IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 4 May 2023, accepted 27 May 2023, date of publication 5 June 2023, date of current version 14 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3282839

RESEARCH ARTICLE

Enhancing Rainy Weather Driving: Deep Unfolding Network With PGD Algorithm for Single Image Deraining

CHAO HU[1] AND HUIWEI WANG[2,3]

[1]College of Electronic and Information Engineering, Southwest University, Chongqing 400715, China
[2]Key Laboratory of Intelligent Information Processing, Chongqing Three Gorges University, Chongqing 404100, China
[3]Chongqing Innovation Center, Beijing Institute of Technology, Chongqing 401120, China

Corresponding author: Huiwei Wang (hwwang@swu.edu.cn)

This work was supported in part by the China Postdoctoral Science Foundation under Grant 2022M720453, and in part by the Science and Technology Research Program of Chongqing Municipal Education Commission under Grant KJZD-M202201204.

ABSTRACT Although deep learning enables excellent visual perception performance for autonomous driving, its robustness to heavy weather is still worthy of attention since it is prone to forgetting previously learned information. In this paper, we focus on the deraining task from images based on single images of street scenes to improve the perception of autonomous driving in the rain, in which we degrade the rain image to a clean background image by using a deep unfolding network (DUN) combined with the proximal gradient descent (PGD) algorithm and introducing a gradient estimation strategy and a proximal mapping module. In the gradient descent module, we flexibly perform gradient descent on complex images by selectively replacing the degradation matrix. And in the proximal mapping module, we introduce an internal feature fusion module to fuse each stage's local and global features to improve feature extraction efficiency, and an inter-stage feature fusion module to fuse each stage with the condensed features of the previous stage to reduce information loss during iteration. Finally, we evaluated our method on a synthetic dataset and also utilized real complex rain images for qualitative analysis. In addition, we combined high-level perception tasks, i.e., target detection and semantic segmentation, for autonomous driving to compare the perceptual effectiveness of autonomous driving before and after removing the rain. Experimental results demonstrate that our model not only outperforms existing efficient rain removal networks and produces a noticeable improvement in visual quality, but also significantly enhances the perception performance of autonomous driving in rainy weather for both the combined target detection task and the semantic segmentation task.

INDEX TERMS Deraining, PGD, deep unfolding network, driving automatically in rain.

I. INTRODUCTION

Inclement conditions are challenging for autonomous vehicles for several reasons. Rain can obscure and confuse sensors, hide markings on the road, and make a car perform differently. Beyond this, bad weather represents a difficult test for artificial intelligence algorithms. As the key component, the perception subsystem is the eyes of the autonomous driving system, responsible for detecting information about surrounding obstacles and markings on the road. A number of applications, including lane line detection, vehicle identification, and environment perception, require precise feature learning from street images employing vision-based perception functions like target detection and semantic segmentation. At the same time, the accuracy and robustness of the perception system is a critical factor in the safety of autonomous driving decisions. However, on rainy days, adverse weather conditions can significantly degrade image quality and obscure background objects, thus affecting the overall performance of autonomous driving perception.

The associate editor coordinating the review of this manuscript and approving it for publication was Mingbo Zhao.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

Rain removal tasks for images can be used to improve autonomous driving perception in specific adverse weather conditions.

The rain removal work can simply be seen as a restoration process for the rainy image. At this point, the degradation process of rain can be modeled in the form of a linear expression about the degradation matrix A and the additive noise ε, as shown in the following equation:

$$y = Aw + \varepsilon. \tag{1}$$

In this case, the rain removal problem can be paraphrased as getting a clean image w from the rainy image y. The objective is to restore image areas affected by rain streaks and large rainwater accumulations to a clean state. For single-image rain removal methods, they are broadly classified into model-based optimization methods [1], [2], [3], [4], [5], [6] and deep learning-based methods [7], [8], [9], [10], [11]. Now, traditional model-based optimization methods are gradually being replaced by deep learning-based methods because deep learning has a more powerful learning ability and a better image mapping ability. However, most deep neural networks (DNNs) use a black-box design and lack interpretability, which also leads to the optimization of the network often relying on the stacking of different modules, making the network more complex and the operation slower. Relatively, deep unfolding networks (DUN) [17], [18], [19], [20], [21] can be seen as a bridge between traditional models and deep learning, which proposes interpretable end-to-end parameter optimization, providing better detail recovery and faster inference. However, most of the designs derive solutions from known degradation processes, making them not universal for complex real-world situations due to unclear correlation signals and spatial distribution. Moreover, the DUN network treats images as input and output between adjacent stages. On one hand, it is unable to extract precise features for each stage, and on the other hand, it is difficult to avoid information loss in the information transfer between stages, which leads to serious distortion of the final image information obtained.

Considering the above limitations, we design an end-to-end network by combining model-based optimization methods with deep learning through the deep unfolding of the iterative process of the PGD algorithm to achieve fast and accurate rain removal. Unlike general DUNs, for each stage, we integrate the gradient estimation strategy into the PGD algorithm to predict the gradient in unknown situations. Considering complex realistic scenarios, we designed intra-stage information integration to further improve the ability of each stage to extract image features, and inter-stage information pathways to combine the information condensed in the previous stage with the information in the current stage to address the deficiencies of information loss inherent in DUN. In this way, the efficiency of removing rain is improved and the performance of autonomous driving perception is enhanced.

Overall, the main contributions of this paper are summarized as:

- We use the PGD algorithm as the underlying optimization model for rain removal and unfold it in depth for iterative rain removal. Compared with the existing work, our model converges faster, and then can handle more complex real-world scenarios, and is more suitable for high-level perception tasks combined with autonomous driving.
- We propose an internal feature fusion module, which introduces dilated convolutions to extract local features at different scales and fuses global features with local features at each stage to improve the efficiency of feature extraction by the model.

The rest of this paper is structured as follows. Section II introduces the related work and preliminaries for overviewing the baseline approach. Section III details the proposed rain removal method. Section IV evaluates the performance of the proposed rain removal method through comparative experiments and presents the experimental results. Finally, section V draws the conclusion and concludes the paper.

