scientific reports

Check for updates

OPEN

SMFR-Net: simple multi-domain flare removal network

Shaofeng Liu[1], Guorong Chen[1,2]✉, Weijie Zhang[1], Qingru Zhang[1], Jinmei Zhang[1] & Jian Wang[1]

When strong light enters the lens, multiple internal reflections and scattering can cause flare, significantly degrading image quality and affecting the performance of downstream vision tasks. In practical photography, flare is often caused by multiple strong light sources, resulting in artifacts such as bright streaks or diffuse halos, which often cover large areas of the image. To effectively remove flare, the network needs to have a large receptive field. However, although the native Transformer architecture has global modeling capability, its computational complexity grows with the square of the image resolution, making it difficult to apply on resource-constrained devices. The windowed attention mechanism, as a compromise, improves computational efficiency but limits the receptive field to within the window, making it difficult to achieve true global perception. To address these issues, we propose a simple multi-domain image flare removal network-SMFR-Net, which achieves state-of-the-art (SOTA) performance with 7.981M parameters. Specifically, SMFR-Net consists of an encoder that jointly models the frequency and spatial domains, and a decoder with a simplified structure. The encoder first enhances global contextual awareness using a frequency domain module with Fourier Transform, then further expands the receptive field through a spatial domain module combined with multi-scale dilated convolutions, and introduces a Channel-Spatial Attention Mechanism to precisely locate the flare regions. The decoder, based on this, discards frequency domain modeling and simplifies the structure to reduce redundant computation. Furthermore, we design a structure-aware composite loss function for the network to improve overall performance. Experimental results show that SMFR-Net outperforms existing methods on the Flare7K++ real-world test set, synthetic test set, and several real-world scenes across most metrics, demonstrating superior flare removal performance and good application potential with its simple and efficient structure.

Keywords Image Flare Removal, Fourier Transform, Attention Mechanism, Lightweight Network

In natural scene photography and computer vision perception, lens flare is a common and complex imaging degradation phenomenon, usually caused by strong light sources (such as the sun or artificial lights) entering the camera, resulting in unwantedreflections and scattering within the lens system. These interfering light rays do not participate in the normal imaging process but are projected onto the sensor surface along abnormal paths, disrupting the structure and brightness distribution of the image. Depending on the manifestation and cause of the flare in the image, it is typically classified into two types: stray flare and reflected flare[1]. The former is usually caused by scattering phenomena due to dust, stains, or scratches on the lens surface, often appearing as bright streaks or overexposed regions extending along the light path; the latter is typically caused by multiple reflections within the lens group, forming bright spots with regular geometric shapes, such as polygonal halos or star-shaped light spots. Both types of flare interfere with the structural information of the image, degrade visual quality, and significantly affect downstream tasks such as semantic segmentation[2], object detection[3], and monocular depth prediction[4]. As shown in Fig. 1, the presence of flare causes severe interference with downstream tasks: it induces significant structural recognition errors in semantic segmentation and results in a loss of depth information in depth estimation, underscoring the great importance of its effective suppression for downstream applications.

To mitigate the impact of lens flare on image quality, early research primarily focused on optical optimization at the hardware level. Camera systems typically introduce anti-reflective coatings (AR coatings) on optical element surfaces to reduce reflectivity, thereby suppressing multiple reflections and interference caused by strong light sources within the lens system. These coatings are based on the principle of phase-cancellation interference and can effectively reduce the intensity of reflected light within specific wavelength ranges, thus enhancing image contrast. However, AR coatings generally function only under specific incident angles and

[1]School of Computer Science and Engineering, Chongqing University of Science and Technology, Chongqing 401331, China. [2]Chongqing Institute of Intelligent Mathematics and Autonomous Intelligence, Chongqing University of Science and Technology, Yubei,Chongqing 401127, China. ✉email: 2003057@cqust.edu.cn

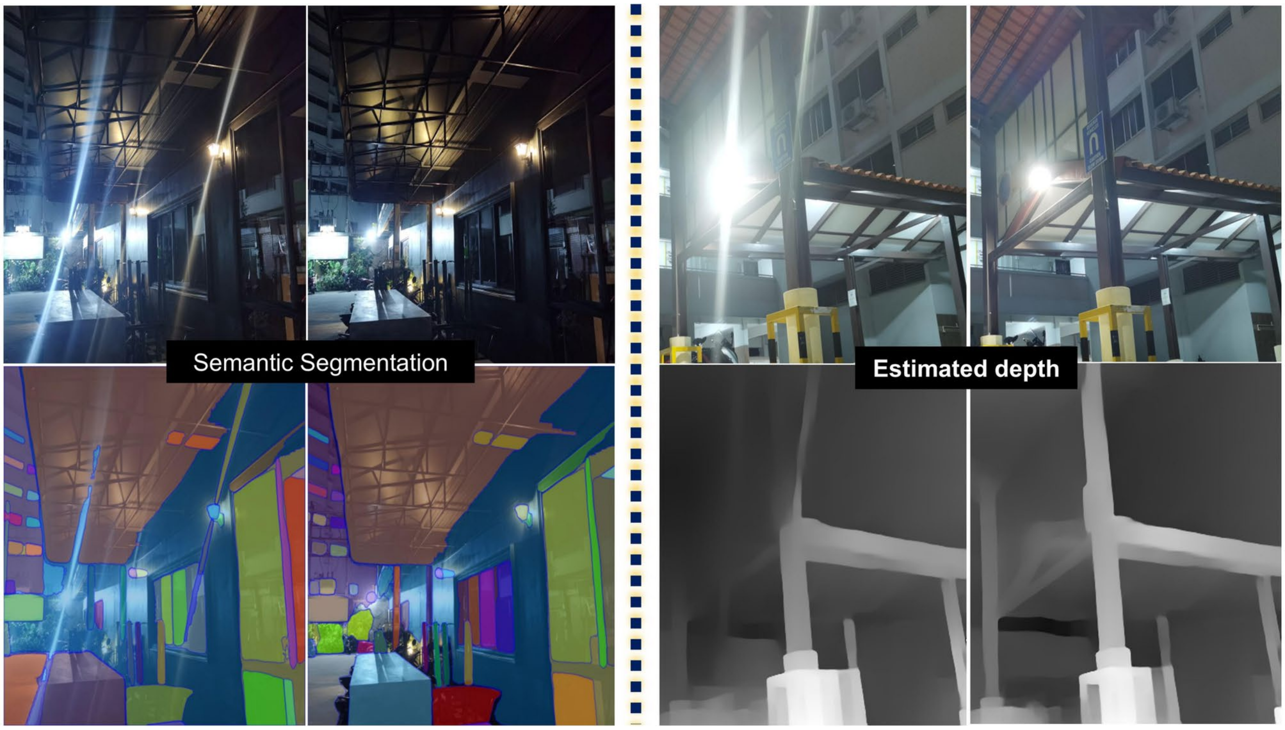

Fig. 1. The negative impact of lens flare on different downstream tasks. The left panel shows semantic segmentation results generated by Segment Anything[5], and the right panel shows monocular depth estimation results from Vision Transformers for Dense Prediction[4]. In both tasks, the flare-corrupted input (top-left of each set) leads to an erroneous output (bottom-left), while the output from the flare-free ground truth (bottom-right) is accurate.

spectral bands, and their high cost limits large-scale deployment. Another common approach is the use of lens hoods or the optimization of lens barrel designs to physically block off-axis incoming light, thereby reducing the interference of stray light in the imaging process. These methods can partially suppress the occurrence of bright artifacts, but their effectiveness is constrained by scene composition and light source positions, making them less adaptable to complex and dynamic natural lighting conditions. Moreover, hardware-based approaches are essentially pre-capture suppression strategies, which are incapable of addressing flare artifacts in already captured images. As a result, they exhibit inherent limitations in practical applications.

To overcome the limitations of hardware-based methods in adaptability and post-processing, researchers have proposed a variety of software-based approaches for image flare removal. Most traditional methods adopt a two-stage strategy: first detecting potential flare regions in the image, followed by restoration and reconstruction of the affected areas. Early works typically relied on explicit modeling based on image brightness, shape, or spatial features. For example, Chabert et al.[6] constructed candidate flare regions using multi-thresholding and contour feature extraction, and completed reconstruction via sample-based image inpainting; Vitoria et al.[7] detected overexposed local features to generate masks for flare suppression; Asha et al.[8] focused on strong highlights in the background caused by sunlight or flickering sources to formulate a targeted flare-filling strategy. Although such methods perform significantly well in handling flare with regular shapes or single-type artifacts, they rely on handcrafted features and struggle to handle real-world flare phenomena that are complex, asymmetric, and spatially variant.In addition, some approaches attempted to model the point spread function (PSF) and restore occluded regions via deconvolution[9]. However, these methods usually assume spatial invariance and circular symmetry of the flare patterns, which limits their applicability in practice. Due to the diversity of flare in terms of intensity, shape, and position, as well as the ambiguous boundaries between flare and naturally bright regions, traditional image-processing-based methods often suffer from high false detection rates and weak generalization, making them insufficient for robust image quality restoration in complex scenes.

