Power Efficient Video Super-Resolution on Mobile NPUs with Deep Learning, Mobile AI & AIM 2022 challenge: Report

Andrey Ignatov, Radu Timofte, Cheng-Ming Chiang, Hsien-Kai Kuo, Yu-Syuan Xu, Man-Yu Lee, Allen Lu, Chia-Ming Cheng, Chih-Cheng Chen, Jia-Ying Yong, Hong-Han Shuai, Wen-Huang Cheng, Zhuang Jia, Tianyu Xu, Yijian Zhang, Long Bao, Heng Sun, Diankai Zhang, Si Gao, Shaoli Liu, Biao Wu, Xiaofeng Zhang, Chengjian Zheng, Kaidi Lu, Ning Wang, Xiao Sun, HaoDong Wu, Xuncheng Liu, Weizhan Zhang, Caixia Yan, Haipeng Du, Qinghua Zheng, Qi Wang, Wangdu Chen, Ran Duan, Ran Duan, Mengdi Sun, Dan Zhu, Guannan Chen, Hojin Cho, Steve Kim, Shijie Yue, Chenghua Li, Zhengyang Zhuge, Wei Chen, Wenxu Wang, Yufeng Zhou, Xiaochen Cai, Hengxing Cai, Kele Xu, Li Liu, Zehua Cheng, Wenyi Lian, and Wenjing Lian *

Abstract. Video super-resolution is one of the most popular tasks on mobile devices, being widely used for an automatic improvement of low-bitrate and low-resolution video streams. While numerous solutions have been proposed for this problem, they are usually quite computationally demanding, demonstrating low FPS rates and power efficiency on mobile devices. In this Mobile AI challenge, we address this problem and propose the participants to design an end-to-end real-time video super-resolution solution for mobile NPUs optimized for low energy consumption. The participants were provided with the REDS training dataset containing video sequences for a 4X video upscaling task. The runtime and power efficiency of all models was evaluated on the powerful MediaTek Dimensity 9000 platform with a dedicated AI processing unit capable of accelerating floating-point and quantized neural networks. All proposed solutions are fully compatible with the above NPU, demonstrating an up to 500 FPS rate and 0.2 [Watt / 30 FPS] power consumption. A detailed description of all models developed in the challenge is provided in this paper.

Keywords: Mobile AI Challenge, Video Super-Resolution, Mobile NPUs, Mobile AI, Deep Learning, MediaTek, AI Benchmark

* Andrey Ignatov, Radu Timofte, Cheng-Ming Chiang, Hsien-Kai Kuo, Hong-Han Shuai and Wen-Huang Cheng are the main Mobile AI & AIM 2022 challenge organizers. The other authors participated in the challenge.

Appendix A contains the authors’ team names and affiliations.

Mobile AI 2022 Workshop website:
https://ai-benchmark.com/workshops/mai/2022/


1 Introduction

The widespread of mobile devices and the increased demand for various video streaming services from the end users led to a pressing need for video super-resolution solutions that are both efficient and low power device-friendly. A large number of accurate deep learning-based solutions have been introduced for video super-resolution \cite{57,59,62,50,49} in the past. Unfortunately, most of these solutions while achieving very good fidelity scores are lacking in terms of computational efficiency and complexity as they are not optimized to meet the specific hardware constraints of mobile devices. Such an optimization is key for processing tasks related to image \cite{16,35} and video \cite{58} enhancement on mobile devices. In this challenge, we add efficiency-related constraints on the developed video super-resolution solutions that are validated on mobile NPUs. We employ REDS \cite{57}, a popular video super-resolution dataset.

The deployment of AI-based solutions on portable devices usually requires an efficient model design based on a good understanding of the mobile processing units (e.g., CPUs, NPUs, GPUs, DSP) and their hardware particularities, including their memory constraints. We refer to \cite{29,25} for an extensive overview of mobile AI acceleration hardware, its particularities and performance. As shown in these works, the latest generations of mobile NPUs are reaching the performance of older-generation mid-range desktop GPUs. Nevertheless, a straightforward deployment of neural networks-based solutions on mobile devices is impeded by (i) a limited memory (i.e., restricted amount of RAM) and (ii) a limited or lacking support of many common deep learning operators and layers. These impeding factors make the processing of high resolution inputs impossible with the standard NN models and require a careful adaptation or re-design to the constraints of mobile AI hardware. Such optimizations can employ a combination of various model techniques such as 16-bit / 8-bit \cite{5,39,38,71} and low-bit \cite{4,67,36,53} quantization, network pruning and compression \cite{5,21,15,60}, device- or NPU-specific adaptations, platform-aware neural architecture search \cite{11,64,70,68}, etc.

The majority of competitions aimed at efficient deep learning models use standard desktop hardware for evaluating the solutions, thus the obtained models rarely show acceptable results when running on real mobile hardware with many specific constraints. In this Mobile AI challenge, we take a radically different approach and propose the participants to develop and evaluate their models directly on mobile devices. The goal of this competition is to design a fast and power efficient deep learning-based solution for video super-resolution problem. For this, the participants were provided with a large-scale REDS \cite{57} dataset containing the original high-quality and downscaled by a factor of 4 videos. The efficiency of the proposed solutions was evaluated on the MediaTek Dimensity 9000 mobile SoC with a dedicated AI Processing Unit (APU) capable of running floating-point and quantized models. The overall score of each submission was computed based on its fidelity, power consumption and runtime results, thus balancing between the image reconstruction quality and the efficiency of the model. All solutions developed in this challenge are fully compatible with the
Fig. 1. Loading and running custom TensorFlow Lite models with AI Benchmark application. The currently supported acceleration options include Android NNAPI, TFLite GPU, Hexagon NN, Qualcomm QNN, MediaTek Neuron and Samsung ENN delegates as well as CPU inference through TFLite or XNNPACK backends. The latest app version can be downloaded at https://ai-benchmark.com/download

TensorFlow Lite framework [65], thus can be executed on various Linux and Android-based IoT platforms, smartphones and edge devices.

