IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 22 November 2023, accepted 6 December 2023, date of publication 8 December 2023,
date of current version 15 December 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3341041*

RESEARCH ARTICLE

# Learned Kernel-Based Interpolation for Efficient RGBW Remosaicing

**AN GIA VIEN, (Graduate Student Member, IEEE), AND CHUL LEE, (Member, IEEE)**
Department of Multimedia Engineering, Dongguk University, Seoul 04620, South Korea

Corresponding author: Chul Lee (chullee@dongguk.edu)

This work was supported in part by the National Research Foundation of Korea (NRF) funded by the Korean Government (MSIT) under Grant NRF-2022R1F1A1074402, and in part by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under the Artificial Intelligence Convergence Innovation Human Resources Development funded by MSIT under Grant IITP-2023-RS-2023-00254592.

**ABSTRACT** RGBW remosaicing is an interpolation technique that converts RGBW images captured using RGBW color filtering arrays into Bayer images. Although recent learning-based approaches using convolutional neural networks have shown substantial performance improvements, most algorithms require high computational and memory complexities, which limit their practical applicability. In this work, we propose an efficient and effective RGBW remosaicing algorithm based on learned kernel-based interpolation. First, the proposed algorithm extracts deep feature maps from input RGBW images. Then, we develop a learned kernel-based interpolation module composed of local and non-local interpolation blocks that generates two intermediate Bayer images. Specifically, the local interpolation block learns local filters to recover a Bayer image, whereas the non-local interpolation block recovers a Bayer image by estimating the non-local filters of dynamic shapes. Finally, a reconstructed Bayer image is obtained by combining the complementary information from the intermediate Bayer images using a spatially weighted fusion block. Experimental results demonstrate that the proposed algorithm achieves comparable or even better performance than state-of-the-art algorithms while providing the lowest computational and memory complexities.

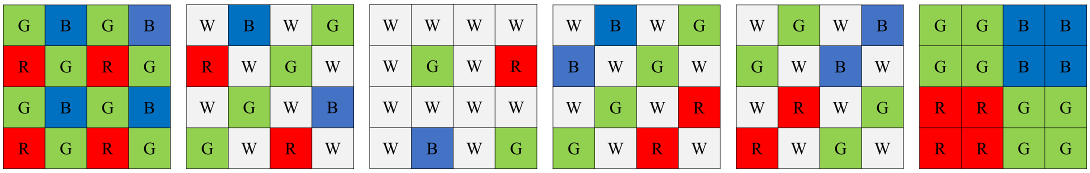
**INDEX TERMS** RGBW color filter array (CFA), Bayer CFA, remosaicing, learned kernel-based interpolation.

## I. INTRODUCTION

Despite recent advances in digital imaging systems, capturing high-quality images in low-light environments using conventional cameras remains challenging because of the limited sensitivity of RGB filters in the Bayer color filter array (CFA) [1], [2]. To overcome this limitation, several new CFA patterns with additional panchromatic, or white (W), pixels that have a higher transmittance of visible light by increasing the amount of light reaching each pixel, also known as RGBW CFA, have been developed [1], [3], [4], [5], [6], [7]. Figure 1 shows examples of common RGBW CFA patterns used in conventional cameras. As W pixels exhibit higher light efficiency than RGB color pixels, the signal-to-noise ratio of images captured by RGBW CFAs significantly increases, thereby improving the image quality, especially in low-light environments. Despite the advantages of RGBW CFAs over the Bayer CFA in light efficiency, image signal processing (ISP) pipelines, especially for demosaicing, have been developed and optimized for the Bayer CFA. Therefore, to increase the practical use of RGBW CFAs in cameras, various demosaicing algorithms have been developed to reconstruct full-color images from RGBW images [8].

A straightforward approach is to develop a demosaicing algorithm for RGBW CFAs [2], [8], [11], [12], [13], [14], [15]. However, demosaicing RGBW images, also known as RGBW demosaicing, is more challenging than Bayer demosaicing. Because W pixels have no color information

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



(a) (b) (c) (d) (e) (f)

**FIGURE 1.** **Examples of common CFA patterns. (a) Bayer CFA [9], (b) Sony-RGBW CFA [6], (c) Cannon-RGBW CFA [5], (d) Kodak-RGBW CFA [4], (e) shifted Kodak-RGBW CFA [10], and (f) Quad-Bayer CFA.[1]**

and the use of W pixels reduces the spatial resolution of the RGB information, as shown in Figures 1(b)–(e), the reconstruction of color information at each pixel location is more challenging. Nevertheless, several RGBW demosaicing algorithms have been developed because of their practical importance. For example, in [2], [8], [11], [13], [14], and [15], color reconstruction models for RGBW demosaicing were developed based on inter-channel correlations and image priors of RGBW CFAs. Despite their effectiveness in color reconstruction, they require high computational complexity to solve optimization problems. Another approach to RGBW demosaicing was developed based on pansharpening, which recasts color reconstruction as a fusion of W and RGB images [12], [16], [17]. However, this approach relies heavily on pansharpening algorithms, which are intractable for deployment in ISP pipelines.

Another effective approach is to convert RGBW images into Bayer images and then use the existing ISP pipeline to generate full-color images, which is known as RGBW sensor remosaicing or simply remosaicing. Due to their advantages of reusing existing ISP pipelines, various remosaicing algorithms [18], [19], [20], [21] have been developed based on the properties of specific CFAs. Early attempts at remosaicing have tried to formulate optimization problems based on prior knowledge of natural images and solve them. However, solving numerical optimization generally demands considerable computational resources, limiting their use in practical applications. Recently, a high-quality dataset of aligned RGBW and Bayer images was constructed for RGBW sensor remosaic challenges [10], [22], which can facilitate the development of deep learning-based algorithms by learning end-to-end mappings between RGBW and Bayer images. However, although learning-based algorithms [10], [22], [23], [24], [25] yield high remosaicing performance, they require high computational and memory complexities because of the large number of network parameters, as shown in Figure 2. This limits their applicability in commercial cameras, such as smartphones. Therefore, developing an efficient remosaicing algorithm while maintaining performance is highly desirable.

In this work, we develop a simple yet effective RGBW remosaicing algorithm based on learned kernel-based interpolation. The proposed algorithm consists of a feature extractor, learned kernel-based interpolation, and spatially weighted fusion modules. First, the feature extractor extracts the deep feature maps of an input RGBW image for interpolation. Next, the learned kernel-based interpolation module generates two Bayer images using local and non-local interpolation blocks. Specifically, the local interpolation block recovers a Bayer image by learning local filters, whereas the non-local interpolation block estimates the dynamic shapes of the filters to recover a Bayer image. Finally, the spatially weighted fusion module adaptively merges these intermediate images from the interpolation modules to generate a reconstructed Bayer image. Experimental results demonstrate that the proposed algorithm achieves comparable or even better performance while incurring the lowest computational and memory complexities compared with state-of-the-art learning-based algorithms [10], [22]. Figure 2 shows that the proposed algorithm achieves the best performance in terms of M4 with the least number of parameters and the fastest runtime.

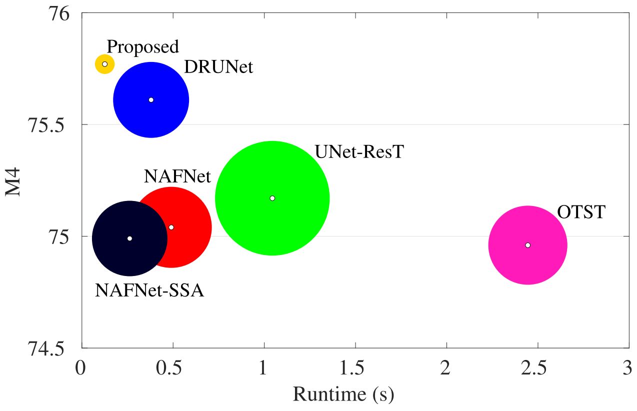


**FIGURE 2.** **Comparison of performance and runtime of the proposed algorithm and recent deep learning-based RGBW remosaicing algorithms on the MIPI-RGBW dataset [22]. The size of the circles depicts the numbers of network parameters.**

In summary, we make the following contributions:

- We propose a simple yet effective kernel learning-based RGBW remosaicing algorithm that converts RGBW images into Bayer images by adaptively combining local and non-local information extracted from an input RGBW image.
- We develop a learned kernel-based interpolation algorithm that learns dynamic local kernels and adaptive offsets to effectively recover pixels in a Bayer image by

---

[1] https://www.sony.com/en/SonyInfo/News/Press/201807/18-060E

extracting valid information from an RGBW image both locally and non-locally.
- We experimentally demonstrate that the proposed algorithm achieves state-of-the-art remosaicing performance while significantly reducing the model size and runtime, thus achieving a graceful tradeoff between performance and complexity.

