MIPI 2024 Challenge on Nighttime Flare Removal: Methods and Results

Yuekun Dai[1] Dafeng Zhang[2] Xiaoming Li[1] Zongsheng Yue[1] Chongyi Li[3] Shangchen Zhou[1]
Ruicheng Feng[1] Peiqing Yang[1] Zhezhu Jin[2] Guanqun Liu[2] Chen Change Loy[1]

Abstract

The increasing demand for computational photography and imaging on mobile platforms has led to the widespread development and integration of advanced image sensors with novel algorithms in camera systems. However, the scarcity of high-quality data for research and the rare opportunity for in-depth exchange of views from industry and academia constrain the development of mobile intelligent photography and imaging (MIPI). Building on the achievements of the previous MIPI Workshops held at ECCV 2022 and CVPR 2023, we introduce our third MIPI challenge including three tracks focusing on novel image sensors and imaging algorithms. In this paper, we summarize and review the Nighttime Flare Removal track on MIPI 2024. In total, 170 participants were successfully registered, and 14 teams submitted results in the final testing phase. The developed solutions in this challenge achieved state-of-the-art performance on Nighttime Flare Removal. More details of this challenge and the link to the dataset can be found at https://mipi-challenge.org/MIPI2024.

1. Introduction

Lens flare, an optical phenomenon, arises when intense light scatters or reflects within a lens system, manifesting as a distinct radial-shaped bright area and light spots in captured photographs. In mobile platforms such as monitor lenses, smartphone cameras, UAVs, and autonomous driving cameras, daily wear and tear, fingerprints, and dust can function as a grating, exacerbating lens flare and making it particularly noticeable at night. Thus, flare removal algorithms are highly desired.

Flares can be categorized into three main types: scattering flares, reflective flares, and lens orbs. In this competition, we mainly focus on removing the scattering flares, as they are the most prevalent type of nighttime image degradation. Early attempts at scattering flare removal were made by Wu *et al*. [22], who proposed a dataset with physically-based synthetic flares and flare photos taken in a darkroom. However, these flares have obvious domain gap with real-captured nighttime flares. To address this issue, Dai *et al*. [2, 3] propose a new dataset Flare7K++ which is specifically designed for nighttime scenes. Additionally, various other efforts have been pursued, including flare removal in multi-light scenarios [28], in raw image formats [9], and smartphone reflective flare [5]. However, due to variations in lens structures and the diversity of lens protectors, existing lens flare datasets struggle to cover all types of lens flare comprehensively. This occasionally results in 'out of distribution' occurrences of lens flare in real-world captures for specific type of lenses. In response to the growing demand among smartphone and lens manufacturers, this competition focuses on developing lens-specific lens flare removal methods. In addition to the Flare7K++ dataset, we provide 600 aligned flare-corrupted/flare-removed image pairs specifically for certain smartphone's rear camera. Furthermore, in order to mimic the commonly-used high resolutions in the industry, all training set and test set images' resolutions are set to 2K.

We hold this challenge in conjunction with the third MIPI Challenge which will be held on CVPR 2024. Similar to the previous MIPI challenge [4, 15, 16, 29], we are seeking an efficient and high-performance image restoration algorithm to be used for recovering flare-corrupted images. MIPI 2024 consists of three competition tracks:

- **Few-shot RAW Image Denoising** is geared towards training neural networks for raw image denoising in scenarios where paired data is limited.
- **Demosaic for HybridEVS Camera** is to reconstruct HybridEVS's raw data which contains event pixels and defect pixels into RGB images.
- **Nighttime Flare Removal** is to improve nighttime image quality by removing lens flare effects.

2. MIPI 2024 Nighttime Flare Removal

To facilitate the development of efficient and high-performance flare removal solutions, we provide a high-quality dataset to be used for training and testing and a set of evaluation metrics that can measure the performance of developed solutions. This challenge aims to advance research on nighttime flare removal.

[1]S-Lab, Nanyang Technological University
[2]Samsung Research China
[3]Nankai University

2.1. Datasets

Our competition provides a paired flare-corrupted/flare-free dataset that contains 600 aligned training images in 2K resolution. Participants can train a pixel-to-pixel network with this dataset for flare removal. The validation set and testing set consist of 50 and 50 pairs of images, respectively. The input images from the validation and testing set are provided and the ground truth images are not available to participants. In addition, participants can also use Flare7k++ [3] as an additional training dataset and its released checkpoint. The Flare7k++ provides 5,000 synthetic flare images in 1440×1440, 962 real flare images in 756×1008, and 23,949 background images. Flare images can be added to the flare-free background images to synthesize paired data for training.

2.2. Evaluation

In this competition, we mainly focus on the perceptual similarity of the flare-removed image and flare-free ground truth. Thus, we choose to use the Learned Perceptual Image Patch Similarity (LPIPS) [26] as our main evaluation metric. Besides, Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) [19] are also listed as references. Participants can view these metrics of their submission to optimize the model's performance.

2.3. Challenge Phase

The challenge consisted of the following phases:

1. Development: The registered participants get access to the data and baseline code, and are able to train the models and evaluate their running time locally.
2. Validation: The participants can upload their models to the remote server to check the fidelity scores on the validation dataset, and to compare their results on the validation leaderboard.
3. Testing: The participants submit their final results, code, models, and factsheets.