II. RELATIVE WORK AND PRELIMINARIES

A. TRADITIONAL MODEL-BASED RAIN REMOVAL METHODS

The traditional model-based rain removal methods usually transform into maximizing a posteriori estimation task, that is:

$$\hat{w} = \arg\max_{w} \log P(w|y) + \log P(w) \tag{2}$$

where $\log P(w|y)$ and $\log P(w)$ represent the fidelity term and regularization term of the data, respectively. The fidelity is generally replaced by ℓ_2-norm, i.e. the objective function of the rain removal task for optimization can be written in the following recognizable form:

$$\hat{w} = \arg\min_{w} \left(\frac{1}{2} \|y - Aw\|_2^2 + \lambda h(w) \right) \tag{3}$$

where λ as a hyperparameter, determines the weight of the regularization function $h(w)$. The data fidelity term determines where the rain removal of the task goes as expected, while the regularization term limits the complexity of the model, allowing the model to balance complexity and performance.

Obviously, the traditional model-based approaches rely more on the study of the optical properties of the rain pattern, and then the a priori model of the rain streaks is established by estimating the rainy images, which is then optimized by designing an appropriate regularizer [1], [2], [3], [4], [5], [6]. Specifically, an adaptive method had been proposed in the literature [1] for single-image enhancement, it designs an image classifier to determine whether an image is degraded or not and processes the image according to the chromaticity component values. The authors in the paper [2] use rainy images for dictionary learning and optimization of encoding, by continuously optimizing the dictionary and encoding, an image is finally divided into a sum of two encodings of

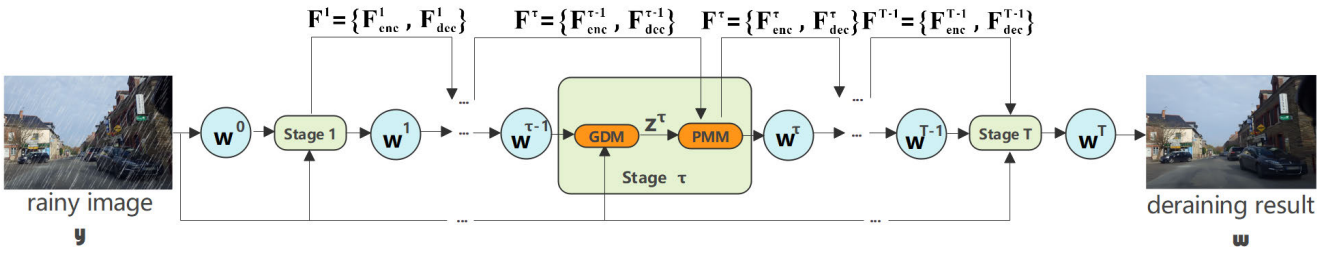


FIGURE 1. The overall framework of the deep unfolding network combined with the PGD algorithm, consisting of several stages consisting of GDM and PMM, each corresponding to one iteration of the PGD algorithm, culminating in the end-to-end single image de-rain task.

a dictionary which correspond to the encoding of rain and the encoding of the background. The method in the paper [3] is based on the idea of block learning using the Gaussian Mixture model (GMM) to portray the prior knowledge of background and rain layers separately, and then the effectiveness of the method in the actual de-rain situation is verified by a combined defogging method. Although this method obtains the best results in the model so far, its estimated background image is partially missing detailed information leading to unclear images. Besides, the method in the paper [4] proposed building a model for rain removal by inscribing the rain line through a low-rank model while using the TV model as a constraint on the background image. The method in the paper [5] uses kernel regression to detect rain lines and then removes the rain lines in the detected rain line region using non-local mean filtering. However, these methods are highly dependent on the a priori knowledge of rain streaks and are not universally applicable to complex real-world environments.

B. DEEP LEARNING-BASED RAIN REMOVAL METHODS

Deep learning network DNN has a more powerful learning ability and an image mapping ability compared with model-based rain removal methods. It learns powerful a priori knowledge from large-scale datasets and can use different DNNs to extract layer features and rain streak information of the rainy image to get a mapping from rainy images to clear images. By now, many useful deep-learning rain removal networks have been proposed [7], [8], [9], [10], [11], [30], [31], [27], [35]. Specifically, LRCnet [35] uses a new encoder-decoder-based network in which a novel low-rank convolution(LR-Conv) for image representation and a residual dense connection (RDC) for feature fusion between encoding and decoding are proposed, yielding excellent image denoising results. The deep detail network [7] is applied to the rain removal problem by simplifying the learning form of the deep residual network. The progressive rain removal network [8] uses recursive ideas for the input and output of the network, taking the stage-by-stage results and the original images with rain as input into each ResNet, and then finally outputting the predicted residual images. SPANet [30] utilizes an attention unit-based network model for removing rain in a local to global manner, etc. Currently, most deep learning methods use a fully supervised training model, but there is no shortage of excellent unsupervised and semi-supervised training networks [12], [13], [14], [15], [16]. Specifically, RR-GAN [12] employs an unsupervised training mode to obtain rain streak information, by using a recursive memory module that exploits attentional mechanisms using the multi-scale attentional memory generator MAMG circular recursive. DerainCycleGAN [13] adopts a two-branch network for unsupervised rain removal. Semi-DerainGAN [14] uses a semi-supervised rain streak information learner based on shared parameters and achieve a strong generalization power to the real SID task. However, regardless of the training mode used, each of these approaches has its own advantages and disadvantages, and all of them are assembled using network modules from existing deep learning tools to learn the background layer directly in an end-to-end format, resulting in ignoring the inherent a priori stripe structure of rainy images and making it lack significant interpretability in the network architecture.

C. DEEP UNFOLDING NETWORK

Deep unfolding networks exploit the feature that a series of recursive DNN modules can equivalently replace traditional iterative optimization algorithms as a bridge that can connect traditional models to deep learning. There have been many studies that have used deep unfolding networks to achieve high performance. For example, Zhang et al. [17] unfolded MAP inference via a semi-quadratic splitting algorithm to obtain a fixed number of iterations consisting of alternating solved data subproblems and a priori subproblems, and solved them with neural modules to obtain a trainable end-to-end iterative network. Nah et al. [18] constructed an implicit gradient flow by cascading and used it for image restoration, Xiong et al. [19], on the other hand, obtained a new method for solving problems such as face feature point detection by unrolling each gradient descent iteration of a non-linear regression model using linear regression. Although deep unfolding networks connect models with deep learning and provide new feasibility in several areas, most unfolding networks that are constantly iterating also suffer from the limited ability to extract features and loss of information during iteration. These issues need to be addressed.