In recent years, deep learning methods have achieved remarkable progress in image restoration and other visual tasks, providing new solutions to the problem of lens flare removal. Wu et al.[10] combined physical modeling to synthesize the first training dataset for flare removal and proposed the SIFR method based on the U-Net architecture[11], enabling end-to-end training. However, due to the relatively simplified data generation rules, there exists a significant domain gap between the synthesized samples and real-world scenes, which limits the generalization capability of the model. To alleviate the difficulty of acquiring paired data, Qiao et al.[12] proposed an unsupervised generative training framework, which employs a dual-mask prediction mechanism to separately model the light source and flare regions, and incorporates light source information to guide the flare removal process. This enables effective training on unpaired data. Subsequently, Dai et al.[1] constructed the widely used nighttime flare removal dataset Flare7K, and further extended it to Flare7K++[13] by incorporating real captured flare patterns, significantly enhancing the model's adaptability to multiple types of strong scattering degradations. With the Transformer architecture[14] and its variant Swin Transformer[15] gaining widespread attention in image modeling tasks, a series of image restoration networks such as Uformer[16] and Restormer[17] have been developed. These methods have achieved outstanding performance in tasks such as image denoising, deraining, and general image restoration. Building upon this, Zhang et al. proposed FF-Former[18], which introduces the Fast Fourier Convolution (FFC) module to construct Spatial Frequency Blocks (SFB). Through frequency-domain modeling, the method enhances the model's ability to perceive global dependencies and effectively improves restoration quality in nighttime scenes with strong lens flare. In another direction, Kotp et al.[19] proposed a two-stage architecture that integrates depth estimation and image restoration. They utilize the scene depth map predicted by the DPT network as structural guidance and feed it together with the input image into the Uformer[16] network, thereby enhancing the model's ability to distinguish between real image content and flare artifacts. This approach improves image reconstruction accuracy and generalization in real-world scenes, demonstrating the potential of depth-aware guidance in lens flare removal tasks. In addition to multimodal guidance, other advanced paradigms have also shown great potential in the field of image restoration. Among them, diffusion models, which have attracted significant attention in recent years[20], are increasingly being applied to low-level visual tasks due to their powerful generative priors and high-quality sample generation capabilities. For instance, WaveDM[21] innovatively combines wavelet transforms with the diffusion process, enabling more effective restoration of image structure and texture details by denoising across different frequency sub-bands. Furthermore, advanced attention mechanisms have also demonstrated their importance in specific removal tasks. DeSeal[22] serves as a case in point, designing a semantic-aware attention mechanism that can precisely locate and remove seals from document images while preserving the background content.

Although existing methods have achieved certain success, balancing large-scale flare modeling with computational efficiency remains a core challenge in current research. On one hand, Convolutional Neural Networks (CNNs), due to the limitation of local receptive fields, typically require deep stacking or multi-scale strategies to expand their perceptual range, which is inefficient and yields limited effectiveness when dealing with large-area, diffuse flares. On the other hand, some advanced architectures with powerful global modeling capabilities also face computational efficiency bottlenecks. For instance, diffusion models, which are based on an iterative sampling process, often require hundreds to thousands of inference steps to generate high-quality results, leading to extremely high inference latency. Meanwhile, the standard Transformer architecture[14], despite its capability for single-step global modeling and long-range dependency capture, suffers from a self-attention mechanism with $O(N^2)$ quadratic computational complexity, making it difficult to apply to high-resolution images and severely restricting its practical deployment. To reduce computational costs, Swin Transformer[15] restricts attention computation to local windows. While this alleviates the computational bottleneck to some extent, it also sacrifices the global receptive field, making it difficult to perform interaction and modeling across the entire image. To address these issues, this paper proposes a simple multi-domain flare removal network-SMFR-Net (Simple Multi-domain Flare Removal network). The model is designed with simplicity and efficiency in mind, and enhances the receptive field through collaborative modeling in both the frequency and spatial domains. While maintaining a parameter count of only 7.981M, it achieves high-quality image reconstruction and superior performance. The main contributions of this paper are as follows:

- This paper proposes a structurally concise multi-domain architecture for image flare removal–SMFR-Net. The encoder integrates a Frequency Domain Modulation (FDM) module and a Multi-Scale Grouped Dilated Convolution (MGDC) module to achieve joint modeling of frequency and spatial domain features. Additionally, a lightweight Channel-Spatial Attention Module (CSAM) is designed to enhance the model's responsiveness to flare regions. The decoder adopts an asymmetric structure and is simplified at this stage by retaining only the MGDC module and introducing components such as the Simple Channel Attention (SCA) module, effectively controlling model complexity.
- A novel structure-aware composite loss function tailored for flare removal is proposed, which leads to significant improvements in quantitative evaluation metrics.
- **SOTA performance:** Extensive benchmark results demonstrate that the proposed method outperforms existing approaches and exhibits strong generalization ability in real-world scenarios.

Methods

In this section, we provide a detailed introduction to the SMFR-Net we proposed. First, we present its overall encoder-decoder architecture. Then, we delve into the core building block of the network: the Simple Multi-domain Encoder Block (SMEBlock), specifically designed for flare removal, and its key components. Finally, we introduce the simplified decoder module,the Simple Multi-scale Decoder Block (SMDBlock), along with the composite loss function used for optimization.

Overall architecture

As shown in Fig. 2, SMFR-Net adopts a structurally simple encoder-decoder architecture and introduces a global residual learning strategy[23] to stabilize the training process and improve image reconstruction accuracy. Given an input flare image I_{flare}, the network learns a residual mapping function $F(\cdot)$, and the restored clean image I_{clean} is defined as:

$$I_{\text{clean}} = I_{\text{flare}} + F(I_{\text{flare}}) \tag{1}$$

Unlike traditional symmetric architectures, SMFR-Net adopts a task-oriented, modularly differentiated design in its encoder and decoder. The encoder focuses on enhancing feature representation capabilities, while the

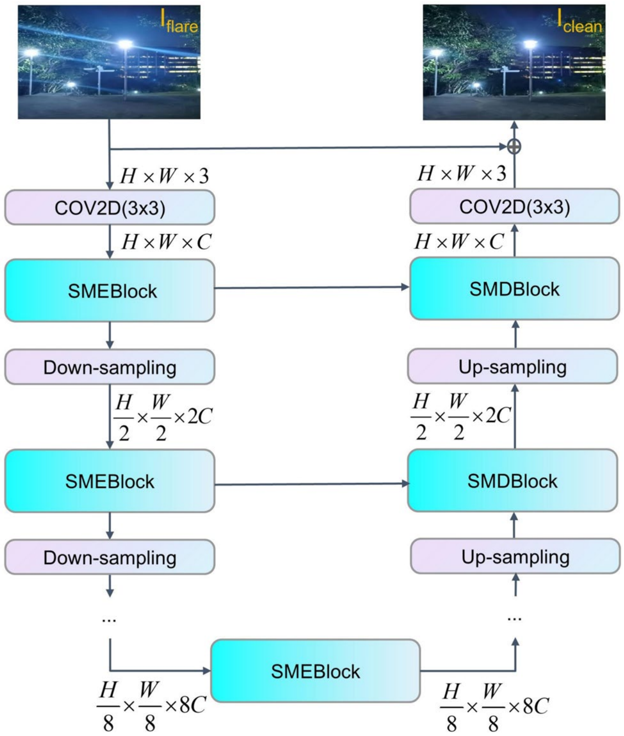

Fig. 2. The overall network architecture of SMFR-Net.

decoder is dedicated to efficient structural reconstruction and detail restoration, thereby achieving a good balance between model performance and computational efficiency. In the encoding path, SMFR-Net stacks multiple SMEBlocks to progressively extract features, with a bottleneck module placed in the middle of the backbone to integrate high-level semantic information. The decoding path, in turn, uses multiple SMDBlocks to gradually restore the spatial structure. Skip connections are introduced to fuse shallow and deep features, effectively mitigating the problems of gradient vanishing and information degradation. Finally, the output features are spatially upsampled via PixelShuffle operations to restore the original image resolution.

SMEBlock

This section mainly introduces the SMEBlock, the core unit in the encoding path of SMFR-Net (as shown in Fig. 3(a)). The SMEBlock is composed of three modules: FDM, MGDC, and CSAM. It adopts a two-stage design: first, it performs a preliminary extraction of global features in the frequency domain through the FDM module; then, it further extracts global features and local details in the spatial domain by combining the MGDC and CSAM modules.

Frequency Domain Modulation module (FDM)

The Fast Fourier Transform (FFT) maps an image from the spatial domain to the frequency domain, such that each frequency component contains the image's global information. Therefore, FFT naturally possesses an infinite theoretical receptive field, which gives it great potential for modeling long-range dependencies and global artifacts (such as large-scale flare). To fully leverage the advantage of global perception in the frequency domain while ensuring model simplicity, we propose a Frequency Domain Modulation (FDM) module that operates entirely in the frequency domain and is applied directly to the input features. As shown in Fig. 3(c), this module aims to perform global modeling on the input features at a low computational cost. The core idea of FDM is to process only the magnitude spectrum, which primarily encodes content and contrast information, while keeping the phase spectrum unchanged to preserve the crucial structural and positional information. The module first applies a 2D Fast Fourier Transform to the input features $X \in \mathbb{R}^{B \times C \times H \times W}$ to obtain their complex frequency-domain representation:

$$F(x) = Me^{j\Phi} \tag{2}$$

where M represents the magnitude spectrum, and Φ is the phase spectrum. Subsequently, we perform adaptive channel re-weighting on the magnitude spectrum M through a simple channel attention mechanism to focus on feature channels with more abundant information. The enhanced magnitude spectrum $\hat{M}$ is calculated as follows:

