSDIR [46] dataset are utilized for this challenge. Specifically, the DIV2K dataset consists of 1,000 diverse 2K resolution RGB images, which are split into a training set of 800 images, a validation set of 100 images, and a test set of 100 images. The LSDIR dataset contains 86,991 high-resolution high-quality images, which are split into a training set of 84,991 images, a validation set of 1,000 images, and a test set of 1,000 images. In this challenge, the corresponding LR DIV2K and LSDIR images are generated by bicubic downsampling with a down-scaling factor of $\times 4$. The training images from DIV2K and LSDIR are provided to the participants of the challenge. During the validation phase, 100 images from the DIV2K validation set and 100 images from the LSDIR validation set, forming the DIV2K_LSDIR_valid set, which is made available to participants. During the test phase, 100 images from the DIV2K test set and another 100 images from the LSDIR test set are used, forming the DIV2K_LSDIR_test set. Throughout the entire challenge, the testing HR images remain hidden from the participants.

2.2. RLFN Baseline Model

The Residual Feature Distillation Network (RLFN) [36] serves as the baseline model in this challenge. The aim is to improve its efficiency in terms of runtime, number of parameters, and FLOPs, while at least maintaining 26.90 dB on the DIV2K_LSDIR_valid dataset and 26.99 dB on the DIV2K_LSDIR_test dataset.

The main idea within RLFN is the use of three convolutional layers for residual local feature learning to simplify feature aggregation, which achieves a good trade-off between model performance and inference time. Moreover, the popular contrastive loss was explored and RLFN proposed that the selection of intermediate features of its feature extractor has a great influence on the overall performance. Specifically, the initial feature extraction is carried out by a 3×3 convolution that generates coarse features from the input LR image. The second part of RLFB consists of four RLFBs, stacked in a chain-like manner, to progressively refine the extracted features. After gradual refinement by the RLFBs, all intermediate features are combined using a 1×1 convolution layer. An additional 3×3 convolution layer is then utilized to smooth the aggregated features. Finally, the super-resolved images are generated by pixel shuffle operation.

The baseline RLFN emerges as the winner of the NTIRE2022 Challenge on Efficient Super-Resolution [36]. The quantitative performance and efficiency metrics of RLFN are given in Table 1, and summarized as follows: (1) The number of parameters is 0.317M. (2) The average PSNRs on validation (DIV2K 100 valid images and LSDIR 100 valid images) and testing (DIV2K 100 test images and LSDIR 100 test images) sets of this challenge are 29.96 dB and 27.07 dB, respectively. (3) The runtime averaged 11.77 ms on the validation and test set with PyTorch 1.13.1+cu117 on a single NVIDIA GeForce RTX 3090 GPU. (4) The number of FLOPs for an input of size 256×256 is 19.67G.

2.3. Tracks and Competition

This challenge aims to devise a network that reduces one or several aspects such as runtime, parameters, and FLOPs, while at least maintaining the 26.90 dB on the DIV2K_LSDIR_valid dataset, and 26.99 dB on the DIV2K_LSDIR_test dataset with a common GPU (*i.e.,* NVIDIA GeForce RTX 3090 GPU).

Main Track: Overall Performance. The aim is to obtain a network design/solution with the best overall performance in terms of inference runtime, FLOPS, and parameters on a common GPU while being constrained to maintain or improve the threshold PSNR results.

Sub-Track 1: Runtime Performance. The aim is to obtain a network design/solution with the lowest inference time (runtime) on a common GPU while being constrained to maintain or improve over the baseline method RLFN in terms of number of parameters, FLOPs, and the threshold PSNR result.

Sub-Track 2: FLOPs Performance. The aim is to obtain a network design/solution with the lowest amount of FLOPs on a common GPU while being constrained to maintain or improve the inference runtime, the parameters, and the threshold PSNR results.

Sub-Track 3: Parameters Performance. The aim is to obtain a network design/solution with the lowest amount of parameters on a common GPU while being constrained to maintain the FLOPs, the inference time (runtime), and the threshold PSNR results.

Challenge phases: *(1) Development and validation phase*: Participants were given access to 800 LR/HR training image pairs and 200 LR/HR validation image pairs from the DIV2K and the LSDIR datasets. Additional 84,991 LR/HR training image pairs from the LSDIR dataset are also provided to the participants. The RLFN model, pretrained parameters, and validation demo script are available on GitHub https://github.com/Amazingren/NTIRE2024_ESR, allowing participants to benchmark their models' runtime on their systems. Participants could upload their HR validation results to the evaluation server to calculate the PSNR of the super-resolved image produced by their models and receive immediate feedback. The corresponding number of parameters, FLOPs, and runtime will

be computed by the participants. *(2) Testing phase*: In the final testing phase, participants were granted access to 100 LR testing images from DIV2K and 100 LR testing images from LSDIR, while the HR ground-truth images remained hidden. Participants submitted their super-resolved results to the Codalab evaluation server and emailed the code and factsheet to the organizers. The organizers verified and ran the provided code to obtain the final results, which were then shared with participants at the end of the challenge.

Evaluation protocol: Quantitative evaluation metrics included validation and testing PSNRs, runtime, FLOPs, and the number of parameters during inference. PSNR was measured by discarding a 4-pixel boundary around the images. The average runtime during inference was computed on the 200 LR validation images and the 200 LR testing images. The average runtime on the validation and testing sets served as the final runtime indicator. FLOPs are evaluated on an input image of size 256×256. Among these metrics, runtime was considered the most important. Participants were required to maintain a PSNR of at least 26.90 dB on the DIV2K_LSDIR valid dataset, and 26.99 dB on the DIV2K_LSDIR test dataset during the challenge. The constraint on the testing set helped prevent overfitting on the validation set. It's important to highlight that methods with a PSNR below the specified threshold (*i.e.,* , 26.90 dB on DIV2K_LSDIR_valid and, 26.99 dB on DIV2K_LSDIR_test) will not be considered for the subsequent ranking process. It is essential to meet the minimum PSNR requirement to be eligible for further evaluation and ranking. A code example for calculating these metrics is available at https://github.com/Amazingren/NTIRE2024_ESR.

To better quantify the rankings, we have designed a scoring function for three evaluation metrics in this challenge: runtime, FLOPs, and parameters. This scoring aims to convert the performance of each metric into corresponding scores to make the rankings more significant. Especially, the score for each separate metric (*i.e.,* Runtime, FLOPs, and parameter) for each sub-track is calculated as:

$$Score\_Metric = \frac{exp(2 \times Metric_{TeamX})}{Metric_{Baseline}}, \quad (1)$$

based on the score of each metric, the final score used for the main track is calculated as:

$$\begin{aligned} Score\_Final = {} & w_1 \times Score\_Runtime \\ & + w_2 \times Score\_FLOPs \\ & + w_3 \times Score\_Params, \end{aligned} \quad (2)$$

where w_1, w_2, and w_3 are set to 0.7, 0.15, and 0.15, respectively. This setting is intended to incentivize participants to design a method that prioritizes speed efficiency while maintaining a reasonable model complexity.

3. Challenge Results

The final test results and rankings are presented in Table 1. The table also includes the baseline method RLFN [36] for comparison. In Sec.4, the methods evaluated in Table 1 are briefly explained, while the team members are listed in A. The performance of different methods is compared from four different perspectives including the runtime, FLOPs, the parameters, and the overall performance. Furthermore, in order to promote a fair competition emphasizing efficiency, the criteria for image reconstruction quality in terms of test PSNR are set to 26.90 and 26.99 on the DIV2K_LSDIR_valid and DIV2K_LSDIR_test sets, resp.

Runtime. In this challenge, runtime stands as the paramount evaluation metric. XiaomiMM's solution emerges as the frontrunner with the shortest runtime among all entries in the efficient SR challenge, securing the top position. Following closely, the Cao Group and BSR claim the second and third spots, respectively. Remarkably, the average runtime of the top three solutions on both the validation and test sets remains below 10 ms. Impressively, the first 15 teams present solutions with an average runtime below 13 ms, showcasing a continuous enhancement in the efficiency of image SR networks. Despite the slight differences in runtime among the top three teams, the challenge retains its competitive edge. Furthermore, the XiaomiMM team achieves the highest PSNR on both the validation and test sets among the top three teams.

FLOPs. FLOPs, representing the number of floating-point operations, serve as a critical metric for assessing model complexity. In this sub-track, PiXupt secures the top position, followed by XJU_100 th Ann and VPEG_C in second and third places, respectively. Remarkably, the disparity among the top three methods is minimal, underscoring their competitiveness and proficiency in managing model complexity. However, it's noteworthy that both PiXupt and XJU_100th Ann exhibit relatively high runtimes. Further exploration is warranted to address and mitigate such challenges.

Parameters. Parameters serve as another critical metric for assessing model complexity, which was also evaluated in this challenge. As shown in Table 1, XJU_100th Ann, VPEG_C, and ZHEstar secured the first three places. However, akin to the FLOPs sub-track, the runtime of these methods lags significantly behind that of the other methods. Further exploration is warranted to address and mitigate such challenges.

Overall evaluation. In the final assessment, performance is meticulously evaluated based on an aggregate metric that intricately weaves together runtime, FLOPs, and the number of parameters. Notably, the XiaomiMM Group emerges triumphant, securing the coveted top spot under this comprehensive metric, with the Cao Group and BSR clinching the 2nd and 3rd places, respectively. This outcome un-

Table 1. Results of Ninth NTIRE 2024 Efficient SR Challenge. The performance of the solutions is compared thoroughly from three perspectives including the runtime, FLOPs, and the number of parameters. The underscript numbers associated with each metric score denote the ranking of the solution in terms of that metric. For runtime, "Val." is the runtime averaged on DIV2K_LSDIR_valid validation set. "Test" is the runtime averaged on a test set with 200 images from DIV2K_LSDIR_test set, respectively. "Ave." is averaged on the validation and test datasets. "#Params" is the total number of parameters of a model. "FLOPs" denotes the floating point operations. Main Track combines all three evaluation metrics. The ranking for the main track is based on the score calculated via Eq. 2, and the ranking for other sub-tracks is based on the score of each metric score via Eq. 1. Please note that **this is not a challenge for PSNR improvement. The "validation/testing PSNR" is not ranked. For all the scores, the lower, the better**.

Teams	PSNR [dB]		Runtime [ms]			FLOPs	#Params	Sub-Track Scores			Main-Track	
	Val.	Test	Val.	Test	Ave.	[G]	[M]	Runtime	FLOPs	#Params	Score	Ranking
XiaomiMM	26.94	27.01	5.62	5.57	5.59	0.151	9.83	$2.59_{(1)}$	$2.59_{(7)}$	$2.72_{(8)}$	2.61	1
Cao Group	26.90	27.00	10.99	5.76	8.37	0.215	13.05	$4.15_{(2)}$	$3.88_{(13)}$	$3.77_{(13)}$	4.05	2
BSR	26.90	27.00	11.96	6.80	9.38	0.218	11.95	$4.93_{(3)}$	$3.96_{(14)}$	$3.37_{(10)}$	4.55	3
VPEG_O	26.90	27.01	12.20	7.06	9.63	0.212	13.86	$5.14_{(4)}$	$3.81_{(12)}$	$4.09_{(14)}$	4.78	4
CMVG	26.90	27.01	12.58	7.46	10.02	0.202	12.17	$5.49_{(5)}$	$3.58_{(10)}$	$3.45_{(12)}$	4.90	5
LeESR	26.91	27.02	13.21	8.04	10.62	0.165	9.75	$6.08_{(7)}$	$2.83_{(9)}$	$2.69_{(6)}$	5.09	6
AdvancedSR	26.91	27.02	12.94	8.04	10.49	0.263	16.20	$5.94_{(6)}$	$5.26_{(17)}$	$5.19_{(17)}$	5.73	7
ECNU_MViC	26.90	27.00	14.10	8.87	11.49	0.163	9.78	$7.05_{(10)}$	$2.80_{(8)}$	$2.70_{(7)}$	5.76	8
HiSR	26.91	27.03	13.69	8.83	11.26	0.208	11.99	$6.78_{(9)}$	$3.71_{(11)}$	$3.38_{(11)}$	5.81	9
MViC_SR	26.90	27.00	14.53	9.23	11.88	0.138	8.16	$7.53_{(11)}$	$2.39_{(6)}$	$2.29_{(5)}$	5.97	10
LVTeam	26.91	27.02	13.79	8.71	11.25	0.266	16.34	$6.77_{(8)}$	$5.36_{(18)}$	$5.27_{(18)}$	6.33	11
Fresh	26.90	27.00	14.52	9.29	11.90	0.245	14.97	$7.56_{(12)}$	$4.69_{(16)}$	$4.58_{(16)}$	6.68	12
Lanzhi	26.93	27.02	14.86	9.53	12.19	0.318	19.70	$7.94_{(13)}$	$7.44_{(20)}$	$7.41_{(20)}$	7.79	13
Supersr	26.90	27.01	15.16	9.90	12.53	0.298	18.67	$8.40_{(15)}$	$6.55_{(19)}$	$6.67_{(19)}$	7.87	14
MeowMeowMeow	26.92	27.03	16.18	10.95	13.57	0.238	14.47	$10.03_{(16)}$	$4.49_{(15)}$	$4.35_{(15)}$	8.35	15
Just Try	26.90	27.00	15.23	9.75	12.49	0.380	24.81	$8.35_{(14)}$	$11.00_{(22)}$	$12.46_{(23)}$	9.36	16
VPEG_C	26.90	27.03	18.76	13.31	16.03	0.084	4.97	$15.24_{(17)}$	$1.70_{(3)}$	$1.66_{(2)}$	11.17	17
VPEG_E	26.90	27.01	21.21	15.74	18.48	0.093	5.89	$23.09_{(20)}$	$1.80_{(5)}$	$1.82_{(4)}$	16.71	18
BU-ESR	27.00	27.11	19.47	14.07	16.77	0.433	27.05	$17.28_{(18)}$	$15.36_{(24)}$	$15.65_{(24)}$	16.75	19
Lasagna	26.90	27.00	20.11	14.59	17.35	0.657	41.21	$19.08_{(19)}$	$63.12_{(25)}$	$66.03_{(26)}$	32.73	20
ZHEstar	26.93	27.04	31.07	24.66	27.87	0.090	5.81	$113.87_{(21)}$	$1.76_{(4)}$	$1.81_{(3)}$	80.25	21
BlingBling	26.93	27.04	40.75	34.63	37.69	0.424	20.17	$604.72_{(22)}$	$14.51_{(23)}$	$7.77_{(21)}$	426.64	22
PiXupt	26.91	27.00	52.91	44.21	48.56	0.060	9.84	$383.46e1_{(23)}$	$1.46_{(1)}$	$2.72_{(9)}$	268.49e1	23
Minimalist	26.91	27.02	53.37	47.02	50.20	0.346	20.65	$506.53e1_{(24)}$	$8.87_{(21)}$	$8.16_{(22)}$	354.82e1	24
XJU_100th Ann	26.90	27.02	61.89	55.78	58.84	0.069	4.39	$219.74e2_{(25)}$	$1.55_{(2)}$	$1.56_{(1)}$	153.82e2	25
MagicSR	27.03	27.17	461.32	443.38	452.35	1.019	38.47	$241.07e31_{(26)}$	$619.57_{(26)}$	$49.98_{(25)}$	168.75e31	26
The following methods are not ranked since their validation/testing PSNR (underlined) is not on par with the threshold.												
FireWork	<u>26.86</u>	<u>26.96</u>	13.26	7.99	10.62	0.441	28.85	6.08	16.16	18.79	9.50	
DIRN	<u>26.22</u>	<u>26.33</u>	21.45	15.96	18.71	0.097	6.48	24.01	1.84	1.93	17.38	
VIP	<u>26.89</u>	27.02	25.60	19.87	22.73	0.088	5.56	47.61	1.74	1.76	33.85	
DASE-IDEALab	<u>26.89</u>	27.00	38.95	33.42	36.19	0.088	4.50	468.34	1.74	1.58	328.34	
ACVLAB	<u>25.31</u>	<u>25.45</u>	206.29	196.24	201.26	0.753	55.89	712.28e12	115.68	293.76	499.60e12	
SPAN-T	26.92	<u>26.98</u>	5.91	6.04	5.98	0.131	8.54	2.76	2.29	2.38	2.63	
hajnal	27.00	<u>26.95</u>	13.40	7.94	10.67	0.243	14.89	6.13	4.63	4.54	5.67	
KLETech CEVI_Lowlight_Hypnotise	<u>24.53</u>	<u>24.75</u>	315.17	302.12	308.64	16.698	1174.94	597.94e20	566.23e43	763.95e49	114.59e49	
RLFN (Baseline)	26.96	27.07	14.35	9.19	11.77	0.317	19.67	7.39	7.39	7.39	7.39	

derscores the meticulous craftsmanship and ingenuity embedded within their methodologies. With runtime bearing significant weight in the scoring system, it's noteworthy how closely the overall performance of each method mirrors their rankings in the runtime sub-track, reflecting the discernible impact of efficiency optimization on overall success. Indeed, our overarching objective this year has been to incentivize participants to embark on a quest for speed and efficiency in their designs—a mission that has undoubtedly borne fruit in the form of groundbreaking advancements and innovative solutions.

PSNR. While MagicSR, BU-ESR, ZHEstar, and BingBing showcase impressive PSNR performance—traditionally regarded as a cornerstone metric in method evaluation—MagicSR notably attains a remarkable 27.17 dB, closely followed by BU-ESR at 27.11 dB, and ZHEstar and BingBing both achieving 27.04 dB on the DIV2K_LSDIR_test set. However, amidst these accolades, it is paramount to underscore the overarching focus of this challenge: efficiency in super-resolution. Thus, in alignment with this objective, we opted to relax the PSNR threshold to a stringent lower bound of 26.90 and 26.99 for ranking on both the DIV2K_LSDIR_valid and DIV2K_LSDIR_test sets. This strategic adjustment aims to underscore the importance of balancing performance with efficiency. Notably, a total of 26 teams successfully met this revised requirement, showcasing their adeptness in navigating the delicate equilibrium between quality and computational efficiency. While several teams, such as SPAN-T, hajnal, FireWork, DIRN, and VIP, exhibit commendable efficiency performance, it is lamentable that they fell short of meeting the PSNR threshold, highlighting the multi-

faceted challenges inherent in the pursuit of efficient super-resolution excellence.

3.1. Main Ideas

Throughout this challenge, several techniques have been proposed to enhance the efficiency of deep neural networks for image super-resolution (SR) while striving to maintain optimal performance. The choice of techniques largely depends on the specific metrics that a team aims to optimize. Below, we outline some typical ideas that have emerged:

- **Parameter-free attention mechanism is validated as a useful technique to enhance computational efficiency** [22, 82]. Specifically, XiaomiMM proposed a swift parameter-free attention network based on parameter-free attention, which achieves the lowest runtime while maintaining a decent PSNR performance.
- **Re-parameterization** [20] [22, 82] **is commonly used in this challenge**. Usually, a normal convolutional layer with multiple basic operations (3×3 convolution, 1×1 operation, first and second-order derivative operators, skip connections) is parameterized during network training. During inference, the multiple operations that reparameterize a convolution could be merged back into a single convolution. *e.g.,* All the top three teams (*i.e.,* XiaomiMM, Cao Group, and BSR) used this operation in their solutions.
- **Incorporating multi-scale information and hierarchical module design are proven strategies for effectively fusing critical information**. For instance, solutions such as PiXupt, XJU_100th Ann, and ZHEstar have successfully utilized multi-scale residual connections and hierarchical module designs to enhance their performance.
- **Network pruning plays an important role**. It is observed that a couple of teams (*i.e.,* CMVG, AdvancedSR, and HiSR) used network pruning techniques to slightly compress a network. This leads to more lightweight architecture without heavy performance drop.
- **Exploration with new network architectures is conducted.** Besides the common CNN or Transformers, the state space model (*i.e.,* vision mamba [29]) was tried by BlingBling for the first time in this challenge.
- **Various other techniques are also attempted**. Some teams also proposed solutions based on neural architecture search, vision transformers, and advanced training strategies.

3.2. Fairness

To uphold the integrity and fairness of the efficient SR challenge, a series of rules were meticulously crafted, primarily focusing on the dataset utilized for training the network. Firstly, participants were granted permission to train their models with additional external datasets, such as Flickr2K, thereby fostering a diverse and comprehensive training regimen. However, to ensure unbiased evaluation, the use of the additional DIV2K and LSDIR validation sets comprising either high-resolution (HR) or low-resolution (LR) images was strictly prohibited during training. This measure was implemented to preserve the integrity of the validation set, which served as a crucial yardstick for assessing the overall performance and generalizability of the proposed networks. Furthermore, training with DIV2K and LSDIR test LR images was unequivocally forbidden, safeguarding the sanctity of the test dataset and maintaining the sanctity of the evaluation process. Lastly, employing advanced data augmentation strategies during training was deemed a fair and equitable approach, empowering participants to optimize their models while adhering to the established guidelines and regulations.