The remainder of this paper is organized as follows. Section II reviews the related work. Section III describes the proposed RGBW remosaicing algorithm, and Section IV discusses the experimental results. Finally, Section V concludes the paper.

## II. RELATED WORK

### A. RGBW DEMOSAICING

RGBW demosaicing algorithms have been developed based on Bayer demosaicing and can be broadly divided into two categories according to how the properties of RGBW CFAs are exploited. Algorithms in the first category attempt to derive mathematical models by incorporating handcrafted model information based on the properties of RGBW CFAs. For example, the linear relationship between W and RGB pixels [8], [11], frequency structures of CFAs [26], image colorization [13], sparse representation [14], and low-dimensional manifolds [15] have been employed. However, these algorithms require substantial computational resources to solve optimization problems, making them unsuitable to deploy on resource-constrained mobile devices. Algorithms in the second category generate full-color images based on pansharpening [12], [16], [17]. Specifically, they first decomposed an RGBW image into a full-resolution W image and a low-resolution RGB image and then interpolated the missing W pixels and color pixels in each image. The interpolated full-resolution W and low-resolution RGB images are then combined using a pansharpening algorithm to obtain the final full-color image. However, the efficiency and performance of these algorithms depend on the choice of Bayer demosaicing, interpolation, and pansharpening algorithms, making their use on the ISP pipeline in practical applications challenging.

Recently, several deep learning-based RGBW demosaicing algorithms that learn end-to-end mappings from RGBW images to full-color images have been developed. For example, Chakrabarti [27] developed a two-path neural network that jointly learns the RGBW CFA pattern and color reconstruction. Sharif and Jung [28] proposed two sequential networks based on pansharpening to reconstruct full-color images of sparse RGBW CFAs. However, the construction of real image pairs of RGBW CFAs and their corresponding full-color images is more difficult than that of the Bayer CFA [8], which limits the generalization ability of learning-based algorithms in real-world applications.

### B. RGBW REMOSAICING

An alternative approach is RGBW remosaicing, which converts RGBW images into Bayer images and then uses existing ISP pipelines. Owing to this advantage, various remosaicing algorithms have been developed. Early attempts were made to exploit the prior knowledge of natural color images [18] or employ spatial filters [19], [20], [21]. Recently, with the construction of a dataset of real image pairs for RGBW and RGB CFAs [10], several deep learning-based RGBW remosaicing algorithms have been developed [10], [22], [25]. In addition, because RGBW remosaicing can be regarded as image restoration in that it reconstructs a Bayer image by recovering missing color pixels, state-of-the-art networks [23], [24], modules or blocks [29], [30], [31], [32], and transformer layers [33], [34] for image restoration have been employed for RGBW remosaicing. For example, Fan et al. [25] employed transformer layers to exploit long-range dependencies in input images by computing the similarity between pixels. However, existing learning-based algorithms focus only on performance improvement without considering efficiency, which is crucial for practical applications.

Another related work similar to RGBW remosaicing is Quad-Bayer remosaicing [10], [35], [36], [37], [38]—a conversion from Quad-Bayer CFAs, which contain four adjacent pixels clustered with the same-colored pixels in Figure 1(f), to Bayer CFAs. However, because Quad-Bayer CFAs have different numbers of RGB pixels from RGBW CFAs and do not have W pixels, the image priors of CFAs for remosaicing should differ from those of RGBW CFAs.

### C. LEARNED KERNEL-BASED INTERPOLATION

The learned kernel-based interpolation technique generates dynamic filters to synthesize a target image for various image processing and computer vision tasks [39], [40], [41], [42], [43], [44], [45]. For example, Jia et al. [39] proposed a filter-generating network that learns filter coefficients to generate future frames adaptive to motion patterns. Gharbi et al. [40] developed an image enhancement algorithm that estimates filter coefficient maps for color transformation. In [41], [42], and [43], spatiotemporal filter coefficients were dynamically learned to interpolate a new frame between temporally adjacent video frames. Cheng and Chen [45] developed a learned kernel-based frame interpolation algorithm that learns deformable offsets for adaptive kernel shapes. Vien and Lee [44] developed a dynamic interpolation network that learns local filter coefficients to combine useful information and restore missing information. In this work, we develop an RGBW remosaicing algorithm based on learned kernel-based interpolation that learns the dynamic filter coefficients and adaptive kernel shapes to take into account an irregular grid of W pixels.

## III. PROPOSED ALGORITHM

Figure 3 shows an overview of the proposed algorithm, which is composed of three main modules: a feature extractor, learned kernel-based interpolation, and spatially weighted fusion. The feature extractor first extracts the input feature map $\mathcal{F}_{in}$ from the input RGBW image $\mathbf{I}_{in}$. Then, the

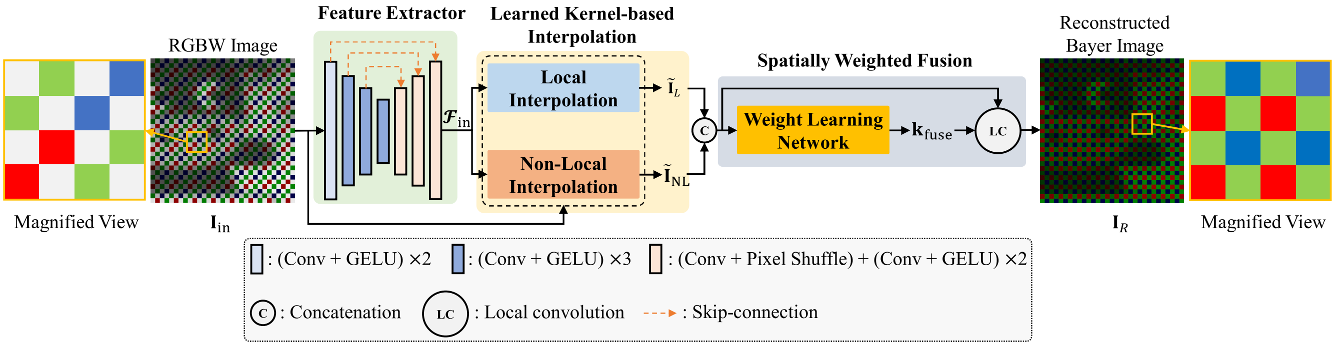


**FIGURE 3.** Overview of the proposed algorithm. Given an RGBW image $\mathbf{I}_{in}$, the proposed algorithm first extracts the feature map $\mathcal{F}_{in}$. Then, these features are used to obtain two intermediate images $\tilde{\mathbf{I}}_L$ and $\tilde{\mathbf{I}}_{NL}$ via local and non-local interpolation blocks, respectively. Finally, the spatially weighted fusion block adaptively fuses $\tilde{\mathbf{I}}_L$ and $\tilde{\mathbf{I}}_{NL}$ to synthesize the final remosaiced Bayer image $\mathbf{I}_R$.

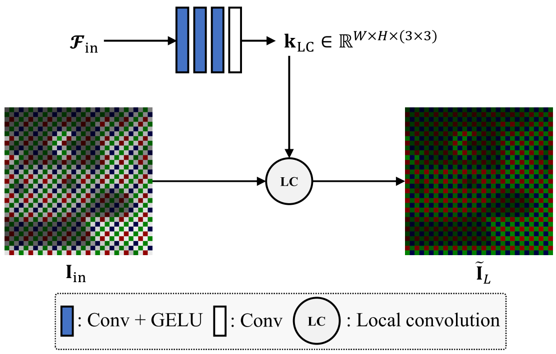


**FIGURE 4.** Architecture of the local interpolation block.

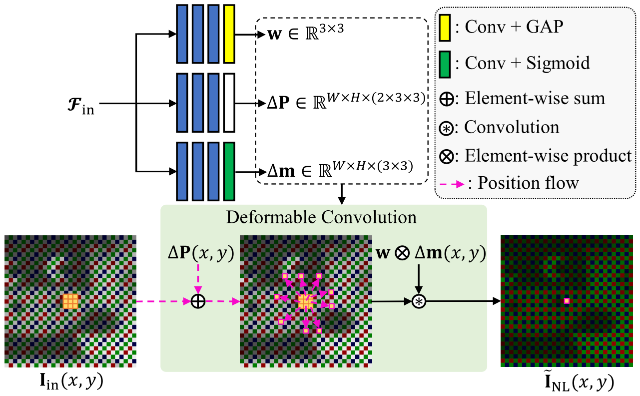


**FIGURE 5.** Architecture of the non-local interpolation block.

learned kernel-based interpolation module generates two intermediate Bayer images, $\tilde{\mathbf{I}}_L$ and $\tilde{\mathbf{I}}_{NL}$, by learning the dynamic weights and offsets simultaneously via local and non-local interpolation, respectively. Finally, the spatially weighted fusion module learns spatial weight maps to merge $\tilde{\mathbf{I}}_L$ and $\tilde{\mathbf{I}}_{NL}$ and obtains the final reconstructed Bayer image $\mathbf{I}_R$.

