Asymmetric Dual-Decoder U-Net for Joint Rain and Haze Removal

YUAN FENG*, College of Science, Zhejiang University of Technology, China
YAOJUN HU*, College of Computer Science and Technology, Zhejiang University, China
PENGFEI FANG†, School of Computer Science and Engineering, Southeast University, China
SHENG LIU, College of Computer Science, Zhejiang University of Technology, China
YANHONG YANG and SHENGYONG CHEN, College of Computer Science and Technology, Tianjin University of Technology, China

This work studies the multi-weather restoration problem. In real-life scenarios, rain and haze, two often co-occurring common weather phenomena, can greatly degrade the clarity and quality of the scene images, leading to a performance drop in the visual applications, such as autonomous driving. However, jointly removing the rain and haze in scene images is ill-posed and challenging, where the existence of haze and rain and the change of atmosphere light, can both degrade the scene information. Current methods focus on the contamination removal part, thus ignoring the restoration of the scene information affected by the change of atmospheric light. We propose a novel deep neural network, named Asymmetric Dual-decoder U-Net (ADU-Net), to address the aforementioned challenge. The ADU-Net produces both the contamination residual and the scene residual to efficiently remove the contamination while preserving the fidelity of the scene information. Extensive experiments show our work outperforms the existing state-of-the-art methods by a considerable margin in both synthetic data and real-world data benchmarks, including RainCityscapes, BID Rain, and SPA-Data. For instance, we improve the state-of-the-art PSNR value by 2.26/4.57 on the RainCityscapes/SPA-Data, respectively. Codes will be made available freely to the research community.

CCS Concepts: • **Computing methodologies** → **Reconstruction**.

Additional Key Words and Phrases: Joint rain and haze removal, Asymmetric Dual-decoder U-Net (ADU-Net), contamination residual, scene residual

1 INTRODUCTION

When photographing in bad weather, the quality of outdoor scene images can be greatly degraded by the contamination, i.e., rain, haze or snow, etc. distributed in the air. Such contamination absorbs or disperses the scene light, thereby reducing the contrast and color fidelity of the scene image. Hence, the existence of contamination significantly affects many real-world vision systems, such as scene recognition, object tracking, semantic segmentation, etc, and all of these vision systems are essential for autonomous driving [7, 13, 60]. In other words, such outdoor vision systems, which works efficiently in ideal weather condition, will suffer a

*Both authors contributed equally to this research.
†The author is corresponding author.

Authors' addresses: Yuan Feng, fy@ieee.org, College of Science, Zhejiang University of Technology, 288 Liuhe Rd, Hangzhou, Zhejiang, China, 310023; Yaojun Hu, yaojunhu@zju.edu.cn, College of Computer Science and Technology, Zhejiang University, 866 Yuhangtang Rd, Hangzhou, Zhejiang, China, 310058; Pengfei Fang, School of Computer Science and Engineering, Southeast University, Nanjing, Jiangsu, China, 210096, fangpengfei@seu.edu.cn; Sheng Liu, College of Computer Science, Zhejiang University of Technology, 288 Liuhe Rd, Hangzhou, Zhejiang, China, 310023, edliu@zjut.edu.cn; Yanhong Yang, yyh_03@163.com; Shengyong Chen, College of Computer Science and Technology, Tianjin University of Technology, Tianjin, China, sy@ieee.org.

1551-6857/2023/10-ART
https://doi.org/10.1145/3628451

plummet due to complex real-world weather conditions. Therefore, it is essential to develop algorithms to restore images contaminated by different contaminants as a pre-processor for such outdoor vision systems.

In this work, we focus on a real yet less-investigated scenario, the co-occurrence of the rain and haze in the scenes. Both image rain removal and haze removal are challenging low-level computer vision tasks. Many efforts have been made to solve the individual rain removal and haze removal tasks [48, 52, 56]. However, only a few works consider removing the rain and haze jointly in scene images [18, 21, 47]. In the real-world scenario, it is a very common situation that the rain and haze co-occur in the rainfall environment (see Fig. 1a) [17]. Along with rain streaks and raindrops, the uneven haze will also obscure the image, interfering with the perception of the environment. Such a scenario brings challenges to the outdoor vision systems that are required to jointly remove the rain and haze in images.

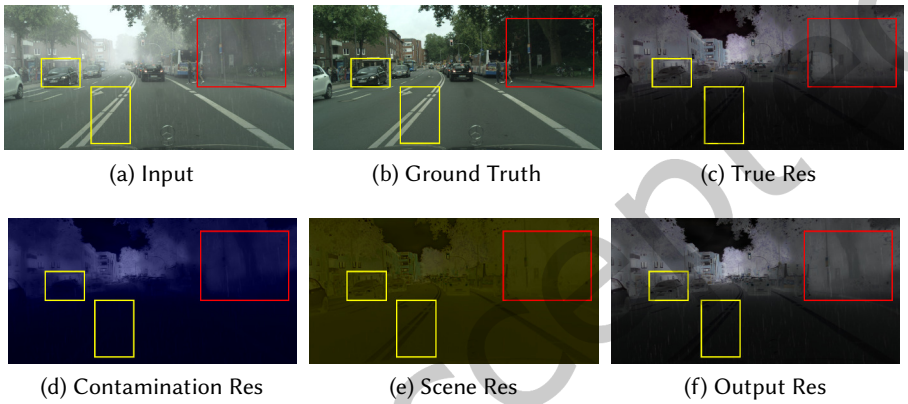


Fig. 1. Example of a scene image and its residual maps. (a) is the input image and (b) is the ground truth from RainCityscapes dataset. Image in (c) is the difference between (a) and (b). (d) and (e) are the contamination residual and scene residual. (f) is the result of (a)+(b). "Res" indicates "Residual". The contamination and scene details are included in the red and yellow boxes, respectively (zoom in to find the details).

The existing methods for single-image rain and haze removal can be roughly categorized into two categories: priority knowledge-oriented approaches and data-driven approaches. The prior knowledge-based image rain removal [24, 31, 36] and haze removal methods [15, 19, 63] are mostly based on the physical imaging models. However, such solutions suffer from the robustness issue when deployed into real-world scenarios [32, 62]. Recent advances in deep learning demonstrate dramatic success in haze removal [9, 27, 43] and rain removal [40, 49, 59]. Learning-based methods in both fields have achieved cutting-edge performance on synthetic datasets. However, methods designed for certain contamination cannot handle the complex real-world scenario with the co-occurrence of rain and haze in the natural scenes. Recent studies also pointed out the necessity of joint-removal, such as Han *et al.* [18] decompose rain and haze by a Blind Image Decomposition Network, and Kim *et al.* [25] remove rain and haze by a frequency-based model. A new dataset for the purpose of benchmarking joint rain and haze removal, named RainCityScapes, is also proposed to facilitate research on this important task [21]. Thus such a joint-removal task becomes an open problem in the community and calls for further study.

Recent advances in low-level computer vision have made remarkable progress, where a well-trained deep neural network can almost perfectly remove the contamination in outdoor scene images. However, no existing work considers paying attention to the scene difference in the restoration process. We observe that the true residual, obtained by (Input − Ground Truth) (see Fig. 1c), contains the scene information. That is, a neural

network designed to focus on contamination may suffer from a gap in recovering the scenes. Such a gap motivates us to develop a unified method to remove the contamination and compensate for the scene information in one go.

In real-world scenarios, the weather condition is complex, that is, different components, such as rain streaks and haze, may co-occur in the scenes. The occurrence of some components, i.e., heavy haze, impacts the atmospheric light. As a consequence, the scene information at the photometric level can be degraded. Physically speaking, along with removing contamination in the image, it is also necessary to restore scene information affected by the change of atmospheric light. To address this issue, we propose a novel dual-branch architecture, called Asymmetric Dual-decoder U-Net (ADU-Net). The ADU-Net consists of a single-branch encoder and an asymmetric dual-branch decoder. In the asymmetric dual-branch architecture, one branch, the contamination residual branch, is designed to remove the contamination (see Fig. 1d). Another branch, the scene residual branch, is required to perform the recovery of scene information (see Fig. 1e). The contamination residual branch, equipped with a novel channel feature fusion (CFF) module and window multi-head self-attention (W-MSA), produces the contamination residual. The special design allows the branch to focus more on the local foreground information in the image, thus extracting the contamination residual. The scene branch, powered by a novel global channel feature fusion (GCFF) module and shift-window multi-head self-attention (SW-MSA) mechanism, aims to compensate for the scene information. Unlike the contamination residual branch, the scene residual branch is designed to focus more on the global contextual information in the image, thus extracting the scene residual. The joint efforts of contamination residual and scene residual separate the rain and haze from the input scene image, while preserving the scene of the image (see Fig. 1f). The proposed ADU-Net can effectively remove the different contamination in the images and compensate for the scene information on multiple benchmark datasets, including RainCityscapes [21], BID rain [18] and SPA-Data [49].

Our contribution can be summarized as follows:

- We propose a novel yet efficient neural architecture, ADU-Net, to jointly remove rain and haze in scene images.
- We present an asymmetric dual-decoder, which removes the contamination while compensating for the scene information of the image. To the best of our knowledge, this is the first work to consider the recovery of scene information in deraining and dehazing tasks.
- Extensive experiments, including quantitative studies and qualitative studies, are conducted to evaluate the effectiveness of the ADU-Net. Empirical evaluation shows our method outperforms the current state-of-the-art methods by a considerable margin.

