NTIRE 2025 Image Denoising ($\sigma = 50$) Challenge Factsheet
Bias-Tuning Enables Efficient Image Denoising

1. Introduction

This factsheet template is meant to structure the description of the contributions made by each participating team in the NTIRE 2025 challenge on image denoising with noise level $\sigma = 50$.

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 (Lei Sun, Yawei Li, and Radu Timofte)** by email.

2. Email final submission guide

To: lei.sun@insait.ai
cshguo@gmail.com
yawei.li@vision.ee.ethz.ch
Radu.Timofte@uni-wuerzburg.de
cc: your_team_members
Title: NTIRE 2025 Image Denoising 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
Alwaysu

b) team leader's name and email address
Jun Cheng, jcheng24@hust.edu.cn

c) rest of the team members
Shan Tan

d) user names on NTIRE 2025 CodaLab competitions
alwaysu

e) Code, pretrained model, and factsheet download command, e.g. `git clone ...`, `wget ...`
Code and factsheet download command:
git clone https://github.com/alwaysuu/Challenge_GaussDn50
pretrained model download command:
wget https://drive.google.com/file/d/1_JDH6XSiWiAhmjPP_rsTISJ9YT6PTgSZ/view?usp=sharing

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

- Please provide different urls in e) and f)

wget https://drive.google.com/file/d/1PwPRMqmCH4FP54Z345Tm7AArASqRWGvq/view?usp=sharing

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 noisy and denoised 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
  Alwaysu
- Team leader name
  Jun Cheng
- Team leader address, phone number, and email
  1037 Luo Yu Road, Wuhan, 430074 China
  (+86) 13554028461
  jcheng24@hust.edu.cn
- Rest of the team members
  Shan Tan
- Team website URL (if any)
  N/A
- Affiliation
  Huazhong University of Science and Technology
- Affiliation of the team and/or team members with NTIRE 2025 sponsors (check the workshop website)
  No
- User names and entries on the NTIRE 2025 Codalab competitions (development/validation and testing phases)
  alwaysu
- Best scoring entries of the team during development/validation phase
  PSNR/SSIM: 30.72/0.860
- Link to the codes/executables of the solution(s)
  See Section 2

4.2. Method details

Method: Our objective is to achieve efficient Gaussian denoising based on pre-trained denoisers. Our core idea, initially proposed in transfer learning [1], is freezing pre-trained denoisers and only fine-tuning *existing or newly added bias parameters* during adaptation, thus maintaining the knowledge of pre-trained models and reducing tuning cost.

We choose the Restormer [2] model trained to remove the same $i.i.d.$ Gaussian noise ($\sigma = 50$) *without intensity clipping* as our baseline. As this pre-trained Restormer did not clip noisy images' intensities into the normal range, i.e., [0, 255], it performs poorly in *clipped* noisy images, resulting in low PSNR/SSIM (27.47/0.79 on DIV2K validation) and clear artifacts. After embedding learnable bias parameters into this freezing Restormer (except LayerNorm modules) and fine-tuning the model, satisfactory denoising results can be obtained, and the resultant PSNR increases by over 3dB (evaluated on DIV2K validation set). During the inference, we further enhance the denoiser via self-ensemble [3] and *patch stitching*. When dealing with high-resolution (HR) noisy images, we process them via overlapping patches with the same patch size as the training phase. We stitch these overlapping denoised patches via linear blending, as introduced in image stitching [4].

Training details: We fine-tune this *bias-version* Restormer using the PSNR loss function and AdamW optimizer combined with batch size 2, patch size 256×256, learning rate $3e^{-4}$ (cosine annealed to $1e^{-6}$), $200k$ iterations and geometric augmentation. The training dataset consists of 800 images from DIV2K training set and 1,000 images from LSDIR training set.

Inference details: The patch size and overlapping size during patch stitching are 256×256 and 16, respectively.

Additional details:

- Total method complexity

  Total number of parameters: 26.25M; Total number of learnable bias parameters: 0.014M; FLOPs: 140.99G (evaluated on image with shape $256 \times 256 \times 3$)

- Which pre-trained or external methods / models have been used (for any stage, if any)

  The pre-trained Restormer model is from this link: https://drive.google.com/drive/folders/1Qwsjyny54RZWa7zC4Apg7exixLBo4uF0.
  We found that various pre-trained denoisers, including three noise-specific models and one noise-blind model, resulted in similar denoising performance on clipped noisy images after bias-tuning.

- Which additional data has been used in addition to the provided NTIRE training and validation data (at any

stage, if any)
Our bias-tuning did not use additional data except DIV2K and LSDIR. We also note that the pre-trained Restormer utilized a combined set of 800 images from DIV2K, 2,650 images of Flickr2K, 400 BSD500 images and 4,744 images from WED.

- Training description
  See above
- Testing description
  See above
- Quantitative and qualitative advantages of the proposed solution
  Bias-tuning is parameter-efficient.
- Results of the comparison to other approaches (if any)
  N/A
- Results on other benchmarks (if any)
  N/A
- Novelty degree of the solution and if it has been previously published
  Minor. We built upon existing works [1–4].
- It is OK if the proposed solution is based on other works (papers, reports, Internet sources (links), etc). It is ethically wrong and a misconduct if you are not properly giving credits and hide this information.
  We agree.

5. Other details

- Planned submission of a solution(s) description paper at NTIRE 2025 workshop.
  No
- General comments and impressions of the NTIRE 2025 challenge.
  Good
- What do you expect from a new challenge in image restoration, enhancement and manipulation?
  It's good enough
- Other comments: encountered difficulties, fairness of the challenge, proposed subcategories, proposed evaluation method(s), etc.
  No

References

[1] H. Cai, C. Gan, L. Zhu, and S. Han, "Tinytl: Reduce memory, not parameters for efficient on-device learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11285–11297, 2020. 2, 3

[2] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5728–5739, 2022. 2, 3

[3] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, "Enhanced deep residual networks for single image super-resolution," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 136–144, 2017. 2, 3

[4] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *International journal of computer vision*, vol. 74, pp. 59–73, 2007. 2, 3