RepNet-VSR: Reparameterizable Architecture for High-Fidelity Video Super-Resolution

Biao Wu Diankai Zhang Shaoli Liu Si Gao Chengjian Zheng Ning Wang State Key Laboratory of Mobile Network and Mobile Multimedia Technology, ZTE, China

{wu.biao,zhang.diankai,liu.shaoli,gao.si,zheng.chengjian,wangning}@zte.com.cn

Abstract

As a fundamental challenge in visual computing, video super-resolution (VSR) focuses on reconstructing highdefinition video sequences from their degraded lowresolution counterparts. While deep convolutional neural networks have demonstrated state-of-the-art performance in spatial-temporal super-resolution tasks, their computationally intensive nature poses significant deployment challenges for resource-constrained edge devices, particularly in real-time mobile video processing scenarios where power efficiency and latency constraints coexist. In this work, we propose a Reparameterizable Architecture for High Fidelity Video Super Resolution method, named RepNet-VSR, for real-time 4x video super-resolution. On the REDS validation set, the proposed model achieves 27.79 dB PSNR when processing $180p \rightarrow 720p$ frames in 103 ms per 10 frames on a MediaTek Dimensity NPU. The competition results demonstrate an excellent balance between restoration quality and deployment efficiency. The proposed method scores higher than the previous champion algorithm of MAI video super-resolution challenge.

1. Introduction

Over recent years, convolutional neural networks (CNNs) have demonstrated unprecedented performance across diverse computer vision domains. Despite their empirical success, practical deployment of these models remains constrained by their substantial parameter counts and floating-point operations (FLOPs), typically requiring execution on cloud-based GPU servers. This limitation conflicts with growing industrial demands for on-device AI capabilities in mobile ecosystems, thereby driving cross-disciplinary research into efficient CNN deployment strategies for edge devices. When deploying AI-based solutions on mobile devices, it is important to pay attention to the characteristics of mobile NPUs and DSPs in order to design efficient models. [1, 2] provide an extensive overview of

smartphone AI acceleration hardware and its performance. Based on the results reported in these papers, the latest mobile NPUs are close to the results of mid-range desktop GPUs published not long ago. Thanks to the development of edge chip computing power and edge AI accelerators, edge AI algorithms have flourished.

Video super-resolution (VSR) has emerged as a pivotal technology for enhancing visual content across industries, enabling the reconstruction of high-resolution (HR) video sequences from low-resolution (LR) inputs. Its applications span diverse domains: streaming platforms leverage VSR to upscale legacy content while minimizing bandwidth consumption; surveillance systems utilize it to clarify critical details in low-light footage for forensic analysis; and mobile devices employ real-time VSR to refine zoomed-in videos or stabilize recordings without specialized hardware. In medical imaging, VSR enhances endoscopic video clarity for precise diagnostics, while autonomous vehicles rely on it to interpret distant objects in adverse weather conditions. Despite these transformative use cases, deploying VSR on resource-constrained edge devices, such as smartphones, drones, or IoT cameras, introduces significant technical hurdles.

In order to promote the development of video superresolution on mobile terminals, MAI2025 has a real-time video super-resolution challenge track on an actual mobile accelerator. The challenge seeks to establish optimal equilibrium between reconstruction fidelity (quantified by PSNR) and operational efficiency (measured through endto-end latency), requiring solutions to be able to balance accuracy-efficiency.

In this work, we propose a Reparameterizable Architecture for High Fidelity Video Super Resolution method, named RepNet-VSR, for real-time 4x video super-resolution. Demonstrating state-of-the-art efficiencyaccuracy co-optimization, our reparameterized architecture achieves 27.79 dB PSNR on the REDS validation dataset for 4× video super-resolution (180p \rightarrow 720p) in 103 ms/f on a MediaTek Dimensity NPU. In summary, our main contributions are as follows:

- We use NAS search to search for the best parameter configuration between PSNR and FLOPs.
- We employ 1×1 convolutional layers for dimensionality reduction, enhancing computational efficiency while preserving feature representation capabilities.
- Multi-level feature fusion via channel concatenation enhances representational capacity by combining lowlevel textures and high-level semantics.
- We enhance the representation capabilities of network through structural reparameterization, using more complex networks during training and merging them into a common convolution during inference to optimize representation capabilities and computational efficiency.

