Learning Rich Information for Quad Bayer Remosaicing and Denoising

Jun Jia[1], Hanchi Sun[1], Xiaohong Liu[2(✉)], Longan Xiao[3], Qihang Xu[3], and Guangtao Zhai[1(✉)]

[1] Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University, Shanghai, China
{jiajun0302,shc15522,zhaiguangtao}@sjtu.edu.cn
[2] John Hopcroft Center for Computer Science, Shanghai Jiao Tong University, Shanghai, China
xiaohongliu@sjtu.edu.cn
[3] Shanghai Transsion Information Technology, Shanghai, China
{longan.xiao1,qihang.xu}@transsion.com

Abstract. In this paper, we propose a DNNs-based solution to jointly remosaic and denoise the camera raw data in Quad Bayer pattern. The traditional remosaic problem can be viewed as an interpolation process that converts the Quad Bayer pattern to a normal CFA pattern, such as the RGGB one. However, this process becomes more challenging when the input Quad Bayer data is noisy. In addition, the limited amount of data available for this task is not sufficient to train neural networks. To address these issues, we view the remosaic problem as a bayer reconstruction problem and use an image restoration model to remove noises while remosaicing the Quad Bayer data implicitly. To make full use of the color information, we propose a two-stage training strategy. The first stage uses the ground-truth RGGB Bayer map to supervise the reconstruction process, and the second stage leverages the provided Image Signal Processor (ISP) to generate the RGB images from our reconstructed bayers. With the use of color information in the second stage, the quality of reconstructed bayers is further improved. Moreover, we propose a data pre-processing method including data augmentation and bayer rearrangement. The experimental results show it can significantly benefit the network training. Our solution achieves the best KLD score with one order of magnitude lead, and overall ranks the second in Quad Joint Remosaic and Denoise @ MIPI-challenge.

Keywords: Quad Bayer · Remosaicing · Denoising · Data augmentation

1 Introduction

In recent years, the increasing demand for the smartphone camera performance has accelerated the high imaging quality of image sensors for smartphones. One

© The Author(s), under exclusive license to Springer Nature Switzerland AG 2023
L. Karlinsky et al. (Eds.): ECCV 2022 Workshops, LNCS 13805, pp. 175–191, 2023.
https://doi.org/10.1007/978-3-031-25072-9_12

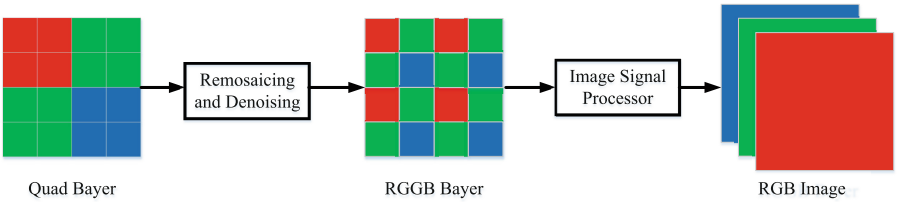

Fig. 1. The overall pipeline of this paper. The input of our model is a Quad Bayer map and the output is a RGGB Bayer map.

of the trends is the multi-pixel, which improves the image resolution by reducing the size of each pixel and arranging more pixels. However, there is a trade-off relationship between pixel miniaturization and decrease in sensitivity. For example, when capturing photos in low-illuminance environments, the weak sensitivities of sensors may decrease the imaging quality. Under this background, a new Color Filter Array (CFA) pattern called Quad Bayer is invented to achieve a good trade-off between the pixel size and the sensitivities of sensors. The Quad Bayer can minimize the decrease in the sensitivities of sensors even if the pixel size is small, which can improve the imaging quality under low light.