$$\hat{M} = M \odot \sigma(\text{Conv}_{1\times1}(\text{GAP}(M))) \tag{3}$$

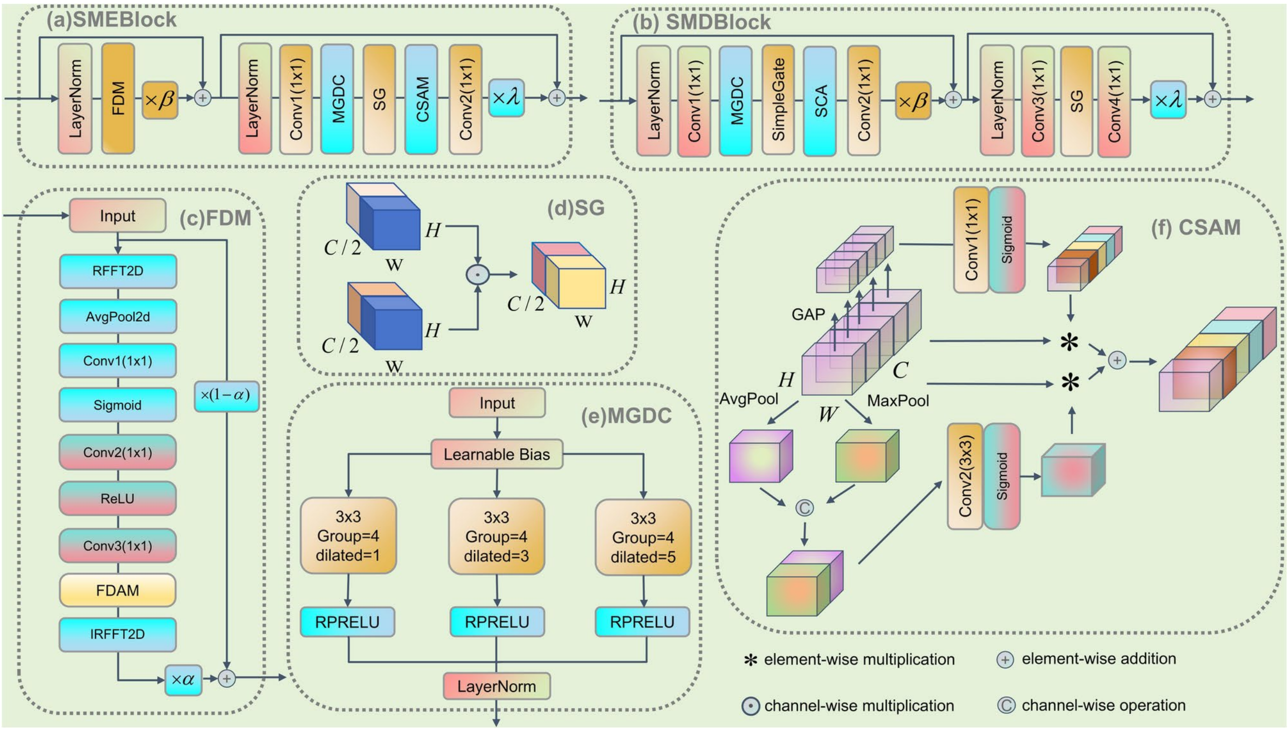

Fig. 3. The structure of core components in SMFR-Net. (**a**) The Simple Multi-domain Encoder Block (SMEBlock). (**b**) The Simple Multi-scale Decoder Block (SMDBlock). (**c**) The Frequency Domain Modulation Module (FDM). (**d**) The Simple Gate Module (SG). (**e**) The Multi-scale Grouped Dilated Convolution Module (MGDC). (**f**) The Channel-Spatial Attention Module (CSAM).

where GAP( ) represents global average pooling, $\mathrm{Conv}_{1\times1}(\cdot)$ refers to a 1×1 convolution, $\odot$ denotes element-wise multiplication, and $\sigma(\cdot)$ represents the Sigmoid function. The enhanced magnitude spectrum $\hat{M}$ is then passed into a lightweight MLP composed of two 1×1 convolutions and a LeakyReLU activation function δ to extract deeper features:

$$M_{\mathrm{processed}} = \mathrm{Conv}_{1\times1}(\delta(\mathrm{Conv}_{1\times1}(\hat{M}))) \tag{4}$$

In order to adaptively adjust the response based on frequency position, we designed the Frequency Distance Adjustment Mechanism (FDAM). This mechanism first defines a fixed, normalized frequency distance map D_{freq}, which represents the distance from each frequency point (u, v) to the spectral center (DC component). Then, a very lightweight convolutional network f_θ is used to learn from this distance map, generating a learnable modulation weight map $W_{freq} = f_\theta(D_{freq})$. The final magnitude spectrum is fine-tuned via the following equation:

$$M_{\mathrm{out}} = M_{\mathrm{processed}} \odot (1 + \gamma \cdot W_{\mathrm{freq}}) \tag{5}$$

where the hyperparameter γ (set to 0.1 in this paper) controls the adjustment strength. Finally, we reconstruct the complex spectrum with the modulated magnitude M_{out} and the original phase Φ, and then transform it back to the spatial domain via the inverse Fourier Transform $\mathscr{F}^{-1}$ to obtain the frequency-enhanced features x_{freq}:

$$x_{\mathrm{freq}} = \mathscr{F}^{-1}(M_{\mathrm{out}} e^{j\Phi}) \tag{6}$$

This feature is then passed through a learnable gating fusion mechanism to be combined with the original input x, yielding the final output of the FDM, x_{out}:

$$x_{\mathrm{out}} = \sigma(\alpha) \cdot x_{\mathrm{freq}} + (1 - \sigma(\alpha)) \cdot x \tag{7}$$

Multi-scale Grouped Dilated Convolution (MGDC)

After processing through the FDM, this paper further designs a multi-scale grouped dilated convolution module to simultaneously expand the receptive field and enhance the perception of local fine-grained details. This module is constructed based on dilated convolution[24] and achieves multi-scale structural information extraction by setting different dilation rates for parallel convolutional branches. The overall structure is shown in Fig. 3(e). The MGDC module first introduces a learnable channel-wise bias term $\beta \in \mathbb{R}^C$ to the features

$X \in \mathbb{R}^{B\times C\times H\times W}$ processed by the FDM, obtaining the shifted features $\tilde{X} = X + \beta$. Subsequently, $\tilde{X}$ is simultaneously sent into three parallel sets of 3×3 grouped dilated convolutions. The dilation rates for these convolutions are set to 1, 3, and 5, respectively, aiming to capture multi-scale structural information from local details to a larger context. To effectively control computational complexity, we set the number of groups for each convolution to a fixed value $g = 4$, which means each convolutional kernel only operates on its assigned channel subset, thereby reducing computational overhead. The outputs of the convolutions are then processed by the RPRELU[25] activation function, fused, and finally passed into a Layer Normalization layer to further enhance feature stability and network convergence speed. The overall expression for this module can be simplified as:

$$Y = \mathrm{LN}\left(\sum_{i=1}^{3} \mathrm{RPRELU}\left(\mathrm{Conv}_{3\times 3}^{(d_i,g)}(X+\beta)\right)\right) \tag{8}$$

where $d_i \in \{1, 3, 5\}$ represents the dilation rate of different branches, and $\mathrm{Conv}_{3\times 3}^{(d_i,g)}$ denotes the convolution operation with dilation rate d_i and number of groups g. After the MGDC processing, we use SimpleGate(SG)[26] to replace the activation function.

Channel-Spatial Attention Module (CSAM)
To enhance the model's perceptual capability for flare regions, this paper proposes a lightweight Channel-Spatial Attention Module (CSAM). As shown in Fig. 3(f), this module combines the lightweight design of SCA[26] with the spatial modeling capability of CBAM[27], employing parallel channel and spatial attention branches. This design ensures computational efficiency while effectively enhancing the modeling ability for flare regions, thereby improving the model's adaptability to complex lighting scenes. The channel attention branch is based on the SCA structure, introducing a 1×1 convolution and a Sigmoid gating mechanism, which makes the response weight of each channel learnable, expressed as:

$$M_c = \sigma(\mathrm{Conv}_{1\times 1}(\mathrm{AvgPool}(X))) \tag{9}$$

The spatial attention branch borrows from the design of CBAM[27]. It employs channel-wise average pooling and max pooling for feature compression. The concatenated results are then passed through a 3×3 convolution to extract spatial saliency for generating the spatial attention map:

$$M_s = \sigma\left(\mathrm{Conv}_{3\times 3}\left([\mathrm{AvgPool}_c(X) \| \mathrm{MaxPool}_c(X)]\right)\right) \tag{10}$$

Finally, the two attention maps are applied to the input feature map through element-wise weighting, yielding the fused output:

$$X' = (X \cdot M_c) + (X \cdot M_s) \tag{11}$$

SMDBlock

Compared to the encoder, which focuses on expanding the receptive field, our decoder design places greater emphasis on feature reconstruction and detail restoration, while also striving for simplicity and efficiency. To this end, we drew inspiration from the block design in NAFNet[26]. This design deconstructs complex processing modules into two more concise core components: a global attention module and a feed-forward network (FFN). Its computation process can be represented by the following equations:

$$z_1 = \mathrm{Attention}(\mathrm{LayerNorm}(x)) + x \tag{12}$$

$$z_2 = \mathrm{FFN}(\mathrm{LayerNorm}(z_1)) + z_1 \tag{13}$$

where x is the input feature and z_2 is the output feature. On this basis, we further incorporate functionally specific enhancement modules to improve the model's performance and adaptability. As shown in Fig. 3(b), the input feature $X \in \mathbb{R}^{B\times C\times H\times W}$ first undergoes normalization and then enters the global branch pathway. Initially, a 1×1 convolution is applied to expand the channel dimension, followed by the MGDC module to achieve multi-scale global contextual modeling.The output is then passed through the SG module[26] and combined with the SCA for channel re-weighting,before a final a 1×1 convolution restores the original dimension, forming the first residual branch.The FFN path adopts the intermediate features after normalization and sequentially applies a 1×1 convolution, the SG module, and another 1×1 convolution to construct the pre-FFN path, forming the second residual branch. These two residual branches are scaled by learnable factors β and λ, respectively, and then added to the input feature X as residual connections to obtain the final output Y. The computation process of the decoder module is formulated as:

$$Y = X + \beta \cdot F_{\mathrm{global}}(X) + \lambda \cdot F_{\mathrm{ffn}}(X) \tag{14}$$

Here, $F_{\mathrm{global}}(\cdot)$ and $F_{\mathrm{ffn}}(\cdot)$ represent the mapping functions of the global and FFN branches, respectively, defined as:

$$F_{\mathrm{global}}(X) = \mathrm{Conv}_{1\times 1}(\mathrm{SCA}(\mathrm{SG}(\mathrm{MGDC}(\mathrm{Conv}_{1\times 1}(\mathrm{LN}(X)))))) \tag{15}$$

$$F_{\text{ffn}}(X) = \text{Conv}_{1\times1}(\text{SG}(\text{Conv}_{1\times1}(\text{LN}(X)))) \tag{16}$$

where $\text{LN}(\cdot)$ denotes 2D Layer Normalization and $\text{Conv}_{1\times1}(\cdot)$ denotes a 1×1 convolution. The learnable scaling factors β and λ are used to balance the contribution weights of the global and FFN branches in the final output.