3. Challenge Results

Among 170 registered participants, 14 teams successfully submitted their results, code, and factsheets in the final test phase. Out of these, 12 teams have contributed their solutions to this report. Table 1 reports the final test results and rankings of the teams. Only two teams train their models with extra data of real-captured nighttime background images. The methods evaluated in Table 1 are briefly described in Section 4 and the team members are listed in Appendix. Finally, the MiAlgo_AI team is the first place winner of this challenge, while BigGuy team win the second place and SFNet-FR team is the third place, respectively.

4. Methods

MiAlgo_AI This team proposes a Progressive Perception Diffusion Network (PPDN). By implementing a two-stage network architecture, it generates visually high-quality results in a progressive strategy. Specifically, in the first stage, there is a diffusion module that aims to remove the flares of the input to the greatest extent feasible. Inspired by [10], they use the IR-SDE as the base diffusion module, which can avoid generating smooth results during deflaring. In the second stage, they utilize the AOT Block [25] as the fundamental enhancement module to amplify the details of the flat domain in the output of the first stage and recover the content of the flare texture. Significantly, the output of the diffusion module serves as crucial conditional information. The output of the inpainting module has high-quality visualization effects. The entire pipeline is shown in Figure 1.

When generating a training dataset, it contains a synthetic dataset and a real dataset. Borrowing from [3], they additionally collected several high-quality nighttime photos at two resolutions and added compound flares as a synthetic dataset. A paired flare dataset containing 600 aligned training images provided by the event group is the real dataset. Upon examining the validation images, the authors noticed a variation in brightness between the input and output images. Furthermore, to simulate more realistic flare-corrupted images when encountering heavier input fog, they augmented the base image with additional light and local haze

When training, they first train the diffusion module using synthetic data generated online and real dataset for about 400,000 iterations. Then they fix the weight of the diffusion module and train the enhancement module for about 300,000 iterations with a batch size of 4. The initial learning rate is $lr = 1e - 4$, and cosine annealing is used to reduce the learning rate. To optimize the model's capacity for extracting global information, the team abstains from employing a random cropping strategy on the input image. The training process is executed using the power of 2 Tesla A100 GPUs for approximately 3 days.

BigGuy This team designs a one-stage Restormer-like Structure [24], making full use of the hierarchical multi-scale information from low-level features to high-level features. To ease the training procedure and facilitate the information flow, an efficient Transformer for image restoration is utilized to model global connectivity and is still applicable to large images. Since flares can take up a large portion of the image, and possibly the entire image, during the removal of nighttime flares, it is critical to have a large receptive area. However, conventional window-based transformer methods limit the receptive field within the window, thus limiting their ability to capture global features. Also,

Table 1. Results of MIPI 2024 challenge on nighttime flare removal. 'Runtime' for per image is tested and averaged across the validation datasets, and the image size is 1440×1920. 'Params' denotes the total number of learnable parameters.

Team Name	Metric			Params (M)	Runtime (s)	Platform	Extra data	Ensemble
	LPIPS*	PSNR	SSIM					
MiAlgo_AI	$0.1435_{(1)}$	$22.15_{(1)}$	$0.7075_{(2)}$	141.61	8.0	NVIDIA Tesla A100	Yes	-
BigGuy	$0.1502_{(2)}$	$21.50_{(7)}$	$0.6996_{(7)}$	26.13	30.0	NVIDIA RTX 3090	-	self-ensemble
SFNet-FR	$0.1518_{(3)}$	$21.74_{(3)}$	$0.7188_{(1)}$	383.42	0.017	NVIDIA RTX 3090Ti	-	-
LVGroup_HFUT	$0.1620_{(4)}$	$21.71_{(5)}$	$0.7041_{(4)}$	/	0.055	NVIDIA RTX 4090	-	-
NativeCV	$0.1688_{(5)}$	$21.39_{(8)}$	$0.6929_{(8)}$	/	/	/	-	-
CILAB-IITMadras	$0.1697_{(6)}$	$21.70_{(6)}$	$0.7042_{(3)}$	61.40	0.70	NVIDIA RTX 4090	-	model-ensemble
Xdh-Flare	$0.1703_{(7)}$	$21.99_{(2)}$	$0.7005_{(5)}$	24.47	1.78	NVIDIA RTX 4090	Yes	-
Fromhit	$0.1713_{(8)}$	$21.24_{(9)}$	$0.6850_{(10)}$	/	1.256	NVIDIA RTX A6000	-	-
UformerPlus	$0.1732_{(9)}$	$21.73_{(4)}$	$0.6997_{(6)}$	38.79	0.32	NVIDIA RTX 3090	-	-
GoodGame	$0.1813_{(10)}$	$20.85_{(10)}$	$0.6881_{(9)}$	19.47	0.73	NVIDIA RTX 3090	-	-
IIT-RPR	$0.1926_{(11)}$	$20.66_{(11)}$	$0.6775_{(11)}$	20.47	1.72	NVIDIA RTX 2080Ti	Yes	-
LSCM-HK	$0.1926_{(12)}$	$22.66_{(12)}$	$0.6775_{(12)}$	20.47	/	NVIDIA RTX 3090	-	-
Hp_zhangGeek	$0.2332_{(13)}$	$19.74_{(13)}$	$0.6538_{(13)}$	2.11	0.13	NVIDIA RTX 4090	-	self-ensemble
Lehaan	$0.6749_{(14)}$	$16.27_{(14)}$	$0.4655_{(14)}$	/	/	NVIDIA RTX 4050	-	-

