NTIRE 2025 the 2nd Restore Any Image Model (RAIM) in the Wild Challenge: Supplementary Material

Jie Liang Radu Timofte Qiaosi Yi Zhengqiang Zhang Shuaizheng Liu
Lingchen Sun Rongyuan Wu Xindong Zhang Hui Zeng Lei Zhang Tianyu Hao
Lin Wang Zhe Xiao Pengzhou Ji Shu-Peng Zhong Xiangming Wang
Xiangming Wang Shu-Peng Zhong Jiaqi Yan Sishun Pan Ce Wang
Yibin Huang ZhanSheng Wang Haobo Liang Zhenghao Pan Jinjian Wu
Yushen Zuo Yuanbo Zhou

1. Teams and Methods

This section briefly describes the participating teams and their proposed methods for the two tracks. We only provide the methods of the top six teams of each track.

1.1. Track 1

1.1.1 Team MiAlgo

Team MiAlgo proposed a two-stage raw image restoration pipeline. Their method integrates transformer-based and GAN-based models for joint denoising, demosaicking, and detail enhancement, with robustness to noise, exposure, and sensor defects.

Architecture. As shown in Fig. 1, the proposed method consists of a two-stage end-to-end pipeline. Stage 1 uses Restormer to jointly perform denoising and demosaicking. Its transformer-based self-attention mechanism enables modeling of long-range dependencies and better preservation of semantic features. Stage 2 applies a GAN model for texture enhancement, where the discriminator follows the Real-ESRGAN [21] design. The generator in this stage is initialized with Stage 1 parameters to boost convergence and performance. In Phase 3, the team compressed and distilled their Phase 2 model using a lightweight UNet architecture enhanced with basic transformer blocks and two Restormer modules in the bottleneck. MWRCAN [11] is used as the Stage 2 generator to enable efficient multi-scale restoration.

Data Augmentation. The team used both the official paired dataset and an internal ultra-high-resolution dataset (4K–6K) consisting of 1200 images, including 1000 general scenes and 200 night portraits. The data degradation pipeline includes gamma correction, AWB gain removal, CCM adjustment, blurring, noise, downsampling, and ISO-based noise augmentation using darkening and dgain transformation. To simulate sensor defects, random Bayer pattern defect augmentation was also applied.

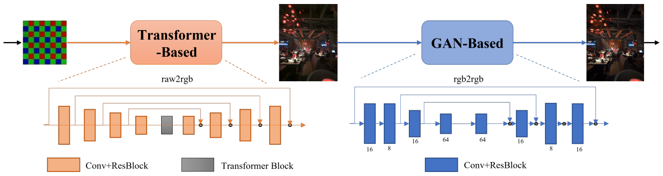


Figure 1. The two-stage pipeline proposed by team MiAlgo.

Training Details. The training was conducted in two stages. Stage 1 was trained with $\mathcal{L}_1$ loss for 300K iterations at a resolution of 256×256, followed by 100K iterations at 512×512 using a loss function composed of L2 + 0.1 Perceptual + 0.1 LPIPS + 0.1 SSIM. Stage 2 was trained with a composite loss of L2 + 0.1 Perceptual + 0.01 GAN + 4 LPIPS, using a learning rate of 1×10^{-5}. The models were implemented in PyTorch and trained on 8 NVIDIA A100 GPUs.

Performance and Efficiency. The method achieved the highest score in Phase 2 with a PSNR-based score of 92.56. The Phase 2 model has 52.24M parameters and 5351.47 GFLOPs (for 1024×1024 input), with an average inference time of 2651 ms per image. The Phase 3 model is significantly lighter, with 4.62M parameters and 372.09 GFLOPs, and runs at 320 ms per image (3072×4096 resolution) on A100.

Robustness and Generality. The proposed solution demonstrates strong robustness in real-world scenarios, especially in the presence of extreme exposure and Bayer pattern defects. The lightweight design of Phase 3 ensures practical deployment with minimal compromise in performance.

Additional Information. No external or pre-trained models were used. No ensemble or fusion strategies were applied. The method is novel and has not been published.

The team does not plan to submit a paper for NTIRE 2025. They suggest releasing the evaluation metrics earlier in future competitions for better tuning and transparency.

1.1.2 Team IID-AI

Team IID-AI employed a data-centric strategy and adopted XRestormer [8] as the backbone network to address the domain gap between training and testing data. The model is trained with a carefully designed multi-loss function to achieve high-quality raw image enhancement.

Data Synthesis. Initial experiments trained on Phase 1 paired data showed strong performance on Phase 2 paired testing, but significant degradation on Phase 3 unpaired data. The root cause was identified as a domain shift: DSLR-captured (16-bit, daytime) training data versus smartphone-captured (10-bit, nighttime) testing data. To mitigate this gap, over 700 clean RAW images were captured using smartphones under low-ISO and short-exposure conditions to simulate noise-free ground truth. A degradation pipeline was then applied, including demosaicing, Gaussian blurring (based on PSF parameters), downsampling, and synthetic noise generation using ISO 1600, 3200, and 6400-level noise. GT sRGB images were generated using AAHD demosaicing for improved edge sharpness and texture fidelity. This pipeline effectively supports joint denoising and demosaicking.

Network Design. XRestormer [8] was selected as the backbone for its proven effectiveness in low-level vision tasks and efficient computation. To adapt it to the raw-to-RGB task, the final convolutional layer was modified to output 12 channels, followed by a pixelshuffle layer for resolution doubling and RGB conversion. The input–output skip connection was removed due to channel mismatch.