3.3. Conclusions

Several conclusions can be drawn from the analysis of different solutions as follows. Firstly, The competition for the efficient image SR community is still fierce. This year the challenge had 262 registered participants, and 34 teams made valid submissions. All the proposed methods improve the state-of-the-art for efficient SR. Secondly, reparameterization and network compression play an important role in efficient SR. More exploration is still encouraged to further improve the model's efficiency with these techniques. Thirdly, regarding the training, the adoption of large-scale dataset [46] for pre-training improves the accuracy of the network and for most of the methods, the training of the network proceeds in several phases with increased patch size and reduced learning rate. Fourthly, the state space model was explored for the first time in the challenge, which may draw a new model choice for the upcoming works. Finally, by jointly considering runtime, FLOPs, and the number of parameters, it is possible to design a balanced model that optimizes more than one evaluation metric. Finally, as computational capabilities advance, the optimization of models for runtime, FLOPs, and parameter count will become increasingly crucial. With ongoing developments in hardware and algorithmic efficiency, there is a strong likelihood of even more sophisticated and resource-efficient models emerging in the field of super-resolution (SR). We anticipate that the pursuit of efficiency in SR will persist and be further explored, leading to continued progress and innovation in the field.

4. Challenge Methods and Teams

4.1. XiaomiMM

Method. The authors propose the Swift Parameter-free Attention Network (SPAN) [75], a highly efficient SISR model that balances parameter count, inference speed, and image quality. As shown in Figure 1, SPAN consists of 6

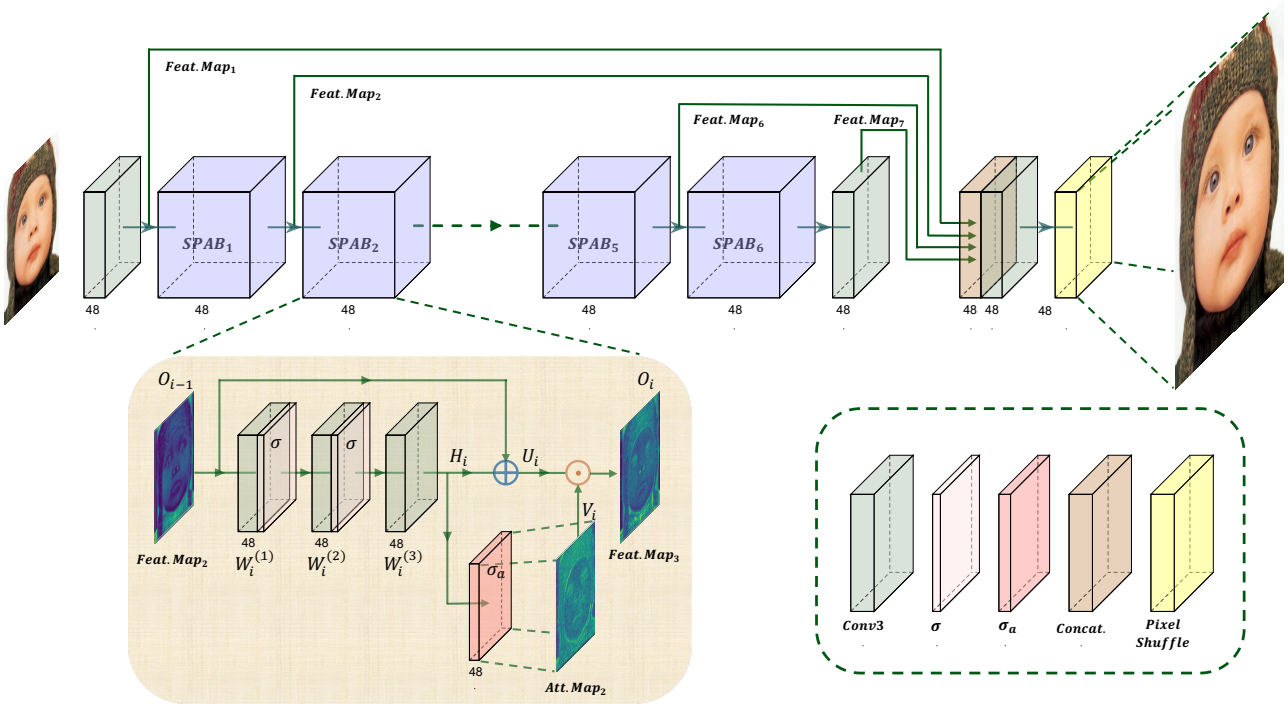


Figure 1. *Team XiaomiMM:* The proposed SPAN architecture [75]. The brown area indicates the internal structure of each SPAB module. $Att.Map_2$ denotes the generated attention map.

consecutive SPABs and each SPAB block extracts progressively higher-level features sequentially through three convolutional layers with C'-channeled $H' \times W'$-sized kernels (In our model, they choose $H' = W' = 3$). The extracted features H_i are then added with a residual connection from the input of SPAB, forming the pre-attention feature map U_i for that block. The features extracted by the convolutional layers are passed through an activation function $\sigma_a(\cdot)$ that is symmetric about the origin to obtain the attention map V_i. The feature map and attention map are element-wise multiplied to produce the final output $O_i = U_i \odot V_i$ of the SPAB block, where $\odot$ denotes element-wise multiplication. They use $W_i^{(j)} \in R^{C' \times H' \times W'}$ to represent the kernel of the j-th convolutional layer of the i-th SPAB block and σ to represent the activation function following the convolutional layer. Then the SPAB block can be expressed as:

$$\begin{aligned} O_i &= F_{W_i}^{(i)}(O_{i-1}) = U_i \odot V_i, \\ U_i &= O_{i-1} \oplus H_i, \quad V_i = \sigma_a(H_i), \\ H_i &= F_{c,W_i}^{(i)}(O_{i-1}), \\ &= W_i^{(3)} \otimes \sigma(W_i^{(2)} \otimes \sigma(W_i^{(1)} \otimes O_{i-1})), \end{aligned} \tag{3}$$

where $\oplus$ and $\otimes$ represent the element-wise sum between extracted features and residual connections, and the convolution operation, respectively. $F_{W_i}^{(i)}$ and $F_{c,W_i}^{(i)}$ are the function representing the i-th SPAB and the function representing the 3 convolution layers of i-th SPAB with parameters $W_i = (W_i^{(1)}, W_i^{(2)}, W_i^{(3)})$, respectively. $O_0 = \sigma(W_0 \otimes I_{\text{LR}})$ is a C'-channeled $H \times W$ feature map from the C-channeled $H \times W$-sized low-resolution input image I_{LR} undergone a convolutional layer with 3×3 sized kernel W_0. This convolutional layer ensures that each SPAB has the same number of channels as input. The whole SPAN neural network can be described as

$$\begin{aligned} I_{\text{HR}} &= F(I_{\text{LR}}) = \text{PixelShuffle}[W_{f2} \otimes O)], \\ O &= \text{Concat}(O_0, O_1, O_5, W_{f1} \otimes O_6), \end{aligned} \tag{4}$$

where O is a $4C'$-channeled $H \times W$-sized feature map with multiple hierarchical features obtaining by concatenating O_0 with the outputs of the first, fifth, and the convolved output of the sixth SPAB blocks by C'-channeled 3×3-sized kernel W_{f1}. O is processed through a 3×3 convolutional layer to create an r^2C channel feature map of size $H \times W$. Then, this feature map goes through a pixel shuffle module to generate a high-resolution image of C channels and dimensions $rH \times rW$, where r represents the super-resolution factor. The idea of computing attention maps directly without parameters from features extracted by convolutional layers led to two design considerations for our neural network: the choice of activation function for computing the attention map and the use of residual connections, more details about activation function and SPAB module are in [75].

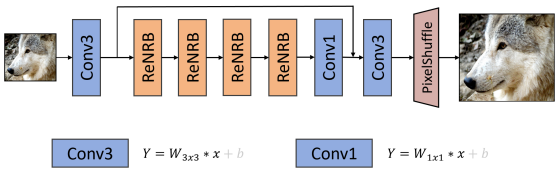


Figure 2. *Team Cao Group:* The structure of R2Net

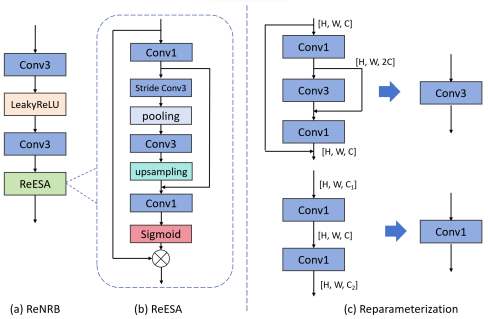


Figure 3. *Team Cao Group:* The structure of ReNRB and reparameterization in R2Net

Training Details. The dataset utilized for training comprises DIV2K and LSDIR. During each training batch, 64 HR RGB patches are cropped, measuring 256×256, and subjected to random flipping and rotation. The learning rate is initialized at 5×10^{-4} and undergoes a halving process every 2×10^5 iterations. The network undergoes training for a total of 10^6 iterations, with the L1 loss function being minimized through the utilization of the Adam optimizer [35]. They repeated the aforementioned training settings four times after loading the trained weights. Subsequently, fine-tuning is executed using the L1 and L2 loss functions, with an initial learning rate of 1×10^{-5} for 5×10^5 iterations, and an HR patch size of 512. They conducted the finetuning on four models utilizing both L1 and L2 losses, employed batch sizes of 64 and 128, and integrated these four models to obtain the ultimate model.

4.2. Cao Group

Method. The overall architecture of their network is shown in Figure 2, which is inspired by previous leading methods, DIPNet [84] and SRN [79]. They propose a Double Reparameterization Network (R2Net). Specifically, they build upon the SRN framework by combining Residual Blocks (RB) and Enhanced Spatial Attention (ESA) to form a new feature extraction module with reparameterization, named reNRB, as shown in Figure 3. They remove the residual connections within RB and the 1x1 convolutions on the residual connections in ESA, retaining only the global residual connections. Furthermore, different from

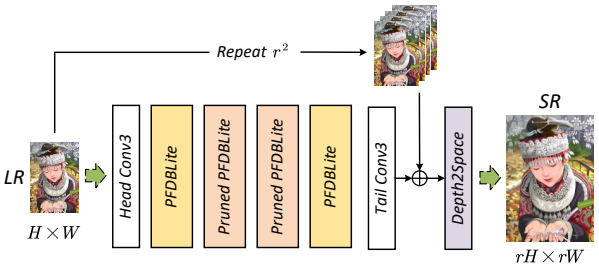


Figure 4. *Team BSR:* PFDNLite Architecture.

SRN [79], they find that preserving the last convolution before the global residual connection significantly boosts performance. For the sake of lightweight design, they set its kernel size to 1x1. Notably, they eliminate biases in all convolutional layers, as this not only significantly accelerates inference speed and reduces parameters but also slightly enhances network performance.

As for the reparameterization technique, they adopt RRRB [24] for all 3x3 convolutions. And inspired by the Feed Forward Network (FFN) module structure in transformer, they propose a reparameterization module to enhance the performance of 1x1 convolutions by expanding the intermediate channel, thereby harnessing the representation capability of complex structures during optimization, as shown in Figure 3(c).

Training Details. They train the network on RGB channels and augment the training data with random flipping and rotation. The number of ESA channels is set to 16, while the number of feature channels is set to 48. Following previous methods, the training process is divided into three stages:

1. In the first stage, they randomly crop 256x256 HR image patches from ground truth images, with a batch size of 32. They use the Adam optimizer, setting $\beta 1$ = 0.9 and $\beta 2$ = 0.999, and minimize the Charbonnier loss function. The initial learning rate is set to 2e-4, with a cosine learning rate decay strategy. The number of iterations is set to 2e-6.
2. In the second stage, they increase the size of the HR image patches to 512x512, with other settings remaining the same as in the first stage.
3. In the third stage, the batch size is set to 64, and L2 loss is adopted to optimize over 2e-6 iterations. The initial learning rate is set to 2e-5.

Throughout the entire training process, they employ an Exponential Moving Average (EMA) strategy to enhance the robustness of training.

4.3. BSR

Method. Inspired by ABPN [23] and PFDN [47], the PFDNLite, as shown in Fig. 4, consists of two PFDBLite blocks and two pruned-PFDBLite blocks. For $\times 4$ SR, the input image is repeated r^2 times and then added to the final

feature. Based on EFDN [78] and PFDN [47], they modify a more lightweight partial feature distillation block, dubbed PFDBLite to chase faster feature extraction. Generally, they execute two modifications focusing on the reparameterizable convolution and attention module. For the convolution block, they employ RepMBConv, which squeezes the MobileNetv3 block into vanilla convolution for better trade-offs between performance and memory access. Moreover, they add a reparameterizable point-wise convolution to cooperate with the middle RepMBConv as an approximation of partial convolution [8]. For the attention module, they propose a local attention (LocalAttn), which applies a local gate and MaxPool-based importance map to modulate input features. As illustrated in Fig. 5, they provide the details of RepMBConv and LocalAttn.

Additionally, as exhibited in Fig. 6, the Pruned-PFDBLite is similar to PFDBLite but drops the second RepMBConv of PFDBLite and decreases the output channels of the first RepMBConv from 48 to 24. Besides, they add the point-wise convolution after the first RepMBConv.

Training details. The training process contains two stages with four steps. And the training dataset is the DIV2K_LSDIR_train [1].

I. At the first stage, they only use PFDBLite blocks in the PFDNLite.

- Step1. HR patches of size 256×256 are randomly cropped from HR images, and the mini-batch size is set to 96. L1 loss with AdamW optimizer is used and the initial learning rate is set to 0.0005 and halved at every 100k iterations. The total iterations is 500k.
- Step2. HR patches of size 256×256 are randomly cropped from HR images, and the mini-batch size is set to 96. Charbonieer loss with AdamW optimizer is used and the initial learning rate is set to 0.0003 and halved at every 100k iterations. The total iterations is 500k.
- Step3. HR patches of size 480×480 are randomly cropped from HR images, and the mini-batch size is set to 64. MSE loss with AdamW optimizer is used and the initial learning rate is set to 0.0001 and halved at every 100k iterations. The total iterations is 500k.

II. At the second stage, they replace the second and the third PFDBLite block with Pruned-PFDBLite and use the weight of PFDBLite to initialize Pruned-PFDBLite.

- Step4. HR patches of size 480x480 are randomly cropped from HR images, and the mini-batch size is set to 64. MSE loss with AdamW optimizer is used and the initial learning rate is set to 0.0001 and halved at every 100k iterations. The total iterations is 500k.

4.4. PiXupt

Method. The PiXupt team proposed the Hierarchical Attention Residual Network (HARN), as shown in Fig. 7. HARN adopts a similar basic framework to [34, 48, 52, 62]. However, HARN uses the Hierarchical Self-Attention Module (HSAM) instead of the original spatial attention in [62], and uses the Hierarchical Separable Residual Block (HSRB) instead of the original Blueprint Shallow Residual Block (BSRB) in [48, 62]. As shown in the Fig. 9 (b), HSAM first divides the inputs into four groups from the channel dimension, where the first group performs the self-attention calculation within a large window, and then the output features are fused as hidden state with the inputs of the next group using a Gate Recurrent Unit[16] (GRU). Such an approach allows feature information to be shared between different groups, which can help different groups to use the large window information without having to use the large window for each head. Secondly, as shown in Fig. 9 (a), the proposed HSRB improves on the depth-wise convolution used in the BSRB. HSRB first fuses the channel information of the input features using point-wise convolution, then groups the features and uses different sizes of depth-wise convolution kernels for different groups and connects them using a hierarchical structure to extract richer local features. HSAM and HSRB are contained in the basic module of HARN, Hierarchical Attention Distillation Block (HADB), as shown in Fig. 8.

Training Details. The proposed HARN has 4 HADBs, in which the number of feature channels is set to 20. The details of training steps are as follows:

1. Pretraining on DIV2K[71]: HR patches of size 256 × 256 are randomly cropped from HR images, and the mini-batch size is set to 64. The model is trained by minimizing L1 loss function with Adam optimizer. The initial learning rate is set to 2×10^{-3} and halved at $\{100k, 500k, 800k, 900k, 950k\}$-iteration. The total number of iterations is 1000k.
2. Finetuning on 800 images of DIV2K and the first 10k images of LSDIR. HR patch size and mini-batch size are set to 384 × 384 and 32, respectively. The model is fine-tuned by minimizing Charbonnier loss function. The initial learning rate is set to 5×10^{-4} and halved at $\{100k, 500k, 800k, 900k, 950k\}$-iteration. The total number of iterations is 1000k.
3. Finetuning on 800 images of DIV2K and the first 10k images of LSDIR again. HR patch size and the mini-batch size are set to 384 × 384 and 32, respectively. The model is fine-tuned by minimizing the L2 loss function. The initial learning rate is set to 2×10^{-4} and halved at $\{100k, 300k, 600k\}$-iteration. The total number of iterations is 650k.

4.5. XJU_100th Ann

Method. They propose an attention guidance distillation network (AGDN) for efficient image super-resolution, which is influenced by existing studies such as IMDN[34], RFDN[52], BSRN[48], and MDRN[62], and further im-

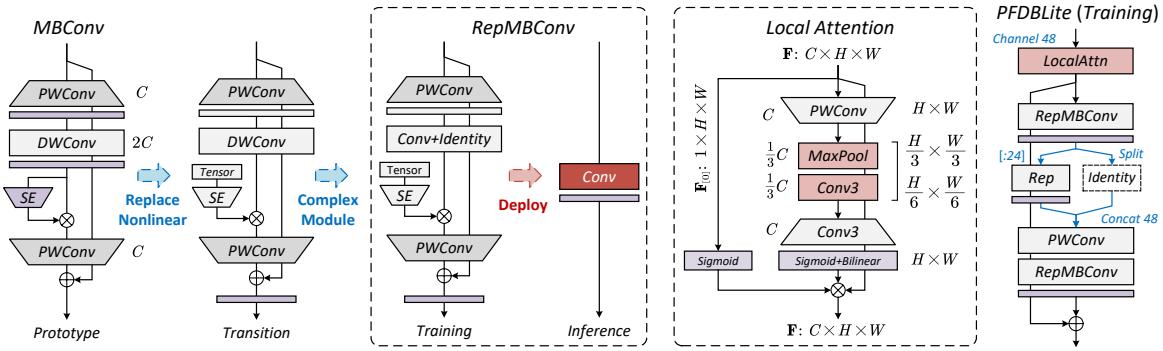


Figure 5. *Team BSR:* PFDBLite Block.

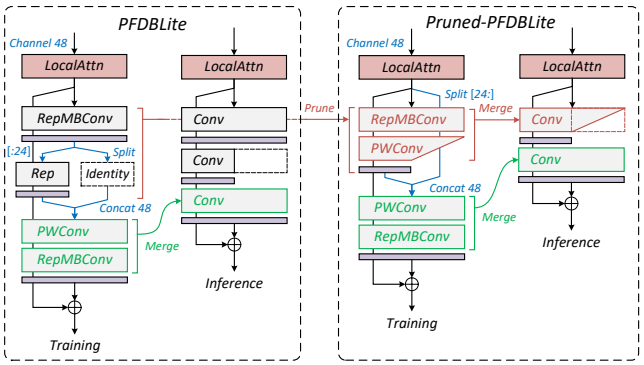


Figure 6. *Team BSR:* Pruned PFDBLite Block.

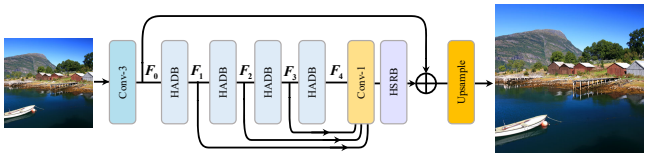


Figure 7. *Team PiXupt:* The whole framework of Hierarchical Attention Residual Network (HARN)

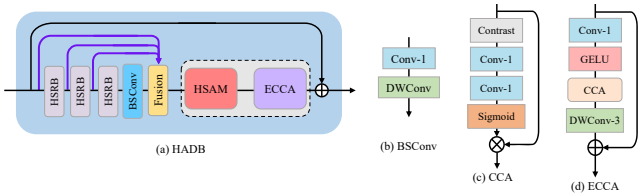


Figure 8. *Team PiXupt:* Hierarchical Attention Distillation Block (HADB).

proved based on these studies. Fig.10 illustrates the overall architecture of their network, which has been extensively validated in previous studies.

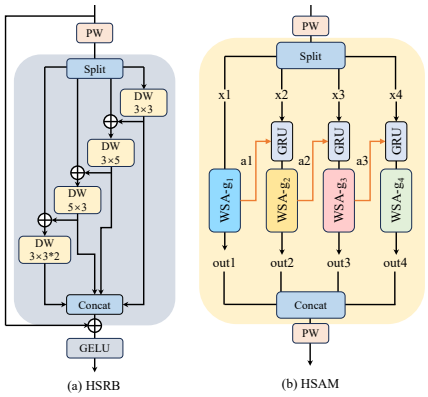


Figure 9. *Team PiXupt:* (a) Hierarchical Separable Residual Block (HSRB). (b) Hierarchical Self-Attention Module (HSAM).

