QSAM-Net: Rain Streak Removal by Quaternion Neural Network With Self-Attention Module

Vladimir Frants, Sos Agaian, Fellow, IEEE, and Karen Panetta, Fellow, IEEE

Abstract—Real-world images captured in remote sensing, image or video retrieval, and outdoor surveillance are often degraded due to poor weather conditions, such as rain and mist. These conditions introduce artifacts that make visual analysis challenging and limit the performance of high-level computer vision methods. In time-critical applications, it is vital to develop algorithms that automatically remove rain without compromising the quality of the image contents. This article proposes a novel approach called QSAM-Net, a quaternion multi-stage multiscale neural network with a self-attention module. The algorithm requires significantly fewer parameters by a factor of 3.98 than the real-valued counterpart and state-of-the-art methods while improving the visual quality of the images. The extensive evaluation and benchmarking on synthetic and real-world rainy images demonstrate the effectiveness of QSAM-Net. This feature makes the network suitable for edge devices and applications requiring near real-time performance. Furthermore, the experiments show that the improved visual quality of images also leads to better object detection accuracy and training speed.

Index Terms—Deep learning, object detection, quaternion image processing, quaternion neural networks, rain removal.

I. INTRODUCTION

Adverse weather conditions (haze, snow, rain, fog) are among the most prevalent reasons for the limited use of automatic video surveillance, crowd counting, accident detection, person re-identification, computational photography, and other computer vision tasks [1], [2], [3], [17], [4], [5], [6], [7], [46]. A preprocessing step is necessary to limit the impact of weather conditions on the rest of the image analysis system. In the real-world applications mentioned above, image de-raining is highly desirable because of several factors that a) rainy weather often causes poor visibility, contrast reduction, and color modification, b) the removal of rain streaks typically produces over-smoothed images and leads to lost image details, c) the lack of robust prior-models of both rain-streaks and background and d) the image rain removal is a highly ill-posed problem [8]. Image de-raining aims to create a sharp, clean image from a rainy input image. Therefore, it is essential to develop algorithms that automatically remove these artifacts, without degrading the rest of the image’s contents. Current deraining efforts can be grouped into model-based and data-driven approaches.

Model-based de-raining algorithms make use of the prior information about rain distribution. They exploit images’ sparsity and local similarity properties of the rain by using sparse dictionaries and Gaussian Mixture Models (GMM) [9], [10], [11]. Data-driven de-raining methods use deep neural networks trained on end-to-end synthetic and clean/rainy image pairs. Various network architectures were explored, including classical CNNs, recurrent neural networks, and generative adversarial networks [11], [12], [13], [14], [15]. Recently, multiscale multi-stage methods such as MPRNet, HINet, MAXIM, and Restormer showed high performance on image processing tasks, including rain streak removal [12], [13], [16], [43], [51]. Despite the significant progress, both model-based and data-driven approaches have their drawbacks. Because of the inadequacy of existing models, some methods remove non-rain-related vertical textures and generate underexposed images [2], [7]. This problem is especially apparent with supervised learning. Modern methods use different network architectures, assumptions, and priors but fail when previously unseen conditions occur [6].

Real-world rainy images are diverse and have complex structures. Raindrops have a variety of sizes, types, densities, and orientations [2], [3], [8]. But the synthetic datasets lack diversity and are limited to the cases of rain of light and mild intensity—Wang et al. attempt to synthesize a real-world dataset using a simple video-based deraining technique [17]. GAN-based methods address the issue by better modeling the dataset distributions and generating images with better illumination, more accurate colors, and better contrast, but produce artifacts when faced with a distribution significantly different from the training distribution [4]. Semi-supervised and unsupervised methods model the rain distribution based on real-world data. This provides tools to generate high-quality synthetic datasets, but many methods cannot show the expected generalization ability, especially in the case of real-world images [5]. Advanced complex deep-learning architectures coupled with infrared imaging, multi-modal input data, and video data improve the situation but are expensive and have limited use in real-world scenarios [8], [9]. Below, the major limitations of the current state-of-the-art de-raining methods are summarized:
1) **Overfitting, over-smoothing, and unnatural hue change.** Overfitting occurs due to the limited generalization ability of models trained on large, diverse datasets, resulting in poor application performance [2, 6]. Over-smoothing manifests itself in textured background regions because of inefficient rain modeling and the similarity between rain streaks and background textures [3]. The presence of mist distorts the color information, which many methods cannot restore [4].

2) **Limited ability to model complex rain patterns, such as overlapping rain streaks and mist.** Streak accumulation over a distance leads to the mist/fog effects [5].

3) **Lack of high-quality realistic datasets:** Data-driven methods depend highly on the training datasets’ quality and size [6]. The current training datasets are limited in size and variation of the rain distortions, affecting the results’ quality.

4) **Lack of focus on consequent processing steps:** Even though CNNs perform well in many real-life applications, they need to improve in color image processing purposes/tasks. Addressing these challenges requires novel approaches to better handle complex rain patterns and address current datasets’ limitations while incorporating color image processing tasks.