D. PROXIMAL GRADIENT DESCENT ALGORITHM

Consider the minimization of functions of the form

$$\arg\min_{w\in\mathcal{W}} \; g(w) + h(w) \tag{4}$$

where $g(w)$ is convex and differentiable and $h(w)$ is convex but non-smooth. In such a circumstance, the non-smooth term $h(w)$ may cause gradient descent to fail and converge to an incorrect minima. Generally, subgradient methods are used to minimize the non-differentiable term. The drawback of using subgradient methods is that they converge far more slowly than gradient descent. In order to find a fast convergent gradient descent like algorithms, the proximal gradient descent algorithm is invented to solve this type of issue. Since it is inconvenient to compute the gradient, the proximal operator is designed to use as the proximal gradient, and then the gradient descent work can be performed well. Taking the non-differentiable function $h(w) = \lambda \|w\|_1$ as an example, for any vector w, the proximal operator can be expressed as:

$$\begin{aligned}\text{prox}_{\mu,h(\cdot)}(z) &= \arg\min_{w}\left(\frac{1}{2}\|w-z\|_2^2 + \lambda\mu\|w\|_1\right)\\ &= S_{\mu,\lambda}(z)\end{aligned} \tag{5}$$

where $\text{prox}_{\mu,h(\cdot)}(z)$ denotes the proximal operator on the variable w and the function $h(\cdot)$, and μ denotes the step size of the proximal gradient descent. $S_{\mu,\lambda}(z)$ denotes the soft threshold function for the variable w, where λ as a hyper-parameter, determines the weight of the function $h(w)$. The formula indicates that for any given $w \in \mathbb{R}^n$, the solution $w^* = \text{prox}_{\mu,h(\cdot)}(z)$ that minimizes $\frac{1}{2}\|w-z\|_2^2 + \lambda\mu\|w\|_1$ can be found.

For the optimization problem $\min_{w\in\mathcal{W}} g(w) + h(w)$ in (4), the iterative update is composed of the following two steps:

1) Gradient step: starting at $w^{\tau-1}$, take a step in the direction of the gradient of the differentiable part $g(w)$, i.e., $z^\tau = w^{\tau-1} - \mu\nabla g(w^{\tau-1})$, where $\nabla(\cdot)$ represents the differential operator;
2) Evaluate prox operator: starting at z^τ, evaluate the proximal operator of the non-smooth part $h(w)$, i.e., $w^\tau = \text{prox}_{\mu,h(\cdot)}(z^\tau)$.

Hence, the whole iterative update of proximal gradient descent methods for the variable w can be expressed as:

$$\begin{aligned}w^\tau &= \text{prox}_{\mu,h(\cdot)}(w^{\tau-1} - \mu\nabla g(w^{\tau-1}))\\ &= S_{\mu,\lambda}(w^{\tau-1} - \mu\nabla g(w^{\tau-1}))\end{aligned} \tag{6}$$

where the superscript τ of the variable indicates the current number of iterations. For completeness and readability, we give the pseudocode of the PGD algorithm as shown below:

III. PROPOSED APPROACH

The single image deraining problem has intrigued scientists and engineers in the artificial intelligence field for the last five years because of their ability to significantly improve the performance of vision tasks in rainy environments. However, most existing methods still have two major drawbacks. First, the rain streaks in a single rainy image are seriously coupled with the background information, which leads to the failure of many methods to correctly identify the rain streaks, and further leads to the loss of texture details in the rain removal results. Second, the calculation cost is expensive, and not conducive to practical application. To overcome these issues, we propose a PGD-based deep unfolding network for the rain removal problem, in which our model takes rainy images as input and obtains clean images directly.

Algorithm 1 PGD

Input: maxiterations or a small number ϵ
Output: $\hat{w}$
Initialize w, λ, μ, set $\tau = 1$
while $J^\tau - J^{\tau-1} < \epsilon$ or $t <$ maxiterations **do**
  update w^τ using $w^\tau = S_{\mu,\lambda}(w^{\tau-1} - \mu\nabla g(w^{\tau-1}))$
  compute $J^\tau(w) = g(w^{\tau-1} + \lambda h(w^{\tau-1}))$
  $\tau \leftarrow \tau + 1$
end while

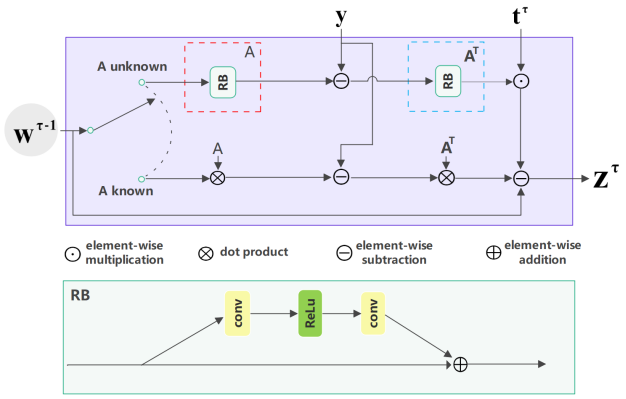


FIGURE 2. The overall framework of the deep unfolding network combined with the PGD algorithm, consisting of several stages consisting of GDM and PMM, each corresponding to one iteration of the PGD algorithm, culminating in the end-to-end single image de-rain task.

A. PGD ITERATIVE RAIN REMOVAL PROCESS

Recalling the optimization model of rain removal tasks in (3), $g(w) = \frac{1}{2}\|y-Aw\|_2^2$ denotes the data fidelity term. Clearly, (3) can be written in the form of (4). By the PGD algorithm composed of the gradient descent step and the proximal evaluation step, the rain removal task can be approximately solved by an iterative process:

$$\begin{aligned}w^\tau &= \arg\min_{w}\left(\frac{1}{2}\left\|w-(w^{\tau-1}-\mu\nabla g(w^{\tau-1}))\right\|_2^2 + \lambda h(w)\right)\\ &= \text{prox}_{\lambda,h(\cdot)}(w^{\tau-1}-\mu\nabla g(w^{\tau-1})).\end{aligned} \tag{7}$$

Thus, substituting the gradient $\nabla g(w^{\tau-1}) = A^T(Aw^{\tau-1} - y)$ into the above equation yields:

$$w^\tau = \text{prox}_{\lambda,h(\cdot)}(w^{\tau-1}-\mu A^T(Aw^{\tau-1} - y)). \tag{8}$$

In this paper, we utilize a PGD-based deep unfolding network to address rain removal tasks, in which we set up multiple iteration stages to calculate the process of PGD update. Each stage consists of a gradient descent module (GDM) and a proximal mapping module (PMM). Figure 1 illustrates the framework of our PGD-based deep unfolding network.