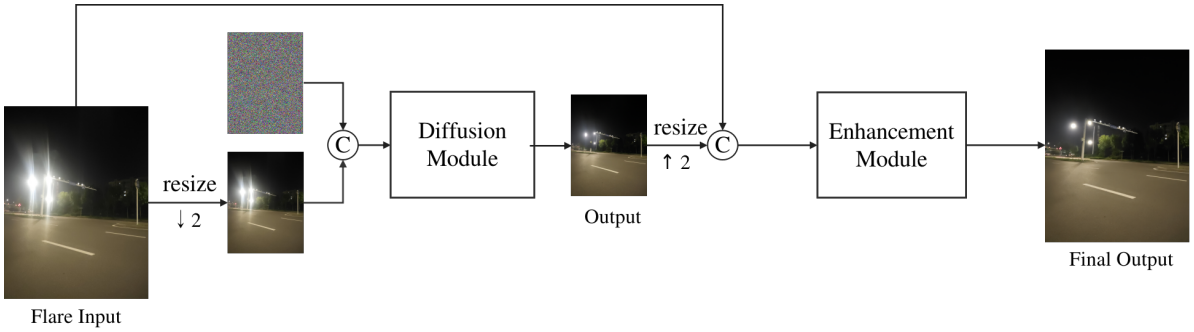

Figure 1. The network architecture of MiAlgo_AI.

to allow the network to focus more on the flare region, this team used a difference algorithm to obtain a mask between the input image and the ground truth to compute the loss function.

The training is using the Adam optimizer [8], with $lr = 1e-5$ and default decay parameters. The optimized objective is a mixture of terms, combining the L1 loss, VGG [13] loss, mask loss, and LPIPS loss comparing the output restored image to the reference image.

SFNet-FR This team proposes *SFNet*, a solution based on multiple-level frequency-band decomposition, performed in both the RGB spatial domain and the image frequency domain. Figure 2 offers a graphical representation of the *SFNet* model, with the encoder module *(center)* and the paired decoder *(right)*. The solution follows the UNet structure [12], with multiple frequency-band skip connections, such as the high-frequency RGB domain *(red)*, or Haar Discrete Wavelet Transform (DWT) *(orange)* skip connection, with another preserving low-frequency combined domains features *(blue)*.

The information preserved through the aforementioned skip connections is the output of the dual domain splitting performed at the encoder level. In the RGB domain, the splitting is performed through a module combining the *AvgPooling* and the *MaxPooling* operators, while the splitting in the frequency domain is done through a Haar DWT operator. The solution builds on previous work [17], with a considerable model complexity allocated to the modules processing high-frequency information, while the lower complexity features are refined through simpler modules.

At the decoder level, the multi-domain high-frequency information is fused in a Stereo Channel Attention module [1], while the low-frequency features are processed separately, then receiving the enhanced high-frequency information as compensation.

The training is using the Adam optimizer [8], with $lr = 2e-4$ and default decay parameters. The inputs are cropped into 320×320 with a batch size of 1. The optimized objective is a mixture of terms, combining the L1 loss with a VGG [13] loss, and a gradient loss comparing the Sobel gradient of the output restored image to the gradient of the reference image.

This builds SFNet as a capable solution for the flare removal task which, trained on the Flare7K++ dataset and the challenge data, can achieve a significant performance level with consistent results for all the evaluated metrics. As a single-stage solution, solving the nighttime flare removal in an end-to-end fashion (without using any self-ensemble or model-ensemble structures), SFNet represents a good trade-

off in terms of achieved performance for the characteristic computational cost. The solution is one of the fastest compared to the other competitors, being able to perform real-time flare removal on a consumer-grade GPU, the Nvidia RTX 3090Ti.

LVGroup_HFUT This team proposes to divide the whole training dataset into different subsets based on different distributions, then train the neural network model on each subset separately, and finally integrate the obtained results to realize flare removal. Specifically, this team refers to NAFNet [1] and FCL-GAN [27] to construct the model, and then divide the provided training data into two subsets (according to the resolution due to the difference in their distributions) and train the models separately to obtain two pre-trained models, Fig.3 demonstrates the detailed architecture of this team.

Training description. The proposed architecture of this team is based on PyTorch 2.2.1 and an NVIDIA 4090 with 24G memory. They set 2500 epochs for training with batch size 4, using AdamW with β_1=0.9 and β_2=0.999 for optimization. The initial learning rate was set to 0.0002, which was reduced by half every 50 epochs. For data augment, they first randomly crop the image to 768×768 and then perform a horizontal flip with probability 0.5. Besides, as mentioned above, two different models were trained separately to fit different distributions (different resolutions in the experiment).

Test description. Similarly to the training stage, the test image with the original resolution is fed into the two models according to their distributions for inference to obtain the results, and finally the obtained results are combined.

CILAB-IITMadras This team proposes to ensemble 3 Uformers using different metrics and methodologies. Flare Removal Uformer GAN(FRUGAN) utilizes UFormer[21] as a generator and a multiscale discriminator that utilizes both adversarial and feature-matching loss to remove flare from the given image. They have used three discriminators similar to the network in pix2pixhd[18] operating at different image scales as shown in Fig.4.

A combination of adversarial loss, multiscale discriminator loss, L1, and perceptual losses is used to train FRUGAN. To improve the results of FRUGAN, Uformer-1, and Uformer-2 were trained on losses $L_{mse} + L_{LPIPS}$ and L_{LPIPS} respectively. The complete ensemble model uses weights $w1 = 0.60$, $w2 = 0.25$, and $w3 = 0.15$ as depicted in Fig.5.