The attention mechanism: This module has become increasingly relevant in recent years due to its ability to enhance the interpretability of deep learning models and improve their performance on complex tasks. As a result, the attention mechanism has gained significant attention in the research community and many recent breakthroughs in various domains, including speech recognition, natural language processing, and computer vision [18]. This module serves multiple purposes, such as enhancing a network’s expression ability by emphasizing relevant features and suppressing irrelevant information. It can effectively remove unnecessary color features and noise and adaptively enhance valuable information from input features. Moreover, the attention mechanism can improve model training performance and reduce computational complexity, making it an essential tool in designing efficient neural networks [18, 19].

The motivation for utilizing quaternion-valued neural networks stems from their proven superiority over real-valued neural networks in various applications such as image, speech, signal processing, image compression, objective image quality assessment, and object recognition [20], [21], [22], [23], [24], [25], [26], [27], [28]. However, they have yet to be widely used for low-level image processing tasks. Quaternion neural networks are a model where neuron computations rely on quaternion numbers, which contain one real and three separate imaginary components. This representation is particularly suitable for efficiently processing color (R, G, B) images, as it preserves the color relationships between the R, G, and B channels. This is a critical limitation of real-valued CNNs (RCNNs), which fail to capture color information when trained on grayscale images, making them impractical in heterogeneous conditions [20].

Furthermore, QNNs can achieve state-of-the-art performance while reducing the number of training parameters, which leads to a natural solution to the overfitting phenomenon [27], [28]. This makes them promising for real-time applications and implementation on edge devices. Moreover, Quaternion CNNs can better capture and model complex spatial relationships within the input data, making them particularly suited for processing challenging rain-streak images. Rain-streak images can be difficult to process because of the complex patterns and textures created by rain droplets. However, QCNNs can leverage their ability to model complex spatial relationships to effectively remove the rain streaks from the image while preserving its underlying structure and features.

This work aims to leverage the advantages of quaternion image processing and quaternion-valued neural networks in a low-level image preprocessing framework to enhance computer vision systems’ performance in adverse weather conditions, such as rain and fog. Our contributions are threefold:

1) We propose a novel QSAM-Net combining multiscale, multi-stage self-attention architecture to remove single-image rain streaks. This is the first time QNN has been utilized for rain removal.

2) We introduce a novel quaternion algebra-based image processing pipeline, QSAM-C-Net, which enhances image color, visibility, and perceived quality. It effectively addresses the distortions of poor weather conditions, including rain streaks, mist, fog, and low contrast.

3) Extensive quantitative and qualitative experiments demonstrate the superior performance of QSAM-Net compared to real-valued analog and existing state-of-the-art methods in two important ways. Firstly, the proposed approach achieves up to 2% improvement over the state-of-the-art in PSNR and SSIM, up to 3 points according to SSEQ [29], and up to 2 points according to BRISQUE [30]. Moreover, Section IV shows that this improvement is achieved with fewer parameters and faster training. The approach has been evaluated on a diversity of datasets, including Test100 [31], Test100L [32], Test100H [32], Test1200 [33], and Test2800 [34]. Secondly, our experiments demonstrate a 3% improvement in object detection accuracy in poor weather conditions on Rain in Driving (RID) and Rain in Surveillance (RIS) datasets [3].

The rest of the paper is organized as follows. Section II presents an overview of previous work on single-image rain-streak removal and quaternion neural networks for image processing. Section III describes the proposed QSAM-Net architecture. Section IV presents and discusses the experimental results. Section V concludes the paper.

### II. Previous Work

In this section, existing image de-raining methods are discussed, as well as recent progress in quaternion neural networks.

#### A. Single Image Deraining

To approach the rain streak removal task, a straightforward model of the rainy image \( I \) is represented as a sum of the rain-free scene \( J \), and a sparse streak image \( S \) [3]:

\[
I = J + S
\]  

(1)
Multi-Stage approaches progressively aim to recover a clean image by employing a lightweight subnetwork several times. MPRNet uses a three-stage framework with shared features [12]. Each stage has access to the features generated by the previous step and the input image. HINet investigates the impact of the normalization layer in MPRNet-like architecture on low-level image processing tasks reaching state-of-the-art performance on rain streak removal [13]. Multi-stage approaches deliver excellent performance but are parameter-heavy. It makes them a good starting point for developing a quaternion-algebra-based method.

B. Quaternion Neural Networks

Quaternion algebra is a valuable tool for multichannel signal processing because of its ability to maintain the relations among dimensions [22]. It prevents information loss and enhances the capabilities of signal processing techniques [23], [22]. Quaternion convolution neural networks (QCNN) use quaternion parameters and redefine the convolution operation using the Hamiltonian product. QCNNs are superior to real-valued CNNs in various image processing and computer vision tasks [28]. Using quaternion convolution reduces the number of trainable parameters by 75%, improving performance.