### A. FEATURE EXTRACTOR

Given an input RGBW image $\mathbf{I}_{in} \in \mathbb{R}^{W\times H\times 1}$, where $W$ and $H$ denote the width and height of the input image, respectively, the feature extractor extracts the deep feature map $\mathcal{F}_{in} \in \mathbb{R}^{W\times H\times 32}$ for local as well as non-local interpolation. We employ U-Net [46], which contains an encoder and a decoder, as the feature extractor, as shown in Figure 3. Specifically, the encoder contains eleven convolution blocks, each of which is composed of a convolution layer with a Gaussian error linear unit (GELU) [47] activation function. It takes an RGBW image $\mathbf{I}_{in}$ and extracts a 4-level feature pyramid with 32, 64, 128, and 256 feature channels. A convolution layer with a $2\times 2$ kernel and a stride of 2 is used for downsampling. The decoder then takes 4-level feature maps via skip-connections and generates the reconstructed feature map $\mathcal{F}_{in}$ in a coarse-to-fine manner through six convolution blocks. A convolution layer with a pixel shuffle operator [48] is used for upsampling.

### B. LEARNED KERNEL-BASED INTERPOLATION

#### 1) OVERVIEW

Several deep networks and modules for image restoration tasks have been applied to RGBW remosaicing to restore missing color pixels in target Bayer images [10], [22], [25]. Although they achieved superior performances, their computational complexity is too high for use in resource-constrained devices. In addition, recent researches [25], [49] found that exploiting local as well as non-local information of inputs improves restoration performance. In this work, we develop an efficient yet effective remosaicing module based on learned kernel-based interpolation by exploiting both local and non-local information to fully utilize W pixels. Specifically, the proposed interpolation module consists of local and non-local interpolation, which are described in detail below.

#### 2) LOCAL INTERPOLATION

As the RGBW CFA has a high proportion of W pixels, which contain a large amount of useful information in the input, effectively extracting meaningful information from the W pixels is essential for effective remosaicing [22]. To this end, we develop a local interpolation block that enables the network to dynamically estimate the relative importance of neighboring pixels. Figure 4 illustrates the architecture of the proposed local interpolation block, which takes the extracted feature map $\mathcal{F}_{in}$ and RGBW image $\mathbf{I}_{in}$ and then adaptively combines them with the neighboring pixels to output the intermediate Bayer image $\widetilde{\mathbf{I}}_L$.

The proposed local interpolation block comprises four convolution layers for estimating the local filter coefficients $\mathbf{k}_{LC} \in \mathbb{R}^{W \times H \times (3\times3)}$, where $W$ and $H$ denote the width and height of the input feature map, respectively. Three $3\times3$ convolution layers with GELU [47] activation functions and a single $3 \times 3$ convolution layer are stacked to generate the filter coefficients. The number of channels in the first three convolution layers is 32, and that in the last convolution layer is 9. Thus, the local interpolation block fuses $3 \times 3$ neighboring pixels adaptively in the RGBW image $\mathbf{I}_{in}$. Specifically, for each pixel $(x, y)$ of $\mathbf{I}_{in}$, we obtain the intermediate Bayer image $\widetilde{\mathbf{I}}_L$ via local convolution (LC) as

$$\widetilde{I}_L(x, y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} k_{LC}(i, j) I_{in}(x + i, y + j), \tag{1}$$

where $(i, j)$ denote local coordinates around $(x, y)$, and the filter coefficients are normalized, *i.e.*, $\sum_{i=-1}^{1} \sum_{j=-1}^{1} k_{LC}(i, j) = 1$.

### 3) NON-LOCAL INTERPOLATION

As shown in Figure 1, RGBW CFAs contain significantly more W pixels than R, G, and B pixels, and more G pixels than R and B pixels. Thus, the interpolation using fixed local kernels in the previous section may fail to accurately estimate the corresponding color pixels because of insufficient color information in the local region. To address this limitation, we develop a non-local interpolation algorithm based on deformable convolution [45], [50], which allows the network to use more pixels outside the rectangular grid of the kernel window.

Figure 5 illustrates the proposed non-local interpolation block, which consists of three branches with the same architecture. Each branch estimates different elements for the deformable convolution. The first branch estimates the global filter coefficients $\mathbf{w} \in \mathbb{R}^{3\times3}$ for the input pixels. The second branch learns 2D offsets $\Delta\mathbf{P} = \{\Delta\mathbf{p}_h, \Delta\mathbf{p}_v\} \in \mathbb{R}^{W \times H \times (2\times3\times3)}$ for all pixels, where $\Delta\mathbf{p}_h$ and $\Delta\mathbf{p}_v$ denote the horizontal and vertical offsets, respectively. These offsets enable non-local interpolation by resampling the pixels outside the rectangular window for convolution. Finally, the third branch learns a modulation mask $\Delta\mathbf{m} \in \mathbb{R}^{W \times H \times (3\times3)}$ that enables the network to focus on the more important pixels, which improves the performance of deformable convolution [50]. In addition, global average pooling (GAP) is employed in the first branch to obtain a global kernel, while a sigmoid function is used in the third branch to constrain the values in $\Delta\mathbf{m}$ in the range of [0, 1]. Similar to the local interpolation block, the number of channels in the first three convolution layers of each branch is 32.

Then, the deformable convolution block in Figure 5 takes the estimated filter coefficients $\mathbf{w}$, offsets $\Delta\mathbf{P}$, and modulation masks $\Delta\mathbf{m}$ and adaptively combines non-local pixels in $\mathbf{I}_{in}$ to generate an intermediate Bayer image $\widetilde{I}_{NL}(x, y)$ at the pixel location $(x, y)$ as

$$\widetilde{I}_{NL}(x, y) = \sum_{i=-1}^{1} \sum_{j=-1}^{1} w(i, j) \Delta m(x, y, i, j) \times I_{in}\big(x + i + \Delta p_h(i, j), y + j + \Delta p_v(i, j)\big). \tag{2}$$

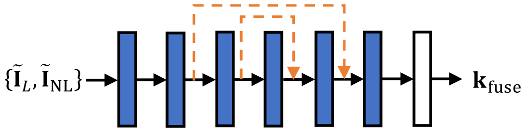


**FIGURE 6.** Architecture of the weight learning network.

### C. SPATIALLY WEIGHTED FUSION

In Figure 3, two intermediate images $\widetilde{\mathbf{I}}_L$ and $\widetilde{\mathbf{I}}_{NL}$ are interpolated using the local and non-local interpolation blocks, respectively. Since the two images are independently interpolated using different blocks, they have different characteristics with complementary information. Therefore, to effectively preserve complementary information from the two images $\widetilde{\mathbf{I}}_L$ and $\widetilde{\mathbf{I}}_{NL}$, we combine them to synthesize the final Bayer image $\mathbf{I}_R$.

To this end, we develop a spatially weighted fusion module to estimate the dynamic filter coefficients $\mathbf{k}_{fuse}$ that determine the adaptive weights between $\widetilde{\mathbf{I}}_L$ and $\widetilde{\mathbf{I}}_{NL}$ for fusion. Figure 6 shows the architecture of the weight learning network. It contains seven $3\times3$ convolution layers with symmetric skip connections to estimate $\mathbf{k}_{fuse}(x, y) \in \mathbb{R}^{W \times H \times 2 \times (d\times d)}$ at each pixel location $(x, y)$, where $d$ denotes the size of the kernel—$d \times d$ neighboring pixels in each image are fused. Each convolution layer is followed by a GELU activation function except for the last convolution. The number of channels in the first six convolution layers is 32, while that of the last convolution layer is two.

Finally, the fused Bayer image $\mathbf{I}_R$ is obtained via LC as

$$I_R(x, y) = \sum_{l=1}^{2} \sum_{i} \sum_{j} k_{fuse}(i, j, l) \widetilde{I}(x + i, y + j, l), \tag{3}$$

where $(i, j)$ are the local coordinates around $(x, y)$ and $\widetilde{\mathbf{I}} = \{\widetilde{\mathbf{I}}_L, \widetilde{\mathbf{I}}_{NL}\} \in \mathbb{R}^{W \times H \times 2}$. The coefficients are normalized, *i.e.*, $\sum_{l=1}^{2} \sum_i \sum_j k_{fuse}(i, j, l) = 1$.

### D. LOSS FUNCTIONS

To train the proposed algorithm, we define the total loss $\mathcal{L}_{total}$ as the weighted sum of the Bayer loss $\mathcal{L}_{Bayer}$, reconstruction loss $\mathcal{L}_r$, SSIM loss $\mathcal{L}_{SSIM}$, and perceptual loss $\mathcal{L}_{per}$ as

$$\mathcal{L}_{total} = \mathcal{L}_{Bayer} + \lambda_r \mathcal{L}_r + \lambda_{SSIM} \mathcal{L}_{SSIM} + \lambda_{per} \mathcal{L}_{per}, \tag{4}$$

where $\lambda_r$, $\lambda_{SSIM}$, and $\lambda_{per}$ are the hyperparameters to balance the four losses.

