The Tenth NTIRE 2025 Image Denoising Challenge Report

Lei Sun*   Hang Guo*   Bin Ren*   Luc Van Gool*   Radu Timofte*   Yawei Li*

Xiangyu Kong   Hyunhee Park   Xiaoxuan Yu   Suejin Han   Hakjae Jeon   Jia Li

Hyung-Ju Chun   Donghun Ryou   Inju Ha   Bohyung Han   Jingyu Ma

Zhijuan Huang   Huiyuan Fu   Hongyuan Yu   Boqi Zhang   Jiawei Shi   Heng Zhang

Huadong Ma   Deepak Kumar Tyagi   Aman Kukretti   Gajender Sharma

Sriharsha Koundinya   Asim Manna   Jun Cheng   Shan Tan   Jun Liu   Jiangwei Hao

Jianping Luo   Jie Lu   Satya Narayan Tazi   Arnim Gautam   Aditi Pawar

Aishwarya Joshi   Akshay Dudhane   Praful Hambarde   Sachin Chaudhary

Santosh Kumar Vipparthi   Subrahmanyam Murala   Jiachen Tu   Nikhil Akalwadi

Vijayalaxmi Ashok Aralikatti   Dheeraj Damodar Hegde   G Gyaneshwar Rao   Jatin Kalal

Chaitra Desai   Ramesh Ashok Tabib   Uma Mudenagudi   Zhenyuan Lin   Yubo Dong

Weikun Li   Anqi Li   Ang Gao   Weijun Yuan   Zhan Li   Ruting Deng

Yihang Chen   Yifan Deng   Zhanglu Chen   Boyang Yao   Shuling Zheng

Feng Zhang   Zhiheng Fu   Anas M. Ali   Bilel Benjdira   Wadii Boulila   JanSeny

Pei Zhou   Jianhua Hu   K. L. Eddie Law   Jaeho Lee   M. J. Aashik Rasool

Abdur Rehman   SMA Sharif   Seongwan Kim   Alexandru Brateanu   Raul Balmez

Ciprian Orhei   Cosmin Ancuti   Zeyu Xiao   Zhuoyuan Li   Ziqi Wang   Yanyan Wei

Fei Wang   Kun Li   Shengeng Tang   Yunkai Zhang   Weirun Zhou   Haoxuan Lu

Abstract

This paper presents an overview of the NTIRE 2025 Image Denoising Challenge ($\sigma = 50$), highlighting the proposed methodologies and corresponding results. The primary objective is to develop a network architecture capable of achieving high-quality denoising performance, quantitatively evaluated using PSNR, without constraints on computational complexity or model size. The task assumes independent additive white Gaussian noise (AWGN) with a fixed noise level of 50. A total of 290 participants registered for the challenge, with 20 teams successfully submitting valid results, providing insights into the current state-of-the-art in image denoising.

* L. Sun (lei.sun@insait.ai, INSAIT, Sofia University "St. Kliment Ohridski"), H. Guo, B. Ren (bin.ren@unitn.it, University of Pisa & University of Trento, Italy), L. Van Gool, R. Timofte, and Y. Li were the challenge organizers, while the other authors participated in the challenge. Appendix A contains the authors' teams and affiliations.
NTIRE 2025 webpage: https://cvlai.net/ntire/2025/.
Code: https://github.com/AHupuJR/NTIRE2025_Dn50_challenge.

1. Introduction

Image denoising is a fundamental problem in low-level vision, where the objective is to reconstruct a noise-free image from its degraded counterpart. During image acquisition and processing, various types of noise can be introduced, such as Gaussian noise, Poisson noise, and compression artifacts from formats like JPEG. The presence of these noise sources makes denoising a particularly challenging task. Given the importance of image denoising in applications such as computational photography, medical imaging, and remote sensing, continuous research efforts are necessary to develop more efficient and generalizable denoising solutions [20, 61].

To further advance research in this area, this challenge aims to promote the development of denoising methods. A widely used benchmark for fair performance evaluation is the additive white Gaussian noise (AWGN) model, which serves as the standard setting in this competition.

As part of the New Trends in Image Restoration and Enhancement (NTIRE) 2025 workshop, we organized the Image Denoising Challenge. The objective is to restore clean images from inputs corrupted by AWGN with a noise level of $\sigma = 50$. This competition seeks to foster innovative

solutions, establish performance benchmarks, and explore emerging trends in the design of image denoising networks, we hope the methods in this challenge will shed light on image denoising.

This challenge is one of the NTIRE 2025 Workshop associated challenges on: ambient lighting normalization [54], reflection removal in the wild [57], shadow removal [53], event-based image deblurring [48], image denoising [49], XGC quality assessment [37], UGC video enhancement [45], night photography rendering [18], image super-resolution (x4) [12], real-world face restoration [13], efficient super-resolution [44], HR depth estimation [58], efficient burst HDR and restoration [27], cross-domain few-shot object detection [19], short-form UGC video quality assessment and enhancement [29, 30], text to image generation model quality assessment [22], day and night raindrop removal for dual-focused images [28], video quality assessment for video conferencing [23], low light image enhancement [38], light field super-resolution [56], restore any image model (RAIM) in the wild [34], raw restoration and super-resolution [16] and raw reconstruction from RGB on smartphones [17].

2. NTIRE 2025 Image Denoising Challenge

The objectives of this challenge are threefold: (1) to stimulate advancements in image denoising research, (2) to enable a fair and comprehensive comparison of different denoising techniques, and (3) to create a collaborative environment where academic and industry professionals can exchange ideas and explore potential partnerships.

In the following sections, we provide a detailed overview of the challenge, including its dataset, evaluation criteria, challenge results, and the methodologies employed by participating teams. By establishing a standardized benchmark, this challenge aims to push the boundaries of current denoising approaches and foster innovation in the field.

2.1. Dataset

The widely used DIV2K [2] dataset and LSDIR [31] dataset are utilized for the challenge.

DIV2K dataset comprises 1,000 diverse RGB images at 2K resolution, partitioned into 800 images for training, 100 images for validation, and 100 images for testing.

LSDIR dataset consists of 86,991 high-resolution, high-quality images, with 84,991 images allocated for training, 1,000 images for validation, and 1,000 images for testing.

Participants were provided with training images from both the DIV2K and LSDIR datasets. During the validation phase, the 100 images from the DIV2K validation set were made accessible to them. In the test phase, evaluation was conducted using 100 images from the DIV2K test

set and an additional 100 images from the LSDIR test set. To ensure a fair assessment, the ground-truth noise-free images for the test phase remained hidden from participants throughout the challenge.

2.2. Tracks and Competition

The goal is to develop a network architecture that can generate high-quality denoising results, with performance evaluated based on the peak signal-to-noise ratio (PSNR) metric.

Challenge phases *(1) Development and validation phase*: Participants were provided with 800 clean training images and 100 clean/noisy image pairs from the DIV2K dataset, along with an additional 84,991 clean images from the LSDIR dataset. During the training process, noisy images were generated by adding Gaussian noise with a noise level of $\sigma = 50$. Participants had the opportunity to upload their denoising results to the CodaLab evaluation server, where the PSNR of the denoised images was computed, offering immediate feedback on their model's performance. *(2) Testing phase*: In the final test phase, participants were given access to 100 noisy test images from the DIV2K dataset and 100 noisy test images from the LSDIR dataset, while the corresponding clean ground-truth images remained concealed. Participants were required to submit their denoised images to the CodaLab evaluation server and send their code and factsheet to the organizers. The organizers then verified the submitted code and ran it to compute the final results, which were shared with participants at the conclusion of the challenge.

Evaluation protocol The primary objective of this challenge is to promote the development of accurate image denoising networks. Hence, PSNR and SSIM metrics are used for quantitative evaluation, based on the 200 test images. A code example for calculating these metrics can be found at https://github.com/AHupuJR/NTIRE2025_Dn50_challenge. Additionally, the code for the submitted solutions, along with the pre-trained weights, is also provided in this repository. Note that computational complexity and model size are not factored into the final ranking of the participants.

3. Challenge Results

Table 1 presents the final rankings and results of the participating teams. Detailed descriptions of each team's implementation are provided in Sec.4, while team member information can be found in Appendix A. SRC-B secured first place in terms of PSNR, achieving a 1.25 dB advantage over the second-best entry. SNUCV and BuptMM ranked second and third, respectively.

Team	Rank	PSNR (primary)	SSIM
SRC-B	1	31.20	0.8884
SNUCV	2	29.95	0.8676
BuptMM	3	29.89	0.8664
HMiDenoise	4	29.84	0.8653
Pixel Purifiers	5	29.83	0.8652
Alwaysu	6	29.80	0.8642
Tcler Denoising	7	29.78	0.8632
cipher_vision	8	29.64	0.8601
Sky-D	9	29.61	0.8602
KLETech-CEVI	10	29.60	0.8602
xd_denoise	11	29.58	0.8597
JNU620	12	29.55	0.8590
PSU team	12	29.55	0.8598
Aurora	14	29.51	0.8605
mpu_ai	15	29.30	0.8499
OptDenoiser	16	28.95	0.8422
AKDT	17	28.83	0.8374
X-L	18	26.85	0.7836
Whitehairbin	19	26.83	0.8010
mygo	20	24.92	0.6972

Table 1. Results of NTIRE 2025 Image Denoising Challenge. PSNR and SSIM scores are measured on the 200 test images from DIV2K test set and LSDIR test set. Team rankings are based primarily on PSNR.

3.1. Participants

This year, the challenge attracted 290 registered participants, with 20 teams successfully submitting valid results. Compared to the previous challenge [32], the SRC-B team's solution outperformed the top-ranked method from 2023 by 1.24 dB. Notably, the results achieved by the top six teams this year surpassed those of their counterparts from the previous edition, establishing a new benchmark for image denoising.

3.2. Main Ideas and Architectures

During the challenge, participants implemented a range of novel techniques to enhance image denoising performance. Below, we highlight some of the fundamental strategies adopted by the leading teams.

1. **Hybrid architecture performs well.** All the models from the top-3 teams adopted a hybrid architecture that combines transformer-based and convolutional-based network. Both Global features from the transformer and local features from the convolutional network are useful for image denoising. SNUCV further adopted the model ensemble to push the limit.

2. **Data is important.** This year's winning team, SRC-B adopted a data selection process to mitigate the influence of data imbalance, and also select high-quality images in

the dataset for training instead of training on the whole DIV2K and LSDIR dataset.

3. **The devil is in the details.** Wavelet Transform loss [25] is utilized by the winning team, which is proven to help the model escape from local optima. Tricks such as a progressive learning strategy also work well. A higher percentage of overlapping of the patches during inference also leads to higher PSNR. Ensemble techniques effectively improve the performance.

4. **New Mamba-based Design.** SNUCV, the second-ranking team, leveraged the MambaIRv2 framework to design a hybrid architecture, combining the efficient sequence modeling capabilities from Mamba with image restoration objectives.

5. **Self-ensemble or model ensembling** is adopted to improve the performance by some of the teams.

3.3. Fairness

To uphold the fairness of the image denoising challenge, several rules were established, primarily regarding the datasets used for training. First, participants were allowed to use additional external datasets, such as Flickr2K, for training. However, training on the DIV2K validation set, including either high-resolution (HR) or low-resolution (LR) images, was strictly prohibited, as this set was designated for evaluating the generalization ability of the models. Similarly, training with the LR images from the DIV2K test set was not permitted. Lastly, employing advanced data augmentation techniques during training was considered acceptable and within the scope of fair competition.

4. Challenge Methods and Teams

4.1. Samsung MX (Mobile eXperience) Business & Samsung R&D Institute China - Beijing (SRC-B)

4.1.1. Model Framework

The proposed solution is shown in figure 1. In recent years, the Transformer structure has shown excellent performance in image denoising tasks due to its advantages in capturing global context.

However, it is found that pure Transformer architectures are relatively weak in recovering local features and details. On the other hand, CNN-based methods excel in detail recovery but struggle to effectively capture global context information. Therefore, they designed a network that combines the strengths of the transformer network Restormer [59] and the convolutional network NAFnet [10]. They first extract global features using the Transformer network and then enhance detail information using the convolutional network. The denoising network's structure follows Restormer, while the detail enhancement network draws inspiration from NAFNet. Finally, they dynamically fuse the
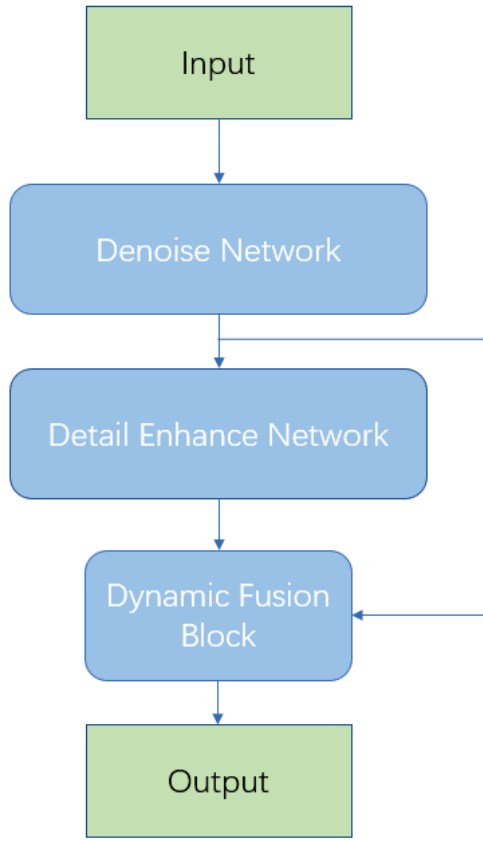
Figure 1. Framework of the hybrid network proposed by Team SRC-B.

two features from transformer network and convolutional network through a set of learnable parameters to balance denoising and detail preservation like in , thereby improving the overall performance of image denoising.

4.1.2. Dataset and Training Strategy

Dataset. Three datasets are used in total: the DIV2K dataset, the LSDIR dataset, and a self-collected custom dataset consisting of 2 million images. The specific ways in which they utilized these training sets across different training phases will be detailed in the training details section. In the final fine-tuning phase, they construct a high quality dataset consist of 1000 images from LSDIR, 1000 images from the custom dataset and all 800 images from DIV2K. The data selection process including:

- Image resolution: Keep only images with a resolution greater than 900x900.
- Image quality: Keep only images that rank in the top 30% for all three metrics: Laplacian Var, BRISQUE, and NIQE.
- Semantic selection: To achieve semantic balance, they

conducted a semantic selection based on Clip [43] features to ensure that the dataset reflects diverse and representative content across various scene categories.

Training. The model training consists of three stages. In the first stage, they pre-train the entire network using a custom dataset of 2 million images, with an initial learning rate of $1e^{-4}$ and a training time of approximately 360 hours. In the second stage, they fine-tune the detail enhancement network module using the DIV2K and LSDIR datasets, with an initial learning rate of $1e^{-5}$ and a training duration of about 240 hours, which enhanced the model's ability to restore details. In the third stage, they select 1,000 images from the custom dataset, 1,000 images from the LSDIR data, and 800 images from DIV2K as the training set. With an initial learning rate of $1e^{-6}$, they fine-tuned the entire network for approximately 120 hours.

The model is trained by alternately iterating L1 loss, L2 loss, and Stationary Wavelet Transform(SWT) loss[25]. They found that adding SWT loss during training helps the model escape from local optima. They also perform progressive learning where the network is trained on different image patch sizes gradually enlarged from 256 to 448 and 768. As the patch size increases, the performance can gradually improve. The model was trained on an A100 80G GPU.

4.2. SNUCV

Method. As shown in Figure 2, the network architecture they utilized consists of MambaIRv2 [21], Xformer [60], and Restormer [59]. These networks were first trained on Gaussian noise with a standard deviation of 50. Subsequently, the outputs of these networks are concatenated with the noisy image, which is then used as input to the ensemble model. In addition to the output, the features from the deepest layers of these networks are also concatenated and integrated into the deepest layer features of the ensemble network. This approach ensures that the feature information from the previous networks is preserved and effectively transferred to the ensemble network without loss. The ensemble model is designed based on Xformer, accepting an input with 12 channels. Its deepest layer is structured to incorporate the concatenated features of the previous models. These concatenated features are then processed through a 1×1 convolution to reduce the channel dimension back to that of the original network, thus alleviating the computational burden. Additionally, while Xformer and Restormer reduce the feature size in their deep layer, MambaIRv2 retains its original feature size without reduction. To align the sizes for concatenation, the features of MambaIRv2 were downscaled by a factor of 8 before being concatenated.