This challenge is a part of the Mobile AI & AIM 2022 Workshops and Challenges consisting of the following competitions:

- Power Efficient Video Super-Resolution on Mobile NPUs
- Quantized Image Super-Resolution on Mobile NPUs [27]
- Learned Smartphone ISP on Mobile GPUs [33]
- Efficient Single-Image Depth Estimation on Mobile Devices [20]
- Realistic Bokeh Effect Rendering on Mobile GPUs [34]
- Super-Resolution of Compressed Image and Video [72]
- Reversed Image Signal Processing and RAW Reconstruction [6]
- Instagram Filter Removal [43]

The results and solutions obtained in the previous MAI 2021 Challenges are described in our last year papers:

- Single-Image Depth Estimation on Mobile Devices [17]
- Learned Smartphone ISP on Mobile NPUs [14]
- Real Image Denoising on Mobile GPUs [15]
- Quantized Image Super-Resolution on Mobile NPUs [26]
- Real-Time Video Super-Resolution on Mobile GPUs [61]
- Quantized Camera Scene Detection on Smartphones [18]
2 Challenge

In order to design an efficient and practical deep learning-based solution for the considered task that runs fast on mobile devices, one needs the following tools:

1. A large-scale high-quality dataset for training and evaluating the models. Real, not synthetically generated data should be used to ensure a high quality of the obtained model;
2. An easy way to check the runtime and debug the model locally without any constraints as well as the ability to get the runtime and energy consumption on the target evaluation platform.

This challenge addresses all the above issues. Real training data, tools, runtime and power consumption evaluation options provided to the challenge participants are described in the next sections.

2.1 Dataset

In this challenge, we use the REDS [57] dataset that serves as a benchmark for traditional video super-resolution task as it contains a large diversity of content and dynamic scenes. Following the standard procedure, we use 240 videos for training, 30 videos for validation, and 30 videos for testing. Each video has sequences of length 100, where every sequence contains video frames of $1280 \times 720$ resolution at 24 fps. To generate low-resolution data, the videos were bicubically downsampled with a factor of 4. The low-resolution video data is then considered as input, and the high-resolution — are the target.

2.2 Local Runtime Evaluation

When developing AI solutions for mobile devices, it is vital to be able to test the designed models and debug all emerging issues locally on available devices. For this, the participants were provided with the AI Benchmark application [25,29] that allows to load any custom TensorFlow Lite model and run it on any Android device with all supported acceleration options. This tool contains the latest versions of Android NNAPI, TFLite GPU, MediaTek Neuron, Hexagon NN, Qualcomm QNN and Samsung ENN delegates, therefore supporting all current mobile platforms and providing the users with the ability to execute neural networks on smartphone NPUs, APUs, DSPs, GPUs and CPUs.

To load and run a custom TensorFlow Lite model, one needs to follow the next steps:

1. Download AI Benchmark from the official website\footnote{https://ai-benchmark.com/download} or from the Google Play\footnote{https://play.google.com/store/apps/details?id=org.benchmark.demo} and run its standard tests.
2. After the end of the tests, enter the PRO Mode and select the Custom Model tab there.
3. Rename the exported TFLite model to \textit{model.tflite} and put it into the Download folder of the device.
4. Select mode type (\textit{INT8}, \textit{FP16}, or \textit{FP32}), the desired acceleration/inference options and run the model.

These steps are also illustrated in Fig. [1]

2.3 Runtime Evaluation on the Target Platform

In this challenge, we use the \textit{MediaTek Dimensity 9000} SoC as our target runtime evaluation platform. This chipset contains a powerful APU [14] capable of accelerating floating point, INT16 and INT8 models, being ranked first by AI Benchmark at the time of its release. Within the challenge, the participants were able to upload their TFLite models to a dedicated validation server connected to a real device and get an instantaneous feedback: the power consumption and the runtime of their solution on the Dimensity 9000 APU or a detailed error log if the model contains some incompatible operations. The models were parsed and accelerated using MediaTek Neuron delegate\footnote{https://github.com/MediaTek-NeuroPilot/tflite-neuron-delegate}. The same setup was also used for the final model evaluation. The participants were additionally provided with a detailed model optimization guideline demonstrating the restrictions and the most efficient setups for each supported TFLite op. A comprehensive tutorial demonstrating how to work with the data and how to train a baseline MobileRNN model on the provided videos was additionally released to the participants:\footnote{https://github.com/MediaTek-NeuroPilot/mai22-real-time-video-sr}

2.4 Challenge Phases

The challenge consisted of the following phases:

I. \textit{Development}: the participants get access to the data and AI Benchmark app, and are able to train the models and evaluate their runtime locally;

II. \textit{Validation}: the participants can upload their models to the remote server to check the fidelity scores on the validation dataset, to get the runtime and energy consumption on the target platform, and to compare their results on the validation leaderboard;

III. \textit{Testing}: the participants submit their final results, codes, TensorFlow Lite models, and factsheets.

2.5 Scoring System

All solutions were evaluated using the following metrics: \footnote{https://github.com/MediaTek-NeuroPilot/tflite-neuron-delegate}
## Challenge Results

From above 160 registered participants, 11 teams entered the final phase and submitted valid results. TFLite models, codes, executables and factsheets. Table 1 summarizes the final challenge results and reports PSNR, SSIM and runtime and power consumption numbers for each submitted solution on the final test dataset and on the target evaluation platform. The proposed methods are described in section 4 and the team members and affiliations are listed in Appendix A.