Loss function design

To enhance the model's reconstruction capability in regions affected by strong light interference and improve overall perceptual quality, we designed a structure-aware composite loss function for the training phase. Specifically, this loss function consists of three components: L1 loss, perceptual loss[23], and multi-scale structural similarity loss (MS-SSIM)[28]. The final loss is defined as follows:

$$\mathscr{L}_{\text{total}} = \lambda_1 \cdot \mathscr{L}_{\text{pixel}} + \lambda_2 \cdot \mathscr{L}_{\text{percep}} + \lambda_3 \cdot \mathscr{L}_{\text{MS-SSIM}} \tag{17}$$

where $\lambda_1 = 0.5$, $\lambda_2 = 0.5$, and $\lambda_3 = 0.2$ are the weighting coefficients for each respective loss term.

We employ the Mean Absolute Error (MAE) as the basic pixel-wise loss, encouraging the network to perform precise reconstruction in the pixel space. It is defined as:

$$\mathscr{L}_{\text{pixel}}(\hat{I}, I) = \frac{1}{N}\sum_{i=1}^{N} |\hat{I}_i - I_i| \tag{18}$$

where $\hat{I}$ and I denote the predicted image and the corresponding ground truth (GT) image, respectively. To enhance the semantic consistency and subjective visual quality of the flare-removed results, we introduce a perceptual loss based on VGG19. This loss extracts features from multiple layers of the predicted and ground truth images, and computes the L1 distance in the feature space:

$$\mathscr{L}_{\text{percep}} = \sum_{l \in \mathscr{L}} w_l \cdot \left\| \phi_l(\hat{I}) - \phi_l(I) \right\|_1 \tag{19}$$

where $\phi_l(\cdot)$ denotes the feature map from the l-th layer of the VGG network, and w_l is the weight for this specific feature layer. In this work, the 2nd, 7th, 12th, 21st, and 30th layers of VGG19 are selected as perceptual layers.

To further enhance texture and structure restoration, the MS-SSIM loss is introduced. It measures the structural similarity between images across multiple scales:

$$\mathscr{L}_{\text{MS-SSIM}} = 1 - \text{MS-SSIM}(\hat{I}, I) \tag{20}$$

where an MS-SSIM value closer to 1 indicates a higher structural similarity. We use SSIM computed at five scales with weighted averaging, where the weights are set to [0.0448, 0.2856, 0.3001, 0.2363, 0.1333].

The composite loss described above collaboratively guides the training process, significantly improving the flare removal performance. Experimental results demonstrate that it outperforms single-loss training strategies.

Experiments

Datasets

To train our flare removal model, we primarily used the Flare7K++[13] dataset. This dataset consists of two parts, Flare7K and Flare-R: Flare7K contains 5,000 simulated scattering flare images and 2,000 simulated reflective flare images, while Flare-R supplements this with 962 real flare patterns. We utilize its dynamic synthesis pipeline to generate paired training samples by randomly selecting backgrounds from the 23,949 natural images in the Flickr24K[29] dataset and sampling flare patterns and their corresponding light sources with equal probability from Flare7K and Flare-R. To enhance the model's adaptability to real-world nighttime scenes, we also additionally introduced 600 real images from FlareReal600[30]. However, this portion of data serves only as a minor supplement, accounting for approximately 2.44% of the total training samples, which exceed 24,000. The vast majority of the training data still originates from Flare7K++.

Prior to training, we apply a series of complex data augmentation operations, with the detailed parameters shown in Table 1, to the images from the Flare7K[13], Flare-R[13], and FlareReal600[30] datasets. The entire process strictly distinguishes the processing for base images and flare images: first, all base images and flare images undergo an initial random gamma correction ($\gamma \sim U(1.8, 2.2)$) and random flips (horizontal or vertical). Subsequently, to simulate diverse flare morphologies, only the flare images are subjected to a series of exclusive geometric and appearance transformations, including random rotation (0° to 360°), translation (up to 50 pixels), scaling (0.8 to 1.1 times), shear ($\pm10^\circ$), and Gaussian blur ($\sigma \sim U(0.1, 3)$), after which they are center-cropped to 256×256. Meanwhile, only the base images are randomly cropped to the same size and are enhanced to simulate the physical characteristics of the sensor by adding Gaussian noise ($\sigma \approx 0.01 \times \chi^2(1)$) and multiplying by a random gain ($g \sim U(0.5, 1.2)$). The processed flare component is added to the enhanced base image to generate the low-quality (LQ) input. Finally, both the LQ image and the enhanced base image, which serves as the ground truth (GT), are subjected to a reverse gamma correction, ultimately forming the training pair $\langle \text{LQ},\ \text{GT} \rangle$ normalized to the range $[0, 1]$.

During the testing phase, we evaluate our model on two standard test sets provided by Flare7K++[13]. The first is the Flare7K++ real test dataset, which contains 100 pairs of real nighttime images captured under diverse

Transformation Type	Transformation Range
Gamma transformation	$\gamma \sim U(1.8, 2.2)$
Rotation	$\theta \sim U(0, 360^\circ)$
Translation	$t \sim U(-50\text{ px}, 50\text{ px})$
Shear	$\alpha \sim U(-10^\circ, 10^\circ)$
Scaling	$s \sim U(0.8, 1.1)$
Blurring (Gaussian)	$\sigma \sim U(0.1, 3)$
Flip	Horizontal or vertical (random)
RGB gain	$g \sim U(0.5, 1.2)$
Gaussian noise	$\sigma \approx 0.01 \times \chi^2(1)$

Table 1. Training data augmentation parameters.

lighting conditions and flare patterns. The second is the Flare7K++ synthetic test dataset, which we use to further verify the model's generalization ability on synthetically generated flare images. In addition to these public datasets, we captured our own real-world nighttime scenes using an iPhone 15 Pro and a Xiaomi 13 smartphone. This allows us to evaluate the robustness and practicality of our model on unlabeled images from real-world scenarios.

Training settings

The entire training process is conducted on a single NVIDIA TITAN RTX GPU with 24 GB of memory. Our model is an image restoration network based on an encoder-decoder architecture, where the encoder and decoder are composed of [1, 2, 3] and [3, 1, 1] residual blocks, respectively. We train the model using the AdamW optimizer with an initial learning rate of 1×10^{-3}, a batch size of 8, and for a total of 100,000 steps. To enhance stability during the initial training phase, we employ a warm-up strategy for the first 5,000 steps, gradually increasing the learning rate. Subsequently, we use the MultiStepLR scheduler to decay the learning rate by a factor of 0.5 at three predefined milestones (30k, 60k, and 90k steps). This strategy helps mitigate training fluctuations and improves the final convergence accuracy.

Evaluation metrics

Most existing flare removal methods evaluate on images with a resolution of 512×512. To ensure a fair and consistent comparison, all test images are uniformly cropped or resized to 512×512 and normalized to the range [0, 1]. We adopt three widely-used metrics to evaluate image restoration quality: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM)[31], and Learned Perceptual Image Patch Similarity (LPIPS)[32]. To more comprehensively assess the model's performance in removing different types of flare components, we also introduce two local evaluation metrics proposed by Dai et al.[13]: S-PSNR and G-PSNR. These metrics independently evaluate the regions of strong glare and stripe diffusion, respectively. In addition to these quantitative metrics, we report the number of parameters and FLOPs for each model to assess their computational cost. We also conduct a qualitative analysis on real-world nighttime images to demonstrate the model's ability to suppress complex lighting interference while preserving structural details.

Results

To comprehensively verify the effectiveness and superiority of the proposed SMFR-Net in the image deglare task, we selected several representative existing methods for performance comparison. These methods cover traditional image enhancement techniques, direct glare removal methods, and various recently proposed end-to-end image restoration network architectures. Specifically, the comparison methods include: the glare removal method proposed by Wu[10]; the end-to-end restoration network proposed by Dai[13]; the nighttime lighting enhancement method proposed by Sharma[33]; several well-known image restoration networks trained on the Flare7K++ or Flare7K datasets, such as U-Net[11], HINet[36], MPRNet[35], Restormer[17], Uformer[16], and NAFNet[26]; the Uformer+ND method based on depth estimation by Kotp and Torki[19]; as well as recent state-of-the-art methods SPDDNet[37] and LPFSformer[38], ensuring a comprehensive comparison. Detailed evaluation results can be found in Tables 2 and 4.

The results indicate that SMFR-Net exhibits leading performance on both the real and synthetic test sets of Flare7K++, significantly outperforming most mainstream methods. Compared to the current state-of-the-art methods, SMFR-Net improves PSNR, G-PSNR, and S-PSNR by 0.114 dB, 0.048 dB, and 0.065 dB, respectively, on the real test set. It is worth noting that most models in Table 2 were trained only on the Flare7K++ dataset, whereas SMFR-Net incorporates both Flare7K++ and FlareReal600 to leverage additional diversity. To isolate the impact of data differences and objectively validate the effectiveness of the model architecture itself, we conducted a fair comparison by training a version of SMFR-Net using only the Flare7K++ dataset. As shown in Table 3, while the absence of diversity from FlareReal600 resulted in a slight decrease in some metrics compared to the dual-dataset training, this version of SMFR-Net still significantly outperforms most mainstream methods.