Training details. They first train the denoising networks, and then we incorporate the frozen denoising networks to train the ensemble model. Both the denoising
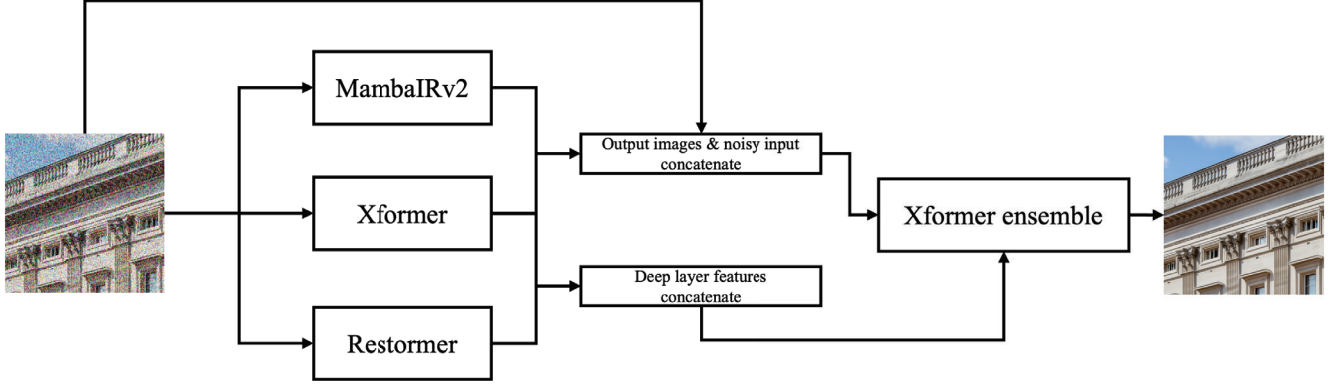
Figure 2. The overview of the deep ensemble pipeline proposed by Team SNUCV.

models and the ensemble model were trained exclusively using the DIV2K [2] and LSDIR [31] datasets. Training was performed using the AdamW [39] optimizer with hyperparameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a learning rate of 3×10^{-4}. All models were trained for a total of 300,000 iterations. For denoising models, Restormer and Xformer were trained using a progressive training strategy to enhance robustness and efficiency. Patch sizes were progressively increased as $[128, 160, 192, 256, 320, 384]$, with corresponding batch sizes of $[8, 5, 4, 2, 1, 1]$. In contrast, MambaIRv2 was trained with a more constrained setup due to GPU memory limitations, utilizing patch sizes of $[128, 160]$ and batch sizes of $[2, 1]$. The ensemble model was trained with a progressive patch size schedule of $[160, 192, 256, 320, 384, 448]$ and corresponding batch sizes of $[8, 5, 4, 2, 1, 1]$. The denoising models were trained using L1 loss, while the ensemble model was trained using a combination of L1 loss, MSE loss, and high frequency loss.

Inference details. During the final inference stage to derive test results, they utilized a self-ensemble technique. Furthermore, inference was conducted using a patch-based sliding-window approach. Patch sizes were set at [256, 384, 512], with corresponding overlap values of [48, 64, 96]. The resulting outputs were subsequently averaged to optimize performance. This self-ensemble approach, while significantly increasing computational cost, substantially enhances performance.

4.3. BuptMM

Description. In recent years, the Transformer architecture has been widely used in image denoising tasks. In order to further explore the superiority of the two representative networks, Restormer [59] and HAT [11], they propose a dual network & post-processing denoising model that combines the advantages of the former's global attention mechanism and the latter's channel attention mechanism.

As shown in Fig. 3, our network is divided into two

stages. In the first stage, they use DIV2K [2] and LSDIR [31] training sets to train Restormer [59] and HAT [11] respectively, and then enhance the ability of Restormer [59] through TLC [36] technology during its reasoning stage. In the second stage, they first use the Canny operator to perform edge detection on the images processed by the two models. They take an OR operation on the two edge images, and then XOR the result with the edge of HAT to obtain the edge difference between the two images. For this part of the edge difference, they use the result obtained by HAT [11] as the standard for preservation. Finally, they take the average of the other pixels of HAT [11] and Restormer [59] to obtain the final result.

They used the DIV2K [2] and LSDIR [31] datasets to train both the Restormer [59] and HAT [11] simultaneously. They employed a progressive training strategy for the Restormer [59] with a total of 292000 iterations, where the image block size increased from 128 to 384 with a step size of 64. They also used progressive training strategy for the HAT [11], where the image block size increased from 64 to 224. They did not use any other datasets besides the datasets mentioned above during the process. During the training phase, they spent one day separately training the Reformer [59] and HAT [11], they trained two models using 8 NVIDIA H100 GPUs. They conducted the inference process on the H20 test set, with a memory usage of 15G. The average inference time for a single image from the 200 test sets was 4.4 seconds, while the average time for morphological post-processing was within 1 second.

4.4. HMiDenoise

The network is inspired by the HAT [11] model architecture, and the architecture is optimized for the task specifically. The optimized denoising network structure(D-HAT) is shown in Fig 4.

The dataset utilized for training comprises DIV2K and LSDIR. To accelerate training and achieve good performance, they initially train on a small scale (64x64) with
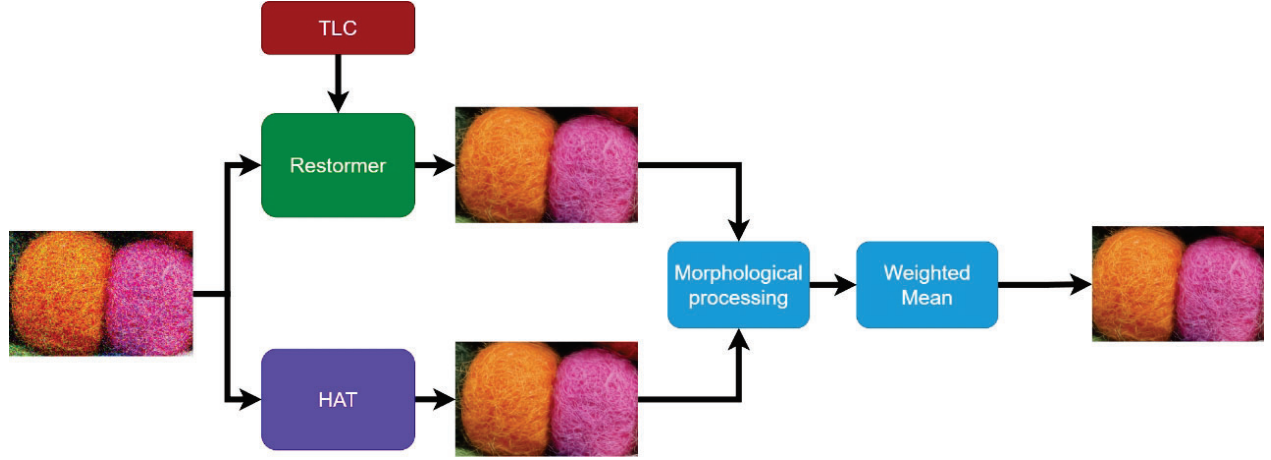
Figure 3. The model architecture of DDU proposed by Team BuptMM.
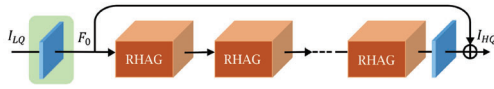

Figure 4. Model architecture of DB-HAT proposed by Team HMi-Denoise.

batch size 16, then on a medium scale (128x128) with batch size 1, and finally optimize on a larger scale (224x224) with batch size 1. As the patch size increases, the performance can gradually improve. The learning rate is initialized at 4 $\times 10^{-4}$ and decays according to the cosine annealing strategy during the training. The network undergoes training for a total of $2{\times}10^5$ iterations, with the L2 loss function being minimized through the utilization of the Adam optimizer. Subsequently, fine-tuning is executed using the L2 loss and SSIM loss functions, with an initial learning rate of 5×10^{-5} for $2{\times}10^5$ iterations. They repeated the aforementioned fine-tune settings two times after loading the trained weights. All experiments are conducted with the PyTorch 2.0 framework on 8 H100 GPUs.

4.5. Pixel Purifiers

Architecture. Restormer architecture [59], as shown in Fig. 5(a), is an efficient transformer and it uses the multi-Dconv head transposed attention block (MDTA) for channel attention and the gated Dconv feedforward network (GDFN) for the feedforward network. MDTA block applies self-attention across channels rather than the spatial dimension to compute cross-covariance across channels to generate an attention map encoding the global context implicitly. Additionally, depth-wise convolutions are used to emphasize on the local context before computing feature covariance to produce the global attention map. GDFN block introduces a novel gating mechanism and depth-wise con-

volutions to encode information from spatially neighboring pixel positions, useful for learning local image structure for effective restoration.

Training Techniques. They have conducted extensive experiments to evaluate the effectiveness of our approach (as shown in Fig. 5(b)). The network is trained using the DIV2K and LSDIR datasets only with L1 loss function. To enhance generalization and mitigate overfitting, they apply randomized data augmentation during training, including horizontal flipping, vertical flipping, and rotations of $90°$, $180°$, and $270°$. A fixed patch size of 256×256 is maintained for both training and inference to preserve global context. For optimization, they used the AdamW optimizer in conjunction with the CosineAnnealingRestartCyclicLR scheduler, with an initial learning rate 1×10^{-4}. Training is done using 8 NVIDIA Tesla V100 GPUs. Additionally, they leveraged Hard Dataset Mining for model fine-tuning, specifically targeting training patches where the loss exceeded a predefined threshold. This technique, discussed in detail in the following section, further enhanced the performance of our baseline model.

Hard Dataset Mining. To further enhance PSNR, they employed a hard dataset mining technique inspired by [3] for fine-tuning. Specifically, training patches with loss value exceeding a predefined threshold is selected for transfer learning on our base trained model. To preserve the model's generalization while refining its performance on challenging samples, they applied a learning rate that was 100 times smaller than the initial training rate.

DIV2K and LSDIR Datasets Ratio. As the model is to be trained and tested on two datasets (DIV2K and LSDIR), they first analysed their characteristics. DIV2K is relatively small and generalised with 800 training images while LSDIR is significantly large dataset with 84k+ training images, primarily consisting of high texture images. Consid-
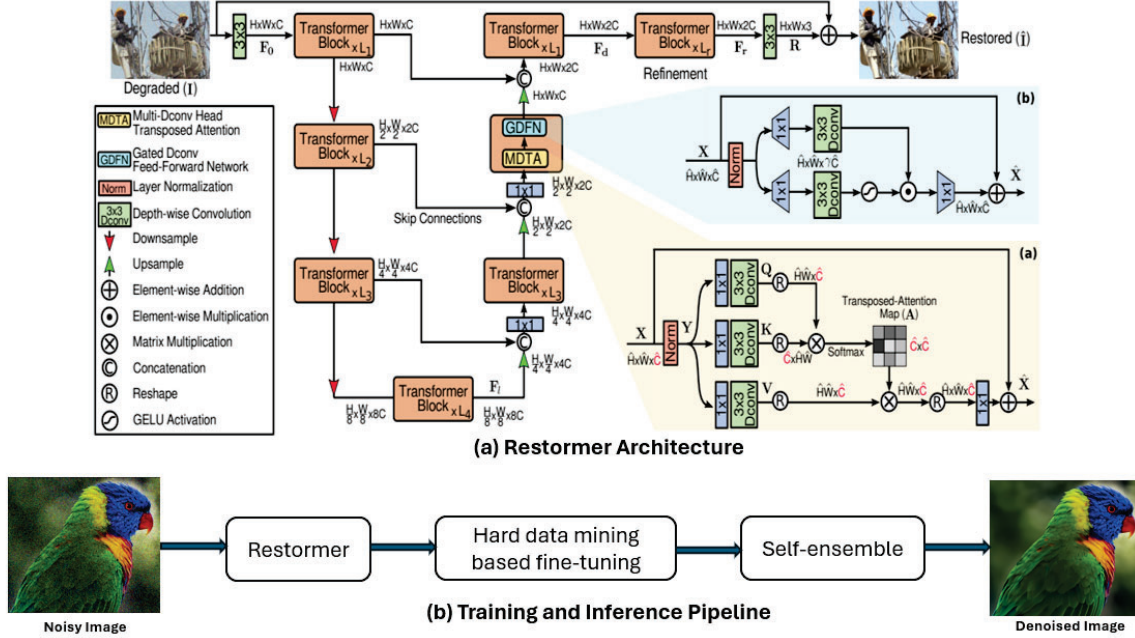
Figure 5. Block Diagram for Image Denoising using Restormer architecture along with Hard data mining and Ensemble Techniques (Team Pixel Purifiers).

ering the dataset charatectertics and our dataset ratio experiments, they found that DIV2K to LSDIR ratio of 12:88 during training helps to improve overall PSNR and generalise the model better for both validation and test datasets.

Overlapping Percentage During Inference. Using a small overlap of 5% during inference with a patch size of 256×256 (same as the training patch size to preserve global context) resulted in improved inference speed. However, despite applying boundary pixel averaging, minor stitching artifacts is observed, leading to a decline in PSNR performance. To mitigate these artifacts, they increased the overlap to 20% with original 256×256 patch size, which resulted in PSNR improvement.

Ensemble Technique at Inference. Ensemble techniques played a crucial role by effectively boosting performance. They used the Self Ensemble Strategy, specifically test-time augmentation ensemble [35] where multiple flips and rotations of images were used before model inference. The model outputs are averaged to generate the final output image.

4.6. Alwaysu

Method: Our objective is to achieve efficient Gaussian denoising based on pre-trained denoisers. Our core idea, termed *Bias-Tuning*, initially proposed in transfer learning [8], 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.

They choose the Restormer [59] 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). They found that various pre-trained Gaussian denoisers from [59], including three noise-specific models and one noise-blind model, resulted in similar denoising performance on clipped noisy images after Bias-Tuning.

During the inference, they further enhance the denoiser via self-ensemble [35] and *patch stitching*. When dealing with high-resolution (HR) noisy images, they process them via overlapping patches with the same patch size as the training phase. They stitch these overlapping denoised patches via linear blending, as introduced in image stitching [7].

Training details: They 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. They 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.

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

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

4.7. Tcler_Denosing

Building upon the work of Potlapalli et al. [42], they propose a novel transformer-based architecture for image restoration, termed PromptIR-Dn50. This architecture adopts a U-shaped encoder-decoder network structure, incorporating progressive downsampling and upsampling operations. Specifically tailored for denoising tasks under additive white Gaussian noise (AWGN) with a noise level of sigma=50, PromptIR-Dn50 leverages the strengths of the PromptGenBlock with targeted modifications. In this framework, the PromptGenBlock is adapted by explicitly incorporating sigma=50 as an input parameter, ensuring the model is optimized for the specific noise level and achieves superior performance in denoising tasks.

Inspired by the advancements in MambaIRv2 [21], they further introduce a specialized variant, MambaIRv2-Dn50, designed for image restoration tasks. This architecture also adopts a U-shaped encoder-decoder structure but integrates two key innovations: the Attentive State-space Equation (ASE) and Semantic Guided Neighboring (SGN) modules. These components address the causal scanning limitations inherent in traditional Mamba frameworks while maintaining linear computational complexity. Unlike prior approaches that rely on multi-directional scanning, MambaIRv2-Dn50 achieves non-causal global perception through single-sequence processing, making it highly efficient and well-suited for vision tasks.

To further enhance the performance of image restoration, they propose a fusion strategy that combines the strengths of PromptIR-Dn50 and MambaIRv2-Dn50. By integrating the outputs of these two architectures, the fused model leverages the noise-specific optimization of PromptIR-Dn50 and the global perception capabilities of MambaIRv2-Dn50. This hybrid approach ensures robust and high-quality restoration results, effectively addressing the challenges posed by sigma=50 AWGN noise.

The architecture follows a progressive training strategy as in Restormer [59], where input resolutions gradually increase from 64×64 to 112×112. This progressive learning scheme enhances feature adaptation across scales without compromising training stability.