### Table 1. Mobile AI 2022 Power Efficient Video Super-Resolution challenge results and final rankings. During the runtime and power consumption measurements, the models were upscaling video frames from 180×320 to 1280×720 pixels on the MediaTek Dimensity 9000 chipset. Teams MVideoSR and ZX_VIP are the challenge winners.

| Team           | Author         | Framework          | Model Size, KB | PSNR   | SSIM   | Runtime, ms | Power Consumption, W@30FPS | Final Score |
|----------------|----------------|--------------------|----------------|--------|--------|-------------|---------------------------|-------------|
| MVideoSR       | OptimizePrime  | PyTorch / TensorFlow 17 | 27.34 | 0.7982 | 3.04 | 0.10 | 90.9 |
| ZX_VIP         | erick          | PyTorch / TensorFlow 20 | 27.54 | 0.7867 | 3.24 | 0.20 | 87.4 |
| FightX         | ex             | TensorFlow 11      | 27.34 | 0.7816 | 3.41 | 0.20 | 85.4 |
| XITU-MIGU SUPER | xinchengliang  | TensorFlow 20      | 27.77 | 0.7953 | 3.25 | 0.22 | 85.1 |
| DOICT-ABRO     | Ductile        | TensorFlow 40      | 27.71 | 0.7820 | 1.97 | 0.24 | 84.0 |
| GunMedia Group | steveki        | TensorFlow 135     | 27.71 | 0.7820 | 1.97 | 0.24 | 84.0 |
| NCUT Vipo      | rej            | TensorFlow 35      | 27.46 | 0.7822 | 1.39 | 0.46 | 75.6 |
| Mortar ICT     | work mai       | TensorFlow 75      | 22.91 | 0.7546 | 1.76 | 0.36 | 70.0 |
| RedCat AutoX   | roxin          | TensorFlow 62      | 27.71 | 0.7945 | 7.26 | 0.53 | 69.5 |
| ZoomH          | shermanliang   | TensorFlow 1810    | 28.19 | 0.8171 | 26.8 | 3.73 | -89.3 |
| SuperDash      | xiaoxuan       | TensorFlow 1810    | 28.40 | 0.8171 | 26.8 | 3.73 | -89.3 |

### Notes:
- Peak Signal-to-Noise Ratio (PSNR) measuring fidelity score,
- Structural Similarity Index Measure (SSIM), a proxy for perceptual score,
- The energy consumption on the target Dimensity 9000 platform,
- The runtime on the Dimensity 9000 platform.

The score of each final submission was evaluated based on the next formula:

\[
\text{Final Score} = \alpha \cdot \text{PSNR} + \beta \cdot (1 - \text{power consumption}),
\]

where \(\alpha = 1.66\) and \(\beta = 50\). Besides that, the runtime of the solutions should be less than 33 ms when reconstructing one Full HD video frame on the Dimensity 9000 NPU (thus achieving a real-time performance of 30 FPS), otherwise their final score was set to 0. The energy consumption was computed as the amount of watts consumed when processing 30 subsequent video frames (in 1 second).

During the final challenge phase, the participants did not have access to the test dataset. Instead, they had to submit their final TensorFlow Lite models that were subsequently used by the challenge organizers to check both the runtime and the fidelity results of each submission under identical conditions. This approach solved all the issues related to model overfitting, reproducibility of the results, and consistency of the obtained runtime/accuracy values.
3.1 Results and Discussion

All solutions proposed in this challenge demonstrated a very high efficiency, achieving a real-time performance of more than 30 FPS on the target MediaTek Dimensity 9000 platform. The majority of models followed a simple single-frame restoration approach to improve the runtime and power efficiency. Only one model (from GenMedia Group team) used all 10 input video frames as an input, and two teams (RedCat AutoX and 221B) proposed RNN-based solutions that used two and three subsequent input video frames, respectively. All designed models have shallow architectures with small convolutional filters and channel sizes. The majority of networks used the depth-to-space (pixel shuffle) op at the end of the model instead of the transposed convolution to avoid the computations when upsampling the images. The sizes of all designed architectures are less than 200 KB, except for the solution from team SuperDash that submitted the baseline MobileRNN network.

Teams MVideoSR and ZX_VIP are the challenge winners. The speed of their models is over 300 FPS on the MediaTek Dimensity 9000 APU, while the power consumption is less than 0.1 watt when processing 30 video frames. Both networks used a single-frame upsampling structure with one residual block and a pixel shuffle op at the end. In terms of the runtime, one of the best results was achieved by team BOE-IOT-AIBD, which solution demonstrated over 500 FPS on MediaTek’s APU while also showing quite decent fidelity scores. The best fidelity-runtime balance was achieved by the solution proposed by team GenMedia Group: its PSNR and SSIM scores are just slightly behind a considerably larger and slower MobileRNN baseline, though its runtime is just around 3 milliseconds per one video frame, same as for the solutions from the first two teams. One can also notice that RNN-based solutions are considerably slower and more power demanding, though they do not necessarily lead to better perceptual and fidelity results. Therefore, we can conclude that right now the standard single-or multi-frame restoration CNN models are more suitable for on-device video super-resolution in terms of the balance between the runtime, power efficiency and visual video restoration quality.

4 Challenge Methods

This section describes solutions submitted by all teams participating in the final stage of the MAI 2022 Real-Time Video Super-Resolution challenge.

4.1 MVideoSR

In order to meet the low power consumption requirement, the authors used a simple network structure shown in Fig. 2 that has 6 layers, of which only 5 have learnable parameters (4 conv layers and a PReLU activation layer). All intermediate layers have 6 feature channels, pixel shuffle operation is used to upscale the image to the target resolution without performing any computations.
When choosing the activation function, the authors compared the commonly used ReLU, Leaky ReLU and Parametric ReLU (PReLU) activations, and the results implied that the PReLU can boost the performance by about 0.05 dB due to its higher flexibility. At the same time, PReLU op has nearly the same power consumption as ReLU / LeakyReLU, thus this activation layer was selected.