On the synthetic test set, SMFR-Net also demonstrates a clear advantage, achieving a PSNR of 30.276, which surpasses the Uformer and Kotp methods by 0.778 dB and 0.703 dB, respectively. Furthermore, the SSIM score

Dataset		**Flare7K++ real test dataset**						
Metrics		**PSNR↑**	**SSIM↑**	**LPIPS↓**	**G-PSNR↑**	**S-PSNR↑**	**Params (M)**	**MACs (G)**
Input		**22.561**	**0.856**	**0.0777**	**19.555**	**13.104**	-	-
Previous Synthesis Pipelines	FF-Former†[18]	27.350	0.901	0.0440	-	-	-	-
	Sharma[33]	20.492	0.826	0.1115	17.790	12.685	22.365	285.12
	Wu[10]	24.613	0.871	0.0598	21.772	16.728	34.526	261.901
	Flare7K[1]	26.978	0.890	0.0466	23.507	21.563	20.429	159.643
Flare7K++	Zhou et al.[34]	25.184	0.872	0.0548	22.112	20.543	20.628	327.347
	Restormer*[17]	27.597	0.897	0.0447	23.828	22.452	*2.981*	*57.975*
	MPRNet*[35]	27.036	0.893	0.0481	23.490	22.267	3.642	567.187
	U-net[11]	27.189	0.894	0.0452	23.527	22.647	34.527	261.953
	NAFNet[26]	27.042	0.888	0.0556	24.098	22.459	67.788	252.314
	Uformer[16]	27.633	0.894	0.0428	23.949	22.603	20.601	164.361
	HINet[36]	27.548	0.892	0.0464	24.081	22.907	88.674	685.127
	Kotp and Torki[19]	27.662	0.897	0.0422	23.987	22.847	129.306	271.419
	SPDDNet[37]	28.033	0.903	0.0420	24.537	23.614	25.620	105.010
	LPFSformer[38]	*28.238*	*0.905*	0.0422	*24.793*	*23.876*	13.733	525.442
Flare7K++ FlareReal600	**SMFR-Net-L (ours)**	28.225	**0.907**	*0.0403*	24.760	23.832	**2.152**	**31.228**
	SMFR-Net (ours)	**28.352**	**0.907**	**0.0384**	**24.841**	**23.941**	7.981	103.888

Table 2. Comparison results on the Flare7K++ real test dataset. Best results are highlighted in **Bold**, second-best in *Italic*. * denotes models with reduced parameters due to limited GPU memory. † indicates methods without released code, for which metrics are reported from the original paper and may be incomplete. Note: The average inference time per image for SMFR-Net and SMFR-Net-L is 0.0825 s and 0.0412 s, respectively, measured on an NVIDIA TITAN RTX (24 GB) GPU.

Dataset		**Flare7K++ real test dataset**						
Metrics		**PSNR↑**	**SSIM↑**	**LPIPS↓**	**G-PSNR↑**	**S-PSNR↑**	**Params (M)**	**MACs (G)**
Input		**22.561**	**0.856**	**0.0777**	**19.555**	**13.104**	-	-
Flare7K++	**SMFR-Net-L (ours)**	28.223	*0.906*	0.0405	24.763	23.802	**2.152**	**31.228**
	SMFR-Net (ours)	**28.354**	**0.907**	*0.0389*	*24.831*	*23.938*	7.981	103.888
Flare7K++ FlareReal600	**SMFR-Net-L (ours)**	28.225	**0.907**	0.0403	24.760	23.832	**2.152**	**31.228**
	SMFR-Net (ours)	*28.352*	**0.907**	**0.0384**	**24.841**	**23.941**	7.981	103.888

Table 3. Performance of SMFR-Net trained on Flare7K++ vs. Flare7K++ with FlareReal600. Best results are highlighted in **bold** and second best in *italic*

Dataset	**Flare7K++ synthetic test dataset**				
Metrics	**PSNR↑**	**SSIM↑**	**LPIPS↓**	**G-PSNR↑**	**S-PSNR↑**
Input	**22.561**	**0.856**	**0.0777**	**19.555**	**13.104**
NAFNet[26]	27.818	0.946	0.0333	23.388	22.267
Kotp and Torki[19]	29.573	0.961	0.0205	24.879	24.458
Uformer[16]	29.498	*0.962*	0.0210	24.686	24.115
SMFR-Net-L (ours)	*29.649*	*0.962*	*0.0199*	*24.914*	*24.793*
SMFR-Net (ours)	**30.276**	**0.966**	**0.0177**	**25.561**	**25.545**

Table 4. Comparison results on the Flare7K++ synthetic test dataset. All models were trained on the combined Flare7K++ and FlareReal600 datasets. Best results are highlighted in **bold**, second-best in *italic*.

increases to 0.966, while the LPIPS value decreases to 0.0177. These results strongly indicate that SMFR-Net is more effective at restoring image details in regions affected by strong glare and streak diffusion.

At the same time, SMFR-Net demonstrates excellent computational efficiency. The model only requires 7.981M parameters and 103.888G FLOPs, maintaining superior performance while significantly reducing computational costs compared to high-complexity networks, such as Kotp (129.306M / 271.419G) and HINet (88.674M / 685.127G).It is worth noting that, due to the high computational cost of the original MPRNet and Restormer models, this study adopted their lightweight versions for comparison. At the same time, we also proposed our own lightweight model–SMFR-Net-Light (SMFR-Net-L)–whose network width is 32, with parameters and FLOPs of 2.152M and 31.228G, respectively. Although the performance of this model is slightly lower than the full version of SMFR-Net, it still significantly outperforms other comparison methods.

In qualitative analysis, SMFR-Net also demonstrates clear and stable visual results on the Flare7K++ real test images (Fig. 4) and the self-collected validation set (Fig. 5), particularly excelling in restoring details in large-scale glare regions, further confirming its practical application value and structural preservation capability.

Ablation study

In addition to validating the overall performance of the backbone model, we conducted a series of ablation experiments to explore the impact of individual components and training strategies. The first group of experiments mainly focuses on evaluating the effectiveness of the designed modules, specifically MGDC, FDM, CSAM, and the structure-aware composite loss function. To achieve this, we systematically removed each individual module or loss term, built corresponding control models, and compared their performance with the full SMFR-Net on the Flare7K++ real test dataset. The results are shown in Fig. 6 and Table 5.

Among all the comparison models, the complete SMFR-Net consistently outperforms in all metrics, fully demonstrating the synergistic effect of frequency domain modeling, dilated convolutions, and attention mechanisms in improving glare region modeling and image structure restoration. Furthermore, the structure-aware composite loss function significantly enhances perceptual consistency and subjective visual quality.

The second set of experiments aims to explore the impact of different encoder-decoder combinations on model performance and complexity. We construct multiple combinations using SMEBlock, NAFBlock, and SMDBlock, and compare them with the final model architecture (SMFR-Net). To improve training efficiency, the channel number was reduced from 64 to 16 during experiments, so the overall performance is slightly lower than the full configuration. As shown in Table 6, SMFR-Net (ours), which adopts the SMEBlock + SMDBlock combination under the full configuration, achieves the best performance, outperforming all other combinations across multiple key metrics, while maintaining a good balance between performance and efficiency with 7.981M parameters and 103.888G FLOPs.

In comparison, SMEBlock + NAFBlock exhibits a slight advantage in G-PSNR (24.279), but falls short of SMFR-Net in PSNR, LPIPS, and other subjective and objective metrics; while All NAFBlock, despite having the lowest parameter count (6.439M) and computation (92.735G), shows a significant drop in performance. In summary, the combination of SMEBlock and SMDBlock can more effectively model image structures and restore details under strong light interference, achieving an ideal balance between performance and complexity, and validating its rationality and superiority as the final backbone architecture.

