NTIRE 2025 Efficient SR Challenge Factsheet
-Espanded SPAN for Efficient Super-Resolution-

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

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

Yan Wang
ByteDance
Shanghai, China
wangyan.my@bytedance.com

Hongyu An
ByteDance
Beijing, China
anhongyu@bytedance.com

Liou Zhang
ByteDance
Beijing, China
zhangliou@bytedance.com

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

1. Team details

- Team name: MBGA
- Team leader name: Qing Wang[1]
- Team leader address: Shenzhen, China;
  phone: (+86) 18826077684;
  email: wangqing.keen@bytedance.com
- Rest of the team members: Yi Liu[2], Yang Wang[2], Hongyu An[2], Liou Zhang[2], Shijie Zhao[2]
- Team website URL (if any)
- Affiliation: [1]Bytedance
- Affiliation of the team and/or team members with NTIRE 2025 sponsors (check the workshop website): Not Applicable
- User names: MakeBicubicGreatAgain (26.90/26.99)
- Best scoring entries **Validation: 26.90** , **Test: 26.99**
- https://github.com/lookinmyeyestellmewhy/nt25_mbga

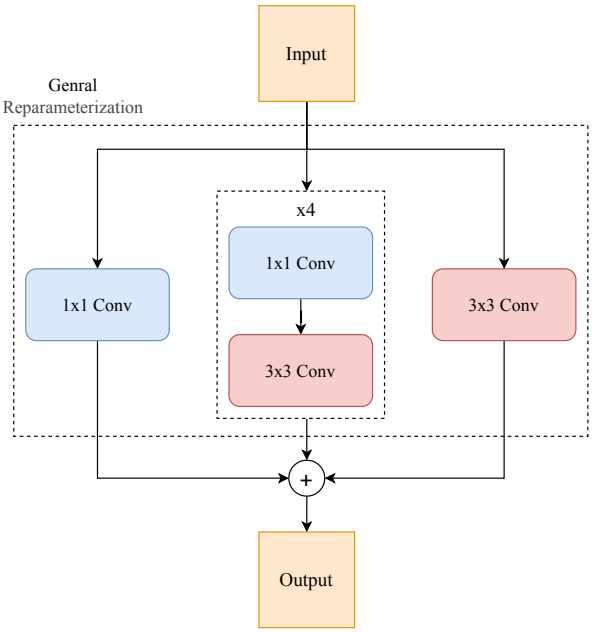

Figure 1. General Reparameterization.

2. Method details

Based on SPAN [4], we propose ESPAN.

Architecture: Through evaluations of depth-channel combinations in SPAN on an A6000 GPU, we determined that setting the number of channels to 32 yields higher efficiency than 28 channels. To reduce parameters and FLOPs, a depth of 6 was adopted. Additionally, a 9×9 convolution replaced the conventional 3×3 convolution at the network's input stage since we find that 9×9 convolution is faster than 3x3 convolution on A6000.

General Reparameterization: Inspired by MobileOne [3] and RepVGG [1], we propose a generalized reparameterization block(Fig. 1). The block consists of four 1×1-3×3 convolution branches, one 1×1 convolution branch, and one 3×3 convolution branch. Skip connections were omitted due to empirical observations of training in-

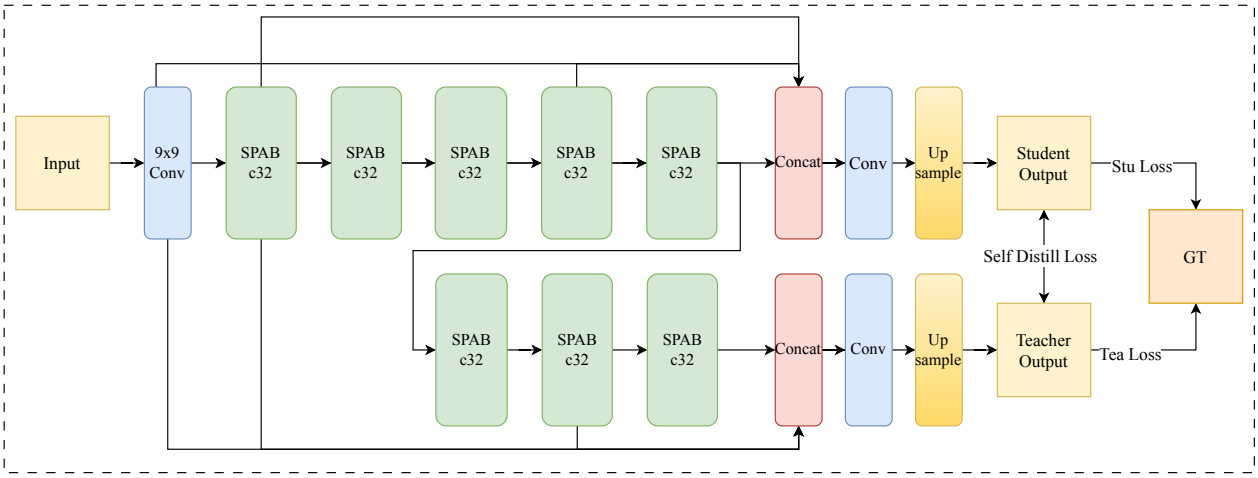

Figure 2. ESPAN with self distillation.

stability. While additional duplicated branches or 3×3-1×1 convolution branches are feasible, the current configuration was found to offer superior performance consistency during optimization.