Compared to the normal CFA patterns, such as the RGGB Bayer pattern, the four adjacent pixels of a Quad Bayer are clustered with the same color filters. As shown in Fig. 1, the Quad Bayer data has three kinds of color filters and the pixels within a 2×2 neighborhood have the same color filters. The Quad Bayer has two modes for low and normal light. When capturing photos under low light, the binning mode enhances the sensitivities of sensors by averaging the four pixels within a 2×2 neighborhood, which can improve the imaging quality. As a tradeoff, the spatial resolution is halved. When capturing photos under normal light, the output bayer is supposed to have the same spatial resolution as the input Quad Bayer data. Thus, the original Quad Bayer data needs to be converted to a normal CFA pattern and then fed to the Image Signal Processor (ISP). This converting is an interpolation process called remosaic. The traditional remosaic algorithms are implemented based on hardware. For example, Sony Semiconductor Solutions Corporation (SSS) handles remosaic by installing an array conversion circuit on the image sensor chip[1]. Compared to hardware-based algorithms, software-based remosaic algorithms can be more flexibly applied to different devices. A good remosaic algorithm should be able to get the normal bayer output from the Quad Bayer data with least artifacts, such as moire pattern, false color, and so forth.

However, there are two challenges when designing a remosaic algorithm. The first challenge is that the remosaic problem is difficult when the input Quad Bayer data becomes noisy. Thus, the solution of jointly remosaicing and denoising is in demand for real-world applications. However, denoising and remosaicing are two separate tasks, which makes it difficult to combine them into one algorithm. To address this challenge, we view this process as a reconstruction problem from the

[1] https://www.sony-semicon.com/en/technology/mobile/quad-bayer-coding.html.

noisy Quad Bayer map to the clean RGGB Bayer map. Inspired by the success of deep neural networks (DNNs) in image reconstruction tasks, we propose to use a DNNs-based model to remove noises while implicitly rearranging the Quad Bayer map. We present the overall pipeline in real applications in Fig. 1. In addition, we propose a two-stage training strategy to make full use of the color information in both the bayer domain and the RGB domain. As shown in Fig. 2, the first stage uses the ground-truth RGGB Bayer maps to supervise the training of the reconstruction process. After that, the second stage applies the ISP provided by organizers to generate RGB images from the reconstructed bayers. This fine-tuning stage can further improve the quality of our reconstruction.

To train a robust DNNs-based model, we need sufficient training data. However, there is no public dataset that currently contains the paired noisy Quad Bayer data and clean RGGB Bayer data. Thus, the limited amount of data available for this task is the second challenge. Although the organizers of Quad Joint Remosaic and Denoise @MIPI-challenge provide a training set that includes the 210 paired noisy Quad Bayer data and clean RGGB Bayer data, they are not sufficient for training. To address this challenge, we propose a data pre-processing method that employs data augmentation and bayer rearrangement to expand and unify the training samples. The experimental results show this pre-processing can significantly benefit the network training.

After developing and validating our solution, we submit the trained model to Quad Joint Remosaic and Denoise @MIPI-challenge. Our solution is ranked second in the final test phase and achieves the best KLD score. To summarize, our contributions include:

- We propose a DNNs-based model with a two-stage training strategy to jointly remosaic and denoise for Quad Bayer data. By leveraging the two-stage training strategy, the model can make full use of the color information in both the bayer domain and the RGB domain, and the reconstruction quality can be further improved.
- We propose a data pre-processing method including data augmentation and bayer rearrangement. The experimental results show this pre-processing can significantly benefit the network training.
- We submit our solution to Quad Joint Remosaic and Denoise @MIPI-challenge. Our solution is ranked the second in the final test phase and achieves the best KLD score. Codes: https://github.com/jj199603/MIPI2022-QuadBayer

2 Related Works

2.1 Denoising

Image denoising has been greatly concerned in the past few decades. Currently, the image denoising can be classified as two categories, *i.e.,* traditional methods and deep learning based methods.

In traditional image denoising methods, image analysis and processing are usually based on transcendental images. Common methods are 3D transform-domain filtering (BM3D) [10], non-local means (NLM) [4], sparse coding [2], etc. The non-local similarity approach [4] used a non-local algorithm with shared similarity patterns, and the same strategy is applied to [15]. The application of image denoising was implemented using weighted nuclear norm minimization in [18], statistical properties of images were used to remove noise in [43], and scale mixtures of Gaussians were used in the wavelet domain for denoising in [41]. Dictionary learning methods [13] relied on sparse learning [33] from images to obtain a complete dictionary. Traditional denoising methods have certain denoising effectiveness based on reasonable use of image information but the traditional image denoising method is limited because it cannot be extended to all real scenes.