2. Related Work

In recent years, numerous efficient methods have been developed for image super-resolution tasks [3-6], demonstrating robust performance even on hardware with constrained computational capabilities. Following the pioneering work of SRCNN [7], which introduced convolutional neural networks (ConvNets) to super-resolution, FS-RCNN [8] achieved substantial acceleration in single image super-resolution (SISR) networks by utilizing the original low-resolution input and reducing convolution kernel sizes. DRCN [9] and DRRN [10] attempted to minimize model parameters through recurrent architectures but incurred heavy computational overhead. To address this limitation, IDN [11] and IMDN [12] implemented streamlined information fusion strategies to lower both parameter counts and computational demands, whereas RFDN [13] introduced a residual feature distillation mechanism. The recent surge in transformer-based models has also led to the integration of self-attention (SA) [14] into lightweight SR frameworks. For instance, SwinIR [15] adopted windowbased SA, while ESRT [16] devised an efficient transformer variant termed Efficient Multi-Head Attention (EMHA) for SISR. Despite their effectiveness in modeling long-range dependencies, these transformer-based approaches remain computationally intensive.

Video super-resolution (VSR) poses significant challenges due to the necessity of extracting and integrating complementary information from multiple misaligned video frames to achieve high-quality video restoration. A widely used strategy is the sliding-window framework [17–19], which restores each video frame by leveraging neighboring frames within a limited temporal window. In contrast, recurrent frameworks aim to harness long-term dependencies through iterative propagation of latent features across sequential frames. Broadly speaking, these techniques [20–22] achieve superior model compactness compared to sliding-window approaches. However, fundamental challenges persist in preserving long-range temporal dependencies and ensuring robust feature alignment across frames within recurrent architectures. BasicVSR [23] employs bidirectional propagation to holistically integrate temporal information across the full input sequence for enhanced reconstruction fidelity. Notably, it utilizes optical flow estimation to achieve precise spatiotemporal feature alignment through adaptive warping operations. In RRN [24], the model processes sequential frames through hidden state inputs, employing identity mapping in the state propagation to effectively retain texture information throughout the network architecture. Although the above methods achieve high reconstruction quality, they are inefficient on resource-constrained mobile devices.

Neural Architecture Search (NAS) [25–27] automates the design of deep learning architectures for image super-resolution, replacing manual trial-and-error approaches. Recent works demonstrate that NAS can discover lightweight, efficient, or high-performance SR networks tailored to specific constraints (e.g, computational cost, parameters, or PSNR/SSIM metrics). Recent advances in super-resolution have exploited structural reparameterization [28–31] to employ more complex networks during training and merge them into a single vanilla convolution at inference time.

3. Approach

3.1. Real-Time Video Super-Resolution Challenge Constraints

The input tensor of model must process 10 subsequent video frames with a shape of $[1 \times 180 \times 320 \times 30]$, where:

- The first dimension (1) represents the batch size.
- The second and third dimensions (180×320) correspond to the height and width of the input frames from the REDS dataset.
- The fourth dimension (30) combines the 3 RGB color channels and 10 frames (3 × 10 = 30 channels).

The output tensor should have a shape of $[1\times720\times1280\times30]$, maintaining the same batch size and channel count while upscaling the spatial resolution to 720×1280 .

The final score for this challenge, as defined in Formula (1), is determined by two key metrics:

- Reconstruction quality of the output results.
- Model runtime performance on the target MediaTek NPU.

$$Score(PSNR, runtime) = \frac{2^{2 \cdot (PSNR-27)}}{runtime}$$
(1)

The scoring formula demonstrates that doubling the computational efficiency on the target platform yields a quality improvement equivalent to a 0.5dB gain in PSNR. This necessitates prioritizing both image quality restoration and computational performance optimization in network design. To address this dual challenge, we implemented a systematic experimental approach. Our methodology begins with a comprehensive evaluation of state-of-the-art super-resolution techniques, followed by a detailed exposition of our novel algorithm's architectural advantages and performance enhancements.