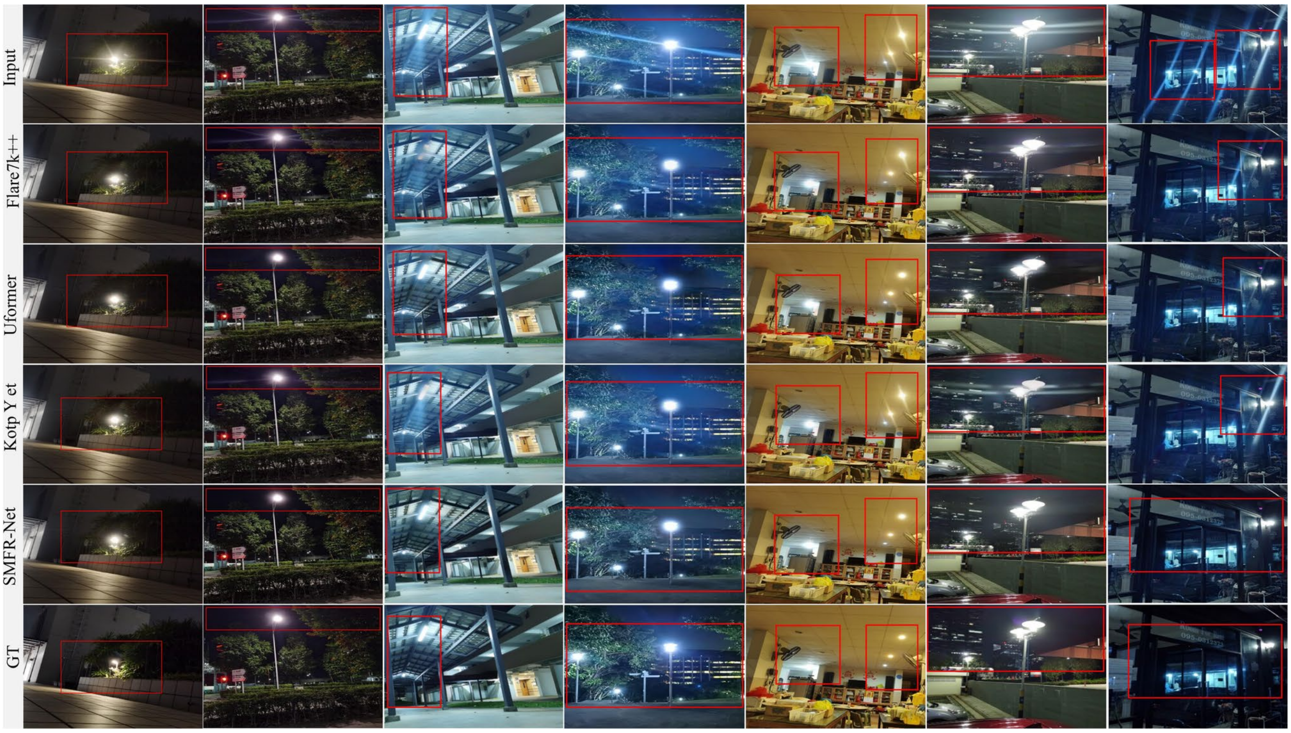

Fig. 4. Visual comparison of flare removal results by different methods on the Flare7K++ real test dataset. Red boxes highlight significant differences in artifact suppression among the methods. Under challenging conditions with multiple strong light sources, most existing methods struggle to effectively remove large-area flare artifacts. In contrast, SMFR-Net significantly suppresses glare interference and restores clear structural details, demonstrating superior capability in large-receptive-field modeling and image restoration.

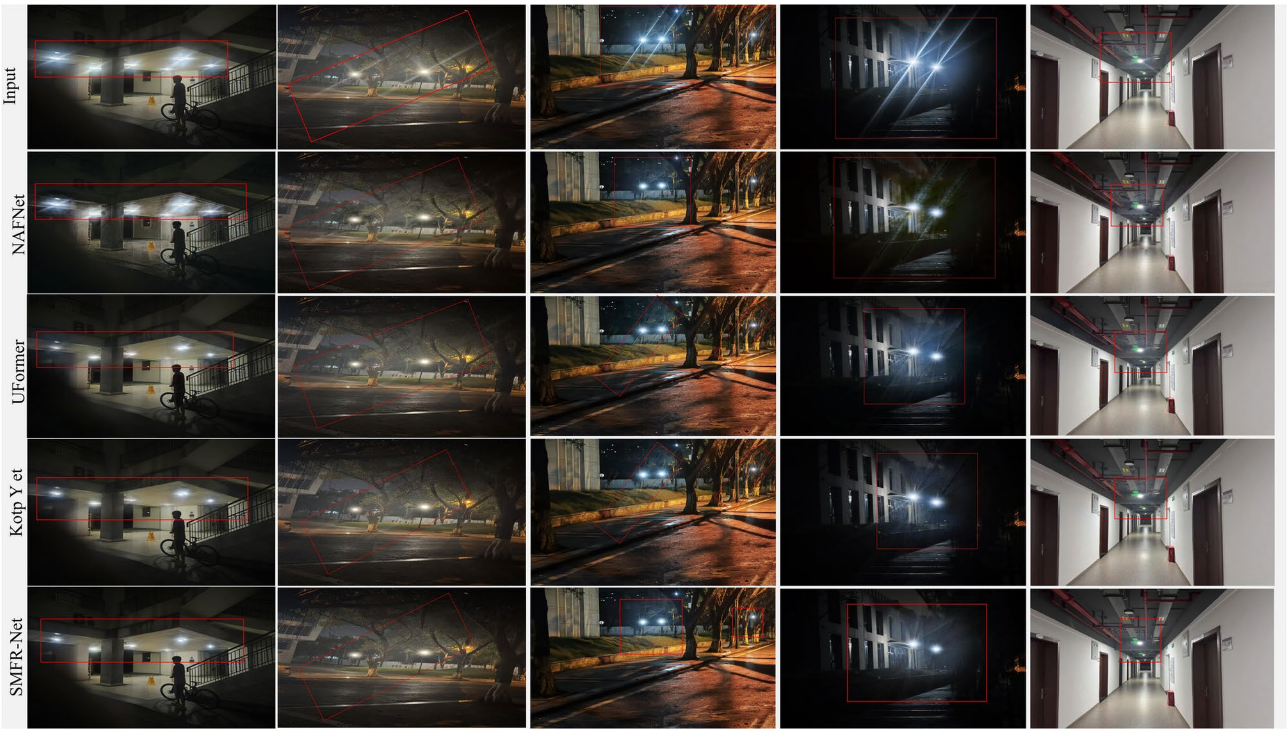

Fig. 5. Visual comparison of flare removal results on real-world images captured by Xiaomi 13 and iPhone 15 Pro. SMFR-Net is compared with NAFNet, Uformer, and Kotp et al.[19] under various challenging lighting conditions. Red boxes highlight regions with significant visual differences, where SMFR-Net achieves the best glare removal performance.

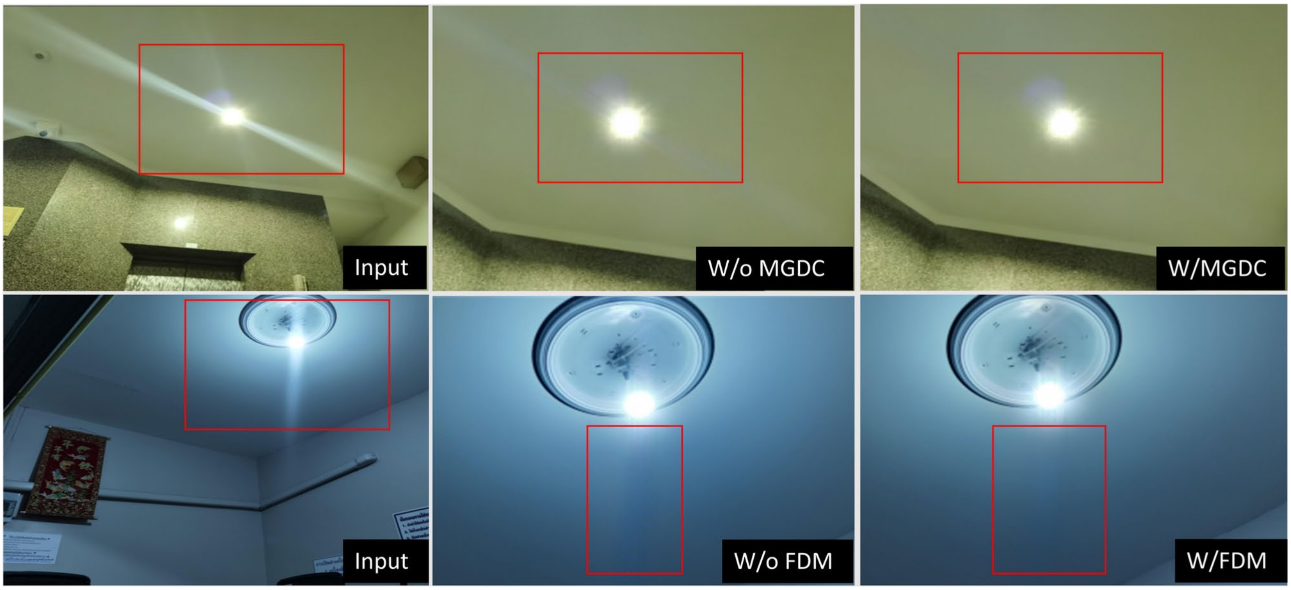

Fig. 6. Visual comparison of flare artifacts with and without MGDC and FDM modules. Red boxes highlight regions where the absence of each module leads to more pronounced flare artifacts, confirming their effectiveness in suppression.

In the third group of experiments, we compare different combinations of loss functions to evaluate their impact on model performance. Specifically, we conducted a systematic ablation study to analyze each loss component and its corresponding weighting coefficient. As shown in Table 7, the experimental results illustrate our step-wise process to determine the final configuration. We begin with a baseline model using only L1 loss ($\lambda_1 = 1.0$). Upon introducing the perceptual loss, we drew from successful practices in the image deglaring

Models	PSNR↑	SSIM↑	LPIPS↓	G-PSNR	S-PSNR↑	Params (M)	MACs (G)
w/o FDM	28.141	0.901	0.0406	24.854	23.247	**5.490**	88.423
w/o CSAM	28.202	0.903	0.0387	24.735	23.694	7.484	103.831
w/o MGDC	27.912	0.899	0.0436	24.675	23.057	7.438	**85.659**
w/o Loss	28.178	0.904	0.0430	24.753	23.016	7.981	103.888
SMFR-Net (ours)	**28.352**	**0.907**	**0.0384**	**24.841**	**23.941**	7.981	103.888

Table 5. Ablation study of key modules and the loss function in SMFR-Net on the Flare7K++ real test dataset. Best results are highlighted in bold.

Models	PSNR↑	SSIM↑	LPIPS↓	G-PSNR↑	S-PSNR↑	Params (M)	MACs (G)
SMEBlock NAFBlock	27.833	0.896	0.0438	24.279	22.345	7.777	95.885
NAFBlock SMDBlock	27.691	0.895	0.0456	24.028	22.516	7.081	95.676
All SMDBlock	27.577	0.896	0.0436	23.997	22.398	6.981	110.963
All SMEBlock	27.835	0.896	0.0435	**24.370**	22.740	8.420	98.826
All NAFBlock	27.317	0.893	0.0464	24.027	21.891	**6.439**	**92.735**
SMFR-Net (ours)	**27.925**	**0.901**	**0.0407**	24.204	**22.894**	7.981	103.888

Table 6. Comparison of different encoder–decoder combinations on model performance and complexity on the Flare7K++ real test dataset. Best results are highlighted in bold.