They have trained the proposed Uformer model by randomly cropping the images from the competition dataset to 800*800 images and then resizing it to 512×512 images. They further did data augmentations of random horizontal and vertical flips and random rotations of up to 5°. The model was trained with Adam optimizer with β_1 set to 0.9 and β_2 set to 0.999. This team found that the model started over-fitting after 50 epochs. Both the discriminator and the generator were updated after every iteration. The FRUGAN model was trained on NVIDIA A100 with 40GB VRAM.

Similarly, the other Uformer models were trained by resizing the input images to 512*512. The model was trained with Adam optimizer with β_1 set to 0.9 and β_2 set to 0.999. The learning rate was set to 1e-4. They have trained for 100 epochs. The Uformer models were trained on NVIDIA RTX 4090 with two 24GB VRAM GPUs.

Xdh-Flare This team adopts Uformer architecture [20] and makes several improvements in the dataset and loss function to remove nighttime flare. The authors observe that the data distribution in the target domain differs from that in the source domain, primarily manifesting in the discrepancy between the proportion of light sources and flare tones training set images and the testing set. In response to the data disparity between the target and source domains, the authors adopt a strategy of augmenting the dataset in the target domain to reduce domain gaps. Following the method of synthesizing data from Flare7k++ [3], they add flare images provided by Flare7k++ to the flare-free background images provided by BracketFlare [5] to synthesize paired data for training. They observe the flare tones of all images in the dataset around light sources, selecting flare images based on a similar distribution of flare tones in the flare images, completing the augmentation of the dataset from the source domain to the target domain.

They use L1 loss, SSIM loss, and perceptual loss to remove flare regions. Given an image with flares I_{input}, Uformer network outputs flare-free image I_g and flare image I_f. To better recover the flare-free image and flare image, the authors add I_f and I_g , then calculate the L1 loss with I_{input}. The total L1 loss is represented as:

$$\text{loss}_{L1} = \|I_{input} - (I_g + I_f)\| * w + \|I_o - I_g\| \quad (1)$$

where w is a hyper-parameter set to be 2.

Using the SSIM loss function can make the generated images closer to the real images in terms of structure, improving the quality of the generated images. SSIM Loss is defined as:

$$\text{loss}_{SSIM} = 1 - \text{SSIM}(I_o, I_g) \quad (2)$$

Using perceptual loss can measure perceptual similarity, improving the similarity between the output images and input images. $f(\cdot)$ represents the perceptual extraction network, and they use AlexNet pretrained on ImageNet to extract features:

$$\text{loss}_{per} = \|f(I_o) - f(I_g)\| * w + \|f(I_g + I_f) - f(I_{input})\| \quad (3)$$

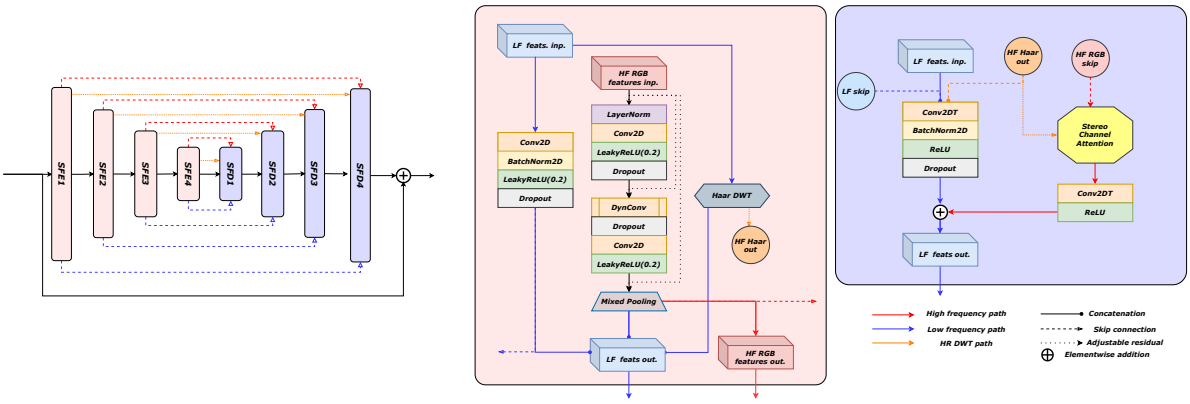

Figure 2. A graphical representation of the proposed SFNet *(left)*, with a detailed representation of the Spatial-Frequency Encoder (SFE) *(center)*, and the Spatial-Frequency Decoder (SFD) *(right)*.

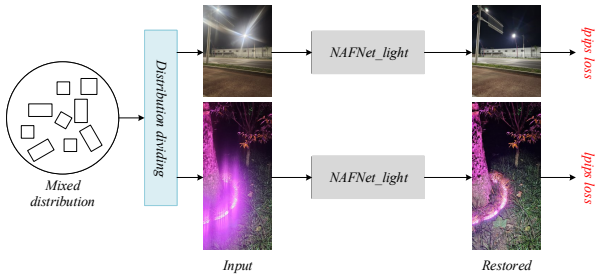

Figure 3. The network architecture of LVGroup_HFUT team.

where w is a hyper-parameter set to be 2.

The total loss is the weighted sum of the three types of loss:

$$\text{loss}_{total} = \alpha * \text{loss}_{L1} + \beta * \text{loss}_{SSIM} + \gamma * \text{loss}_{per} \quad (4)$$

where α, β, and γ are respectively set to 1, 0.01, and 1 in their experiments.