3.2. Analysis

In previous iterations of the MAI Video Super-Resolution Challenge, participants demonstrated innovative approaches by developing computationally efficient algorithms that achieved an optimal equilibrium between visual reconstruction fidelity and runtime performance. The proposed EVSRNet [32] framework employs neural architecture search (NAS) technology to automatically optimize and identify optimal training parameters within its novel network architecture. Team Diggers [33]introduced a bidirectional recurrent neural network (BRNN) architecture for video super-resolution, which leverages temporal information by incorporating feature maps from both preceding and subsequent frames to enhance the resolution reconstruction of the current target frame. The MVideoSR team [34] developed an ultra-compact neural network architecture featuring merely four convolutional layers and a 6-channel configuration for efficient feature extraction. This lightweight model was optimized through a multistage training methodology, achieving unprecedented computational efficiency and the lowest recorded power consumption on designated hardware platforms. RCBSR [34] adopts a re-parameterized ECB architecture that employs multi-branch network structures during the training phase, which are subsequently consolidated into a single 3×3 convolutional layer for inference. This design strategy effectively optimizes the balance between achieving higher PSNR metrics and maintaining computational efficiency in practical deployment. While these algorithms demonstrate notable performance advantages, they present specific operational limitations. The bidirectional recursive architecture achieves superior reconstruction quality, yet exhibits significant computational inefficiency during execution. Another prevalent issue arises in implementations employing channel compression through 3×3 convolution prior to depth-tospace operations. Particularly in 4× upscaling frameworks, this convolutional layer - designed to reduce feature channels from 48 to 3 - creates a critical processing bottleneck due to its position in the computational pipeline. The architectural requirement for high-dimensional feature transformation prior to spatial reorganization substantially impacts real-time processing capabilities.

3.3. Network Architecture

Motivated by these findings, we propose RepNet-VSR - a re-parameterizable architecture. This approach specifically addresses the identified computational bottlenecks while maintaining reconstruction fidelity through multilevel feature fusion. As shown in Figure 1, we use channel concatenation to integrate multi-level features of deep and shallow network layers to enhance the feature representation capability of the model. To optimize computational efficiency, we incorporate 1×1 convolutional layers for dimensionality reduction throughout the fusion process. Specifically, prior to the depth-to-space operation, we substitute conventional 3×3 convolutions with 1×1 convolutions to achieve efficient channel compression from 48 to 3 dimensions. This architectural modification can speed up upsampling by a factor of 4 while improving reconstruction quality.

3.4. Reparameterization

Reparameterization is a technique in neural network design where models employ intricate architectures during training phase (e.g, multi-branch with parallel operations) that are subsequently transformed into equivalent but computationally efficient structures during deployment. This transformation usually involves combining multiple parameterized layers (e.g. convolutional blocks and skip connections) into a single convolutional layer. As shown in Figure 2, The proposed reparameterization module adopts a sequential architecture comprising three key operations. Initially, a 1×1 convolutional layer expands the channel dimension by a factor of 4, followed by a 3×3 convolutional layer that extracts spatial features in this elevated feature space. Subsequently, a 1×1 convolutional layer reduces the channel dimension back to its original size, completing the bottleneck structure. In order to maintain dimensionality consistency, a residual connection is implemented using 1×1 convolution. This architecture design effectively deepens the network while maintaining computational efficiency, thereby enhancing the feature extraction capabilities of the model. The synergistic combination of channel expansion/reduction and residual learning mechanisms ultimately strengthens the representational capabilities of the model without introducing significant computational overhead.

3.5. Neural Architecture Search

To achieve the best balance between reconstruction quality and runtime efficiency, we use FGNAS [35] to automat-



Figure 1. RepNet-VSR architecture overview.



Figure 2. RepConv overview.

Factor	Search Space
Convolution types	Normal
Convolution kernel sizes	3
Activation functions	ReLU
The number of channels	0,1,,32
The number of Repblock	0,1,,8

Table 1. The search space of operations.

ically search between reparameterized modules and channels. The search space presented in Table 1 was systematically established, with a focused exploration of channel count variations and reparameterized module quantities to identify optimal parameter configurations within the proposed architectural framework. This method achieves an effective balance between computational efficiency and reconstruction fidelity through NAS.

