NTIRE 2023 Real-Time Super-Resolution

SEU_CNII
Daheng Yin, Baijun Chen, Mengyang Liu*

March 19, 2023

This factsheet template is meant to structure the description of the contributions made by each participating team in the NTIRE 2023 Real-Time Super-Resolution Challenge. The provided information, the codes/executables and the achieved performance on the testing data are used to decide the awardees of the NTIRE 2023 Real-Time Super-Resolution Challenge.

1. The models will be benchmarked as indicated in `https://github.com/eduardzamfir/NTIRE23-RTSR#evaluation-of-your-submission`.
2. The test set is similar to the validation set, including only 4K native images. Reproducibility is a must. **We will only run and benchmark the models, we will not release or open-source them.** Open sourcing the solution is a must to receive the prizes, up to the team.
3. The winners, the awardees and the top ranking teams will be invited to co-author the NTIRE 2023 Real-Time Super-Resolution Challenge report and to submit papers with their solutions to the NTIRE 2023 workshop. Detailed descriptions are much appreciated: brief description of all models and experiments tested, ablations, visualizations, things that did not work, datasets, pre-trained models, etc. You can share qualitative results via shared folder at google drive, dropbox, etc.

If you participate in both Track 1 and 2, you only need to submit 1 factsheet. This factsheet, and the source codes/models/executables (as specified here) must be sent to **all of the NTIRE 2023 Real-Time Super-Resolution Challenge organizers** by email. We accept download links from your website, google drive, dropbox, etc. When using cloud services, please remember to allow sharing and consider the traffic limit of your service provider. If there is any problem, we will contact the team leader.

*School of Computer Science and Engineering, Southeast University

Organizer name	Email
Marcos Conde	marcos.conde@uni-wuerzburg.de
Eduard Zamfir	eduard-sebastian.zamfir@uni-wuerzburg.de
Radu Timofte	radu.timofte@uni-wuerzburg.de

Email final submission guide

```
 To:  marcos.conde@uni-wuerzburg.de,
eduard-sebastian.zamfir@uni-wuerzburg.de,
radu.timofte@uni-wuerzburg.de
cc:  your_team_members
Title:  NTIRE 2023 Real-Time Super-Resolution Challenge - TEAM_NAME
```

Body contents should include:

a) the challenge name (AND TRACKS): NTIRE 2023 Real-Time Super-Resolution Challenge

b) team name

c) team leader's name and email address

d) rest of the team members

e) team name and user names on CodaLab competitions

f) executable/models/source code attached or download links. We will run and benchmark the models following `https://github.com/eduardzamfir/NTIRE23-RTSR#evaluation-of-your-submission`.

g) factsheet attached. You can also provide a link to **overleaf**. Only we, the organizers, will be able to read it. Factsheet must be a compiled pdf file together with a zip with .tex factsheet source files (including figures with good resolution).

1 Team details

- Team name: SEU_CNII
- Team leader name: Daheng Yin
- Team leader institution and email: School of Computer Science and Engineering, Southeast University; yindaheng98@seu.edu.cn
- Rest of the team members: Baijun Chen, Mengyang Liu
- Team website URL (if any): N/A
- Affiliations: School of Computer Science and Engineering, Southeast University
- Usernames on the NTIRE 2023 Real-Time Super-Resolution Codalab leaderboard (development/validation and testing phases): yindaheng98

- Link to the codes/executables of the solution(s) following `https://github.com/yindaheng98/NTIRE23-RTSR`

2 Contribution details

- Title of the contribution
  PRFDN: High Parallelism Distillation Network For Image Super-resolution
- General method description (Summary)
  We proposed Parallel RFDN (PRFDN) based on the pre-trained RFDN[2] as is shown in figure 1. Our method disentangles the sequentially computed trunks in RFDN into branches (figure 1b) and performs re-parametrization to make these branches inference in parallel on single devices. After that, we further perform pruning on the model (figure 1d) and finetune it to achieve higher performance.
- References: RFDN[2], Torch-Pruning[1]
- have you tested previously published methods? (yes/no) If yes, please specify which methods and the results/problems you found. N/A
- Other methods and baselines tested (even if results were not top competitive). N/A

3 Global Method Description

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.
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

(a) RFDN (b) Branching

(c) Re-parametrization (d) Pruning

Figure 1: Transform a pre-trained RFDN into PRFDN

Input	Training Time	Attention	Quantization	# Params. (M)	GPU
(256,256,3)	12h	No	Yes	5 Million	RTX 3070 8G

Table 1: FILL THIS TABLE PLEASE

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.

Please fill the following table specifying the technical information (besides writing it), should take 1 minute.

4 Technical details

- Language: Python
- Framework: Pytorch, Torch-Pruning[1]
- Optimizer: AdaM
- Learning rate: 1e-5 before re-parametrization, 1e-6 after re-parametrization
- GPU: RTX 3070 8G
- Datasets for training: LSDIR and DIV2K

Any particularities of the solution for this competition in comparison to other SR challenges (if applicable). N/A

5 Other details

- Planned submission of a solution(s) description paper at NTIRE 2023 workshop [YES/ NO]. YES
- General comments and impressions of the NTIRE 2023 Real-Time Super-Resolution Challenge (we appreciate your feedback to improve in future editions).

  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.

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] 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.