**TABLE 1.** Quantitative comparison of the proposed algorithm with the conventional algorithms on the test sets using five quality metrics. For each metric, the best result is **boldfaced**, while the second best is underlined.

| | MIPI-RGBW | | | | | Kodak | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR (↑) | SSIM (↑) | LPIPS (↓) | KLD (↓) | M4 (↑) | PSNR (↑) | SSIM (↑) | LPIPS (↓) | KLD (↓) | M4 (↑) |
| DRUNet | 39.79 | 0.9839 | 0.0254 | 0.0273 | 75.61 | **44.38** | 0.9906 | 0.0146 | 3.2492 | 9.23 |
| NAFNet | 39.77 | 0.9838 | 0.0285 | 0.0349 | 75.04 | 43.53 | 0.9905 | 0.0146 | 3.3671 | 8.35 |
| UNet-ResT | 39.75 | 0.9838 | 0.0283 | 0.0318 | 75.17 | 44.26 | 0.9907 | 0.0156 | 3.2672 | 9.06 |
| NAFNet-SSA | 39.80 | 0.9840 | 0.0277 | 0.0382 | 74.99 | 44.14 | **0.9911** | **0.0132** | 3.4691 | 7.89 |
| OTST | **39.91** | **0.9841** | 0.0275 | 0.0430 | 74.96 | 44.19 | 0.9910 | 0.0143 | 3.4449 | 8.01 |
| Proposed | 39.68 | 0.9839 | **0.0244** | **0.0206** | **75.77** | 42.61 | 0.9877 | 0.0298 | **3.1191** | **9.57** |

The Bayer loss $\mathcal{L}_{Bayer}$ measures the difference between the reconstructed Bayer image $\mathbf{I}_R$ and its ground-truth $\mathbf{I}_{gt}$ as

$$\mathcal{L}_{Bayer} = \|\mathbf{I}_R - \mathbf{I}_{gt}\|_1. \tag{5}$$

As the estimated Bayer image $\mathbf{I}_R$ should be demosaiced for use in real applications, we compute the losses using the full-color images obtained by demosaicing. To this end, we employ a demosaicing algorithm $\mathcal{G}$ [51] with pretrained weights from [10] and [22] and the gamma function to obtain full-color images as

$$\mathbf{I}_R^{RGB} = \left(\mathcal{G}\left((\mathbf{I}_R)^{1/\gamma}\right)\right)^{\gamma}, \tag{6}$$

$$\mathbf{I}_{gt}^{RGB} = \left(\mathcal{G}\left((\mathbf{I}_{gt})^{1/\gamma}\right)\right)^{\gamma} \tag{7}$$

for $\mathbf{I}_R$ and $\mathbf{I}_{gt}$, respectively. In this work, we fixed $\gamma = 2.2$. We compute the $l_1$-norm as the reconstruction loss $\mathcal{L}_r$ as

$$\mathcal{L}_r = \|\mathbf{I}_R^{RGB} - \mathbf{I}_{gt}^{RGB}\|_1. \tag{8}$$

We employ the SSIM loss $\mathcal{L}_{SSIM}$ to qualify the structure and contrast information, which is given by

$$\mathcal{L}_{SSIM} = 1 - SSIM(\mathbf{I}_R^{RGB}, \mathbf{I}_{gt}^{RGB}), \tag{9}$$

where $SSIM(\cdot)$ is the structural similarity index. Finally, we use the perceptual loss $\mathcal{L}_{per}$ to compare the high-level differences between the two images, which is defined as

$$\mathcal{L}_{per} = LPIPS(\mathbf{I}_R^{RGB}, \mathbf{I}_{gt}^{RGB}), \tag{10}$$

where $LPIPS(\cdot)$ denotes the LPIPS score [52] using a pretrained AlexNet [53].

## IV. EXPERIMENTAL RESULTS

### A. EXPERIMENTAL SETTINGS

#### 1) DATASET

We evaluate the performance of the proposed algorithm on the MIPI-RGBW [10] and Kodak[2] datasets.

- MIPI-RGBW [10]: This dataset contains captured RGBW and Bayer image pairs with a resolution of 1800 × 1200, which are composed of 70 training and 15 validation pairs. We use the validation set for the testing and randomly select ten images from the training pairs for validation; thus, the new training set contains 60 pairs.
- Kodak: This dataset was constructed by synthesizing RGBW and Bayer image pairs using 24 full-color images with a resolution of 3072 × 2048. Specifically, following [12], [27], and [54], we synthesize a panchromatic W image using the linear relation $W = R + G + B$ and then generate a pair of a synthetic RGBW image with the RGBW pattern in Figure 1(e) and a Bayer image with the GBRG pattern in Figure 1(a).

Although we evaluate the proposed algorithm using these two datasets, we use only the MIPI-RGBW dataset for training.

#### 2) IMPLEMENTATION DETAILS

The proposed algorithm comprises three modules: a feature extractor, learned kernel-based interpolation, and spatially weighted fusion. We experimentally found that training these networks in a two-step manner provides better performance than end-to-end training. Specifically, we first train the feature extractor and learned kernel-based interpolation jointly to obtain two intermediate images. Then, with these two modules fixed, we train the spatially weighted fusion module end-to-end. The fixed feature extractor and learned kernel-based interpolation enable training the spatially weighted fusion module to converge faster and the network to improve reconstruction performance.

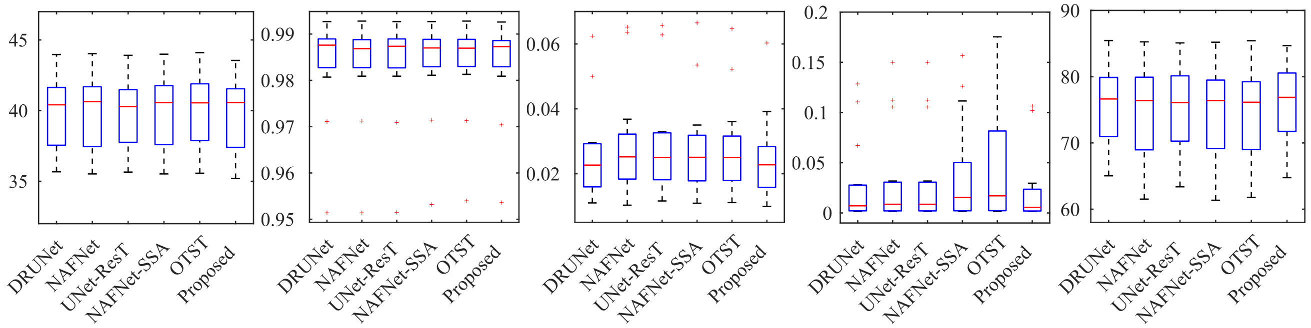
We use the Adam optimizer [55] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, an initial learning rate of $10^{-4}$, and a batch size of 5 for 1000 epochs. The learning rate is halved after the 200th and 300th epochs. We augment the training dataset by a geometric transformation of 90° rotation, random flipping, and cropping of 256 × 256 patches. The hyperparameters $\lambda_r$, $\lambda_{SSIM}$, and $\lambda_{per}$ in (4) are fixed to 1.0, 0.75, and 0.1, respectively, to provide the best overall subjective quality.