2 RELATED WORK

2.1 Single-image Rain Removal

The very first single-image rain removal methods were based on a priori knowledge. Morphological component analysis (MCA) [24] employs bilateral filters to extract high-frequency components from rain images, where the high-frequency components are further decomposed into "rain components" and "non-rain components" through dictionary learning and sparse coding. Luo *et al.* [36] proposed a single-image rain removal algorithm based on mutual exclusion dictionary learning. Gaussian mixture model prior knowledge [31] was utilized to accommodate multiple orientations and scales of rain streaks. In [62], Zhu *et al.* detected the approximate region, where the rain streaks were located, to guide the separation of the rain layer from the background layer. However, early models based on prior knowledge often suffer from a lack of stability in real scenarios [24, 31, 36]. Since 2017, deep learning approaches have been developed for rain removal tasks. Deep detail networks [16] narrowed the mapping from input to output and combined prior knowledge to capture high-frequency details, making the model stay focused on rain streaks information. By adding an iterative information feedback network, JORDER [53] used a binary mapping to locate rain streaks. A non-locally enhanced encoder-decoder structure [28] was proposed to

capture long-range dependencies and leverage the hierarchical features of the convolutional layer. In [30], Li *et al.* proposed a deep recurrent convolutional neural network to remove rain marks located at different depths progressively. A density-aware multi-stream connectivity network was introduced for rain removal in [58]. By adding constraints to the cGAN [23], Zhang *et al.* [59] generated more photo-realistic results. A progressive contextual aggregation network [40] was proposed as a baseline for rain removal. A real-world rain dataset was constructed by Wang *et al.* [49], they also incorporated spatial perception mechanisms into deraining networks. Recently, Zhu *et al.* [61] proposed a gated non-local depth residual network for image rain removal. Yu *et al.*[55] conducted a comprehensive analysis of various aspects of existing rain removal models and their robustness against adversarial attacks. Based on these analyses, they proposed a more robust approach to address this issue.

While significant progress has been made in the research on image rain removal, the existing studies lack consideration for real-world rainy scenarios, limiting their effectiveness in practical applications. In contrast, our methods take a more realistic approach by not only addressing rain streak occlusions commonly encountered in rainy weather but also considering the impact of haze, which is prevalent in the atmosphere, on atmospheric light. By incorporating these factors, our methods offer a more comprehensive and practical solution that better aligns with real-world conditions.

2.2 Single-image Haze Removal

{Similar to image rain removal methods, early work on image dehaze tended to employ statistical methods to acquire prior information by capturing patterns in haze-free images. Representative methods include Dark channel prior [19], color-line prior [15], colour attenuation prior [63], *etc.*. However, prior-based methods tend to distort colors and thus produce undesirable artifacts [15, 19, 63]. In the deep learning era, methods started to not rely on prior knowledge, but to estimate atmospheric light and the transmission map directly. For example, Cai *et al.* [5] proposed an end-to-end dehazing model named DehazeNet, where haze-free images are produced by learning the transmission rate. Similarly, Ren *et al.* [41] employed multi-scale deep neural networks to learn the mapping relationship between foggy images and their corresponding transmission maps, aiming to reduce the error in estimating the transmission maps. AODNet [27] reconstructed the atmospheric scattering model by leveraging an improved convolutional neural network to learn the mapping relationship between foggy and clean pairs. In [57], a single network was proposed to simultaneously learn the intrinsic relationship between transmission maps, atmospheric light, and clean images. Ren *et al.* [42] built an encoder-decoder neural network to enhance the dehazing process. A network with an enhancer and two generators was proposed by Qu *et al.* [39]. Chen *et al.* [9] proposed a patch map-based PMS-Net to effectively suppress the distorted color issue. Dong *et al.* [12] proposed MSBDN (Multi-Scale Boosted Dehazing Network) based on the U-Net architecture, incorporating boosting and error feedback as guiding principles. Although the method achieves good results, it suffers from a large number of parameters. Yeh *et al.* [54] decomposed hazy images into base components and detail components and proposed MSRL-DehazeNet, which based on residual learning and U-Net architecture. Sun *et al.* [46] proposed SADNet based on the attention mechanism using a semi-supervised approach for solving practical problems. Song *et al.* [45] introduced Swin Transformer into image haze removal and proposed DehazeFormer, which achieved significant improvements on multiple datasets. Unlike image rain removal, image dehazing often consider the impact of haze on atmospheric light intensity, which can compensate for the limitations in rain removal methods. Our methods combine these approaches with the research on rain removal, resulting in a more realistic approach that better aligns with real-world scenarios.

2.3 Other Related Works

Unlike previous single-task models, some researchers have also explored the simultaneous enhancement of both rain removal and haze removal in images. Hu *et al.* [21] built an imaging model for rain streaks and haze based on

the visual effect of rain and the scene depth map to synthesize a realistic dataset named RainCityscapes. Han *et al.* [18] constructed a superimposed image dataset and proposed a simple yet general Blind Image Decomposition Network to decompose rain streaks, raindrops, and haze in a blind image decomposition setting. Kim *et al.* [25] proposed a frequency-based model for removing rain and haze, where the frequency-based model divided the input image into high-frequency and low-frequency parts with a guided filter and then employed a symmetric encoder-decoder network to remove rain and haze separately. Kulkarni *et al.* [26] proposed a lightweight network that combines convolutions at different scales with spatial attention and channel attention mechanisms, employing a dual restoration mechanism to handle images affected by various weather conditions. Recently, Li *et al.* [29] used a neural structure search based approach to handle multiple weather situations, however, it has a large number of parameters as it uses multiple encoders for each weather removal task. Chen *et al.* [10] proposed a training approach based on knowledge distillation, considering the perspective of training strategies. They introduced a multi-teacher model and a single-student model, enabling a single model to handle various weather conditions without increasing the parameter size. Valanarasu *et al.* [47] proposed a single transformer-based encoder-decoder network while restored image with a learnable weather type query in the decoder to learn the type of weather degradation. Wang *et al.* [50] enhanced the U-Net architecture by adding a small decoder and a dilated convolution attention module. This enhancement enabled the network to capture both global information and finer details in high-resolution remote sensing images. After an in-depth study of related works, we have identified two primary factors that contribute to image quality degradation: contamination and scene information affected by atmospheric light. To effectively address these factors, we introduce an innovative asymmetric dual-branch structure, allowing independent processing of each category. By separately optimizing contamination removal and scene information recovery, our method achieves enhanced overall performance and improved image quality.

3 METHOD

This section details the proposed method in a top-down fashion: starting from the problem formulation of our application, followed by the architecture of the proposed Asymmetric Dual-decoder U-Net (ADU-Net) and its building block, namely asymmetric dual-decoder block (ADB).

Notations. Throughout the paper, we use bold capital letters to denote matrices or tensors (e.g., $\boldsymbol{X}$), and bold lower-case letters to denote vectors (e.g., $\boldsymbol{x}$).

3.1 Problem Formulation

Let a third-order tensor, $\boldsymbol{I} \in \mathbb{R}^{C\times H\times W}$, denote an input image, where C, H and W present the channel, height, and width of the image, respectively. In our application, both rain and haze are synthesized into the origin scene images as input images. Each input image $\boldsymbol{I}$ is labeled with its ground truth image $\boldsymbol{I}^{\text{gt}}$ without rain and haze in the scene. Our ADU-Net f_θ, consisting of a single branch encoder f_{E}, and an asymmetric dual-decoder f_{AD}, can remove the rain and haze in the input image, such that the output of the ADU-Net, $\boldsymbol{Y} = f_\theta(\boldsymbol{I})$ can restore its ground truth scene $\boldsymbol{I}^{\text{gt}}$. The ADU-Net is trained to learn a set of parameters, θ^*, with minimum empirical objective value $\mathcal{L}(\boldsymbol{I}^{\text{gt}}, \boldsymbol{Y})$.

3.2 Network Overview

We first give a sketch of the proposed ADU-Net. In rain and haze removal applications, one ideal option is to employ the deep neural network to understand the scene of the input image and separate the rain and haze from the input image. In our work, we develop the ADU-Net to remove the rain and haze jointly. As shown in Fig. 2, the ADU-Net is stacked by a single branch encoder and an asymmetric dual-decoder. In the encoder f_{E}, we have

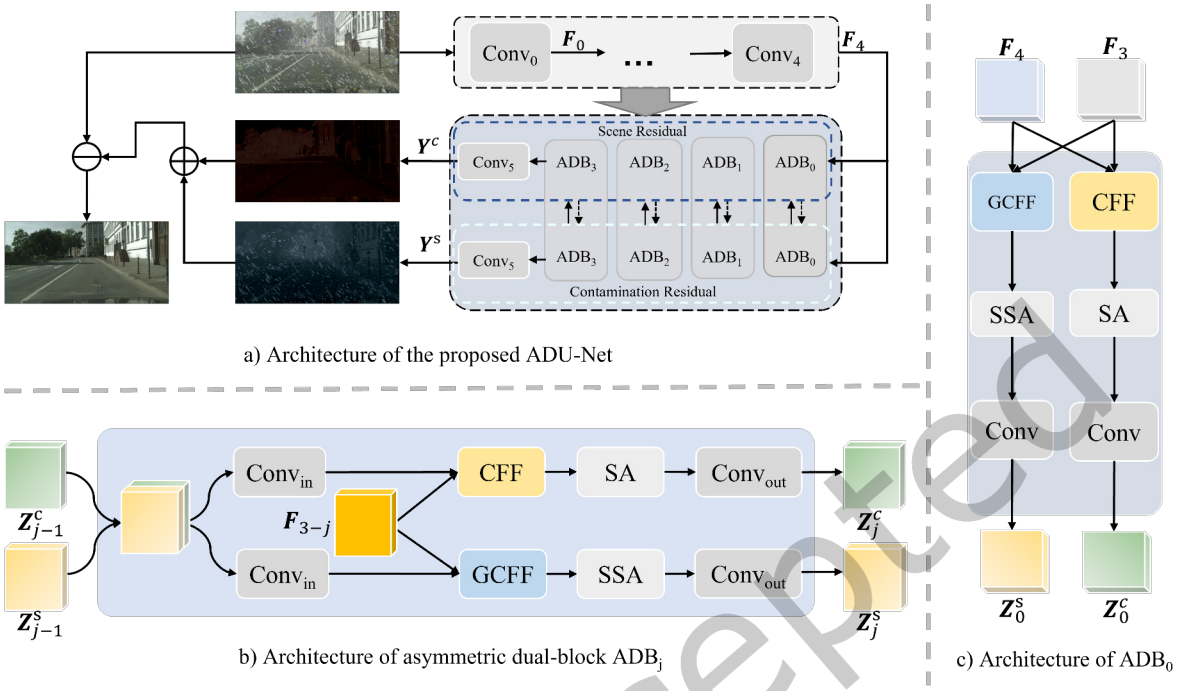


Fig. 2. The network architecture of the proposed ADU-Net, which consists of an encoder f_{E} and an asymmetric dual-decoder f_{AD}. f_{E} has five Conv_i blocks and f_{AD} has four ADB_j blocks and a Conv block. The network is optimized by the SSIM loss function.