B. GRADIENT DESCENT MODULE

Generally, the degradation matrix A in most image restoration tasks is known. In this case, we utilize directly the gradient descent step to calculate the PGD update in (8). To improve the robustness of the model, we set a trainable parameter μ^τ instead of μ in the gradient descent module,

$$z^\tau = w^{\tau-1} - \mu^\tau A^T(Aw^{\tau-1} - y). \tag{9}$$

However, there exist few image restoration tasks that the degradation matrix A is unknown. In this case, we intend to use the data-driven way to approximate the gradient by the following steps:

1) Matrix approximation step: set two independent residual blocks R_A and R_{A^T} without batch normalization to approximate the degradation matrix A and its transpose A^T;
2) Gradient prediction step: perform gradient prediction by replacing the degradation matrix A and its transpose A^T with R_A^τ and $R_{A^T}^\tau$, respectively, at each iteration:

$$z^\tau = w^{\tau-1} - \mu^\tau R_A^\tau(R_{A^T}^\tau w^{\tau-1} - y). \tag{10}$$

Figure 2 shows the gradient descent module associated with two cases that the degradation matrix A is known or unknown.

C. PROXIMAL MAPPING MODULE (PMM)

During each PGD iteration, the image degradation requires feature learning of the current degraded image. So when we compute the gradient descent result z^τ, we need to push it into the proximal mapping module for image feature learning. In order to facilitate us to obtain an end-to-end trainable network, we choose the encoder-decoder network as the architecture of the proximal mapping module.

At each stage of the encoder network, we first use the channel attention module (CAB) [22] to let the information undergo feature learning in the channel dimension, which helps to form a sense of importance for each channel. After that, we use the result f_{global}^τ as a global feature of the corresponding stage. For traditional deep unfolding networks, the information in each layer of the stage cannot avoid problems such as poor feature extraction and information loss during the iteration process. For this reason, we propose an internal feature fusion (IFF) module to improve the feature extraction effect at each stage and an inter-stage feature fusion (ISFF) module to reduce the information loss of features during continuous iterations, respectively. And introduced them into our encoder-decoder network architecture.

1) INTERNAL FEATURE FUSION (IFF) MODULE

In each stage of the encoder network, it is difficult to avoid the loss of local information in the process of image downsampling, so we propose the internal feature fusion module. It can help us reduce local information distortion as much as possible and improve our feature extraction efficiency.

Inspired by [23], for the extraction of encoder features $\hat{F}_{enc}^\tau$ in different scales at the τ-th stage, we first propose a special residual block (DRB) with dilation convolution [24]. Considering that the batch normalization layer can lead to over-smoothing of some specific features, which can degrade the model performance. In addition, it may consume more memory and slow down our computation, so our residual block removes the batch normalization. At the same time, in order to obtain a larger perceptual field and to better extract local features, we introduced a dilated convolution into our residual block, as shown in Figure 3. Our dilated residual block consists of three 3×3 convolutions and two ReLu functions, where the third convolution is a dilated convolution with the dilation rate equal to 2. We pass the downsampled feature data $f_{enc\&n}^\tau$ into our dilated convolutional residual block for further extraction of local features. At this point, we obtain a local encoder feature $f_{1\&n}^\tau$ for the n-th scale of the τ-th stage. After that, we fuse the local feature $f_{1\&n}^\tau$ with the global feature f_{global}^τ of that stage. The formula for the fusion module within the whole stage is shown in the following:

$$\begin{cases} f_{1\&n}^\tau = DRB(f_{enc\&n}^\tau) \\ f_{2\&n}^\tau = f_c\left(f_{1\&n}^\tau, \mathrm{conv}(f_{global}^\tau)\right) \\ \hat{F}_{enc\&n}^\tau = \mathrm{conv}(f_{global}^\tau) \times \mathrm{conv}(f_{2\&n}^\tau) + f_{1\&n}^\tau \\ \quad \times \mathrm{conv}(f_{2\&n}^\tau) \end{cases} \tag{11}$$

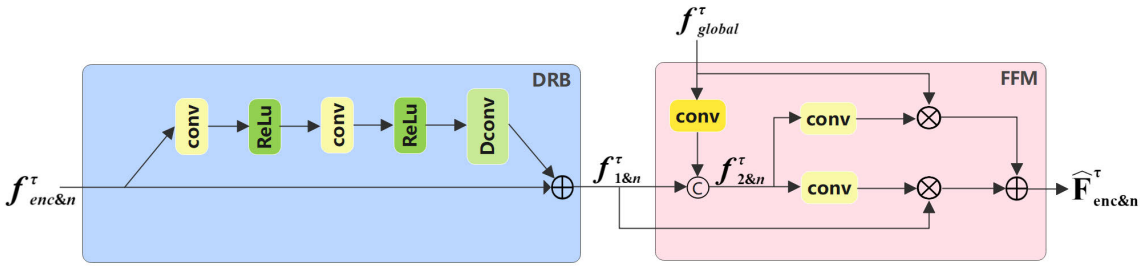


FIGURE 3. Inter-stage information fusion module (IFF) demonstration at stage τ.

where $DRB(\cdot)$ indicates that the variable is running our dilated residual block and $f_c(\cdot)$ represents the concatenation operation.

The global feature f^{τ}_{global} is convolved to match the image size after down-sampling at different scales and then concatenated on the channels to obtain a concatenated feature map with twice the number of channels. We use 1×1 convolution to automatically learn the weights of the two feature maps, and obtain the weights of the two identical channel weights, which correspond to the global and local feature maps respectively. Finally, by multiplying the two weights and the feature maps and adding them together, we obtain the internally fused encoded features $\hat{F}^{\tau}_{enc\&n}$.