#### 3) STATE-OF-THE-ART ALGORITHMS FOR COMPARISONS

We evaluate the RGBW remosaicing performance of the proposed algorithm with those of five top-ranked algorithms in the challenges [10], [22]: DRUNet, NAFNet, and UNet-ResT in MIPI 2022 [10] and NAFNet-SSA and OTST [25] in MIPI 2023 [22]. For a fair comparison, we retrained these algorithms using the training dataset in Section IV-A1 using the parameter settings recommended by the respective

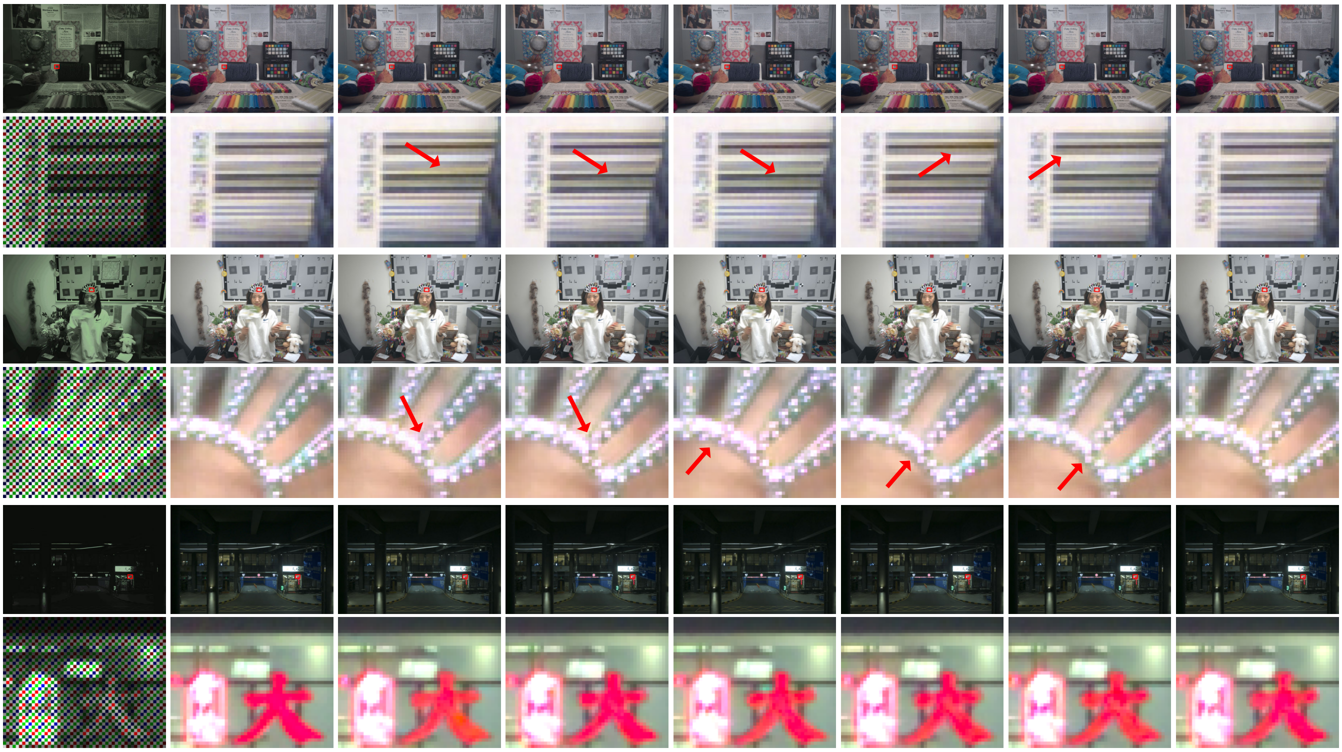
---

[2] https://r0k.us/graphics/kodak/



(a) PSNR (b) SSIM (c) LPIPS (d) KLD (e) M4

**FIGURE 7.** Comparison of box plots for the five metrics in Table 1 on the MIPI-RGBW dataset.



(a) RGBW images (b) Ground-truth (c) DRUNet (d) NAFNet (e) UNet-ResT (f) NAFNet-SSA (g) OTST (h) Proposed

**FIGURE 8.** Qualitative comparison of demosaiced full-color images obtained using a simple ISP [22] on the MIPI-RGBW dataset. The second, fourth, and sixth rows show the enlarged parts for the red rectangles in the first, third, and fifth rows, respectively.

authors. The source code of UNet-ResT was obtained from the author's website,[3] whereas we implemented DRUNet, NAFNet, NAFNet-SSA, and OTST. We will provide the source codes and pretrained models on our project website.[4]

### B. PERFORMANCE COMPARISON

We evaluate the RGBW remosaicing performance using five quality metrics: PSNR, SSIM, LPIPS [52], Kullback-Leibler divergence (KLD), and M4. As each quality metric assesses different aspects of the image quality, using a single metric for performance comparison is difficult. Thus, an overall weighted metric, M4, was developed in the RGBW remosaicing challenge [22] for the overall performance evaluation, which is defined as

$$M4 = \text{PSNR} \times \text{SSIM} \times 2^{(1-\text{LPIPS}-\text{KLD})}. \tag{11}$$

The PSNR, SSIM, and LPIPS scores are computed between full-color images converted from the restored and ground-truth Bayer images using a simple ISP [22], whereas the KLD score is computed directly from the Bayer images. Higher PSNR, SSIM, and M4 scores indicate better performance, whereas lower LPIPS and KLD scores indicate better performance.

---

[3] https://github.com/Joyies/ECCVW22-RGBW-Remosaic-Challenge

[4] https://github.com/viengiaan/RGBW_Remosaicing

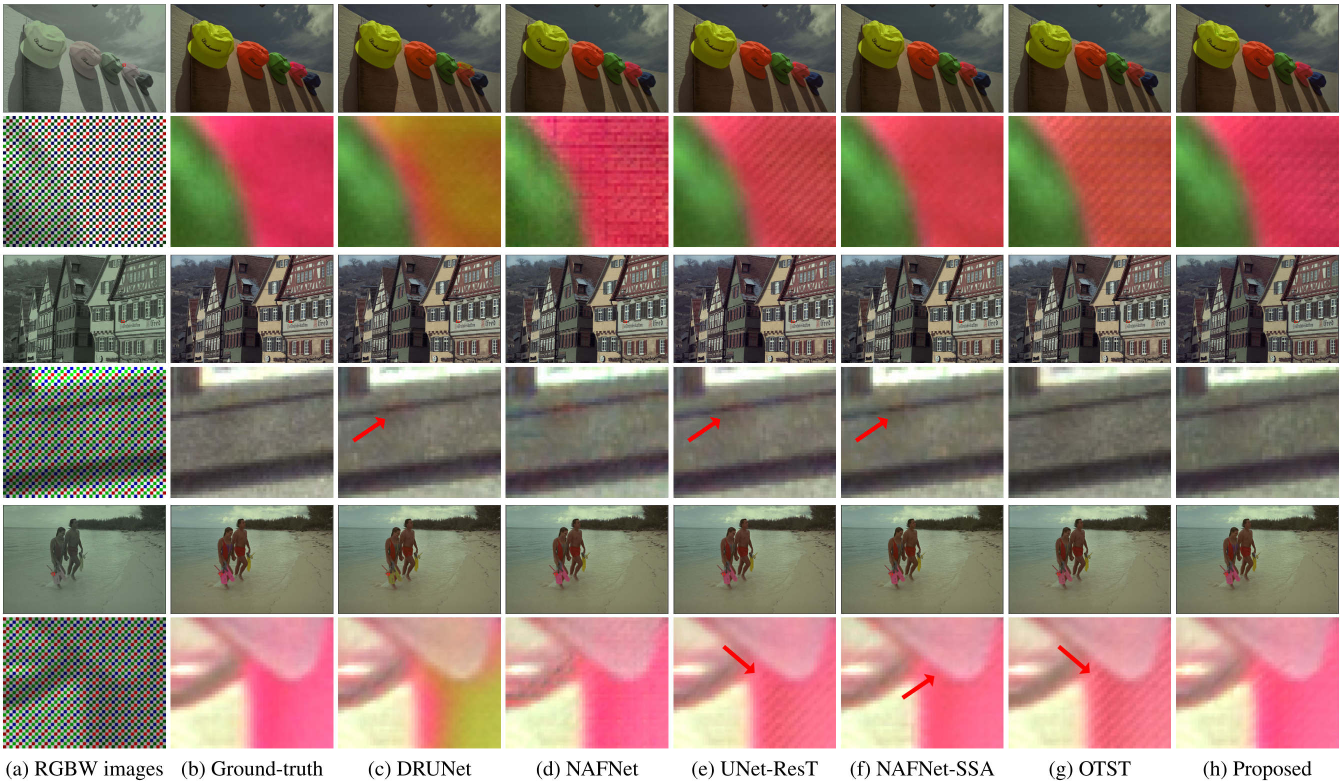

(a) RGBW images (b) Ground-truth (c) DRUNet (d) NAFNet (e) UNet-ResT (f) NAFNet-SSA (g) OTST (h) Proposed

**FIGURE 9.** **Qualitative comparison of demosaiced full-color images obtained using a simple ISP [22] on the Kodak dataset. The second, fourth, and sixth rows show the enlarged parts for the red rectangles in the first, third, and fifth rows, respectively.**

Table 1 quantitatively compares the RGBW remosaicing performances on the MIPI-RGBW and Kodak datasets. First, the proposed algorithm achieves the best KLD and M4 scores on both datasets because it synthesizes the most accurate Bayer images by exploiting both local and non-local information in the RGBW images. Specifically, the proposed algorithm obtains 0.16 and 0.34 higher M4 scores than the second-best algorithm, DRUNet, on the MIPI-RGBW and Kodak datasets, respectively. Second, the proposed algorithm yields the best LPIPS and comparable SSIM and PSNR scores to those of the best algorithm, OTST, on the MIPI-RGBW dataset. Note that, although the proposed algorithm provides inferior PSNR and SSIM scores to conventional algorithms, it produces visually more plausible results as will be shown subsequently. This is because the proposed algorithm is trained to minimize both pixel-wise differences and perceptual quality as described in Section III-D, whereas the conventional algorithms consider only the pixel-wise differences. The effects of different losses on the performance will be analyzed in Section IV-C4.