They have reconsidered the previous network structure, and the feature extraction phase remains a key limiting factor for network performance. The feature distillation block comprises distillation in the pre-phase and enhancement in the post-phase. Thus, improving both distillation and enhancement can significantly boost network performance.

Based on the above analysis, they propose the new attention guidance distillation block (AGDB) with more efficient spatial attention, channel attention and self-attention as the base block of AGDN. As shown in Fig.11, they use the multi-level variance-aware spatial attention (MVSA) and reallocated contrast-aware channel attention (RCCA) as alternatives to the enhanced spatial attention (ESA) [53] and contrast-aware channel attention (CCA) [34], and introduce sparse global self-attention (SGSA) [41] to achieve further feature enhancement.

In MVSA, they consider the impact of multi-level branching and local variance on performance. Multi-level branches with small windows cannot cover a sufficient range of information while using local variance in a single branch can lead to large differences in weights between

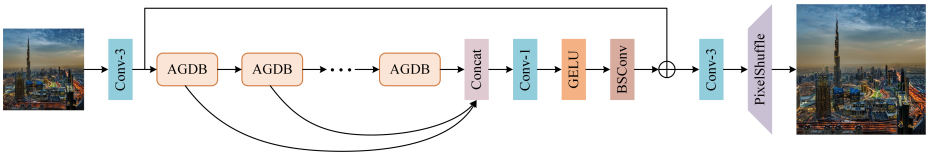


Figure 10. *Team XJU_100th Ann:* The overall architecture of attention guidance distillation network (AGDN).

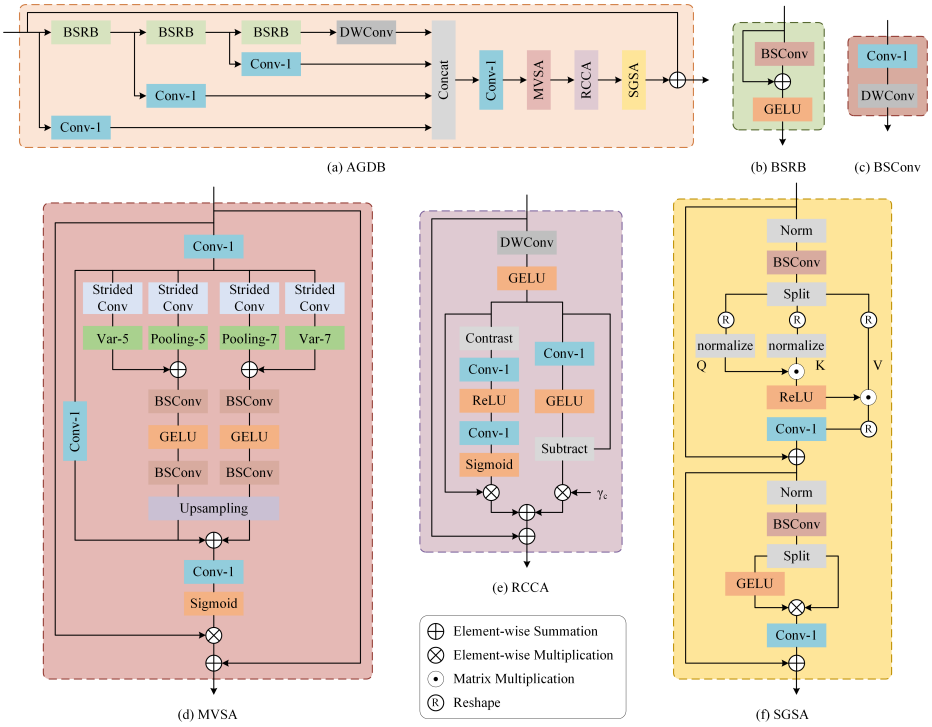


Figure 11. *Team XJU_100th Ann:* The details of each component. (a) AGDB: Attention Guidance Distillation Block; (b) BSRB: Blueprint Shallow Residual Block; (c) BSConv: Blueprint Separable Convolution; (d) MVSA: Multi-level Variance-aware Spatial Attention; (e) RCCA: Reallocated Contrast-aware Channel Attention; (f) SGSA: Sparse Global Self-attention.

branches. Therefore, they improved MDSA [62] to obtain D5 and D7 branches that contain both local variance to better capture structurally information-rich regions while balancing performance and model complexity In RCCA, they not only consider the reallocation of weights across channels by traditional channel attention but also enhance the treatment of common information across all channels. They added complementary branches with 1×1 convolution and GELU activation representations to reallocate complementary channel information, promoting the uniqueness of each channel. Finally, they introduce SGSA for selecting the most useful similarity values, aiming to better utilize essential global features for image reconstruction. In image reconstruction, there is usually a gap in global attention between the training and testing phases. Therefore, they adopt the default enhancement approach of SGSA, which is to apply the test-time localizer converter (TLC) [15] approach during the testing phase.

Training Details. The proposed AGDN has 4 AGDBs, in which the number of feature channels is set to 24. The details of the training steps are as follows:

1. Pretraining on the DIV2K [1] and Flickr2K [51] datasets. HR patches of size 256×256 are randomly cropped from HR images, and the mini-batch size is set to 64. The model is trained by minimizing the L1 loss function with the Adam optimizer. The initial learning rate is set to 2×10^{-3} and halved at {100k, 500k, 800k, 900k, 950k}-iteration. The total number of iterations is 1000k.

2. Finetuning on 800 images of DIV2K and the first 10k images of LSDIR[46]. HR patch size and mini-batch size are set to 384×384 and 32, respectively. The model is fine-tuned by minimizing the L2 loss function. The initial learning rate is set to 5×10^{-4} and halved at 50k iteration. The total number of iterations is 100k.

4.6. VPEG_C

Method. They introduce a self-modulation feature aggregation (SMFA) module as shown in Figure 12 to collaboratively exploit both local and non-local feature interactions for image super-resolution. Specifically, the SMFA module employs an efficient approximation of the self-attention (EASA) branch to model non-local information and uses a local detail estimation (LDE) branch to capture local details. Additionally, they further introduce a partial convolution-based feed-forward network (PCFN) to refine the representative features derived from the SMFA. Given the input feature $F_{in} \in \mathbb{R}^{H\times W\times C}$, where $H\times W$ denotes the spatial size and C is the number of channels, they first apply a 1×1 convolution to the normalized F_{in} to expand the channel, and then split the channel into two parts as inputs to the efficient approximation of self-attention (EASA) and local detail estimation (LDE) branches:

$$\{X, Y\} = \mathcal{S}(Conv_{1\times1}(||F_{in}||_2)), \tag{5}$$

where $||\cdot||_2$ is the L_2 normalization, $Conv_{1\times1}(\cdot)$ denotes a 1×1 convolutional layer, $\mathcal{S}(\cdot)$ denotes a channel splitting operation, and $\{X, Y\} \in \mathbb{R}^{H\times W\times C}$. They then process the features X and Y in parallel via the EASA and LED branches, producing the non-local feature X_l and local feature Y_d, respectively. Finally, they fuse X_l and Y_d together with element-wise addition and feed them into a 1×1 convolution to form a representative output of the SMFA module. This process can be formulated as:

$$F_\rho = F_{in} + Conv_{1\times1}(X_l + Y_d), \tag{6}$$

where $F_\rho \in \mathbb{R}^{H\times W\times C}$ is the output feature.

For an efficient approximation of self-attention, they obtain the low-frequency components through a downsampling operation and feed them into a 3×3 depth-wise convolution to generate non-local structure information $X_s \in \mathbb{R}^{H/8\times W/8\times C}$:

$$X_s = DWConv_{3\times3}(\mathcal{D}(X)), \tag{7}$$

where $\mathcal{D}(\cdot)$ denotes the adaptive max pooling with a scaling factor of 8, $DWConv_{3\times3}(\cdot)$ is a 3×3 depth-wise convolutional layer. To embed global descriptions for modulating non-local representation X_s, they introduce variance $\nu(X) \in \mathbb{R}^{1\times1\times C}$ of the input X and calculate it in the spatial dimension. X_s and $\nu(X)$ are then added and fed into a 1×1 convolution to fuse the information thoroughly:

$$X_m = Conv_{1\times1}(X_s + \nu(X)), \tag{8}$$

where $X_m \in \mathbb{R}^{H\times W\times C}$ represents the modulated feature. This variance modulation mechanism facilitates better exploring non-local information.

Finally, they use the modulated features to aggregate the input feature X for extracting the representative structure information X_l:

$$X_l = X \odot \mathcal{U}(\phi(X_m)), \tag{9}$$

where $\phi(\cdot)$ refers to the GELU activation function [32], $\mathcal{U}(\cdot)$ denotes a nearest upsampling operation, and $\odot$ represents the element-wise product operation.

Training Details. The proposed SMFAN consists of 8 FMBs and the number of channels is set to 24. They first train the proposed SMFAN on the DIV2K [71] and Flickr2K [51] datasets. The cropped LR image size is 96 $\times$ 96 and the mini-batch size is set to 64. The SMFAN is trained by minimizing L1 loss and the frequency loss [13] with Adam optimizer for total of 800,000 iterations. They set the initial learning rate to 2×10^{-3} and the minimum one to 1×10^{-5}, which is updated by the Cosine Annealing scheme [60]. After that, they use the first 10,000 images of LSDIR [46] dataset for fine-tuning. The cropped LR image size is 160 $\times$ 160 and the mini-batch size is set to 32. The fine-tuning stage uses MSE loss and the frequency loss [13] with 500,000 iterations. The initial learning rate is set to 2×10^{-5} and the minimum one to 1×10^{-7}.

4.7. ZHEstar

Method. The ZHEstar team proposed Large Kernel Frequency-enhanced Network (LKFN) [9]. The architecture is shown in Fig.13. It is based on BSRN [48]. They replaced the RBSB in BSRN with their Partial Large Kernl Block (PLKB) shown in Fig.14. Inspired by PConv [8], PLKB first divides the input feature map into two halves in the channel dimension. One half undergoes a 5×5 depth-wise convolution (The third PLKB with a dilation rate of 3.), and the result is then concatenated with the unprocessed other half. A 1×1 convolution is subsequently used to perform data exchange between these two parts. For their Frequency-enhanced Pixel Attention (FPA) module, it first transforms the spatial domain feature map to the frequency domain through Fourier transform, then pass the frequency domain map through a three-layer 1×1 convolution, followed by two LeakyReLUs. The result is added to the initial frequency domain map via a residual connection to obtain the enhanced frequency domain attention map. Then it was transformed back to the spatial domain and multiplied by the input spatial feature map.

Training Details. The proposed LKFN consists of 8 LKFBs and the feature channel is set to 28. The training data includes 800 images from DIV2K [1] and the first 10K images from LSDIR [46]. They use the default parameter settings of the Adan optimizer [81] in the whole process. The training process is as follows:

1. Training with an input patch size of 64×64 and a mini-batch size of 64 from scratch by minimizing the $\mathcal{L}_1$ loss.

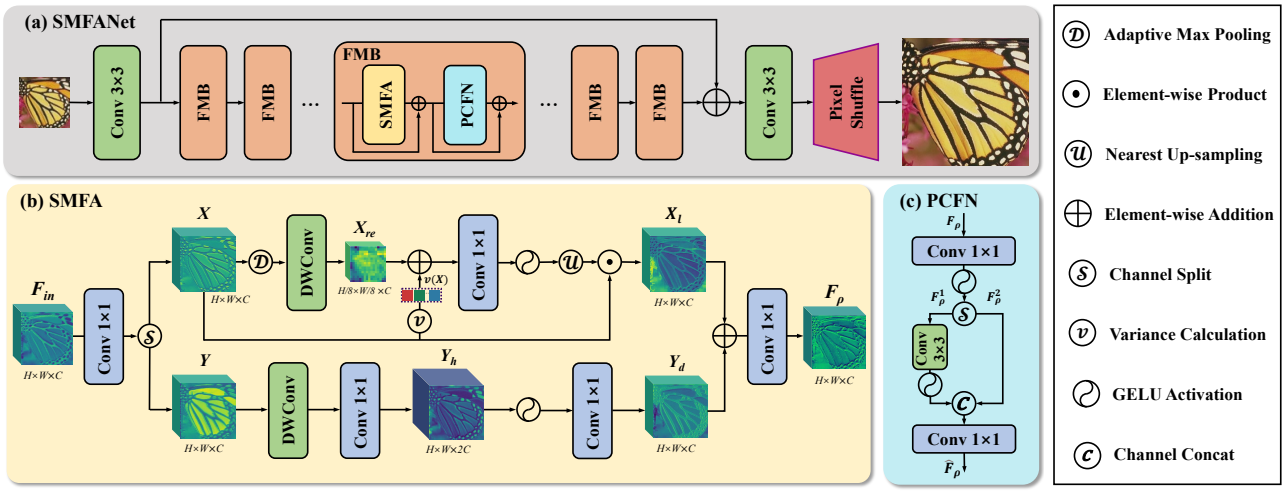


Figure 12. *Team VPEG_C* : An overview of the proposed SMFAN.

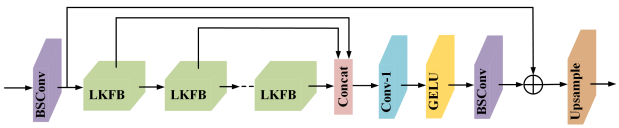


Figure 13. *Team ZHEstar:* The framework of Large Kernel Frequency-enhanced Network (LKFN)

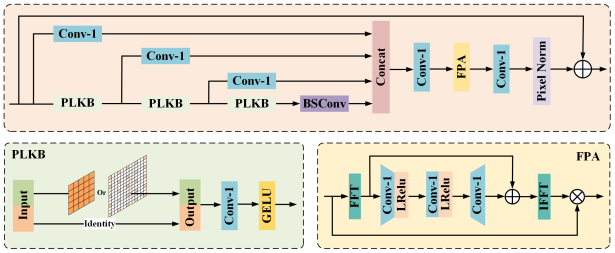


Figure 14. *Team ZHEstar:* Large Kernel Frequency-enhanced Block (LKFB)

The initial learning rate is set to 5×10^{-3}. The learning rate decay is following cosine annealing with T_{max} = total iterations, $\eta_{min} = 1 \times 10^{-7}$. The total number of iterations is 1000K.

2. Finetuning with an input patch size of 120×120 and a mini-batch size of 64 by minimizing the MSE loss. The learning rate is set to 2×10^{-5} during this stage. The total number of iterations is 150K.

4.8. VPEG_O

Method. The VPEG_O team introduces SAFMN++, an improved version of SAFMN [70] for solving efficient SR. This solution mainly concentrates on improving the

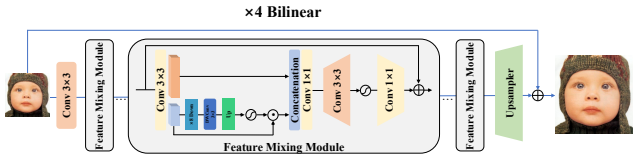


Figure 15. *Team VPEG_O*: The overall network architecture of their proposed SAFMN++.

effectiveness of the spatially adaptive feature modulation (SAFM) [70] layer. Different from the original SAFM, as shown in Fig 15, the improved SAFM (SAFM+) can extract both local and non-local features. In SAFM+, a 3×3 convolution is first utilized to extract local features and a single scale feature modulation is then applied to a portion of the extracted features for non-local feature interaction. After this process, these two sets of features are aggregated by channel concatenation and fed into a 1×1 convolution for feature fusion.

Training details. The proposed SAFMN++ consists of 6 feature mixing modules, and the number of channels is set to 36. They train the proposed SAFMN++ on the LSIDR [46] dataset. The cropped LR image size is 120×120 and the mini-batch size is set to 64. The SAFMN++ is trained by minimizing L1 loss and the frequency loss[13] with Adam optimizer for total 800, 000 iterations. They set the initial learning rate to 3×10^{-3} and the minimum one to 1×10^{-6}, which is updated by the Cosine Annealing scheme [60].

4.9. CMVG

Method. They propose a residual knowledge distillation super-resolution network named RDEN for efficient super-

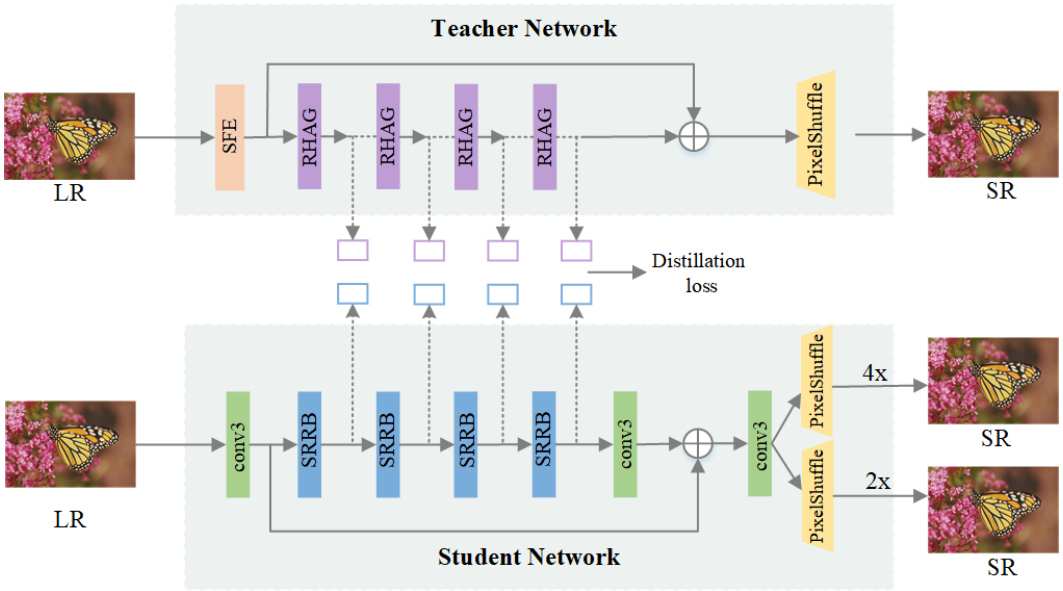


Figure 16. *Team CMVG:* The framework of RDEN

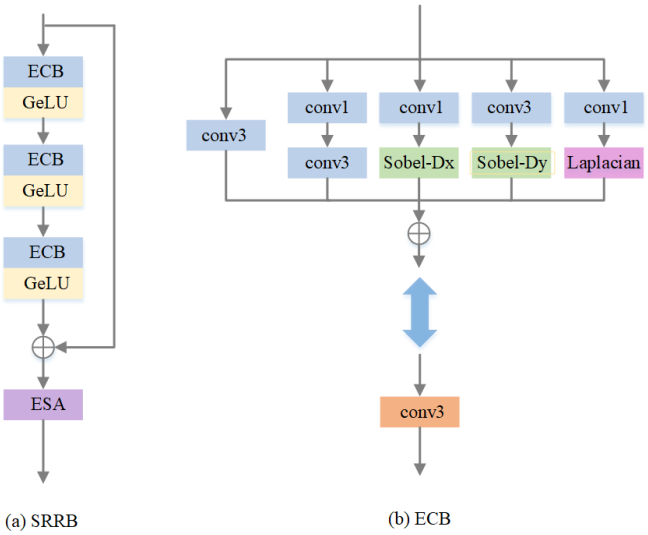


Figure 17. *Team CMVG:* The framework of SRRB

resolution(SR) as shown in Figure.16. The RDEN model is composed of a large teacher network and a lightweight student network, they apply the superior SR model HAT[10] as a teacher network. The distillation training provides additional effective supervision information for student training, and enhance the performance and generalization ability of the student network. The student network includes the shallow feature extraction, the depth feature extraction, and the reconstruction modules. There are four simplified reparameterization residual blocks (SRRB) in depth feature extraction modules, as shown in Figure.17. The reconstruction module includes a $4\times$ and $2\times$ upsampling head. A SRRB block is composed of three residual connected ECB[88] blocks and ESA[36]. They utilize ECB blocks during the training phase, while they can be merged into a single 3x3 convolution layer during inference through mathematical transformation. The number of feature maps of Conv3 and SRRB is set to 38 respectively, after the pruning, the feature maps are reduced to 37 finally. Although the model is small, the knowledge distillation and reparameterization

provide the information compensation, which enables their model to achieve good reconstruction performance. They train the teacher network with L1 loss, the teacher loss is denoted as follows:

$$L_T = \|Y_T^{SR} - Y_T^{HR}\| \tag{10}$$

Where the Y_T^{SR} is SR image from teacher and Y_T^{HR} is HR image. The student network is trained by the distillation loss, L1 loss, and joint supervision loss. They extract features from the 0th, 2nd, 4th, and 5th blocks of the teacher model, and features from each SRRB block, then they use a 1*1 convolution to expand the student feature dimension to match corresponding teacher features. The distillation loss is denoted as follows:

$$L_{distillation} = \lambda_i \sum_{i=0}^{4} \|F_T^i - F_S^i\| + \mu \|Y_T^{SR} - X_S^{SRx4}\| \tag{11}$$

where F_S^i represents the feature map of the output of the i-th block of the student network, and F_T^i is the corresponding features of teacher model. X_S^{SRx4} represents the output of 4x upsample heads in student network. The joint supervision loss is composed of GM loss, FFT loss, and 2x supervision loss, denoted as follows:

$$L_{GM} = \|GM(X_S^{SRx4}) - GM(X_S^{HRx4})\| \tag{12}$$

$$L_{FFT} = \|FFT(X_S^{SRx4}) - FFT(X_S^{HRx4})\| \tag{13}$$

$$L_{X2} = \|X_S^{SRx2} - X_S^{HRx2}\| \tag{14}$$

where $GM(.)$ and $FFT(.)$ respectively represent the gradient map and focal frequency extraction operators, the X_S^{SRx2} denotes the output from 2× upsampling head, the X_S^{HRx4} and X_S^{HRx2} are corresponding 4×and 2× HR image. Finally, the student loss for training student network is :

$$L_S = \alpha\|X_S^{SRx4} - X_S^{HRx4}\| + \beta L_{distillation} + \gamma L_{GM} + \delta L_{FFT} + \epsilon L_{X2} \tag{15}$$

Training Details. They train their model on DIV2K, Flickr2K, and LSDIR datasets, and the multi-stage progressive training strategy is used to optimize and finetune. The progressive training strategy gradually increases patch size, changes loss function, and loads weights from the previous step to improve performance. The training details are described as follows:

Stage1 *Training teacher network:* The teacher network HAT is trained from scratch with teacher loss.