2) INTER-STAGE FEATURE FUSION (ISFF) MODULE

For deep unfolding networks, the process of image deraining is not just a stage of feature learning. Images are transmitted through continuous stages, between which the loss of information is unavoidable. So, we follow [25] and use the inter-stage feature fusion (ISFF) module on different sizes at the same stage to establish information pathways between the stages to solve this problem. $F^{\tau}_{enc\&n}$ and $F^{\tau}_{dec\&n}$ represent the encoded features and decoded features we extracted at stage τ, respectively, as shown in Figure 4.

For the computation of the τ-th stage, we fuse the encoded $F^{\tau-1}_{enc}$ and decoded $F^{\tau-1}_{dec\&n}$ features from the previous stage into the encoding process of this stage by using spatially adaptive normalization. For encoded and decoded features at different scales for stage $\tau - 1$, each of them is added element-by-element after 1×1 convolution to focus on the combination of features across channels, and then α^{τ} and β^{τ} are calculated by two convolutions, respectively, to transform the original encoded feature $F^{\tau-1}_{enc\&n}$ into the fused encoded feature $F^{\tau}_{enc\&n}$, i.e., the fusion process is shown in the following equations:

$$\begin{cases} \alpha^{\tau} = \mathrm{conv}_{\alpha}[\mathrm{conv}(F^{\tau-1}_{enc\&n}) + \mathrm{conv}(F^{\tau-1}_{dec\&n})] \\ \beta^{\tau} = \mathrm{conv}_{\beta}[\mathrm{conv}(F^{\tau-1}_{enc\&n}) + \mathrm{conv}(F^{\tau-1}_{dec\&n})] \\ F^{\tau}_{enc\&n} = \hat{F}^{\tau}_{enc\&n} \odot \alpha^{\tau} + \beta^{\tau} \end{cases} \tag{12}$$

where α and β are not vectors but tensors with spatial dimensions that allow the encoded features of each stage to be fused with the condensed memory of the previous stage, thus producing an information-rich proximal mapping.

Referring to the U-Net network [26] for global path presentation from input to output, we use 2×2 maximum pooling with stride 2 for down-sampling to facilitate us to switch scales. After each down-sampling, an IFF module is connected to improve the efficiency of information extraction. Then we get an initial encoded feature $\hat{F}^{\tau}_{enc\&n}$ at the current scale. As for the operation result of the CAB module at the initial scale, since the scale has not changed, we regard the global feature of this stage f^{τ}_{global} as the encoded result $\hat{F}^{\tau}_{enc\&n}$ at the current scale. And to avoid information loss from inter-stage, we collect information on the encoded and decoded features of different scales and fuse them into the corresponding scales of the next stage, forming an information pathway for each scale between stages. In the end, we extract the w^{τ} and the corresponding images of each stage output with the help of the supervised attention module [27] and pass them to the next stage by subspace mapping [28]. The following Figure 5 shows our idea.

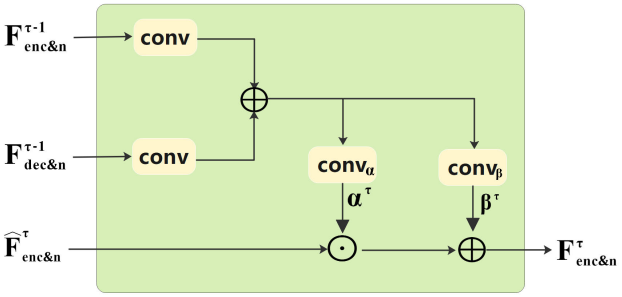


FIGURE 4. Inter-stage feature fusion module (ISFF) demonstration at stage τ.

D. LOSS FUNCTION

For the loss function, instead of using the ℓ_2-norm loss, we propose to train the network with the robust Charbonnier loss function L_{char} [29] to better handle outliers and improve the performance, which is defined as:

$$L_{char} = \sqrt{\|w - w^{\tau}\|^2 + \epsilon^2} \tag{13}$$

where w represents the ground-truth image and ϵ denotes the stop error. Besides, in order to capture global and local differences, we introduce an edge loss function L_{edge} as:

$$L_{edge} = \sqrt{\|\triangle w - \triangle w^{\tau}\|^2 + \epsilon^2} \tag{14}$$

where $\triangle$ represents the Laplace operator.

For the output w^{τ} of each stage, we optimize our end-to-end deep unfolding network using the following loss function:

$$L = \sum_{\tau=1}^{T} [L_{char}(w, w^{\tau}) + \eta L_{edge}(w, w^{\tau})] \tag{15}$$

where T represents the total iteration number, and the parameter η indicates the importance of both losses. In this paper, we set $\epsilon = 10^{-3}$ and $\eta = 0.05$.

E. HIGH-LEVEL TASK EVALUATIONS

The perception performance under rain conditions is crucial for the robustness of the autonomous driving perception system and the safety of autonomous driving. Nowadays, many autonomous driving rain removal tasks are image-based rain removal performance studies, which are seldom combined with the high-level perception tasks of autonomous driving for validation. The decision-making system for autonomous driving is mostly based on high-level perception tasks, such