The authors first trained the proposed network to perform a 2× image super-resolution using the same REDS dataset. For this, the last conv layer of the model was modified to have weights of size [12, 8, 3, 3]. After pre-training, these weights were kept and repeated 4 times along the channel dimension (Fig. 2, bottom) to comply with the spatial position of the output obtained after pixel shuffle when performing a 4× super-resolution task. This strategy was used to accelerate the convergence of the model on the final task. Overall, the training of the model was done in five stages:

1. First, the authors trained a 2× super-resolution model from scratch as stated above with a batch size of 64 and the target (high-resolution) patch size of 256×256. The model was optimized using the L1 loss and the Adam algorithm for 500K iterations, the initial learning rate was set to 5e–4 and decreases by 0.5 after 200K and 400K iterations.
2. Next, a 4× super-resolution model was initialized with weights obtained during the first stage. The same training setup was used except for the learning rate that was initialized with 5e–5 and decreases by 0.5 after 100K, 300K and 450K iterations. After that, the learning rate was set to 2e–4 and
the model was trained again for 500K iterations, where the learning rate was decreased by 0.5 after 200K iterations.

3. The model was then fine-tuned with the MSE loss function for 1000K iterations with a learning rate of $2 \times 10^{-4}$ decreases by 0.5 after 300K, 600K and 900K iterations.

4. During the next 500K iterations, the model was further fine-tuned with the MSE loss, but the target patch size was changed to $512 \times 512$. This stage took 500K iterations, the learning rate was set to $2 \times 10^{-5}$ and decreases by 0.5 after 100K, 200K, 300K and 400K iterations.

5. Finally, the obtained model was fine-tuned for another 50K iterations on patches of size $640 \times 640$ with a learning rate of $2 \times 10^{-5}$.

4.2 ZX VIP

Inspired by the architecture of the edge-oriented convolution block (ECB) [74], team ZX VIP developed a re-parametrization edge-oriented convolution block.
(RCBSDR) shown in Fig. 3. During the training stage, the authors took the standard ECB block as a basic backbone unit since it can extract expressive features due to its multi-branch structure. PReLU activations were replaced with ReLU in this block for a greater efficiency. During the inference stage, the multi-branch structure of the ECB was merged into one single $3 \times 3$ convolution by using a re-parametrization trick. The authors used the depth-to-space op at the end of the model to produce the final output without any computations. The overall model architecture is shown in Fig. 3 the authors used only one ECB block ($M = 1$) in their final model. The network was optimized in two stages: first, it was trained with a batch size of 64 on 512×512 pixel images augmented with random flips and rotations. Charbonnier loss was used as the target metric, model parameters were optimized for 4000 epochs using the Adam algorithm with a learning rate initialized at 5e–4 and decreases by half every 1000 epochs. In the second stage, the model was fine-tuned with the L2 loss and a learning rate initialized at 2e–4.

### 4.3 Fighter

The architecture proposed by team Fighter is illustrated in Fig. 4. The authors used a shallow CNN model with depthwise separable convolutions and one residual connection. The number of convolution channels in the model was set to 8, the depth-to-space op was used at the end of the model to produce the final output. The model was trained for 1500 epochs with the initial learning rate of 10e–3 dropped by half every 240 epochs.

### 4.4 XJTU-MIGU SUPER

The architecture developed by team XJTU-MIGU SUPER is shown in Fig. 5. The model has a depthwise separable convolutional structure with residual connections. The number of convolution channels is set to 16, and the model includes a depth-to-space operation at the end to produce the final output. The model was trained for 1500 epochs with an initial learning rate of 10e–3 dropped by half every 240 epochs.
Team XJTU-MIGU SUPER proposed a small CNN-based model originated from genetic algorithm shown in Fig. 5. The network has 16 feature channels in the 1st, 2nd and 3rd layers, and 48 channels in the last convolutional layer. ReLu was used as an activation function in the 2nd and 3rd layers, while the 1st and 4th layers were not followed by any activations. The pixel shuffle operation is used at the end of model to produce the final image output while minimizing the number of computations. First, the models are trained to minimize the L1 loss for 200 epochs using the Adam optimizer with a batch size of 4 and the initial learning rate of 1.6e–2. Then, the learning rate was changed to 0.12 and the model was fine-tuned for another 800 epochs with the batch size of 64. Finally, the batch size was changed again to 4 and the model was trained for another 1600 epochs on the whole dataset.

4.5 BOE-IOT-AIBD

Fig. 6. The architecture of the model developed by team BOE-IOT-AIBD.

The solution developed by team BOE-IOT-AIBD is based on the CNN-Net [41] architecture, its structure is illustrated in Fig 6. This model consists of six convolutional layers and one transposed convolutional layer. ReLU activations are used after the first four and the last convolutional layer, the number of feature channels was set to 25. The authors applied model distillation [10] when training the network, and used the RFDN [51] CNN as a teacher model. In the training phase, the MSE loss function was calculated between the main model’s output and the corresponding output from the teacher model, and between the feature maps obtained after the last concat layer of both models having 50 feature channels. The model was trained on patches of size 60 × 80 pixels with a batch size of 4. Network parameters were optimized using the Adam algorithm with a constant learning rate of 1e–4.

4.6 GenMedia Group

The model architecture proposed by team GenMedia Group is inspired by the last year’s top solution [7] from the MAI image super-resolution challenge [26]. The authors added one extra skip connection to the mentioned anchor-based
plain net (ABPN) model, and used concatenation followed by a $1 \times 1$ convolution instead of a pixel-wise summation at the end of the network (Fig. 7). To avoid potential context switching between CPU and GPU, the order of the clipping function and the depth-to-space operation was rearranged. The authors used the mean absolute error (MAE) as the target objective function. The model was trained on patches of size $96 \times 96$ pixels with a batch size of 32. Network parameters were optimized for 1000 epochs using the Adam algorithm with a learning rate of $1 \times 10^{-3}$ decreased by half every 200 epochs. Random horizontal and vertical image flips were used for data augmentation.