five convolutional blocks, with each denoted by Conv_i, $0 \le i \le 4$. The output of each convolutional block is denoted by $\boldsymbol{F}_i = \mathrm{Conv}_i(\boldsymbol{F}_{i-1})$ and $\boldsymbol{F}_{-1} = \boldsymbol{I}$.

Then a following asymmetric dual-decoder f_{AD} aims to recover the scene image without rain and haze (see Fig. 2). The proposed asymmetric dual-decoder is stacked of a set of ADBs, which produce two streams of latent features, denoted by $\boldsymbol{Z}^{\mathrm{c}}_j$ and $\boldsymbol{Z}^{\mathrm{s}}_j$ in the j-th ADB. Specifically, the processing can be formulated as

$$\boldsymbol{Z}^{\mathrm{c}}_0, \boldsymbol{Z}^{\mathrm{s}}_0 = \mathrm{ADB}_0(\boldsymbol{F}_3, \boldsymbol{F}_4), \tag{1}$$

or

$$\boldsymbol{Z}^{\mathrm{c}}_j, \boldsymbol{Z}^{\mathrm{s}}_j = \mathrm{ADB}_j(\boldsymbol{Z}^{\mathrm{c}}_{j-1}, \boldsymbol{Z}^{\mathrm{s}}_{j-1}, \boldsymbol{F}_{3-j}),\ j > 0. \tag{2}$$

After the last ADB, each stream of latent features $\boldsymbol{Z}^{\mathrm{c}}_3$ or $\boldsymbol{Z}^{\mathrm{s}}_3$ is encoded by a convolutional block to recover the channel dimensions into the image space (e.g., $C = 3$), as $\boldsymbol{Y}^{\mathrm{c}} = \mathrm{Conv}_5(\boldsymbol{Z}^{\mathrm{c}}_3)$ and $\boldsymbol{Y}^{\mathrm{s}} = \mathrm{Conv}_5(\boldsymbol{Z}^{\mathrm{s}}_3)$. We denote the $\boldsymbol{Y}^{\mathrm{c}}$ as the contamination residual, and $\boldsymbol{Y}^{\mathrm{s}}$ as the scene residual. Having the $\boldsymbol{Y}^{\mathrm{c}}$ and $\boldsymbol{Y}^{\mathrm{s}}$ at hand, one can obtain the restored scene image $\boldsymbol{Y}$ as

$$\boldsymbol{Y} = \boldsymbol{I} - \boldsymbol{Y}^{\mathrm{c}} - \boldsymbol{Y}^{\mathrm{s}}. \tag{3}$$

The network is optimized by the negative SSIM loss [51] as $\mathcal{L}_{\mathrm{SSIM}} = -\mathrm{SSIM}(\boldsymbol{I}^{\mathrm{gt}}, \boldsymbol{Y})$. Noted the common practice uses both the negative SSIM loss and MSE loss as the objective. Empirically we observed that a negative SSIM loss works better in the proposed ADU-Net, which will be justified in § 4.4.

3.3 Asymmetric Dual-decoder Block

In this part, we will describe the asymmetric dual-decoder f_{AD} in ADU-Net. As shown in Fig. 2, f_{AD} consists of four ADBs and a convolutional block, while the ADBs are stacked by two different instantiations (e.g., ADB_0 vs. $\mathrm{ADB}_j,\ j = 1, 2, 3$). In this following, we will first describe ADB_0, a simple form of the block. Then with minor modifications, we can realize the ADB_j, $j = 1, 2, 3$ on top of the ADB_0.

The ADB_0 is a two branch architecture (see Fig. 2), which receives the $\boldsymbol{F}_3$ and $\boldsymbol{F}_4$ as input, and produces two latent features $\boldsymbol{Z}_0^{\mathrm{c}}$ and $\boldsymbol{Z}_0^{\mathrm{s}}$. In ADB_0, the two latent features are respectively encoded by two branches of network, namely contamination residual net (denoted by g^{c}), and scene residual net (denoted by g^{s}), given by

$$\boldsymbol{Z}_0^{\mathrm{c}} = g^{\mathrm{c}}(\boldsymbol{F}_3, \boldsymbol{F}_4) \tag{4}$$

and

$$\boldsymbol{Z}_0^{\mathrm{s}} = g^{\mathrm{s}}(\boldsymbol{F}_3, \boldsymbol{F}_4). \tag{5}$$

Contamination Residual Net. In the contamination residual net (g^{c}), $\boldsymbol{F}_3$ and $\boldsymbol{F}_4$ are fed to a channel feature fusion (CFF) module to localize the rain and haze areas in the scene image, as

$$\boldsymbol{G}_0^{\mathrm{c}} = \mathrm{CFF}(\boldsymbol{F}_3, \boldsymbol{F}_4). \tag{6}$$

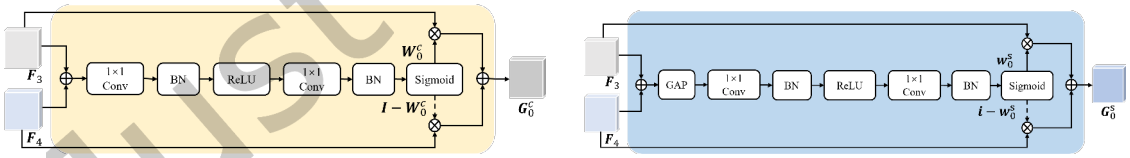
The details of CFF are illustrated in Fig. 3a. Given two feature maps $\boldsymbol{F}_3$ and $\boldsymbol{F}_4$ as input, it first fuses the two inputs by using element-wise addition and then feeds the fused feature maps to 2-layer convolutional blocks to obtain the attention weights, formulated by

$$\boldsymbol{W}_0^{\mathrm{c}} = \sigma\Big(\mathrm{BN}\big(\mathrm{Conv}(\mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}(\boldsymbol{F}_3 \oplus \boldsymbol{F}_4))))\big)\Big), \tag{7}$$

where σ, BN, ReLU are sigmoid function, batch normalization, and rectified linear unit activation, respectively. Here, the kernel size of Conv is 1×1, which can be understood as applying a fully-connected layer to the channel features.

Then we can apply the attention weights to the input feature maps and obtain the fused output, as

$$\boldsymbol{G}_0^{\mathrm{c}} = \big(\boldsymbol{W}_0^{\mathrm{c}} \otimes \boldsymbol{F}_3\big) \oplus \big((\boldsymbol{I} - \boldsymbol{W}_0^{\mathrm{c}}) \otimes \boldsymbol{F}_4\big). \tag{8}$$



(a) Channel Feature Fusion Module (b) Global Channel Feature Fusion Module

Fig. 3. Architecture of the global channel feature fusion module and channel feature fusion module.

The CFF module fuses the input feature maps, and the fusion weights are produced via the channel patterns. We further employ a self-attention mechanism to build the spatially long-range dependencies of the fused feature maps $\boldsymbol{G}_0^{\mathrm{s}}$, given by

$$\boldsymbol{H}_0^{\mathrm{c}} = \text{W-MSA}(\boldsymbol{G}_0^{\mathrm{c}}), \tag{9}$$

where W-MSA is the window multi-head self-attention from the Swin Transformer [35].

Having fusing the input feature maps and being processed by the attention mechanism, we can obtain the contamination residual feature maps as

$$\boldsymbol{Z}_0^{\mathrm{c}} = \mathrm{Conv}(\boldsymbol{H}_0^{\mathrm{c}}). \tag{10}$$

The contamination residual net (g^{c}) aims to attend to the rainy and hazy regions, thereby highlighting the rain and haze components in the contamination residual feature maps.