Loss Functions. A combination of L1 reconstruction loss [16], perceptual loss [12, 28], and FFT-based frequency loss [15] was used. While reconstruction and perceptual losses ensured good quantitative and perceptual quality, checkerboard artifacts were observed in dark regions of Phase 3 test images. This was addressed by adding a frequency-domain loss term. The overall training loss consists of an L1 loss, an FFT loss weighted by 300, and an LPIPS loss weighted by 0.3.

Training Details. The model was implemented using PyTorch and trained on an NVIDIA RTX 4090 GPU. The optimizer was Adam, with a batch size of 2 and a patch size of 512. Training was conducted for 70K iterations with a learning rate of 1×10^{-4}, followed by 30K iterations at 1×10^{-5}.

1.1.3 Team PolyU-AISP

Team PolyU-AISP proposed a lightweight restoration network based on the NAFNet [6] architecture. The method adopts a UNet-like design with layer normalization, convolution, and channel attention in each block. To handle high-resolution inputs, the model processes images in overlapping tiles and fuses them at the output, achieving both computational efficiency and restoration quality.

Architecture. As illustrated in Fig. 2, the proposed pipeline is a compact UNet-style network built upon NAFNet. Each block integrates layer normalization, 1×1 convolution, depthwise convolution, and channel attention. The input image is divided into 2048×2048 overlapping tiles with 64-pixel padding. These tiles are processed independently and then fused using weighted averaging to ensure smooth transitions across tile boundaries. The final model contains 29M parameters, offering a good trade-off between performance and model size.

Training Details. The model is trained on the SIDD dataset [1], following the protocol in HiNet [4]. The training patch size is 256×256, and the network is optimized with the Adam optimizer [14]. The learning rate starts from 10^{-3} and decays to 10^{-7} using cosine annealing over 200K iterations. A tile width of 32 is used during training.

Testing and Inference. During inference, the input is tiled into 2048×2048 overlapping patches with 64 pixels of overlap. Each tile is processed through the network, and the outputs are fused via weighted averaging. The average runtime per image is approximately 2.8 seconds on an NVIDIA RTX 4090 GPU. The method requires only 3K GFLOPs to process a full 4096×3072 image, demonstrating high computational efficiency.

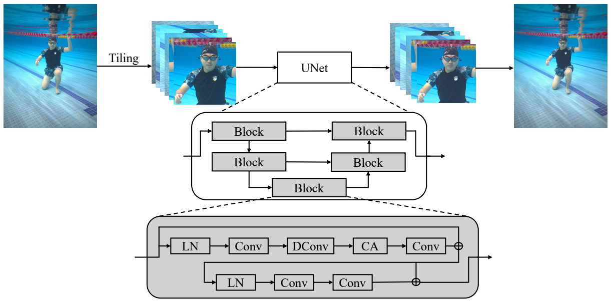
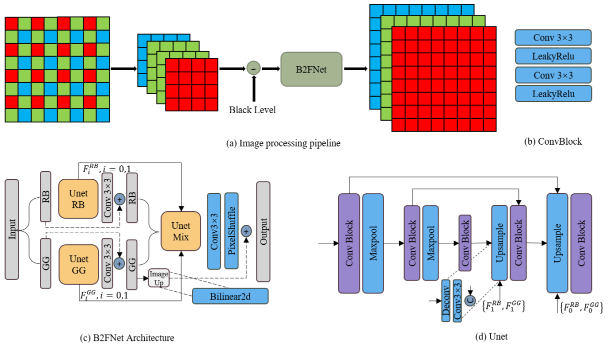


Figure 2. Pipeline of Team POLYU-AISP. LN: Layer Normalization, Conv: 1×1 Convolution, DConv: Depthwise Convolution, CA: Channel Attention (as in NAFNet [6]). Certain details, such as activation functions and element-wise operations, are omitted for clarity.

1.1.4 Team TongJi-IPOE

Team TongJi-IPOE proposed a lightweight and efficient joint raw image denoising and demosaicing solution, named B2FNet (Branching to Fusion Network), designed for low-light image processing. Inspired by the low-light pipeline from Chen et al. [2], the method explicitly separates and



(a) Image processing pipeline

(b) ConvBlock

(c) B2FNet Architecture

(d) Unet

Figure 3. Overview of the Team TongJi-IPOE.

processes the green and red/blue channels in the Bayer pattern, followed by a fusion stage to generate the final sRGB image.

Architecture. As shown in Fig. 3, B2FNet is a three-stage UNet-based structure. The input raw image is first converted to a four-channel RGGB image via demosaicing. Two lightweight UNets are deployed to restore the GG and RB channels separately. Their outputs are then fused by another UNet to generate the final sRGB output. Each UNet consists only of 3 × 3 convolutional layers, max-pooling for downsampling, and deconvolution for upsampling. To avoid gradient vanishing, feature maps from the encoder stages of the two-branch UNets are introduced into the decoder of the fusion UNet.

Processing Pipeline. Given a raw input of size $\mathbb{R}^{H\times W\times 1}$, it is first demosaiced into a four-channel RGGB image $\mathbb{R}^{\frac{H}{2}\times\frac{W}{2}\times 4}$. This is followed by dark-level correction and processed by B2FNet to generate a final sRGB image of size $\mathbb{R}^{H\times W\times 3}$.