Figure 7 shows the box plots for the five metrics presented in Table 1 for the MIPI-RGBW dataset. The red lines and crosses for each metric denote the median values and outliers, respectively. The proposed algorithm achieves the lowest median values for LPIPS and KLD scores in Figures 7(c) and (d), respectively, and the highest median value for M4 in Figure 7(e). In addition, the proposed algorithm produces the smallest number of outliers. These results confirm that the proposed algorithm is more effective and stable than the conventional algorithms.

Figure 8 compares the demosaiced full-color images obtained by applying a simple ISP [22] to the Bayer images reconstructed by each algorithm on the MIPI-RGBW dataset. DRUNet, NAFNet, UNet-ResT, and NAFNet-SSA in Figures 8(c)–(f) generate false-color artifacts in regions with high-frequency textures because they fail to effectively extract useful information from the W pixels for texture restoration. Although OTST in Figure 8(g) produces better results, it still generates visible artifacts in the fourth row. In contrast, the proposed algorithm in Figure 8(h) produces full-color images with less false-color artifacts by accurately recovering textures using both local and non-local interpolation. For example, conventional algorithms yield strong false-color artifacts in the second, fourth, and sixth rows because of inaccurate texture restoration, *e.g.*, the high contrast edges in the fourth row, which are effectively recovered by the proposed algorithm.

Figure 9 shows the RGBW remosaicing results for the Kodak dataset. DRUNet in Figure 9(c) produces severe color distortions, especially in the second and sixth rows, due to the ineffective exploitation of W pixels for the estimation of missing color pixels. By adopting more complicated restoration modules, NAFNet, UNet-ResT, NAFNet-SSA, and OTST in Figures 9(d)–(g), respectively, generate better color tones than DRUNet. However, they still produce strong visible artifacts in the homogeneous regions in the second and sixth

**TABLE 2.** Comparisons of computational complexity in terms of the number of network parameters, GMACC, and runtime in seconds.

| | # Param. (M) | GMACC | Runtime (s) |
|---|---|---|---|
| DRUNet | 32.64 | 4727.0 | 0.3794 |
| NAFNet | 37.17 | 836.3 | 0.4160 |
| UNet-ResT | 74.40 | 5224.7 | 1.0442 |
| NAFNet-SSA | 32.27 | 954.0 | 0.2622 |
| OTST | 35.37 | 1762.5 | 2.4445 |
| Proposed | **2.20** | **721.1** | **0.1257** |

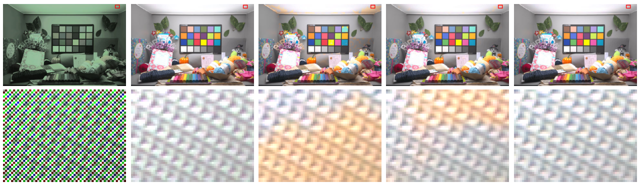

Input Ground-truth RDN DRDN U-Net

**FIGURE 10.** Qualitative comparison of synthesized results obtained using different feature extractors. The second row shows the enlarged parts for the red rectangles in the first row.

rows and color-false artifacts in the fourth row. In contrast, the proposed algorithm in Figure 9(h) yields results that are most similar to the ground-truths in Figure 9(b) with less visible artifacts, which confirms the effectiveness of the proposed learned kernel-based interpolation blocks.

Table 2 compares the numbers of network parameters and giga multiply-accumulate (GMACC) operations and the average runtime of different algorithms to process the test images in the MIPI-RGBW dataset on an Nvidia RTX 3090 GPU. The proposed algorithm is the most efficient for all the complexity measurement metrics. In particular, the proposed algorithm requires 14 times fewer network parameters and 1.3 times fewer GMACCs and is 2.1 times faster than the second-most efficient algorithm, NAFNet-SSA. This superior efficiency is due to the lower complexity of learned kernel-based interpolation compared to that of end-to-end mapping in conventional algorithms. These results indicate that the proposed algorithm provides comparable or even better remosaicing performance than conventional algorithms, while demanding significantly lower computational complexity, thus, achieving a desirable tradeoff between performance and complexity.

### *C. MODEL ANALYSIS*

We conduct several ablation studies to analyze the contributions of the key components of the proposed algorithm. In addition, we analyze the effects of different fusion strategies for the fusion block and each loss component on the synthesis performance. All the experiments are performed using the MIPI-RGBW dataset.

#### 1) FEATURE EXTRACTOR

To analyze the effectiveness of the feature extractor of the proposed algorithm, we train the proposed network using three widely used backbone architectures: residual dense network (RDN) [56], dilated residual dense network (DRDN) [57], and U-Net [46]. Table 3 compares the synthesis performances. U-Net achieves the best performance for all metrics with the smallest network parameters and GMACCs. This indicates that the multiscale feature map extracted by U-Net is more effective for RGBW remosaicing than the single-scale feature maps extracted by the RDN and DRDN. We also visually compare the results obtained using different feature extractors in Figure 10. The RDN and DRDN produce results with false-color artifacts due to ineffective feature extraction, whereas U-Net produces considerably better results by exploiting multiscale rich information.

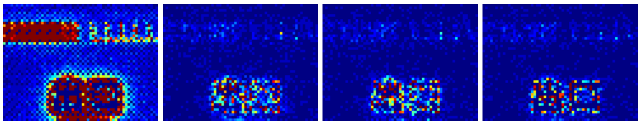

(a) Input (b) Local (c) Non-local (d) Proposed

**FIGURE 11.** Comparison of the error maps for different settings of interpolation.

#### 2) LOCAL AND NON-LOCAL INTERPOLATION

To analyze the effectiveness of different interpolation blocks, local and non-local interpolation, in Figure 3, we train the proposed algorithm using different settings. Table 4 compares the synthesis performances under different settings. First, using only the local interpolation block yields the worst performance for all metrics, because local neighboring pixels may not contain sufficient color information given the high sampling rate of W pixels, to recover the corresponding color pixels accurately. Second, non-local interpolation achieves better performance because it uses more color information outside the rectangular grid of the filter. Finally, combining the results of local and non-local interpolation via spatially weighted fusion significantly improves the performance by exploiting the complementary information extracted from both interpolation blocks. In addition, Figure 11 shows the error maps for each setting in Table 4. Local and non-local interpolation yield complementary results, and the proposed algorithm improves the performance by combining complementary information.

#### 3) FUSION BLOCK

We analyze the effectiveness of the proposed spatially weighted fusion block described in Section III-C by training the proposed algorithm using various fusion strategies, *i.e.*, global $1 \times 1$ convolution, DRDN [57], and spatially weighted fusion using three kernel sizes, *i.e.*, $d = 1$, $d = 3$, and $d = 5$. Note that if $d = 1$, the fusion in (3) becomes the weighted sum of two intermediate images with pixel-wise weights. Table 5 and Figure 12 presents quantitative and qualitative comparisons of the different fusion strategies. The $1 \times 1$ convolution yields the worst performance because it cannot fully exploit the complementary information in

**TABLE 3.** Comparison of synthesis performance using different backbone networks for feature extractor.

| | PSNR (↑) | SSIM (↑) | LPIPS (↓) | KLD (↓) | M4 (↑) | # Param. (M) | GMACC |
|---|---|---|---|---|---|---|---|
| RDN [56] | 38.74 | 0.9838 | 0.0321 | 0.0257 | 73.41 | 2.26 | 4484.5 |
| DRDN [57] | 39.31 | **0.9839** | 0.0297 | 0.0267 | 74.51 | 2.26 | 4888.3 |
| U-Net [46] | **39.68** | **0.9839** | **0.0244** | **0.0206** | **75.77** | 2.20 | 721.1 |

**TABLE 4.** Impacts of local interpolation, non-local interpolation, and spatially weighted fusion on synthesis performance.

| Interpolation | | Spatially Weighted Fusion | PSNR (↑) | SSIM (↑) | LPIPS (↓) | KLD (↓) | M4 (↑) | # Param. (M) | GMACC |
|---|---|---|---|---|---|---|---|---|---|
| Local | Non-local | | | | | | | | |
| ✓ | | | 32.70 | 0.9811 | 0.0539 | 0.0409 | 60.12 | 2.04 | 375.4 |
| | ✓ | | 39.18 | 0.9833 | 0.0305 | 0.0386 | 73.60 | 2.11 | 512.1 |
| ✓ | ✓ | ✓ | **39.68** | **0.9839** | **0.0244** | **0.0206** | **75.77** | 2.20 | 721.1 |