When training, the input images are cropped to 512×512. The authors also use gamma correction and inverse gamma correction to the input images. The authors use the Adam optimizer with an initial learning rate of 0.0001 and a batch size of 4. The network is trained for 500 epochs on the augmented dataset.

Fromhit This team employs an efficient image restoration model, NAFNet[1], as the base model. Specifically, a four-scale CNN encoder and decoder are adopted, and each scale contains two NAFBlocks. Between the encoder and decoder, the authors use four NAFBlocks as a middle block. Then, the authors design a loss function for nighttime flare removal. During training, the authors minimize the sum of two losses, L_1 loss encourages the predicted flare-free image to be close to the ground truth both photometrically and perceptually. Like [23], the perceptual loss is computed by feeding the predicted flare-free image and ground-truth through a pre-trained VGG-19 network[13]. This team does not process light sources. For training, the authors randomly cropped 512×512 patches from the training images as inputs. The mini-batch size is set to 4 and the whole network is trained for 1×10^5 iterations. The learning rate is initialized as 1×10^-4, and the authors use ADAM as the optimizer with $\beta_1 = 0.9$, and $\beta_2 = 0.99$,

UformerPlus The team proposes an effective nighttime flare removal pipeline. Firstly, they employed the strong image restoration model Uformer [20] as the base model, which has an encoder, a decoder, and skip connections. The Locally-enhanced Window (LeWin) block is adopting the design in Uformer. And then to leverage the frequency characteristics of the image, the authors introduce the ResFFT-Block [11] after the LeWin block, which is based on Fast Fourier Convolution (FFC), to extract global frequency features for reducing distortions and enhancing details. Moreover, the authors use two NAFBlocks [1] as the refinement module following the last decoder blocks for powerful representation. Finally, some improvements were made to the loss function. Instead of using a fixed loss weight, the team dynamically adjusts the weight ratio of the loss function as the training iterations progress, with an increased emphasis on perceptual loss for better visual results during training. Through model fusion and weighted loss function, the performance of the model was further improved and ultimately achieved competitive results in the challenge.

The loss function comprises both Charbonnier L_1 loss and perceptual loss, with dynamically assigned weights. The inputs are cropped into 512×512 with a batch size of 2, and the Adam optimizer [8] is used. The initial learning rate is set to 1×10^{-4}, and the CosineAnnealingLR scheduler is employed with a maximum of 300,000 iterations and a minimum learning rate of 1×10^{-6} to adjust the learning rate. They also use horizontal and vertical flips for data enhancement. For testing, the authors split the original im-

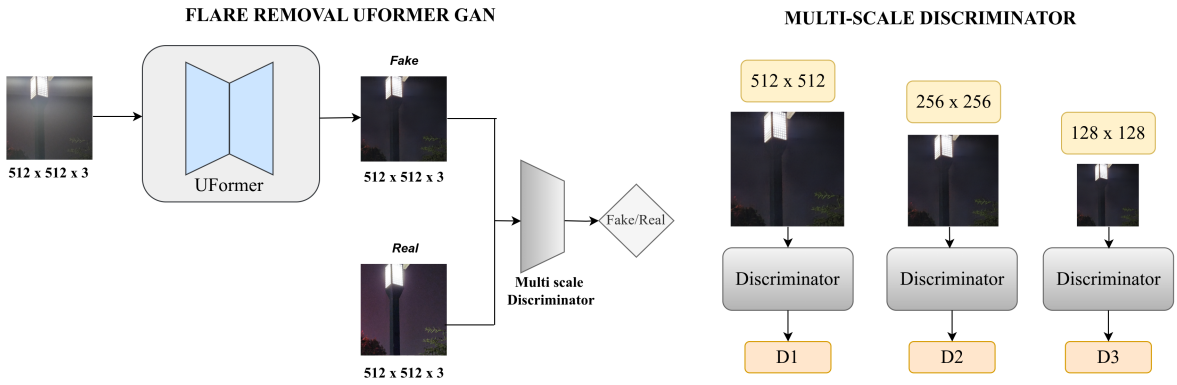

Figure 4. Overview of FRU-GAN Architecture with multi-scale discriminator.

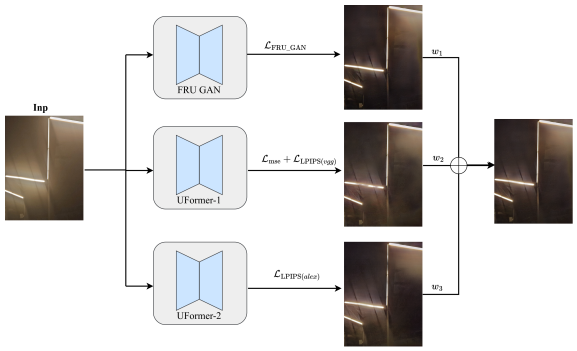

Figure 5. Ensemble Model with blended output.

ages into 512×512 patches and generate the final flare-free images. All experiments were performed on two NVIDIA RTX 3090 GPUs with 24GB memory.

GoodGame This team proposes an efficient flare removal network, based on Restormer[24]. In the model, they use Flare-Aware Transformer Blocks to capture the Flare in the image, and the composition structure is similar to that of Restormer. Residual connections are also used in the model so that the model only needs to learn the changes in the flare and does not need to reconstruct the entire image. The model is also efficient enough to infer large-resolution images. Fig. 6 shows the framework of the entire model.