Training Details. The model is trained solely on the provided dataset, using PyTorch on a single NVIDIA RTX 3090Ti GPU. The optimizer is AdamW with $\beta_1 = 0.9$, $\beta_2 = 0.999$. The training is conducted in three stages: 92K iterations with a fixed learning rate of 1×10^{-4}, followed by 208K iterations using cosine annealing to decay the learning rate to 1×10^{-6}, and finally 300K iterations of fine-tuning at 1×10^{-5}. The training patch size is 128 × 128.

Testing and Efficiency. At inference, raw inputs are first demosaiced to RGGB format, corrected for dark levels, and passed through B2FNet for sRGB conversion. The model contains only 0.39M parameters and requires 23.45 GFLOPs for an input size of 1024 × 1024. Average inference time is 8.3 ms on an NVIDIA A100 GPU.

Training Details. The method is implemented in PyTorch and runs on a single RTX 3090Ti GPU. The pipeline is simple and compact, and suitable for real-time deployment. The full training and fine-tuning process took approximately 24 hours. Given its low complexity, the method holds potential for edge deployment with further optimization.

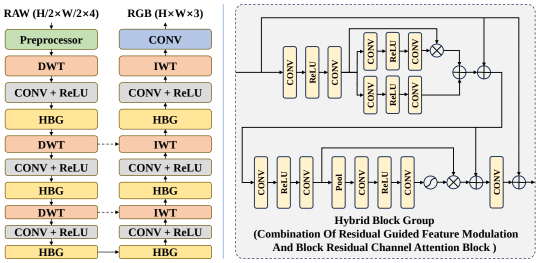


Figure 4. Frame diagram of our method.

1.1.5 Team NJUST-KMG

Team NJUST-KMG proposed a novel restoration framework for low-light raw image denoising and demosaicing, named DWT-Enhanced Hybrid Networks. The approach combines a hybrid attention mechanism with multi-scale feature processing via discrete wavelet transforms (DWT), and is trained via a two-stage pipeline to jointly optimize noise removal and detail preservation.

Architecture. The model employs a U-shaped encoder-decoder structure enhanced with HybridBlockGroups, which alternate between Residual Guided Feature Modulation (ResGFM) blocks and Residual Channel Attention (RCA) blocks. These modules are designed to balance spatial detail preservation and channel-wise feature enhancement. Additionally, DWTForward and DWTInverse modules are incorporated to perform multi-scale feature extraction and reconstruction using discrete wavelet transforms. The architecture is illustrated in Fig. 4.

Training Strategy. The model is trained in two phases using a custom PyTorch framework that supports distributed training and mixed precision for improved efficiency. The first phase focuses on robustness, using heavy data augmentation to train the model against severe low-light noise and artifacts. Augmentations include ISO-based noise modeling (simulating shot and read noise across ISO levels from 3200 to 9600), adaptive Gaussian blur (kernel sizes 3–7 and sigma 0.6–1.2), color channel perturbation (random scaling in [0.8, 1.2]), and dynamic noise amplification (factors between 20 and 30). In the second phase, the model is fine-tuned with lighter augmentations to better preserve texture and fine details while maintaining denoising performance. The dataset used includes 50 images from SIDD [1], 90 images from SID [2], and 20 additional images collected from online sources. The training loss is a composite of L1 loss, SSIM loss, and perceptual losses (LPIPS and DISTS), which together balance pixel accuracy, structural similarity, and perceptual fidelity. The entire training process takes approximately 4 hours on a single NVIDIA RTX 3090 GPU.

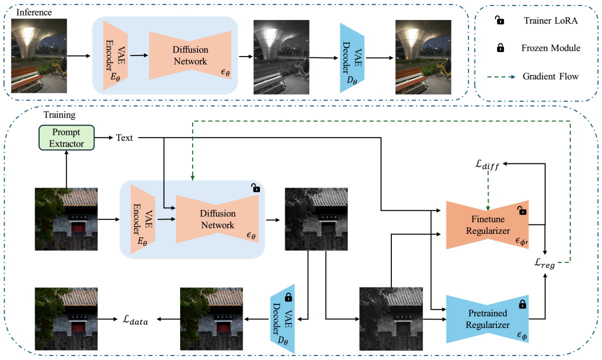


Figure 5. Frame diagram of Team xianggkl.

The lightweight design of the network (only 0.83M parameters) allows for efficient training and deployment, requiring minimal human supervision.

1.1.6 Team xianggkl

Team xianggkl proposed a diffusion-based framework for joint raw image denoising and demosaicing, named Enhanced Degradation Adaptation Diffusion. The method leverages a trainable VAE encoder E_θ, a LoRA fine-tuned two-step diffusion model ϵ_θ, and a frozen VAE decoder D_θ. To guide the diffusion process, text prompts extracted from low-quality images are used as conditioning inputs. These prompts help the model adaptively generate high-quality images by aligning the output distribution with that of natural, clean images using Variational Score Distillation (VSD).

Architecture and Inference. As shown in the Fig. 5, the proposed model is based on a diffusion framework enhanced with a trainable VAE encoder E_θ, a LoRA fine-tuned two-step diffusion network ϵ_θ, and a frozen VAE decoder D_θ. During training, the diffusion output is regularized by two networks—one frozen and one fine-tuned—via VSD in the latent space. The overall objective function combines a data loss $\mathcal{L}_{\text{data}}$ (comprising MSE and LPIPS losses) and a regularization loss $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{data}} + \lambda_2 \mathcal{L}_{\text{reg}}$, where λ_2 controls the trade-off between fidelity and distribution alignment. At inference, only E_θ, ϵ_θ, and D_θ are used. The prompt extractor and CLIP encoder are removed and replaced with a fixed empty-string embedding. The decoder operates in a fast VAE decoding mode, achieving 37.48 seconds per image of size 3072 × 4096.