For optimization, they employ the Adam optimizer with an initial learning rate of 1e-4, combined with a CosineAn-
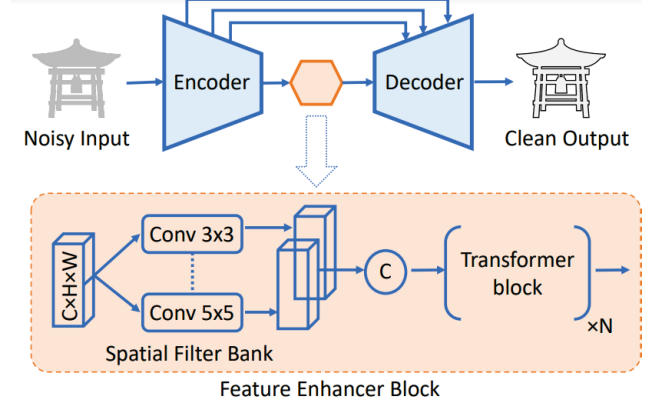

Figure 6. Proposed Pureformer encoder-decoder architecture for image denoising proposed by Team cipher vision. The input noisy image is processed through a multi-level encoder, a feature enhancer block, and a multi-level decoder. Each encoder and decoder level employs *xN* transformer blocks [62], consisting of Multi-Dconv Head Transposed Attention (MDTA) and Gated-Dconv Feed-Forward Network (GDFN) blocks. The feature enhancer block, placed in the latent space, expands the receptive field using a spatial filter bank. The multi-scale features are then concatenated and refined through *xN* transformer blocks to enhance feature correlation and merge multi-scale information effectively.

nealingRestartCyclicLR schedule to adjust the learning rate dynamically during training. The model is trained using a combination of Charbonnier loss and Gradient-weighted L1 loss, which effectively balances pixel-wise accuracy and edge preservation. The weights for those two losses are 0.8 and 0.2, respectively. They use the DIV2K [2] and LSDIR [31] datasets exclusively during the training phase, where horizontally and vertically flipping, rotation, USM sharpen [55] are used to augment the input images of our model.

During the testing phase, the input size is fixed at 112×112, and self-ensemble techniques [50] are applied to further enhance the model's performance. This approach ensures robust denoising results and improved generalization to unseen data.

In summary, MambaIRv2-Dn50 introduces a tailored state-space model-based architecture for denoising tasks, leveraging progressive learning, advanced loss functions, and self-ensemble techniques to achieve state-of-the-art performance on sigma=50 AWGN noise.

4.8. cipher_vision

As shown in Figure 6, they employ a Transformer-based encoder-decoder architecture featuring a four-level encoder-decoder structure designed to restore images degraded by Gaussian noise ($\sigma = 50$). This architecture is optimized to capture both local and global features, significantly enhancing the quality of input images. The hierarchical structure of the model includes four levels, containing

[4, 6, 6, 8] Transformer blocks respectively. Each Transformer block includes Multi-Dconv Head Transposed Attention (MDTA) followed by a Gated-Dconv feed-forward network (GDFN), enabling the model to capture long-range feature dependencies effectively. Additionally, skip connections are utilized to link the encoder and decoder, preserving spatial details and ensuring efficient feature reuse throughout the network. The feature enhancer block in the latent space processes latent features through the filter bank, and extracted multi-scale features are concatenated and passed through the transformer blocks as shown in Figure 6.

Training Details Our training strategy uses the datasets DIV2K (1000) and LSDIR (86,991). They leverage small patch-based training and data augmentation techniques to optimize the Pureformer. The training process uses the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a learning schedule that includes a linear warmup for 15 epochs followed by cosine annealing. The batch size is set to 4, consisting of $4 \times 3 \times 128 \times 128$ patches, and training is conducted on 2×A100 GPUs. Data augmentation techniques such as random cropping, flips, 90° rotations, and mixup are applied. They use L1 Loss to optimize the parameters.

Testing Strategy For inference, they use the datasets DIV2K (100) and LSDIR (100). Testing is performed using 512×512 patches. To enhance robustness, they employ self-ensemble testing with rotational transformations. The input image is rotated by 0°, 90°, 180°, and 270°, processed through the trained model, and rotated back to its original orientation. The final prediction is obtained by averaging the outputs of all four rotations.

4.9. A Two-Stage Denoising Framework with Generalized Denoising Score Matching Pretraining and Supervised Fine-tuning (Sky-D)

Problem Formulation In natural image denoising, we aim to recover a clean image $\mathbf{X}_0 \in \mathbb{R}^d$ from its noisy observation $\mathbf{X}_{t_{\text{data}}} \in \mathbb{R}^d$. The noisy observation can be modeled as:

$$\mathbf{X}_{t_{\text{data}}} = \mathbf{X}_0 + \sigma_{t_{\text{data}}}\mathbf{N}, \tag{1}$$

where $\sigma_{t_{\text{data}}} > 0$ denotes the noise standard deviation at level t_{data}, and $\mathbf{N} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$ represents the noise component.

Our approach consists of two stages: (1) self-supervised pretraining using Generalized Denoising Score Matching (GDSM) and (2) supervised fine-tuning. This two-stage approach enables us to leverage both noisy data and clean labels effectively.

4.9.1. Self-Supervised Pretraining with Generalized Denoising Score Matching

For the pretraining stage, we adopt the Generalized Denoising Score Matching (GDSM) framework introduced in Corruption2Self (C2S) [51]. This approach enables effective

learning directly from noisy observations without requiring clean labels.

Forward Corruption Process Following [51], we define a forward corruption process that systematically adds additional Gaussian noise to $\mathbf{X}_{t_{\text{data}}}$:

$$\mathbf{X}_t = \mathbf{X}_{t_{\text{data}}} + \sqrt{\sigma_t^2 - \sigma_{t_{\text{data}}}^2}\, \mathbf{Z},$$
$$\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d), \quad t > t_{\text{data}}, \tag{2}$$

where σ_t is a monotonically increasing noise schedule function for $t \in (t_{\text{data}}, T]$, with T being the maximum noise level.

Generalized Denoising Score Matching Loss The GDSM loss function [51] is formulated as:

$$J(\theta) = \mathbb{E}_{\mathbf{X}_{t_{\text{data}}},t,\mathbf{X}_t}\Big[\big\|\gamma(t, \sigma_{t_{\text{target}}})\, \mathbf{h}_\theta(\mathbf{X}_t, t)$$
$$+\delta(t, \sigma_{t_{\text{target}}})\, \mathbf{X}_t - \mathbf{X}_{t_{\text{data}}}\big\|^2\Big], \tag{3}$$

where t is sampled uniformly from $(t_{\text{data}}, T]$ and the coefficients are defined by:

$$\gamma(t, \sigma_{t_{\text{target}}}) := \frac{\sigma_t^2 - \sigma_{t_{\text{data}}}^2}{\sigma_t^2 - \sigma_{t_{\text{target}}}^2}$$
$$\delta(t, \sigma_{t_{\text{target}}}) := \frac{\sigma_{t_{\text{data}}}^2 - \sigma_{t_{\text{target}}}^2}{\sigma_t^2 - \sigma_{t_{\text{target}}}^2}. \tag{4}$$

The parameter $\sigma_{t_{\text{target}}}$ controls the target noise level, with $\sigma_{t_{\text{target}}} = 0$ representing maximum denoising (complete noise removal).

Reparameterization for Improved Training Stability To enhance training stability and improve convergence, we employ the reparameterization strategy proposed in [51]. Let $\tau \in (0, T']$ be a new variable defined by:

$$\sigma_\tau^2 = \sigma_t^2 - \sigma_{t_{\text{data}}}^2,$$
$$T' = \sqrt{\sigma_T^2 - \sigma_{t_{\text{data}}}^2}. \tag{5}$$

The original t can be recovered via:

$$t = \sigma_t^{-1}\left(\sqrt{\sigma_\tau^2 + \sigma_{t_{\text{data}}}^2}\right). \tag{6}$$

Under this reparameterization, the loss function becomes:

$$J'(\theta) = \mathbb{E}_{\mathbf{X}_{t_{\text{data}}},\tau,\mathbf{X}_t}\Big[\big\|\gamma'(\tau, \sigma_{t_{\text{target}}})\, \mathbf{h}_\theta(\mathbf{X}_t, t)$$
$$+\delta'(\tau, \sigma_{t_{\text{target}}})\, \mathbf{X}_t - \mathbf{X}_{t_{\text{data}}}\big\|^2\Big], \tag{7}$$

where the coefficients are:

$$\gamma'(\tau, \sigma_{t_{\text{target}}}) = \frac{\sigma_\tau^2}{\sigma_\tau^2 + \sigma_{t_{\text{data}}}^2 - \sigma_{t_{\text{target}}}^2},$$
$$\delta'(\tau, \sigma_{t_{\text{target}}}) = \frac{\sigma_{t_{\text{data}}}^2 - \sigma_{t_{\text{target}}}^2}{\sigma_\tau^2 + \sigma_{t_{\text{data}}}^2 - \sigma_{t_{\text{target}}}^2}. \quad (8)$$

This reparameterization ensures uniform sampling over τ and consistent coverage of the noise level range during training, leading to smoother and faster convergence.

4.9.2. Supervised Fine-tuning

After pretraining with GDSM, we propose to fine-tune the model with a supervised approach. Unlike traditional methods that train from scratch using clean labels, our approach leverages the knowledge gained during pretraining to enhance performance.

Supervised Fine-tuning Loss Given paired training data $\{(\mathbf{X}_{t_{\text{data}}}^i, \mathbf{Y}^i)\}_{i=1}^N$, where $\mathbf{X}_{t_{\text{data}}}^i$ is the noisy observation and $\mathbf{Y}^i$ is the corresponding clean target, we formulate the supervised fine-tuning loss as:

$$\mathcal{L}_{\text{sup}}(\theta) = \frac{1}{N} \sum_{i=1}^N \left\| \mathbf{h}_\theta(\mathbf{X}_{t_{\text{data}}}^i, t_{\text{data}}) - \mathbf{Y}^i \right\|^2. \quad (9)$$

This formulation directly optimizes the network to map noisy observations to clean targets. By initializing θ with the pretrained weights from the GDSM stage, we enable more effective and stable fine-tuning.

4.9.3. Time-Conditioned Diffusion Model Architecture

Our approach employs the same time-conditioned diffusion model architecture used in [51], which is based on the U-Net architecture enhanced with time conditioning and the Noise Variance Conditioned Multi-Head Self-Attention (NVC-MSA) module. The model's denoising function $\mathbf{h}_\theta : \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ maps a noisy input $\mathbf{X}_t$ and noise level t to an estimate of the clean image $\mathbf{X}_0$.

The time conditioning is implemented through an embedding layer that transforms the noise level t into a high-dimensional feature vector, which is then integrated into the convolutional layers via adaptive instance normalization. This enables the model to dynamically adjust its denoising behavior based on the noise level of the input.

The NVC-MSA module extends standard self-attention by conditioning the attention mechanism on the noise variance, allowing the model to adapt its attention patterns based on the noise characteristics of the input. This adaptation enhances the model's ability to denoise effectively across different noise levels and patterns.

Algorithm 1: Two-Stage Training Procedure for GDSM Pretraining and Supervised Fine-tuning

---

Require: Training data from DIV2K and LSDIR, max noise level T, learning rates α_1, α_2
Ensure: Trained denoising model $\mathbf{h}_\theta$
1: **// Phase 1: Self-supervised Pretraining with GDSM**
2: Initialize network parameters θ randomly
3: **repeat**
4:     Sample minibatch $\{\mathbf{X}_{t_{\text{data}}}^i\}_{i=1}^m$ from DIV2K and LSDIR training sets
5:     Sample noise level $\tau \sim \mathcal{U}(0, T')$
6:     Sample Gaussian noise $\mathbf{Z} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}_d)$
7:     Compute $t = \sigma_t^{-1}\left(\sqrt{\sigma_\tau^2 + \sigma_{t_{\text{data}}}^2}\right)$
8:     Generate corrupted samples: $\mathbf{X}_t = \mathbf{X}_{t_{\text{data}}} + \sigma_\tau \mathbf{Z}$
9:     Compute coefficients $\gamma'(\tau, \sigma_{t_{\text{target}}})$ and $\delta'(\tau, \sigma_{t_{\text{target}}})$
10:     Compute GDSM loss $J'(\theta)$ according to Eq. (7)
11:     Update parameters: $\theta \leftarrow \theta - \alpha_1 \nabla_\theta J'(\theta)$
12: **until** convergence or maximum iterations reached
13: **// Phase 2: Supervised Fine-tuning**
14: Initialize network parameters θ with pretrained weights from Phase 1
15: **repeat**
16:     Sample paired minibatch $\{(\mathbf{X}_{t_{\text{data}}}^i, \mathbf{Y}^i)\}_{i=1}^m$ from DIV2K and LSDIR training sets
17:     Compute supervised loss:
    $\mathcal{L}_{\text{sup}}(\theta) = \frac{1}{m} \sum_{i=1}^m \|\mathbf{h}_\theta(\mathbf{X}_{t_{\text{data}}}^i, t_{\text{data}}) - \mathbf{Y}^i\|^2$
18:     Update parameters: $\theta \leftarrow \theta - \alpha_2 \nabla_\theta \mathcal{L}_{\text{sup}}(\theta)$ $\{\alpha_2 < \alpha_1$ for stable fine-tuning$\}$
19: **until** convergence or maximum iterations reached
20: **return** Trained model $\mathbf{h}_\theta$

4.9.4. Training Procedure

As outlined in Algorithm 1, our approach combines self-supervised pretraining with supervised fine-tuning to leverage the strengths of both paradigms. The GDSM pretraining phase enables the model to learn robust representations across diverse noise levels without clean labels, establishing a strong initialization for subsequent supervised learning. This knowledge transfer accelerates convergence during fine-tuning and enhances generalization to noise distributions not explicitly covered in the supervised data. The time-conditioned architecture further facilitates this adaptability by dynamically adjusting denoising behavior based on input noise characteristics. To our knowledge, this represents the first application of GDSM as a pretraining strategy for natural image denoising, offering a principled approach to combining self-supervised and supervised learning objectives for this task.

4.9.5. Implementation Details

We implement our two-stage training procedure with a progressive learning strategy similar to that proposed in [59], gradually increasing image patch sizes to capture multiscale features while maintaining computational efficiency.

Table 2. Progressive Training Schedule

Stage	Patch Size	Batch	Learning Rate
1	256^2	48	1×10^{-3}
2	384^2	24	3×10^{-4}
3	512^2	12	1×10^{-4}
4	Mixed*	4	5×10^{-5}

*Randomly selected from $\{512^2, 768^2, 896^2\}$ per batch

As detailed in Algorithm 1, each stage consists of both self-supervised pretraining and supervised fine-tuning phases.

For the GDSM pretraining, we set the maximum corruption level $T = 10$, which provides sufficient noise coverage while maintaining training stability. To determine the data noise level t_{data}, we incorporate standard noise estimation techniques from the `skimage` package [52]. While we could explicitly set t_{data} to correspond to specific noise levels (e.g., 50/255), we found that automated estimation suffices for good performance. In future work, more tailored approaches for specific noise level denoising could be implemented.

For optimization, we employ the AdamW optimizer with gradient clipping to stabilize training, coupled with a cosine annealing learning rate scheduler. Our progressive training schedule (see Table 2) gradually increases patch sizes while adjusting batch sizes and learning rates accordingly. We initialize each stage with weights from the previous stage, setting a maximum of 20 epochs per stage with early stopping based on validation performance. Due to computational time constraints, we note that the network training for the final stage of progressive learning had not yet fully converged when reporting our results.

This progressive approach allows the model to initially learn basic denoising patterns on smaller patches where more diverse samples can be processed in each batch, then gradually adapt to larger contextual information in later stages. We train our models using the DIV2K [2] and LS-DIR [31] training datasets, while validation is performed on their respective validation sets, which remain completely separate from training.

Throughout the entire training process, we maintain the same time-conditioned model architecture, leveraging its ability to handle varying noise levels both during self-supervised pretraining and supervised fine-tuning. The self-supervised pretraining with GDSM establishes robust initialization across diverse noise conditions, while the supervised fine-tuning further refines the model's performance on specific noise distributions of interest.

4.9.6. Inference Process

During standard inference, given a noisy observation $\mathbf{X}_{t_{\text{data}}}$, we obtain the denoised output directly from our trained model:

$$\hat{\mathbf{X}} = \mathbf{h}_{\theta^*}(\mathbf{X}_{t_{\text{data}}}, t_{\text{data}}), \tag{10}$$

However, to maximize denoising performance for high-resolution images without requiring additional model training, we incorporate two advanced techniques: geometric self-ensemble and adaptive patch-based processing.