With the development and application of data-driven deep learning, image denoising is given a new processing method. A growing number of researchers are designing novel network architectures based on CNN and transformers, improving the accuracy and versatility of image denoising. [5] first introduced the multilayer perceptron (MLP) in image denoising and achieves the comparable performance to BM3D. To mimic the real images, many synthetic noise methods are proposed, such as Poissonian-Gaussian noise model [16], Gaussian mixture model [54], camera process simulation [46], and genetic algorithm [8]. Since then, several large physical noise datasets have been generated, such as DND [40] and SIDD [1].

In addition, the real image denoising method is also considered. Researchers first tried the methods previously applied to synthetic noisy datasets on real datasets with model adaptation and tuning [53]. Among them, AINDNet [24] adopted the transfer learning from comprehensive denoising to real denoising, and achieved satisfactory results.

The VDN [48] network architecture based on U-Net [42] was proposed, which assumed that the noise follows an inverse gamma distribution. However, the distribution of noise in the real world is often more complex, so this hypothesis did not apply to many application scenarios. Subsequently, DANet [50] abandoned the hypothesis of noise distribution and used the GAN framework to train the model. Two parallel branches were also used in the structure: one for noise removal and the other for noise generation. A potential limitation was that the training of the GAN-based model was unstable and thus took longer to converge [3]. DANet also used the U-Net architecture in the parallel branch.

Zhang *et al.* [53] recently proposed FFDNet, a denoising network using supervised learning, which connected noise levels as a mapping of noisy images and demonstrated the spatially invariant denoising of real noise with oversmoothed details. MIRNet [51] proposed a general network architecture for image enhancement, such as denoising and super-resolution, with many new building blocks that extracted, exchanged, and exploited multi-scale feature information. InvDN [31] transformed noisy inputs into low-resolution clean images and implicit representations containing noise. To remove noise and restore a clear image,

InvDN replaced noise implicit representation with another implicit representation extracted from previous distribution during noise reduction.

2.2 ISP and Demosaicing

A typical camera ISP assembly line uses a large number of image processing blocks to reconstruct sRGB images from raw sensor data. In another study [27], a two-stage depth network was proposed to replace the camera ISP. The entire ISP of the Huawei P20 smartphone was proposed to be replaced by a combination of deep mode and extensive global functional operations [21]. In [44], the authors proposed a CNN model to suppress image noise and exposure correction to the images captured by smartphone cameras.

Image demosaicing is considered as a low-level ISP task aimed at recreating the CFA pattern of RGB images. However, in practical applications, the image sensors can be affected by noises, which can also lead to the corruption of the final image reconstruction results during demosaicing [29]. In recent work, therefore, the focus has been on the need for a combination of demagnetization and denoising, rather than traditional sequential operations.

In the last four decades, signal processing methods have been widely used for the demosaicing problem [37,47] or resort to the frequency information to improve zipper effect [11,34]. Early methods used frequency method to design the aliasing-free filters [12]. In order to improve near-edge performance, a median chromatic aberration filter [20] was performed and a gradient based approach [39] was widely used. While many methods used chromatic aberrations, Monno recommends using color residuals, starting with a bilinear interpolation of the G channel and then improving the red and blue residuals [38].

However, traditional image processing method can not produce good image quality, easy to produce visual artifacts. More and more demosaicing methods exploit the technology of machine learning; see Energy-based Minimization in [25] or Heide's Complete ISP Modeling [19].

In the last five years, deep learning has become more and more important in low-level visual tasks [23,30,45] than human cognitive abilities. The work related to image demosaicing is summarized below.

Garbi *et al.* presented Bayer with the first end-to-end solution that combined noise reduction and demosaicing [17]. After receiving Bayer images, they extracted four RGGB channels and linked them to a estimated noise channel. The five channels were used as the low-resolution inputs for CNN. They then used a simple network structure similar to VDSR with stacked convolution and global residual paths. Before the final convolution, they also connected the original spliced Bayer plane to the feature mapping of the previous sample. Their main contribution is a data-driven approach to demosaicing the data and publishing a new training dataset by hard patch mining using HDR-VDP2 and moiré detection metric to detect artifact-prone patches [35].

Liu *et al.* [29] proposed an approach based on deep learning, supplemented by density maps and green channel guidance. In [26], the majority-minimization method were merged into a residual denoising network. A deep network was

trained using thousands of images to achieve state-of-the-art results [17]. In addition to these supervised learning methods, [14] attempted to address JDD through the unsupervised learning of large numbers of images.