Sifkas et al. used a QCNN for the segmentation of Byzantine inscriptions. Chen et al. [23] show the superiority of QCNN in image classification and denoising problems. QCNN outperforms a similar real-valued network by 0.89 dB on image denoising and classification accuracy by 4%. Yin et al. [28] derived quaternion batch normalization and introduced a quaternion attention mechanism.

The resulting QCNN demonstrates superior results on double JPEG compression detection tasks. Parcollet et al. evaluated QCNN on CIFAR-10 and CIFAR-100 classification and the KITTI image segmentation dataset [29]. QCNN requires 50% more training time but uses four times fewer parameters and shows slight performance gains. In [27], Parcollet et al. show the ability of quaternion convolutional encoder-decoder (QCAE) to reconstruct a color image from a grayscale input image successfully. At the same time, a similar real-valued network (CAE) fails.

As demonstrated a significant amount of evidence to support the claim that using quaternion algebra generally improves performance, especially in the case of color image processing.

III. PROPOSED QSAM-NET APPROACH

Below we present a quaternion neural architecture QSAM-Net for rain streak removal. Also, we describe a quaternion image processing pipeline QSAM-C-Net to improve the visibility of fine details in the image.

A quaternion number \( \hat{q} \in \mathbb{H} \) extends the concept of complex numbers by introducing one real (a) and three imaginary \((b, c, d)\) components in the form \( \hat{q} = a + bi + cj + dk; \) where \( a, b, c, d \in \mathbb{R} \) and \((i, j, k)\) form the quaternion union basis, where \( i^2 = j^2 = k^2 = ijk = -1. \)
The color input image of the size $W$ by $H$ pixels is represented as a quaternion matrix $\hat{I} \in \mathbb{H}^{H \times W}$:

$$\hat{I} = L + R\hat{I} + G\hat{j} + B\hat{k}$$ (2)

Where $L, R, G, B \in \mathbb{R}^{H \times W}$ are matrices representing luminosity, red, green, and blue channels, respectively. Similarly, intermediate feature maps are represented as a group of quaternion-valued matrices. This image representation’s main advantage is preserving the interrelationship and structural information between the R, G, and B channels [21], [22].

An algebra on $\mathbb{H}$ defines operations among quaternion numbers, such as addition, conjunction, and modulus, similar to the algebra on complex numbers [22].

The Hamiltonian product defines the non-commutative multiplication of two quaternions $\hat{x} = a_1 + b_1\hat{i} + c_1\hat{j} + d_1\hat{k}$ and $\hat{y} = a_2 + b_2\hat{i} + c_2\hat{j} + d_2\hat{k}$ as:

$$\hat{x} \otimes \hat{y} = (a_1a_2 - b_1b_2 - c_1c_2 - d_1d_2) + (a_1b_2 + b_1a_2 + c_1d_2 - d_1c_2)\hat{i} + (a_1c_2 - b_1d_2 + c_1a_2 - d_1b_2)\hat{j} + (a_1d_2 + b_1c_2 - c_1b_2 - d_1a_2)\hat{k}$$ (3)

In Quaternion Neural Networks (QNN), the Hamilton product ($\otimes$) replaces the real-valued dot product as a transform between two quaternion-valued feature maps. It allows the maintenance and exploitation of relations within components of a quaternion feature map. In Quaternion CNN (QCNN), the convolution of a quaternion input $\hat{q} = q_0 + q_1\hat{i} + q_2\hat{j} + q_3\hat{k}$ and kernel $W = W_0 + W_1\hat{i} + W_2\hat{j} + W_3\hat{k}$ is defined as:

$$\hat{q}' = \hat{W} \otimes \hat{q}$$ (4)

Typically, the quaternion convolution is implemented as a grouped real-valued convolution. A $C$-channel quaternion feature map is represented as $4 \cdot C$-channel real-valued feature map. The first $C$ channels represent real components of quaternion feature maps, and the following three groups of $C$ channels each represent $i, j,$ and $k$-components. The components of weight $\hat{W}$ convolved with multiple quaternion inputs.

A. Quaternion Multiscale Multi-Stage Streak Removal Network

The architecture of the QSAM-Net is presented in Fig. 2. We use an encoder-decoder multiscale network with skip connections. Both the encoder and decoder are residual blocks with quaternion convolution and quaternion instance normalization layers. We apply two stages consequently. Each stage has access to the original rainy image $I$, and the ground truth image $J$, during the training. The stages are interconnected by Cross-Stage Feature Fusion (CSFF) [12] modified to handle quaternion feature maps and the newly proposed Quaternion Self-Attention Module (QSAM). The network aims to estimate the residual rain-streak image $S$ given rainy input $I$.

For the real-valued case, it is defined as:

$$h(x) = \begin{cases} x & \text{if } x > 0 \\ \lambda x & \text{otherwise} \end{cases}$$ (5)

Where $\lambda$ is the gain parameter. We define the quaternion version by splitting along the components of a quaternion as:

$$\text{LeakyReLU}(\hat{q}) = h(a) + h(b)\hat{i} + h(c)\hat{j} + h(d)\hat{k}$$ (6)

Where $\hat{q} = a + b\hat{i} + c\hat{j} + d\hat{k}$ in an input quaternion-valued feature map.