Scene Residual Net. Since we can observe from the contamination residual ($\boldsymbol{Y}^{\mathrm{c}}$) that it contains the scene information along with the rain and haze, we develop a scene residual net (g^{s}), that can compensate for the removed scene information in the image. In doing so, the global channel feature fusion (GCFF) module is proposed to capture valuable global scene information of the image, and fuse features, as

$$\boldsymbol{G}_0^{\mathrm{s}} = \mathrm{GCFF}(\boldsymbol{F}_3, \boldsymbol{F}_4). \tag{11}$$

As shown in Fig. 3b, $\boldsymbol{F}_3$ and $\boldsymbol{F}_4$ are first fused, and summarized to its global feature, as

$$\boldsymbol{m}_0^{\mathrm{s}} = \mathrm{GAP}(\boldsymbol{F}_3 \oplus \boldsymbol{F}_4), \tag{12}$$

where GAP indicates the global average pooling and $\boldsymbol{m}_0^{\mathrm{s}}$ indicates the resultant vector. Then a 2-layer convolutional block is used to modulate per element of the global feature $\boldsymbol{m}_0^{\mathrm{s}}$, written by:

$$\boldsymbol{w}_0^{\mathrm{s}} = \sigma\Big(\mathrm{BN}\big(\mathrm{Conv}(\mathrm{ReLU}(\mathrm{BN}(\mathrm{Conv}(\boldsymbol{m}_0^{\mathrm{s}}))))\big)\Big). \tag{13}$$

We can thereby fuse the input feature maps as:

$$\boldsymbol{G}_0^{\mathrm{s}} = \big(\boldsymbol{w}_0^{\mathrm{s}} \otimes \boldsymbol{F}_3\big) \oplus \big((\boldsymbol{i} - \boldsymbol{w}_0^{\mathrm{s}}) \otimes \boldsymbol{F}_4\big). \tag{14}$$

where $\boldsymbol{i}$ indicates a unit vector with the same size as $\boldsymbol{w}_0^{\mathrm{s}}$.

After GCFF, we employ the shift-window multi-head self-attention (SW-MSA) to enhance the spatial interaction of the feature maps and obtain the scene residual features, described by

$$\boldsymbol{H}_0^{\mathrm{s}} = \text{SW-MSA}(\boldsymbol{G}_0^{\mathrm{s}}), \tag{15}$$

and

$$\boldsymbol{Z}_0^{\mathrm{s}} = \mathrm{Conv}(\boldsymbol{H}_0^{\mathrm{s}}). \tag{16}$$

Instantiation of $\boldsymbol{ADB}_j$. The difference between $\mathrm{ADB}_j, j \neq 0$ and ADB_0 is that ADB_0 receives two feature maps as input, while $\mathrm{ADB}_j, j \neq 0$ includes three feature maps as input. To adapt the architecture of ADB_0 to $\mathrm{ADB}_j, j \neq 0$, we make minor modification (see Fig. 2). Specifically, for any block, ADB_j, its input includes the output from $j-1$-th ADB blcok, e.g., $\boldsymbol{Z}_{j-1}^{\mathrm{c}}, \boldsymbol{Z}_{j-1}^{\mathrm{s}} \in \mathbb{R}^{d \times h \times w}$, and from the $3-j$-th convolutional encoder, e.g., $\boldsymbol{F}_{3-j}$. We first concatenate the $\boldsymbol{Z}_{j-1}^{\mathrm{c}}, \boldsymbol{Z}_{j-1}^{\mathrm{s}}$, and reduce its dimension from $2d \times h \times w$ to $d \times h \times w$, as

$$\bar{\boldsymbol{Z}}_{j-1} = \mathrm{Concat}(\boldsymbol{Z}_{j-1}^{\mathrm{c}}, \boldsymbol{Z}_{j-1}^{\mathrm{s}}). \tag{17}$$

and

$$\tilde{\boldsymbol{Z}}_{j-1}^{\mathrm{c}} = \mathrm{Conv}_{\mathrm{in}}(\bar{\boldsymbol{Z}}_{j-1}),\ \tilde{\boldsymbol{Z}}_{j-1}^{\mathrm{s}} = \mathrm{Conv}_{\mathrm{in}}(\bar{\boldsymbol{Z}}_{j-1}). \tag{18}$$

where $\mathrm{Conv}_{\mathrm{in}}$ indicates a 2-layer convolution block with batch normalization and rectified linear unit activation. Here, the kernel size of convolution layers is 3×3.

With $\boldsymbol{F}_{3-j}$, the output of ADB_j can be obtained as

$$\begin{aligned} \boldsymbol{Z}_j^{\mathrm{c}} &= g^{\mathrm{c}}(\tilde{\boldsymbol{Z}}_{j-1}^{\mathrm{c}}, \boldsymbol{F}_{3-j}) \\ &= \mathrm{Conv}_{\mathrm{out}}\Big(\text{W-MSA}\big(\mathrm{CFF}(\tilde{\boldsymbol{Z}}_{j-1}^{\mathrm{c}}, \boldsymbol{F}_{3-j})\big)\Big) \end{aligned} \tag{19}$$

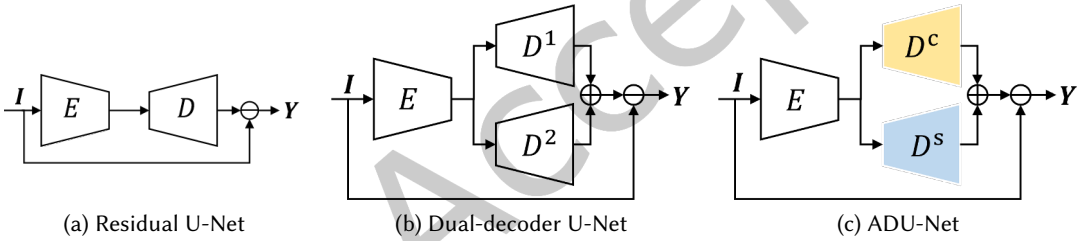
and

$$\begin{aligned} Z_j^{\mathrm{s}} &= g^{\mathrm{s}}(\tilde{Z}_{j-1}^{\mathrm{s}}, \boldsymbol{F}_{3-j}) \\ &= \mathrm{Conv}_{\mathrm{out}}\Big(\mathrm{SW\text{-}MSA}\big(\mathrm{GCFF}(\tilde{Z}_{j-1}^{\mathrm{s}}, \boldsymbol{F}_{3-j})\big)\Big). \end{aligned} \tag{20}$$

Here, $\mathrm{Conv}_{\mathrm{out}}$ indicates a convolution layer with the kernel size of 3×3 followed by a leaky rectified linear unit and another convolution layer with the kernel size of 1×1 also followed by a leaky rectified linear unit.

In this work, we propose a novel architecture for the rain and haze removal task. Considering the network capacity and hardware overhead, we propose two sizes of networks. One is the lite network, called ADU-Net, and the other is the large network, called ADU-Net-plus. In § 4, we present the details of two architectures. The network performance is also evaluated in § 4.

Remark 1. *The residual U-Net architecture has been used extensively for the rain or haze removal tasks [6], as shown in Fig. 4a. Having the observation that the contamination residual, produced by the decoder, contains the scene information, we aim to develop a dual-decoder U-Net, with one decoder producing the contamination residual, and another one producing the scene residual as a scene compensator. Its initial design is shown in Fig. 4b. Considering the physical property of the contamination and scene information in the input image, we propose a novel network architecture, ADU-Net, where we integrate two decoders with non-identical architectures (see Fig. 4c). We justify our design in § 4.4.*



(a) Residual U-Net (b) Dual-decoder U-Net (c) ADU-Net

Fig. 4. Schematic comparison of the ADU-Net architecture and U-Net-based architectures. (a) is a vanilla architecture of the residual U-Net. (b) is a simple form of the residual U-Net with dual decoders. (c) is the diagram of our method.

4 EXPERIMENTS

In this section, we first give the implementation details of the proposed ADU-Net and ADU-Net-plus. Then the benchmark datasets and evaluation protocol are also introduced. We further compare our network to the state-of-the-art methods and conduct ablation studies to evaluate the superiority of the proposed network and each component. In the final part, we demonstrate substantial qualitative results to analyze the superior performance of our network.

4.1 Implementation Details

Network Architecture. The overall neural architecture of the proposed network is shown in Fig. 2. Table 1 lists the kernel size of the convolutional layers. In the encoder block, the feature maps are processed by the Batch Normalization [22] and ReLU [1] after the convolutional layer, i.e., Conv_0, Conv_1, Conv_2, Conv_3, Conv_4. Then the max-pooling layer is employed to down-sample the feature maps in each layer. In the decoder block, we also list the kernel size in the convolutional layers (see Table 1), and employ the Leaky ReLU as the activation

Table 1. Details of the kernel size in convolution layers. H and W denote the height and width of the input image, respectively.

Layer name		Output size	ADU-Net	ADU-Net-plus
Conv_0		$H \times W$	$\begin{bmatrix} 3\times3, 32 \\ 3\times3, 32 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}$
Conv_1		$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}$
Conv_2		$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}$
Conv_3		$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}$
Conv_4		$\frac{H}{16} \times \frac{W}{16}$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}$
ADB_0		$\frac{H}{8} \times \frac{W}{8}$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}$
ADB_1	Conv_{in}	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}$
	Conv_{out}	$\frac{H}{4} \times \frac{W}{4}$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}$
ADB_2	Conv_{in}	$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}$
	Conv_{out}	$\frac{H}{2} \times \frac{W}{2}$	$\begin{bmatrix} 3\times3, 32 \\ 3\times3, 32 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}$
ADB_3	Conv_{in}	$H \times W$	$\begin{bmatrix} 3\times3, 32 \\ 3\times3, 32 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}$
	Conv_{out}	$H \times W$	$\begin{bmatrix} 3\times3, 16 \\ 3\times3, 16 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 32 \\ 3\times3, 32 \end{bmatrix}$
Conv_5		$H \times W$	$\begin{bmatrix} 3\times3, 3 \\ 3\times3, 3 \end{bmatrix}$	$\begin{bmatrix} 3\times3, 3 \\ 3\times3, 3 \end{bmatrix}$
Parameter size			6.63×10^6	26.45×10^6

function. Having computational efficiency in mind, we develop two neural networks of different scales. The light one is denoted as ADU-Net, while the large one is denoted as ADU-Net-plus. As shown in Table 1, the difference between the two networks is merely the modification to the channel dimensions. The superiority of our network will be evaluated in § 4.3.