4.7 NCUT VGroup

Fig. 8. Model architecture proposed by team NCUT.
Team NCUT VGroup also based their solution on the ABPN model (Fig. 8). To improve the runtime, the authors removed four convolution layers and residual connection, thus the resulting network consists of only three convolution layers. The model was trained to minimize the mean error (MAE) loss on patches of size 64 × 64 pixels with a batch size of 16. Network parameters were optimized for 168K iterations using the Adam algorithm with a learning rate of 1e−3 reduced to 1e−8 with the cosine annealing. Random horizontal and vertical image flips were used for data augmentation.

4.8 Mortar ICT

Team Mortar ICT used a simple CNN model for the considered video super-resolution task (Fig. 9). The proposed network consists of eight 3 × 3 convolutional layers with 32 channels followed by a depth-to-space operation. The authors additionally used a re-parametrization method during the training to expand 3 × 3 convolutions to 3 × 3 + 1 × 1 convs that were fused at the inference stage. The model was trained to minimize the L1 loss on patches of size 64 × 64 pixels. Network parameters were optimized for 100 epochs using the Adam algorithm with a learning rate of 5e−4 that was decreased by half every 30 epochs.

4.9 RedCat AutoX

Team RedCat AutoX used the provided MobileRNN baseline to design their solution (Fig. 10). To improve the model complexity, the authors replaced the residual block in this model with information multi-distillation blocks (IMDBs) and adjusted the number of blocks and base channels to 5 and 8, respectively. The model has a the recurrent residual structure and takes two adjacent video frames \( I_t, I_{t+1} \) as an input to see the forward information. To be specific, at time \( t \) the model take three parts as an input: the previous hidden state \( h_{t-1} \), the current frame at time \( t \), and the forward frame at time \( t + 1 \). A global skip connection, where the input frame is upsampled with a bilinear operation and added to the final output, is used to improve the fidelity and stability of the model. During the training process, L1 loss was used as the target metric. The network was trained for 150K iterations using the Adam algorithm with an initial learning rate of 3e−3 halved every 15K steps. The batch size was set to 8, random clipping, horizontal and vertical flips were used for data augmentation.
Fig. 10. RNN-based model proposed by team RedCat.

4.10 221B

Fig. 11. Recurrent network architecture proposed by team 221B.

Team 221B developed an RNN-based model architecture illustrated in Fig. 11. At each time step $t$, the network accepts three subsequent video frames (previous, current and future) and produces the target frame, which is similar to
a sliding window multi-frame super-resolution algorithm \cite{69,18,55,56}. Inspired by \cite{37}, the authors take advantage of recurrent hidden states to preserve the previous and future information. Specifically, the initial hidden states (forward and backward) are set to 0 and are updated when the window slides to the next frames. Here, the previous frame $x_{t-1}$, the current frame $x_t$ and the forward hidden state are concatenated as a forward group, then the future frame $x_{t+1}$, the current frame $x_t$ and the backward hidden state compose a backward group. Deep features for each group are separately extracted and concatenated to aggregate a multi-frame information to reconstruct the target high-resolution frame. Meanwhile, the extracted features of the forward and backward groups update the corresponding forward and backward hidden states, respectively. The model uses only $3 \times 3$ convolution layers with ReLU activation, a bilinearly upsampled current frame is taken as a residual connection to improve the restoration accuracy \cite{46}.

The model was trained to minimize the Charbonnier loss function on patches of size $64 \times 64$ pixels with a batch size of 16. Network parameters were optimized for 150K iterations using the Adam algorithm with a learning rate of $1e^{-3}$ decreased by half every 50K iterations.

5 Additional Literature

An overview of the past challenges on mobile-related tasks together with the proposed solutions can be found in the following papers:

- Video Super-Resolution: \cite{59,58,23,19}
- Image Super-Resolution: \cite{31,64,3,66}
- Learned End-to-End ISP: \cite{28,32}
- Perceptual Image Enhancement: \cite{31,24}
- Bokeh Effect Rendering: \cite{22,30}
- Image Denoising: \cite{1,2}

Acknowledgements

We thank the sponsors of the Mobile AI and AIM 2022 workshops and challenges: AI Witchlabs, MediaTek, Huawei, Reality Labs, OPPO, Synaptics, Raspberry Pi, ETH Zürich (Computer Vision Lab) and University of Würzburg (Computer Vision Lab).

A Teams and Affiliations

Mobile AI & AIM 2022 Team

*Title:* Mobile AI & AIM 2022 Video Super-Resolution Challenge
Members:
Andrey Ignatov\textsuperscript{1,2} (andrey@vision.ee.ethz.ch), Radu Timofte\textsuperscript{1,2,3} (radu.timofte@uni-wuerzburg.de), Cheng-Ming Chiang\textsuperscript{4} (jimmy.chiang@mediatek.com), Hsien-Kai Kuo\textsuperscript{4} (hsienkai.kuo@mediatek.com), Yu-Syuan Xu\textsuperscript{4} (yu-syuan.xu@mediatek.com), Man-Yu Lee\textsuperscript{4} (my.lee@mediatek.com), Allen Lu\textsuperscript{4} (allen-cl.lu@mediatek.com), Chia-Ming Cheng\textsuperscript{4} (cm.cheng@mediatek.com), Chih-Cheng Chen\textsuperscript{4} (ryan.chen@mediatek.com), Jia-Ying Yong\textsuperscript{4} (jiaying.ee10@nycu.edu.tw), Hong-Han Shuai\textsuperscript{5} (hhshuai@nycu.edu.tw), Wen-Huang Cheng\textsuperscript{5} (whcheng@nycu.edu.tw)