3.6. Loss

FGNAS seeks to simultaneously maximize task performance accuracy and minimize computational resource consumption in the identified model. To achieve this dual objective, our formulation combines two key components: 1) a primary task-specific loss function that optimizes prediction quality, and 2) a regularization term designed to penalize computational overheads associated with network resources including parameter count, floating-point operations (FLOPs), and inference latency. In our work, FLOPs are incorporated as a regularizer to penalize network complexity, primarily because they offer a computationally efficient metric for optimization.The objective function is formally given by formula (2).

$$\min_{\theta,\psi} L(\psi,\theta) + \lambda \cdot R(\psi) \tag{2}$$

4. Experiments

In this part, we will describe the implementation details of our proposed method and report the results on the REDS validation set.

4.1. Datasets

For this challenge, we employ the REDS [36] dataset, a widely used benchmark for traditional video superresolution tasks due to its extensive content diversity and dynamic scene composition. Following established protocols, we partition the dataset into 240 videos for training, 30 for validation, and 30 for testing. Each video consists of 100-frame sequences, with every frame maintaining a native resolution of 1280×720 pixels at 24 frames per second. To create low-resolution inputs, the original videos undergo bicubic downsampling with a scaling factor of 4. The resultant low-resolution sequences serve as model input, while

Model	nc	nb	PSNR	Runtime(ms)	Score
1	8	4	27.39	56.5	0.0303
2	16	4	27.79	89.6	0.0334
3	32	4	28.01	149.7	0.0271
4	16	5	27.83	93.6	0.0323

Table 2. Training results of different models on the REDS validation set.

the corresponding original high-resolution frames function as the reconstruction target.

4.2. Training Configuration

The models are trained using PyTorch Lightning with mixed precision (FP16) enabled by specifying the precision flag in the Trainer, along with the Adam optimizer configured with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We employ a initial learning rate of 5×10^{-4} . A decaying learning rate scheduler is implemented across all stages, featuring a 500-epoch warm-up period followed by linear decay until the learning rate reaches 1×10^{-8} . We randomly extract HR patches with dimensions 384×384 from high-resolution images and corresponding LR patches of size 96×96 from low-resolution images. The training process consists of two distinct phases. During the initial phase, Neural Architecture Search (NAS) is employed to conduct architectural search, optimizing both channel quantities and the count of re-parameterizable modules. The objective function defined in Equation 2 guides this optimization to maintain an optimal equilibrium between image restoration quality (PSNR) and computational efficiency. In the subsequent optimization phase, the best configuration determined in the first phase is retained, and the reconstruction quality of the pretrained model is adjusted using L2 Loss fine-tuning. All hyperparameters remain consistent in both phases to ensure the continuity of training, and the second phase focuses on further performance optimization on the established architectural framework.

4.3. Runtime evaluation

As previously noted, while our training pipeline utilizes PyTorch Lightning, the efficiency evaluation necessitates a TensorFlow Lite (tflite) model. To achieve this, we transfer the PyTorch model weights into an architecturally equivalent TensorFlow implementation by performing dimension transformations across network layers, ultimately generating a floating-point precision tflite model for runtime evaluation. To check the validity of the model, the challenge requires running it using AI Benchmark on a MediaTek-based device, using FP16 mode + MediaTek neuron delegate; if there are other platforms, use NNAPI / Qualcomm QNN delegate), as the final runtime evaluation will use a similar



Figure 3. Qualitative comparison on the REDS val datasets. Zoom in for better visualization.

setup. Given the current unavailability of the specified hardware infrastructure, we are conducting computational efficiency assessments of the selected model using FP16 precision optimization on the Qualcomm Snapdragon 870's GPU platform. This interim evaluation will be followed by formal coordination with the MAI2025 organizing committee

to perform standardized efficiency benchmarking through their designated testing protocols.

4.4. Ablation Studies

We employ a two-phase training strategy. During the initial phase, Neural Architecture Search (NAS) is utilized to conduct preliminary parameter optimization regarding channel quantities and the number of reparameterization modules. Subsequently, we initialize the model with parameters from the preceding phase and perform fine-tuning using an L2-norm loss function for enhanced performance refinement. As indicated in Table 2, we evaluated the computational efficiency of processing 10 frames using FP16 precision mode on the Qualcomm Snapdragon 870 GPU platform. In addition, Table 2 shows several sets of parameter configurations with relatively high scores under the proposed network architecture. Experimental results indicate that the optimal configuration under the proposed architecture occurs with 16 channels and 4 reparameterization modules.