Stage2 *Training student network:* The teacher network is fixed, and they pretrain a 2× network to initialize the student network. Then, the student network is optimized through distillation training by student loss. The initial learning rate is set to 5e-4 and halved at every 60 epochs and the total number of epochs is 500. The batch size and patch size are set to 64 and 256 separately. Data augmentation is also adopted.

Stage3 The finetune steps of the student network are described as follows:

(1) The student model is initialized from Stage 2 and trained with the same settings as in the previous step.

(2) The student model is initialized from Stage 3.1 and trained with the same settings as Stage 3.1, especially since the loss function is only MSE loss.

(3) The student model is initialized from the previous step and finetuned by MSE loss further, it is worth noting that the patch size is set to 512. Other parameter settings are not changed and finally, the student model is finetuned with 640 HR patches and MSE loss.

4.10. LeESR

Method. Inspired by the RLFN [36] and BSRN [48], this team proposed a Separable-Mixable Residual Network (SMRN) as illustrated in Fig.18 for Efficient image Super-Resolution, which can maintain lower parameters and computation while performing faster. Unlike the popular RFDB (see Fig.19(a)) in RFDN [52], the RLFB (see Fig.19(b)) in RLFN, and the ESDB (see Fig.19(c)), the BSConv (see Fig.19(d)) in the proposed SMRB consists of a 1×1 pixel-wise convolution and a depth-wise convolution and considers the intra kernel correlation. Among it, a kernel on a single channel (as a blueprint) is multiplied with different weights (e.g., 1×1 pixel-wise convolution) to obtain the convolution kernels on other different channels. Obviously, the strategy can greatly simplify traditional convolution operations. However, only the kernel of different channels separately is limited, the features of different channels still need to mix for performance improvement. Therefore, as shown in Fig.19(d), a Separable-Mixable Residual Block (SMRB) is designed, which consists of both the blueprint separable operation and the information mixable operation. Specifically, they introduce traditional convolutional kernels to be used in conjunction with BSconv for feature separation and mixable.

Reparameterization has improved the performance of ESR without introducing any inference cost. In Separable-Mixable Residual Block (SMRB), they introduce the Re-

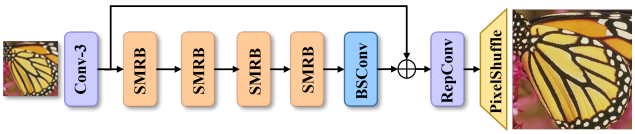


Figure 18. *Team LeESR:* The framework of the proposed Separable-Mixable Residual Network (SMRN).

parameterized Blueprint Separable Block (RBSB) to improve the BSconv representation, while a Re-parameterized Information Mixable Block (RIMB) is designed to replace the regular convolution. Specifically, inspired by SESR [5], during training, the conv in RIMB is a collapsible block which consists of a 3 × 3 convolution and a 1×1 convolution in the expanded space. During inference, they collapse them to a single 3×3 convolution as depicted in Fig.19(d). In image reconstruction, a re-parameterized 3 × 3 convolution (RepConv) as shown in Fig.18 is utilized to replace the single 3 × 3 convolution.

Training Details. The proposed SMRN contains four SMRBs, in which they set the number of feature maps to 48. Also, the channel number of the ESA is set to 16 similar to [36]. Besides, the collapsed channel number in the collapsible block is 256. Throughout the entire training process, they use the Adam optimizer [35], where $\beta 1$ = 0.9 and $\beta 2$ = 0.999. The model is trained for 1000k iterations in each stage. Input patches are randomly cropped and augmented. Data augmentation strategies included horizontal and vertical flips, and random rotations of 90, 180, and 270 degrees. Model training was performed using Pytorch 1.11.0 [66] on one NVIDIA A100 40G GPUs. Specifically, the training strategy consists of several steps as follows.

1. In the starting stage, they train the model from scratch on the 800 images of DIV2K [71] and the first 10k images of LSDIR [46] datasets. The model is trained for total 10^6 iterations by minimizing Charbonnier loss and FFT loss [13]. The HR patch size is set to 256×256, while the mini-batch size is set to 64. They set the initial learning rate to 1×10^{-3} and the minimum one to 1×10^{-6}, which is updated by the Cosine Annealing scheme.

2. In the second stage, they increase the HR patch size to 384. The model is fine-tuned by minimizing the Charbonnier loss and FFT loss. They utilize the MultiStepLR scheduler with a warm-up strategy (2,000 iterations for warm-up), where the initial learning rate is set to 5×10^{-4} and halved at 200k, 400k, 800k-iterations.

3. In the third stage, the model is initialized with the pre-trained weights of Stage 2, and fine-tuned with larger HR patches of size 480x480. Other settings are the same as in the second stage.

4. In the fourth stage, the model is further fine-tuned with 480×480 HR patches, however, the loss function is changed to minimize the combination of L2 loss and FFT loss. Other settings are the same as Stage 3.

4.11. AdvancedSR

Method. The AdvancedSR team proposes a kernel-pruning-based Residual Local Feature Network [36] for efficient SR. The overall network architecture is illustrated in Figure.20. It is a lightweight network consisting of a series of RLFB blocks, similar to RLFN. However, they prune the second convolution in the pruned-RLFB block based on sensitivity analysis. Additionally, the pixel shuffle block is used for image restoration.

Additionally, this team perform the pruning steps based on NTIRE2024 official baseline model [36]. The pruning process consists of two stages. In the first stage, they apply progressive kernel-pruning on the re-parameterized model inspired by UPDP[55]. And in the second stage, they apply bias prune in their model to accelerate the runtime.

Stage 1. Kernel Pruning. They conducted a sensitivity analysis on all the conv3×3 of the RLFN Network. Based on the proportion of responses of the center weights in Conv3, they replaced the conv3 with conv1 progressively. The experiments show that replacing Conv3 with Conv1s scarcely degrades the performance but can enhance runtime efficiency. Despite channel pruning boasting fewer FLOPs, its runtime performance falls short of kernel pruning.

Stage 2. Bias Pruning. After kernel pruning, they obtain a subnet with a modified RLFB network. They preserve the biases of the first conv and ESA module, then retrain the subnet by removing the remaining biases thus further enhancing runtime efficiency.

Training Details. The model is trained on LSDIR dataset[46] and they use RLFN (the official baseline model of NTIRE2024) as their basemodel. The training HR patch size is set as 256×256 with data augmentation such as rotation and horizontal flip in order to enhance the comprehensive ability of the model. They set the batch size as 64 in the training process with total of 500 epochs. The model is trained by minimizing L2 loss with Adam optimizer. The initial learning rate is set to 2e-5 and the learning rate is decayed by half at 100 epochs.

4.12. ECNU_MViC

Method. As shown in Fig. 21, they propose an intermittent feature aggregation network named IFADNet. The architecture comprises of three parts: the shallow feature extraction, the deep feature extraction based on alternating BFEB blocks and RFMB blocks, and the reconstruction stage.
They employ a single 3× 3 convolution to extract the shallow feature $F_s \in \mathbb{R}^{C \times H \times W}$ in the first stage H_F:

$$F_s = H_F(I_i), \tag{16}$$

where I_i, C, H, W are the input image, the embedding channel dimension, height and width of the input, respec-

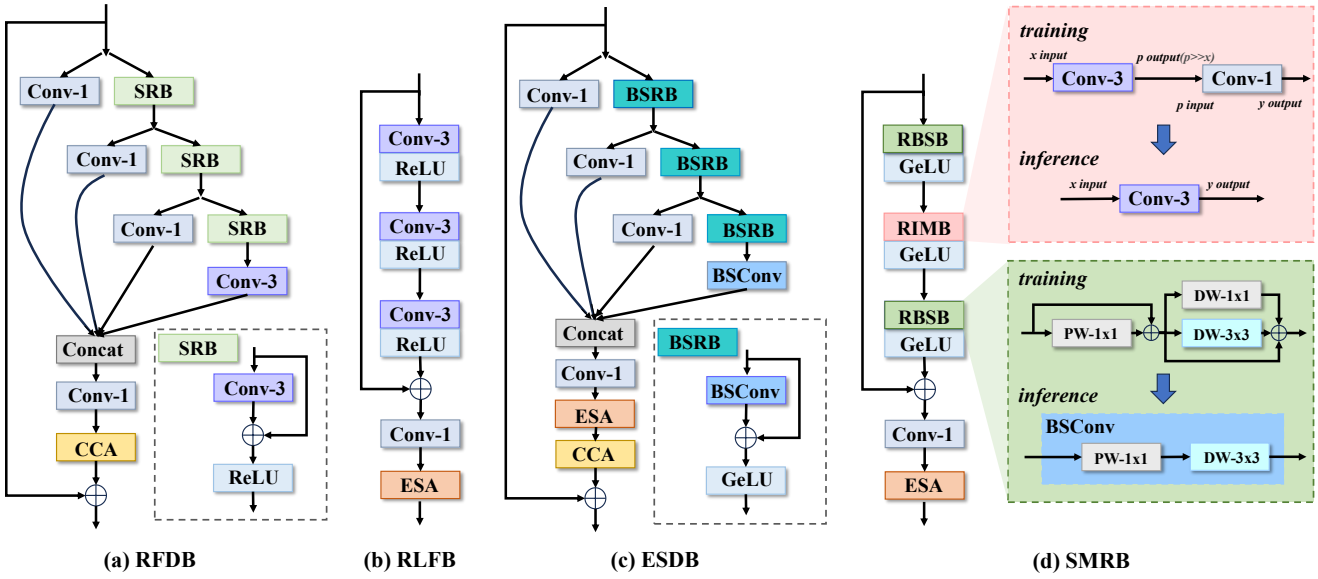


Figure 19. *Team LeESR:* (a) Residual feature distillation block (RFDB). (b) Residual local feature block (RLFB). (c) Efficient separable residual block (ESDB). (d) The proposed Separable-Mixable Residual Block (SMRB).

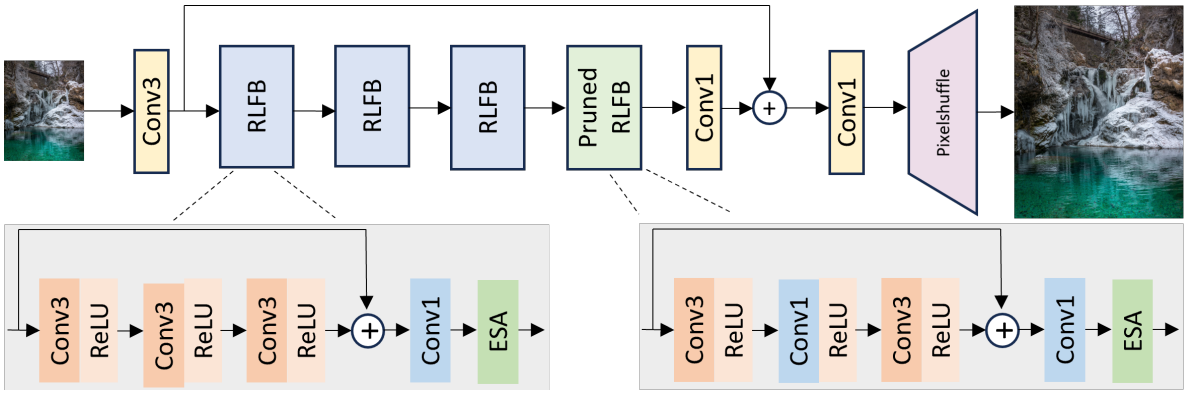


Figure 20. *Team AdvancedSR*: The overall architecture of AdvancedSR.

tively.

During the second stage, six intermittent blocks are used to extract the deep feature $F_d \in \mathbb{R}^{C\times H\times W}$:

$$F_d = H_D(F_s). \tag{17}$$

Specifically, H_D consists of an alternating blueprint feature extraction block (BFEB) and a parameterized feature modulation block (RFMB). The details of BFEB and RFMB will be introduced in the next paragraph. By using F_s and F_d as inputs, the high-quality image I_r is generated in the reconstruction stage denoted by H_R as:

$$I_r = H_R(F_s + F_d), \tag{18}$$

where H_R involves a single 3×3 convolution followed by a pixel shuffle operation [69].

Inspired by the blueprint shallow residual block [48], they designed a blueprint feature extraction block to reduce the computation time, which did not significantly reduce model performance. As shown in Fig. 21 (a), the BFEB contains three blueprint shallow blocks (BSB) which contain a 1×1 point-wise convolution with a 3 × 3 depth-wise convolution followed by GELU activation. The reparameterized feature modulation block (RFMB) consists of three reparameterized convolution blocks (CB) and an enhanced spatial attention block [36] is employed to extract and modulate the deep feature fully. The detailed structure is illustrated in Fig. 22. They further observe that the intermittent setting of blocks significantly reduces model complexity while not largely impairing model effectiveness. At the end of the second stage, the extracted features of each block are concatenated and aggregated using two convolutions.

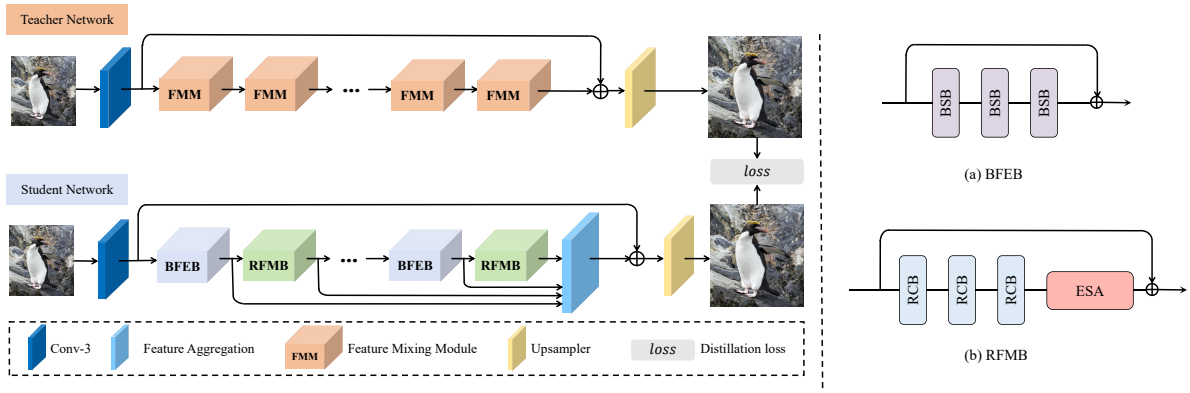


Figure 21. *Team ECNU_MViC:* The pipeline of IFADNet. (a) Structure of RCB. (b) Structure of RFMB.

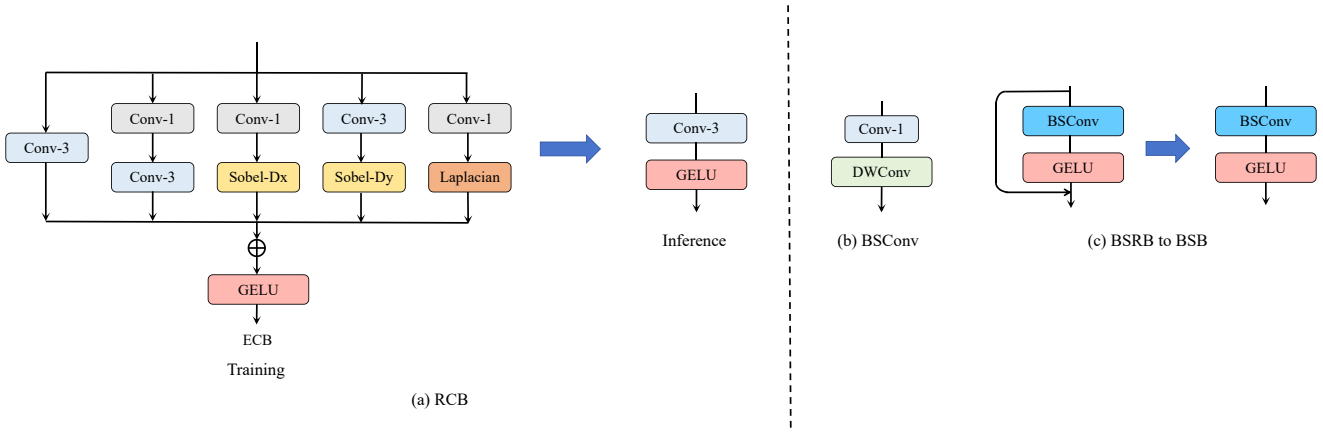


Figure 22. *Team ECNU_MViC:* (a) Structure of BFEB. (b) Structure of BSConv. (c) From BSRB to BSB.

Reparameterization has shown a strong ability to improve the feature presentation. Different from the reparameterization module design of high-level tasks, they design an isotropic edge-oriented convolutional block in their model. As shown in Fig. 22(a), the Sobel-Dx and Sobel-Dy employ isotropic Sobel functions to enhance the network's representation capabilities. For inference, the output is computed in a simplified 3×3 convolution, which significantly reduces computation cost.

Training Details. They use DIV2K [1] and the first 10K images of LSDIR [46] to train their model. The training dataset is augmented with horizontal flips and 90-degree rotations. Knowledge distillation is applied to improve the model performance. They use the large version of pre-trained SAFMN[70] as their teacher model.The student model IFADNet has 6 blocks(3 BFEB and 3 RFMB). The channel of their network is 36. The training details are as follows:

- Training from scratch. The HR patch size is 256. The mini-batch size is set to 64. The model is trained by minimizing L1 loss and distillation loss (also L1 loss) with Adam optimizer [35]. The initial learning rate is set to 2×10^{-3} and halved at $\{100k, 500k, 800k, 900k, 950k\}$-iteration. The total number of iterations is 1000k.
- Finetuning with larger patches. The HR patch size is set to 640. The model is finetuned with MSE loss. Other settings are the same as in the previous step.

4.13. HiSR

Method. They propose the SlimRLFN for the efficient super-resolution task. The network architecture is inspired by the design of RLFN[36], while fully exploring the capacity of reparameterizable convolution, light distillation, and iterative model pruning. The whole architecture is shown in Fig.23, which mainly consists of six SRLFB modules and a pixel shuffle module. Reparameterizable convolutions are utilized in the SRLFB module, aiming to improve the super-resolution capability without introducing any additional parameter overhead during the inference stage. Meanwhile, the network is optimized by the pixel-wise loss such as charbonnier loss or L2 loss, along with the distillation loss provided by a light but efficient teacher model. Last but not

least, they use iterative pruning to shrink the model size while maintaining the promising performance at the last training stage.

Training Details. They choose DIV2K, Flickr2K, and LSDIR datasets as their training datasets, and they augment them with horizontal/vertical flips and rotations during the training stage. They set the batch size as 64 for all training stages and other hyperparameters such as patch size or learning rate are determined by the specific training stage. The whole training process is summarised as follows.

1) Training the teacher model. They choose the large RLFN as their teacher model. They set the patch size as 256×256, and use charbonnier loss and Adam optimizer for optimization. They train the teacher model for 100 epochs and set the initial learning rate as 1e-3. The learning rate decay follows cosine annealing with T_{max} as 100 and eta_{min} as 1e-5.
2) Training the SlimRLFN with light distillation. They set the patch size as 256×256, and use two loss functions for training. The first one is the regular charbonnier loss with ground truth HR image, and the second one is the charbonnier loss between SlimRLFN's output and the teacher's output. They also choose the Adam optimizer for optimization. They train the SlimRLFN model for 100 epochs, and set the initial learning rate as 1e-3. The learning rate decay follows cosine annealing with T_{max} as 100 and eta_{min} as 1e-5. Then they repeat this stage two more times without distillation loss, and the pretrained model is adopted from the last stage.
3) Training the SlimRLFN with a larger patch size progressively. They set the patch size as {384×384, 512×512}, and set the initial learning rate as {5e-4, 2.5e-4} respectively. Each stage's pretrained model is adopted from the last stage, and they train the model under the same patch size three times in total. The other details are the same as before.
4) Training the SlimRLFN with the patch size of 640×640. They use the L2 loss in this stage and set the initial learning rate as 1e-4. They also adopt cosine annealing with T_{max} as 100 and eta_{min} as 1e-6 for learning rate decay.
5) [Optional] SlimRLFN pruning stage. After training from the above several epochs, they adopt the iterative pruning for the SlimRLFN which has obtained promising performance.