Affiliations:
\begin{itemize}
\item [1] Computer Vision Lab, ETH Zurich, Switzerland
\item [2] AI Witchlabs, Switzerland
\item [3] University of Wuerzburg, Germany
\item [4] MediaTek Inc., Taiwan
\item [5] National Yang Ming Chiao Tung University, Taiwan
\end{itemize}

MVideoSR

Title:
Extreme Low Power Network for Real-time Video Super Resolution

Members:
Zhuang Jia (jiazhuang@xiaomi.com), Tianyu Xu (xutianyu@xiaomi.com), Yi-jian Zhang (zhangyijian@xiaomi.com), Long Bao (baolong@xiaomi.com), Heng Sun (sunheng3@xiaomi.com)

Affiliations:
Video Algorithm Group, Camera Department, Xiaomi Inc., China

ZX VIP

Title:
Real-Time Video Super-Resolution Model \cite{9}

Members:
Diankai Zhang (zhang.diankai@zte.com.cn), Si Gao, Shaoli Liu, Biao Wu, Xiaofeng Zhang, Chengjian Zheng, Kaidi Lu, Ning Wang

Affiliations:
Audio & Video Technology Platform Department, ZTE Corp., China

Fighter

Title:
Fast Real-Time Video Super-Resolution

Members:
Xiao Sun (2609723059@qq.com), HaoDong Wu
**Affiliations:**
None, China

**XJTU-MIGU SUPER**

**Title:**
Light and Fast On-Mobile VSR

**Members:**
Xuncheng Liu (liuxuncheng123@stu.xjtu.edu.cn), Weizhan Zhang, Caixia Yan, Haipeng Du, Qinghua Zheng, Qi Wang, Wangdu Chen

**Affiliations:**
School of Computer Science and Technology, Xi’an Jiaotong University, China
MIGU Video Co. Ltd, China

**BOE-IOT-AIBD**

**Title:**
Lightweight Quantization CNN-Net for Mobile Video Super-Resolution

**Members:**
Ran Duan (duanr@boe.com.cn), Ran Duan, Mengdi Sun, Dan Zhu, Guannan Chen

**Affiliations:**
BOE Technology Group Co., Ltd., China

**GenMedia Group**

**Title:**
SkipSkip Video Super-Resolution

**Members:**
Hojin Cho (jin@gengen.ai), Steve Kim

**Affiliations:**
GenGenAI, South Korea

**NCUT VGroup**

**Title:**
EESRNet: A Network for Energy Efficient Super Resolution

**Members:**
Shijie Yue (1161126955@qq.com), Chenghua Li, Zhengyang Zhuge

**Affiliations:**
North China University of Technology, China
Institute of Automation, Chinese Academy of Sciences, China
Mortar ICT

Title: Real-Time Video Super-Resolution Model

Members: Wei Chen (chenwei21s@ict.ac.cn), Wenxu Wang, Yufeng Zhou

Affiliations: State Key Laboratory of Computer Architecture, Institute of Computing Technology, China

RedCat AutoX

Title: Forward Recurrent Residual Network

Members: Xiaochen Cai (caixc@lamda.nju.edu.cn), Hengxing Cai, Kele Xu, Li Liu, Ze-hua Cheng

Affiliations: 1. Paradigm Inc., Beijing, China
2. National University of Defense Technology, Changsha, China
3. University of Oxford, Oxford, United Kingdom

221B

Title: Sliding Window Recurrent Network for Efficient Video Super-Resolution [47]

Members: Wenyi Lian (shermanlian@163.com), Wenjing Lian

Affiliations: Uppsala University, Sweden
Northeastern University, China

References

1. Abdelhamed, A., Afifi, M., Timofte, R., Brown, M.S.: Ntire 2020 challenge on real image denoising: Dataset, methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 496–497 (2020)
2. Abdelhamed, A., Timofte, R., Brown, M.S.: Ntire 2019 challenge on real image denoising: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
3. Cai, J., Gu, S., Timofte, R., Zhang, L.: Ntire 2019 challenge on real image super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
4. Cai, Y., Yao, Z., Dong, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Zeroq: A novel zero shot quantization framework. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13169–13178 (2020)
5. Chiang, C.M., Tseng, Y., Xu, Y.S., Kuo, H.K., Tsai, Y.M., Chen, G.Y., Tan, K.S., Wang, W.T., Lin, Y.C., Tseng, S.Y.R., et al.: Deploying image deblurring across mobile devices: A perspective of quality and latency. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 502–503 (2020)
6. Conde, M.V., Timofte, R., et al.: Reversed Image Signal Processing and RAW Reconstruction. AIM 2022 Challenge Report. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2022)
7. Du, Z., Liu, J., Tang, J., Wu, G.: Anchor-based plain net for mobile image super-resolution. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2021)
8. Fuoli, D., Gu, S., Timofte, R.: Efficient video super-resolution through recurrent latent space propagation. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3476–3485. IEEE (2019)
9. Gao, S., Zheng, C., zhang, X., shaoli liu, Wu, B., Lu, K., Zhang, D., Wang, N.: Rcbr: Re-parameterization convolution block for super-resolution. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2022)
10. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
11. Howard, A., Sandler, M., Chu, G., Chen, L.C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., et al.: Searching for mobilenetv3. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1314–1324 (2019)
12. Hui, Z., Gao, X., Yang, Y., Wang, X.: Lightweight image super-resolution with information multi-distillation network. In: Proceedings of the 27th ACM International Conference on Multimedia. pp. 2024–2032 (2019)
13. Ignatov, A., Byeoung-su, K., Timofte, R.: Fast camera image denoising on mobile gpus with deep learning, mobile ai 2021 challenge: Report. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2021)
14. Ignatov, A., Chiang, J., Kuo, H.K., Sycheva, A., Timofte, R.: Learned smartphone isp on mobile npus with deep learning, mobile ai 2021 challenge: Report. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2021)
15. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: Dslr-quality photos on mobile devices with deep convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3277–3285 (2017)
16. Ignatov, A., Kobyshev, N., Timofte, R., Vanhoey, K., Van Gool, L.: Wespe: weakly supervised photo enhancer for digital cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 691–700 (2018)
17. Ignatov, A., Malivenko, G., Plowman, D., Shukla, S., Timofte, R.: Fast and accurate single-image depth estimation on mobile devices, mobile ai 2021 challenge: Report. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2021)
18. Ignatov, A., Malivenko, G., Timofte, R.: Fast and accurate quantized camera scene detection on smartphones, mobile ai 2021 challenge: Report. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2021)
19. Ignatov, A., Malivenko, G., Timofte, R., Tseng, Y., Xu, Y.S., Yu, P.H., Chiang, C.M., Kuo, H.K., Chen, M.H., Cheng, C.M., Van Gool, L.: Pynet-v2 mobile: Efficient on-device photo processing with neural networks. In: 2021 26th International Conference on Pattern Recognition (ICPR). IEEE (2022)