Geometric Self-Ensemble Following [35], we implement geometric self-ensemble to enhance denoising quality by leveraging the model's equivariance properties. This technique applies a set of geometric transformations (rotations and flips) to the input image, processes each transformed version independently, and then averages the aligned outputs. The approach can be concisely formulated as:

$$\hat{\mathbf{X}}_{\text{GSE}} = \frac{1}{K} \sum_{i=1}^{K} T_i^{-1} \left(\mathbf{h}_{\theta^*} \left(T_i(\mathbf{X}_{t_{\text{data}}}), t_{\text{data}} \right) \right), \tag{11}$$

where $\{T_i\}_{i=1}^{K}$ represents a set of $K = 8$ geometric transformations (identity, horizontal flip, vertical flip, 90°, 180°, and 270° rotations, plus combinations), and T_i^{-1} denotes the corresponding inverse transformation. This approach effectively provides model ensembling benefits without requiring multiple models or additional training.

Adaptive Patch-Based Processing To handle high-resolution images efficiently, we implement an adaptive patch-based processing scheme that dynamically selects appropriate patch sizes based on input dimensions. Algorithm 2 details our complete inference procedure.

Our adaptive patch-based approach dynamically selects from three patch sizes (896×896, 768×768, or 512×512) based on input image dimensions. For each geometric transformation, the algorithm determines whether patch-based processing is necessary. If so, it divides the image into overlapping patches with 50% stride, processes each patch independently, and reconstructs the full image by averaging overlapping regions. This strategy effectively handles high-resolution images while maintaining computational efficiency.

4.10. KLETech-CEVI

Method: The proposed HNNformer method is based on the HNN framework [24], which includes three main modules: the hierarchical spatio-contextual (HSC) feature encoder, Global-Local Spatio-Contextual (GLSC) block, and hierarchical spatio-contextual (HSC) decoder, as shown in Figure 7. Typically, image denoising networks employ feature scaling for varying the sizes of the receptive fields. The varying receptive fields facilitate learning of local-to-global

Algorithm 2: Adaptive Geometric Self-Ensemble Inference

Require: Noisy image $\mathbf{X}_{t_{\text{data}}}$, model $\mathbf{h}_{\theta*}$
Ensure: Denoised image $\hat{\mathbf{X}}$
1: $\mathcal{T} \leftarrow$ {Identity, HFlip, VFlip, Rot90, ...}   ▷ 8 transforms
2: $H, W \leftarrow$ dimensions of $\mathbf{X}_{t_{\text{data}}}$
3: $t_{\text{data}} \leftarrow \begin{cases} \text{estimate\_noise}(\mathbf{X}_{t_{\text{data}}}) & \text{if auto mode} \\ \text{predefined level} & \text{otherwise} \end{cases}$
4: patch_size $\leftarrow \begin{cases} 896 & \text{if } \min(H, W) \geq 896 \\ 768 & \text{if } \min(H, W) \geq 768 \\ 512 & \text{if } \min(H, W) \geq 512 \end{cases}$
5: stride $\leftarrow$ patch_size/2   ▷ 50% overlap
6: outputs $\leftarrow \emptyset$
7: **for all** $T \in \mathcal{T}$ **do**
8:     $\mathbf{X}_T \leftarrow T(\mathbf{X}_{t_{\text{data}}})$
9:     $H_T, W_T \leftarrow$ dimensions of $\mathbf{X}_T$
10:     **if** $\max(H_T, W_T) >$ patch_size **then**
11:         $\text{output}_T, \text{count} \leftarrow \text{zeros}(H_T, W_T)$
12:         Pad $\mathbf{X}_T$ to dimensions divisible by stride
13:         **for** (i, j) in overlapping patch grid **do**
14:             patch $\leftarrow \mathbf{X}_T[i : i + \text{patch\_size}, j : j + \text{patch\_size}]$
15:             result $\leftarrow \mathbf{h}_{\theta*}(\text{patch}, t_{\text{data}})$
16:             Accumulate result and increment count at positions (i, j)
17:         **end for**
18:         $\text{denoised}_T \leftarrow \text{output}_T/\text{count}$
19:     **else**
20:         $\text{denoised}_T \leftarrow \mathbf{h}_{\theta*}(\mathbf{X}_T, t_{\text{data}})$
21:     **end if**
22:     outputs $\leftarrow$ outputs $\cup \{T^{-1}(\text{denoised}_T)\}$
23: **end for**
24: **return** $\hat{\mathbf{X}} \leftarrow \frac{1}{|\mathcal{T}|} \sum_{\text{out} \in \text{outputs}} \text{out}$

variances in the features. With this motivation, they learn contextual information from multi-scale features while preserving high-resolution spatial details. They achieve this via a hierarchical style encoder-decoder network with residual blocks as the backbone for learning. Given an input noisy image x, the proposed multi-scale hierarchical encoder extracts shallow features in three distinct scales and is given as:r

$$F_{si} = ME_s(x) \tag{12}$$

where F_{si} are the shallow features extracted at the ith scale from the sampled space of input noisy image x and ME_s represents the hierarchical encoder at scale s.

Inspired by [60], they propose Global-Local Spatio-Contextual (GLSC) Block, that uses Spatial Attention Blocks (SAB) to learn spatial features at each scale. They also employ a Channel Attention Block (CAB) to fuse the multi-level features. The learned deep features are represented as:

$$D_{si} = GLSC_{si}(F_{si}) \tag{13}$$

where D_{si} is the deep feature at the ith scale, F_{si} are the spatial features extracted at the ith scale, and $GLSC_{si}$ represents Spatial Attention Blocks (SAB) at respective scales.

They decode the deep features obtained at various scales with the proposed hierarchical decoder, given by:

$$d_{si} = MD_{si}(D_{si}) \tag{14}$$

where D_{si} is the deep feature at the ith scale, d_{si} is the decoded feature at the ith scale, and MD_{si} represents the hierarchical decoder. The decoded features and upscaled features at each scale are passed to the reconstruction layers M_r to obtain the denoised image $\hat{y}$. The upscaled features from each scale are stacked and represented as:

$$P = d_{s1} + d_{s2} + d_{s3} \tag{15}$$

where d_{s1}, d_{s2}, and d_{s3} are decoded features at three distinct scales, and P represents the final set of features passed to the Channel Attention Block (CAB) to obtain the denoised image $\hat{y}$.

$$\hat{y} = M_r(P) \tag{16}$$

where $\hat{y}$ is the denoised image obtained from reconstruction layers M_r. They optimize the learning of HNNFormer with the proposed $L_{HNNformer}$, given as:

$$L_{HNNformer} = (\alpha \cdot L_1) + (\beta \cdot L_{VGG}) + (\gamma \cdot L_{MSSSIM}) \tag{17}$$

where α, β, and γ are the weights. They experimentally set the weights to $\alpha = 0.5$, $\beta = 0.7$, and $\gamma = 0.5$. L_{HNN} is a weighted combination of three distinct losses: L_1 loss to minimize error at the pixel level, perceptual loss to efficiently restore contextual information between the ground-truth image and the output denoised image, and multiscale structural dissimilarity loss to restore structural details. The aim here is to minimize the weighted combinational loss L_{HNN} given as:

$$L(\theta) = \frac{1}{N} \sum_{i=1}^{N} \|HNNFormer(x_i) - y_i\|L_{HNN} \tag{18}$$

where θ denotes the learnable parameters of the proposed framework, N is the total number of training pairs, x and y are the input noisy and output denoised images, respectively, and $HNNFormer(\cdot)$ is the proposed framework for image denoising.

4.11. xd_denoise

Implementation details. As shown in Figure 8, They use SCUNet[62] as their baseline model. They employed the PyTorch deep learning framework and conducted experiments on an Ubuntu 20.04 system. The hardware and software setup is as follows: CPU: Intel Xeon Gold 6226R, GPU: Four graphics cards of NVIDIA GeForce RTX 4090,
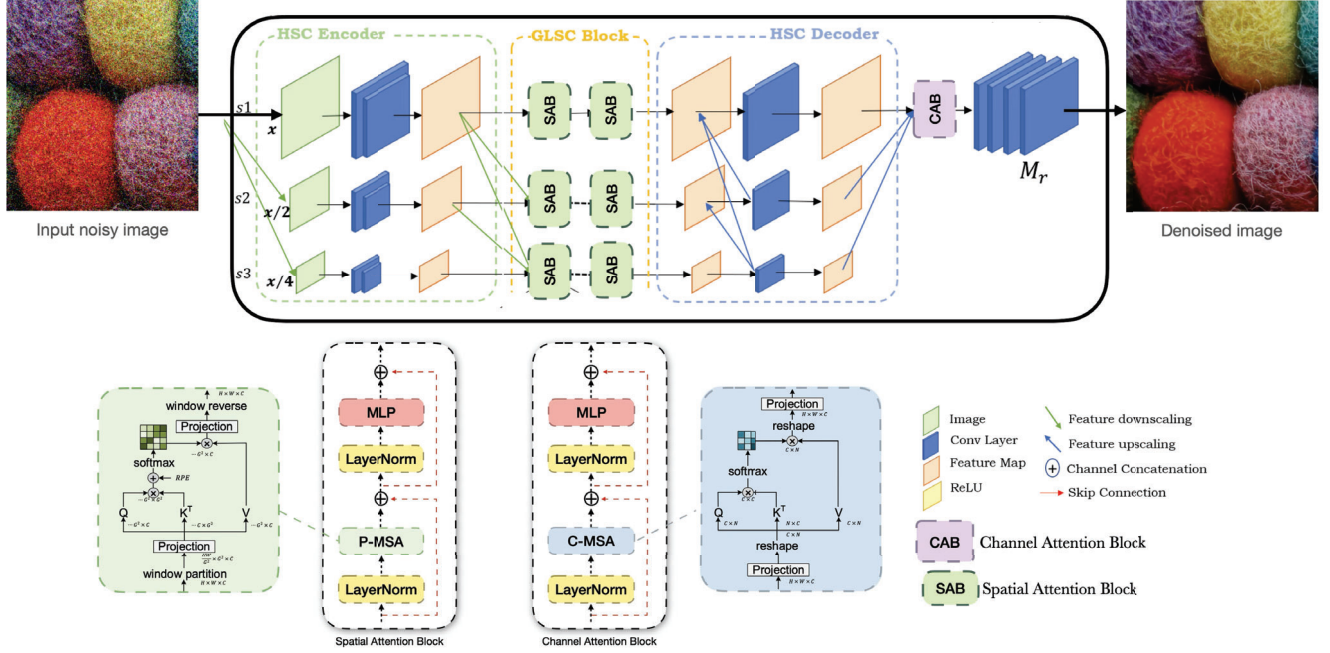
Figure 7. Overview of the HNNFormer proposed by Team KLETech-CEVI: Hierarchical Noise-Deinterlace Transformer for Image Denoising (HNNFormer). The encoder extracts features in three distinct scales, with information passed across hierarchies (green dashed box). Fine-grained global-local spatial and contextual information is learnt through the attention blocks at GLSC (orange dashed box). At the decoder, information exchange occurs in reverse hierarchies (blue dashed box).
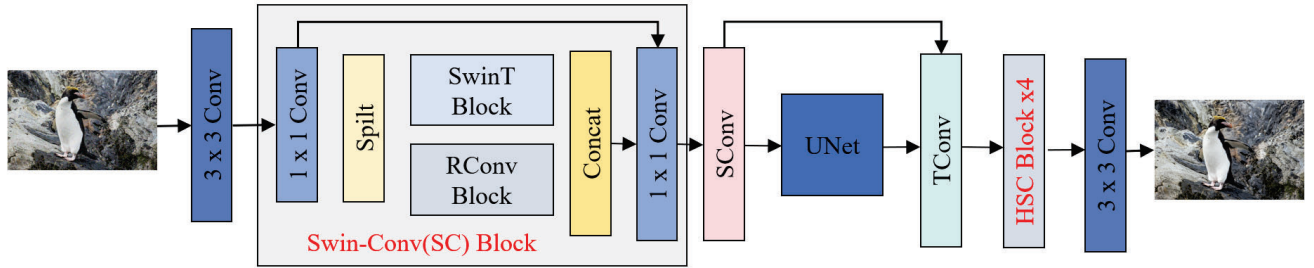

Figure 8. The SCUNet model architecture proposed by Team xd_denoise.

Python version: 3.8.0, PyTorch version: 2.0.0, CUDA version: 11.7. They only use high-definition images from the DIV2K and LSDIR datasets for training and validation. The training set consists of 85791 images (84991 + 800), and the validation set consists of 350 images (250 + 100). They used the Adam optimizer with 100 training epochs, a batch size of 32, and a crop size of 256 × 256. The initial learning rate was set to $1e^{-4}$, with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and no weight decay applied. At epoch 90, the learning rate was reduced to $1e^{-5}$. No data augmentation was applied during training or validation. The model is trained with MSE loss.

Testing description They integrate Test-Time Augmentation(TTA) into their method during testing, including horizontal flip, vertical flip, and 90-degree rotation. They utilized an ensemble technique by chaining three basic U-Net networks and SCUNet, and according to the weights of 0.6 and 0.4, output the results of concatenating the SCUNet model with three UNet models to achieve better performance.

4.12. JNU620

Description. Recently, some research in low-level vision has shown that ensemble learning can significantly improve model performance. Thus, instead of designing a new archi-

tecture, they leverage existing NAFNet [10] and RCAN [63] as basic networks to design a pipeline for image denoising (NRDenoising) based on the idea of ensemble learning, as shown in Fig 9. They find the results are better improved by employing both self-ensemble and model ensemble strategies.
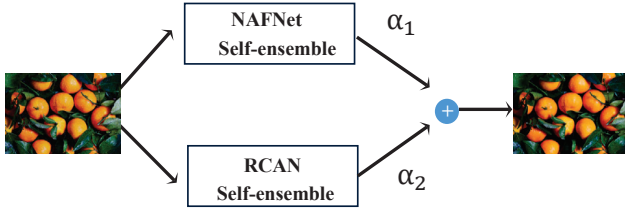

Figure 9. The pipeline of the NRDenoising proposed by Team JNU620.

Implementation details. For the training of NAFNet [10], they utilize the provided DIV2K [2] dataset. The model is trained with MSE loss. They utilize the AdamW optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.9$) for 400K iterations on an NVIDIA Tesla V100 GPU. The initial learning rate is set to 1×10^{-3} and gradually reduces to 1×10^{-7} with the cosine annealing. The training batch is set to 4 and the patch size is 384×384. Random horizontal flipping and rotation are adopted for data augmentation.

For the training of RCAN [63], the provided DIV2K [2] dataset is also employed. The MSE loss is utilized with an initial learning rate of 1×10^{-4}. The Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.99$) is used for 100K iterations. The batch size is 3, and the patch size is 200×200. Data augmentation includes the horizontal flip and the 90-degree rotation.

During inference, they apply a self-ensemble strategy for NAFNet [10] and selectively adopt the TLC [15] method based on the size of input images; For RCAN [63], they utilize a self-ensemble strategy. Finally, the model-ensemble strategy is employed to combine the outputs of NAFNet [10] and RCAN [63].

4.13. PSU_team

General method description. They propose **OptiMalDiff**, a high-fidelity image enhancement framework that reformulates image denoising as an optimal transport problem. The core idea is to model the transition from noisy to clean image distributions via a Schrödinger Bridge-based diffusion process. The architecture (shown in Fig. 10) consists of three main components: (1) a hierarchical Swin Transformer backbone that extracts both local and global features efficiently, (2) a Schrödinger Bridge Diffusion Module that learns forward and reverse stochastic mappings, and (3) a Multi-Scale Refinement Network (MRefNet) designed to progressively refine image details. To enhance realism, they

integrate a PatchGAN discriminator with adversarial training.

Training details. The model is trained from scratch using the DIV2K dataset, without relying on any pre-trained weights. They jointly optimize all modules using a composite loss function that includes diffusion loss, Sinkhorn-based optimal transport loss, multi-scale SSIM and L1 losses, and an adversarial loss. The training spans 300 epochs with a batch size of 8, totaling 35,500 iterations per epoch. The method emphasizes both fidelity and perceptual quality, achieving strong results in PSNR and LPIPS.

4.14. Aurora

They will introduce their algorithm from four aspects: model architecture, data processing methods, training pipeline, and testing pipeline.