To prevent internal covariate shift, we use Instance Normalization [41], defined as:

$$y_{tijk} = \left( \frac{x_{tijk} - \mu_i}{\sqrt{\sigma^2_{t_i} + \epsilon}} \right) \gamma + \beta$$ (7)

Where $\gamma \in \mathbb{R}$ and $\beta \in \mathbb{H}$ are trainable parameters, $(j, k)$, $j = 0..H, k = 0..W$ - pixel coordinates, $i = 0..M$ - feature map index, $t = 0..B, B$ - number of images in the minibatch, $M$ - number of feature maps, $H$ and $W$ - are height and width of the feature map, and $\epsilon$ is a small value, added for stability. The mean and variance for the quaternion case are computed as defined in [37].

B. Quaternion Self-Attention Module (QSAM)

In this subsection, we present a novel quaternion self-attention module (QSAM). It serves to pass the features between the stages of the network. Using quaternion algebra in the attention module...
ensures better use of multichannel information. We illustrate the structure and functioning of the QSAM in the middle part of Fig. 2. Overall, the QSAM module allows the QCNN to identify and emphasize the most relevant information in the input feature map, which is critical for accurate rain-streak removal.

The aim of QSAM is to take the feature map of the previous stage and generate the residual image. Then, the residual is added to the input image to form the restored image used to create the guidance image $S$. The quaternion self-attention module contains three quaternion convolution operations. First, convolution conv1 enriches the input feature map by using the convolutional kernel of size $3 \times 3$. Then the input feature map is transformed to the extent of a single quaternion that is summed up with the input image. Next, this information is used to generate the guidance by the convolution conv3, conv3 after the non-linear operation is used to transform the feature map by the guidance map. Besides QSAM, we use cross-stage feature fusion adapted for the quaternion case [8].

C. Quaternion Image Visibility Enhancement With QSAM-C-NET

Most papers on single-image de-raining ignore the impact of rain removal on the consequent high-level tasks. Also, rain streaks always coincide with introducing mist, blur, and low-light conditions, leading to poor visibility of fine details in the image.

We describe the QSAM-C-Net visibility restoration network and a post-processing procedure to deal with these issues. We separately designed and trained the following blocks: (1) QSAM-Net for rain streak removal, (2) a quaternion neural network for mist removal, and (3) a low light image enhancement block. Each block uses quaternion image representation and quaternion algebra for processing. Following, we apply classical image processing methods (1) Image quality enhancement using a histogram adjustment method, (2) sharpening as described in [42], (3) color correction following the procedure from [44]. The pipeline of QSAM-C-Net is illustrated in Fig. 3.

Low-light enhancement using Retinex decomposition is essential for images taken at night and in poor weather conditions. It also compensates for the general tendency of dehazing and de-raining methods to produce too dark images [44]. The network processes data through decomposition, adjustment, and reconstruction. The input image is at first decomposed into reflectance and illumination. We only employ the decomposition step to extract the color information in our combined cascaded network. We replace all the convolutional layers with the quaternion convolution layer and replace the ReLU activation layer with its quaternion split version. We trained the network on the same LOL datasets containing 500 low/normal-light image pairs following the procedure from the original paper [44].

To carry out the dehazing operation, we use architecture like Gated Context Aggregation Network [45]. We adopted it to the quaternion case. It uses a straightforward and efficient
coder-decoder network with half-resolution intermediate features gathered using dilated convolution [46]. We use three encoders, three decoder full-resolution layers, and four half-resolution blocks in our modified network. We keep the network structure. The convolution operation is replaced with the quaternion version. ReLU activation function is replaced with its quaternion split variant. We use the quaternion variant of Instance Normalization, as described in section B. In addition, we keep the gate prediction convolution real-valued. The network is trained on the OTS synthetic dataset [39] for 12 epochs using Adam [34] optimizer, with a starting learning rate of 0.01 and decayed by 10 every five epochs. The batch size is set to 12. The hyperparameters are chosen experimentally.

IV. EXPERIMENTAL RESULTS

A. Implementation Details

We use $3 \times 3$ convolutional kernels throughout the network. The number of quaternion feature maps in the encoder is set to $16-32-64-128-256$. We use PyTorch [38] and Adam [34] with a minibatch of size 2 to train the network. The training patches are of size $256 \times 256$ pixels, with horizontal flip for data augmentation. Starting learning rate is set to $2 \times 10^{-4}$ and decreased down to $1 \times 10^{-7}$ with cosine annealing strategy [35]. We are interested in the impact of the use of QCNN on the streak removal capability, so we optimize the network end-to-end using a very basic mean squared error (MSE) loss function:

$$\mathcal{L} = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} [I(i,j) - J(i,j)]^2$$

where $H$ and $W$ are the height and width of the output image, $I$ is the rainy image, and $J$ is the rain-free ground truth image.