Training Strategy and Efficiency. Training is conducted in three stages to progressively improve performance and robustness. In Step 1, the model is trained on 256 × 256 crops using four RTX 4090 GPUs (batch size 4) for 60K steps. Step 2 continues training with noise-enhanced data (high ISO noise) for 20K steps on two RTX 4090 GPUs (batch size 2). Step 3 fine-tunes the model on 512 × 512 crops using two A100 GPUs for 50K steps. Despite the large model size and complexity (2124.36 GFLOPs for 512 × 512 input), the system is optimized for deployment with fast inference and minimal runtime overhead.

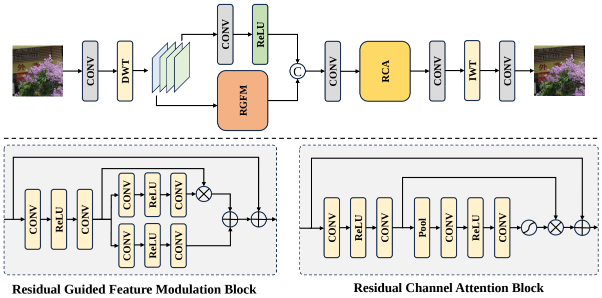


Figure 6. The overall pipeline of the solution proposed by team NulltoZero.

Novelty and Generalization. The proposed method builds upon OSEDiff [23] and incorporates degradation-aware prompt extraction (DAPE) from SeeSR [24]. Notable innovations include noise-adaptive training, CLIP-free inference via fixed prompts, progressive training stages, and the use of VSD for distribution alignment. The model generalizes well across varying lighting and noise conditions and can be adapted to other restoration and degradation tasks. To the best of the authors' knowledge, this solution has not been previously published.

Technical Implementation. The system is implemented in Python using PyTorch and trained on 4× RTX 4090 and 2× A100 GPUs. Each experiment takes approximately 2 days. Validation includes both quantitative metrics (PSNR, SSIM, LPIPS) and qualitative visual inspection. Hyperparameters such as latent tile size are tuned through multiple test runs to optimize output quality.

1.2. Track 2

1.2.1 Team NulltoZero

Team NulltoZero introduces WaveletFusionNet, a deep learning architecture for image detail enhancement, specifically designed for competitive image enhancement tasks.

Architecture.As shown in Fig. 6, the proposed model introduces WaveletFusionNet, a novel deep learning architecture tailored for high-resolution image detail enhancement tasks. The model adopts a multi-stage U-shaped framework built upon multilevel discrete wavelet transforms (DWT)to decompose the input into low-frequency and high-frequency components. This decomposition facilitates effective noise suppression and detail recovery.

To extract and enhance features at different scales, the model employs Hybrid Feature Extraction, Multi-Scale Processing, and Fusion and Reconstruction.

- **Hybrid Feature Extraction Modules**. The network utilizes Residual Channel Attention Groups (RCAGroup) to refine global features and a novel Residual Guided Feature Modulation Block (ResGFMBlock) to adaptively modulate spatial features via a Spatial Feature Transform module.
- **Multi-Scale Processing**. The network utilizes Residual Channel Attention Groups (RCAGroup) to refine global features and a novel Residual Guided Feature Modulation Block (ResGFMBlock) to adaptively modulate spatial features via a Spatial Feature Transform module.
- **Fusion and Reconstruction**. The enhanced low- and high-frequency features are fused using a 1×1 convolution followed by further refinement with attention-based modules and residual connections.

The overall design results in an end-to-end system capable of enhancing image details and suppressing artifacts, making it particularly effective for high-resolution image enhancement tasks. The model has approximately 0.303MB of parameters, which have achieved excellent results in image processing efficiency. The training time of the model can be controlled within 4 hours. When the input size is $1024\times1024\times3$, the reasoning time is about 22.58ms per image on a computer with an RTX3090 GPU.

Data Augmentation. Three main steps are involved. First, the target image is read from a specified path, and a corresponding input image path is generated, and the existence of the input image is verified. Then, a random degradation strategy is applied. If the input image exists and a random probability threshold is met, or when not in training mode, the input image is used directly. Otherwise, a random degradation method is applied, either Gaussian blur or bilateral filtering, which is selected randomly. For Gaussian blur, parameters such as kernel size and sigma are randomly sampled, while for bilateral filtering, sigmaColor, sigmaSpace, and the kernel size are chosen randomly. Lastly, synchronized transformation is performed by generating a random seed to ensure that both the degraded input image and target image undergo identical transformations.

Training Details. The training process using PyTorch starts with loading the training and validation datasets using the DataLoader. The data sets are split based on a set validation ratio. The WaveletFusionNet model is built and trained using the AdamW optimizer. A MultiStepLR scheduler is used to adjust the learning rate during training. The loss function combines $\mathcal{L}_1$ loss, SSIM loss, and a custom NTIRE score loss based on LPIPS and DISTS to support multi-objective optimization. During the validation process, performance is measured using PSNR, SSIM, LPIPS, DISTS, and NIQE. The training loop includes repeated training and validation steps. The checkpoints are saved regularly and the best model is selected on the basis of the validation results. Every five epochs, a visualization is created by randomly selecting an image from the validation set to track the progress of the model.

1.2.2 Team TeleAI-Vision

Team TeleAI-Vision adopts the HAT [7] architecture to perform the image restoration task, benefiting from its strong representation capacity and competitive performance.

