NTIRE 2024 Efficient SR Challenge Factsheet
-Lightening Partial Feature Distillation Network for Efficient Super-Resolution-

Yan Wang
Nankai University
Tianjin, China
wyrmy@foxmail.com

Yi Liu
ByteDance
Shenzhen, China
liuyi.chester@bytedance.com

Qing Wang
ByteDance
Shenzhen, China
wangqing.keen@bytedance.com

Gang Zhang
ByteDance
Shenzhen, China
zhanggang.zzg@bytedance.com

Liou Zhang
ByteDance
Shenzhen, China
zhangliou@bytedance.com

Shijie Zhao
ByteDance
Shenzhen, China
zhaoshijie.0526@bytedance.com

1. Team details

- Team name: BSR
- Team leader name: Yan Wang[1]
- Team leader address: Tianjin, China;
  phone: (+86) 13086676376;
  email: wyrmy@foxmail.com
  cc-email: wangy@nbjl.nankai.edu.cn
- Rest of the team members: Yi Liu[2], Qing Wang[2], Gang Zhang[2], Liou Zhang[2], Shijie Zhao[2]
- Team website URL (if any)
- Affiliation: [1]Nankai University, [2]ByteDance
- Affiliation of the team and/or team members with NTIRE 2024 sponsors (check the workshop website): Not Applicable
- User names: icandle (26.90/27.00)
- Best scoring entries **Validation: 26.90** (not the latest model, submitted by entry #3 on 02/16/2024), **Test: 27.00** (submitted by entry #3 on 03/20/2024)
- https://github.com/icandle/BSR

2. Method details

Based on PFDN [3], we propose PFDNLite.

PFDNLite: Inspired by ABPN [2] and PFDN [3], the PFDNLite (Fig. 1) 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.

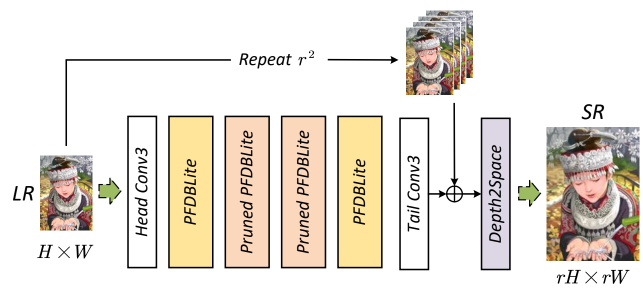

Figure 1. PFDNLite Architecture.

PFDBLite: Based on PFDN [3], we modify a more lightweight partial feature distillation block, dubbed PFDBLite (Fig. 2) to chase faster feature extraction. Generally, we execute two modifications focusing on the reparameterizable convolution and attention module. For the convolution block, we employ RepMBConv, which squeezes the MobileNetv3 block into vanilla convolution for better trade-offs between performance and memory access. Moreover, we add a reparameterizable point-wise convolution to cooperate with the middle RepMBConv as an approximation of partial convolution [1]. For the attention module, we propose a local attention (LocalAttn), which applies a local gate and MaxPool-based importance map to modulate input features. As illustrated in Fig. 2, we provide the details of RepMBConv and LocalAttn.

Pruned-PFDBLite: The Pruned-PFDBLite (Fig. 3) is similar to PFDBLite but dropping the second RepMBConv of PFDBLite and decreasing the output channels of the first RepMBConv from 48 to 24. Besides, we add the point-wise convolution after the first RepMBConv.

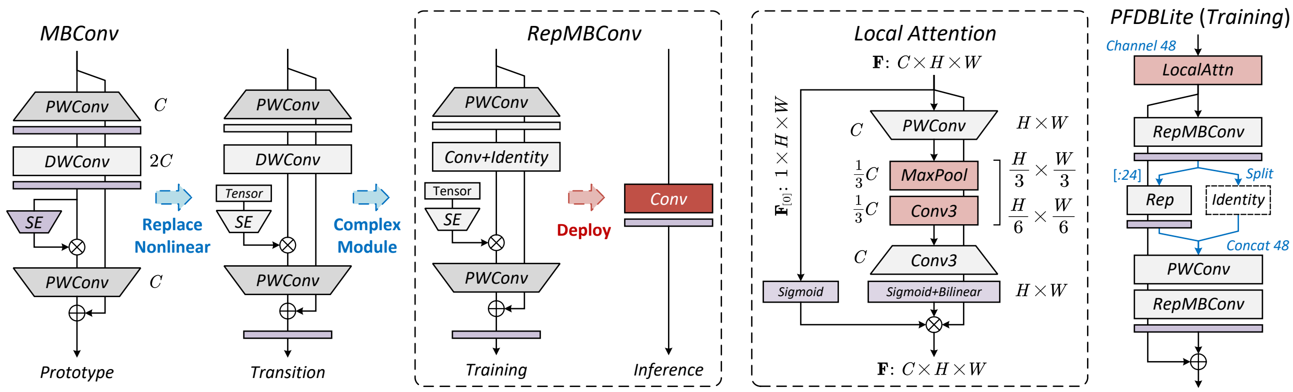

Figure 2. PFDBLite Block.

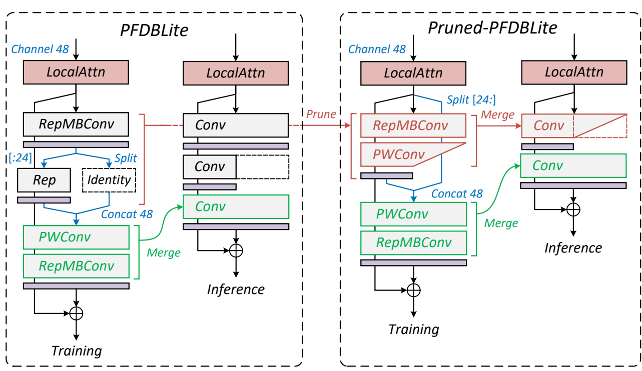

Figure 3. Pruned PFDBLite Block.

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

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

- Step1. HR patches of size 256x256 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 256x256 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 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.

Dataset	RLFN	DIPNet	Ours
LD-valid	26.96	27.00	26.90
LD-test	27.06	26.95	27.00
Test2K	26.20	26.11	26.18
Urban100	24.59	24.25	24.48

Table 1. Performance

II. At the second stage, we 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.

Model complexity: Overall, the parameter number of the PFDNLite is 0.218 M and the FLOPs is 11.95G.

Model performance: Tab. 1 exhibits the PSNR performance of our PFDBLite under four benchmarks, *i.e*., LD-valid/test, Test2K, and Urban100. The results demonstrate the robustness of the proposed framework.

References

[1] 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, 2023. 1

[2] Zongcai Du, Jie Liu, Jie Tang, and Gangshan Wu. Anchor-based plain net for mobile image super-resolution, 2021. 1

[3] Yan Wang. Edge-enhanced feature distillation network for efficient super-resolution, 2022. 1