B. Single Image Rain and Haze Removal on Synthetic and Real-World Datasets

We trained our model using the Rain13k dataset [43], a synthetic dataset that includes 13712 pairs of rainy and rain-free images, combining various previously introduced datasets with diverse synthetic rain conditions. For testing, we used several datasets, including Test100 [31], which contains 100 synthetic images of rainy scenes with light to moderate rainfall conditions; Test100L [32], with similar images but low-intensity rainfall conditions; Test100H [32], featuring heavy rainfall conditions, Test1200 [33], containing 1200 images of rainy scenes, and Test2800 [34], with 2800 images of rainy scenes featuring a wide variety of rain types and orientations. These datasets offer diverse synthetic rain conditions, comprehensively evaluating algorithms to remove or restore rainy scenes. With varying rain intensity and sparsity levels, these datasets offer a range of conditions to evaluate algorithm performance. MSFPN, MPRNet, and HINet are trained on the Rain13k dataset. SS-IRR and VRG are trained using procedures from the original papers. SSIM and PSNR are computed on the Y channel in the YCbCr color space, the standard practice [2]. Metrics BRISQUE [30], SSEQ [29], and NIQE [57] model the human visual system’s perception but do not consider color perception and are computed on the greyscale version of the image only. The metrics mainly consider the naturalness and structural features of the images. Tables I and II present the evaluation results. Table I shows that our deraining methods improve image quality under the SSEQ metric. We use blind image quality metrics and an internet-images dataset [50] (Figs. 4 and 6).

We have tested QSAM-Net on datasets Outdoor-Rain (a combination of haze and rain streaks). We compare with baseline methods for dehazing: EPDN [53], RefineDNet [54]; multi-degradation rain-removal: MPRNET [11], MAXIM [51] and universal: TransWeather [55], Chen et al. [56].

One can see that QSAM-Net is effective for a combination of haze and rain streaks, other methods struggle with this combination. Although Chen et al. and Transweather are suitable for various types of weather, they cannot process the combination of different weather types. The quantitative results of this study are presented in Table II. The table shows the quantitative comparison of our method with state-of-the-art methods on the test1 (rain+fog) dataset based on PSNR and SSIM. For the comparison we take results from QSAM-Net before the post-processing procedures, as they change image lowering PSNR and SSIM.

Table III shows that the quaternion network outperforms state of the art methods in terms of PSNR/SSIM and processing time using a smaller parameter budget and a very limited amount of training. Our quaternion network has the same or better capability to remove the streaks from real-world images. Fig. 5 shows that our method successfully removes streaks while SS-IRR and VRG fail to remove them completely. Using the proposed cascaded network and post-processing pipeline improves the visibility significantly.
Fig. 4. Rainy images from Internet Images dataset processed with different architectures including SS-IRR, MSPF, VRG, MPRNet, HINet, QSAM-Net, and QSAM-C-Net. QSAM-C-Net is qualitatively better at improving visibility than other methods.

**TABLE III**

|                  | Test100 [47] | Rain100H [48] | Rain100L [48] | Test1200 [49] | Test2800 [50] | # parameters | Second/Image |
|------------------|--------------|---------------|---------------|---------------|---------------|--------------|--------------|
|                  | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR | SSIM | PSNR |               |
| Input            | 0.686 | 22.542 | 0.378 | 13.551 | 0.838 | 26.900 | 0.732 | 23.634 | 0.783 | 23.349 | . | . |               |
| SS-IRR           | 0.787 | 22.261 | 0.490 | 16.458 | 0.842 | 24.892 | 0.824 | 26.097 | 0.867 | 28.072 | 23.2 M | 1.2 |               |
| MSPF             | 0.878 | 27.643 | 0.850 | 28.226 | 0.926 | 32.130 | 0.915 | 32.029 | 0.919 | 32.023 | 15.8 M | 1.9 |               |
| VRG              | 0.822 | 23.273 | 0.886 | 30.063 | 0.966 | 34.661 | 0.871 | 28.735 | 0.874 | 28.535 | 0.2 M | 0.8 |               |
| MPRNet           | 0.897 | 30.274 | 0.899 | 30.411 | 0.965 | 36.401 | 0.916 | 32.912 | 0.938 | 33.588 | 3.6 M | 2.5 |               |
| HINet            | 0.906 | 30.270 | 0.893 | 30.633 | 0.969 | 37.205 | 0.918 | 33.016 | 0.940 | 33.834 | 88.6 M | 2.7 |               |
| Restormer        | 0.923 | 32.000 | 0.904 | 31.461 | 0.978 | 38.992 | 0.922 | 33.038 | 0.938 | 33.641 | 27.1 M | 3.1 |               |
| Maxim-2S         | 0.922 | 31.170 | 0.903 | 30.812 | 0.977 | 38.065 | 0.922 | 32.569 | 0.940 | 33.876 | 14.1 M | 5.6 |               |
| QSAM-Net         | **0.926** | **32.208** | **0.906** | **30.885** | **0.979** | **37.337** | **0.924** | **33.819** | **0.948** | **34.112** | **12.2 M** | **0.6** |               |

Fig. 5. Dehazing results from various techniques. QSAM-C-Net shows the best visual quality on haze and rain-streaks.

