NTIRE 2023 Efficient SR Challenge Factsheet
-Partial Feature Distillation Network for Efficient Super-Resolution -

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

Erlin Pan
UESTC
Chengdu, China
wujisixsix6@gmail.com

Qixuan Cai
Tianjin University
Tianjin, China
caiqixuan524@163.com

Xinan Dai
Tianjin University
Tianjin, China
d1577707716@163.com

1. Team details

- KaiBai Group
- Yan Wang[1]
- Addr: Nankai University, Tianjin, China,
  Tel: (+86) 13086676376,
  Email: wyrmy@foxmail.com
- Erlin Pan[2], Qixuan Cai[3], Xinan Dai[3]
- https://github.com/KaiBaiGroup
- [1]Nankai University
  [2]University of Electronic Science and Technology of China
  [3]Tianjin University
- User name: icandle (28.95/27.01).
- Best scoring entries of the team during development/validation phase: (28.95/27.01) for entries #2 on 03/16/23.
- Code has been released at https://github.com/icandle/PFDN.

2. Method details

Network Architecture: The network architecture continues the design of EFDN [4] but removes the skipped connection. The plain design decreases the model's depth and complexity. In Fig. 1, we exhibit the overview of PFDN.

Reparameterizable Convolution: The reparameterization technique plays a significant role in improving the performance of lightweight CNN-based methods. In PFDN, we combine the existing RRRB [1] and a layer-wise loss based on normalized cross-correlation to enhance the layer-wise representation.

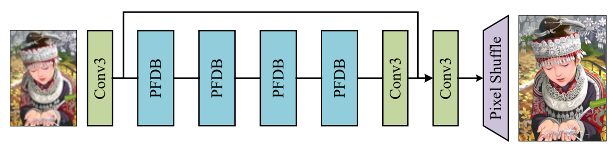

Figure 1. Network architecture of the proposed PFDN.

Partial Feature Distillation: Driven by [3], we design a partial local feature distillation block, dubbed PFDB. The main idea is that the intermediate features share high similarities among different channels [2]. This allows us to process partial features in the middle layer, which can reduce the parameters and FLOPs as well as memory access.

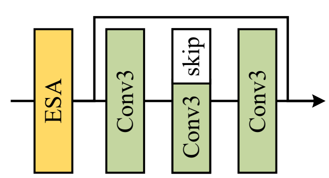

Figure 2. Partial Feature Distillation Block.

Implementation details: To obtain the LR-HR image pairs, we leverage bicubic interpolation to downscale the 2K resolution images from DIV2K and Flickr2K. We augment the training datasets by horizontal flips and 90° rotations. The HR path size and mini-batch size are determined by the training step. The training procedure can be summarized as follows.

1) Training from scratch. The LR patch size is set to 64×64, and the mini-batch size is 96. $\mathcal{L}_1$ loss and Adam optimizer are utilized in optimization. The learning rate is initialized as 5×10^{-4} and halved at $\{$250k, 400k, 450k, 475k$\}$. The total number of iterations is 500k.

2) Repeat training with larger patches. The LR patch size is sequentially set to $\{128\times128, 160\times160, 180\times180\}$, and the initial learning rate is 2×10^{-4}. We reparameterize the model before the $\{180\times180\}$ step.

3) Fine-tuning. The LR patch size and mini-batch size are 240×240 and 128, respectively. The $\mathcal{L}_2$ loss is chosen to promote PSNR value. The learning rate is 1×10^{-5}.

The proposed method is implemented under the PyTorch framework with 4 NVIDIA RTX 3090 GPUs.

References

[1] Zongcai Du, Ding Liu, Jie Liu, Jie Tang, Gangshan Wu, and Lean Fu. Fast and memory-efficient network towards efficient image super-resolution. In *CVPRW*, pages 853–862, 2022. 1

[2] Kai Han, Yunhe Wang, Qi Tian, Jianyuan Guo, Chunjing Xu, and Chang Xu. Ghostnet: More features from cheap operations. In *CVPR*, pages 1580–1589, 2020. 1

[3] 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 *CVPR*, pages 766–776, 2022. 1

[4] Yan Wang. Edge-enhanced feature distillation network for efficient super-resolution. In *CVPRW*, pages 777–785, 2022. 1