**TABLE 5.** Comparison of synthesis performance using different fusion strategies.

| | | PSNR (↑) | SSIM (↑) | LPIPS (↓) | KLD (↓) | M4 (↑) | # Param. (M) | GMACC |
|---|---|---|---|---|---|---|---|---|
| 1 × 1 Conv | | 39.40 | 0.9833 | 0.0306 | 0.0278 | 74.56 | 2.14 | 577.6 |
| DRDN [57] | | 39.62 | 0.9838 | 0.0298 | 0.0230 | 75.27 | 2.36 | 1056.7 |
| Spatially Weighted Fusion | $d = 1$ | **39.68** | **0.9839** | **0.0244** | **0.0206** | **75.77** | 2.20 | 721.1 |
| | $d = 3$ | 39.61 | 0.9837 | 0.0251 | 0.0233 | 75.47 | 2.21 | 741.1 |
| | $d = 5$ | 39.65 | 0.9838 | 0.0247 | 0.0209 | 75.69 | 2.23 | 781.0 |

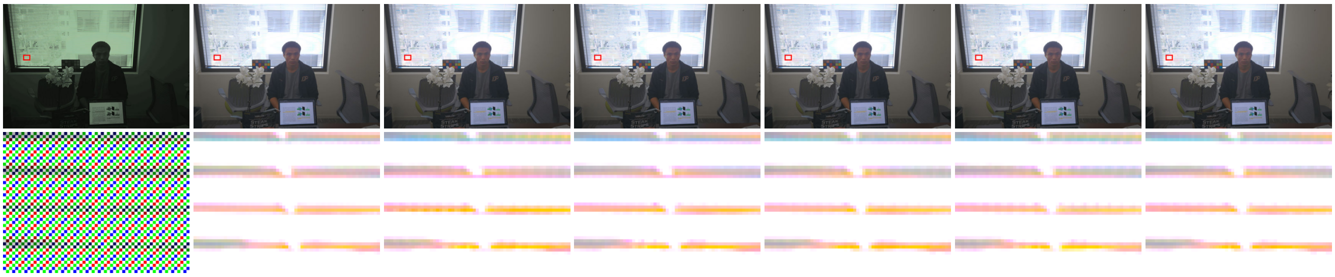

(a) RGBW images (b) Ground-truth (c) 1 × 1 Conv (d) DRDN (e) $d = 1$ (f) $d = 3$ (g) $d = 5$

**FIGURE 12.** Qualitative comparison of results using different fusion strategies. The second row shows the enlarged parts for the red rectangles in the first row.

**TABLE 6.** Impacts of losses on synthesis performance.

| $\mathcal{L}_{\text{Bayer}}$ | $\mathcal{L}_r$ | $\mathcal{L}_{\text{SSIM}}$ | $\mathcal{L}_{\text{per}}$ | PSNR (↑) | SSIM (↑) | LPIPS (↓) | KLD (↓) | M4 (↑) |
|---|---|---|---|---|---|---|---|---|
| ✓ | | | | 39.63 | 0.9835 | 0.0285 | **0.0169** | 75.63 |
| ✓ | ✓ | | | 39.71 | **0.9840** | 0.0286 | 0.0202 | 75.64 |
| ✓ | ✓ | ✓ | | **39.72** | **0.9840** | 0.0286 | 0.0205 | 75.65 |
| ✓ | ✓ | | ✓ | 39.67 | 0.9838 | 0.0248 | 0.0200 | **75.77** |
| ✓ | ✓ | ✓ | ✓ | 39.68 | 0.9839 | **0.0244** | 0.0206 | **75.77** |

each interpolated image; thus, it generates visible artifacts in the result, *e.g.,* the splotchy artifacts around the edges in Figure 12(c). A more complex network, the DRDN, provides better performance than the 1 × 1 convolution. For example, the DRDN in Figure 12(d) generates a result with less splotchy artifacts. However, this comes at the cost of 1.83 times higher complexity in GMACCs compared to 1 × 1 convolution. In contrast, the proposed spatially weighted fusion with $d = 1$ achieves the best quantitative and qualitative performances with comparable complexities. Furthermore, as the kernel size increases, the synthesis performance deteriorates because large kernels use more neighboring pixels for fusion, making learning more difficult. For example, the spatially weighted fusion with $d = 3$ and $d = 5$ in Figures 12(f) and (g), respectively, yield visible artifacts near the edges, indicating that large kernels fail to effectively determine the useful neighboring pixels for fusion.

### 4) LOSS FUNCTIONS

Finally, we conduct ablation studies to analyze the contribution of each component in the loss function in Section III-D by training the proposed algorithm using various combinations of losses. Table 6 compares average scores. Using only the Bayer loss $\mathcal{L}_{Bayer}$ yields the worst performance on PSNR, SSIM, LPIPS, and M4 but the best KLD score. The addition of the losses computed between the full-color images, *i.e.*, reconstruction loss $\mathcal{L}_r$, SSIM loss $\mathcal{L}_{SSIM}$, and perceptual loss $\mathcal{L}_{per}$, improves the performance for PSNR, SSIM, and LPIPS, but slightly degrades the KLD performance. For example, adding $\mathcal{L}_r$ increases the PSNR and SSIM scores by minimizing the pixel-wise differences, and adding both $\mathcal{L}_r$ and $\mathcal{L}_{SSIM}$ achieves the highest PNSR and SSIM scores by preserving the structure and contrast. The combination of $\mathcal{L}_r$ and $\mathcal{L}_{per}$ yields the second-best LPIPS and best M4 scores by preserving high-level context. Finally, the proposed loss $\mathcal{L}_{total}$ achieves the best LPIPS and M4 scores and comparable PSNR, SSIM, and KLD scores to other combinations to generate images with the best overall perceptual quality with comparable fidelity.

## V. CONCLUSION

RGBW remosaicing is an interpolation technique for converting RGBW images into Bayer images to use the existing ISP pipeline to generate full-color images. Although recent learning-based RGBW remosaicing algorithms have shown significant performance improvement, they demand high computational and memory complexities, limiting their practical uses. In this work, we proposed an efficient and effective RGBW sensor remosaicing algorithm based on learned kernel-based interpolation. The proposed algorithm comprises a feature extractor, learned kernel-based interpolation, and spatially weighted fusion modules. We first extracted deep feature maps from an input RGBW image using a feature extractor. The main contribution of this work is the development of a learned kernel-based interpolation module that generates two intermediate Bayer images by exploiting local and non-local pixels in the input RGBW image. Finally, the spatially weighted fusion block aggregates complementary information in the intermediate Bayer images to obtain a reconstructed Bayer image. Experimental results demonstrated that the proposed algorithm provides comparable or even better remosaicing performance than state-of-the-art algorithms while requiring 14 times fewer network parameters, 1.3 times fewer GMACCs, and 2.1 times faster runtime.

## REFERENCES

[1] J. Wang, C. Zhang, and P. Hao, "New color filter arrays of high light sensitivity and high demosaicking performance," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2011, pp. 3153–3156.

[2] J. Li, C. Bai, Z. Lin, and J. Yu, "Automatic design of high-sensitivity color filter arrays with panchromatic pixels," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 870–883, Feb. 2017.

[3] M. Kumar, E. O. Morales, J. E. Adams, and W. Hao, "New digital camera sensor architecture for low light imaging," in *Proc. IEEE Int. Conf. Image Process.*, Nov. 2009, pp. 2681–2684.

[4] J. T. Compton and J. F. Hamilton Jr., "Image sensor with improved light sensitivity," U.S. Patent 2007 0 024 931, Feb. 1, 2007.

[5] H. Honda, Y. Iida, Y. Egawa, and H. Seki, "A color CMOS imager with 4 × 4 white-RGB color filter array for increased low-illumination signal-to-noise ratio," *IEEE Trans. Electron Devices*, vol. 56, no. 11, pp. 2398–2402, Nov. 2009.

[6] I. Hirota, "Method for processing signal of solid-state imaging device, and imaging apparatus," U.S. Patent 8 436 925, Oct. 2, 2013.

[7] S. Hikosaka, "Imaging device and imaging system," U.S. Patent 10 567 712, Oct. 29, 2020.

[8] J. Li, C. Bai, and H. Huang, "Universal demosaicking for interpolation-friendly RGBW color filter arrays," *IEEE Trans. Image Process.*, vol. 32, pp. 3592–3605, 2023.

[9] B. E. Bayer, "Color imaging array," U.S. Patent 3 971 065, Jul. 20, 1976.

[10] Q. Yang et al., "MIPI 2022 challenge on RGBW sensor re-mosaic: Dataset and report," in *Proc. Eur. Conf. Comput. Vis. Workshops*, Oct. 2022, pp. 36–45.

[11] C. Zhang, Y. Li, J. Wang, and P. Hao, "Universal demosaicking of color filter arrays," *IEEE Trans. Image Process.*, vol. 25, no. 11, pp. 5173–5186, Nov. 2016.