Self distillation and progressive learning: Inspired by RIFE [2], self-distillation is incorporated into our training pipeline. The teacher model shares the identical backbone as the student model but includes three extra SPAB blocks appended to the student's backbone(Fig. 2). A self-distillation loss similar to RIFE's formulation is adopted to co-train the teacher and student networks. This design enables the teacher model to learn robust backbone features. After the distillation phase, the student loss and distillation loss components are removed, and the entire teacher model is fine-tuned. Leveraging the pre-trained robust teacher, progressive learning is employed: the extra SPAB blocks are gradually removed from the teacher's backbone, finally resulting in an architecture identical to the original student model.

Frequency-Aware Loss: Since small models have limited parameters, during training, we should make the model focus more on important (or difficult) areas. In our methods, two types of frequency-aware losses are employed. The first type is the DCT loss. We use the discrete cosine transform (DCT) to convert the RGB domain to the frequency domain and then apply the L1 loss to calculate the difference. The other type is the edge loss. We add a blur to the image and then subtract the blurred image from the original one to obtain the high frequency area. Subsequently, the L1 loss is calculated on this high frequency area.

Training details: The training process contains two stages. And the training dataset is the DIV2K_LSDIR_train. General reparameterization is used on the whole process.

I. At the first stage, we use self distillation to train the teacher model.

- Step1. We first train a 2x super-resolution model. HR patches of size 256x256 are randomly cropped from HR images, and the mini-batch size is set to 64. L1 loss and self distillation loss with AdamW optimizer are used and the initial learning rate is set to 0.0001 and halved at every 100k iterations. The total iterations is 500k. This step is repeated twice. And then we follow the same training setting and use 2x super-resolution model as pretrained model to train a 4x super-resolution model. This step is repeated twice.
- Step2. HR patches of size 512x512 are randomly cropped from HR images, and the mini-batch size is set to 16. MSE loss, frequency-aware loss and self distillation loss with AdamW optimizer are used and the initial learning rate is set to 0.0001 and halved at every 100k iterations. The total iterations is 500k. This step is also repeated twice.
- Step3. We only train the teacher model. HR patches of size 512x512 are randomly cropped from HR images, and the mini-batch size is set to 16. MSE loss and frequency-aware loss with AdamW optimizer are used and the initial learning rate is set to 0.00005 and halved at every 100k iterations. The total iterations is 500k. This step is also repeated twice.

II. At the second stage, we use progressive learning to get the final student model.

- Step4. We drop the additional SPAB block one by one. HR patches of size 512x512 are randomly cropped

from HR images, and the mini-batch size is set to 16. L1 loss with AdamW optimizer are used and the initial learning rate is set to 0.0001 and halved at every 100k iterations. The total iterations is 500k.

- Step5. We repeat the following training process many times until convergence. HR patches of size 512x512 are randomly cropped from HR images, and the mini-batch size is set to 16. MSE loss and frequency-aware loss with AdamW optimizer are used and the initial learning rate is set to 0.00005 and halved at every 100k iterations. The total iterations is 500k.

Model complexity: Overall, the parameter number of the ESPAN is 0.1923M and the FLOPs is 12.5556G.

Model performance: val: 26.90db, test: 26.99db

References

[1] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021. 1

[2] Zhewei Huang, Tianyuan Zhang, Wen Heng, Boxin Shi, and Shuchang Zhou. Real-time intermediate flow estimation for video frame interpolation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[3] Pavan Kumar Anasosalu Vasu, James Gabriel, Jeff Zhu, Oncel Tuzel, and Anurag Ranjan. An improved one millisecond mobile backbone. *arXiv preprint arXiv:2206.04040*, 2022. 1

[4] Cheng Wan, Hongyuan Yu, Zhiqi Li, Yihang Chen, Yajun Zou, Yuqing Liu, Xuanwu Yin, and Kunlong Zuo. Swift parameter-free attention network for efficient super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6246–6256, 2024. 1

3. Introduction

This factsheet template is meant to structure the description of the contributions made by each participating team in the NTIRE 2025 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 2025 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 2025 challenge report and to submit papers with their solutions to the NTIRE 2025 workshop. Detailed descriptions are much appreciated.

The factsheet, source codes/executables, trained models should be sent to **all of the NTIRE 2025 challenge organizers (Yawei Li, Bin Ren, Nancy Mehta, and Radu Timofte)** by email.

4. Email final submission guide

To:
yawei.li@vision.ee.ethz.ch
bin.ren@unitn.com
cshguo@gmail.com
zongwei.wu@uni-wuerzburg.de
timofte.radu@gmail.com

Title: NTIRE 2025 Efficient SR Challenge - MBGA - 036

Body contents should include:

a) MBGA

b) Qing Wang, wangqing.keen@bytedance.com

c) Yi Liu, Yan Wang, Hongyu An, Liou Zhang, Shijie Zhao

d) MakeBicubicGreatAgain

e) Code, pretrained model, and factsheet download command: git clone https://github.com/lookinmyeyestellmewhy/nt25_mbga

f) Result download command:

- `gdown https://drive.google.com/file/d/1E4gA_`

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

5. 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.

6. 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.