**C. Real-World Application (Object Detection)**

Rain in Driving (RID) contains 2495 rainy images taken from real-world driving videos. The dataset includes all three types of rain effects: raindrops on the windshield, mist, and rain streaks. The images are taken in various diverse locations. The dataset is labeled with bounding boxes for 5 types of objects: car, person, bus, bicycle, and motorcycle. Rain in Surveillance (RIS) is like RID and contains 2048 rainy images taken from surveillance videos. The images mostly contain rain streams and mist. Both datasets are converted to the COCO format, and the resolution...
of images is changed to 640 on the longer side where necessary [50].

We conduct experiments on a single Nvidia 1080Ti GPU. For evaluation, we use a state-of-the-art network, SCNet trained on COCO dataset [51]. The model uses a multiscale approach to deliver high accuracy in object detection and segmentation tasks. We count a detection denoted by a bounding box with a confidence score equal to or higher than 0.3. We do not consider bounding boxes with the predicted classes that are not available in RID/RDS. The detection results are evaluated with a metric proposed for the COCO challenge mAP50 [50]. The examples of detected objects for threshold 0.3 are given in Fig. 7.

We compared the direct use of pre-trained SCNet on the unpreprocessed rainy image and de-raining method followed by SCNet, our network with SCNet, and our pipeline with SC-Net. Results for average precision mAP50 are presented in Table IV. Fig. 6 shows examples of detection by the different methods. It’s interesting to see that preprocessing the rainy images with a de-raining method before applying object detection using pre-trained SCNet actually decreases the accuracy of many methods, as reported in Table IV. This is consistent with previous findings that preprocessing can sometimes negatively impact the performance of object detection algorithms on rainy images [3]. The reason for this decrease in accuracy may be since de-raining methods can sometimes alter the visual appearance of the rain streaks in ways that are not easily predictable. This can lead to inconsistencies in the appearance of the rain streaks across different images, which can make it difficult for object detection algorithms to learn consistent features that are useful for detecting objects.

D. Ablation Studies

We perform an ablation study to establish the impact of quaternion color representation and the QSAM module. We compared the performance of three different network architectures: (1) a network using real-valued convolution, (2) a network using quaternion convolution without the QSAM, and (3) a network using quaternion convolution with the QSAM module. Results are presented in Table V. The results of this study suggest that using quaternion color representation and quaternion convolution instead of real-valued convolution can improve the performance of a deraining network. This is because quaternion convolution can capture more complex relationships between features than real-valued convolution, leading to more effective feature extraction and representation. Use of QSAM also improves deraining performance. Table VI presents the detection performance as an objective metric of the success of the proposed quaternion pipeline.

The study shows that using the whole pipeline, including both de-raining and dehazing, can further improve the detector’s performance. This is because rain and haze can affect an image’s clarity and visibility, and removing both can lead to better feature extraction and representation for the detector. The quaternion...
network trains faster and outperforms the real-valued network at each step during the training, as shown in Fig. 8.

Overall, the use of quaternion color representation and quaternion convolution, as well as the whole pipeline of de-raining and dehazing, can lead to improved performance in image processing tasks, such as object detection, where image clarity and quality are critical factors.

V. CONCLUSION

This paper presents two novel contributions, the QSAM-Net and the quaternion image processing pipeline QSAM-C-Net, which utilize quaternion image processing to improve the performance of rain removal. The proposed QSAM module improves performance by leveraging the Hamiltonian product and considering the properties of quaternion image representation. This results in up to a 3% increase in SSIM and PSNR in various rain conditions compared to state-of-the-art methods. Additionally, the proposed QSAM-C-Net improves the perceived quality of images taken in adverse weather conditions, measured by SSEQ and BRISQUE, by up to 3 and 2 points, respectively. Extensive ablation studies highlight the significance and impact of each pipeline stage. Furthermore, this proposed neural network demonstrates the benefits of quaternion color representation, quaternion neural networks, and quaternion image processing, addressing limitations of current methods, preserving original image details, improving object detection performance, and emphasizing the importance of rain-streak and mist for automatic image analysis and object detection applications. In the near future, we plan to extend this method to video-based rain removal using an automatic single-image self-learning-based rain streak removal concept.