Network Training. We implement our method using PyTorch deep learning package [37]. All experiments are evaluated on NVIDIA GTX 2080ti GPUs. In the experiments for RainCityscapes [21], BID Rain datasets [18] and NH-HAZE [2], the input images are resized to 512×256. For the SPA-Data, we follow the practice in [49], which uses original images with size of 256×256. The Adam optimization scheme with an initial learning rate of 0.001 is used to optimize the network. We train the network for 100 epochs for RainCityscapes and BID Rain datasets, and 20 epochs for SPA-Data. The learning rate adjustment strategy is employed to realize the learning rate decay, where the learning rate is decayed by a factor of 0.1 when the accuracy of the network does not improve in 5 epochs.

4.2 Datasets and Evaluation Protocol

We evaluate the proposed methods on two synthetic datasets, i.e., RainCityscapes [21], BID Rain [18], and two real-world datasets, i.e., SPA-Data [49], and NH-HAZE [2]. In the following, we will introduce these datasets and the statistics of each dataset are illustrated in Table 2.

RainCityscapes. The RainCityscapes dataset is synthesized from the Cityscapes dataset [11]. It takes 9,432 images synthesized from 262 Cityscapes images as the training set and 1,188 images synthesized from 33 Cityscapes images as the test set. All the selected images of Cityscapes are overcast, without obvious shadow. {Rain streaks and haze are synthesized by different intensity maps. By adjusting the intensity of the rain streaks and haze, each original image can produce 36 different synthesized images. The results of different methods are reported in Table 3.

BID Rain. The BID Rain dataset is also synthesized from the Cityscapes dataset. It samples 2,975 images from the validation set of the Cityscapes dataset as a training set, and 500 images from the test set of the Cityscapes dataset as its test set. This is a complicated dataset as the images contain rain streaks, haze, snow, and raindrops. The rain streaks masks are sampled from Rain100L and Rain100H [53], and the snow masks are sampled from Snow 100K [34]. The haze masks include three different intensities originating from FoggyCityScape [44]. The raindrops are produced from the metaball model [4]. Those weather components are mixed with the images in the Cityscapes dataset using the physical imaging models [4, 19, 34, 44, 53]. In the training set, every image can be mixed with each weather component with random probabilities, and we evaluate our model in six different cases, the combinations of the weather components in each case are as follows (1): rain streaks, (2): rain streaks and snow, (3): rain streaks and light haze, (4): rain streaks and heavy haze, (5): rain streaks, moderate haze and raindrops and (6): rain streaks, snow, moderate haze and raindrops. Refer [18] for more details of six settings. The results of different cases are shown in Table 4.

SPA-Data. The SPA-Data is a real-world dataset, which is cropped from 170 real rain videos, of which 86 videos are collected from StoryBlocks or YouTube, and 84 videos are captured by iPhone X or iPhone 6SP. Those videos cover outdoor fields, suburb scenes, and common urban scenes. This dataset contains 638,492 image pairs for training and 1,000 for testing. The results of SPA-Data are shown in Table 5.

NH-HAZE The NH-HAZE [2] is a valuable dataset for non-homogeneous haze research, as it offers ground truth images for evaluation. The dataset comprises 55 pairs of real-world outdoor scenes, where each pair consists of a hazy image and its corresponding haze-free counterpart. The non-homogeneous haze present in these images has been meticulously generated using a professional haze generator, ensuring an accurate representation of real-life haze conditions. The results on the NH-HAZE dataset are presented in Table 6.

Evaluation Protocol. In our experiments, the network performance is quantitatively evaluated by the peak signal-to-noise ratio (PSNR) and structural similarity (SSIM) metrics. A higher value of PSNR and SSIM indicates a better image recovery performance of the network.

Table 2. The Statistics of Datasets.

Dataset	Train set	Test set	Property		Contamination			
			Synthetic	Real world	Rain streaks	Haze	Snow	Raindrops
RainCityscapes	9,432	1,188	✓		✓	✓		
BID Rain	2,975	500 * 6	✓		✓	✓	✓	✓
SPA-Data	638,492	1,000		✓	✓			
NH-HAZE	40	15		✓		✓		

4.3 Comparison to the State-of-the-Arts

To verify the advance of our method, we compare the performance of our method with current state-of-the-art methods across three datasets.

RainCityscapes. In the RainCityscapes dataset, we compare our methods to the the state-of-the-art rain removal methods including RESCAN [28], PReNet [40], DuRN [33], RCDNet [48], SPANet [49] and MPRNet [56]. We also compare our methods with approaches that jointly remove the rain and haze, i.e., DAF-Net [20], DGNL-Net [21], WiperNet [26], TransWeather [47] and GTRain [3]. The comparison with haze removal methods, like EPDN [39], DCPDN [57], AECR-Net [52], is also conducted. The results are reported in Table 3. We can find that our vanilla solution, i.e. ADU-Net, outperforms the existing state-of-the-art methods. In particular, it improves the PSNR/SSIM values of the DGNL-Net by 1.45/0.0017, indicating the superior design of our method. The plus version of our method, i.e., ADU-Net-plus, again brings performance gain over the AUD-Net, where the ADU-Net-plus improves the PSNR/SSIM values by 0.81/0.0021.

Table 3. Comparison with the State-of-the-Arts Methods of rain removal and haze removal on RainCityscapes dataset. † indicates the network was trained on the RainCityscapes dataset. ‡ indicates the results of the algorithms as reported in [21]. 1st/2nd best in red/blue.

Method		PSNR	SSIM
Input		15.55	0.7722
Haze removal	EPDN‡[39]	26.08	0.9306
	DCPDN‡[57]	28.52	0.9277
	AECRNet†[52]	28.77	0.9350
Rain removal	RESCAN‡[30]	24.49	0.8852
	PReNet†[40]	27.34	0.9497
	DuRN‡[33]	29.43	0.9487
	RCDNet†[48]	30.56	0.8873
	SPANet‡[49]	31.48	0.9656
	MPRNet†[56]	32.33	0.9767
Rain and haze removal	DAF-Net†[20]	30.16	0.9531
	DGNL-Net†[21]	32.38	0.9743
	TransWeather†[47]	29.28	0.9216
	GTRain†[3]	30.19	0.9597
	WiperNet†[26]	30.21	0.9584
	ADU-Net	33.83	0.9784
	ADU-Net-plus	34.64	0.9805

BID Rain. Since the scene in the RainCityscapes dataset only contains rain and haze information, we further evaluate our methods on the challenging dataset, BID Rain, to verify its generalization of working in complicated weather conditions. Table 4 illustrates the comparison of the model performance in each weather condition. We can observe that the proposed ADU-Net can outperform the BIDeN [18] in each of the cases. Especially in cases (2) and (3), the ADU-Net brings the maximum performance gain. One possible explanation is that the proposed ADU-Net is designed with a dual-branch decoder, which is tailored for the images in case (2) including the rain streaks and snow, or that in (3) including rain streaks and a light haze. However, the improvement in the other cases reveals the generalization of our proposal. Along with the ADU-Net, its plus version can significantly improve both PSNR/SSIM values, showing the superiority of our network architecture. In case (4),

Table 4. Comparison with the State-of-the-Arts Methods on BID Rain dataset. † indicates the network was trained on the BID Rain dataset. 1st/2nd best in red/blue.

Case		Input	PReNet	RCDNet	BIDNet	TransWeather	GTrain	ADUNet	ADUNet-plus
(1)	PSNR	25.51	32.69	28.05	31.17	31.88	31.91	34.62	39.05
	SSIM	0.8144	0.9803	0.9527	0.9438	0.9307	0.9596	0.9827	0.9877
(2)	PSNR	18.69	30.52	29.84	29.47	29.37	30.03	32.47	36.48
	SSIM	0.5979	0.9504	0.9351	0.9089	0.8844	0.9178	0.9560	0.9742
(3)	PSNR	17.48	29.65	30.17	28.90	29.46	30.14	31.48	33.75
	SSIM	0.7427	0.9568	0.9536	0.9325	0.9176	0.9470	0.9669	0.9777
(4)	PSNR	11.55	25.80	26.74	26.82	27.51	27.33	26.52	29.30
	SSIM	0.6017	0.9233	0.9210	0.9125	0.8949	0.9222	0.9360	0.9565
(5)	PSNR	14.02	27.36	28.30	27.31	26.94	27.74	28.54	30.32
	SSIM	0.6455	0.9302	0.9285	0.9116	0.8833	0.9191	0.9443	0.9594
(6)	PSNR	12.38	26.56	27.26	26.54	26.22	26.85	27.63	29.66
	SSIM	0.4916	0.9046	0.9005	0.8675	0.8504	0.8857	0.9222	0.9418

Table 5. Comparison with the State-of-the-Arts Methods on SPA-Data dataset. ‡ indicates the results of the algorithms as reported in [48]. † indicates the network was trained on the SPA-data. 1st/2nd best in red/blue.

Method	PSNR	SSIM
Input	34.15	0.9269
RESCAN‡[30]	38.19	0.9707
PReNet‡[40]	40.16	0.9816
SPANet‡[49]	40.24	0.9811
RCDNet‡[48]	41.47	0.9834
TransWeather†[47]	38.31	0.9757
WiperNet†[26]	41.73	0.9905
ADU-Net	44.19	0.9885
ADU-Net-plus	46.04	0.9924

the performance of ADU-Net is lower than that of RCDNet [48], BIDeN [18], TransWeather [47] and GTrain [3]. One possible explanation is that the "heavy haze" covers the scenes, which makes it difficult for our network to produce the scene residual. Nevertheless, this issue is addressed by increasing the parameter size, supported by the performance in ADU-Net-plus.

