NTIRE 2023 Efficient SR Challenge Factsheet
PRFDN: High Parallelism Distillation Network For Image Super-resolution

Daheng Yin, Baijun Chen, Mengyang Liu
School of Computer Science and Engineering, Southeast University
Nanjing, Jiangsu
`yindaheng98, bjchen, myliu@seu.edu.cn`

1. Introduction

This factsheet template is meant to structure the description of the contributions made by each participating team in the NTIRE 2023 challenge on efficient image super-resolution.

Ideally, all the aspects enumerated below should be addressed. The provided information, the codes/executables and the achieved performance on the testing data are used to decide the awardees of the NTIRE 2023 challenge.

Reproducibility is a must and needs to be checked for the final test results in order to qualify for the NTIRE awards.

The main winners will be decided based on overall performance and a number of awards will go to novel, interesting solutions and to solutions that stand up as the best in a particular subcategory the judging committee will decided. Please check the competition webpage and forums for more details.

The winners, the awardees and the top ranking teams will be invited to co-author the NTIRE 2023 challenge report and to submit papers with their solutions to the NTIRE 2023 workshop. Detailed descriptions are much appreciated.

The factsheet, source codes/executables, trained models should be sent to **all of the NTIRE 2023 challenge organizers (Yawei Li, Yulun Zhang, and Radu Timofte)** by email.

2. Email final submission guide

To: yawei.li@vision.ee.ethz.ch
yulun100@gmail.com
timofte.radu@gmail.com
cc: your_team_members
Title: NTIRE 2023 Efficient SR Challenge - TEAM_NAME - TEAM_ID

To get your TEAM_ID, please register at Google Sheet. Please fill in your Team Name, Contact Person, and Contact Email in the first empty row from the top of sheet. Body contents should include:

a) team name

b) team leader's name and email address

c) rest of the team members

d) user names on NTIRE 2023 CodaLab competitions

e) Code, pretrained model, and factsheet download command, e.g. `git clone ...`, `wget ...`

f) Result download command, e.g. `wget ...`

- Please provide different urls in e) and f)

Factsheet must be a compiled pdf file together with a zip with .tex factsheet source files. Please provide a detailed explanation.

3. Code Submission

The code and trained models should be organized according to the GitHub repository. This code repository provides the basis to compare the various methods in the challenge. **Code scripts based on other repositories will not be accepted.** Specifically, you should follow the steps below.

1. Git clone the repository.
2. Put your model script under the `models` folder. Name your model script as `[Your_Team_ID]_[Your_Model_Name].py`.
3. Put your pretrained model under the `model_zoo` folder. Name your model checkpoint as `[Your_Team_ID]_[Your_Model_Name].[pth or pt or ckpt]`
4. Modify `model_path` in `test_demo.py`. Modify the imported models.
5. `python test_demo.py`

Please send us the command to download your code, e.g. `git clone [Your repository link]` When submitting the code, please remove the LR and SR images in `data` folder to save the bandwidth.

4. Factsheet Information

The factsheet should contain the following information. Most importantly, you should describe your method in detail. The training strategy (optimization method, learning rate schedule, and other parameters such as batch size, and patch size) and training data (information about the additional trainning data) should also be explained in detail.

4.1. Team details

- Team name: SEU_CNII
- Team leader name: Daheng Yin
- Team leader address, phone number, and email:
School of Computer Science and Engineering, Southeast University;
+86 188 5189 9135; yindaheng98@seu.edu.cn
- Rest of the team members: Baijun Chen, Mengyang Liu
- Team website URL (if any): N/A
- Affiliation: School of Computer Science and Engineering, Southeast University
- User names and entries on the NTIRE 2023 Codalab competitions (development/validation and testing phases): yindaheng98
- Best scoring entries of the team during development/validation phase:
PSNR 28.993
- Link to the codes/executables of the solution(s): https://github.com/yindaheng98/NTIRE23-RTSR

4.2. Method details

We proposed Parallel RFDN (PRFDN) as is shown in figure 1. Our method consists of four stages to transform a pre-trained RFDN [2] into PRFDN.

1. Branching. To accelerate the inference, we first consider reducing the data dependency in the model to achieve higher parallelism. Our method disentangles the sequentially computed trunks into branches. As is shown in figure 1b, after the branching, the major part of the model will consist of four independent branches that can calculate in parallel. To improve the accuracy, we also design a small SR block (SRFDB) based on the RFDB of RFDN and add them before the input of each branch.

(a) RFDN

(b) Branching

(c) Re-parametrization

(d) Pruning

Figure 1. Transform a pre-trained RFDN into PRFDN

2. Training. Since we only change the data flow but not the structure of RFDB, the pre-trained RFDN parameters can still be loaded into the major part of our branch model (only except for those SRFDBs). To benefit from the pre-training, we load the pre-trained RFDN parameters into our branch model before training our branch model.

3. Re-parametrization. Without much data dependency, branches in our model can be computed in parallel. However, a single GPU cannot compute two or more different models in parallel. To address this issue and achieve parallel computing of multiple branches on a single GPU, we should merge the four branches into a single branch. As is shown in figure 1c, we realize that the major part of these four branches (RFDBs and SRFDBs) have exactly the same structure but different parameters, so we merge and re-parametrized the RFDBs and SRFDBs into a single branch. More specifically, we create a bigger RFDB and a bigger SRFDB with bigger convolution layers and move the weight and bias from the RFDBs and SRFDBs into them. After re-parametrization, we get a bigger model that is equivalent to that of the branches of the branch model computed in parallel on a single GPU.

4. Pruning. The re-parametrized model is big. To further accelerate the inference, we applied channel pruning on our re-parametrized model, as is shown in figure 1d. To accurately extract the channel dependency for pruning, we first replace each concatenate and split operation with an equivalent convolution operation. After replacement, we used Torch-Pruning [1] to prune the model and finetune the model between each pruning step.

5. Technical details

- Language: Python
- Framework: Pytorch, Torch-Pruning [1]

- Optimizer: AdaM
- Learning rate
  Before re-parametrization: 1e-5 in general, if the loss cannot decrease after a lot of epochs, then descrease to 5e-6
  After re-parametrization: 1e-6
- GPU: RTX 3070 8G
- Datasets for training: LSDIR and DIV2K

6. Other details

- Planned submission of a solution(s) description paper at NTIRE 2023 workshop. YES
- General comments and impressions of the NTIRE 2023 challenge.

  Positive impression: Scripts for running the models and measuring metrics are well organized.

  Negative impression: Instructions are not clear (e.g. there are some posts in the forum saying that they cannot find the test set, but none of the organizers respond).

  A small suggestion: Create a group in instant messaging apps (e.g. Discord or Telegram) for convenient communication.
- What do you expect from a new challenge in image restoration, enhancement and manipulation? Real-time video super-resolution; Efficient compressed video restoration.
- Other comments: encountered difficulties, fairness of the challenge, proposed subcategories, proposed evaluation method(s), etc. N/A

References

[1] Gongfan Fang, Xinyin Ma, Mingli Song, Michael Bi Mi, and Xinchao Wang. Depgraph: Towards any structural pruning. *The Thirty-Fourth IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023. 2

[2] 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. 2