Loss Components	Weights ($\lambda_1, \lambda_2, \lambda_3$)	PSNR↑	SSIM↑	LPIPS↓	G-PSNR↑	S-PSNR↑
L_{pixel}	(1.0, –, –)	28.178	0.904	0.0430	24.753	23.016
$L_{\text{pixel}} + L_{\text{percep}}$	(0.5, 0.5, –)	28.207	0.904	0.0401	24.670	23.296
$L_{\text{pixel}} + L_{\text{percep}} + L_{\text{MS-SSIM}}$	(0.5, 0.5, 0.5)	28.105	0.894	0.0420	24.630	23.614
$L_{\text{pixel}} + L_{\text{percep}} + L_{\text{MS-SSIM}}$**(ours)**	**(0.5, 0.5, 0.2)**	**28.352**	**0.907**	**0.0380**	**24.841**	**23.941**

Table 7. Ablation study on loss function components and associated weights on the Flare7K++ real test dataset. Best results are in bold.

field for balancing pixel-level fidelity with perceptual quality, such as in Flare7K++[13], and adopted a balanced weighting of $\lambda_1 = 0.5$ and $\lambda_2 = 0.5$. As the data in the table shows, this combination improves PSNR while reducing LPIPS from 0.0430 to 0.0401. The next step is to introduce and fine-tune the MS-SSIM loss. The experiment shows that assigning a high weight ($\lambda_3 = 0.5$) leads to a decrease in all metrics (e.g., PSNR drops to 28.105 and SSIM to 0.894). Therefore, by reducing its weight to $\lambda_3 = 0.2$, the model achieves the best values across all evaluation metrics, with a PSNR of 28.352 and an SSIM of 0.907. This study indicates that our final weighting coefficients ($\lambda_1 = 0.5, \lambda_2 = 0.5, \lambda_3 = 0.2$) are a well-justified combination of established practices and empirical fine-tuning.

Furthermore, we conducted an ablation study on the CSAM module to evaluate the role of its spatial attention (SA) mechanism in the deglaring task. As shown in Table 8, the results reveal a noteworthy phenomenon: on the Flare7K++ real test set, while removing the spatial attention branch led to a slight increase in PSNR (from 28.352 to 28.463), the perceptually-oriented metrics, such as LPIPS, G-PSNR, and S-PSNR, all exhibited a significant decline. We attribute this seemingly contradictory result to a trade-off where the model sacrifices structural details for a lower pixel-level mean squared error. For instance, G-PSNR decreased from 24.841. to 24.753, indicating that the model's global modeling capability for handling strong light interference was significantly compromised. On the Flare7K++ synthetic test dataset, this trend becomes even more pronounced: the model with spatial attention preserved outperforms the version without it in multiple metrics, including PSNR (30.276), SSIM (0.966), LPIPS (0.0177), and G-PSNR (25.561), further confirming the critical role of the spatial attention mechanism in structural perception and detail restoration.

Additional analyses

To validate the applicability and effectiveness of the proposed glare removal method in real-world visual tasks, we conducted experimental evaluations on two representative tasks: semantic segmentation and object detection.

For semantic segmentation, we employed the Segment Anything Model (SAM) proposed by Meta AI[5]. This model possesses zero-shot segmentation capability, enabling high-quality segmentation without fine-tuning, and is suitable for various visual scenarios. As shown in Fig. 7, we input the original image, the image processed by SMFR-Net, and the glare-free ground truth (GT) image into the SAM model to generate the corresponding semantic segmentation results. The results show that the strong glare region in the original image

Dataset	Models	PSNR↑	SSIM↑	LPIPS↓	G-PSNR↑	S-PSNR↑
Flare7K++ real test dataset	w/o SA	**28.463**	**0.907**	0.0381	24.753	23.694
	CSAM (full)	28.352	**0.907**	**0.0380**	**24.841**	**23.941**
Flare7K++ synthetic test dataset	w/o SA	29.895	0.962	0.0195	25.083	24.846
	CSAM (full)	**30.276**	**0.966**	**0.0177**	**25.561**	**25.545**

Table 8. Comparison results of CSAM (full) vs. w/o SA on Flare7K++ test datasets. Best results are highlighted in bold.

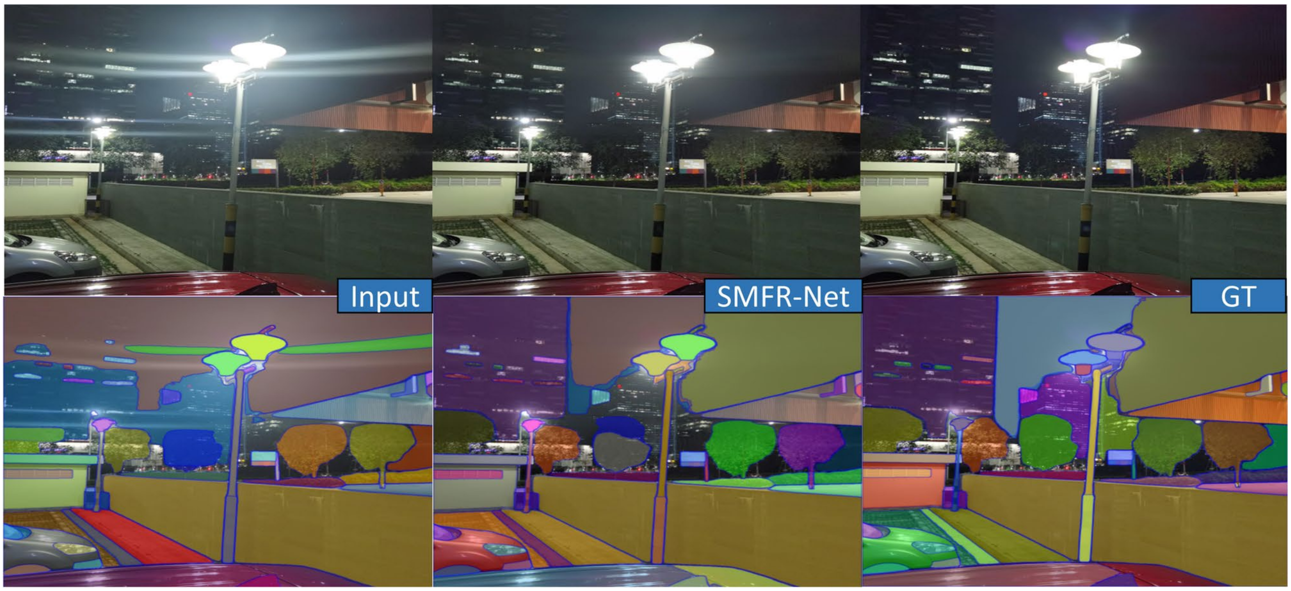

Fig. 7. Semantic segmentation results using SAM before and after SMFR-Net processing. SMFR-Net effectively suppresses glare, enabling more accurate segmentation boundaries and better alignment with ground truth results.

was misidentified as a semantic object, resulting in segmentation errors; whereas after SMFR-Net processing, the glare regions were effectively removed, image structural boundaries became clearer, and semantic partitioning was more accurate and closely aligned with the GT. This verifies the effectiveness of our method in restoring true semantic structures.

For object detection, we selected the medium-scale variant YOLOv11m of the YOLOv11 model[39] for inference evaluation. This model achieves relatively high detection accuracy while maintaining good inference speed, making it suitable for multi-object detection tasks in complex nighttime environments. As shown in Fig. 8, strong light interference in the original image significantly affects the model's perception ability, leading to missed detections or low confidence scores. For example, in the Input4 scene, due to glare occlusion, the motorcycle was detected as "motorcycle" with a confidence score of only 0.56. In contrast, in the image processed by SMFR-Net, the glare interference was effectively suppressed, the confidence of "motorcycle" increased to 0.76, and the detection bounding box aligned better with the object edges.

To further quantify the changes in detection performance, we collected confidence differences of eight targets across four scenes before and after processing. The results indicate that after SMFR-Net processing, target confidence scores generally improved, with an average increase of 0.089, further validating the applicability and effectiveness of our method in real-world visual tasks.

Limitation

Although the SMFR-Net proposed in this paper demonstrates effectiveness across various scenarios, as a model based on supervised learning, its performance is still limited in certain extreme cases. When the scale and intensity of the flare cause the underlying texture and structural information in vast regions of an image to be completely occluded, the model's restoration capability is affected. In situations of complete information loss, a supervised learning model struggles to reconstruct complex, scene-consistent details due to the lack of effective input cues, and its output tends to converge towards blurry or overly-smooth results. To address this challenge, a research direction worth exploring is the combination of the efficient SMFR-Net architecture with generative models. By leveraging the prior knowledge of generative models, it is expected to enable plausible generative inpainting for regions with complete information loss, thereby enhancing the model's restoration capabilities in extreme degradation scenarios.