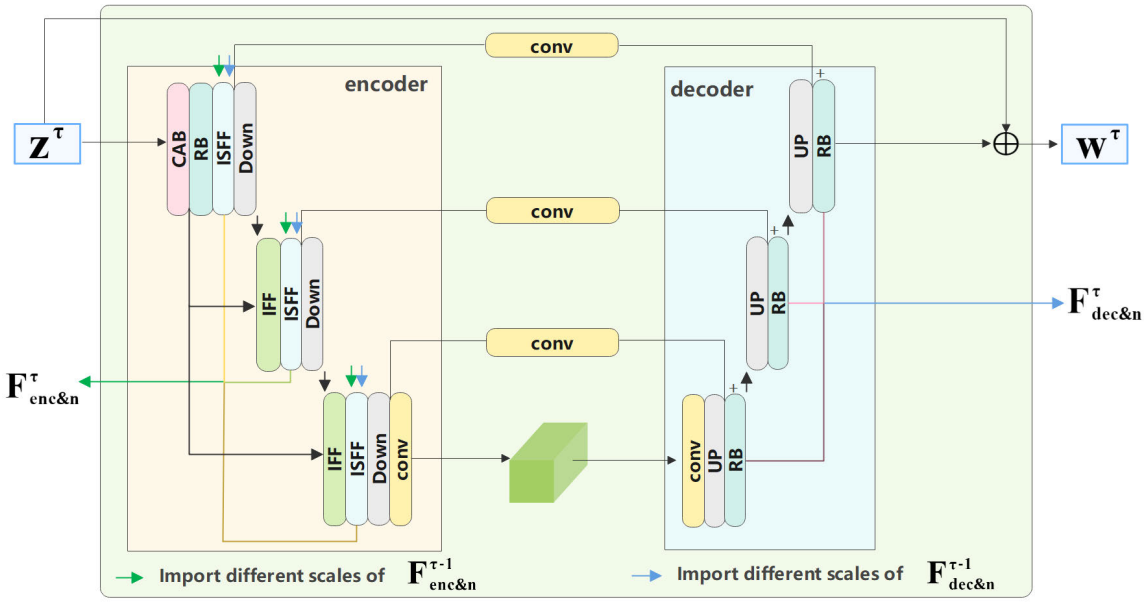


FIGURE 5. Proximal mapping module (PMM) framework demonstration. To distinguish the encoded and decoded features at different scales of the stages, different colors are used. For the τ-th stage, the gradient descent result z^{τ} is passed in, and the τ-th order iteration result w^{τ} is obtained after the encoder-decoder network.

TABLE 1. Comparison of PSNR and SSIM under light rain patterns.

Metrics	Input	RESCAN [30]	SPANet [31]	MPRNet [27]	Ours
PSNR	30.07	32.12	32.97	33.06	**36.12**
SSIM	0.8926	0.9078	0.9169	0.9213	**0.9463**

TABLE 2. Comparison of PSNR and SSIM under moderate rain patterns.

Metrics	Input	RESCAN [30]	SPANet [31]	MPRNet [27]	Ours
PSNR	32.42	34.05	34.22	36.23	**36.88**
SSIM	0.9184	0.9343	0.9279	0.9622	**0.9689**

TABLE 3. Comparison of PSNR and SSIM under heavy rain patterns.

Metrics	Input	RESCAN [30]	SPANet [31]	MPRNet [27]	Ours
PSNR	28.29	30.81	32.03	**32.24**	32.15
SSIM	0.8764	0.8929	0.9100	**0.9217**	0.9168

as target detection and semantic segmentation, for decision setting. Therefore, we further explore the effect of rain on the autonomous driving perception system by combining the rain removal task with the high-level perception task of autonomous driving. We provide the rain removal images to the existing target detection task Yolov5 and the semantic segmentation task PIDNet [32], experimental results and discussion will be provided in Section IV.

IV. EXPERIMENTS AND EVALUATION

We first describe the datasets used in this paper and the training parameters setting in Section IV-A. In Section IV-B, the quantitative and qualitative benchmarking of our method is performed and compared with existing deraining models [30], [31], [27]. In addition, we use real rainfall pictures as a qualitative determination evaluation. In section IV-C, we combine our rain removal network with the target detection task via Yolov5 and the semantic segmentation task via PIDNet [32], then show the comparative results.

A. TRAINING DETAILS

In order to simulate rainy weather scenarios for autonomous driving, we used a collection of large-scale camera images of different city street scenes as a dataset and overlaid the images with simulated raindrop movement trajectories by OpenCV's random noise method, after which the images were classified into three categories according to rainy patterns, i.e., light and moderate as well as heavy rain patterns with a resolution size of 480 $\times$ 320. The synthetic dataset consisted of 22, 500 images for training, 5, 000 images for validation, and 10, 300 images for testing. Here we only use the light rain pattern in the synthetic dataset for parametric training, so only 7, 500 images are picked performing on the NVIDIA RTX 3090 GPU for training. In addition, we set the initial learning rate of the Adam optimizer to 1×10^{-4}, the batch size to 4 for 50 epochs, and the number of stages T to 7. The default parameters associated with each stage remain unchanged except for the first and last stages.

B. EVALUATION OF RAIN REMOVAL EFFECT

We compare our method with three existing rain removal methods: RESCAN [30], SPANet [31], and MPRNet [27]. We show the results of each network in Figure 6, our rain

FIGURE 6. Comparison of rain removal results. The three rows from top to bottom represent light rain patterns, medium rain patterns, and heavy rain patterns. Each column from left to right represents the rain image, RESCAN deraining image, SPANet deraining image, MPRNet deraining image, our network deraining image, and the original image, respectively.

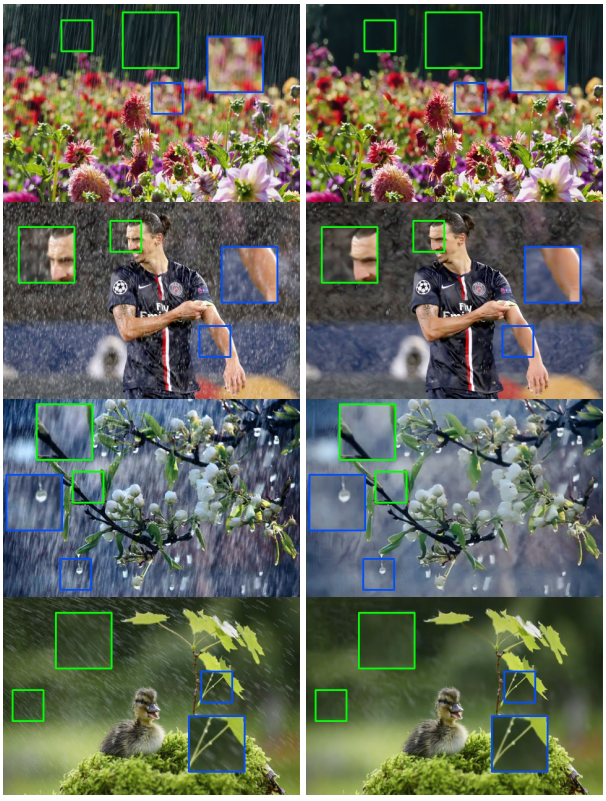


FIGURE 7. Comparison of our rain removal effect for real rain images. The left side of the image set is the rainy images, and the right side is the deraining images.

removal effect is more visually noticeable, removing more rain streaks without destroying the image texture as much as possible.