The planar codec structure with symmetric skip connections (RED-Net) was proposed by Mao *et al.* in [36]. RED-Net used skip connections to connect encoder and counter encoder components, but their network is simple and not multi-resolution. They tried different depths of the network, including deeper ones.

Long *et al.* [32] proposed an image segmentation for a full convolutional networks, and the improved version had multi-scale features that captured the background of U-Net [42] at different resolutions. It is demonstrated that the U-Net architecture performed better than the DnCNN [22] network.

3 Proposed Method

In this Section, we first describe the details of the Quad Bayer pre-processing method including data augmentation and bayer rearrangement in Sect. 3.1. Then, the architecture of the model is presented in Sect. 3.2. Finally, we describe the proposed effective two-stage training strategy in Sect. 3.3.

3.1 Quad Bayer Pre-processing

Bayer Rearrangement. As shown in Fig. 2, a raw image in the Quad Bayer pattern consists of multiple 4×4 Quad Bayer units, and each Quad Bayer unit consists of 4 red units, 8 green units, and 4 blue units. Inspired by the success of the raw image processing methods [7,28], a Quad Bayer map is supposed to be decomposed into four channels: R channel, G1 channel, G2 channel, and B channel. For the convenience of channel decomposition, we first swap the second column and the third column in each 4×4 Quad Bayer unit and then swap the second row and the third row of each unit. After that, the Quad Bayer map is converted to a RGGB-alike bayer map and we decompose the RGGB-alike map into four channel maps that are the R channel, the G_1 channel, the G_2 channel, and the B channel, respectively. This process is presented in the first row of Fig. 2.

The above swapping process only includes simple spatial swapping of Quad Bayer units. Thus, the converted RGGB-alike maps still contain noises. Since the provided official ISP does not support a raw image in the Quad Bayer pattern as the input, we also apply this swapping process to the original Quad Bayer data containing noises for visualization in the remaining sections of this paper.

Data Augmentation. We use data augmentation to expand the training samples. The data augmentation processing is applied to the RGGB-alike map after the spatial swapping other than the original Quad Bayer map. In training, the horizontal flip, the vertical flip, and the transposition (permutation of rows and columns) are randomly applied to the RGGB-alike map. To maintain the RGGB