REFERENCES

[1] Y. Yang et al., “End-to-end rain removal network based on progressive residual detail supplement,” IEEE Trans. Multimedia, vol. 24, pp. 1622–1636, 2022.
[2] W. Yang, R. T. Tan, S. Wang, Y. Fang, and J. Liu, “Single image deraining: From model-based to data-driven and beyond,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 11, pp. 4059–4077, Nov. 2021.
[3] S. Li et al., “Single image deraining: A comprehensive benchmark analysis,” in Proc. IEEE/CVF Comput. Vis. Pattern Recognit., 2019, pp. 3838–3847.
[4] S. Du et al., “A comprehensive survey: Image deraining and stereo-matching task-driven performance analysis,” IET Image Process., vol. 16, no. 1, pp. 11–28, 2022.
[5] X. Yang, H. Li, Y.-L. Fan, and R. Chen, “Single image haze removal via region detection network,” IEEE Trans. Multimedia, vol. 21, no. 10, pp. 2555–2560, Oct. 2019.
[6] Y. Que, S. Li, and H. J. Lee, “Attentive composite residual network for robust rain removal from single images,” IEEE Trans. Multimedia, vol. 23, pp. 3059–3072, 2021.
[7] M. R. Islam and M. Paul, “Video deraining using the visual properties of rain streaks,” IEEE Access, vol. 10, pp. 202–212, 2022.
[8] X. Kang, C.-W. Lin, and Y.-H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” IEEE Trans. Image Process., vol. 21, no. 4, pp. 1742–1755, Apr. 2012.
[9] H. Duan et al., “Develop then rival: A human vision-inspired framework for superimposed image decomposition,” IEEE Trans. Multimedia, early access, doi: 10.1109/TMM.2022.3172882.
[10] 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., 2016, pp. 2736–2744.
[11] A. Mehri, P. B. Ardakani, and A. D. Sappa, “MPRNNet: Multi-path residual network for lightweight image super-resolution,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis., 2021, pp. 2703–2712.
[12] L. Chen, X. Lu, J. Zhang, X. Chu, and C. Chen, “HINet: Half instance normalization network for image restoration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 182–192.
[13] S. A. Hovhanisyan, H. A. Gasparyan, S. S. Agaian, and A. Ghazaryan, “AED-Net: A single image dehazing,” IEEE Access, vol. 10, pp. 12465–12474, 2022.
[14] 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.
[15] S. W. Zamir et al., “Restormer: Efficient transformer for high-resolution image restoration,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 5718–5729.
[16] T. Wang et al., “Spatial attentive single-image deraining with a high quality real rain dataset,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 12262–12271.
[17] C. HengShuai et al., “Attention-guided multi-scale feature fusion network for low-light image enhancement,” Front. Neurorobot., vol. 16, 2022, Art. no. 837208.
[18] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7132–7141.
[19] T. Parcollet, M. Morchid, and G. Linares, “A survey of quaternion neural networks,” Artif. Intell. Rev., vol. 53, no. 4, pp. 2957–2982, 2020.
[20] A. Grigoryan and S. Agaian, “Quaternion and Octonion Color Image Processing with MATLAB,” Bellingham, WA, USA: SPIE, 2018.
[21] A. Grigoryan and S. Agaian, “Optimal restoration of multiple signals in quaternion algebra,” Proc. SPIE, vol. 11399, 2020, Art. no. 1139909.
[22] Y. Chen, X. Xiao, and Y. Zhou, “Low-rank quaternion approximation for color image processing,” IEEE Trans. Image Process., vol. 29, pp. 1426–1439, 2020.
[23] A. Grigoryan and S. Agaian, “Retooling of color imaging in the quaternion algebra,” Appl. Math. Sci.: An Int. J., vol. 1, no. 3, pp. 23–39, 2014.
IEEE TRANSACTIONS ON MULTIMEDIA, VOL. 26, 2024

[24] L. Luo, H. Feng, and L. Ding, “Color image compression based on quaternion neural network principal component analysis,” in Proc. Int. Conf. Multimedia Technol., 2010, pp. 1–4.

[25] A. Kolanam and O. Yadid-Pecht, “Quaternion structural similarity: A new quality index for color images,” IEEE Trans. Image Process., vol. 21, no. 4, pp. 1526–1536, Apr. 2012.

[26] T. Parcollet, M. Morchid, and G. Linarès, “Quaternion convolutional neural networks for heterogeneous image processing,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process., 2019, pp. 8514–8518.

[27] Q. Yin et al., “Quaternion convolutional neural network for color image classification and forensics,” IEEE Access, vol. 7, pp. 20293–20301, 2019.

[28] L. Liu et al., “No-reference image quality assessment based on spatial and spectral entropies,” Signal Process.: Image Commun., vol. 29, no. 8, pp. 856–863, 2014.

[29] A. Mittal, A. K. Moorthy, and A. C. Bovik, “No-reference image quality assessment in the spatial domain,” IEEE Trans. Image Process., vol. 21, no. 12, pp. 4605–4708, Dec. 2012.

[30] H. Zhang, V. Sindagi, and V. M. Patel, “Image de-raining using a conditional generative adversarial network,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 11, pp. 3943–3956, Nov. 2020.

[31] W. Yang et al., “Deep joint rain detection and removal from a single image,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1685–1694.

[32] H. Zhang and V. M. Patel, “Density-aware single image de-raining using a multi-stream dense network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 695–704.

[33] X. Fu et al., “Removing rain from single images via a deep detail network,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 1715–1723.

[34] X. Fu, Q. Qi, Z. Zha, Y. Zhu, and X. Ding, “Rain streak removal via dual graph convolutional network,” in Proc. AAAI Conf. Artif. Intell., 2021, pp. 1–9.