For the quantitative evaluation of rain removal performance, we use standard metrics: the structural similarity index (SSIM) [33] and the peak signal-to-noise ratio (PSNR) [34] for comparison. Generally, the better the rain removal, the larger the SSIM or PSNR values. We divided the synthetic dataset into three test sets according to the rain pattern size and passed them into the rain removal networks RESCAN [30], SPANet [31], MPRNet [27] and our network in turn, and the average SSIM and PSNR obtained are recorded in Table 1, Table 2 and Table 3, where the results that achieve the highest rain removal performance we use the emphasis markers. It can be found that our model outperforms the three networks we listed in terms of rain removal under light and medium rain patterns, and is comparable to the best network in terms of rain removal under heavy rain patterns.

To show more clearly the effect of our IFF module, we removed the IFF module and performed the same operation on the SSIM and PSNR in the light rain mode, and found that the SSIM was reduced by 0.0062 and the PSNR was reduced by 0.09 compared to the original model, thus showing that the addition of the internal feature fusion module had an enhanced effect on the extraction of features.

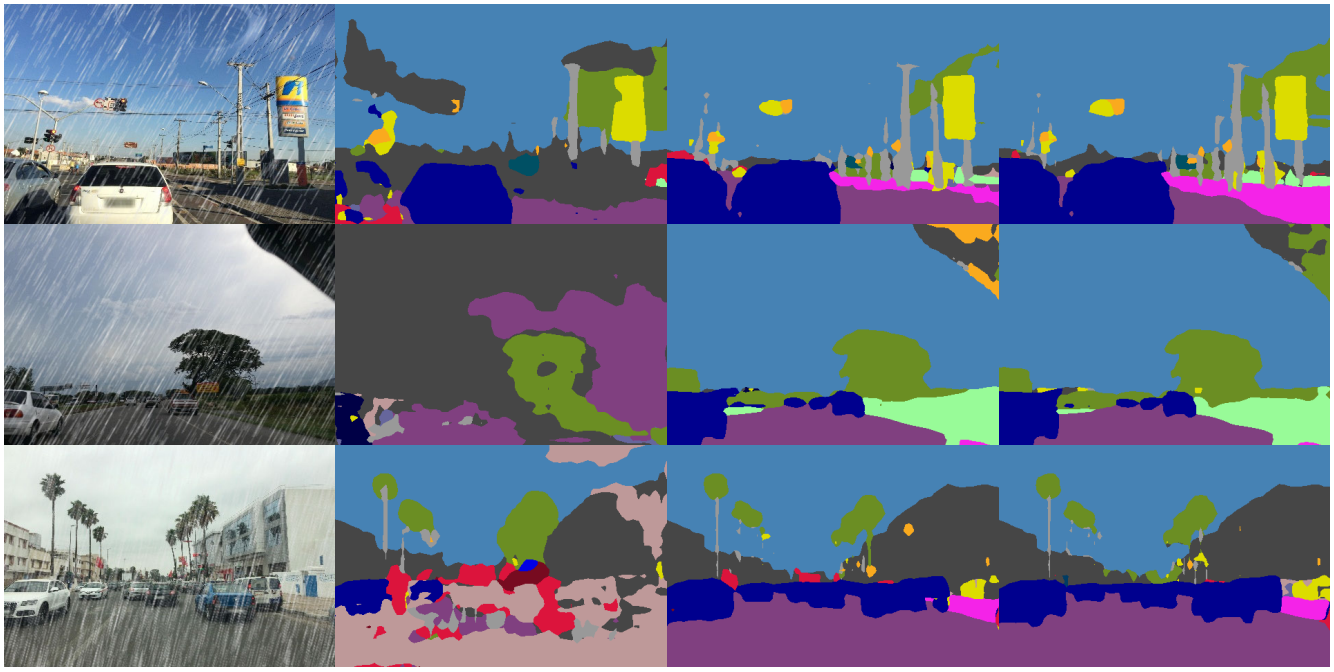
In addition, we also perform the rain removal test by realistic complex rainy images, as shown in Figure 7. We can find that our network can also play an excellent effect when dealing with complex real environments, and has good generalization ability for complex rain types.

C. EVALUATION IN CONJUNCTION WITH HIGH-LEVEL PERCEPTION TASK

For autonomous driving sensing systems, training with a large number of images in rainy conditions is not a sensible way due to the blurring effect of images, and the cost of direct training is expensive. However, if the high-level perception task of autonomous driving is to be tested in combination with the rain removal algorithm, which requires the rain removal algorithm to be able to remove rain quickly so that the autonomous driving can execute commands more quickly. Our rain removal network uses a PGD algorithm,

(a) Generated rain images (b) TD for rain images by Yolov5 (c) TD for rain images by our model (d) TD for original images

FIGURE 8. Comparison results of the rain removal task combined with the Target Detection (TD) task. This task is tested in three different rain patterns as three rows in the figure matrix from top to bottom, i.e., the light rain pattern, the medium rain pattern, and the heavy rain pattern, respectively. The first column is the generated rainy images, the second column is the target detection results for rain images by Yolov5, the third row is the target detection results for rain images after applying our rain removal model, and the fourth column is the target detection results for original images by Yolov5.



(a) Generated rain images (b) SS for rain images by PIDNet (c) SS for rain images by our model (d) SS for original images

FIGURE 9. Comparison results of the rain removal task combined with the Semantic Segmentation (SS) task. This task is tested in three different rain patterns as three rows in the figure matrix from top to bottom, i.e., the light rain pattern, the medium rain pattern, and the heavy rain pattern, respectively. The first column is the generated rain images, the second column is the semantic segmentation results for rainy images by PIDNet only, the third row is the semantic segmentation results for rainy images after applying our rain removal model, and the fourth column is the semantic segmentation results for original images by PIDNet.

which allows for faster convergence and is more suitable for high-level perception tasks in conjunction with autonomous driving. In addition to this, to effectively demonstrate the impact of our rain removal model on the perceptual performance of autonomous driving, we perform an experimental comparison combining target detection and semantic segmentation tasks.

1) EVALUATION IN CONJUNCTION WITH TARGET DETECTION

To evaluate the expansion ability of our rain removal model proposed in this paper, we first test the target detection task by employing Yolov5 as a baseline model, and the test results are shown in Figure 8. The experimental results show that on one hand, our approach achieves a significant perceptual performance improvement in combination with the target detection task, providing a significant visual improvement in the image that more closely resembles real environmental features. On the other hand, our model converges more quickly due to the PGD algorithm, which can be quickly combined with the trained high-level perceptual model, saving a significant amount of training time for the perceptual model.