4.14. MViC_SR

Method. As shown in Figure 24, this team propose a network with lightweight spaced attention (LSANet). The architecture of LSANet consists of the following parts: the shallow feature extraction, the deep feature extraction based on spaced local feature extraction module (LFEM) and reparamaterized spaced attention Block (RSAB), and the reconstruction.

Given the LR input I_{LR}, a single 3×3 convolution is applied to extract the shallow feature $F_0 \in \mathbb{R}^{C\times H\times W}$ in the first part:

$$F_0 = \mathrm{Conv}\left(I_{LR}\right), \tag{19}$$

where I_{LR}, C, H, W are the input LR image, channel dimension, height and width of the input, respectively.

In the second part, they use six alternate blocks to extract the deep feature $F_d \in \mathbb{R}^{C\times H\times W}$:

$$F_d = H_D(F_0). \tag{20}$$

Specifically, H_D is comprised of local feature extraction module (LFEM) and reparamaterized spaced attention Block (RSAB). By taking F_s and F_d as inputs, the HR image I_{HR} is reconstructed with an upsampler as:

$$I_{HR} = H_{RC}(F_0 + F_d), \tag{21}$$

where H_{RC} is the reconstruction module involves a single 3×3 convolution followed by a pixel shuffle operation [69].

Previous methods mostly use several plain convolutions to extract features, which is inefficient with soaring computational complexity. Inspired by the blueprint shallow residual block [48], they design an efficient yet effective local feature extraction module (LFEM) to alleviate computing burden while largely maintaining model performance. As illustrated in Figure 24(a), the LFEM contains three efficient shallow blocks (ESB) which include a 1×1 point-wise convolution with a 3×3 depth-wise convolution followed by GELU activation [32].

Consequently, they propose a reparamaterized spaced attention Block (RSAB) to preserve the representation ability to the adjacent blocks, which is composed of an efficient shallow block and two reparameterized convolution blocks (RCB) followed by an enhanced spatial attention block (ESA)[36] and a convolution. The spaced ESA blocks only used in RSAB are employed for comprehensive extraction and modulation of deep features. They use a 1×1 convolution after the ESA block to further refine the weighted feature and capture local patterns. The detailed structure is shown in Figure 25. Intermittently configurating attention blocks substantially lower model complexity without a noticeable performance drop. At the end of this part, features extracted from all blocks are concatenated and aggregated using two convolutions.

Reparameterization [24, 88] has proven effective in enhancing feature representation without introducing additional computational overhead. Different from the reparameterization module design of high-level tasks, they design an isotropic edge-oriented convolutional block in their model. As shown in Figure 25(a), the Sobel-Dx and Sobel-Dy employ the isotropic Sobel function to improve the representation capabilities of their model. During the inference

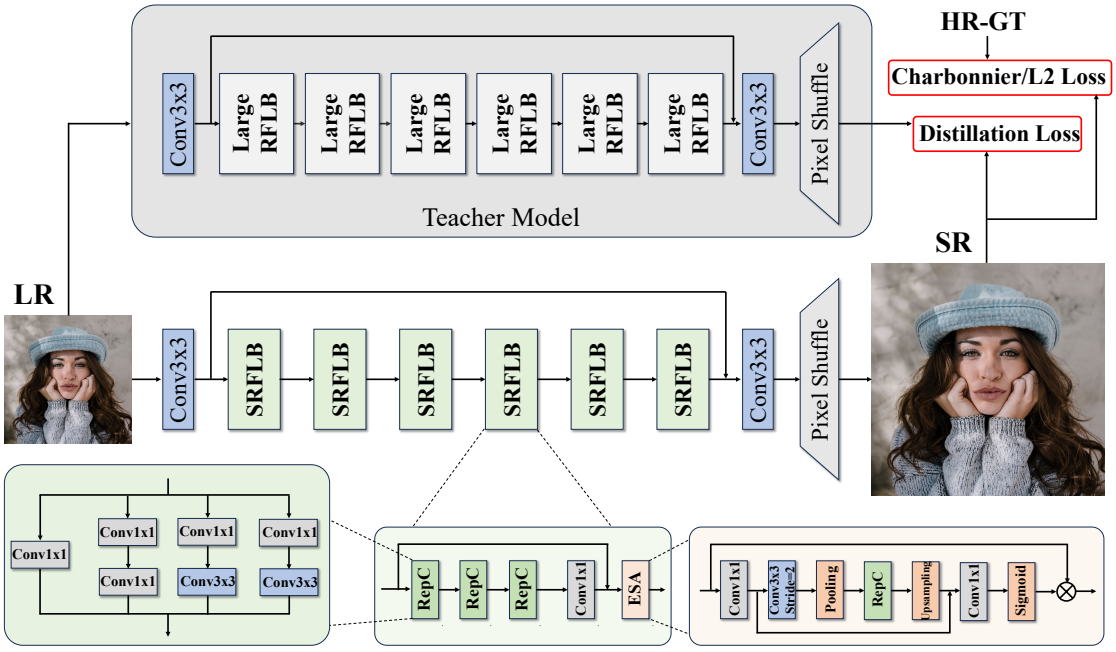


Figure 23. *Team HiSR*: Network architecture of SlimRLFN.

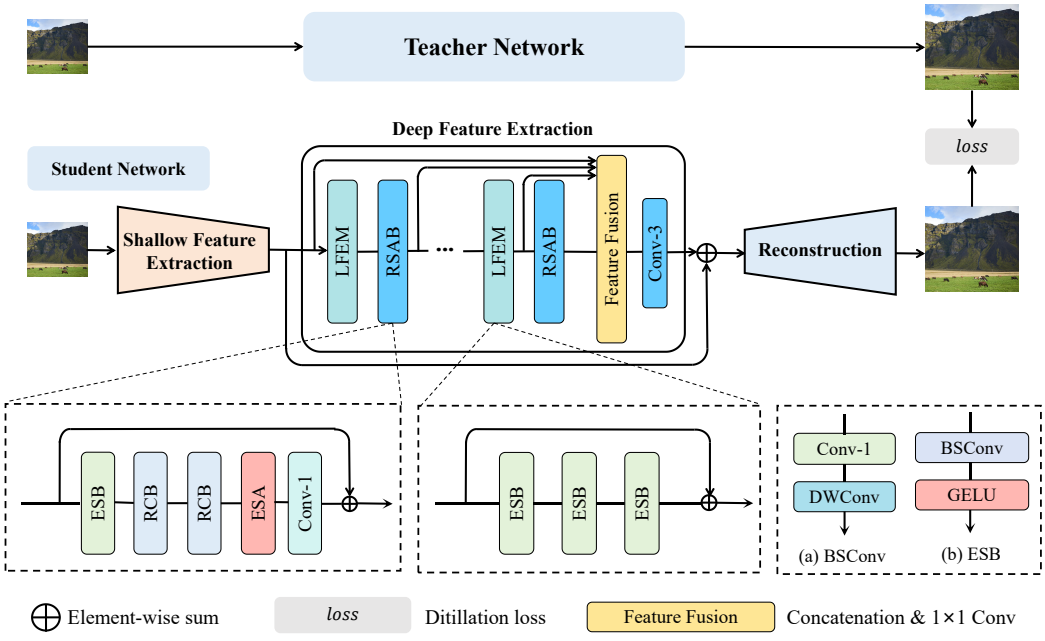


Figure 24. *Team MViC_SR:* The pipeline of LSANet. (a) Structure of BSConv. (b) Structure of ESB.

phase, all branches are combined to a simplified 3×3 convolution, which significantly reduces computation cost.

Training Details. They apply two stages to train their network on DIV2K and the first 10K data of LSDIR. They randomly augment the input with a flip and 90-degree rotation to enhance the robustness of the network. They design a teacher-student distillation strategy for training, which makes the large version of SAFMN [70] as a teacher and the proposed network as a student. Three LFEM and RSAB blocks are stacked alternately in their student network, and the feature channel is 36. The mini-batch size is fixed to 64 in all stages. The details of the two training stages are as

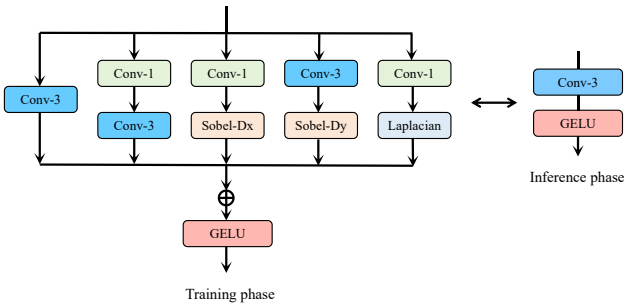


Figure 25. *Team MViC_SR:* The structure of RCB.

follows:

- **Stage 1.** They minimize the L1 loss (student prediction and ground-truth) and distillation loss (student prediction and teacher preditction) to optimize the student network by Adam optimizer [35] for $1000k$ iterations. The initial learning rate is set to 2×10^{-3}, which will be halved at $100k$, $500k$, $800k$, $900k$, $950k$. The HR patch size is 256 during this stage.
- **Stage 2.** They initialize the weight of the student network with the pre-trained student in Stage 1. They enlarge the HR patch size to 640 and minimize MSE loss instead of L1 loss.

4.15. LVTeam

Method. Accurately inferring and reconstructing the fine details missing in the LR images based on the learned models of image textures and structures, while maintaining the efficiency of the inference process, poses a challenging task. Since the provided baseline RLFN [36] is already lightweight, they wanted to further explore the limits of the RLFN model while keeping the performance (26.99 dB). For this purpose, they made two improvements based on the RLFN and formed a novel architecture LightRLFN. Firstly, a shortcut connection between the inputs and the model outputs. To achieve this, they perform $4\times$ bilinear interpolation on the input image to match the resolution of the output. Secondly, they reduced the width of the model (46 channels to 36 channels). As a result, the designed LightRLFN model is lightweight and efficient enough, as shown in fig.26.

Training Details. The proposed architecture is based on PyTorch 2.2.1 and an NVIDIA 2080Ti with 11G memory. They set 400 epochs for training with batch size 32, using AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for optimization. The initial learning rate was set to 0.001, and cosine annealing was used for learning rate adjustment. Regarding the use of the DIV2K [1] and LSDIR [46] datasets, they first copied the DIV2K dataset to 85 times its original number (i.e., 800×85), and then merged it with the LSDIR dataset to form the final training set, which contains a total of 152,991 images. For data augment, they first randomly crop the LR image to 96×96 (the corresponding HR resolution is 384×384) and then perform horizontal flip with probability 0.5. Regarding the loss function, they use the L1 loss and the frequency domain reconstruction loss with 0.1 weights.

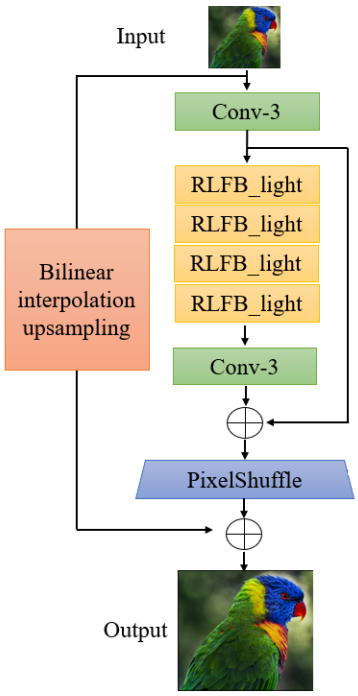


Figure 26. *LvTeam*: The architecture of LightRLFN.

4.16. Fresh

Method. To further reduce the network's parameter count and enhance its efficiency, they propose Depth residual local feature network for super-resolution (DepthRLFN), which is modified from the RLFN [36] As illustrated in Figure 27, they employ depth-wise separable convolutions to replace the conventional convolutions in the RLFB structure of the baseline, and they add the low-resolution (LR) image subjected to bilinear interpolation before the network's output. This approach not only better preserves the details of the original image but also reduces the model's parameter count and enhances the final image quality. Moreover, to mitigate the discrepancies observed between global operation behaviors during training and testing phases, which adversely affect super-resolution performance, their approach integrates the Test-time Local Converter (TCL) architecture as introduced in [36].

Training Details. They trained DepthRLFN using a total of 85,791 image pairs from the DIV2K and LSDIR [46] datasets on PyTorch 2.2.1 and an NVIDIA A40 with 40G memory. The training process was divided into two stages, with DepthRLFN comprising 4 DRLFBs and having

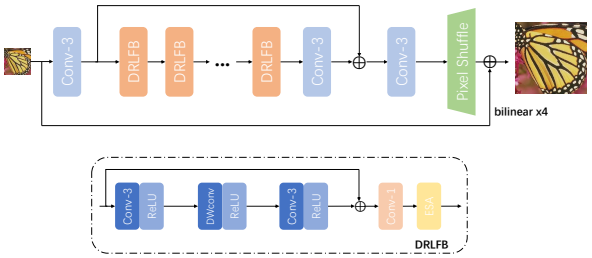


Figure 27. *Team Fresh*: The architecture of DepthRLFN.

a channel count of 64. For data augment, they first randomly crop the image to 192×192 and then perform a horizontal flip with probability 0.5. They set 500,000 iterations for training, using AdamW with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for optimization. They set the initial learning rate to 1×10^{-4} and decay the learning rate by 0.5 every 150, 000 iterations. Additionally, they use L1 loss and Frequency loss for the training phase. The cropped HR image size is 256×256 and the mini-batch size is set to 64 for the finetuning stage. The DepthRLFN is trained by L2 loss for a total of 800,000 iterations. They set the initial learning rate to 1×10^{-5} and decay the learning rate by 0.5 every 250,000 iterations.

4.17. Lanzhi

Method. In order to facilitate more stable model training, they employ an expedited convergence rate and simultaneously enhance the generalization capability of the model. As shown in Figure 28, they introduce a batch normalization layer after the first CONV3+RELU layer in the residual local feature blocks (RLFBs) within the Residual Local Feature Network (RLFN) architecture [36]. By incorporating BN before each convolution, every convolutional layer can benefit from the normalized input features. Additionally, the final convolutional layer can make certain adjustments to pixel biases before incorporating them into the next branch, thereby enhancing performance to a certain extent [54]. Ensemble learning is a powerful

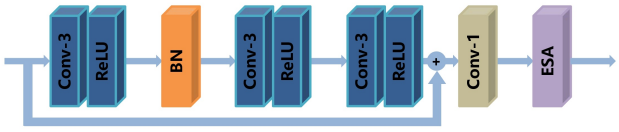


Figure 28. *Team Lanzhi:* RLFB_BN: Residual local feature block with embedded batch normalization layer.

technique to improve model performance and robustness. In the inference stage, they utilize model fusion as a means of its implementation. Specifically, they employ a weighted averaging strategy to fuse the prediction results of the models. By adjusting the weight values, they can flexibly

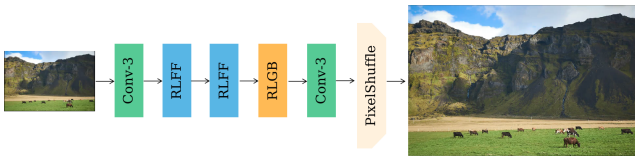


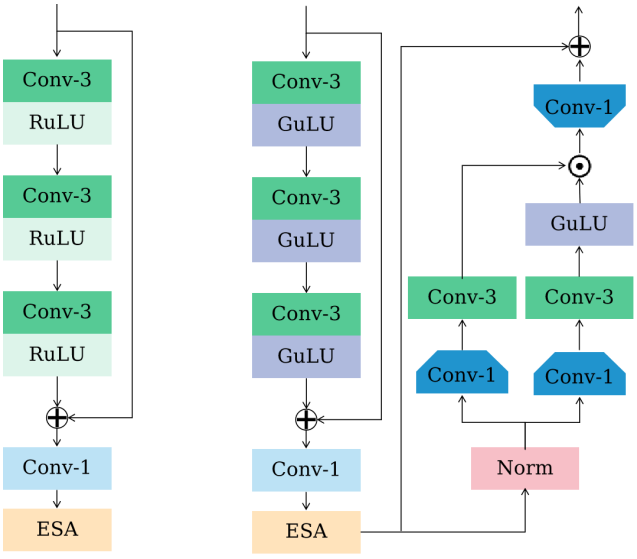
Figure 29. *Team Supersr:*The overall network architecture of our RLFF.

control the influence of each model. Through experiments, they find that this methodology can significantly improve the effect of image super-resolution. They conducted a series of experiments and comparisons between RLFN_BN and the original RLFN model. Through modifications to the RLFN architecture, they find that the runtime, FLOPs, and number of parameters remained comparable. On top of that, with the application of ensemble learning strategy, their approach achieved a certain improvement in the peak signal-to-noise ratio (PSNR) while maintaining comparable structural similarity (SSIM) scores, further demonstrating the feasibility and effectiveness of their approach for efficient image super-resolution.

Training Details. During the training phase, they utilize the training data from the DIV2K dataset, comprising 800 pairs of low-resolution and high-resolution images, as well as a subset of the LSDIR dataset [46], consisting of randomly selected 8,500 pairs of low-resolution and high-resolution images. Before training, they pre-process the images by decoding all PNG files and saving them as binary files. The training process employs the Adam optimizer. On the other hand, they employ a learning rate decay strategy to optimize training stability and adapt to changes in the data distribution. The initial learning rate is set to 2×10^{-4}. The total number of epochs is 1000. The other training configurations, such as the batch size is set as 16 and the RGB range is set as 255. Additionally, they utilize the L1 loss function as it tends to generate sharper images compared to the L2 loss. The implementation of their approach is carried out using the PyTorch framework.

4.18. Supersr

Method. It is designed based on the baseline approach RLFN [36]. As depicted in Fig. 29, they replace the original RLFB block with their Residual Local Feed Forward (RLFF) block, which is efficient for super-resolution tasks. The detailed information will be introduced in the next section. The residual Local Feed Forward (RLFF) block is the core block in our framework. Firstly, they use the GeLU as the activation function. As presented in Restormer [87], the Gated Dconv Feed-Forward Network, by managing the flow of information across the hierarchical levels within the pipeline, empowers each level to concentrate on intricate



(a) RLFB (b) RLFF

Figure 30. *Team Supersr:* (a) The original RLFB module. (b) Our RLFF module.

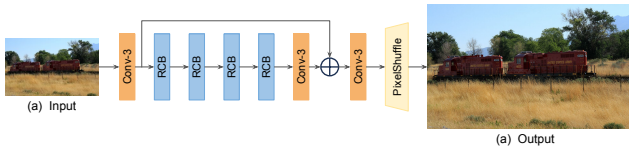


Figure 31. *Team MeowMeowMeow:* Reparameterized Convolutional Network (RCN).

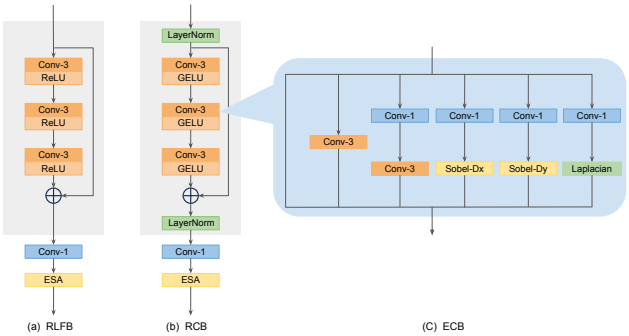


Figure 32. *Team MeowMeowMeow:* Reparameterized Convolutional Block (RCB) and Edge-oriented Convolutional Block (ECB).

details that complement those addressed by other levels. Inspired by this, they put it into the basic module as a subsequent processing of features to improve the feature learning capability of the network further.

Training Details. Two datasets provided are used for training: DIV2K [1] and LSDIR [46]. They use the augmentation strategy of vertical/horizontal flips and 90-degree rotation. They adopt the Adam optimizer algorithm, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. The batch size and patch size are set to 64 and 256. All experiments are conducted on a single NVIDIA RTX 3090 GPU. Only the L1 is used during the training process. It consists of three phases: 1. patch size/learning rate: 256/2e-3 for 100, 000 iterations; 2. patch size/learning rate: 384/1e-3 for 50, 000 iterations; 3. patch size/learning rate: 512/5e-4 for 50, 000 iterations.