Given the excellent performance of generative adversarial networks (GANs) in image generation tasks, and considering that image denoising can also be regarded as a type of generative task, they utilize a generative adversarial network for the denoising task. Specifically, they adopt NAFNet [10] as the generator and have made a series of parameter adjustments. In particular, they increased both the number of channels and the number of modules. Due to the superior performance of the SiLU activation function across various tasks, they replaced the original activation function with SiLU. For the discriminator, they employ a VGG11 architecture without batch normalization (BN) layers, where the ReLU activation function is replaced with LeakyReLU.

In the training stage, they exclusively use the DIV2K and LSDIR datasets [31]. Instead of employing overly complex data augmentation algorithms, they applied simple flipping and rotation techniques for data augmentation. Finally, a patch is cropped from the high-resolution (HR) image, normalized, and then fed into the network.

During training, they progressively trained the model using resolutions of [128, 192, 256]. The model was jointly optimized using L1, L2, and Sobel loss functions. The optimizer and learning rate scheduler used during training were AdamW and CosineAnnealingLR, respectively.

In the inference phase, they employed a self-ensemble strategy and selectively adopted the TLC [14] method to further enhance performance.

4.15. mpu_ai

4.15.1. Method

Existing deep learning-based image restoration methods exhibit inadequate generalization capabilities when faced with a variety of noise types and intensities, thereby significantly impeding their broad application in real-world scenarios. To tackle this challenge, this paper proposes a novel prompt-based learning approach, namely Blind Image Restoration Using Dual-Channel Transformers and
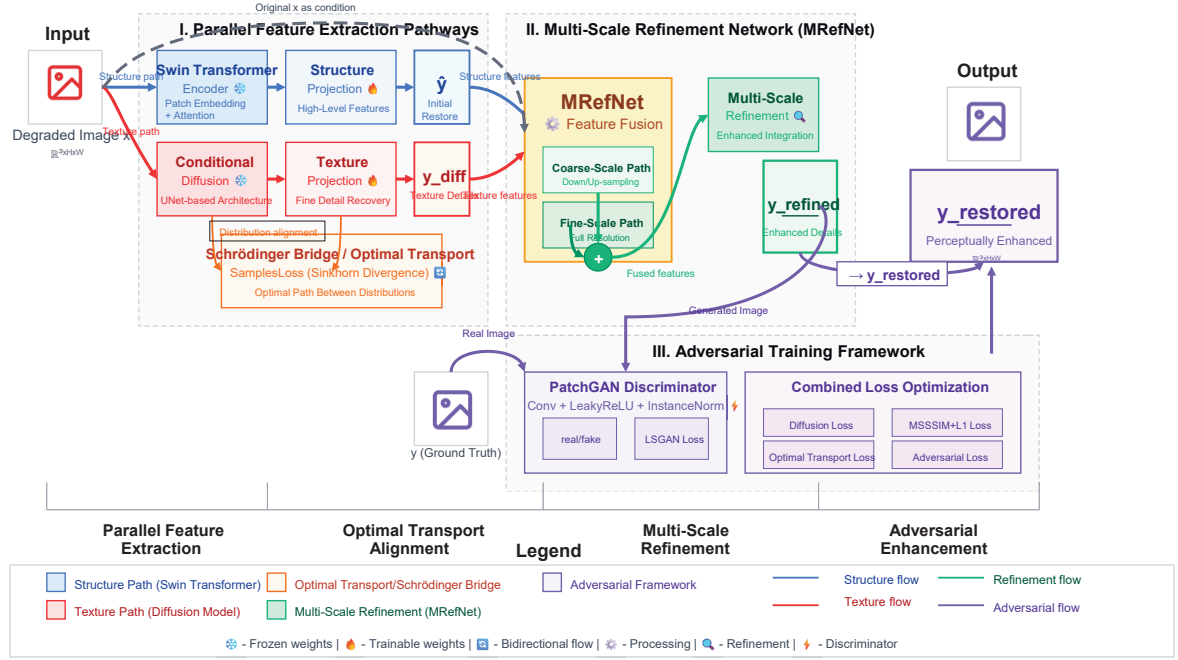
Figure 10. Overview of the OptiMalDiff architecture proposed by PSU_team, combining Schrödinger Bridge diffusion, transformer-based feature extraction, and adversarial refinement.

Multi-Scale Attention Prompt Learning (CTMP), as depicted in Figure 11. The CTMP model features a U-shaped architecture grounded in the Transformer framework, constructed from the enhanced Channel Attention Transformer Block (CATB). During the image restoration process, CTMP adopts a blind image restoration strategy to address diverse noise types and intensities. It integrates an Efficient Multi-Scale Attention Prompt Module (EMAPM) that is based on prompts. Within the EMAPM, an Enhanced Multi-scale Attention (EMA) module is specifically designed. This module extracts global information across different directions and employs dynamic weight calculations to adaptively modulate the importance of features at various scales. The EMA module subsequently fuses the enhanced multi-scale features with the input feature maps, yielding a more enriched feature representation. This fusion mechanism empowers the model to more effectively capture and leverage features at different scales, thereby markedly bolstering its capacity to restore image degradations and showcasing superior generalization capabilities.

4.15.2. Transformer Block Incorporating Channel Attention and Residual Connections

The Transformer Block serves as the cornerstone of their entire model, harnessing the Transformer architecture to extract image features through the self-attention mechanism.

In pursuit of enhanced performance, they have refined the Transformer module by devising a novel architecture that integrates Channel Attention with the self-attention mechanism, thereby combining the strengths of both Transformer and Channel Attention. Specifically, the Transformer focuses on extracting high-frequency information to capture the fine details and textures of images, while Channel Attention excels at capturing low-frequency information to extract the overall structure and semantic information of images. This integration further boosts the image denoising effect. As depicted in Figure 12, the improved Transformer architecture, named the Channel Attention Transformer Block (CATB), primarily consists of the following three modules: Multi-DConv Head Transposed Self-Attention (MDTA), Channel Attention (CA), and Gated-Dconv Feed-Forward Network (GDFN).

The Multi-DConv Head Transposed Self-Attention (MDTA) module enhances the self-attention mechanism's perception of local image features by incorporating multi-scale depthwise convolution operations, effectively capturing detailed image information. The Channel Attention (CA) module, dedicated to information processing along the channel dimension, computes the importance weights of each channel to perform weighted fusion of channel features, thereby strengthening the model's perception of the overall image structure. The Gated-Dconv Feed-Forward
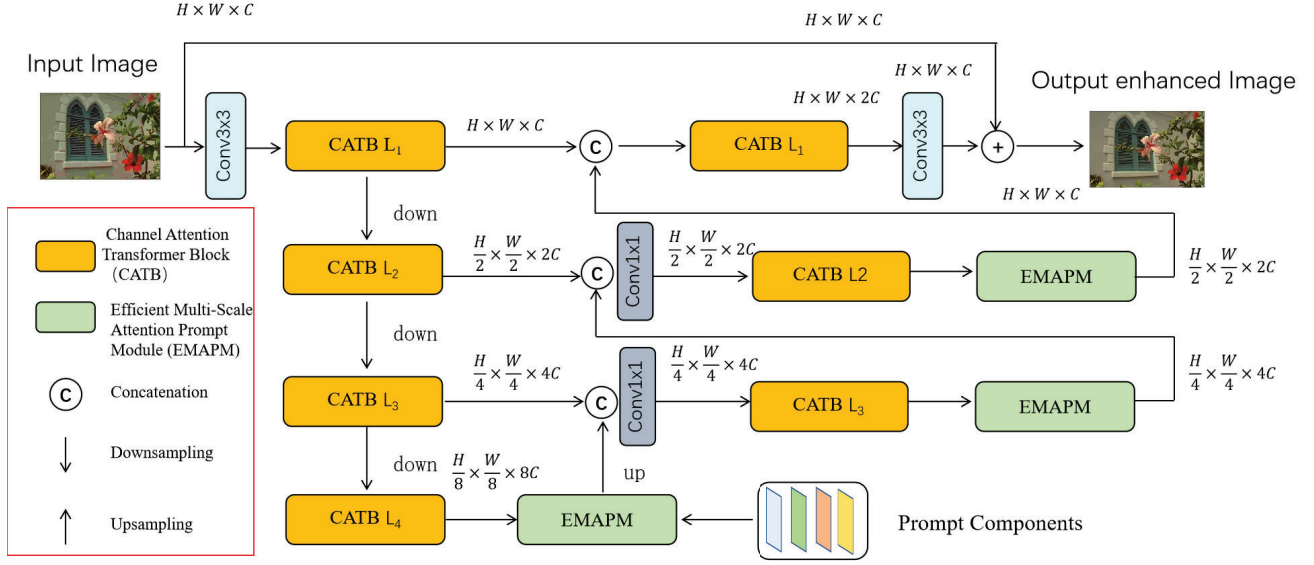
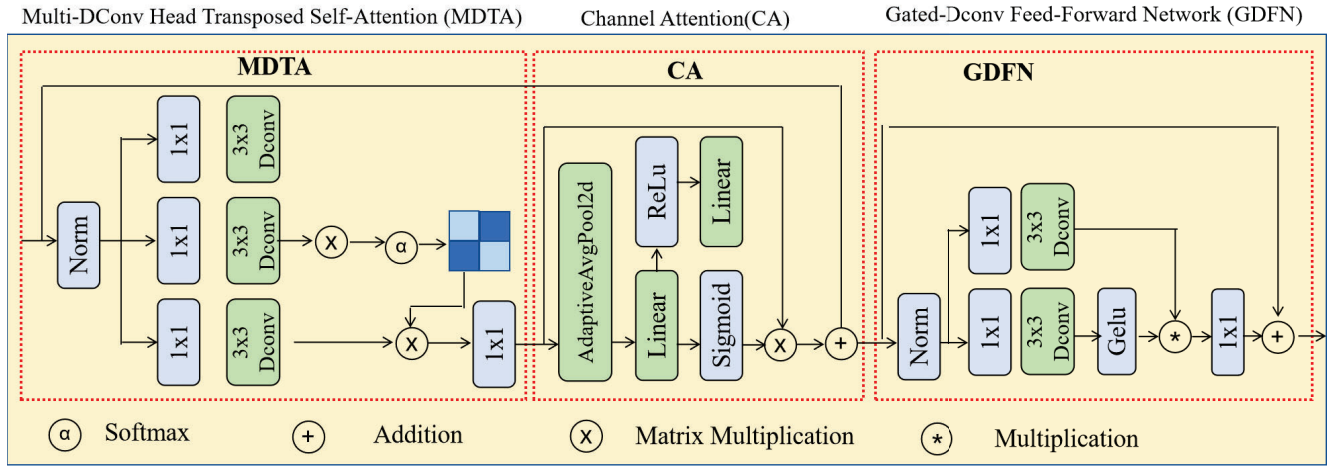Figure 11. The CTMP architecture proposed by Team mpu_ai

Figure 12. The Channel Attention Transformer Block (CATB), proposed by Team mpu_ai

Network (GDFN) module combines the gating mechanism with depthwise convolution operations, aiming to further optimize the nonlinear transformation of features. By introducing the gating mechanism, the model can adaptively adjust the transmission and updating of features based on the dynamic characteristics of the input features, thereby enhancing the flexibility and adaptability of feature representation. Through the synergistic action of these three modules, the improved Transformer architecture can more effectively handle both high-frequency and low-frequency information in images, thereby significantly enhancing the performance of image denoising and restoration.

In image restoration tasks, feature extraction and representation are crucial steps. Traditional convolutional neural networks (CNNs) and Transformer architectures primarily focus on feature extraction in the spatial domain, while paying less attention to the weighting of features in the channel dimension. To address this limitation, they introduce a Channel Attention module in the Transformer Block, creating a Transformer Block that incorporates Channel Attention and Residual Connections. This module weights the channel dimension through global average pooling and fully connected layers, enhancing important channel features while suppressing less important ones. This weighting mechanism enables the model to focus more effectively on key information, thereby improving the quality of restored images. Additionally, the introduction of residual connections further enhances the model's robustness and perfor-

mance. Residual connections ensure that the information of the input features is fully retained after processing by the Channel Attention module by adding the input features directly to the output features. This design not only aids gradient propagation but also retains the original information of the input features when the weighting effect of the Channel Attention module is suboptimal, further boosting the model's robustness.

The proposed model incorporates several key enhancements to improve image restoration quality. Firstly, the Channel Attention Module leverages global average pooling and fully connected layers to selectively enhance important channel features while suppressing less relevant ones. This mechanism enables the model to focus more effectively on critical information, thereby improving the quality of the restored image. Secondly, residual connections are employed to ensure that the original input features are fully retained and added directly to the output features after processing by the Channel Attention Module. This not only aids gradient propagation but also preserves the original information when the weighting effect is suboptimal, thus boosting the model's robustness. Lastly, the LeakyReLU activation function is utilized in the Feed-Forward Network to introduce non-linearity while avoiding the "dying neurons" issue associated with ReLU, further enhancing the model's expressive power. Together, these improvements contribute to a more effective and robust image restoration model.

4.15.3. Efficient Multi-Scale Attention Prompt Module

Addressing multi-scale image degradations is a crucial challenge in image restoration tasks. Traditional feature extraction methods typically capture features at a single scale, neglecting the fusion and interaction of features across multiple scales. To overcome this limitation, they propose a prompt-based blind image restoration approach, incorporating an Efficient Multi-Scale Attention Prompt Module (EMAPM). As be shown in Figure 13, the core of the EMAPM is the Enhanced Multi-scale Attention (EMA) module, which extracts global information in different directions and combines dynamic weight calculations to adaptively adjust the significance of features at various scales, thereby generating a richer feature representation. This design not only enhances the model's adaptability to multi-scale image degradations but also strengthens the expressiveness of features, significantly improving the quality of image restoration. The introduction of the EMA module represents a significant innovation in their image restoration approach. Experimental results validate the effectiveness of the EMA module, demonstrating its ability to substantially boost model performance across multiple image restoration tasks. This innovation not only enhances the model's restoration capabilities but also offers new research directions for image restoration tasks.

The Efficient Multi-Scale Attention Prompt Module (EMAPM) is designed to enhance the model's ability to capture multi-scale features in image restoration tasks. By generating adaptive prompts that focus on different scales and characteristics of the input image, EMAPM allows the model to better handle various types of image degradations. The core components and operations of EMAPM are described as follows:

Module Configuration: To configure the EMAPM, several key parameters are defined:

- **Prompt Dimension** (d_p): This determines the dimension of each prompt vector, which represents the feature space for each prompt.
- **Prompt Length** (L_p): This specifies the number of prompt vectors, which controls the diversity of prompts generated.
- **Prompt Size** (S_p): This sets the spatial size of each prompt vector, which affects the resolution of the prompts.
- **Linear Dimension** (d_l): This is the dimension of the input to the linear layer, which processes the embedding of the input feature map.
- **Factor** (f): This defines the number of groups in the EMA module, which influences the grouping mechanism in the attention process.