Detailed Method Description. In their proposed approach, the team adopts a multi-phase strategy to address the image restoration task. In Phase 2, they utilize the HAT-GAN architecture, leveraging its strong representation capacity and competitive performance to perform high-quality image restoration. To further enhance the model's efficiency, knowledge distillation is applied by using HAT-S as the student network, thereby reducing the model size while preserving performance.

Recognizing the increasing importance of computational efficiency and the demand for deployment-friendly models, the team introduces a more lightweight solution in Phase 3. Specifically, they adopt the Swift Parameter-free Attention Network (SPAN) [19] as the backbone of their final model. In their implementation, the number of feature channels is set to 52 and the upscale factor is fixed at 1, in accordance with the specific requirements of the restoration task.

The training process is carried out in three distinct stages. Unlike the Phase 2 model, which relies solely on the provided paired data, the team begins by pretraining the model using the high-quality and diverse LSDIR dataset [17]. This dataset offers a broad range of scenes, enabling the model to acquire a strong and generalizable representation. During the first two stages, the model is trained on the LSDIR training subset with synthetic degradations, allowing it to learn a robust prior. In the third and final stage, the model is fine-tuned using real paired data provided by RAIM 2024 [18] and RAIM 2025. This step is crucial for adapting the model to the target domain and improving its performance under real-world degradation conditions.

Overall, the experimental results demonstrate the effectiveness of the proposed method and underscore the value of carefully curated training data. By progressively leveraging both synthetic datasets and real paired samples, the model achieves strong generalization capabilities while maintaining high adaptability to real-world degradations.

Training Details. In the phase 2 of their approach, the team trains the HAT model using only the real paired datasets provided by RAIM 2024 [18] and RAIM 2025. To enhance efficiency, knowledge distillation is applied with HAT-S serving as the student network.

In the third phase, with a focus on computational efficiency, the team adopts the Swift Parameter-free Attention Network (SPAN) as the backbone of the model. The complete training process is conducted on 8 GPUs with a batch size of 12. During the first two training stages, the model is trained on the LSDIR dataset [17], which includes synthetically degraded images. Before training, LSDIR images are cropped into patches of size 240 $\times$ 240 with a stride of 120. During training, random patches of size 128 $\times$ 128 are further sampled as model inputs to improve generalization.

The degradation pipeline is based on Real-ESRGAN [20], with additional degradation hyperparameters to introduce more variability. The degradation settings are as follows:

- Gaussian noise probability: 0.5
- Noise range: [0.5, 1]
- Poisson scale range: [0.05, 0.1]
- Gray noise probability: 0.4
- Second blur probability: 0.8
- Gaussian noise probability (second): 0.5
- Noise range (second): [0.5, 1]
- Poisson scale range (second): [0.05, 0.1]
- Gray noise probability (second): 0.4

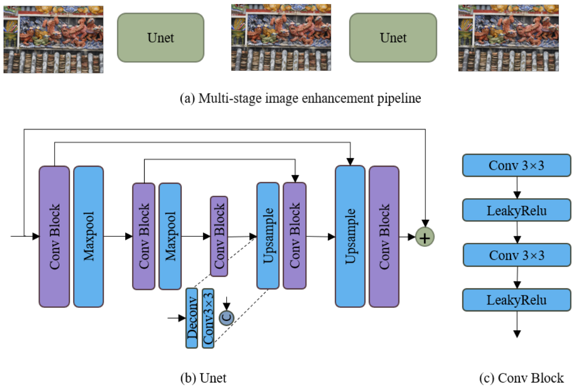
The model is trained in three stages:

- **Stage 1** The model is trained using only the $\mathcal{L}_1$ loss on the LSDIR dataset. This stage runs for approximately 330000 iterations with a learning rate of 1×10^{-4}.
- **Stage 2** The Real-SPAN model is trained using a composite loss function to improve perceptual quality and robustness. The loss function is defined as $\mathcal{L} = \mathcal{L}_{\text{L}_1} + 0.1 \times \mathcal{L}_{\text{Perceptual}} + 4 \times \mathcal{L}_{\text{LPIPS}} + 0.1 \times \mathcal{L}_{\text{GAN}}$. This stage is trained for approximately 400000 iterations with the same
- **Stage 3**. The model is fine-tuned using the real paired data from RAIM 2024 and RAIM 2025. Before training, the RAIM images are cropped into patches of size 480 $\times$ 480 with a stride of 240. The same composite loss function from Stage 2 is used. Fine-tuning is performed for 10000 iterations with a reduced learning rate of 1×10^{-5} to ensure stable convergence.

1.2.3 Team TongJi-IPOE

Team TongJi-IPOE proposed a lightweight method for efficient image detail enhancement/generation on RGB images, named MSUnet (from branching to fusion network, as shown in Fig. 7, a multi-stage image restoration network.

Architecture. Recent research has demonstrated that multi-stage network architectures are highly effective for various image restoration tasks. Compared to directly increasing the number of channels in convolutional layers, cascading multiple networks provides a more lightweight and computationally efficient design [25]. Based on this motivation, the team proposes a multi-stage architecture named MSUnet, as illustrated in Fig. 7 (a). The MSUnet consists of two cascaded U-Net sub-networks. The first U-Net is responsible for performing a preliminary enhancement on the input image, while the second U-Net is designed to further refine the image details and improve the final restoration quality. Both sub-networks adopt a simple yet effective design, consisting solely of 3 $\times$ 3 convolutional layers, max-pooling layers for downsampling, and deconvolution layers for upsampling. To achieve better computational efficiency, the team also introduces a lightweight backbone named **B2FNet**, which serves as the core component of the MSUnet. The B2FNet contains approximately 0.26 M parameters and requires 54.96 GFLOPs to process an image of resolution 1024 $\times$ 1024. The average inference time of the network is 15 ms on a single NVIDIA A100 GPU, demonstrating its suitability for real-time applications and deployment on resource-constrained platforms.