4.19. MeowMeowMeow

Method. Their model is designed based on the baseline approach RLFN [36], as depicted in Figure 31. They replace the RLFB block with a modified version, which they call Reparameterized Convolutional Block (RCB), incorporate two extra Layer Norm layers, and replace ReLU with GELU. Inspired by ECBSR [88], they employ Edge-oriented Convolutional Block (ECB) as their reparameterization module to replace the original 3×3 convolution layer. In the training phase, the ECB module adds five extra branches to the original 3×3 convolution layer as shown in Figure 31. In the inference phase, the ECB module can be converted into a 3×3 convolution layer without any computational overhead.

Table 2. *Team MeowMeowMeow:* Hyperparameters for different training stages.

	Stage 1	Stage 2	Stage 3	Stage 4
PS	196	256	384	512
LR	$2e-3$	$2e-3$	$1e-3$	$2e-4$
Sche	None (fixed LR)	CosineAnnealingRestartLR	CosineAnnealingRestartLR	CosineAnnealingRestartLR
Iter	200,000	50,000	50,000	20,000
Loss	1*L1	1*L1 + 0.001*PSNR	1*L2 + 0.001*PSNR	1*L2 + 0.001*PSNR

Training Details. They use DIV2K [1] and LSDIR [46] datasets for training, as well as vertical/horizontal flips and 90-degree rotation as data augmentation. AdamW is leveraged as the optimizer with a weight decay of 0.01, $\beta_1 = 0.9$, and $\beta_2 = 0.99$. EMA decay is set to 0.999, and batch size is set to 128. They use a warmup iteration of 2000. All training was conducted on RTX A6000 GPU. They used three functions, including standard L1 and L2 loss, and our proposed PSNR loss. The whole training procedure contains four stages. Inspired by PEFT techniques [30, 31], they employ LoRA with rank $r = 4$ to the last stage. Tunable hyperparameters include patch size (PS), learning rate (LR), scheduler (Sche), training iterations (Iter), and loss, across the four stages. They summarize these hyperparameters in Table 2 and Table 3.

Table 3. *Team MeowMeowMeow:* Hyperparameters of CosineAnnealingRestartLR scheduler for stage 2, stage 3 and stage 4.

Parameter	Stage 2	Stage 3	Stage 4
Periods	[12500, 12500, 12500, 12500]	[12500, 12500, 12500, 12500]	[20000]
Restart_weights	[1, 0.9, 0.8, 0.7]	[1, 0.9, 0.8, 0.7]	[1]
Eta_min	$1e-7$	$1e-7$	$1e-7$

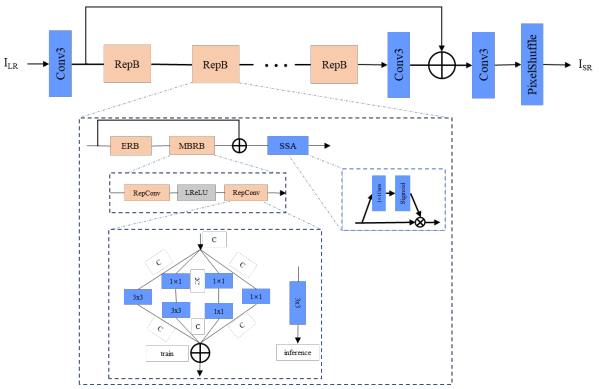


Figure 33. *Team Just Try:* Overall structure of ERRN

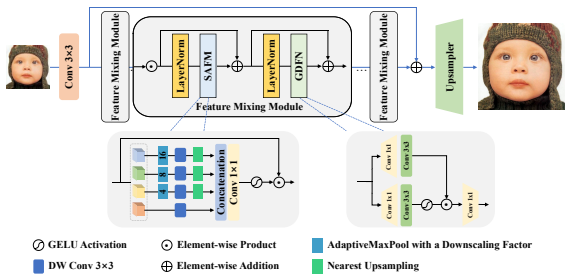


Figure 34. *Team VPEG_E*: The overall network architecture of our proposed EGFMN.

4.20. Just Try

Method. Inspired by RLFN[36], FMEN[24], and DBB[19], they designed a Enhanced Reparameterize Residual Network(ERRN) as shown in the Fig.33. The reparameterize block (RepB) consists of ERB[24], multi-branch reparameterize block (MBRB), and simple spatial attention(SSA). The ERB and MBRB are both reparameterization blocks, which use a complex RepConv structure during the training phase and convert to a 3×3 convolutional during inference. Every RepB uses SSA to enhance the output feature. In ERRN, they use six RepBs, and the number of feature channel C is set to 40.

Training Details. They use DIV2K, LSDIR, and Flickr2K datasets as training datasets. For each mini-batch, they randomly crop 16 patches from the LR images with the size of 64×64. They use a cosine annealing learning scheme, the learning rate is initialized as 2×10^{-4} and the minimum learning rate is 1×10^{-7}, a total of 1000k iterations, the period of cosine is 250k iterations. They use Adam optimizer withβ_1= 0.9,β_2=0.99. The loss function is L1 loss. Then fine-tuning on the same datasets, the LR size is set to 128×128, the initialized learning rate is 1×10^{-4}, a total of 400k iterations, and the period of cosine is 100k iterations. Other settings are the same as above. Final L2 loss is used for fine-tuning on the same datasets. Other settings are the same as above.

4.21. VPEG_E

Method. The VPEG_E team introduces an enhanced gated feature modulation network (EGFMN) for efficient SR, which is modified from the SAFMN [70]. To make the EGFMN more lightweight, the VPEG_E team replaced the used convolutional channel mixer (CCM) with the gated-dconv feed-forward network [87] (GDFN). Figure 34 shows that EGFMN first uses a convolution layer maps the input image to feature space and employs 7 feature mixing modules (FMMs) for learning discriminative feature representation, where each FMM block has a spatially-adaptive feature modulation (SAFM) layer and a GDFN module. To recover the HR target image, the VPEG_E team introduced a global residual connection to learn high-frequency details and employ a lightweight upsampling layer for fast reconstruction, which only contains a 3×3 convolution and a pixel-shuffle [69] layer.

Training Details. The VPEG_E team trains the proposed EGFMN on the LSDIR dataset. The cropped LR image size is 640×640 and the mini-batch size is set to 64. The EGFMN is trained by minimizing L1 loss and the frequency loss [13] with Adam optimizer for total of 600, of 000 iterations. The VPEG_E team set the initial learning rate to 2×10^{-3} and the minimum one to 1×10^{-6}, which is updated by the Cosine Annealing scheme [60].

4.22. BU-ESR

Method. Inspired by [26], this team employs a knowledge distillation method, as shown in Fig.35. Initially, the team utilizes a substantial teacher network equipped with a Hybrid Attention Transformer backbone [10]. This teacher network is specifically designed to learn and distill high-quality features. Its augmented parameter capacity is crucial for capturing nuanced feature information, thereby providing a robust foundation for supervising the student network. In the second stage, this team further trains the student network under the supervision of the pretrained features from the teacher network to enhance its performance. Considering that architectures based on self-attention significantly increase the model's parameter count [27, 28], their work is meticulously designed around the Residual Feature Distillation Network (RFDN) model, which serves as the backbone for the student network. The RFDN framework is notable for its lightweight structure, consisting of four pivotal components: an initial feature extraction convolution, multiple Residual Feature Distillation Blocks (RFDBs) stacked sequentially, a feature fusion layer, and

a final reconstruction block. The process begins with a 3×3 convolutional layer that extracts coarse features from the input low-resolution (LR) image. Then, the core of the RFDN, comprising four RFDBs, is sequentially employed by the team to progressively refine these features. The refined features from each RFDB are amalgamated using a 1×1 convolution layer, followed by an additional 3×3 convolutional layer to enhance the smoothness of the aggregated features. Finally, the high-resolution image is produced through a pixel shuffle operation by the team.

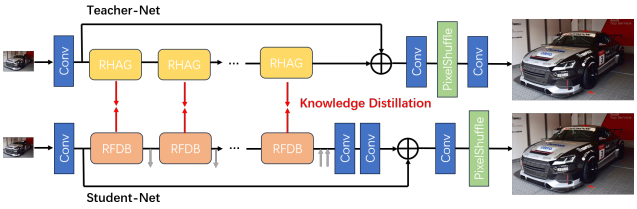


Figure 35. *Team BU-ESR:* Their teacher network is designed based on a Hybrid Attention Transformer backbone, while their student network is implemented using an RFDN network. The dilation loss provides latent space feature supervision to enhance the lightweight RFDN's image super-resolution results.

Training Details. This team utilized two datasets, DIV2K and LSDIR, for their experiments. They augmented the training dataset with geometric transformations, including vertical and horizontal flips and 90-degree rotations, to enhance the model's comprehensive abilities. During the Teacher Model Training Phases, the model was initially trained from scratch. High-resolution (HR) patches of size 192×192 were randomly cropped from HR images, with a mini-batch size of 16. The training employed the smooth L1 loss function and the Adam optimizer. Considering the impact of the choice of learning rate on the results [25], the team made multiple adjustments to the initial learning rate and finally set it at an initial learning rate of 2×10^{-4}. The training spanned 40,000 epochs. Subsequently, the student model was fine-tuned using pre-trained weights, with the initial learning rate reduced to 1×10^{-4}, and the training ran for an additional 100 epochs. This stage retained the same settings as the initial phase but incorporated a loss function that includes smooth L1, MS-SSIM loss, perceptual loss, and teacher-student dilation loss.

4.23. Lasagna

Method. This team propose an efficient enhanced residual network (EERN) for efficient image super-resolution, the primary architecture of which is illustrated in the Fig.36. Although the EDSR[51] network achieves high performance in the field of single-image super-resolution, its network structure is relatively bulky. Therefore, they have adopted a strategy of reducing the number of blocks to lighten the model. However, merely decreasing the number of blocks are not able to meet the performance standards set by the competition. Therefore, they have made certain adjustments to its structure by incorporating an ESA[52] after each block, and to reduce the parameters, they have eliminated several convolutional layers. Additionally, they appended a convolutional layer at the end of the ERB block for the purpose of fine-tuning. This convolutional layer was not included at the onset of the training process but was incorporated during the final fine-tuning phase to participate in the computation. Simultaneously, to increase inference speed and reduce GPU memory usage, this team have removed the residual connections in the RB modules of EDSR. Experiments show that removing residual connections has minimal impact on the module's performance. Moreover, considering that excessive changes in the number of feature channels are detrimental to performance in lightweight networks, they opted to replace two successive shuffle2 operations with a single shuffle4 operation to avoid the final convolutional reduction of channels from 256 to 3. Lastly, they trained the network using a knowledge distillation approach. Due to the mismatch in the number of feature channels between the teacher model and the student model, they did not utilize the loss of feature maps. Instead, the loss was calculated based on the outputs of both models.
Training Details. The number of ERB modules and the number of its feature channels were set to 4 and 84, respectively. They trained a total of 1700 epochs to bring the model to convergence. The process was divided into three stages. In the first stage, they initiated the learning rate at 1e-4, employing a cosine annealing strategy to decrease the learning rate to 5e-7 by the 400th epoch. During this cycle, they utilized L1 loss, with the dataset limited to DIV2K. In the second stage, the starting learning rate was set to 1e-5, again using a cosine annealing method to reduce the learning rate to 5e-7 by the 300th epoch. Moreover, in this cycle, PSNR loss was employed for fine-tuning, and the dataset was expanded to include both DIV2K and folders named 0001000 to 0010000 from the LSDIR dataset. In the final stage, the convolutional layer was introduced and initialized to zero. This stage comprised a total of 1000 epochs, with all other settings being identical to those of the second stage.

4.24. BlingBling

Method. In this work, the BlingBling team propose Distillated Vision Mamba SR (DVMSR), a novel lightweight Image SR network that incorporates Vision Mamba and a distillation strategy. The framework of their DVMSR is illustrated in Fig.37. It consists of three main modules: feature extraction convolution, multiple stacked Residual State Space Blocks (RSSBs), and a reconstruction module. The feature extraction convolution employs a 3×3 convolu-

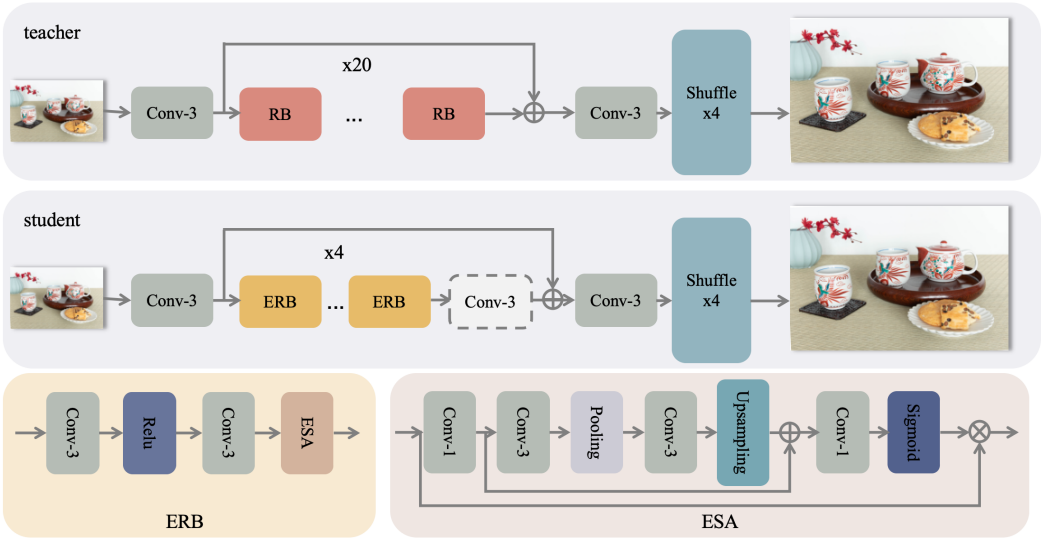


Figure 36. *Team Lasagna:* The structure of the proposed EERN.

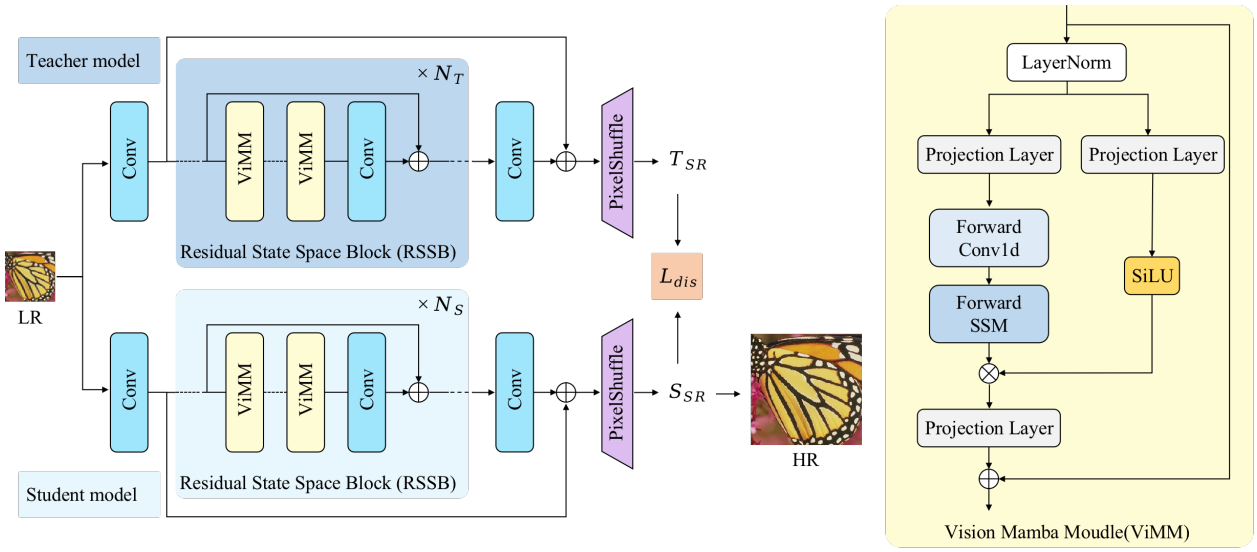


Figure 37. *Team BlingBling:*The overall network architecture of our DVMSR, as well as Vision Mamba Module (ViMM).

tional layer to extract shallow features. Subsequently, deep features are extracted by several stacked Vision Mamba Modules (ViMMs) [29, 90], which leverage the long-range modeling capability of Mamba to effectively reduce computational complexity. A 3×3 convolutional layer refines the features extracted from the ViMMs at the end of each RSSB. The final stage involves the aggregation of shallow and deep features using a long skip connection, followed by the use of a sub-pixel convolution layer to upsample the feature and reconstruct the high-quality image. The teacher and student models have similar network structures, with variations in the number of RSSBs, ViMMs, and channels.

Training Details. They employ DIV2K [71] and LSDIR [46] to construct the training dataset. The High-Resolution (HR) images are cropped to 256×256 patches for the training procedure. They use $\mathcal{L}_1$ loss with the Adam optimizer for the network optimization. The initial learning rate is set to 2×10^{-4}. The total number of iterations is 500k. They adopt a multi-step learning rate strategy, where the learning rate will be halved when the iteration reaches 250,000, 400,000, 450,000, and 475000, respectively. In the teacher learning phase, they utilize the DIV2K dataset with 2K resolution to train the teacher network, which comprises 6 RSSB and 2 ViMM blocks with 180 channels. The teacher

network can learn the rich representation knowledge for the distillation stage. During the distillation training phase, they merge the DIV2K and LSDIR datasets for the student network, which contains 4 RSSB and 2 ViMM blocks with 60 channels. The teacher network remains fixed during this process. They employ $\mathcal{L}_1$ loss to align the student network feature with the teacher network feature. This process can transfer the knowledge of the teacher network to the student network. Formally,

$$\begin{aligned} \mathcal{L}_{out} &= \lambda_{dis}\mathcal{L}_{dis} + \lambda_1\mathcal{L}_1, \\ \mathcal{L}_{dis} &= \|\mathcal{T}(I_{LR}) - \mathcal{S}(I_{LR})\|_1, \\ \mathcal{L}_1 &= \|I_{HR} - \mathcal{S}(I_{LR})\|_1, \end{aligned} \tag{22}$$

where λ_{dis} and λ_1 represents the coefficient of the $\mathcal{L}_{dis}$ loss function and the coefficient of the $\mathcal{L}_1$ loss function, respectively. They are set 1. $\mathcal{T}$ represents the function of our teacher network and $\mathcal{S}$ denotes the function of our proposed network. I_{LR} and I_{HR} are the input LR images and the corresponding ground-truth HR images, respectively. More information of $\mathcal{L}_{dis}$ can be seen from Fig.37.

4.25. Minimalist

Method. Given the inherently lightweight nature of the baseline Residual Local Feature Network (RLFN), the Minimalist aims to enhance its capabilities by introducing strategic modifications to its layers and incorporating additional methods aimed at enhancing its capacity to capture wider spatial and more semantically rich information. A pivotal contribution to their architectural advancements lies in the integration of a novel module termed Efficient Multi-head Spatial Attention (MHSA), alongside a refined iteration of the Residual Local Feature Block (RLFB), drawing inspiration from the foundational work presented in RLFN [36]. The comprehensive layout of their team proposed methodology is depicted in Figure 38. The proposed Efficient Multi-head Spatial Attention (MHSA) mechanism introduces a novel approach by partitioning feature maps into multiple heads. Each head is dedicated to capturing spatial dependencies within distinct groups, thereby facilitating a more thorough and efficient representation of features. This concept of spatial attention draws inspiration from prior work such as [36], while the integration of multi-head convolution mixture-normalization is similarly influenced by the findings in [74]. The incorporation of spatial attention across various groupings within different heads enables the model to discern diverse features at different group levels. Subsequently, they leverage the distilled knowledge from multi-head learning and employ layer normalization to facilitate effective learning with enhanced gradient smoothness. Additionally, their Efficient Residual Local Feature Block (ERLFB) iteratively enhances feature representations through the utilization of local spatial attention and channel-wise attention mechanisms, dynamically assigning priority to specific spatial and channel dimensions. To manage the potential increase in parameters resulting from the introduction of new layers, they integrated group convolution, effectively balancing the trade-off between parameter efficiency and performance, thereby maintaining inference speed comparable to that of the RLFB block.

Training Details. The Low-Resolution (LR) images are cropped to 48×48 patches for the training procedure. They use MSE loss with the Adam optimizer for the network optimization. The initial learning rate is set to 0.001. The total number of epochs is 500 and the batch size is 64. To prevent the explosion of gradients they apply the threshold 10 for the clip gradient norm. They adopt a multi-step learning rate strategy, where the learning rate will decrease to 60% of the current learning rate after each 50 epoch. Data augmentation strategies included horizontal flips, and random rotations of 90, 180, and 270 degrees.