Mathematical Formulation: Given an input feature map $x \in \mathbb{R}^{B \times C \times H \times W}$, where B is the batch size, C is the number of channels, and $H \times W$ is the spatial dimension, the operations within EMAPM are defined as follows:

1. **Compute Embedding:** The embedding of the input feature map is computed by averaging the spatial dimensions.

$$\text{emb} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} x_{:,:,i,j} \in \mathbb{R}^{B \times C} \qquad (19)$$

2. **Linear Layer and Softmax:** The embedding is passed through a linear layer followed by a softmax function to generate prompt weights.

$$\text{prompt\_weights} = \text{softmax}(\text{linear\_layer}(\text{emb})) \in \mathbb{R}^{B \times L_p} \tag{20}$$

3. **Generate Prompt:** The prompts are generated by weighting the prompt parameters with the prompt weights and then summing them up. The prompts are then interpolated to match the spatial dimensions of the input feature map.

$$\text{prompt} = \sum_{k=1}^{L_p} \text{prompt\_weights}_{:,k} \cdot \text{prompt\_param}_k \in \mathbb{R}^{B \times d_p \times S_p \times S_p}$$

$$\tag{21}$$

$$\text{prompt} = \text{F.interpolate}(\text{prompt}, (H, W), \text{mode} = \text{"bilinear"})$$
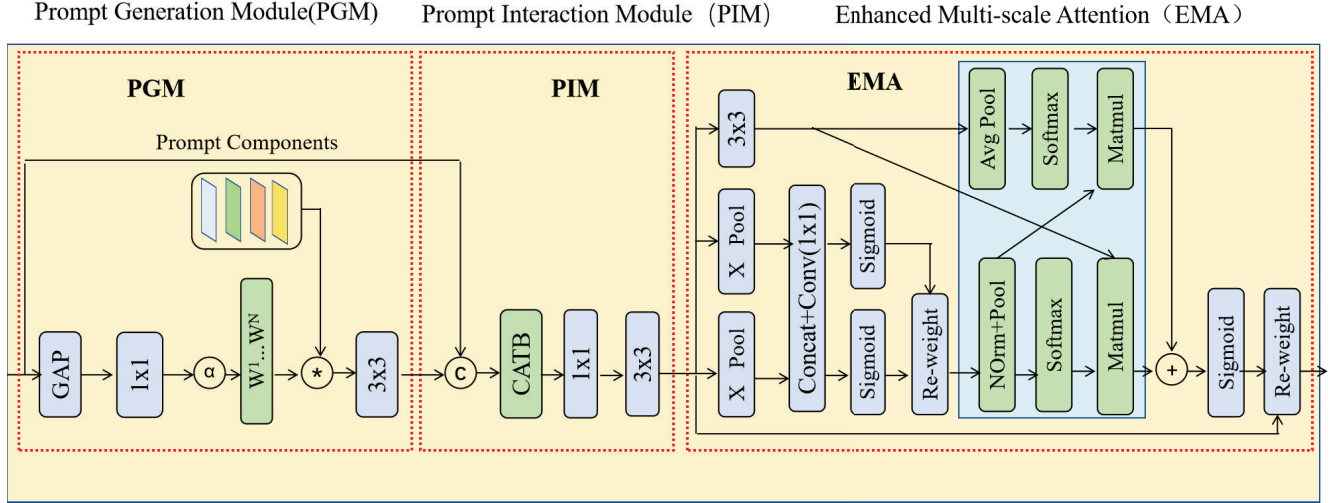
$$\tag{22}$$

Figure 13. Efficient Multi-Scale Attention Prompt Module (EMAPM), proposed by Team mpu_ai.

4. **Enhance Prompt using EMA:** The prompts are enhanced using the Enhanced Multi-scale Attention (EMA) module, which refines the prompts by incorporating multi-scale attention.

$$\text{enhanced\_prompt} = \text{EMA}(\text{prompt}) \in \mathbb{R}^{B \times d_p \times H \times W} \tag{23}$$

5. **Conv3x3:** Finally, the enhanced prompts are processed through a 3x3 convolutional layer to further refine the feature representation.

$$\text{enhanced\_prompt} = \text{conv3x3}(\text{enhanced\_prompt}) \in \mathbb{R}^{B \times d_p \times H \times W} \tag{24}$$

4.15.4. Experiments

In this section, they conducted a series of extensive experiments to comprehensively demonstrate the superior performance of the proposed CTMP model across multiple datasets and benchmarks. The experiments covered a variety of tasks, including denoising and deblocking of compressed images, and were compared with previous state-of-the-art methods. Additionally, they reported the results of ablation studies, which strongly validated the effectiveness of the Channel Attention Transformer Block (CATB) and the Enhanced Multi-scale Attention Prompt Module (EMAPM) within the CTMP architecture.

The CTMP framework is end-to-end trainable without the need for pretraining any individual components. Its architecture consists of a 4-level encoder-decoder, with each level equipped with a different number of Transformer modules, specifically [4, 6, 6, 8] from level 1 to level 4. They placed a Prompt module between every two consecutive decoder levels, resulting in a total of 3 Prompt modules across the entire PromptIR network, with a total of 5 Prompt components. During training, the model was trained with a batch size of 2, leveraging the computational power of a Tesla T4 GPU. The network was optimized through L1 loss, using the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with a learning rate of 2×10^{-4}. To further enhance the model's generalization ability, they used 128×128 cropped blocks as input during training and augmented the training data by applying random horizontal and vertical flips to the input images.

The proposed model in this paper exhibits the following characteristics in terms of overall complexity: It consists of approximately 35.92 million parameters and has a computational cost of 158.41 billion floating-point operations (FLOPs). The number of activations is around 1,863.85 million, with 304 Conv2d layers. During GPU training, the maximum memory consumption is 441.57 MB, and the average runtime for validation is 25,287.67 seconds.

4.15.5. Dataset

To comprehensively evaluate the performance of the CTMP algorithm in image restoration tasks, they conducted experiments in two critical areas: image denoising and deblocking of compressed images. For training, they selected the high-quality DIV2K dataset, which comprises 800 high-resolution clean images with rich textures and details, providing ample training samples to enable the model to perform well under various degradation conditions [2]. Additionally, they used 100 clean/noisy image pairs as the validation set to monitor the model's performance during training and adjust the hyperparameters.

During the testing phase, they chose several widely used datasets, including Kodak, LIVE1, and BSDS100, to comprehensively assess the algorithm's performance. The Kodak dataset consists of 24 high-quality images with diverse scenes and textures, commonly used to evaluate the visual

effects of image restoration algorithms [1]. The LIVE1 dataset contains a variety of image types and is widely used for image quality assessment tasks, effectively testing the algorithm's performance under different degradation conditions [47]. The BSDS100 dataset includes 100 images with rich textures and edge information, providing a comprehensive evaluation of the algorithm's performance in image restoration tasks [41].

By testing on these representative datasets, they were able to comprehensively evaluate the CTMP algorithm's performance across different degradation types and image conditions, ensuring its effectiveness and reliability in practical applications.

4.16. OptDenoiser

Method They introduce a two-stage transformer-based network that effectively maps low-resolution noisy images to their high-resolution counterparts, as depicted in Fig. 14. The proposed framework comprises two independent encoder-decoder blocks (EDBs) and Multi-Head correlation blocks to generate visually coherent images [46]. To enhance reconstruction efficiency, they integrate illumination mapping [46] guided by Retinex theory [26]. Additionally, they conduct a theory, an in-depth evaluation of the effectiveness of illumination mapping in general image reconstruction tasks, including image denoising. Therefore, their framework integrates the Retinexformer [9] network as the first stage. In the context of image denoising, Retinexformer surpasses conventional denoisers such as UFormer, Restormer, and DnCNN. However, like other denoising methods, Retinexformer encounters challenges, including jagged edges, blurred outputs, and difficulties in capturing and representing complex structures in noisy inputs. To address these obstacles, they incorporate the MHC, followed by an additional EDB in their framework. This design effectively exploits feature correlations from intermediate outputs, enabling more accurate reconstruction with improved structural fidelity and texture preservation. Furthermore, they integrate a perceptual loss function with luminance-chrominance guidance [46] to mitigate color inconsistencies, ensuring visually coherent and perceptually refined reconstructions.

4.16.1. Global Method Description

Training Procedure: During the training phase, input images were randomly cropped into 512×512 patches and subsequently downscaled to 128×128 to enhance the model's ability to capture spatial features effectively. A fixed learning rate of 0.0001 was maintained throughout the training process. The model was trained exclusively on the LSDIR and DIV2K datasets, without the inclusion of any additional training, validation, or testing data.
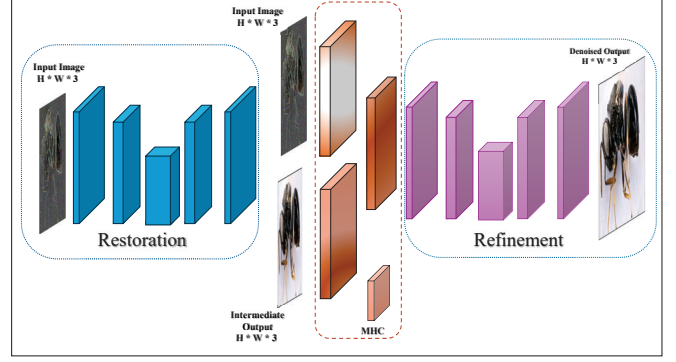

Figure 14. Overview of the two-stage OptDenoiser framework for image denoising.

4.16.2. Technical details

The proposed solution is implemented with the *PyTorch* framework. The networks were optimized using the *Adam* optimizer, where the hyperparameters were tuned as $\beta_1 = 0.9$, $\beta_2 = 0.99$, and the learning rate was set to 1×10^{-4}. They trained their model using randomly cropped image patches with a constant batch size of 4, which takes approximately 72 hours to complete. All experiments were conducted on a machine equipped with an *NVIDIA RTX 3090 GPU*.

4.17. AKDT

Method. The team utilizes their existing network *Adaptive Kernel Dilation Transformer* [5] (**AKDT**), published at VISAPP 2025, with code published at https://github.com/albrateanu/AKDT. Figure 15 presents the architecture of **AKDT**. It proposes a novel convolutional structure with learnable dilation rates: the Learnable Dilation Rate (**LDR**) Block, used to formulate the Noise Estimator (**NE**) Module, which is leveraged within the self-attention and feed-forward mechanisms.

LDR. The Learnable Dilation Rate module lies at the foundation of **AKDT** and helps the model effectively pick optimal dilation rates for convolutional kernels. Given an input feature map $\mathbf{F}_{\text{in}} \in \mathbb{R}^{H \times W \times C}$, it is formulated as the weighted concatenaton of N dilated convolutions:

$$\mathbf{F}_{\text{LDR}} = conv1 \times 1 \left(\mathbf{concat}_{i=1}^{N} \alpha_i \times conv3 \times 3_i (\mathbf{F}_{\text{in}}) \right) \quad (25)$$

where **concat** represents the channel-wise concatenation operation. The specific dilation rates picked for LDR are a hyperparameter that is carefully chosen to balance between performance and computational efficiency.

NE. The Noise Estimator integrates both global and local context understanding through its unique structure. This module consists of two distinct parallel components: the Global and Local **LDR** modules with selected dilation rates for capturing global and local structure. It is defined as:
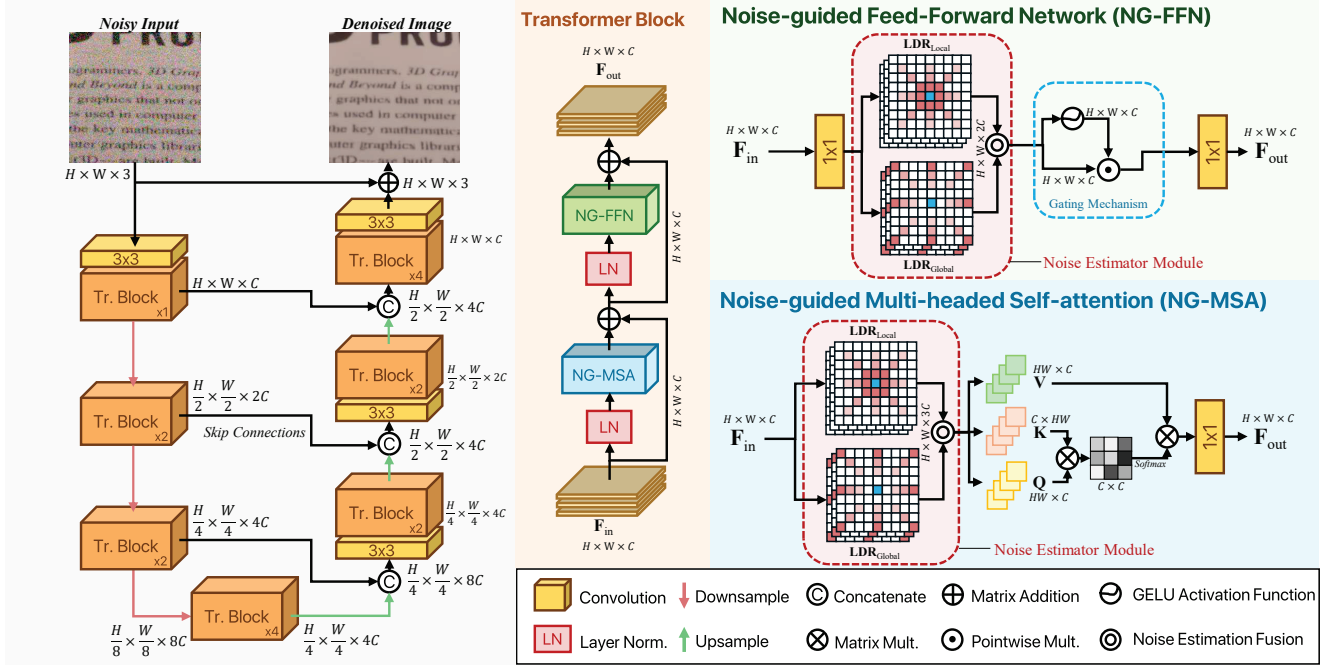
Figure 15. Overall framework of **AKDT** - Adaptive Kernel Dilation Transformer.

$$\mathbf{NE} = \varrho\left(\mathbf{LDR}_{\text{Global}}, \mathbf{LDR}_{\text{Local}}\right) \qquad (26)$$

where ϱ is the Noise Estimation Fusion operation that merges global and local noiseless feature context.

NG-MSA. To ensure efficiency in their Noise-guided Multi-headed Self-Attention, they utilize the Transposed Multi-headed Self-Attention mechanism [59] as baseline. They then integrate their proposed **NE** module for the **Q,K,V** extraction phase, to ensure self-attended feature maps are produced utilizing noiseless context. Therefore, given the input feature map $\mathbf{F}_{\text{in}} \in \mathbb{R}^{H \times W \times C}$, they can define this process as:

$$\{\mathbf{Q,K,V}\} = \mathbf{NE}(\mathbf{F}_{\text{in}}), \quad \mathbf{Q,K,V} \in \mathbb{R}^{HW \times C} \qquad (27)$$

Then, **Q,K** are used to compute the self-attention map by matrix multiplication and Softmax activation, which is then applied to **V** to obtain the final self-attended feature map.

NG-FFN. The Noise-guided Feed-forward Network also utilizes the **NE** module for noise-free feature extraction context. It consists of a series of convolutional layers with a gating mechanism used to selectively apply non-linear activations. The noise-free features, obtained from projecting the input through their **NE** will be referred to as $\mathbf{F}_{\text{NE}} \in \mathbb{R}^{H \times W \times C}$. Consequently, the feed-forward process can be described as:

$$\mathbf{F}_{\text{NG-FFN}} = \phi\left(W_1 \mathbf{F}_{\text{NE}}\right) \odot W_2 \mathbf{F}_{\text{NE}} + \mathbf{F}_{\text{NE}}, \qquad (28)$$

here ϕ denotes the GELU activation function, $\odot$ represents element-wise multiplication, and W_1, W_2 are the learnable parameters of the parallel paths.

Implementation. **AKDT** is implemented by PyTorch. They only use the DIV2K dataset for training. The model is trained using the Adam Optimizer for 150k iterations, with an initial learning rate set at $2e - 4$ which gradually decreases through a Cosine Annealing scheme. Each iteration consists of a batch of $4\ 600 \times 600$ randomly-cropped image patches that undergo data augmentation (random flipping/rotation). To optimize their network, they utilize a hybrid loss function capable to capture pixel-level, multi-scale and perceptual differences [6] [4]. Testing is performed via standard inference, without additional enhancement techniques.

4.18. X-L

General method description. To ensure performance while reducing computational overhead, they adopted the following strategy: leveraging two leading approaches, Xformer [60] and SwinIR [33], the pipeline is shown in Fig. 16. They directly utilized their pre-trained models to perform self-ensemble, generating two output results. Then, they conducted model ensemble on these two outputs, integrating the results between models to obtain the final reconstruction result.

Training details. They do not require additional training; instead, they directly leverage existing methods and their pre-trained models for inference. This approach not
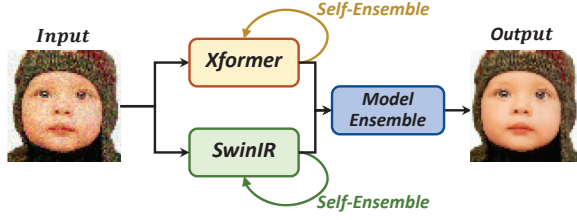
Figure 16. Overview of the MixEnsemble pipeline proposed by Team X-L.

only saves significant computational resources and time but also fully utilizes the excellent models and valuable expertise available in the field. By directly employing these pre-trained models, they can quickly generate high-quality predictions while avoiding the high costs and complexity associated with training models from scratch.