(a) Multi-stage image enhancement pipeline

(b) Unet

(c) Conv Block

Figure 7. Overview of the MSUnet proposed by Team TongJi-IPOE.

Training Details. The entire training process is conducted using a single NVIDIA GeForce RTX 3090Ti GPU, and the implementation is based on the PyTorch framework. The team employs the AdamW optimizer to train the proposed network, with the hyperparameters set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The training is carried out in two stages. In the first stage, the model is trained for 92000 iterations using a fixed learning rate of 1×10^{-4}. In the second stage, the model undergoes further training for 208000 iterations. During this stage, the learning rate is gradually decreased following a cosine annealing schedule, with a minimum learning rate of 1×10^{-6}. Throughout both training stages, only the official dataset provided by the RAIM organizers is used. The training inputs are cropped into patches of size 128 $\times$ 128 to balance memory usage and training efficiency.

1.2.4 Team NJUST-KMG

Method Description. The team optimizes the MW-ISPNet architecture [11] to develop a lightweight model suitable for real-world applications, particularly in resource-constrained environments. Specifically, the number of channels in the downsampling stages is adjusted to 16, 32, and 32, while only a single intermediate layer is retained to simplify the network structure. Correspondingly, the upsampling stages are configured with 32, 32, and 16 channels. To enhance the model's ability to capture fine-grained image details and textures, several auxiliary loss functions are introduced in addition to the standard $\mathcal{L}_1$ loss. The SSIM loss [22] is employed to preserve structural consistency, with a loss weight of 0.15. Furthermore, LPIPS [27] and DISTS [9] losses are incorporated to improve perceptual and texture-level fidelity, both with weights set to 1.0. The combination of these loss functions significantly improves the model's capacity to restore realistic and detailed image content.

Training and Testing Details. To improve generalization and robustness, data augmentation techniques are applied during training. These include randomly cropping the input images to a resolution of 1024×1024 pixels, as well as performing random horizontal and vertical flips to diversify the training samples. In the third phase of development, the team observed that training exclusively with the LR and GT image pairs provided by the organizers often resulted in the generation of severe false textures, which are undesirable in real-world applications. Analysis suggests that this issue arises because of the significant loss of detail and texture information in the LR images, which increases the learning difficulty for the network. To mitigate this problem, a data pre-processing strategy is proposed to generate LR images that better match the statistical distribution of real mobile phone photographs. As illustrated in Fig. 8, a slight boundary blur is applied to the GT images. To preserve potential noise in flat regions, the blurring is restricted to edge areas only. The process is as follows: the Canny edge detection algorithm is used to extract an edge mask from the GT image. Then, a Gaussian-blurred version of the GT image is multiplied by the edge mask and added to the original GT image weighted by the inverse mask $(1 - \text{mask})$, resulting in a synthesized LR image that preserves noise in flat areas while reducing sharp transitions at edges. This preprocessing strategy effectively reduces false texture artifacts and enhances the model's ability to generate high-quality images that align more closely with real-world visual characteristics.

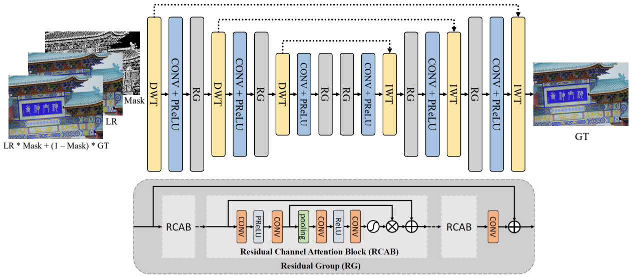


Figure 8. Overall training framework of the method proposed by the team NJUST-KMG.

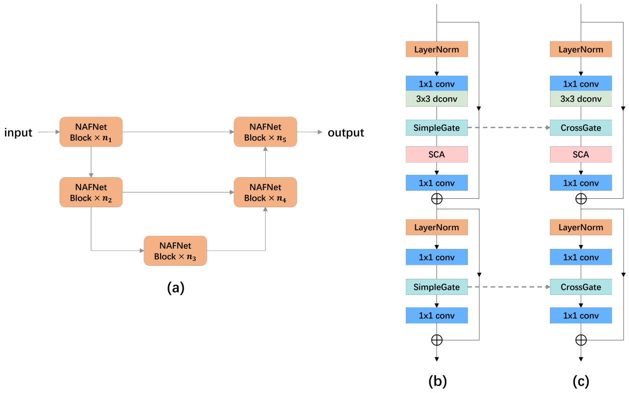


Figure 9. Overall framework proposed by iAM_IR. (a) represents the U-Net architecture of NAFNet, (b) represents the original NAFNet block, and (c) represents our improved NAFNet block.

1.2.5 Team iAM_IR

Team iAM_IR proposed a two-stage image restoration model (TSIRM) to address simulated degradation in phase 2 and real-world degradation in phase 3. In the first stage, the model is pre-trained using extensive simulated data. In the second stage, the model is fine-tuned using GT.