[12] C. Kwan, B. Chou, L. M. Kwan, and B. Budavari, "Debayering RGBW color filter arrays: A pansharpening approach," in *Proc. IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf.*, Oct. 2017, pp. 94–100.

[13] S. H. Lee, P. Oh, and M. G. Kang, "Three dimensional colorization based image/video reconstruction from white-dominant RGBW pattern images," *Digit. Signal Process.*, vol. 93, pp. 87–101, Oct. 2019.

[14] C. Bai and J. Li, "Convolutional sparse coding for demosaicking with panchromatic pixels," *Signal Process., Image Commun.*, vol. 77, pp. 20–27, Sep. 2019.

[15] M. Kang and M. Jung, "Low-dimensional manifold model for demosaicking from a RGBW color filter array," *Signal, Image Video Process.*, vol. 14, no. 1, pp. 143–150, Feb. 2020.

[16] C. Kwan and B. Chou, "Further improvement of debayering performance of RGBW color filter arrays using deep learning and pansharpening techniques," *J. Imag.*, vol. 5, no. 8, p. 68, Aug. 2019.

[17] C. Kwan, J. Larkin, and B. Ayhan, "Demosaicing of CFA 3.0 with applications to low lighting images," *Sensors*, vol. 20, no. 12, p. 3423, Jun. 2020.

[18] L. Condat, "A generic variational approach for demosaicking from an arbitrary color filter array," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Nov. 2009, pp. 1625–1628.

[19] J. Gu, P. J. Wolfe, and K. Hirakawa, "Filterbank-based universal demosaicking," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2010, pp. 1981–1984.

[20] Q. Tian, S. Lansel, J. E. Farrell, and B. A. Wandell, "Automating the design of image processing pipelines for novel color filter arrays: Local, linear, learned (L3) method," *Proc. SPIE*, vol. 9023, Mar. 2014, Art. no. 90230K.

[21] H. Jiang, Q. Tian, J. Farrell, and B. A. Wandell, "Learning the image processing pipeline," *IEEE Trans. Image Process.*, vol. 26, no. 10, pp. 5032–5042, Oct. 2017.

[22] Q. Sun et al., "MIPI 2023 challenge on RGBW remosaic: Methods and results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2023, pp. 2877–2884.

[23] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 6360–6376, Oct. 2022.

[24] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2022, pp. 17–33.

[25] Z. Fan, X. Wu, F. Meng, Y. Wu, and F. Zhang, "OTST: A two-phase framework for joint denoising and remosaicing in RGBW CFA," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2023, pp. 2832–2841.

[26] M. Singh and T. Singh, "Linear universal demosaicking of regular pattern color filter arrays," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2012, pp. 1277–1280.

[27] A. Chakrabarti, "Learning sensor multiplexing design through back-propagation," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2016, pp. 3089–3097.

[28] S. M. A. Sharif and Y. J. Jung, "Deep color reconstruction for a sparse color sensor," *Opt. Exp.*, vol. 27, no. 17, pp. 23661–23681, Aug. 2019.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[30] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan, and S. Liu, "NBNet: Noise basis learning for image denoising with subspace projection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2021, pp. 4894–4904.

[31] Y. Cai, J. Lin, H. Wang, X. Yuan, H. Ding, Y. Zhang, R. Timofte, and L. V. Gool, "Degradation-aware unfolding half-shuffle transformer for spectral compressive imaging," in *Proc. Adv. Neural Inf. Process. Syst.*, Nov. 2022, pp. 37749–37761.

[32] Y. Cai, J. Lin, X. Hu, H. Wang, X. Yuan, Y. Zhang, R. Timofte, and L. Van Gool, "Mask-guided spectral-wise transformer for efficient hyperspectral image reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2022, pp. 17481–17490.

[33] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.

[34] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Oct. 2021, pp. 9992–10002.

[35] J. Jia, H. Sun, X. Liu, L. Xiao, Q. Xu, and G. Zhai, "Learning rich information for Quad Bayer remosaicing and denoising," in *Proc. Eur. Conf. Comput. Vis. Workshops*, Oct. 2022, pp. 175–191.

[36] X. Wu, Z. Fan, J. Zheng, Y. Wu, and F. Zhang, "Learning to joint remosaic and denoise in quad Bayer CFA via universal multi-scale channel attention network," in *Proc. Eur. Conf. Comput. Vis. Workshops*, Oct. 2022, pp. 147–160.

[37] H. Zeng, K. Feng, J. Cao, S. Huang, and Y. Zhao, "Inheriting Bayer's legacy: Joint remosaicing and denoising for quad Bayer," 2023, *arXiv:2303.13571*.

[38] T. Wu, K. Sugimoto, M. Tanaka, and Y. Oguro, "High dynamic range imaging with multi-exposure binning on quad Bayer color filter array," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2023, pp. 2015–2019.

[39] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2016, pp. 667–675.

[40] M. Gharbi, J. Chen, J. T. Barron, S. W. Hasinoff, and F. Durand, "Deep bilateral learning for real-time image enhancement," *ACM Trans. Graph.*, vol. 36, no. 4, pp. 1–12, Jul. 2017.

[41] S. Niklaus, L. Mai, and F. Liu, "Video frame interpolation via adaptive separable convolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 261–270.

[42] H. Choi and I. V. Bajic, "Deep frame prediction for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 1843–1855, Jul. 2020.

[43] H. Lee, T. Kim, T.-Y. Chung, D. Pak, Y. Ban, and S. Lee, "AdaCoF: Adaptive collaboration of flows for video frame interpolation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 5315–5324.

[44] A. G. Vien and C. Lee, "Exposure-aware dynamic weighted learning for single-shot HDR imaging," in *Proc. Eur. Conf. Comput. Vis.*, Oct. 2022, pp. 435–452.

[45] X. Cheng and Z. Chen, "Multiple video frame interpolation via enhanced deformable separable convolution," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 10, pp. 7029–7045, Oct. 2022.

[46] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Med. Imag. Comput. Comput.-Asiated Intervent.*, Nov. 2015, pp. 234–241.

[47] D. Hendrycks and K. Gimpel, "Bridging nonlinearities and stochastic regularizers with Gaussian error linear units," 2016, *arXiv:1606.08415*.

[48] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1874–1883.

[49] C. Zheng, Y. Zhang, J. Gu, Y. Zhang, L. Kong, and X. Yuan, "Cross aggregation transformer for image restoration," in *Proc. Adv. Neural Inf. Process. Syst.*, Oct. 2022, pp. 25478–25490.

[50] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 9300–9308.

[51] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Trans. Graph.*, vol. 35, no. 6, pp. 1–12, Dec. 2016.

[52] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[53] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2012, pp. 1097–1105.

[54] P.-H. Su, P.-C. Chen, and H. H. Chen, "Compensation of spectral mismatch to enhance WRGB demosaicking," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2015, pp. 68–72.

[55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, Sep. 2015.

[56] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 7, pp. 2480–2495, Jul. 2021.

[57] Z. Liu, W. Lin, X. Li, Q. Rao, T. Jiang, M. Han, H. Fan, J. Sun, and S. Liu, "ADNet: Attention-guided deformable convolutional network for high dynamic range imaging," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2021, pp. 463–470.

**AN GIA VIEN** (Graduate Student Member, IEEE) received the B.S. degree in mathematics and computer science from the University of Science, Vietnam National University, Ho Chi Minh City, Vietnam, in 2015, and the M.S. degree in computer engineering from Pukyong National University, Busan, South Korea, in 2019. He is currently pursuing the Ph.D. degree with the Department of Multimedia Engineering, Dongguk University, Seoul, South Korea. His current research interests include image restoration, image enhancement, and high dynamic range imaging.

**CHUL LEE** (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea University, Seoul, South Korea, in 2003, 2008, and 2013, respectively.

From 2002 to 2006, he was with Biospace Inc., Seoul, where he involved in the development of medical equipment. From 2013 to 2014, he was a Postdoctoral Scholar with the Department of Electrical Engineering, Pennsylvania State University, University Park, PA, USA. From 2014 to 2015, he was a Research Scientist with the Department of Electrical and Electronic Engineering, The University of Hong Kong, Hong Kong. From 2015 to 2019, he was an Assistant Professor with the Department of Computer Engineering, Pukyong National University, Busan, South Korea. In March 2019, he joined the Department of Multimedia Engineering, Dongguk University, Seoul, where he is currently an Associate Professor. His current research interests include image processing and computational imaging with an emphasis on restoration and high dynamic range imaging.

Dr. Lee received the Best Paper Award from the *Journal of Visual Communication and Image Representation*, in 2014. He is an Editorial Board Member of the *Journal of Visual Communication and Image Representation*. He is also an APSIPA Distinguished Lecturer for the term 2023–2024.