2) EVALUATION IN CONJUNCTION WITH SEMANTIC SEGMENTATION

We also applied our approach to another high-level perception task, i.e., the semantic segmentation task. The state-of-the-art PIDNet [32] is used for this task as the baseline model, thus making the test more realistic and reasonable. Figure 9 shows a comparison of the semantic segmentation before and after removing the rain, and it is clear that the rain removal effect of our model brings a significant improvement to the semantic segmentation task and is close to the semantic segmentation results for the original images without rain.

V. CONCLUSION

In this paper, we performed a single image rain removal task using a PGD algorithm combined with DUN. Our approach integrates the advantages of both model-based and deep learning-based methods by incorporating a gradient estimation strategy with a proximal mapping module at each stage of the iteration to degrade complex rain streaks and compensate for the lack of information extraction within each stage and the loss of information between stages. Therefore, our method can be more generalized and accurate when dealing with the degradation of complex rainy images. Also, the extremely fast processing speed can effectively combine the rain removal task with the high-level perception task of autonomous driving and improve the robustness of the autonomous driving perception system.

REFERENCES

[1] Z.-Y. Liu, H.-R. Jiang, and H.-W. Xu, "Low-quality image enhancement algorithms in extreme weather conditions," *Comput. Eng. Appl.*, vol. 53, no. 8, pp. 193–198, 2017.

[2] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Santiago, Chile, Dec. 2015, pp. 3397–3405.

[3] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2736–2744.

[4] Y. Chen and C. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, NSW, Australia, Dec. 2013, pp. 1968–1975.

[5] J. Kim, C. Lee, J. Sim, and C. Kim, "Single-image deraining using an adaptive nonlocal means filter," in *Proc. IEEE Int. Conf. Image Process.*, Melbourne, VIC, Australia, Sep. 2013, pp. 914–917.

[6] L. Zhu, C. Fu, D. Lischinski, and P. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 2545–2553.

[7] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 1715–1723.

[8] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: A better and simpler baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 3932–3941.

[9] R. Li, L. Cheong, and R. T. Tan, "Heavy rain image restoration: Integrating physics model and conditional adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 1633–1642.

[10] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Trans. Image Process.*, vol. 26, no. 6, pp. 2944–2956, Jun. 2017.

[11] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 695–704.

[12] K. Javed, G. Hussain, and T. Seong, "RRGAN: Removing rain using generative neural network," in *Proc. Int. Conf. Green Human Inf. Technol.*, Jan. 2019, pp. 287–289.

[13] Y. Wei, Z. Zhang, Y. Wang, M. Xu, Y. Yang, S. Yan, and M. Wang, "DerainCycleGAN: Rain attentive CycleGAN for single image deraining and rainmaking," *IEEE Trans. Image Process.*, vol. 30, pp. 4788–4801, 2021.

[14] Y. Wei, Z. Zhang, Y. Wang, H. Zhang, M. Zhao, M. Xu, and M. Wang, "Semi-deraingan: A new semi-supervised single image deraining," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Shenzhen, China, Jul. 2021, pp. 1–6.

[15] A. Akram and N. Khan, "U.S.-GAN: On the importance of ultimate skip connection for facial expression synthesis," 2021, *arXiv:2112.13002*.

[16] H. Huang, A. Yu, and R. He, "Memory oriented transfer learning for semi-supervised image deraining," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 7728–7737.

[17] K. Zhang, L. Van Gool, and R. Timofte, "Deep unfolding network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 3214–3223.

[18] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 257–265.

[19] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 532–539.

[20] Q. Ning, W. Dong, G. Shi, L. Li, and X. Li, "Accurate and lightweight image super-resolution with model-guided deep unfolding network," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 240–252, Feb. 2021.

[21] W. Dong, X. Li, L. Zhang, and G. Shi, "Sparsity-based image denoising via dictionary learning and structural clustering," in *Proc. CVPR*, Colorado Springs, CO, USA, Jun. 2011, pp. 457–464.

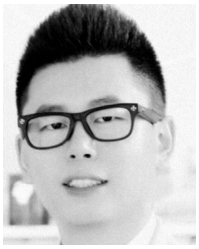
[22] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 3–19.

[23] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1132–1140.

[24] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*.

[25] T. Park, M. Liu, T. Wang, and J. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 2332–2341.
[26] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," 2015, *arXiv:1505.04597*.
[27] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M. Yang, and L. Shao, "Multi-stage progressive image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 14816–14826.
[28] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan, and S. Liu, "NBNet: Noise basis learning for image denoising with subspace projection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 4894–4904.
[29] P. Charbonnier, L. Blanc-Feraud, G. Aubert, and M. Barlaud, "Two deterministic half-quadratic regularization algorithms for computed imaging," in *Proc. 1st Int. Conf. Image Process.*, 1994, pp. 168–172.
[30] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proc. ECCV*, Sep. 2018, pp. 254–269.
[31] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 12262–12271.
[32] J. Xu, Z. Xiong, and S. P. Bhattacharyya, "PIDNet: A real-time semantic segmentation network inspired by PID controllers," 2022, *arXiv:2206.02066*.
[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[34] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, Jun. 2008.
[35] J. Ren, Z. Zhang, R. Hong, M. Xu, H. Zhang, M. Zhao, and M. Wang, "Robust low-rank convolution network for image denoising," in *Proc. 30th ACM Int. Conf. Multimedia*, New York, NY, USA, Oct. 2022, pp. 6211–6219.

CHAO HU is currently pursuing the bachelor's degree with Southwest University, Chongqing, China. His major concern is information security. His research interests include deep learning, image target detection, image restoration, and image processing.



HUIWEI WANG received the B.S. degree in information and computing science and the M.E. degree in computer application from Chongqing Jiaotong University, Chongqing, China, in 2008 and 2011, respectively, and the Ph.D. degree in computer science from Chongqing University, Chongqing, in 2014.

He is currently an Associate Researcher with the College of Electronic and Information Engineering, Southwest University, Chongqing. From 2014 to 2016, he was a Postdoctoral Research Associate with Texas A&M University at Qatar, Doha, Qatar, where he was also a Program Aide, from 2012 to 2013. His current research interests include neural networks, multiagent networks, wireless sensor networks, and automatic driving technology.