Architecture. Considering the efficiency and practical deployment requirements, the proposed model is built upon NAFNet [5], with several modifications to improve feature representation and model performance. As illustrated in Fig. 9, the model adopts a U-Net-like architecture composed of multi-scale NAFNet blocks. In the original NAFNet design, the GELU activation function is replaced with the computationally efficient SimpleGate operation, and the self-attention mechanism is substituted with a lightweight channel attention module. These changes significantly improve computational efficiency while preserving the benefits of transformer-based architectures. However, SimpleGate alone may result in limited feature interaction. To address this limitation, the team introduces an improved activation mechanism named CrossGate, as shown in Fig. 10. Unlike SimpleGate, the CrossGate module enables more effective fusion of complementary feature information without reducing channel dimensionality, thereby enhancing the model's representation capacity.

Two-Stage Training Strategy. Due to the limited availability of real paired training data, a two-stage training strategy is adopted. In the first stage, the model is pretrained using synthetic paired data generated from high-quality public datasets. Specifically, the team uses the DIV8K [10] dataset and 1,000 facial images from FFHQ [13]. To simulate realistic degradations, a simplified version of the BSRGAN [26] pipeline is adopted, in which JPEG compression is removed to reduce artifacts. The commonly used

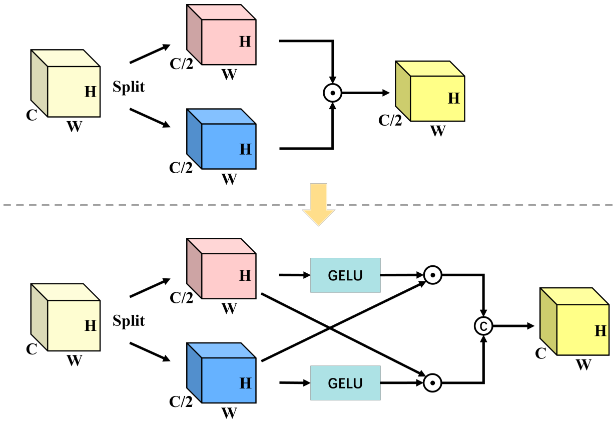


Figure 10. Structure comparison between SimpleGate and the CrossGate proposed by iAM_IR.

Real-ESRGAN [20] second-order degradation pipeline is avoided due to its tendency to introduce overly severe distortions. Pretraining is conducted in two steps: first, 20,000 iterations are trained using only $\mathcal{L}_1$ loss; then, perceptual loss and GAN loss are introduced, with the loss function defined as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + 1.0 \times \mathcal{L}_{\text{perceptual}} + 0.1 \times \mathcal{L}_{\text{GAN}},$$

and training continues for an additional 150000 iterations. In the second stage, fine-tuning is performed using available high-quality reference images. In Phase 2, where GT data is available, the team employs all reference-based evaluation metrics from the competition as supervisory objectives to directly optimize image quality and perceptual fidelity. In Phase 3, where no GT is provided, pseudo GT images are generated using FeMaSR [3] (×2 version), a state-of-the-art real image restoration model. The generated pairs are then used to fine-tune the network. To further enhance perceptual quality, the weights of the perceptual and GAN losses are increased to:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_1 + 1.5 \times \mathcal{L}_{\text{perceptual}} + 0.2 \times \mathcal{L}_{\text{GAN}}.$$

1.2.6 Team MiAlgo

Phase2. The MiAlgo team fine-tuned the model for approximately 20000 iterations using the loss function $\mathcal{L}_2 + 0.1 \cdot \mathcal{L}_{\text{perceptual}} + 0.01 \cdot \mathcal{L}_{\text{GAN}} + 4 \cdot \mathcal{L}_{\text{lpips}}$. The fine-tuning process was conducted within the RealESRGAN [20] training framework, employing a learning rate of 1e-5. For training data, the degraded data from the RAIM2024 [18] challenge was combined with the 50 paired data samples provided by RAIM2025, in a 50:50 ratio in the training filelist.

Phase3. As shown in Figure 11, the Phase 3 model is built upon the Phase 2 architecture, adopting a lightweight UNet structure enhanced with Haar wavelet-based downsampling and upsampling (×2). The design is inspired by the MWRCAN model [11], where each UNet block contains one ResBlock, with convolutional layers between ResBlocks and wavelet modules to adjust channel dimensions. Given the high quality of Phase 3 test data, only minor edits were necessary. To maintain fidelity to the input, a global residual connection was added into the network design to ensure the output remains closely aligned with the original image. This efficient configuration results in a model size of only 0.075 MB and 19.83 GFLOPs for $3 \times 1024 \times 1024$ inputs. On an A100 GPU, it processes this input in 5.95 ms and handles $3 \times 3040 \times 4032$ images in 58.90 ms.

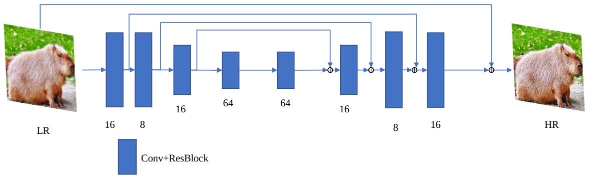


Figure 11. The tiny unet in phase3 applied by the team MiAlgo.

The training dataset consisted of two parts: one inherited from Phase 2, and another newly synthesized. The latter was motivated by the observation that 50% of the test images were high-quality female portraits. Approximately 5000 curated DSLR portrait of young women were curated and used as ground truth. These images were converted to RAW format and augmented with noise. Subsequently, the Track 1 Phase 2 model was employed to render these images into RGB format, serving as the low-quality images. Similarly, the GT used in phase 2 was also processed in the same manner to generate low-quality images.