4.5. Test on MediaTek's NPU

In the final phase of our evaluation, we conducted a comprehensive efficiency assessment on the dedicated Neural Processing Unit (NPU) of MediaTek.

• Quality-PSNR Progression:

Our model (MAI2025) achieves a 1.8% improvement in PSNR over EVSRNet (27.79 vs. 27.42) and a 1.9% improvement over RCBSR [34] (27.79 vs. 27.28), setting a new standard for reconstruction quality and outperforming RCBSR based on ECB reparameterization.

• Computational Efficiency: CPU Utilization:

Our solution shows comparable CPU latency to EVS-RNet (273ms vs 271ms), though 2.44× slower than RCBSR's CPU implementation (273ms vs 112ms) NPU Acceleration:

Maintains parity with EVSRNet (103ms) while achieving 7.5% slower inference than RCBSR (103ms vs 95.8ms) on the NPU of MediaTek.

• Score Metric Analysis:

Our model achieves a 66.7% higher composite score than EVSRNet (0.029 vs. 0.0174) and 88.3% improvement over RCBSR.

From Table 3, it can still achieve a better qualityefficiency trade-off despite longer CPU runtime.

4.6. Visual Comparision

Our proposed RepNetVSR algorithm was evaluated against bicubic interpolation, EVSRNet, and RCBSR in

Model	year	PSNR	CPU Runtime	NPU Runtime	Score
EVSRNet	MAI2021	27.42	271	103	0.0174
RCBSR	MAI2022	27.28	112	95.8	0.0154
our	MAI2025	27.79	273	103	0.029

Table 3. Inference time (ms) of RepNet-VSR on MediaTek NPU.

terms of subjective visual quality, with comparative results demonstrating its significant superiority in perceptual performance, as shown in Figure 3.

5. Conclusion

In this paper, we introduce RepNet-VSR, a reparameterizable architecture designed to optimize inference efficiency while maintaining high restoration quality in video super-resolution (VSR). Our method leverages reparameterization to streamline computational overhead during deployment, enabling real-time performance on mobile devices. When evaluated on the REDS validation set, RepNet-VSR achieves a PSNR of 27.79 dB for 4x super-resolution (180p \rightarrow 720p), processing each 10 frames in 103 ms on a MediaTek Dimensity NPU. These results underscore its exceptional balance between reconstruction fidelity and computational efficiency, addressing the critical challenge of latency-sensitive mobile device applications. In addition, RepNet-VSR surpasses previous competitive algorithms in the MAI Video Super-Resolution Challenge, demonstrating its practical advantages in deployment scenarios.

References

- Andrei Kulik Seungsoo Yang Ke Wang Felix Baum Max Wu Lirong Xu Andrey Ignatov, Radu Timofte and Luc Van Gool. Ai benchmark: All about deep learning on smartphones in 2019.
- [2] William Chou Ke Wang Max Wu Tim Hartley Andrey Ignatov, Radu Timofte and Luc Van Gool. Ai benchmark: Running deep neural networks on android smartphones.
- [3] Byungkon Kang Namhyuk Ahn and Kyung-Ah Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In Proceedings of the European Conference on Computer Vision, pages 252–268, 2018.
- [4] Jie Liu Jie Tang Gangshan Wu Zongcai Du, Ding Liu and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. page 853–862, 2022.
- [5] Juncheng Li Fei Wu Huimin Lu Guangwei Gao, Wenjie Li and Yi Yu. Feature distillation interaction weighting network for lightweight image super-

resolution. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 36, pages 661–669, 2022.