20. Ignatov, A., Malivenko, G., Timofte, R., et al.: Efficient single-image depth estimation on mobile devices, mobile ai & aim 2022 challenge: Report. In: European Conference on Computer Vision (2022)

21. Ignatov, A., Patel, J., Timofte, R.: Rendering natural camera bokeh effect with deep learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 418–419 (2020)

22. Ignatov, A., Patel, J., Timofte, R., Zheng, B., Ye, X., Huang, L., Tian, X., Dutta, S., Purohit, K., Kundula, P., et al.: Aim 2019 challenge on bokeh effect synthesis: Methods and results. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3591–3598. IEEE (2019)

23. Ignatov, A., Sycheva, A., Timofte, R., Tseng, Y., Xu, Y.S., Yu, P.H., Chiang, C.M., Kuo, H.K., Chen, M.H., Cheng, C.M., Van Gool, L.: Microisp: Processing 32mp photos on mobile devices with deep learning. In: European Conference on Computer Vision (2022)

24. Ignatov, A., Timofte, R.: Ntire 2019 challenge on image enhancement: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

25. Ignatov, A., Timofte, R., Chou, W., Wang, K., Wu, M., Hartley, T., Van Gool, L.: Ai benchmark: Running deep neural networks on android smartphones. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018)

26. Ignatov, A., Timofte, R., Denna, M., Younes, A.: Real-time quantized image super-resolution on mobile npus, mobile ai 2021 challenge: Report. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2021)

27. Ignatov, A., Timofte, R., Denna, M., Younes, A., et al.: Efficient and accurate quantized image super-resolution on mobile npus, mobile ai & aim 2022 challenge: Report. In: European Conference on Computer Vision (2022)

28. Ignatov, A., Timofte, R., Ko, S.J., Kim, S.W., Uhm, K.H., Ji, S.W., Cho, S.J., Hong, J.P., Mei, K., Li, J., et al.: Aim 2019 challenge on raw to rgb mapping: Methods and results. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3584–3590. IEEE (2019)

29. Ignatov, A., Timofte, R., Kulik, A., Yang, S., Wang, K., Baum, F., Wu, M., Xu, L., Van Gool, L.: Ai benchmark: All about deep learning on smartphones in 2019. In: 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW). pp. 3617–3635. IEEE (2019)

30. Ignatov, A., Timofte, R., Qian, M., Qiao, C., Lin, J., Guo, Z., Li, C., Leng, C., Cheng, J., Peng, J., et al.: Aim 2020 challenge on rendering realistic bokeh. In: European Conference on Computer Vision. pp. 213–228. Springer (2020)

31. Ignatov, A., Timofte, R., Van Vu, T., Minh Luu, T., X Pham, T., Van Nguyen, C., Kim, Y., Choi, J.S., Kim, M., Huang, J., et al.: Pirm challenge on perceptual image enhancement on smartphones: Report. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops. pp. 0–0 (2018)