Training began from scratch using the full dataset. The model was first optimized with $\mathcal{L}_2$ loss for 10k steps, followed by 800k steps using a composite loss: $\mathcal{L}_2 + 0.1 \cdot \mathcal{L}_{\text{perceptual}} + 0.01 \cdot \mathcal{L}_{\text{GAN}}$. A learning rate of $1e^{-5}$, batch size of 32, and input size of 512 were employed across 4 GPUs. To emphasize portrait quality, the dataset was rebalanced to include 80% portrait images. A learning rate of $1e^{-6}$ was used for an additional 40k steps. In the final stage, another 10k steps were trained with USM-enhanced ground truths and a reduced learning rate of $1e^{-7}$. Despite some training data being blurrier than the test set, such degradation was found to improve the clarity of non-portrait scenes.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1692–1700, 2018. 2, 3

[2] Chen Chen, Qifeng Chen, Jia Xu, and Vladlen Koltun.

Learning to see in the dark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3291–3300, 2018. 2, 3

[3] Chaofeng Chen, Xinyu Shi, Yipeng Qin, Xiaoming Li, Xiaoguang Han, Tao Yang, and Shihui Guo. Real-world blind super-resolution via feature matching with implicit high-resolution priors. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 1329–1338, 2022. 8

[4] Liangyu Chen, Xin Lu, Jie Zhang, Xiaojie Chu, and Chengpeng Chen. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 182–192, 2021. 2

[5] 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. 7

[6] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII*, pages 17–33. Springer, 2022. 2

[7] Xiangyu Chen, Xintao Wang, Jiantao Zhou, Yu Qiao, and Chao Dong. Activating more pixels in image super-resolution transformer. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 22367–22377, 2023. 5

[8] Xiangyu Chen, Zheyuan Li, Yuandong Pu, Yihao Liu, Jiantao Zhou, Yu Qiao, and Chao Dong. A comparative study of image restoration networks for general backbone network design. In *European Conference on Computer Vision*, pages 74–91. Springer, 2024. 2

[9] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *IEEE transactions on pattern analysis and machine intelligence*, 44(5):2567–2581, 2020. 7

[10] Shuhang Gu, Andreas Lugmayr, Martin Danelljan, Manuel Fritsche, Julien Lamour, and Radu Timofte. Div8k: Diverse 8k resolution image dataset. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3512–3516. IEEE, 2019. 7

[11] Andrey Ignatov, Radu Timofte, Zhilu Zhang, Ming Liu, Haolin Wang, Wangmeng Zuo, and et al. Aim 2020 challenge on learned image signal processing pipeline. In *Computer Vision–ECCV 2020 Workshops: Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 152–170, 2020. 1, 6, 8

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14*, pages 694–711. Springer, 2016. 2

[13] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4401–4410, 2019. 7

[14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2

[15] Patrick Krawczyk, Marvin Gaertner, Andreas Jansche, Timo Bernthaler, and Gerhard Schneider. Artifact generation when using perceptual loss for image deblurring. *Authorea Preprints*, 2023. 2

[16] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018. 2

[17] 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*, pages 1775–1787, 2023. 5, 6

[18] Jie Liang, Radu Timofte, Qiaosi Yi, Shuaizheng Liu, Lingchen Sun, Rongyuan Wu, Xindong Zhang, Hui Zeng, Lei Zhang, Yibin Huang, Shuai Liu, Yongqiang Li, Chaoyu Feng, Xiaotao Wang, Lei Lei, Yuxiang Chen, Xiangyu Chen, Qiubo Chen, Fengyu Sun, Mengying Cui, Jiaxu Chen, Zhenyu Hu, Jingyun Liu, Wenzhuo Ma, Ce Wang, Hanyou Zheng, Wanjie Sun, Zhenzhong Chen, Ziwei Luo, Fredrik K. Gustafsson, Zheng Zhao, Jens Sjölund, Thomas B. Schön, Xiong Dun, Pengzhou Ji, Yujie Xing, Xuquan Wang, Zhanshan Wang, Xinbin Cheng, Jun Xiao, Chenhang He, Xiuyuan Wang, Zhi-Song Liu, Zimeng Miao, Zhicun Yin, Ming Liu, Wangmeng Zuo, and Shuai Li. Ntire 2024 restore any image model (raim) in the wild challenge. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 6632–6640, 2024. 5, 8

[19] Cheng Wan, Hongyuan Yu, Zhiqi Li, Yihang Chen, Yajun Zou, Yuqing Liu, Xuanwu Yin, and Kunlong Zuo. Swift parameter-free attention network for efficient super-resolution. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 6246–6256, 2024. 5

[20] Xintao Wang, Liangbin Xie, Chao Dong, and Ying Shan. Real-esrgan: Training real-world blind super-resolution with pure synthetic data. In *International Conference on Computer Vision Workshops (ICCVW)*. 6, 8

[21] 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. 1

[22] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 7

[23] Rongyuan Wu, Lingchen Sun, Zhiyuan Ma, and Lei Zhang. One-step effective diffusion network for real-world image super-resolution. *Advances in Neural Information Processing Systems*, 37:92529–92553, 2024. 4

[24] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *Proceedings of*

the IEEE/CVF conference on computer vision and pattern recognition, pages 25456–25467, 2024. 4

[25] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 6

[26] Kai Zhang, Jingyun Liang, Luc Van Gool, and Radu Timofte. Designing a practical degradation model for deep blind image super-resolution. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4791–4800, 2021. 7

[27] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[28] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 2