They trained 300,000 iterations to take the model to convergence. Progressive learning was used during the training process, from the initial batch size of 16 and patch size (resize) of 128, to the final batch size of 2 and patch size of 384. During the training process, gradually increase the patch size of the image and reduce the batch size, so that the model can learn more details of the image. They choose `AdamW` as an optimizer, set the initial learning rate to 3×10^{-4}, and introduce a weight decay of $1e-4$. At the same time, they adopted the cosine annealing learning rate scheduler (`CosineAnnealingLR`), where $T_{\max}$ is set to 500 and the minimum learning rate is set to $1e-6$.In terms of loss, they used L1 loss, Fourier L1 loss, and Lpips loss, with Lpips loss accounting for the largest proportion.

IIT-RPR This team designs a method, based on U-former model architecture. FADU-Net shown in Figure 7, trained from the ground up. The synthesis of training images involves utilizing Flickr24K as background images and incorporating 5k scattering flare images from Flare7K. An innovative night data augmentation strategy (Night Data Aug) is implemented for background images, featuring four modes, randomly selected for each image during training. The provided competition's images are also used as the validation dataset during the training of the architecture. The loss function is a combination of L1 loss and perceptual loss, with distinct weights assigned to areas inside and outside the flare.
During training, a patch size of 512 is utilized, and the Adam optimizer is employed with an initial learning rate of 0.0001. This team conducts training for 1200K iterations and observes that extending the training duration may further enhance results. Although, the proposed pipeline demonstrates its efficacy by achieving a PSNR of 20.66 and LPIPS of 0.1926 on the MIPI challenge's test dataset, showcasing the team's adeptness in addressing the complex task of nighttime flare removal. It outputs the two images i.e., the predicted flare and the predicted image. The predicted flare image shows the reflective and scattering flare present in the input image. And predicted image is the output image without the flares containing only the light source. This architecture is trained on the 11GB NVIDIA GeForce RTX 2080 Ti GPU for 6 days and 13 hours.

Hp_zhangGeek This team designs a conditional variational autoencoder (CVAE) [14] for removing nighttime flares. Specifically, for the nighttime flare removal task, CVAEs can contribute significantly due to their ability to model complex data distributions and generate high-quality, diverse outputs conditioned on given inputs. They also de-

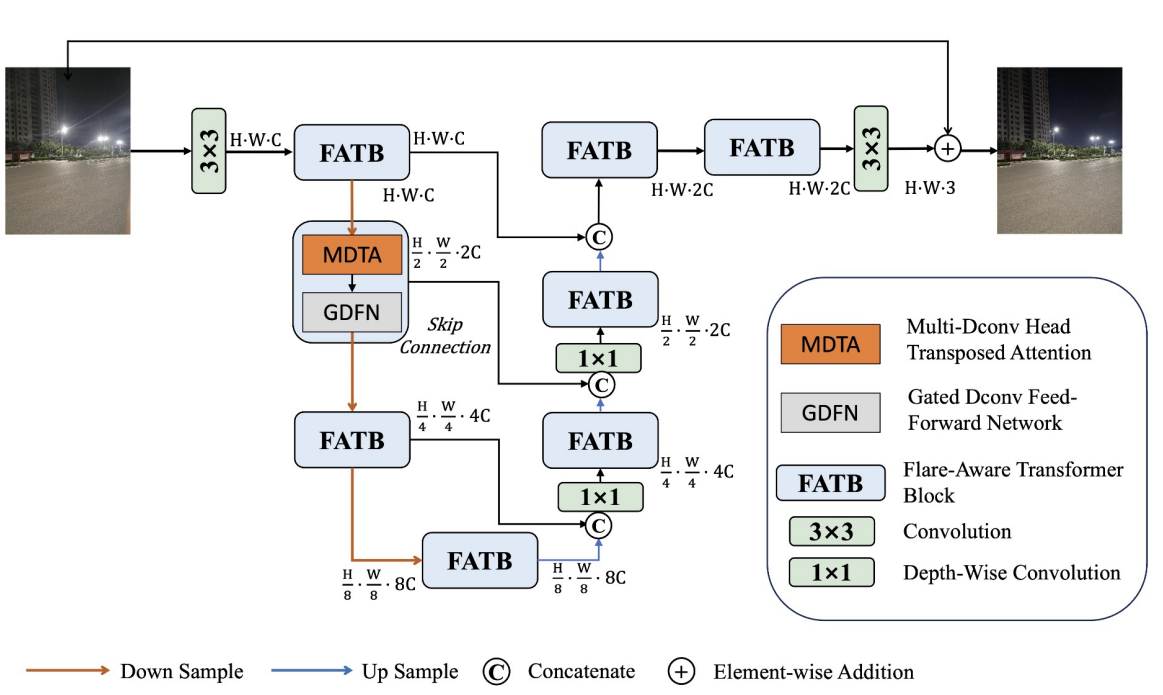

Figure 6. The network architecture of GoodGame team. MDTA and GDFN are the same as Restormer[24].

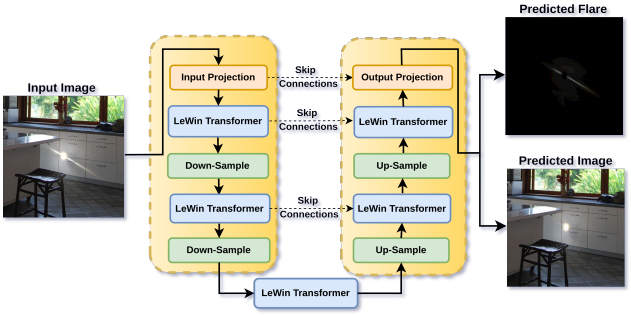

Figure 7. The network architecture of FADU-Net / IIT-RPR team.

signed an Adaptive Normalization Module (ANM) to enhance the details of input features.