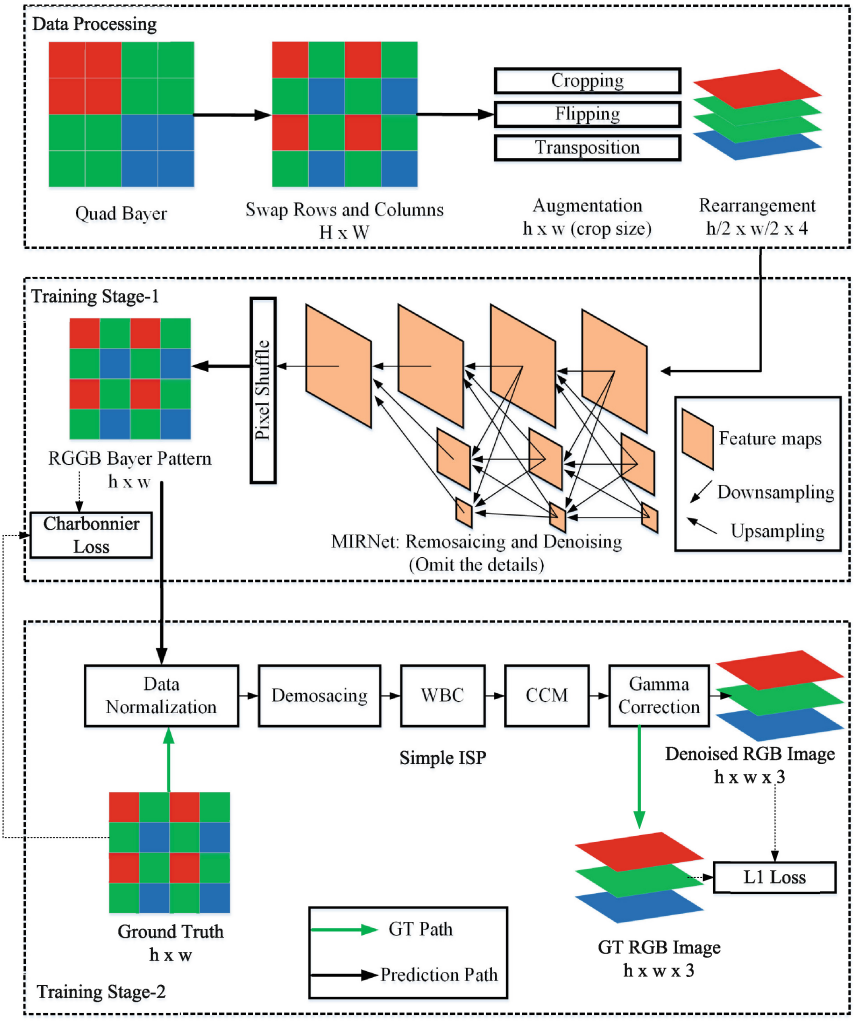

Fig. 2. The solution pipeline of this paper. In the data pre-processing stage, we use data augmentation to expand training samples and convert a Quad Bayer map to four channel RGGB maps. In the first stage, the ground-truth RGGB Bayer maps are used for supervision. In the second stage, the ground-truth RGB image generated by the ISP system is used for supervision. (Color figure online)

pattern after applying augmentations, a cropping-based post-processing method inspired by [28] is used to unify the CFA pattern after these augmentations.

– **vertical flip:** after vertically flipping, a RGGB-alike map is converted to a GBRG-alike map. If we directly decompose the GBRG-alike map into four channels, the channel order will be changed. Thus, we remove the first and last rows of the flipped GBRG-alike map to convert it to a new RGGB-alike map. The details of this process are presented in Fig. 3(a).

- **horizontal flip:** after horizontally flipping, a RGGB-alike map is converted to a GRBG-alike map. To unify the pattern for channel decomposition, we remove the first and last columns of the flipped GRBG-alike map to convert it to a new RGGB-alike map. The details of this process are presented in Fig. 3(b).
- **transposition:** the rows and the columns of the original RGGB-alike map are permuted after transposition, but the pattern is still RGGB. The traditional data augmentation methods also include rotations of 90°, 180°, 270°. However, rotations of 90°, 180°, and 270° can be obtained by a combination of flip and transposition. For instance, the clockwise rotation of 90° is equivalent to transposing and then flipping horizontally. Thus, transposition can be viewed as the elemental operation of rotation. The details of transposition are shown in Fig. 3(c).

In addition, to improve the training efficiency, we randomly crop the augmented RGGB-alike map to h × w. The starting coordinates of the cropping need to be even numbers to maintain the RGGB pattern.