[35] K. Jiang et al., “Multiscale progressive fusion network for single image deraining,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 8343–8352.

[36] A. Greenblatt and S. Agaian, “Introducing quaternion multi-valued neural networks with numerical examples,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 326–342, 2018.

[37] P. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. 33rd Int. Conf. Neural Inf. Process. Syst., 2019, pp. 8026–8037.

[38] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

[39] I. Loshchilov and F. Hutter, “SGDR: Stochastic gradient descent with warm restarts,” 2016, arXiv:1608.03983.

[40] D. Ulyanov et al., “Instance normalization: The missing ingredient for fast stylization,” 2016, arXiv:1607.08022.

[41] J. G. Schavemaker, M. J. Reinders, J. J. Gerbrands, and E. Backer, “Image sharpening by morphological filtering,” Pattern Recognit., vol. 33, no. 6, pp. 997–1012, 2000.

[42] D. Tao, W. Wang, and W. Yang, “Deep retinex decomposition for low-light enhancement,” in Proc. Brit. Mach. Vis. Conf., 2018, pp. 1–12.

[43] S. Badireva et al., “Comprehensive analysis and benchmarking of de-raining techniques through a rain model approach,” Proc. SPIE, vol. 12100, pp. 135–146, 2022.

[44] D. Chen et al., “Gated context aggregation network for image dehazing and deraining,” in Proc. IEEE Winter Conf. Appl. Comput. Vis., 2019, pp. 1375–1383.

[45] Y. Wei et al., “Revisiting dilated convolution: A simple approach for weakly- and semi-supervised segmentation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018, pp. 7268–7277.

[46] V. Frants, S. Agaian, and K. Panetta, “QCNN-H: Single-image dehazing using quaternion neural networks,” IEEE Trans. Cybern., early access, 2011, doi:10.1109/TCYB.2013.2338640.

[47] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, “Semi-supervised transfer learning for image rain removal,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 3872–3881.

[48] H. Wang et al., “From rain generation to rain removal,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2021, pp. 14786–14796.

[49] T. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis., 2014, pp. 740–755.

[50] T. Vu, H. Kang, and C. D. Yoo, “SCNet: Training inference sample consistency for instance segmentation,” in Proc. AAAI Conf. Artif. Intell., 2021, vol. 35, no. 3, pp. 2701–2709.

[51] Z. Tu et al., “Maxim: Multi-axis MLP for image processing,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 5759–5770.

[52] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng, “AO-Dnet: All-in-one dehazing network,” in Proc. IEEE Int. Conf. Comput. Vis., 2017, pp. 4780–4788.

[53] Y. Qu, Y. Chen, J. Huang, and Y. Xie, “Enhanced pix2pix dehazing network,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 8152–8160.

[54] S. Zhao, L. Zhang, Y. Shen, and Y. Zhou, “RefineDNet: A weakly supervised refinement for single image dehazing,” IEEE Trans. Image Process., vol. 30, pp. 3391–3404, 2021.

[55] J. M. Jose Valanarasu, R. Yasalar, and V. M. Patel, “Transweather: Transformer-based restoration of images degraded by adverse weather conditions,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 2343–2353.

[56] W.-T. Chen et al., “Learning multiple adverse weather removal via two-stage knowledge learning and multi-contrastive regularization: Toward a unified model,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2022, pp. 17632–17641.

[57] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a completely blind image quality analyzer,” IEEE Signal Process. Lett., vol. 22, no. 3, pp. 209–212, Mar. 2013.

Vladimir Frants received the bachelor’s degree in electrical engineering from the South Russian University of Economics and Service, St. Petersburg, Russia, in 2011, and the M.S. degree in electrical engineering from the Don State Technical University, Rostov-on-Don, Russia, in 2013. He is currently working the Ph.D. degree in computer science with The Graduate Center CUNY, New York, NY, USA. His research interests include artificial intelligence, computer vision, image processing, and machine learning.

Sos Agaian (Fellow, IEEE) is currently a Distin-

Karen Panetta (Fellow, IEEE) received the B.S. degree in computer engineering from Boston University, Boston, MA, USA, and the M.S. and Ph.D. degrees in electrical engineering from Northeastern University, Boston, MA, USA. She is currently the Dean with Graduate Engineering Education and a Professor with the Department of Electrical and Computer Engineering. She is also an Adjunct Professor of computer science with Tufts University, Medford, MA, USA, and the Director for the Dr. Panetta’s Vision and Sensing System Laboratory. Her research interests include system identification, control, and biomedical imaging. The technologies he invented have been adopted by multiple institutions, including the U.S. Government, and commercialized by industry. He is an Associate Editor for several journals, including IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS, Journal of Electrical and Computer Engineering, and the International Journal of Digital Multimedia Broadcasting (both: Hindawi Publishing Corporation), and Journal of Electronic Imaging (IS&T and SPIE). He is a distinguished IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS Lecturer. He is also the Fellow of SPIE, IS&T, AIAA, and AAAS.