As depicted in Fig. 8 (a), they adopt U-Net architecture [12] in the encoder that progressively downsamples the image into a more compact representation. The Prior network (abbreviated as Pr in Fig. 8 (a)) shares the same structure as the encoder. This network is designed to learn a prior distribution of the latent variables. In the network, the prior network models the distribution of latent variables that is from the flare-corrupted images. Inspired by PUIE-Net [6] and U-GAT-IT [7], the adaptive normalization module (ANM) takes the latent representation from the encoder and refines it. As shown in Fig. 8 (b), the adaptive normalization module involves adjusting the feature distribution of the latent representation to a state that is more conducive to generating a clean image without flares. Using the refined latent

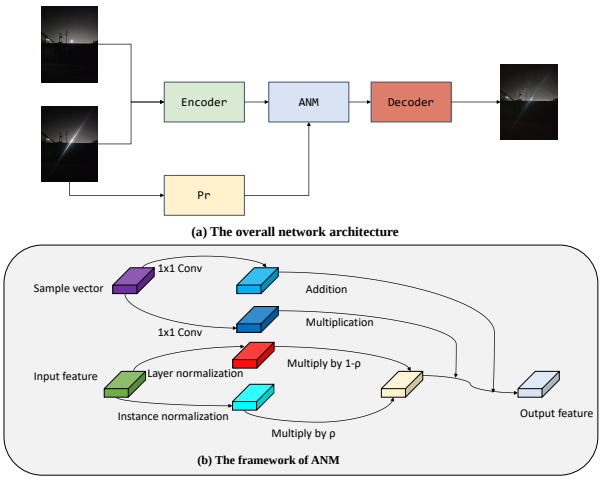

Figure 8. The overall network architecture of HP_zhang Geek team.

representation from the ANM, the decoder network reconstructs the image. The decoder effectively reverses the encoding process, upscaling the latent representation back to the original image dimensions to reproduce the image without the lens flare.

This team only used the 600 image pairs for training. In cases where the dataset is relatively small, they adopt a 90/10 split to maximize the amount of training data while still having a validation set to monitor overfitting and per-

formance. They resize the images to 256×256 and apply horizontal flipping, vertical flipping, and rotation randomly for data augmentation. The batch size is 16. They train the network for 700 epochs. For the training phase, the loss function for training is formulated as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{re}} + \alpha\mathcal{L}_{\text{kl}} + \beta\mathcal{L}_{\text{per}}, \tag{5}$$

where $\mathcal{L}_{\text{re}}$ is the reconstruction loss, $\mathcal{L}_{\text{kl}}$ is the KL divergence, and $\mathcal{L}_{\text{per}}$ is the perceptual loss, α and β are learnable parameters.

The experimental setup for the computational framework was implemented on an Ubuntu-based workstation equipped with a single NVIDIA RTX 4090 card. The duration of the model training, inclusive of the validation phase, spanned approximately 16 hours. The inference time of the model is approximately 6.3 milliseconds per image. It is pertinent to note that the inference protocol entailed sampling the model 20 times for each image. The final output was derived by computing the mean across these 20 samples, ensuring robustness and stability in the generated results. This methodical approach to inference underscores the model's efficacy in handling the variability inherent in the data, thereby contributing to the reliability of the outcomes.

Lehaan This team utilizes the Uformer model [21] as a flare-erasing module coupled with AOT-GAN for image inpainting. Uformer employs a hierarchical encoder-decoder structure akin to UNet but substitutes convolutional layers with Transformer blocks. Key aspects of Uformer include the Locally-enhanced window (LeWin) Transformer block for localized context capture and the Multi-scale restoration modulator for feature adjustment at various scales. AOT-GAN (Aggregation of Contextual Transformation - GAN) enhances context reasoning through AOT blocks in the generator, facilitating the aggregation of contextual transformations from different image areas for accurate inpainting. AOT blocks are a novel approach for convolutional neural networks designed to enhance context reasoning. They achieve this by splitting a large kernel into smaller ones, each specializing in a specific number of output channels. These sub-kernels then analyze the input using different dilation rates, allowing them to focus on varying areas of the image. Finally, the outputs from all sub-kernels are merged, enabling the AOT block to consider the input from various perspectives and capture richer contextual information. This approach has shown promise in improving tasks like image inpainting.

While the Uformer model can sufficiently remove the flare, it also inadvertently removes the pixels that were behind the flares. Hence, an inpainting module is used to inpaint back the image that was removed too. To specify the region required for inpainting, image differencing followed by thresholding is done in order to get the regions affected by UFormer. This is given as the mask to AOT-GAN while inpainting.

The complexity of the Uformer method comprises several stages: Self Attention: Time complexity - $O(n^2d^2)$, Space complexity - $O(n^2d)$. Feed-Forward Network: Time complexity - $O(2nd^2)$, Space complexity - $O(nd)$. Layer Normalization and Residual Connection: Constant time and space complexity - $O(1)$. Multi-Head Attention (MHA): Time complexity - $O(nh^2d^2)$. Total time complexity:$O(n^2hd^2)$, Space Complexity $O(n^2d + nhd)$.