SPA-Data. We also evaluate our methods in the large-scale dataset, SPA-Data. We compare our methods to the existing state-of-the-art methods in Table 5, including RESCAN[30], PReNet[40], SPANet[49], RCDNet[48] and WiperNet[26]. As shown in Table 5, the proposed methods outperform the existing methods by a large margin. For example, the improvements read of 2.72/0.0051 (PSNR/SSIM) from ADU-Net and 4.57/0.0090 from ADU-Net-plus, as compared to RCDNet, showing the strong performance of our network architecture. Indeed, although ADU-Net exhibits a slightly lower SSIM compared to WiperNet by 0.0020, its superiority in PSNR by 2.46 highlights its excellent performance. Furthermore, ADU-Net-plus outperforms WiperNet, leading in both PSNR and SSIM by 4.31 and 0.0019, respectively. These findings affirm the robustness and efficacy of our proposed methods for image rain removal in real-world scenarios.

Table 6. Comparison with the State-of-the-Arts Methods on NH-HAZE dataset. ‡ indicates the results of the algorithms as reported in [45]. † indicates the network was trained on the SPA-data. $1^{st}/2^{nd}$ best in red/blue.

Method	PSNR	SSIM
Input	11.48	0.4023
DehazeNet‡[5]	16.62	0.524
FFA-Net‡[38]	19.87	0.692
MSBDN‡[12]	19.23	0.706
AECRNet‡[52]	19.88	0.717
DehazeFormer-S‡[45]	20.47	0.731
ADU-Net	28.37	0.887
ADU-Net-plus	29.46	0.890

NH-HAZE. To showcase the effectiveness of our approach, we conducted experiments using the real-world NH-HAZE dataset [2]. In Table 6, we compare our methods with state-of-the-art techniques, including MSBDN [12], FFA-Net [38], AECRNet [52], and DehazeFormer [45]. The results, as presented in Table 6, demonstrate significant performance improvements with our proposed methods surpassing existing approaches by a wide margin. For instance, when compared to DehazeFormer-S, ADU-Net and ADU-Net-plus exhibit remarkable improvements of 7.9/0.156 (PSNR/SSIM) and 8.99/0.159, respectively. These outcomes highlight the strong performance of our network architecture.

Comparison of Model Complexity and Time Cost. In addition to analyzing PSNR and SSIM, we also compare the complexity of our methods with the existing state-of-the-art methods in Table 7, including PReNet [40], RCDNet [48], AECRNet [52] and DGNL-Net [21]. We employ GFLOPs (Giga Floating Point Operations) to quantify the complexity of the model. Additionally, we assess the run-time complexity by averaging the training time per epoch (s/epoch). It is evident from the table that ADU-Net stands out with the smallest GFLOPs and the second shortest s/epoch. Although our model's runtime is slightly higher compared to DGNL-Net [21], we have demonstrated superior performance on both PSNR and SSIM. In our proposed methods, the convolutional layers play a significant role in the overall computational load. We are proud to share that ADU-Net achieves an impressive 50% reduction in kernel size compared to ADU-Net-plus while maintaining an exceptional performance level of 95%. We believe this significant reduction in complexity makes ADU-Net a more efficient and practical choice for various applications. However, in cases where computational resources are abundant and time is not a constraint, choosing ADU-Net-plus to achieve better performance is also a viable option.

Table 7. Model Complexity and Run-time Cost. The PSNR/SSIM are the results of the RainCityscapes dataset. $1^{st}/2^{nd}$ best in red/blue.

Model	GFLOPs	s/epoch	PSNR	SSIM
PReNet[40]	132.88	2145.45	27.34	0.9497
RCDNet[48]	48.46	3509.27	30.56	0.8873
AECRNet[52]	86.09	1113.68	28.77	0.9350
DGNL-Net[21]	39.53	837.28	32.38	0.9743
ADU-Net	31.65	1017.21	33.83	0.9784
ADU-Net-plus	125.71	1126.49	34.64	0.9805

4.4 Ablation Study

In this section, we conduct thorough ablation studies to verify the effectiveness per component in the proposed network. All studies in this section are conducted using ADU-Net on the RainCityscapes dataset.

Loss Function. In our implementation, the network is optimized by the negative SSIM loss, i.e., $\mathcal{L}_{\text{SSIM}}$. While in many practices of the low-level computer vision tasks, the MSE loss i.e., $\mathcal{L}_{\text{MSE}}$, is employed [14]. In this study, we evaluate the effectiveness of each loss function. As shown in Table 8, we can find that each of the loss functions works better for our rain and haze removal task, and the network performance training from the two-loss functions is similar. However, the multi-task training, which optimizes the loss functions jointly, will degrade the network performance, indicating that the network may be saturated using one loss function, and the joint training will harm the network.

Table 8. Comparison of the effectiveness of Loss Functions. We use **bold** to indicate best the result.

Loss Function	$\mathcal{L}_{\text{MSE}}$	$\mathcal{L}_{\text{SSIM}}$	$\mathcal{L}_{\text{MSE}} + \mathcal{L}_{\text{SSIM}}$
PSNR	33.17	**33.83**	33.74
SSIM	0.9720	**0.9784**	0.9774

Effect of Dual-branch Architecture. Our work naively proposes a dual-branch architecture, i.e., asymmetric dual-decoder U-Net, for rain and haze removal tasks. In this study, we will justify the effectiveness of the dual-branch design in our task (shown in Fig 4). Table 9 shows the empirical comparison of three architectures, i.e., Residual U-Net, Dual-decoder U-Net, and the proposed ADU-Net. The "Residual U-Net" represents the structure shown in Figure 4a, while the "Dual-Decoder U-Net" also represents the structure depicted in Figure 4b. Table 9 verifies our design is reasonable, where the dual-decoder U-Net outperforms the vanilla version of the residual U-Net and our ADU-Net can further bring the performance gain to the dual-decoder U-Net.

Table 9. Effect of dual-branch architecture in rain and haze removal. We use **bold** to indicate best the result.

Model	PSNR	SSIM
Residual U-Net	31.64	0.9712
Dual-decoder U-Net	32.26	0.9724
ADU-Net	**33.83**	**0.9784**

The above study shows our design flow is reasonable. We further evaluate the effectiveness of the contamination residual branch and scene residual branch in ADU-Net (see the results in Table 10). As compared to the Residual U-Net, each branch can improve its performance, showing the effectiveness of the proposed residual branch. Also, we can observe that the combination of the proposed residual branches can achieve further improvement, indicating that those two decoders learn complementary features of the image. In Table 10, the first row, "Residual U-Net," is the same as in Table 9. The second row, "+Contamination residual branch," represents the model where the decoder of Residual U-Net is replaced with the Contamination residual branch, and similarly, the third row, "+Scene residual branch," represents the model where the decoder of Residual U-Net is replaced with the Scene residual branch. From the experimental results, it can be observed that each branch contributes to the improvement of PSNR and SSIM. However, combining both branches in the model leads to greater improvement.

Effect of Self-attention Module. As for Table 11, we aim to demonstrate that using W-MSA and SW-MSA in both symmetric decoders is superior to using either one alone. The first row (Dual-decoder U-Net) represents the

Table 10. Effect of the dual-branch decoder in ADU-Net. We use **bold** to indicate best the result.

Model	PSNR	SSIM
Residual U-Net	31.64	0.9712
+ Contamination residual branch	32.30	0.9725
+ Scene residual branch	32.94	0.9744
ADU-Net	**33.83**	**0.9784**

architecture shown in Figure 4b. The second row (+W-MSA) indicates that W-MSA is used in both symmetric decoders. Compared to the first row, there is an improvement of 0.44/0.0037 in PSNR/SSIM, indicating that using W-MSA alone brings only limited improvement. Similarly, the third row (+SW-MSA) indicates that SW-MSA is used in both symmetric decoders. Compared to the first row, there is an improvement of 0.51/0.0036 in PSNR/SSIM, showing that using SW-MSA alone also brings a modest improvement. The third row (+W-MSA&SW-MSA) represents the utilization of both W-MSA and SW-MSA in the two branches. There is an improvement of 0.74/0.0040 in PSNR/SSIM compared to the first row. However, it should be noted that in our experiments, we focused on saving the model with the best PSNR during training and did not specifically optimize for SSIM. Therefore, we believe that the marginal decrease in SSIM does not necessarily indicate a decline in model performance. We acknowledge that the benefits brought by simultaneously using both W-MSA and SW-MSA may not be significant.

Table 11. Effect of Self-attention Module. We use **bold** to indicate best the result.

Model	PSNR	SSIM
Dual-decoder U-Net	32.26	0.9724
+ W-MSA	32.70	0.9761
+ SW-MSA	32.77	0.9760
+ W-MSA&SW-MSA	**33.00**	**0.9764**

Effect of Feature Fusion Module. In the proposed architecture of the ADU-Net, each decoder block has two information flows, respectively encoding the contamination residual and scene residual (see Fig. 2 and Fig. 2). Each information flow yields the feature fusion w.r.t. the concern of physical properties. In this study, we evaluate our design. Table 12 ablations the effectiveness of the feature fusion blocks. Each of the CFF or GCFF can improve the accuracy by about 0.2 PSNR value. However, combining those two blocks can further bring an outstanding performance gain on top of the individual one, around 0.6 PSNR value. This can greatly verify the good practice of the feature fusion blocks in our design.

4.5 Application: Semantic Segmentation

To demonstrate the effectiveness of our approach for application, we conducted an evaluation of our approach using the RainCityscapes dataset, chosen for its comprehensive assessment of overall image restoration. The experimental results are presented in Table 13. As a baseline, we used DeepLabV3 [8] and performed semantic segmentation on the RainCityscapes dataset. The evaluation metrics included IoU_{class}(Intersection-over-Union for classes), $iIoU_{class}$ (instance-level Intersection-over-Union for classes), $IoU_{category}$(Intersection-over-Union for categories), $iIoU_{category}$(instance-level Intersection-over-Union for categories), and accuracy.