- [6] Yunchu Yang Zheng Hui, Xinbo Gao and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network. *In Proceedings* of the 27th ACM International Conference on Multimedia, pages 2024–2032, 2019.
- [7] C. C.; He K.; Dong, C.; Loy and X. 2015. Tang. Image super-resolution using deep convolutional networks. page 38(2): 295–307.
- [8] Chen Change Loy Chao Dong and Xiaoou Tang. Accelerating the super-resolution convolutional neural network. 2016.
- [9] Jung Kwon Lee Jiwon Kim and Kyoung Mu Lee. Deeply-recursive convolutional network for image super-resolution. 2016.
- [10] Jian Yang Ying Tai and Xiaoming Liu. Image superresolution via deep recursive residual network. 2017.
- [11] Xiumei Wang Zheng Hui and Xinbo Gao. Fast and accurate single image super-resolution via information distillation network.
- [12] Yunchu Yang Zheng Hui, Xinbo Gao and Xiumei Wang. Lightweight image super-resolution with information multi-distillation network.
- [13] Jie Tang Jie Liu and Gangshan Wu. Residual feature distillation network for lightweight image superresolution.
- [14] Shi Guo Xindong Zhang, Hui Zeng and Lei Zhang. Efficient long-range attention network for image superresolution.
- [15] Shizhuo Liu Xiaobing Wang Dafeng Zhang, Feiyu Huang and Zhezhu Jin. Swinfir:revisiting the swinir with fast fourier convolution and improved training for image super-resolution.
- [16] et al. Lu, Zhisheng. Transformer for single image super-resolution.
- [17] Greg Shakhnarovich Muhammad Haris and Norimichi Ukita. Recurrent back-projection network for video superresolution.
- [18] Yun Fu Yapeng Tian, Yulun Zhang and Chenliang Xu. Tdan: Temporally deformable alignment network for video super-resolution.

- [19] Ke Yu Chao Dong Xintao Wang, Kelvin C.K. Chan and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks.
- [20] Shuhang Gu Dario Fuoli and Radu Timofte. Efficient video super-resolution through recurrent latent space propagation.
- [21] Wei Wang Yan Huang and Liang Wang. Bidirectional recurrent convolutional networks for multi-frame superresolution.
- [22] Shuhang Gu Songjiang Li Shengjin Wang Takashi Isobe, Xu Jia and Qi Tian. Video superresolution with recurrent structure-detail network.
- [23] Ke Yu Chao Dong Kelvin C.K. Chan, Xintao Wang and Chen Change Loy. Basicvsr: The search for essential components in video super-resolution and beyond.
- [24] Xu Jia ShengjinWang. Takashi Isobe, Fang Zhu. Revisiting temporal modeling for video superresolution.
- [25] Zhang et al. Neural architecture search for lightweight non-local networks.
- [26] Chu et al. Fast, accurate, and lightweight superresolution with neural architecture search.
- [27] Mei et al. Multi-objective neural architecture search for fast and accurate image super-resolution.
- [28] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13733–13742, 2021.
- [29] Jie Liu, Jie Tang, and Gangshan Wu. Ecbsr: Edge-oriented convolution block for real-time superresolution. *IEEE Transactions on Image Processing*, 31:4529–4543, 2022.
- [30] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Repsr: Training efficient vggstyle super-resolution networks with structural reparameterization. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5720–5729, 2022.
- [31] Yi Zhang, Kun Li, Wenqi Li, and Yunhui Liu. Swinrep: Swin-transformer with structural reparameterization for efficient image restoration. arXiv preprint arXiv:2303.15348, 2023.

- [32] Liu, Shaoli et al. Evsrnet: Efficient video superresolution with neural architecture search. *IEEE Conference on Computer Vision and Pattern Recognition* (*CVPR*), 2021.
- [33] Romero A., Kim H., Timofte R., Ho C. M., and Meng Z. et al. Real-time video super-resolution on smartphones with deep learning, mobile ai 2021 challenge: report. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.
- [34] Andrey Ignatov, Radu Timofte, Cheng-Ming Chiang et al. Power efficient video super-resolution on mobile npus with deep learning, mobile ai & aim 2022 challenge: Report. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.
- [35] Kim H., Hong S., and Han B. et al. Fine-grained neural architecture search. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019.
- [36] Seokil Hong Gyeongsik Moon Sanghyun Son Radu Timofte Seungjun Nah, Sungyong Baik and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and superresolution: Dataset and study. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pages 0–0, 2019.