4.26. MagicSR

Method. MagicSR team adopts the structure of the Swin Transformer with window self-attention (WSA) for single image super-resolution (SR). However, the plain WSA ignores the broad regions when reconstructing high-resolution images due to a limited receptive field, and requires intensive computations due to the nature of its structure. To overcome these problems, they propose context-aware neighbor local windows, inspired by N-Gram [14], to produce neighboring embeddings and interact with each other by sliding WSA to produce the context-aware features before window partitioning. As illustrated in Fig. 39, the proposed MagicSR-Light consists of five components: a shallow module (a 3 × 3 convolution), three hierarchical encoder stages (with patch-merging) that contain CST blocks (context-aware Swin transformer modules), PCD bottleneck (pixel-shuffle, concatenation, depth-wise convolution, point-wise projection), a small decoder stage with CSTs, and an image reconstruction module. They employ CSTs by using the context-aware neighbor local windows and the scaled-cosine attention proposed by Swin V2 [59]. PCD bottleneck, which takes multi-scale outputs of the encoder, is a variant of bottleneck from U-Net. They adopt a decoder module composed of a CST block and a reconstruction layer to produce the final RGB output. They train their model in an adversarial way [61] to improve the robustness. Given a low-resolution (LR) image I, a shallow module (comprising a 3 × 3 convolution) is employed to extract pertinent features. These features subsequently traverse three encoder stages, each comprising n_i CST Blocks, and undergo a 2 × 2 patch-merging process, with the exception of the final stage. Notably, the patch-merging mechanism mirrors that of the Swin Transformer [59], albeit with a dimensional re-

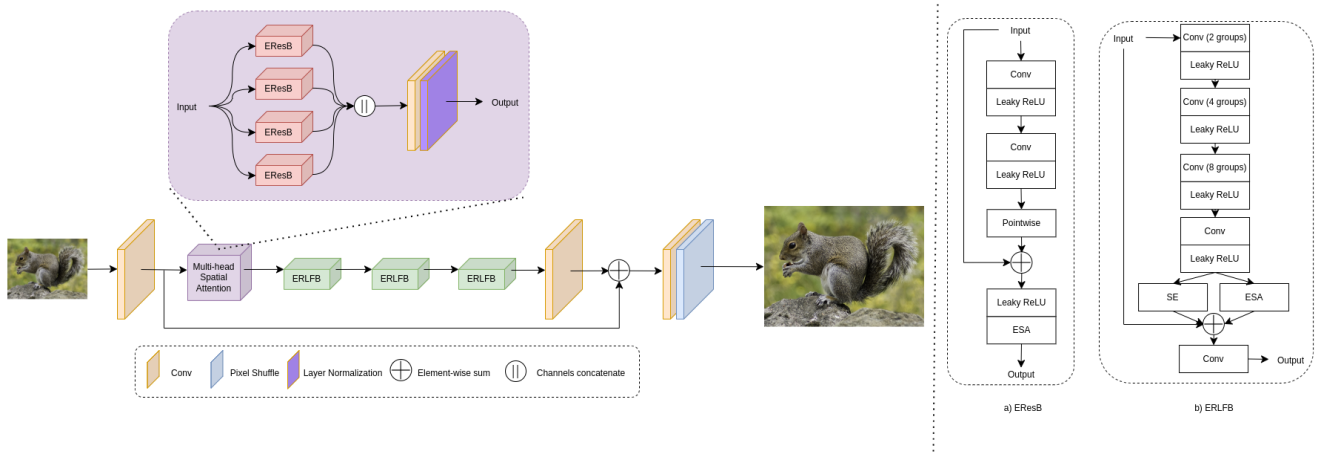


Figure 38. *Team Minimalist:* The overall EMaxGMan architecture is shown on the left side, with a) EResB and b) ERLFB on the right side.

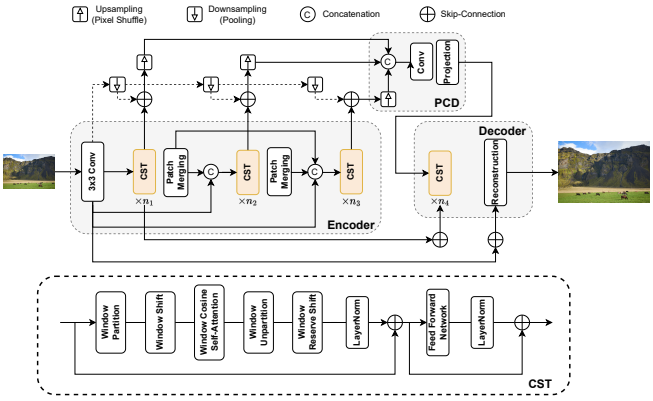


Figure 39. *Team MagicSR:* Overall architecture of MagicSR-Light and CST. In the CST module, they adopt an asymmetric U-Net decoder architecture consisting of pixel-shuffle, concatenation, depth-wise convolution, and point-wise (linear) projection. This variant efficiently integrates multi-scale outputs from the encoder part, encompassing the shallow module.

duction from 4D to D instead of 2D. This reduction in network dimensionality via patch merging results in a notable decrease in attention computation requirements for the CST. In their model, they set $n_1, n_2, n_3 = 6, 4, 4$.

As illustrated in the bottom of Fig. 39, their Context-aware Swin Transformer (CST) adopts scaled-cosine attention and post-normalization proposed in SwinV2 [59]. The window size of the WSA module is set to 8 by default. In window partitioning, they implement the context-aware neighbor local windows by following [14]. This algorithm is also identically applied to other Swin Transformer models (SwinIR-light [49]), focusing only on better performances. The window shifts are operated in the even-numbered blocks, the same as in the Swin Transformer. The decoder module comprises n_4 CST blocks along with a reconstruction layer, where they set n_4 to 6. The reconstruction module is structured with a convolution layer, followed by a pixel-shuffler, and another convolution layer for output conversion. The input to this module incorporates a global skip connection originating from the shallow module of the encoder.

4.27. DIRN

Method. Despite the rapid development of neural network architecture, convolution remains the mainstay of deep neural networks. In recent years, depth-wise separable convolution [33] has been proposed to speed up deep models. Duo Li.[40] introduced the involution operation to match visual patterns concerning the location and reduce inter-channel redundancy. The reverse function of involution enhances efficiency and reduces parameters. A visual and efficient involution kernel belonging to a specific location can be generated by only considering the feature vector of the corresponding location, as shown in Figure. 40(c). By sharing the involution kernel along the channel dimension, the redundancy of the kernels is reduced. Considering the focus of involution on visual performance and self-attention, in the remaining blocks, they use involution along with depth convolution to extract deep features. They propose that the structure of network blocks varies according to the depth. Therefore depth convolution is used more in the initial blocks. As shown in Figure.40(a), the Depthwise convolution and Involution Residual Block (DIRB) block uses three layers of Invo/DConv 3×3 + LeakyReLU to extract deep features. Feature maps are then connected along the channel dimension. Inspired by IMDN, they modified the Contrast-Aware Channel Attention Block (CCA) [34] and used it to evaluate the contrast degree of feature maps and

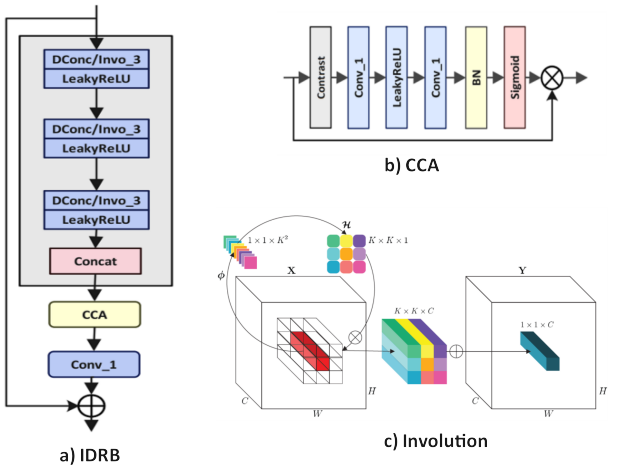


Figure 40. *Team DIRN:* (a) Depthwise convolution and Involution Residual Block(DIRB). (b) Contrast-Aware Channel Attention Block (CCA). (c) Involution[40].

improve performance. They also used a 1×1 convolution to reduce the number of channels and connections left at the end of the block.

Training Details. They are using DIV2K[71] and LSDIR[46] images as the dataset. HR image patches of size 96×96 are randomly cropped from HR images, and the small batch size is set to 16. They adopt the Adam optimizer and for the loss function, they use the L1 loss to measure the difference between the SR images and the ground truth. In DIRN, they set the number of DIRBs to 6. To investigate the Combination of Involution and Depthwise convolution at different depths, four different models were considered. In each of the models, as the network deepens, involution is used more than Depthwise convolution in the DIRB block, and in none of the models, the structure of the blocks is the same. Experiments with 100 epochs and evaluation results with validation dataset DIV2K_LSDIR_valid are presented. According to the results, it is appropriate to use Depthwise convolution in the initial blocks where feature maps have fewer changes and are similar to pixel values. As the network gets deeper, feature maps have undergone many changes along the network, and by ensuring correlation using Depthwise convolution, using involution as a visual operator can yield better results. According to the results of the fourth model, it performed better than other models and they chose it as the main model. The DIRN model is lightweight with 97K parameters and 6.48G for flops.

4.28. ACVLAB

Method. The method of ACVLAB is based on [39], which maintains performance with a smaller number of parameters by incorporating high-frequency prior information. It includes three key components: high-frequency enhanced residual block (HFERB), shifted rectangular window attention block (SRWAB) for capturing high-frequency information to capture global information, and hybrid fusion Blocks (HFB) are designed to enhance the global representation. They reduce the network depth and adapt the small convolutional kernels in the deep feature extraction stage to further reduce the number of parameters.

Training Details. They use LSDIR [46] and DIV2K [71] dataset for training. Their training stage can be divided into two stages. Throughout the entire training process, they adapt the Adam optimizer with β_1 = 0.9, and β_2 = 0.999 and train for 500000 iterations in each stage. The learning rate is set to 2e-4, the multi-step learning scheduler is also used. The learning rate is halved at the [250000, 400000, 450000, 475000] iterations respectively. Weight decay is not applied. For the training stage, HR patches of size 256 $\times$ 256 are randomly cropped from HR images. The random horizontal flips and the random rotation are used for augmentation. In the first stage, they use the L1-loss for the model optimization with a batch size of 16. In the second stage, they use the MSE-loss for enhancement. The model was implemented using Pytorch 1.13.1 and trained on a single NVIDIA-GeForce-RTX-3090.

4.29. KLETech-CEVI_Lowlight_Hypnotise

Method. In this work, the team proposes an architecture named Efficient SRGAN for image super-resolution, aimed at achieving improved performance in terms of both reconstruction accuracy and computational efficiency. Image super-resolution techniques like [65], [67], [11], [2] aim at improving the resolution of images. Real-ESRGAN, an extension of the Enhanced Super-Resolution Generative Adversarial Network (ESRGAN), presents a powerful architecture for image super-resolution tasks. The architecture incorporates a high-order degradation model, to simulate complex real-world degradations accurately. Additionally, Real-ESRGAN addresses artifacts such as ringing and overshooting during synthesis, leading to improved visual quality in the output images. To further enhance performance, Real-ESRGAN utilizes a U-Net discriminator with spectral normalization, increasing discriminator capability and stabilizing training dynamics. Real-ESRGAN is trained on synthetic data, for improved generalization on real-world image restoration challenges. They extend their work by training Real-ESRGAN [77] on custom weighted combinational loss function as shown in Equation 23. However, Real-ESRGAN suffers from substantial texture information loss in the reconstructed image. To overcome this, they propose to use VGG-19 Perceptual loss inspired from [18], combined with $L1$ loss.

$$\mathcal{L}_{SR} = \alpha * \mathcal{L}_{VGG} + \beta * \mathcal{L}_1 \tag{23}$$

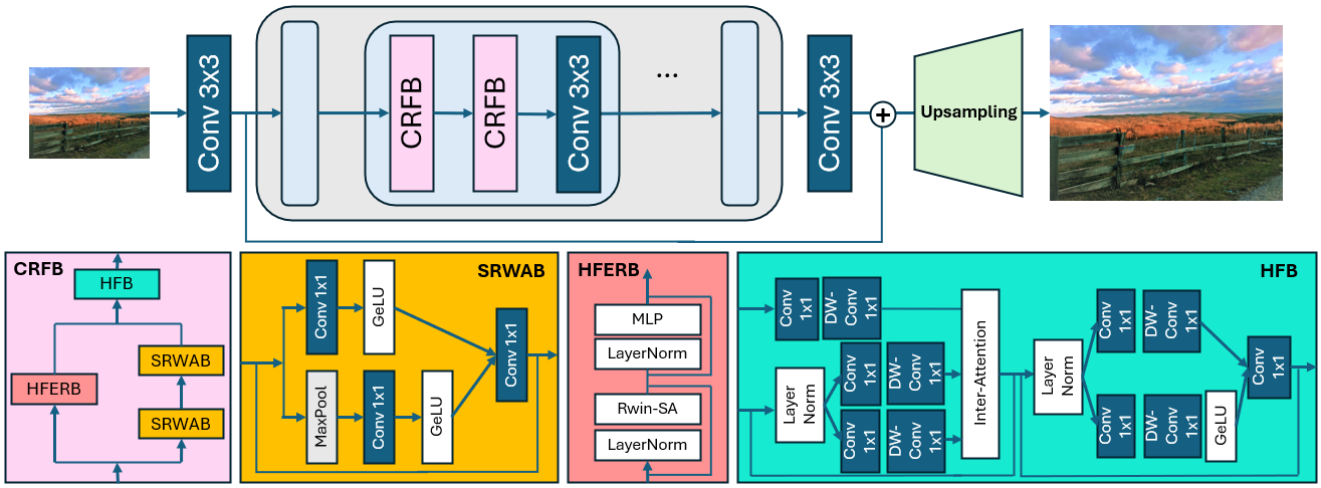


Figure 41. *Team ACVLAB:* The overall architecture.

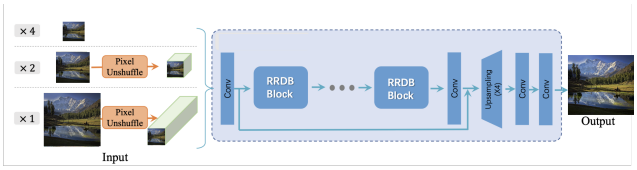


Figure 42. *Team KLETech-CEVI_Lowlight_Hypnotise:* Overview architecture diagram of Efficient SRGAN, illustrating the network structure and flow of information during the super-resolution process. (Figure reproduced from [77])

where, $\mathcal{L}_{VGG}$ is VGG-19 perceptual loss [38], and is given as,

$$\mathcal{L}_{VGG} = \frac{1}{WH} \sum i = 1^{W} \sum j = 1^{H} (\phi(\hat{y})i, j - \phi(x)i, j) \tag{24}$$

where, $\phi(.)$ is the activation of j^{th} layer of network ϕ when processing on image x. W and H are width and height of image. α and β are weights to the losses, and are set to 0.7 and 0.5 heuristically.

Training Details: They train the proposed methodology using the dataset provided by NTIRE 2024 Efficient Super-Resolution Challenge. They train the model using Python and PyTorch frameworks, on a patch resolution of 339*510, with a batch size of 8. They use Adam optimizer with $\beta 1$ set to 0.9 and $\beta 2$ set to 0.999. They train the model for 1000 epochs at a learning rate of 0.0002. During testing, they use full-resolution images (339*510), on a single RTX 3090 GPU. The average testing time for a single image on the full resolution is 0.9s on RTX 3090 GPU.

Acknowledgments

This work was partially supported by the Humboldt Foundation. We thank the NTIRE 2024 sponsors: Meta Reality Labs, OPPO, KuaiShou, Huawei and University of Würzburg (Computer Vision Lab).

A. Teams and affiliations

NTIRE 2024 team

Title: NTIRE 2024 Efficient Super-Resolution Challenge

Members:
Bin Ren[1,2] (bin.ren@unitn.it),
Yawei Li[3] (yawei.li@vision.ee.ethz.ch),
Nancy Mehta[4] (nancy.mehta@uni-wuerzburg.de),
Radu Timofte[3,4] (Radu.Timofte@uni-wuerzburg.de)

Affiliations:
[1] University of Pisa, Italy
[2] University of Trento, Italy
[3] Computer Vision Lab, ETH Zürich, Switzerland
[4] University of Würzburg, Germany

XiaomiMM

Title: Swift Parameter-free Attention Network for Efficient Image Super-Resolution

Members:
Hongyuan Yu[1] (yuhyuan1995@gmail.com),
Cheng Wan[2], Yuxin Hong[3], Bingnan Han[1], Zhuoyuan Wu[1], Yajun Zou[1], Yuqing Liu[1], Jizhe Li[1], Keji He[4], Chao Fan[5], Heng Zhang[1], Xiaolin Zhang[1], Xuanwu Yin[1], Kunlong Zuo[1]

Affiliations:
[1] Multimedia Department, Xiaomi Inc.

[2] Georgia Institute of Technology
[3] Lanzhou University
[4] Institute of Automation, Chinese Academy of Sciences
[5] Beijing University of Technology

Cao Group

Title: Double Reparameterization Network for Efficient Image Super-Resolution
Members:
Bohao Liao[1] (liaobh@mail.ustc.edu.cn),
Peizhe Xia[1], Long Peng[1], Zhibo Du[1], Xin Di[1], Wangkai Li[1], Yang Wang[1], Wei Zhai[1], Renjing Pei[2], Jiaming Guo[2], Songcen Xu[2], Yang Cao[1], Zhengjun Zha[1]
Affiliations:
[1] University of Science and Technology of China
[2] Huawei Noah's Ark Lab

BSR

Title: Lightening Partial Feature Distillation Network for Efficient Super-Resolution
Members:
Yan Wang[1] (wyrmy@foxmail.com),
Yi Liu[2], Qing Wang[2], Gang Zhang[2], Liou Zhang[2], Shijie Zhao[2]
Affiliations:
[1] Nankai University
[2] ByteDance Inc.

PiXupt

Title: Hierarchical Attention Residual Network for Efficient Image Super-Resolution
Members:
Yanyu Mao[1] (bolt35982@gmail.com),
Ruilong Guo[1], Nihao Zhang[1], Qian Wang[1,2]
Affiliations:
[1] Xian University of Posts and Telecommunications, Xi'an, China
[2] National Engineering Laboratory for Cyber Event Warning and Control Technologies

XJU_100th Ann

Title: Attention Guidance Distillation Network for Efficient Image Super-Resolution
Members:
Shuli Cheng[1] (cslxju@xju.edu.cn),
Hongyuan Wang[1], Ziyan Wei[1], Qingting Tang[1], Liejun Wang[1], Yongming Li[1]
Affiliations:
[1] School of Computer Science and Technology, Xinjiang University, Ürümqi, China

VPEG_C

Title: SMFAN: A Lightweight Self-Modulation Feature Aggregation Network for Efficient Image Super-Resolution
Members:
Mingjun Zheng[1] (jun_zmj@qq.com),
Long Sun[1], Jinshan Pan[1], Jiangxin Dong[1], Jinhui Tang[1]
Affiliations:
[1] Nanjing University of Science and Technology

ZHEstar

Title: Large Kernel Frequency-enhanced Network for Efficient Single Image Super-Resolution
Members:
Jiadi Chen (zjnucjd@163.com),
Huanhuan Long, Chunjiang Duanmu
Affiliations:
Zhejiang Normal University, Jinhua, China

VPEG_O

Title: SAFMN++: Improved Feature Modulation Network for Efficient Image Super-Resolution
Members:
Long Sun[1] (cs.longsun@gmail.com),
Jinshan Pan[1], Jiangxin Dong[1], Jinhui Tang[1]
Affiliations:
[1] Nanjing University of Science and Technology

CMVG

Title: Multi-supervisory Residual Knowledge Distillation Network for Efficient Super-resolution
Members:
Xin Liu[1] (liuxin9976@163.com),
Min Yan[1], Qian Wang[1]
Affiliations:
[1] China Mobile Research Institute

LeESR

Title: Separable-Mixable Residual Network for Efficient image Super-Resolution
Members:
Menghan Zhou[1] (zhoumh3@lenovo.com),
Yiqiang Yan[1]
Affiliations:
[1] Lenovo Research

AdvancedSR

Title: Progressive Kernel Pruning for Efficient Super-Resolution
Members:
Yixuan Liu[1] (yixuanl@amd.com),
Wensong Chan[1], Dehua Tang[1], Dong Zhou[1], Li Wang[1], Lu Tian[1], Barsoum Emad[1]
Affiliations:
[1] Advanced Micro Devices, Inc., Beijing, China

ECNU_MViC

Title: Intermittent Feature Aggregation with Distillation for Efficient Super-Resolution
Members:
Bohan Jia[1] (10205501423@stu.ecnu.edu.cn),
Jincheng Liao[1], Junbo Qiao[1,2], Yunshuai Zhou[1], Yun Zhang[2,3], Wei Li[2], Shaohui Lin[1]
Affiliations:
[1] East China Normal University
[2] Huawei Noah's Ark Lab
[3] The Hong Kong University of Science and Technology

HiSR

Title: SlimRLFN: Making RLFN Smaller and Faster Again
Members:
Shenglong Zhou[1] (slzhou96@mail.ustc.edu.cn),
Binbin Chen[2]
Affiliations:
[1] University of Science and Technology of China
[2] Huazhong University of Science and Technology