32. Ignatov, A., Timofte, R., Zhang, Z., Liu, M., Wang, H., Zuo, W., Zhang, J., Zhang, R., Peng, Z., Ren, S., et al.: Aim 2020 challenge on learned image signal processing pipeline. arXiv preprint arXiv:2011.04994 (2020)
33. Ignatov, A., Timofte, R., et al.: Learned smartphone isp on mobile gpus with deep learning, mobile ai & aim 2022 challenge: Report. In: European Conference on Computer Vision (2022)
34. Ignatov, A., Timofte, R., et al.: Realistic bokeh effect rendering on mobile gpus, mobile ai & aim 2022 challenge: Report (2022)
35. Ignatov, A., Van Gool, L., Timofte, R.: Replacing mobile camera isp with a single deep learning model. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 536–537 (2020)
36. Ignatov, D., Ignatov, A.: Controlling information capacity of binary neural network. Pattern Recognition Letters 138, 276–281 (2020)
37. Isobe, T., Zhu, F., Jia, X., Wang, S.: Revisiting temporal modeling for video super-resolution. arXiv preprint arXiv:2008.05765 (2020)
38. Jacob, B., Kligys, S., Chen, B., Zhu, M., Tang, M., Howard, A., Adam, H., Kalenichenko, D.: Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2704–2713 (2018)
39. Jain, S.R., Gural, A., Wu, M., Dick, C.H.: Trained quantization thresholds for accurate and efficient fixed-point inference of deep neural networks. arXiv preprint arXiv:1903.08066 (2019)
40. Kappeler, A., Yoo, S., Dai, Q., Katsaggelos, A.K.: Video super-resolution with convolutional neural networks. IEEE Transactions on Computational Imaging 2(2), 109–122 (2016)
41. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1646–1654 (2016)
42. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
43. Kınlı, F.O., Menteş, S., Özcan, B., Kirac, F., Timofte, R., et al.: Aim 2022 challenge on instagram filter removal: Methods and results. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2022)
44. Lee, Y.L., Tsung, P.K., Wu, M.: Technology trend of edge ai. In: 2018 International Symposium on VLSI Design, Automation and Test (VLSI-DAT). pp. 1–2. IEEE (2018)
45. Li, Y., Gu, S., Gool, L.V., Timofte, R.: Learning filter basis for convolutional neural network compression. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 5623–5632 (2019)
46. Li, Y., Zhang, K., Timofte, R., Van Gool, L., Kong, F., Li, M., Liu, S., Du, Z., Liu, D., Zhou, C., et al.: Ntire 2022 challenge on efficient super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1062–1102 (2022)
47. Lian, W., Lian, W.: Sliding window recurrent network for efficient video super-resolution. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2022)
48. Lian, W., Peng, S.: Kernel-aware raw burst blind super-resolution. arXiv preprint arXiv:2112.07315 (2021)
49. Liang, J., Cao, J., Fan, Y., Zhang, K., Ranjan, R., Li, Y., Timofte, R., Van Gool, L.: Vrt: A video restoration transformer. arXiv preprint arXiv:2201.12288 (2022)
50. Liu, H., Ruan, Z., Zhao, P., Dong, C., Shang, F., Liu, Y., Yang, L., Timofte, R.: Video super-resolution based on deep learning: a comprehensive survey. Artificial Intelligence Review pp. 1–55 (2022)
51. Liu, J., Tang, J., Wu, G.: Residual feature distillation network for lightweight image super-resolution. In: European Conference on Computer Vision. pp. 41–55. Springer (2020)

52. Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, K.T., Sun, J.: Metapruning: Meta learning for automatic neural network channel pruning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3296–3305 (2019)

53. Liu, Z., Wu, B., Luo, W., Yang, X., Liu, W., Cheng, K.T.: Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In: Proceedings of the European conference on computer vision (ECCV). pp. 722–737 (2018)

54. Lugmayr, A., Danelljan, M., Timofte, R.: Ntire 2020 challenge on real-world image super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 494–495 (2020)

55. Luo, Z., Li, Y., Cheng, S., Yu, L., Wu, Q., Wen, Z., Fan, H., Sun, J., Liu, S.: Bsr: Improving burst super-resolution with swin transformer and flow-guided deformable alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 998–1008 (2022)

56. Luo, Z., Yu, L., Mo, X., Li, Y., Jia, L., Fan, H., Sun, J., Liu, S.: Ebsr: Feature enhanced burst super-resolution with deformable alignment. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 471–478 (2021)

57. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Mu Lee, K.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

58. Nah, S., Son, S., Timofte, R., Lee, K.M.: Ntire 2020 challenge on image and video deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 416–417 (2020)

59. Nah, S., Timofte, R., Gu, S., Baik, S., Hong, S., Moon, G., Son, S., Mu Lee, K.: Ntire 2019 challenge on video super-resolution: Methods and results. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)

60. Obukhov, A., Rakhuba, M., Georgoulis, S., Kanakis, M., Dai, D., Van Gool, L.: T-basis: a compact representation for neural networks. In: International Conference on Machine Learning. pp. 7392–7404. PMLR (2020)

61. Romero, A., Ignatov, A., Kim, H., Timofte, R.: Real-time video super-resolution on smartphones with deep learning, mobile ai 2021 challenge: Report. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2021)

62. Sajjadi, M.S., Venulapalli, R., Brown, M.: Frame-recurrent video super-resolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6626–6634 (2018)

63. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: 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. pp. 1874–1883 (2016)

64. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2820–2828 (2019)
65. TensorFlow-Lite: [https://www.tensorflow.org/lite](https://www.tensorflow.org/lite)
66. Timofte, R., Gu, S., Wu, J., Van Gool, L.: Ntire 2018 challenge on single image super-resolution: Methods and results. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. pp. 852–863 (2018)
67. Uhlich, S., Mauch, L., Cardinaux, F., Yoshiyama, K., Garcia, J.A., Tiedemann, S., Kemp, T., Nakamura, A.: Mixed precision dnns: All you need is a good parametrization. arXiv preprint arXiv:1905.11452 (2019)
68. Wan, A., Dai, X., Zhang, P., He, Z., Tian, Y., Xie, S., Wu, B., Yu, M., Xu, T., Chen, K., et al.: Fnnetv2: Differentiable neural architecture search for spatial and channel dimensions. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12965–12974 (2020)
69. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
70. Wu, B., Dai, X., Zhang, P., Wang, Y., Sun, F., Wu, Y., Tian, Y., Vajda, P., Jia, Y., Keutzer, K.: Fnnet: Hardware-aware efficient convnet design via differentiable neural architecture search. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10734–10742 (2019)
71. Yang, J., Shen, X., Xing, J., Tian, X., Li, H., Deng, B., Huang, J., Hua, X.s.: Quantization networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7308–7316 (2019)
72. Yang, R., Timofte, R., et al.: Aim 2022 challenge on super-resolution of compressed image and video: Dataset, methods and results. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2022)
73. Yue, S., Li, C., Zhuge, Z., Song, R.: Eesnet: A network for energy efficient super-resolution. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2022)
74. Zhang, X., Zeng, H., Zhang, L.: Edge-oriented convolution block for real-time super resolution on mobile devices. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 4034–4043 (2021)