Table 12. Effect of Feature Fusion Module. We use **bold** to indicate best the result.

Model	PSNR	SSIM
Dual-decoder U-Net	32.26	0.9724
w/o GCFF&CFF	33.00	0.9764
+ CFF	33.25	0.9770
+ GCFF	33.21	0.9773
ADU-Net	**33.83**	**0.9784**

We conducted four types of experiments: original rainy images from the RainCityscapes dataset (Rainy Images), rainy images derained by ADU-Net (Rain-free Images (ADU-Net)), rainy images derained by ADU-Net-plus (Rain-free Images (ADU-Net-plus)), and ground truth images from the RainCityscapes dataset (Rain-free Images (ground truth)). As shown in Table 13, the utilization of ADU-Net for removing rain and haze from the images resulted in improvements across various evaluation metrics. The IoU_{class} metric showed a notable improvement of 0.1047, while the $iIoU_{class}$ increased by 0.0842. Furthermore, the $IoU_{category}$ experienced a significant boost of 0.1241, and the $iIoU_{category}$ demonstrated an even more substantial enhancement of 0.1896. Additionally, the accuracy metric showed a notable increase of 0.0754. Similar positive advancements were observed for ADU-Net-plus across all metrics.

It is noteworthy that the experimental results of the rainy images derained by ADU-Net (Rain-free Images (ADU-Net)) and ADU-Net-plus (Rain-free Images (ADU-Net-plus)) were comparable. Considering the minimal difference of only 0.81/0.0021 in terms of PSNR and SSIM, it indicates that there is a limited benefit in the application of semantic segmentation models when the image restoration reaches a certain level. Further improvements are necessary to achieve better results.

Table 13. Effect of ADU-Net on semantic segmentation. $1^{st}/2^{nd}$ best in red/blue.

Model	IoU_{class}	$iIoU_{\text{class}}$	IoU_{category}	$iIoU_{\text{category}}$	Accuracy
Rainy Images	0.3649	0.1116	0.6293	0.3045	0.7823
Rain-free Images(ADU-Net)	0.4696	0.1958	0.7534	0.4941	0.8577
Rain-free Images(ADU-Net-plus)	0.4724	0.1951	0.7561	0.5005	0.8589
Rain-free Images(ground truth)	0.4841	0.2039	0.7644	0.5265	0.8625

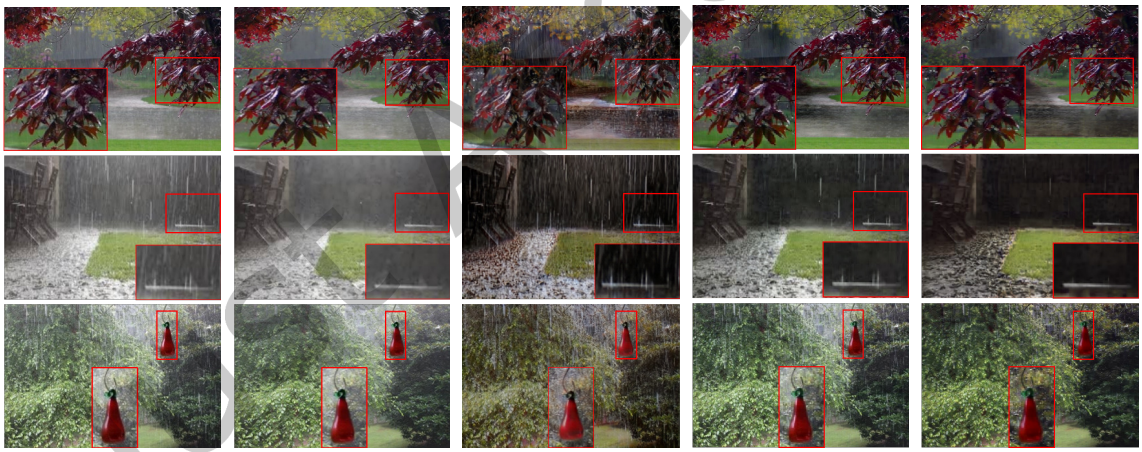
4.6 Visualization

Along with the quantitative analysis in the above paragraphs, we further conduct qualitative analysis to verify the superiority of our work. In this study, we first illustrate the rain and haze removal performance between our work and existing SOTA methods in synthetic datasets (see Fig. 5). Various real-world outdoor scenes are also evaluated (see Fig. 6). The generalization of the proposed ADU-Net is further evaluated by removing other contamination, e.g. only rain in Fig. 7, or rain and snow in Fig. 8.

The first study is evaluated on the RainCityscapes dataset. We compare our method with the state-of-the-art methods, including PReNet [40], AECR-Net [52] and DGNL-Net [21]. As shown in Fig. 5, our method can produce a much clear scene image (see the red box for details). For example, in the fourth row of Fig. 5, our method removes most of the haze and produces a clear shape of the tree branches. While other methods fail to recover the tree branches. This clearly shows the superiority of our method.

(a) Input (b) PReNet (c) AECR-Net (d) DGNL-Net (e) Ours (f) GroundTruth

Fig. 5. Visualization of contamination removal performance on the RainCityscapes. The first column (a) is the input image. We compare our method with state-of-the-art algorithms, including PReNet [40], AECR-Net [52] and DGNL-Net [21]. (f) is the ground truth.



(a) Input (b) PReNet (c) AECR-Net (d) DGNL-Net (e) Ours

Fig. 6. Visualization of contamination removal performance on real-world images with rain and haze. The first column (a) is the input image. We compare our method with state-of-the-art algorithms, including PReNet [40], AECR-Net [52] and DGNL-Net [21].

In the second study, we conduct the analysis on real-world images[1] used in [49], to justify the potential of our method in real scenarios. We again compare our method to PReNet, AECR-Net, and DGNL-Net. For a fair

[1]147 real rain images collected from Internet.

comparison, each method adopts publicly available fine-tune weights trained on their own datasets. As can be observed from Fig. 6, the scene images, generated by our method, are more clear and more realistic than those from other methods. For example, as compared to the rain removal network PReNet, our method can also remove the haze in real-world scenes. The hues of the recovered scene from our method are also more realistic than that from the dehazing network AECR-Net and reflective details of the scenes are maintained by our method. As compared to DGNL-Net, the closest work to ours, {our ADU-Net can remove more rain streaks (the second row) or haze (the third row) and retain more scene details (the first row). This study can vividly show the effectiveness of our method in real scenarios.

Fig. 7. Visualization of the contamination removal on the BID Rain dataset. The images in BID Rain are synthesized with rain streaks, raindrops, snow, and haze. The **first row** is the input image. The **second row** and **third row** are contamination residual and scene residual. The **fourth row** and **fifth row** are the clean image and ground truth.

To demonstrate the generalization of our dual-decoder architecture in separating different contamination, we show the residual produced by different branches. Fig. 7 shows the results of our method on the BID Rain dataset. The first row is the input image. The second row and third row present the masks of contamination residual and scene residual. The fourth row and fifth row are the generated images and the ground truth. We can find that our method separates the contamination (e.g., snow or haze) and scene clearly, and produces high-quality scene images. A similar observation is also made in the real-world images from Internet-Data in Fig. 8. This study also verifies our motivation that most of the contamination components in the image are included in the contamination residual while the scene residual contains more detail of the scene including building structures and driveway lines. This analysis again illustrates the superior generalization of the proposed method.

5 CONCLUSION

In this paper, we propose ADU-Net, the first module involving two residual branches, for the joint rain and haze removal task. Unlike previous work focusing on the contamination removal only, ADU-Net recalls the importance

Fig. 8. Visualization of the contamination removal on real-world rain images. The **first row** is the input image. The **second row** and **third row** are contamination residual and scene residual. The **fourth row** is the clean image.

of restoring the scene information affected by the change of atmospheric light. By leveraging our proposed scene residual and contamination residual, ADU-Net can produce clear scene images. The superiority of ADU-Net is evaluated by extensive experiments, and the proposed ADU-Net outperforms the current state-of-the-art approaches significantly across three benchmark datasets and tasks. We believe our study will serve as a strong baseline for future work, and inspire more research work in the line of joint rain and haze removal task.

ACKNOWLEDGMENTS

This work is supported by the Natural Science Foundation of Zhejiang Province under Grant LGG21F030011 and the National Natural Science Foundation of China under Grant 61972355. This work is also supported by the the National Natural Science Foundation of China under Grant 62306070 and by the Southeast University Start-Up Grant for New Faculty under Grant 4009002309. Furthermore, the work is also supported by the Big Data Computing Center of Southeast University.

REFERENCES

[1] Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375* (March 2018).

[2] Codruta O Ancuti, Cosmin Ancuti, and Radu Timofte. 2020. NH-HAZE: An image dehazing benchmark with non-homogeneous hazy and haze-free images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 444–445.

[3] Yunhao Ba, Howard Zhang, Ethan Yang, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Celso M. de Melo, Suya You, Stefano Soatto, Alex Wong, and Achuta Kadambi. 2022. Not Just Streaks: Towards Ground Truth For Single Image Deraining. In *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part VII* (Tel Aviv, Israel). Springer-Verlag, Berlin, Heidelberg, 723–740.

[4] James F. Blinn. 1982. A Generalization of Algebraic Surface Drawing. *ACM Transactions on Graphic* 1, 3 (July 1982), 235–256.

[5] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. 2016. DehazeNet: An End-to-End System for Single Image Haze Removal. *IEEE Transactions on Image Processing* 25, 11 (November 2016), 5187–5198.

[6] Chenghao Chen and Hao Li. 2021. Robust Representation Learning with Feedback for Single Image Deraining. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7738–7747.