MViC_SR

Title: LSANet: Efficient Image Super-Resolution with Lightweight Spaced Attention Mechanism
Members:
Jincheng Liao[1] (71265901062@stu.ecnu.edu.cn),
Bohan Jia[1], Junbo Qiao[1,2], Yunshuai Zhou[1], Yun Zhang[2,3], Wei Li[2], Shaohui Lin[1]
Affiliations:
[1] East China Normal University
[2] Huawei Noah's Ark Lab
[3] The Hong Kong University of Science and Technology

LVTeam

Title: Explore RLFN's Extreme for Efficient Super Resolution
Members:
Suiyi Zhao[1] (meranderzhao@gmail.com),
Zhao Zhang[1], Bo Wang[1], Yan Luo[1], Yanyan Wei[1]
Affiliations:
[1] Hefei University of Technology

Fresh

Title: Depth Residual Local Feature Network for Efficient Super-Resolution
Members:
Feng Li[1] (fengli@hfut.edu.cn),
Mingshen Wang[1], Yawei Li[1], Jinhan Guan[1], Dehua Hu[1]
Affiliations:
[1] Hefei University of Technology

Lanzhi

Title: Residual Local Feature Block with Batch Normalization
Members:
Jiawei Yu[1] (yujiawei@nudt.edu.cn),
Qisheng Xu[1], Tao Sun[1], Long Lan[1], Kele Xu[1]
Affiliations:
[1] National University of Defense Technology

Supersr

Title: Residual Local Feed Forward for Effcient Image Super-resolution
Members:
Xin Lin[1] (linxin@stu.scu.edu.cn),
Jingtong Yue[1], Lehan Yang[2], Shiyi Du[3], Lu Qi[4], Chao Ren[1]
Affiliations:
[1] Sichuan University
[2] The University of Sydney
[3] Carnegie Mellon University
[4] The University of California, Merced

MeowMeowMeow

Title: Reparameterized Convolutional Network for Efficient Image Super-Resolution
Members:
Zeyu Han[1] (hanzeyu2001@outlook.com),
Yuhan Wang[1], Chaolin Chen[1]
Affiliations:
[1] Sichuan University

Just Try

Title: Enhanced Reparameterize Residual Network for Efficient Image Super-Resolution
Members:

Haobo Li (1114940412@qq.com),

VPEG_E

Title: Enhanced Gated Feature Modulation for Efficient Super-Resolution

Members:
Zhongbao Yang[1] (yangzhongbao40@gmail.com),
Long Sun[1], Lianhong Song[1], Jinshan Pan[1], Jiangxin Dong[1], Jinhui Tang[1]
Affiliations:
[1] Nanjing University of Science and Technology

BU-ESR

Title: Boosting Residual Feature Distillation Networks through Knowledge Dilation

Members:
Xingzhuo Yan[1] (ayx1sgh@bosch.com),
Minghan Fu[2]
Affiliations:
[1] Bosch Investment Ltd.
[2] University of Saskatchewan

Lasagna

Title: Efficient Enhanced Residual Network for Efficient Image Super-Resolution

Members:
Jingyi Zhang[1] (jingyizhang0806@163.com),
Baiang Li[1], Qi Zhu[2], Xiaogang Xu[3,4], Dan Guo[1], Chunle Guo*[5]
Affiliations:
[1] Hefei University of Technology
[2] University of Science and Technology of China
[3] The Chinese University of Hong Kong
[4] Zhejiang University
[5] Nankai University

BlingBling

Title: DVMSR: Distillated Vision Mamba Super-Resolution

Members:
Xiaoyan Lei[1] (xyan_lei@163.com),
Jie Liu[1], Weilin Jia[1], Weifeng Cao[1], Wenlong Zhang[2]
Affiliations:
[1] Zhengzhou University of Light Industry
[2] The Hong Kong Polytechnic University

Minimalist

Title: Efficient Image Super-Resolution with Lightweight Multi-head Spatial Attention

Members:
Manoj Pandey[1] (pandeymanoj@deltax.ai),
Maksym Chernozhukov[1], Giang Le[1]
Affiliations:
[1] DeltaX

MagicSR

Title Context-aware Transformer for Efficient Image Super-resolution

Members:
Yanhui Guo[1] (guoy143@mcmaster.ca),
Hao Xu[1]
Affiliations:
[1] McMaster University

DIRN

Title: Deep combination of Depthwise and Involution for the residual network of super-resolution image

Members:
Akram Khatami-Rizi [1] (akramkhatami67@gmail.com),
Ahmad Mahmoudi-Aznaveh [2]
Affiliations:
[1] Cyberspace Research Institute of Shahid Beheshti University of Iran

ACVLAB

Title: Solution for NTIRE 2024 Efficient SR Challenge

Members:
Chih-Chung Hsu[1] (cchsu@gs.ncku.edu.tw),
Chia-Ming Lee[1] , Yi-Shiuan Chou[1]
Affiliations:
[1] Institute of Data Science, National Cheng Kung University

KLETech-CEVI_Lowlight_Hypnotise

Title: *Efficient SRGAN Towards Super-Resolution of Images*

Members:
Amogh Joshi[1] (joshiamoghmukund@gmail.com),
Nikhil Akalwadi[1,3], Sampada Malagi[1,3], Palani Yashaswini[1,2], Chaitra Desai[1,3], Ramesh Ashok Tabib[1,2], Ujwala Patil[1,2], Uma Mudenagudi[1,2]
Affiliations:
[1] Center of Excellence in Visual Intelligence (CEVI), KLE Technological University, Hubballi, Karnataka, INDIA

[2] School of Electronics and Communication Engineering, KLE Technological University, Hubballi, Karnataka, INDIA
[3] School of Computer Science and Engineering, KLE Technological University, Hubballi, Karnataka, INDIA

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 3, 9, 11, 12, 18, 21, 23

[2] Hanadi Al-Mekhlafi and Shiguang Liu. Single image super-resolution: a comprehensive review and recent insight. *Frontiers of Computer Science*, 18(1):181702, 2024. 29

[3] Cosmin Ancuti, Codruta O Ancuti, Florin-Alexandru Vasluianu, Radu Timofte, et al. NTIRE 2024 dense and non-homogeneous dehazing challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[4] Nikola Banić, Egor Ershov, Artyom Panshin, Oleg Karasev, Sergey Korchagin, Shepelev Lev, Alexandr Startsev, Daniil Vladimirov, Ekaterina Zaychenkova, Dmitrii R Iarchuk, Maria Efimova, Radu Timofte, Arseniy Terekhin, et al. NTIRE 2024 challenge on night photography rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[5] Kartikeya Bhardwaj, Milos Milosavljevic, Liam O'Neil, Dibakar Gope, Ramon Matas, Alex Chalfin, Naveen Suda, Lingchuan Meng, and Danny Loh. Collapsible linear blocks for super-efficient super resolution. *Proceedings of Machine Learning and Systems*, 4:529–547, 2022. 16

[6] Christopher M Bishop and Nasser M Nasrabadi. *Pattern recognition and machine learning*. Springer, 2006. 2

[7] Nicolas Chahine, Marcos V. Conde, Sira Ferradans, Radu Timofte, et al. Deep portrait quality assessment. a NTIRE 2024 challenge survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[8] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don't walk: Chasing higher flops for faster neural networks. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023. 9, 12

[9] Jiadi Chen, Chunjiang Duanmu, and Huanhuan Long. Large kernel frequency-enhanced network for efficient single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2024. 12

[10] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22367–22377, 2023. 14, 24

[11] Yuantao Chen, Runlong Xia, Kai Yang, and Ke Zou. Mffn: image super-resolution via multi-level features fusion network. *The Visual Computer*, 40(2):489–504, 2024. 29

[12] Zheng Chen, Zongwei WU, Eduard Sebastian Zamfir, Kai Zhang, Yulun Zhang, Radu Timofte, Xiaokang Yang, et al. NTIRE 2024 challenge on image super-resolution (×4): Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[13] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021. 12, 13, 16, 24

[14] Haram Choi, Jeongmin Lee, and Jihoon Yang. N-gram in swin transformers for efficient lightweight image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2071–2081, 2023. 27, 28

[15] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu. Improving image restoration by revisiting global information aggregation. In *Proceedings of European Conference on Computer Vision*, pages 53–71. Springer, 2022. 11

[16] Junyoung Chung, Çaglar Gülçehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *ArXiv*, abs/1412.3555, 2014. 9

[17] Marcos V. Conde, Florin-Alexandru Vasluianu, Radu Timofte, et al. Deep raw image super-resolution. a NTIRE 2024 challenge survey. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[18] Chaitra Desai, Nikhil Akalwadi, Amogh Joshi, Sampada Malagi, Chinmayee Mandi, Ramesh Ashok Tabib, Ujwala Patil, and Uma Mudenagudi. Lightnet: Generative model for enhancement of low-light images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, pages 2231–2240, 2023. 29

[19] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Diverse branch block: Building a convolution as an inception-like unit. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10886–10895, 2021. 24

[20] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021. 6

[21] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16×16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 2

[22] Jie Du, Kai Guan, Yanhong Zhou, Yuanman Li, and Tianfu Wang. Parameter-free similarity-aware attention module for medical image classification and segmentation. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2022. 6

[23] Zongcai Du, Jie Liu, Jie Tang, and Gangshan Wu. Anchor-based plain net for mobile image super-resolution. In *Pro-*

ceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2494–2502, 2021. 8

[24] Zongcai Du, Ding Liu, Jie Liu, Jie Tang, Gangshan Wu, and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 853–862, 2022. 8, 19, 24

[25] Minghan Fu and Fang-Xiang Wu. Qlabgrad: A hyperparameter-free and convergence-guaranteed scheme for deep learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 12072–12081, 2024. 25

[26] Minghan Fu, Yanhua Duan, Zhaoping Cheng, Wenjian Qin, Ying Wang, Dong Liang, and Zhanli Hu. Total-body low-dose ct image denoising using a prior knowledge transfer technique with a contrastive regularization mechanism. *Medical Physics*, 50(5):2971–2984, 2023. 24

[27] Minghan Fu, Meiyun Wang, Yaping Wu, Na Zhang, Yongfeng Yang, Haining Wang, Yun Zhou, Yue Shang, Fang-Xiang Wu, Hairong Zheng, et al. A two-branch neural network for short-axis pet image quality enhancement. *IEEE Journal of Biomedical and Health Informatics*, 2023. 24

[28] Minghan Fu, Na Zhang, Zhenxing Huang, Chao Zhou, Xu Zhang, Jianmin Yuan, Qiang He, Yongfeng Yang, Hairong Zheng, Dong Liang, et al. Oif-net: An optical flow registration-based pet/mr cross-modal interactive fusion network for low-count brain pet image denoising. *IEEE Transactions on Medical Imaging*, 2023. 24

[29] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 2, 6, 26

[30] Zeyu Han, Fangrui Zhu, Qianru Lao, and Huaizu Jiang. Zero-shot referring expression comprehension via structural similarity between images and captions. *arXiv preprint arXiv:2311.17048*, 2023. 23

[31] Zeyu Han, Chao Gao, Jinyang Liu, Sai Qian Zhang, et al. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*, 2024. 23

[32] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 12, 19

[33] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 28

[34] Zheng Hui, Xinbo Gao, Yunchu Yang, and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the ACM International Conference on Multimedia*, pages 2024–2032, 2019. 9, 10, 28

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 8, 16, 18, 21

[36] Fangyuan Kong, Mingxi Li, Songwei Liu, Ding Liu, Jingwen He, Yang Bai, Fangmin Chen, and Lean Fu. Residual local feature network for efficient super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 766–776, 2022. 2, 3, 4, 14, 15, 16, 17, 18, 19, 21, 22, 23, 24, 27

[37] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 2

[38] Christian Ledig, Lucas Theis, Ferenc Husz'ar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, 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*, pages 4681–4690, 2017. 30

[39] Ao Li, Le Zhang, Yun Liu, and Ce Zhu. Feature modulation transformer: Cross-refinement of global representation via high-frequency prior for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12514–12524, 2023. 29

[40] Duo Li, Jie Hu, Changhu Wang, Xiangtai Li, Qi She, Lei Zhu, Tong Zhang, and Qifeng Chen. Involution: Inverting the inherence of convolution for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12321–12330, 2021. 28, 29

[41] Xiang Li, Jiangxin Dong, Jinhui Tang, and Jinshan Pan. Dlgsanet: lightweight dynamic local and global self-attention networks for image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12792–12801, 2023. 10

[42] Xin Li, Kun Yuan, Yajing Pei, Yiting Lu, Ming Sun, Chao Zhou, Zhibo Chen, Radu Timofte, et al. NTIRE 2024 challenge on short-form UGC video quality assessment: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[43] Yawei Li, Shuhang Gu, Kai Zhang, Luc Van Gool, and Radu Timofte. DHP: Differentiable meta pruning via hypernetworks. In *Proceeding of the European Conference on Computer Vision*, pages 608–624. Springer, 2020. 2

[44] Yawei Li, Wen Li, Martin Danelljan, Kai Zhang, Shuhang Gu, Luc Van Gool, and Radu Timofte. The heterogeneity hypothesis: Finding layer-wise differentiated network architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2144–2153, 2021. 2

[45] Yawei Li, Yuchen Fan, Xiaoyu Xiang, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. Efficient and explicit modelling of image hierarchies for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18278–18289, 2023. 2

[46] Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. Lsdir: A large scale dataset for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023. 3, 6, 11, 12, 13, 16, 18, 21, 22, 23, 26, 29

[47] Yawei Li, Yulun Zhang, Luc Van Gool, Radu Timofte, et al. NTIRE 2023 challenge on efficient super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023. 8, 9

[48] Zheyuan Li, Yingqi Liu, Xiangyu Chen, Haoming Cai, Jinjin Gu, Yu Qiao, and Chao Dong. Blueprint separable residual network for efficient image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 833–843, 2022. 9, 12, 15, 17, 19

[49] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 1833–1844, 2021. 2, 28

[50] Jie Liang, Qiaosi Yi, Shuaizheng Liu, Lingchen Sun, Rongyuan Wu, Xindong Zhang, Hui Zeng, Radu Timofte, Lei Zhang, et al. NTIRE 2024 restore any image model (RAIM) in the wild challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[51] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1132–1140, 2017. 11, 12, 25

[52] Jie Liu, Jie Tang, and Gangshan Wu. Residual feature distillation network for lightweight image super-resolution. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 41–55. Springer, 2020. 9, 15, 25

[53] Jie Liu, Wenjie Zhang, Yuting Tang, Jie Tang, and Gangshan Wu. Residual feature aggregation network for image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2359–2368, 2020. 10

[54] Jie Liu, Jie Tang, and Gangshan Wu. Adadm: Enabling normalization for image super-resolution. *arXiv preprint arXiv:2111.13905*, 2021. 22

[55] Ji Liu, Dehua Tang, Yuanxian Huang, Li Zhang, Xiaocheng Zeng, Dong Li, Mingjie Lu, Jinzhang Peng, Yu Wang, Fan Jiang, et al. Updp: A unified progressive depth pruner for cnn and vision transformer. *AAAI*, 2024. 16

[56] Xiaohong Liu, Xiongkuo Min, Guangtao Zhai, Chunyi Li, Tengchuan Kou, Wei Sun, Haoning Wu, Yixuan Gao, Yuqin Cao, Zicheng Zhang, Xiele Wu, Radu Timofte, et al. NTIRE 2024 quality assessment of AI-generated content challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[57] Xiaoning Liu, Zongwei Wu, Ao Li, Florin-Alexandru Vasluianu, Yulun Zhang, Shuhang Gu, Le Zhang, Ce Zhu, Radu Timofte, et al. NTIRE 2024 challenge on low light image enhancement: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[58] Zechun Liu, Haoyuan Mu, Xiangyu Zhang, Zichao Guo, Xin Yang, Tim Kwang-Ting Cheng, and Jian Sun. MetaPruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019. 2

[59] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022. 27, 28

[60] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. In *ICLR*, 2017. 12, 13, 24

[61] Fangzhou Luo, Xiaolin Wu, and Yanhui Guo. And: Adversarial neural degradation for learning blind image super-resolution. *Advances in Neural Information Processing Systems*, 36, 2024. 27

[62] Yanyu Mao, Nihao Zhang, Qian Wang, Bendu Bai, Wanying Bai, Haonan Fang, Peng Liu, Mingyue Li, and Shengbo Yan. Multi-level dispersion residual network for efficient image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1660–1669, 2023. 9, 11

[63] Nancy Mehta, Akshay Dudhane, Subrahmanyam Murala, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Adaptive feature consolidation network for burst super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1279–1286, 2022. 2

[64] Nancy Mehta, Akshay Dudhane, Subrahmanyam Murala, Syed Waqas Zamir, Salman Khan, and Fahad Shahbaz Khan. Gated multi-resolution transfer network for burst restoration and enhancement. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22201–22210. IEEE, 2023. 2

[65] Uma Mudenagudi, Subhashis Banerjee, and Prem Kumar Kalra. Space-time super-resolution using graph-cut optimization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):995–1008, 2011. 29

[66] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 16

[67] Ujwala Patil, Ramesh Ashok Tabib, Channabasappa M. Konin, and Uma Mudenagudi. Evidence-based framework for multi-image super-resolution. In *Recent Findings in Intelligent Computing Techniques*, pages 413–423, Singapore, 2018. Springer Singapore. 29

[68] Bin Ren, Yahui Liu, Yue Song, Wei Bi, Rita Cucchiara, Nicu Sebe, and Wei Wang. Masked jigsaw puzzle: A versatile position embedding for vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20382–20391, 2023. 2

[69] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 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, pages 1874–1883, 2016. 17, 19, 24

[70] Long Sun, Jiangxin Dong, Jinhui Tang, and Jinshan Pan. Spatially-adaptive feature modulation for efficient image super-resolution. In *ICCV*, 2023. 13, 18, 20, 24

[71] Radu Timofte, Eirikur Agustsson, Luc Van Gool, Ming-Hsuan Yang, Lei Zhang, et al. NTIRE 2017 challenge on single image super-resolution: Methods and results. In *CVPR Workshops*, 2017. 9, 12, 16, 26, 29

[72] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022. 2

[73] Florin-Alexandru Vasluianu, Tim Seizinger, Zhuyun Zhou, Zongwei WU, Cailian Chen, Radu Timofte, et al. NTIRE 2024 image shadow removal challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[74] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017. 27

[75] Cheng Wan, Hongyuan Yu, Zhiqi Li, Yihang Chen, Yajun Zou, Yuqing Liu, Xuanwu Yin, and Kunlong Zuo. Swift parameter-free attention network for efficient super-resolution. *arXiv preprint arXiv:2311.12770*, 2023. 6, 7

[76] Longguang Wang, Yulan Guo, Juncheng Li, Hongda Liu, Yang Zhao, Yingqian Wang, Zhi Jin, Shuhang Gu, Radu Timofte, et al. NTIRE 2024 challenge on stereo image super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[77] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021. 29, 30

[78] Yan Wang. Edge-enhanced feature distillation network for efficient super-resolution. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, pages 777–785, 2022. 9

[79] Yucong Wang and Minjie Cai. A single residual network with esa modules and distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1970–1980, 2023. 8

[80] Yingqian Wang, Zhengyu Liang, Qianyu Chen, Longguang Wang, Jungang Yang, Radu Timofte, Yulan Guo, et al. NTIRE 2024 challenge on light field image super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[81] Xingyu Xie, Pan Zhou, Huan Li, Zhouchen Lin, and Shuicheng Yan. Adan: Adaptive nesterov momentum algorithm for faster optimizing deep models. *arXiv preprint arXiv:2208.06677*, 2022. 12

[82] Lingxiao Yang, Ru-Yuan Zhang, Lida Li, and Xiaohua Xie. Simam: A simple, parameter-free attention module for convolutional neural networks. In *International conference on machine learning*, pages 11863–11874. PMLR, 2021. 6

[83] Ren Yang, Radu Timofte, et al. NTIRE 2024 challenge on blind enhancement of compressed image: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[84] Lei Yu, Xinpeng Li, Youwei Li, Ting Jiang, Qi Wu, Haoqiang Fan, and Shuaicheng Liu. Dipnet: Efficiency distillation and iterative pruning for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1692–1701, 2023. 8

[85] Pierluigi Zama Ramirez, Fabio Tosi, Luigi Di Stefano, Radu Timofte, Alex Costanzino, Matteo Poggi, et al. NTIRE 2024 challenge on HR depth from images of specular and transparent surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[86] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 2

[87] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 22, 24

[88] Xindong Zhang, Hui Zeng, and Lei Zhang. Edge-oriented convolution block for real-time super resolution on mobile devices. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 4034–4043, 2021. 14, 19, 23

[89] Zhilu Zhang, Shuohao Zhang, Renlong Wu, Wangmeng Zuo, Radu Timofte, et al. NTIRE 2024 challenge on bracketing image restoration and enhancement: Datasets, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2024. 2

[90] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 26

[91] Barret Zoph and Quoc V Le. Neural architecture search with reinforcement learning. In *Proceedings of International Conference on Learning Representations*, 2017. 2

[92] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8697–8710, 2018. 2