For model training Mini-Batch size: 8, Epochs: 1000, Training workers: 4, Evaluation workers: 4, Dataset: Flare7k++, Optimizer: AdamW, learning rate = 10e-3 and default decay parameters, Weight decay: 0.02, GPU: NVIDIA RTX 4050 Laptop GPU.

5. Conclusions

In this report, we review and summarize the methods and results of MIPI 2024 challenge on Nighttime Flare Removal. The participants have made significant contributions to this challenging track, and we express our gratitude for the dedication of each participant.

References

[1] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, 2022. 3, 4, 5

[2] Yuekun Dai, Chongyi Li, Shangchen Zhou, Ruicheng Feng, and Chen Change Loy. Flare7k: A phenomenological nighttime flare removal dataset. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1

[3] Yuekun Dai, Chongyi Li, Shangchen Zhou, Ruicheng Feng, Yihang Luo, and Chen Change Loy. Flare7k++: Mixing synthetic and real datasets for nighttime flare removal and beyond. *arXiv preprint arXiv:2306.04236*, 2023. 1, 2, 4

[4] Yuekun Dai, Chongyi Li, Shangchen Zhou, Ruicheng Feng, Qingpeng Zhu, Qianhui Sun, Wenxiu Sun, Chen Change Loy, Jinwei Gu, Shuai Liu, et al. Mipi 2023 challenge on nighttime flare removal: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 1

[5] Yuekun Dai, Yihang Luo, Shangchen Zhou, Chongyi Li, and Chen Change Loy. Nighttime smartphone reflective flare removal using optical center symmetry prior. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 1, 4

[6] Zhenqi Fu, Wu Wang, Yue Huang, Xinghao Ding, and Kai-Kuang Ma. Uncertainty inspired underwater image enhancement. In *European Conference on Computer Vision*. Springer, 2022. 7

[7] Junho Kim, Minjae Kim, Hyeonwoo Kang, and Kwang Hee Lee. U-gat-it: Unsupervised generative attentional net-

works with adaptive layer-instance normalization for image-to-image translation. In *International Conference on Learning Representations*, 2019. 7

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3, 5

[9] Fengbo Lan and Chang Wen Chen. Tackling scattering and reflective flare in mobile camera systems: A raw image dataset for enhanced flare removal. *arXiv preprint arXiv:2307.14180*, 2023. 1

[10] Ziwei Luo. Image restoration with mean-reverting stochastic differential equations. In *International Conference on Machine Learning*, 2023. 2

[11] Xintian Mao, Yiming Liu, Wei Shen, Qingli Li, and Yan Wang. Deep residual fourier transformation for single image deblurring. *arXiv preprint arXiv:2111.11745*, 2021. 5

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015. 3, 7

[13] K Simonyan and A Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations (ICLR 2015)*. Computational and Biological Learning Society, 2015. 3, 5

[14] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. *Advances in Neural Information Processing Systems*, 2015. 6

[15] Qianhui Sun, Qingyu Yang, Chongyi Li, Shangchen Zhou, Ruicheng Feng, Yuekun Dai, Wenxiu Sun, Qingpeng Zhu, Chen Change Loy, Jinwei Gu, et al. Mipi 2023 challenge on rgbw remosaic: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 1

[16] Qianhui Sun, Qingyu Yang, Chongyi Li, Shangchen Zhou, Ruicheng Feng, Yuekun Dai, Wenxiu Sun, Qingpeng Zhu, Chen Change Loy, Jinwei Gu, et al. Mipi 2023 challenge on rgbw fusion: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2870–2876, 2023. 1

[17] Florin Vasluianu and Radu Timofte. Efficient video enhancement transformer. In *IEEE International Conference on Image Processing*, 2022. 3

[18] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 4

[19] 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. 2

[20] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restorationn. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 4, 5

[21] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 4, 8

[22] Yicheng Wu, Qiurui He, Tianfan Xue, Rahul Garg, Jiawen Chen, Ashok Veeraraghavan, and Jonathan T. Barron. How to train neural networks for flare removal. In *IEEE International Conference on Computer Vision*, 2021. 1

[23] Yicheng Wu, Qiurui He, Tianfan Xue, Rahul Garg, Jiawen Chen, Ashok Veeraraghavan, and Jonathan T Barron. How to train neural networks for flare removal. In *IEEE International Conference on Computer Vision*, 2021. 5

[24] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 6, 7

[25] Yanhong Zeng, Jianlong Fu, Hongyang Chao, and Baining Guo. Aggregated contextual transformations for high-resolution image inpainting. *IEEE Transactions on Visualization and Computer Graphics*, 2022. 2

[26] Richard Zhang, Phillip Isola, Alexei A. Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[27] Suiyi Zhao, Zhao Zhang, Richang Hong, Mingliang Xu, Yi Yang, and Meng Wang. Fcl-gan: A lightweight and real-time baseline for unsupervised blind image deblurring. In *30th ACM International Conference on Multimedia*, 2022. 4

[28] Yuyan Zhou, Dong Liang, Songcan Chen, Sheng-Jun Huang, Shuo Yang, and Chongyi Li. Improving lens flare removal with general-purpose pipeline and multiple light sources recovery. In *IEEE International Conference on Computer Vision*, 2023. 1

[29] Qingpeng Zhu, Wenxiu Sun, Yuekun Dai, Chongyi Li, Shangchen Zhou, Ruicheng Feng, Qianhui Sun, Chen Change Loy, Jinwei Gu, Yi Yu, et al. Mipi 2023 challenge on rgb+ tof depth completion: Methods and results. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 1