4.19. Whitehairbin

4.19.1. Introduce

Their method is based on the Refusion[40] model proposed in previous work, and they trained it on the dataset provided by this competition to validate its effectiveness. The Refusion model itself is a denoising method based on the diffusion model framework. Its core idea is to guide the reverse diffusion process by learning the noise gradient (score function) at different time steps t. Within the Refusion framework, they can still flexibly choose NAFNet or UNet as the neural network backbone architecture to adapt to different computational resources and performance requirements. NAFNet is known for its efficiency, while UNet excels in preserving details. The denoising process follows a stochastic differential equation (SDE) approach, which calculates the score function by predicting the noise residual and iteratively removes noise. Through training and validation on the competition dataset, their method ultimately achieved a test performance of PSNR 27.07 and SSIM 0.79.

4.19.2. Method details

General method description Their proposed denoising method is based on a diffusion model framework, where the network is designed to estimate the noise gradient (*score function*) at different time steps t to guide the reverse diffusion process. The core architecture consists of a neural backbone, which can be either NAFNet, selected based on a trade-off between computational efficiency and denoising quality.

NAFNet features a lightweight structure optimized for high-speed image restoration, incorporating a self-gated activation mechanism (SimpleGate), simplified channel attention (SCA), and depth-wise convolutions, making it highly efficient. UNet, on the other hand, is a widely adopted architecture for image denoising, leveraging an encoder-decoder structure with skip connections to preserve spatial details while extracting multi-scale features.

The denoising process follows a stochastic differential equation (SDE) approach, where Gaussian noise $\mathcal{N}(0, \sigma_t^2 I)$ is added to the clean image x_0 during the forward diffusion process, and the network is trained to predict the noise residual $s_\theta(x_t, t)$. This predicted noise is used to compute the *score function*, which guides the reverse diffusion process, progressively removing noise through an iterative update step:

$$x_{t-1} = x_t - 0.5 \cdot \sigma_t^2 \cdot \text{score}(x_t, t) \cdot dt.$$

To improve sampling efficiency, they integrate an ODE-based sampling strategy, which allows for faster denoising while maintaining high restoration quality. Additionally, they employ a cosine noise schedule, which ensures a smooth noise transition across time steps and improves training stability. The network is optimized using a custom loss function that minimizes the deviation between the predicted noise and the true noise, ensuring precise score estimation.

Training is conducted with the Lion optimizer, incorporating a learning rate scheduler for improved convergence. To enhance computational efficiency, they apply mixed precision training, reduce time steps T, and utilize lightweight backbone networks, striking a balance between high-quality denoising and efficient execution.

Training description They trained their diffusion-based denoising model on a mixed dataset composed of DIV2K and LSDIR, which contained high-resolution images with diverse textures and content. The dataset was augmented with random cropping, horizontal flipping, and other data augmentation techniques to improve model generalization.

The backbone network was selected from either NAFNet, with the feature channel width set to 64. They experimented with different channel sizes and determined that 64 channels provided a good balance between performance and computational efficiency.

They employed theLion optimizer with $\beta_1 = 0.95$ and $\beta_2 = 0.98$ to ensure faster convergence and better stability during training. The learning rate was initialized at 2×10^{-4} and was reduced by half after every 200k iterations using a CosineAnnealingLR scheduler to achieve smoother convergence.

The loss function was a Matching Loss designed to minimize the distance between the predicted and true noise residuals. This function integrated L1 and L2 components, weighted dynamically based on the noise variance at different time steps to stabilize the training across different diffusion levels.

They applied mixed precision training with automatic gradient scaling to accelerate training while reducing memory usage. The model was trained for a total of 800k itera-
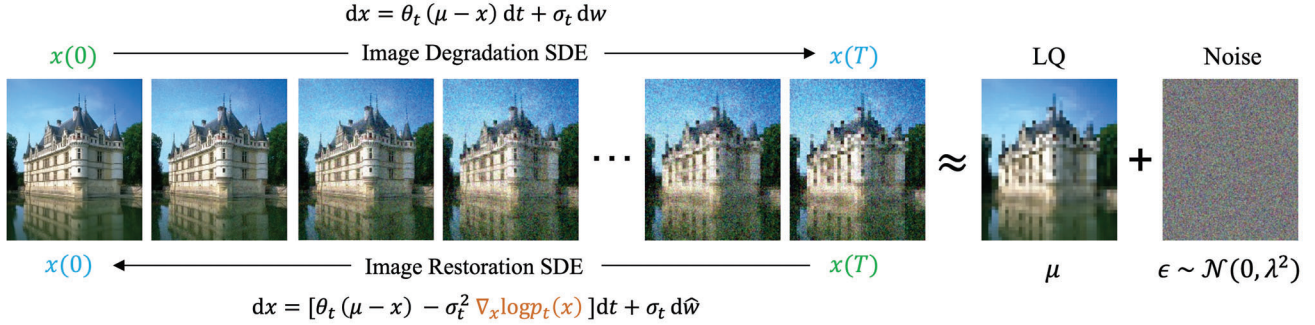
$$dx = \theta_t(\mu - x)\,dt + \sigma_t\,dw$$

Figure 17. Diffusion model for image denoising from Team Whitehairbin.

tions, and each batch contained 16 cropped patches of size 128×128. Training was conducted using a single NVIDIA RTX 4090 GPU, and the entire process took approximately 36 hours to complete.

To ensure robust noise modeling, a cosine noise schedule was adopted, which progressively adjusted the noise level throughout the training process, allowing the model to better capture high-frequency details during the denoising phase.

Testing description During the training phase, they validated the model using the official validation dataset provided by the NTIRE 2025 competition. The validation set included images with Gaussian noise of varying intensities, and the model was assessed based on both PSNR and SSIM metrics.

Upon completing 800k iterations, the model achieved a peak PSNR of 26.83 dB and an SSIM of 0.79 on the validation dataset, indicating effective noise suppression and structure preservation.

After training was completed, the model was rigorously tested using the official test set to verify its effectiveness in real-world scenarios. They conducted multiple test runs with different noise levels to ensure model robustness across various conditions. The test results confirmed that the model performed consistently well in Gaussian noise removal, maintaining high PSNR and SSIM values across diverse image types.

To further evaluate the performance, they applied both SDE-based and ODE-based sampling methods during inference. ODE sampling provided a faster and more deterministic denoising process, while SDE sampling yielded more diverse results. The final submitted model leveraged ODE sampling to achieve a balance between quality and inference speed.

4.20. mygo

U-Net adopts a typical encoder-decoder structure. The encoder is responsible for downsampling the input image, extracting features at different scales to capture the global in-

formation and semantic features of the image. The decoder performs upsampling, restoring the feature maps to the original image size and progressively recovering the detailed information of the image. This architecture enables U-Net to achieve rich global semantic information while accurately restoring image details when processing high-definition images, thereby realizing high-precision segmentation.

The U-Net architecture is characterized by its symmetric encoder-decoder structure with skip connections. In the encoder (or contracting path), the network progressively downsamples the input image through multiple convolutional layers interspersed with max-pooling operations. This process allows the model to extract hierarchical features at various scales, capturing both the global context and semantic information of the image.

In the decoder (or expansive path), the network employs transposed convolutions (or upsampling layers) to gradually upscale the feature maps back to the original image resolution. During this process, the decoder receives additional information from the encoder via skip connections, which concatenate corresponding feature maps from the encoder to those in the decoder. This mechanism helps in refining the output by incorporating fine-grained details and spatial information, which are crucial for accurate image restoration or segmentation.

This design ensures that U-Net can effectively handle high-resolution images by leveraging both the broad contextual understanding gained from the encoder and the detailed spatial information preserved through the skip connections. Consequently, this dual capability of capturing global semantics and local details makes U-Net particularly powerful for tasks that require precise image segmentation. The uniqueness of U-Net lies in its skip connections. These skip connections directly transfer feature maps of the same scale from the encoder to the corresponding layers in the decoder. This mechanism allows the decoder to utilize low-level feature information extracted by the encoder, aiding in the better recovery of image details. When processing high-definition images, these low-level features contain abundant

edge, texture, and other detail information, which is crucial for accurate image segmentation.

Compared to Fully Convolutional Networks (FCNs), U-Net stands out because of its use of skip connections. FCN is also a commonly used model for image segmentation, but lacks the skip connections found in U-Net, resulting in poorer performance in recovering detailed image information. When processing high-definition images, FCNs can produce blurry segmentation results with unclear edges. In contrast, U-Net can better preserve the details of the image through its skip connections, thereby improving the accuracy of segmentation.

Our model resizes all images to 512*512 for training, which facilitates the rapid extraction of image features and effectively reduces the usage of video memory (VRAM). Next, they feed the images into the network model and compute the loss of the output images. In particular, their loss function incorporates both MSE (mean squared error) and SSIM (structured similarity index measure), allowing the model to focus on pixel-level accuracy during training while also emphasizing the structural features of the images. This dual approach improves the overall performance of the model. They use the Adam optimizer for training, which dynamically adjusts the learning rate during the training process based on the first and second moments of the gradients. This allows it to automatically select the appropriate step sizes for each parameter, leading to more efficient convergence compared to fixed learning rate methods. Additionally, Adam helps reduce the overall memory footprint by maintaining only a few extra parameters per weight, contributing to its efficiency in practical applications. In particular, they employ an early stopping mechanism to avoid redundant computations.

It is worth mentioning that they have implemented an early stopping mechanism. This approach helps prevent overfitting by halting the training process when the performance on a validation set stops improving, thus avoiding unnecessary computations and saving computational resources. Early stopping monitors a chosen metric (such as validation loss) and stops training when no improvement is observed over a predefined number of epochs, effectively reducing the risk of overfitting and ensuring efficient use of computational resources.
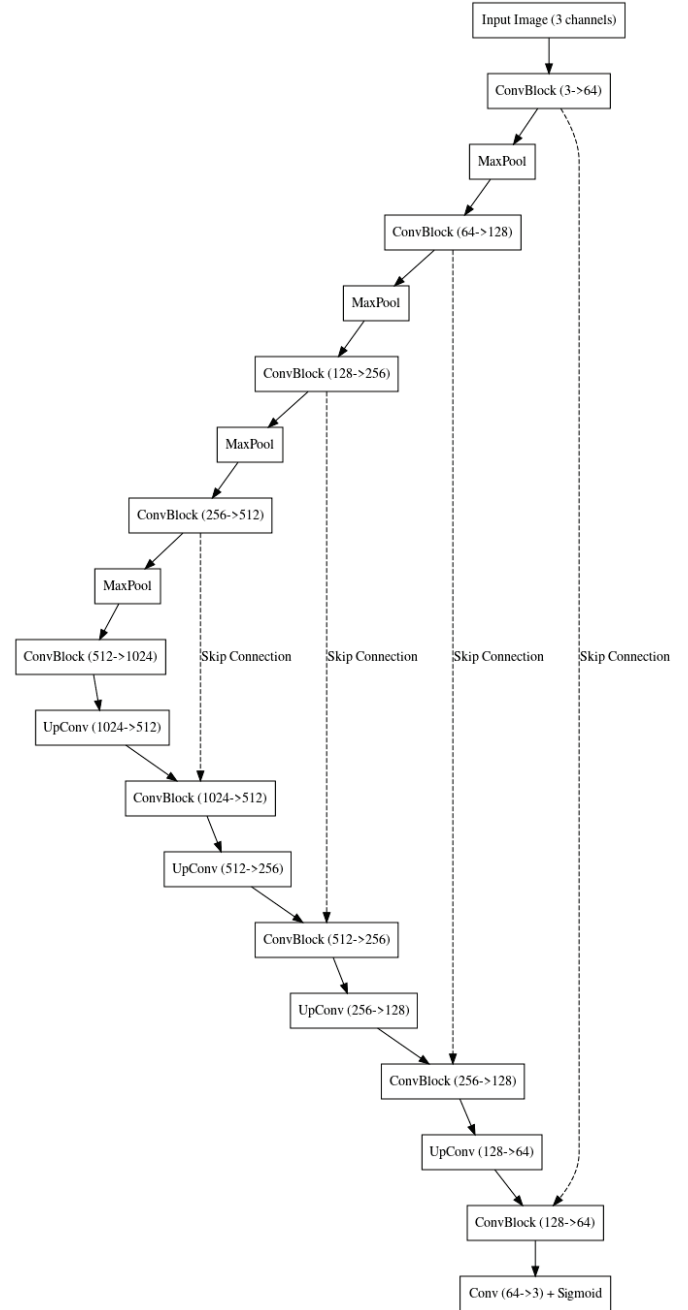
Acknowledgments

Figure 18. Unet model architecture from Team mygo.

A. Teams and affiliations

NTIRE 2025 team

Title: NTIRE 2025 Image Denoising Challenge
Members:
Lei Sun[1] (lei.sun@insait.ai),
Hang Guo[2] (cshguo@gmail.com),
Bin Ren[1,3,4] (bin.ren@unitn.it),

Luc Van Gool[1] (vangool@vision.ee.ethz.ch),
Radu Timofte[5] (Radu.Timofte@uni-wuerzburg.de)
Yawei Li[6] (li.yawei.ai@gmail.com),
Affiliations:
[1] INSAIT,Sofia University,"St.Kliment Ohridski", Bulgaria
[2] Tsinghua University, China
[3] University of Pisa, Italy
[4] University of Trento, Italy
[5] University of Würzburg, Germany
[6] ETH Zürich, Switzerland

Samsung MX (Mobile eXperience) Business & Samsung R&D Institute China - Beijing (SRC-B)

Title: Dynamic detail-enhanced image denoising framework

Members:
Xiangyu Kong[1] (xiangyu.kong@samsung.com), Hyunhee Park[2], Xiaoxuan Yu[1], Suejin Han[2], Hakjae Jeon[2], Jia Li[1], Hyung-Ju Chun[2]
Affiliations:
[1] Samsung R&D Institute China - Beijing (SRC-B)
[2] Department of Camera Innovation Group, Samsung Electronics

SNUCV

Title: Deep ensemble for Image denoising

Members:
Donghun Ryou[1] (dhryou@snu.ac.kr), Inju Ha[1], Bohyung Han[1]
Affiliations:
[1] Seoul National University

BuptMM

Title: DDU——Image Denoising Unit using transformer and morphology method

Members:
Jingyu Ma[1] (whalemjy@bupt.edu.cn), Zhijuan Huang[2], Huiyuan Fu[1], Hongyuan Yu[2], Boqi Zhang[1], Jiawei Shi[1], Heng Zhang[2], Huadong Ma[1]
Affiliations:
[1] Beijing University of Posts and Telecommunications
[2] Xiaomi Inc., China

HMiDenoise

Title: Hybrid Denosing Method Based on HAT

Members:
Zhijuan Huang[1](huang_199109@163.com), Jingyu Ma[2], Hongyuan Yu[1], Heng Zhang[1], Huiyuan Fu[2], Huadong Ma[2]
Affiliations:

[1] Xiaomi Inc.
[2] Beijing University of Posts and Telecommunications

Pixel Purifiers

Title: Denoiser using Restormer and Hard Dataset Mining

Members:
Deepak Kumar Tyagi[1] (deepak.tyagi@samsung.com), Aman Kukretti[1], Gajender Sharma[1], Sriharsha Koundinya[1], Asim Manna[1]
Affiliations:
[1] Samsung R&D Institute India - Bangalore (SRI-B)

Alwaysu

Title: Bias-Tuning Enables Efficient Image Denoising

Members:
Jun Cheng [1] (jcheng24@hust.edu.cn), Shan Tan [1]
Affiliations:
[1] Huazhong University of Science and Technology

Tcler_Denosing

Title: Tcler Denoising

Members:
Jun Liu[1,2] (jun63.liu@tcl.com), Jiangwei Hao[1,2], Jianping Luo[1,2], Jie Lu[1,2]
Affiliations:
[1] TCL Corporate Research
[2] TCL Science Park International E City - West Zone, Building D4

cipher_vision

Title: Pureformer: Transformer-Based Image Denoising

Members:
Satya Narayan Tazi[1] (satya.tazi@ecajmer.ac.in), Arnim Gautam[1], Aditi Pawar[1], Aishwarya Joshi[2], Akshay Dudhane[3], Praful Hambadre[4], Sachin Chaudhary[5], Santosh Kumar Vipparthi[5], Subrahmanyam Murala[6],
Affiliations:
[1] Government Engineering College Ajmer
[2] Mohamed bin Zayed University of Artificial Intelligence gence, Abu Dhabi
[3] University of Petroleum and Energy Studies, Dehradun
[4] Indian Institute of Technology, Mandi
[5] Indian Institute of Technology, Ropar
[6] Trinity College Dublin, Ireland

Sky-D

Title: A Two-Stage Denoising Framework with Generalized Denoising Score Matching Pretraining and Supervised Fine-tuning