[7] Long Chen, Wujing Zhan, Wei Tian, Yuhang He, and Qin Zou. 2019. Deep Integration: A Multi-Label Architecture for Road Scene Recognition. *IEEE Transactions on Image Processing* 28, 10 (October 2019), 4883–4898.

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587* (2017).

[9] Wei-Ting Chen, Jian-Jiun Ding, and Sy-Yen Kuo. 2019. PMS-Net: Robust Haze Removal Based on Patch Map for Single Images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11673–11681.

[10] Wei-Ting Chen, Zhi-Kai Huang, Cheng-Che Tsai, Hao-Hsiang Yang, Jian-Jiun Ding, and Sy-Yen Kuo. 2022. Learning Multiple Adverse Weather Removal via Two-stage Knowledge Learning and Multi-contrastive Regularization: Toward a Unified Model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 17632–17641.

[11] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. The Cityscapes Dataset for Semantic Urban Scene Understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3213–3223.

[12] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. 2020. Multi-Scale Boosted Dehazing Network With Dense Feature Fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2154–2164.

[13] Heng Fan, Liting Lin, Fan Yang, Peng Chu, Ge Deng, Sijia Yu, Hexin Bai, Yong Xu, Chunyuan Liao, and Haibin Ling. 2019. LaSOT: A High-Quality Benchmark for Large-Scale Single Object Tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5369–5378.

[14] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Huang, and Xinghao Ding. 2018. Residual-Guide Network for Single Image Deraining. In *Proceedings of the 26th ACM International Conference on Multimedia*. Association for Computing Machinery, 1751–1759.

[15] Raanan Fattal. 2014. Dehazing Using Color-Lines. *ACM Transactions on Graphics* 34, 1 (November 2014), 1–14.

[16] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. 2017. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1715–1723.

[17] Kshitiz Garg and Shree K. Nayar. 2007. Vision and Rain. *International Journal of Computer Vision* 75, 1 (October 2007), 3–27.

[18] Junlin Han, Weihao Li, Pengfei Fang, Chunyi Sun, Jie Hong, Mohammad Ali Armin, Lars Petersson, and Hongdong Li. 2022. Blind Image Decomposition. In *European Conference on Computer Vision*.

[19] Kaiming He, Jian Sun, and Xiaoou Tang. 2011. Single Image Haze Removal Using Dark Channel Prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 12 (December 2011), 2341–2353.

[20] Xiaowei Hu, Chi-Wing Fu, Lei Zhu, and Pheng-Ann Heng. 2019. Depth-Attentional Features for Single-Image Rain Removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8014–8023.

[21] Xiaowei Hu, Lei Zhu, Tianyu Wang, Chi-Wing Fu, and Pheng-Ann Heng. 2021. Single-Image Real-Time Rain Removal Based on Depth-Guided Non-Local Features. *IEEE Transactions on Image Processing* 30 (January 2021), 1759–1770.

[22] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of the 32nd International Conference on International Conference on Machine Learning* (Lille, France) *(ICML'15, Vol. 37)*. JMLR.org, 448–456.

[23] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. 2017. Image-to-Image Translation with Conditional Adversarial Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5967–5976.

[24] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. 2012. Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition. *IEEE Transactions on Image Processing* 21, 4 (December 2012), 1742–1755.

[25] Dong Hwan Kim, Woo Jin Ahn, Myo Taeg Lim, Tae Koo Kang, and Dong Won Kim. 2021. Frequency-Based Haze and Rain Removal Network (FHRR-Net) with Deep Convolutional Encoder-Decoder. *Applied Sciences* 11, 6 (March 2021).

[26] Ashutosh Kulkarni and Subrahmanyam Murala. 2022. Wipernet: A lightweight multi-weather restoration network for enhanced surveillance. *IEEE Transactions on Intelligent Transportation Systems* 23, 12 (2022), 24488–24498.

[27] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. 2017. AOD-Net: All-in-One Dehazing Network. In *Proceedings of the IEEE International Conference on Computer Vision*. 4780–4788.

[28] Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. [n. d.]. Non-locally enhanced encoder-decoder network for single image de-raining. In *Proceedings of the 26th ACM International Conference on Multimedia*. 1056–1064.

[29] Ruoteng Li, Robby T. Tan, and Loong-Fah Cheong. 2020. All in One Bad Weather Removal Using Architectural Search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3172–3182.

[30] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. 2018. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision*. 262–277.

[31] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. 2016. Rain streak removal using layer priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2736–2744.
[32] Chuping LIANG, Yidan FENG, Haoran XIE, Mingqiang WEI, and Xuefeng YAN. 2021. Prior-based single image rain and haze removal. *Journal of ZheJiang University (Science Edition)* 48, 3 (May 2021), 270–281.
[33] Xing Liu, Masanori Suganuma, Zhun Sun, and Takayuki Okatani. 2019. Dual residual networks leveraging the potential of paired operations for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7000–7009.
[34] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. 2018. DesnowNet: Context-Aware Deep Network for Snow Removal. *IEEE Transactions on Image Processing* 27, 6 (June 2018), 3064–3073.
[35] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. 2021. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *IEEE/CVF International Conference on Computer Vision*. 9992–10002.
[36] Yu Luo, Yong Xu, and Hui Ji. 2015. Removing Rain from a Single Image via Discriminative Sparse Coding. In *IEEE International Conference on Computer Vision*. 3397–3405.
[37] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., 8024–8035.
[38] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. 2020. FFA-Net: Feature fusion attention network for single image dehazing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 11908–11915.
[39] Yanyun Qu, Yizi Chen, Jingying Huang, and Yuan Xie. 2019. Enhanced Pix2pix Dehazing Network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 8152–8160.
[40] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. 2019. Progressive Image Deraining Networks: A Better and Simpler Baseline. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3932–3941.
[41] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. 2016. Single Image Dehazing via Multi-scale Convolutional Neural Networks. In *European Conference on Computer Vision*. Springer, 154–169.
[42] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. 2018. Gated Fusion Network for Single Image Dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3253–3261.
[43] Wenqi Ren, Jinshan Pan, Hua Zhang, Xiaochun Cao, and Ming-Hsuan Yang. 2020. Single Image Dehazing via Multi-scale Convolutional Neural Networks with Holistic Edges. *International Journal of Computer Vision* 128, 1 (January 2020), 240–259.
[44] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. 2018. Semantic Foggy Scene Understanding with Synthetic Data. *International Journal of Computer Vision* 126 (September 2018), 973–992.
[45] Yuda Song, Zhuqing He, Hui Qian, and Xin Du. 2023. Vision transformers for single image dehazing. *IEEE Transactions on Image Processing* 32 (2023), 1927–1941.
[46] Ziyi Sun, Yunfeng Zhang, Fangxun Bao, Ping Wang, Xunxiang Yao, and Caiming Zhang. 2022. Sadnet: Semi-supervised single image dehazing method based on an attention mechanism. *ACM Transactions on Multimedia Computing, Communications, and Applications* 18, 2 (2022), 1–23.
[47] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M. Patel. 2022. TransWeather: Transformer-Based Restoration of Images Degraded by Adverse Weather Conditions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2353–2363.
[48] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. 2020. A Model-Driven Deep Neural Network for Single Image Rain Removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 3103–3112.
[49] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. 2019. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 12262–12271.
[50] Ying Wang, Yuexing Peng, Wei Li, George C. Alexandropoulos, Junchuan Yu, Daqing Ge, and Weixu Xiang. 2022. DDU-Net: Dual-Decoder-U-Net for Road Extraction Using High-Resolution Remote Sensing Images. *IEEE Transactions on Geoscience and Remote Sensing* 60 (2022), 1–12.
[51] Zhou Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13, 4 (April 2004), 600–612.
[52] Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. 2021. Contrastive Learning for Compact Single Image Dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10546–10555.
[53] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. 2017. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1685–1694.
[54] Chia-Hung Yeh, Chih-Hsiang Huang, and Li-Wei Kang. 2020. Multi-Scale Deep Residual Learning-Based Single Image Haze Removal via Image Decomposition. *IEEE Transactions on Image Processing* 29 (2020), 3153–3167.
[55] Yi Yu, Wenhan Yang, Yap-Peng Tan, and Alex C. Kot. 2022. Towards Robust Rain Removal Against Adversarial Attacks: A Comprehensive Benchmark Analysis and Beyond. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6013–6022.

[56] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. 2021. Multi-Stage Progressive Image Restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 14816–14826.

[57] He Zhang and Vishal M. Patel. 2018. Densely Connected Pyramid Dehazing Network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3194–3203.

[58] He Zhang and Vishal M Patel. 2018. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 695–704.

[59] He Zhang, Vishwanath Sindagi, and Vishal M. Patel. 2020. Image De-Raining Using a Conditional Generative Adversarial Network. *IEEE Transactions on Circuits and Systems for Video Technology* 30, 11 (November 2020), 3943–3956.

[60] Hang Zhang, Han Zhang, Chenguang Wang, and Junyuan Xie. 2019. Co-Occurrent Features in Semantic Segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 548–557.

[61] Lei Zhu, Zijun Deng, Xiaowei Hu, Haoran Xie, Xuemiao Xu, Jing Qin, and Pheng-Ann Heng. 2021. Learning Gated Non-Local Residual for Single-Image Rain Streak Removal. *IEEE Transactions on Circuits and Systems for Video Technology* 31, 6 (June 2021), 2147–2159.

[62] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. 2017. Joint bi-layer optimization for single-image rain streak removal. In *Proceedings of the IEEE International Conference on Computer Vision*. 2545–2553.

[63] Qingsong Zhu, Jiaming Mai, and Ling Shao. 2015. A Fast Single Image Haze Removal Algorithm Using Color Attenuation Prior. *IEEE Transactions on Image Processing* 24, 11 (November 2015), 3522–3533.