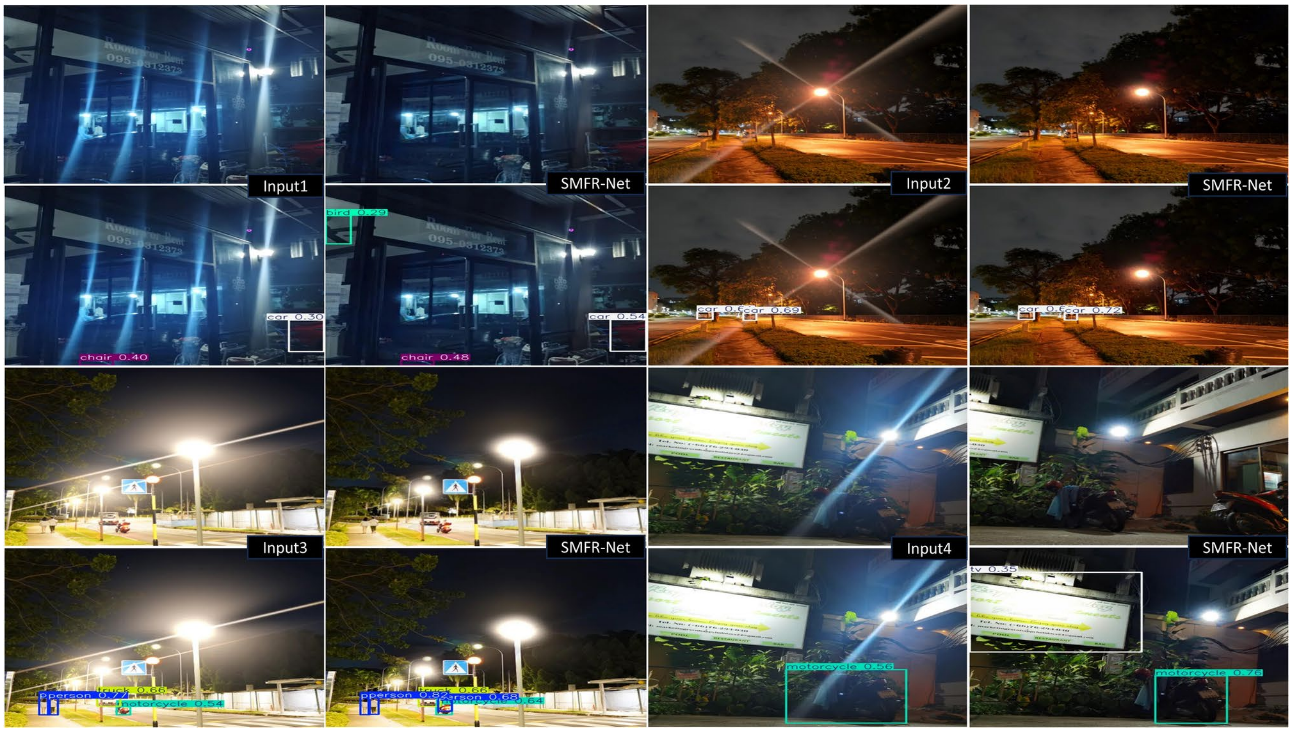

Fig. 8. YOLOv11m detection results before and after SMFR-Net processing. SMFR-Net reduces glare interference, improving detection confidence and bounding box accuracy across four nighttime scenes.

Conclusion

This paper proposes a multi-domain flare removal network–SMFR-Net, aiming to provide a simple and efficient solution. By designing an encoder for multi-domain modeling and a decoder for efficient restoration, SMFR-Net expands the receptive field while maintaining a low computational cost. SMFR-Net contains only 7.981M parameters and achieves significant removal effects in various scenarios, demonstrating its advantages in performance. Furthermore, this paper also proposes a lightweight version of SMFR-Net–SMFR-Net-L, containing only 2.152M parameters, which effectively reduces the computational burden and is more suitable for resource-constrained devices. Experimental results show that although the performance of the lightweight version is slightly lower than the full version, SMFR-Net-L still exhibits good results in the flare removal task and surpasses most existing methods on both the Flare7K++ test set and in real-world application scenarios.

Data availability

The datasets used in this study are publicly available. Flare7K++ is available at https://github.com/ykdai/Flare7K under the S-Lab License 1.0. FlareReal600 is available at https://github.com/Zdafeng/FlareReal600.

Received: 22 July 2025; Accepted: 19 September 2025
Published online: 24 October 2025

References

1. Dai, Y., Li, C., Zhou, S., Feng, R. & Loy, C. C. Flare7k: A phenomenological nighttime flare removal dataset. *Adv. Neural Inf. Process. Syst.* **35**, 3926–3937 (2022).
2. Thoma, M. A survey of semantic segmentation. *arXiv preprint* arXiv:1602.06541 (2016).
3. Zou, Z., Chen, K., Shi, Z., Guo, Y. & Ye, J. Object detection in 20 years: A survey. *Proc. IEEE* **111**, 257–276 (2023).
4. Ranftl, R., Bochkovskiy, A. & Koltun, V. Vision transformers for dense prediction. *ArXiv preprint* arXiv:2103.13413(2021).
5. Kirillov, A. et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026 (2023).
6. Chabert, F. *Automated lens flare removal* (Stanford University, In Technical report (Department of Electrical Engineering, 2015).
7. Vitoria, P. & Ballester, C. Automatic flare spot artifact detection and removal in photographs. *J. Math. Imaging Vis.* **61**, 515–533 (2019).
8. Asha, C., Bhat, S. K., Nayak, D. & Bhat, C. Auto removal of bright spot from images captured against flashing light source. In *2019 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, 1–6 (IEEE, 2019).
9. Seibert, J. A., Nalcioglu, O. & Roeck, W. Removal of image intensifier veiling glare by mathematical deconvolution techniques. *Med. Phys.* **12**, 281–288 (1985).
10. Wu, Y. et al. How to train neural networks for flare removal. In *Proceedings of the IEEE/CVF international conference on computer vision*, 2239–2247 (2021).
11. Ronneberger, O., Fischer, P. & Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241 (Springer, 2015).
12. Qiao, X., Hancke, G. P. & Lau, R. W. Light source guided single-image flare removal from unpaired data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4177–4185 (2021).

13. Dai, Y. et al. Flare7k++: Mixing synthetic and real datasets for nighttime flare removal and beyond. *IEEE Trans. Pattern Anal. Mach. Intell.* **46**, 7041–7055 (2024).
14. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017).
15. Liu, Z. et al. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022 (2021).
16. Wang, Z. et al. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 17683–17693 (2022).
17. Zamir, S. W. et al. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5728–5739 (2022).
18. Zhang, D. et al. Ff-former: Swin fourier transformer for nighttime flare removal. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2824–2832 (2023).
19. Kotp, Y. & Torki, M. Flare-free vision: Empowering uformer with depth insights. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2565–2569 (IEEE, 2024).
20. Huang, Y. et al. Diffusion model-based image editing: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **47**(6), 4409-4437. (2025).
21. Huang, Y. et al. Wavedm: Wavelet-based diffusion models for image restoration. *IEEE Trans. Multimed.* **26**, 7058–7073 (2024).
22. Liu, Y., Huang, J. & Chen, S. Deseal: Semantic-aware seal2clear attention for document seal removal. *IEEE Signal Process. Lett.* **30**, 1702–1706 (2023).
23. Johnson, J., Alahi, A. & Fei-Fei, L. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, 694–711 (Springer, 2016).
24. Yu, F. & Koltun, V. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122 (2015).
25. Liu, Z., Shen, Z., Savvides, M. & Cheng, K.-T. Reactnet: Towards precise binary neural network with generalized activation functions. In *European conference on computer vision*, 143–159 (Springer, 2020).
26. Chen, L., Chu, X., Zhang, X. & Sun, J. Simple baselines for image restoration. In *European conference on computer vision*, 17–33 (Springer, 2022).
27. Woo, S., Park, J., Lee, J.-Y. & Kweon, I. S. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, 3–19 (2018).
28. Wang, Z., Simoncelli, E. P. & Bovik, A. C. Multiscale structural similarity for image quality assessment. In *The thrity-seventh asilomar conference on signals, systems & computers, 2003*, vol. 2, 1398–1402 (Ieee, 2003).
29. Zhang, X., Ng, R. & Chen, Q. Single image reflection separation with perceptual losses. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4786–4794 (2018).
30. Zhang, D. Flarereal600: Real-captured paired dataset for nighttime flare removal. https://github.com/Zdafeng/FlareReal600 (2024). Accessed: 2025-07-2.
31. Wang, Z., Bovik, A. C., Sheikh, H. R. & Simoncelli, E. P. Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**, 600–612 (2004).
32. Zhang, R., Isola, P., Efros, A. A., Shechtman, E. & Wang, O. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595 (2018).
33. Sharma, A. & Tan, R. T. Nighttime visibility enhancement by increasing the dynamic range and suppression of light effects. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognitio*n, 11977–11986 (2021).
34. Zhou, Y. et al. Improving lens flare removal with general-purpose pipeline and multiple light sources recovery. In *Proceedings of the IEEE/CVF international conference on computer vision*, 12969–12979 (2023).
35. Zamir, S. W. et al. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 14821–14831 (2021).
36. Chen, L., Lu, X., Zhang, J., Chu, X. & Chen, C. Hinet: Half instance normalization network for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 182–192 (2021).
37. Qi, K., Wang, B. & Liu, Y. A self-prompt based dual-domain network for nighttime flare removal. *Eng. Appl. Artif. Intell.* **144**, 110103 (2025).
38. Chen, G.-Y. et al. Lpfsformer: Location prior guided frequency and spatial interactive learning for nighttime flare removal. *IEEE Trans. Circuits Syst. Video Technol. 35*(4), 3706-3718. (2024).
39. Khanam, R. & Hussain, M. Yolov11: An overview of the key architectural enhancements. *arXiv preprint* arXiv:2410.17725 (2024).

Author contributions

Shaofeng Liu conceived the study, conducted the experiments, and contributed to writing. Guorong Chen supervised the project, identified issues, and revised the manuscript. Weijie Zhang assisted in writing. Qingru Zhang contributed to the evaluation and visualization. Jinmei Zhang and Jian Wang reviewed the manuscript. All authors approved the final manuscript.

Funding

This work was supported by the Graduate Student Innovation Program of Chongqing University of Science and Technology (grant no. YKJCX2421505) and the Chongqing Technology Innovation and Application Development Project (grant no. 2023TIAD-GPX0007).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to G.C.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by-nc-nd/4.0/.

© The Author(s) 2025, corrected publication 2026