Members:
Jiachen Tu[1] (jtu9@illinois.edu)
Affiliations:
[1] University of Illinois Urbana-Champaign

KLETech-CEVI

Title: HNNFormer: Hierarchical Noise-Deinterlace Transformer for Image Denoising
Members:
Nikhil Akalwadi[1,3] (nikhil.akalwadi@kletech.ac.in), Vijayalaxmi Ashok Aralikatti[1,3], Dheeraj Damodar Hegde[2,3], G Gyaneshwar Rao[2,3], Jatin Kalal[2,3], Chaitra Desai[1,3], Ramesh Ashok Tabib[2,3], Uma Mudenagudi[2,3]
Affiliations:
[1] School of Computer Science and Engineering, KLE Technological University
[2] School of Electronics and Communication Engineering, KLE Technological University
[3] Center of Excellence in Visual Intelligence (CEVI), KLE Technological University

xd_denoise

Title: SCUNet for image denoising
Members:
Zhenyuan Lin[1] (linzhenyuan@stu.xidian.edu.cn), Yubo Dong[1], Weikun Li[2], Anqi Li[1], Ang Gao[1]
Affiliations:
[1] Xidian University
[2] Guilin University Of Electronic Technology

JNU620

Title: Image Denoising using NAFNet and RCAN
Members:
Weijun Yuan[1] (yweijun@stu2022.jnu.edu.cn), Zhan Li[1], Ruting Deng[1], Yihang Chen[1], Yifan Deng[1], Zhanglu Chen[1], Boyang Yao[1], Shuling Zheng[2], Feng Zhang[1], Zhiheng Fu[1]
Affiliations:
[1] Jinan University
[2] Guangdong University of Foreign Studies

PSU_team

Title: OptimalDiff: High-Fidelity Image Enhancement Using Schrödinger Bridge Diffusion and Multi-Scale Adversarial Refinement

Members:
Anas M. Ali[1] (aaboessa@psu.edu.sa), Bilel Benjdira[1],
Wadii Boulila[1]

Affiliations:
[1] Robotics and Internet-of-Things Laboratory, Prince Sultan University, Riyadh, Saudi Arabia

Aurora

Title: GAN + NAFNet: A Powerful Combination for High-Quality Image Denoising
Members:
JanSeny (1225049871@qq.com), Pei Zhou

mpu_ai

Title: Enhanced Blind Image Restoration with Channel Attention Transformers and Multi-Scale Attention Prompt Learning
Members:
Jianhua Hu[1] (p2412994@mpu.edu.mo), K. L. Eddie Law[1]
Affiliations:
[1] Macao Polytechnic University

OptDenoiser

Title: Towards two-stage OptDenoiser framework for image denoising.
Members:
Jaeho Lee [1] (jaeho.lee@opt-ai.kr), M.J. Aashik Rasool[1], Abdur Rehman[1], SMA Sharif[1], Seongwan Kim[1]
Affiliations:
[1] Opt-AI Inc, Marcus Building, Magok, Seoul, South Korea

AKDT

Title: High-resolution Image Denoising via Adaptive Kernel Dilation Transformer
Members:
Alexandru Brateanu[1] (alexandru.brateanu@student.manchester.ac.uk), Raul Balmez[1], Ciprian Orhei[2], Cosmin Ancuti[2]
Affiliations:
[1] University of Manchester - Manchester, United Kingdom
[2] Polytechnica University Timisoara - Timisoara, Romania

X-L

Title: MixEnsemble
Members:
Zeyu Xiao[1] (zeyuxiao1997@163.com), Zhuoyuan Li[2]
Affiliations:
[1] National University of Singapore
[2] University of Science and Technology of China

Whitehairbin

Title: Diffusion-based Denoising Model

Members:
Ziqi Wang[1] (wangziqi-7@outlook.com), Yanyan Wei[1], Fei Wang[1], Kun Li[1], Shengeng Tang[1], Yunkai Zhang[1]

Affiliations:
[1] Hefei University of Technology, China

mygo

Title: High-resolution Image Denoising via Unet neural network
Members:
Weirun Zhou[1] (1764772710@qq.com), Haoxuan Lu[2]

Affiliations:
[1] Xidian University
[2] China University of Mining and Technology

References

[1] Kodak dataset. http://r0k.us/graphics/kodak/. 19

[2] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 126–135, 2017. 2, 5, 8, 11, 14, 18

[3] Yuval Becker, Raz Z Nossek, and Tomer Peleg. Make the most out of your net: Alternating between canonical and hard datasets for improved image demosaicing. *CoRR*, 2023. 6

[4] Alexandru Brateanu and Raul Balmez. Kolmogorov-arnold networks in transformer attention for low-light image enhancement. In *2024 International Symposium on Electronics and Telecommunications (ISETC)*, pages 1–4. IEEE, 2024. 20

[5] Alexandru Brateanu, Raul Balmez, Adrian Avram, and Ciprian Orhei. Akdt: Adaptive kernel dilation transformer for effective image denoising. *Proceedings Copyright*, 418: 425. 19

[6] Alexandru Brateanu, Raul Balmez, Ciprian Orhei, Cosmin Ancuti, and Codruta Ancuti. Enhancing low-light images with kolmogorov–arnold networks in transformer attention. *Sensors*, 25(2):327, 2025. 20

[7] Matthew Brown and David G Lowe. Automatic panoramic image stitching using invariant features. *International journal of computer vision*, 74:59–73, 2007. 7

[8] Han Cai, Chuang Gan, Ligeng Zhu, and Song Han. Tinytl: Reduce memory, not parameters for efficient on-device learning. *Advances in Neural Information Processing Systems*, 33:11285–11297, 2020. 7

[9] Yuanhao Cai, Hao Bian, Jing Lin, Haoqian Wang, Radu Timofte, and Yulun Zhang. Retinexformer: One-stage retinex-based transformer for low-light image enhancement. In *Pro-ceedings of the IEEE/CVF international conference on computer vision*, pages 12504–12513, 2023. 19

[10] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European conference on computer vision*, pages 17–33. Springer, 2022. 3, 14

[11] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 22367–22377, 2023. 5

[12] Zheng Chen, Kai Liu, Jue Gong, Jingkai Wang, Lei Sun, Zongwei Wu, Radu Timofte, Yulun Zhang, et al. NTIRE 2025 challenge on image super-resolution (×4): Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[13] Zheng Chen, Jingkai Wang, Kai Liu, Jue Gong, Lei Sun, Zongwei Wu, Radu Timofte, Yulun Zhang, et al. NTIRE 2025 challenge on real-world face restoration: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[14] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu. Revisiting global statistics aggregation for improving image restoration. *arXiv preprint arXiv:2112.04491*, 2(4):5, 2021. 14

[15] Xiaojie Chu, Liangyu Chen, Chengpeng Chen, and Xin Lu. Improving image restoration by revisiting global information aggregation. In *European Conference on Computer Vision*, pages 53–71. Springer, 2022. 14

[16] Marcos Conde, Radu Timofte, et al. NTIRE 2025 challenge on raw image restoration and super-resolution. In *Proceed-ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[17] Marcos Conde, Radu Timofte, et al. Raw image reconstruction from RGB on smartphones. NTIRE 2025 challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[18] Egor Ershov, Sergey Korchagin, Alexei Khalin, Artyom Panshin, Arseniy Terekhin, Ekaterina Zaychenkova, Georgiy Lobarev, Vsevolod Plokhotnyuk, Denis Abramov, Elisey Zhdanov, Sofia Dorogova, Yasin Mamedov, Nikola Banic, Georgii Perevozchikov, Radu Timofte, et al. NTIRE 2025 challenge on night photography rendering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[19] Yuqian Fu, Xingyu Qiu, Bin Ren Yanwei Fu, Radu Timofte, Nicu Sebe, Ming-Hsuan Yang, Luc Van Gool, et al. NTIRE 2025 challenge on cross-domain few-shot object detection: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[20] Shuhang Gu and Radu Timofte. A brief review of image denoising algorithms and beyond. *Inpainting and Denoising Challenges*, pages 1–21, 2019. 1

[21] Hang Guo, Yong Guo, Yaohua Zha, Yulun Zhang, Wenbo Li, Tao Dai, Shu-Tao Xia, and Yawei Li. Mambairv2: Attentive state space restoration. *arXiv preprint arXiv:2411.15269*, 2024. 4, 8

[22] Shuhao Han, Haotian Fan, Fangyuan Kong, Wenjie Liao, Chunle Guo, Chongyi Li, Radu Timofte, et al. NTIRE 2025 challenge on text to image generation model quality assessment. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[23] Varun Jain, Zongwei Wu, Quan Zou, Louis Florentin, Henrik Turbell, Sandeep Siddhartha, Radu Timofte, et al. NTIRE 2025 challenge on video quality enhancement for video conferencing: Datasets, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[24] Amogh Joshi, Nikhil Akalwadi, Chinmayee Mandi, Chaitra Desai, Ramesh Ashok Tabib, Ujwala Patil, and Uma Mudenagudi. Hnn: Hierarchical noise-deinterlace net towards image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3007–3016, 2024. 11

[25] Cansu Korkmaz and A Murat Tekalp. Training transformer models by wavelet losses improves quantitative and visual performance in single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6661–6670, 2024. 3, 4

[26] Edwin H Land and John J McCann. Lightness and retinex theory. *Journal of the Optical society of America*, 61(1):1–11, 1971. 19

[27] Sangmin Lee, Eunpil Park, Angel Canelo, Hyunhee Park, Youngjo Kim, Hyungju Chun, Xin Jin, Chongyi Li, Chun-Le Guo, Radu Timofte, et al. NTIRE 2025 challenge on efficient burst hdr and restoration: Datasets, methods, and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[28] Xin Li, Yeying Jin, Xin Jin, Zongwei Wu, Bingchen Li, Yufei Wang, Wenhan Yang, Yu Li, Zhibo Chen, Bihan Wen, Robby Tan, Radu Timofte, et al. NTIRE 2025 challenge on day and night raindrop removal for dual-focused images: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[29] Xin Li, Xijun Wang, Bingchen Li, Kun Yuan, Yizhen Shao, Suhang Yao, Ming Sun, Chao Zhou, Radu Timofte, and Zhibo Chen. NTIRE 2025 challenge on short-form ugc video quality assessment and enhancement: Kwaisr dataset and study. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[30] Xin Li, Kun Yuan, Bingchen Li, Fengbin Guan, Yizhen Shao, Zihao Yu, Xijun Wang, Yiting Lu, Wei Luo, Suhang Yao, Ming Sun, Chao Zhou, Zhibo Chen, Radu Timofte, et al. NTIRE 2025 challenge on short-form ugc video quality assessment and enhancement: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[31] Yawei Li, Kai Zhang, Jingyun Liang, Jiezhang Cao, Ce Liu, Rui Gong, Yulun Zhang, Hao Tang, Yun Liu, Denis Demandolx, et al. Lsdir: A large scale dataset for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2023. 2, 5, 8, 11, 14

[32] Yawei Li, Yulun Zhang, Radu Timofte, Luc Van Gool, Zhijun Tu, Kunpeng Du, Hailing Wang, Hanting Chen, Wei Li, Xiaofei Wang, et al. Ntire 2023 challenge on image denoising: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1905–1921, 2023. 3

[33] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 20

[34] Jie Liang, Radu Timofte, Qiaosi Yi, Zhengqiang Zhang, Shuaizheng Liu, Lingchen Sun, Rongyuan Wu, Xindong Zhang, Hui Zeng, Lei Zhang, et al. NTIRE 2025 the 2nd restore any image model (RAIM) in the wild challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[35] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017. 7, 11

[36] Jingbo Lin, Zhilu Zhang, Yuxiang Wei, Dongwei Ren, Dongsheng Jiang, Qi Tian, and Wangmeng Zuo. Improving image restoration through removing degradations in textual representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2866–2878, 2024. 5

[37] Xiaohong Liu, Xiongkuo Min, Qiang Hu, Xiaoyun Zhang, Jie Guo, et al. NTIRE 2025 XGC quality assessment challenge: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[38] Xiaoning Liu, Zongwei Wu, Florin-Alexandru Vasluianu, Hailong Yan, Bin Ren, Yulun Zhang, Shuhang Gu, Le Zhang, Ce Zhu, Radu Timofte, et al. NTIRE 2025 challenge on low light image enhancement: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[39] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[40] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Refusion: Enabling large-size realistic image restoration with latent-space diffusion models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 1680–1691, 2023. 21

[41] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *IEEE International Conference on Computer Vision (ICCV)*, pages 416–423, 2001. 19

[42] Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. *Advances in Neural Information Processing Systems*, 36:71275–71293, 2023. 8

[43] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 4

[44] Bin Ren, Hang Guo, Lei Sun, Zongwei Wu, Radu Timofte, Yawei Li, et al. The tenth NTIRE 2025 efficient super-resolution challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[45] Nickolay Safonov, Alexey Bryntsev, Andrey Moskalenko, Dmitry Kulikov, Dmitriy Vatolin, Radu Timofte, et al. NTIRE 2025 challenge on UGC video enhancement: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[46] SMA Sharif, Abdur Rehman, Zain Ul Abidin, Rizwan Ali Naqvi, Fayaz Ali Dharejo, and Radu Timofte. Illuminating darkness: Enhancing real-world low-light scenes with smartphone images. *arXiv preprint arXiv:2503.06898*, 2025. 19

[47] H. R. Sheikh, M. F. Sabir, and A. C. Bovik. Live image quality assessment database release 2. `http://live.ece.utexas.edu/research/quality/`, 2006. 19

[48] Lei Sun, Andrea Alfarano, Peiqi Duan, Shaolin Su, Kaiwei Wang, Boxin Shi, Radu Timofte, Danda Pani Paudel, Luc Van Gool, et al. NTIRE 2025 challenge on event-based image deblurring: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[49] Lei Sun, Hang Guo, Bin Ren, Luc Van Gool, Radu Timofte, Yawei Li, et al. The tenth ntire 2025 image denoising challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[50] Radu Timofte, Rasmus Rothe, and Luc Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1865–1873, 2016. 8

[51] Jiachen Tu, Yaokun Shi, and Fan Lam. Score-based self-supervised MRI denoising. In *The Thirteenth International Conference on Learning Representations*, 2025. 9, 10

[52] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. 11

[53] Florin-Alexandru Vasluianu, Tim Seizinger, Zhuyun Zhou, Cailian Chen, Zongwei Wu, Radu Timofte, et al. NTIRE 2025 image shadow removal challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[54] Florin-Alexandru Vasluianu, Tim Seizinger, Zhuyun Zhou, Zongwei Wu, Radu Timofte, et al. NTIRE 2025 ambient lighting normalization challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[55] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1905–1914, 2021. 8

[56] Yingqian Wang, Zhengyu Liang, Fengyuan Zhang, Lvli Tian, Longguang Wang, Juncheng Li, Jungang Yang, Radu Timofte, Yulan Guo, et al. NTIRE 2025 challenge on light field image super-resolution: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[57] Kangning Yang, Jie Cai, Ling Ouyang, Florin-Alexandru Vasluianu, Radu Timofte, Jiaming Ding, Huiming Sun, Lan Fu, Jinlong Li, Chiu Man Ho, Zibo Meng, et al. NTIRE 2025 challenge on single image reflection removal in the wild: Datasets, methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[58] Pierluigi Zama Ramirez, Fabio Tosi, Luigi Di Stefano, Radu Timofte, Alex Costanzino, Matteo Poggi, Samuele Salti, Stefano Mattoccia, et al. NTIRE 2025 challenge on hr depth from images of specular and transparent surfaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2025. 2

[59] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 3, 4, 5, 6, 7, 8, 10, 20

[60] Jiale Zhang, Yulun Zhang, Jinjin Gu, Jiahua Dong, Linghe Kong, and Xiaokang Yang. Xformer: Hybrid x-shaped transformer for image denoising. *arXiv preprint arXiv:2303.06440*, 2023. 4, 12, 20

[61] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 1

[62] Kai Zhang, Yawei Li, Jingyun Liang, Jiezhang Cao, Yulun Zhang, Hao Tang, Deng-Ping Fan, Radu Timofte, and Luc Van Gool. Practical blind image denoising via swin-conv-unet and data synthesis. *Machine Intelligence Research*, 20(6):822–836, 2023. 8, 12

[63] Yulun Zhang, Kunpeng Li, Kai Li, Lichen Wang, Bineng Zhong, and Yun Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018. 14