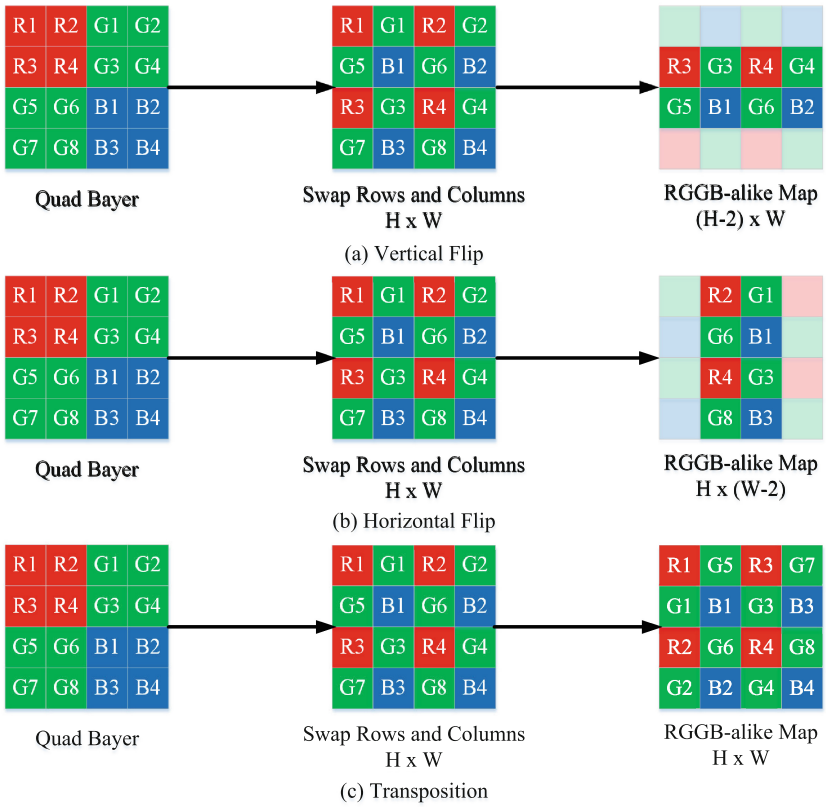

Fig. 3. The details of the data augmentations used in this paper.

3.2 Network for Jointly Remosaicing and Denoising

In this paper, we view the remosaicing problem as a reconstruction process from a Quad Bayer map to a RGGB Bayer map. Inspired by the success of image restoration methods based on deep neural networks [9,49,51,52], we use a DNNs-based neural network to remove noises while remosaicing the Quad pattern to the RGGB pattern implicitly.

We select MIRNet [51] to remove noises and reconstruct RGGB Bayer maps. MIRNet [51] consists of multi-scale feature extraction paths. Feature fusion is carried out among feature maps of different scales to fully learn the multi-scale features of the image. Compared to the models based on auto-encoder [9], MIRNet contains a high-resolution feature path where the feature maps are not downsampled, thus more detailed information is preserved. Through the feature fusion among the path of different scales, the high-level semantic features extracted with small resolutions and the low-level features extracted with large resolutions are fully fused and learned. MIRNet exploits multiple attention mechanisms to fuse the multi-scale features, such as spatial attention and channel attention. We present the overall architecture of this model in Fig. 2, omitting the details such as the attention modules.

The input of MIRNet is a $\frac{h}{2} \times \frac{w}{2} \times 4$ map generated from data augmentation and bayer rearrangement. The output of MIRNet is also a $\frac{h}{2} \times \frac{w}{2} \times 4$ map. We apply PixelShuffle operation to the output to generate a h $\times$ w $\times$ 1 RGGB map which represents the reconstructed RGGB Bayer data.

3.3 Two-Stage Training Strategy

We propose an effective two-stage training strategy to improve the reconstruction quality. As shown in Fig. 2, in the first stage, the model described in Sect. 3.2 is used to jointly remosaic the Quad Bayer data to the RGGB Bayer data and denoise it. The ground-truth RGGB Bayer map is used for supervision.

After the training in the first stage, we concatenate the ISP provided by the challenge organizers to the jointly remosaicing and denoising network. The provided ISP applies normalization, demosaicing, white balance correction, color correction, and gamma correction to a raw image. The output of our network and the ground-truth RGGB Bayer map are processed to generate the corresponding RGB images. Then, the generated ground-truth RGB image is used as a color supervision to finetune our network. In this stage, we freeze the parameters of ISP. Compared to the bayer data, the RGB image contains more color information, which can further improve the quality of the reconstructed raw data.

Loss Functions. In the first training stage, we use *Charbonnier* loss [6] to optimize the parameters of our model, which is defined as:

$$L_{Charbon} = \sqrt{||R_{rec} - R_{gt}||^2 + \epsilon^2} \tag{1}$$

where R_{rec} represents the reconstructed raw data in the RGGB Bayer pattern, R_{gt} represents the ground-truth RGGB Bayer map, and ϵ is hyper-parameter which is set to 10^{-3}. In the second training stage, we compute the L_1 loss between the RGB images generated from the reconstructed raw data and the ground-truth raw data.

4 Experimental Results

4.1 Dataset

The dataset used in experiments is provided by the organizers of the MIPI2022 challenge. The dataset in the development phase is divided into a training set and a validation set. Both the training set and the validation set include samples of three noise levels: 0 dB, 24 dB, and 42 dB. The training set of each noise level includes 70 Quad Bayer files and the corresponding ground-truth files in the RGGB pattern. The validation set of each noise level only includes 15 Quad Bayer files without the corresponding ground-truth files. Thus, the training set includes 270 Quad Bayer samples and the validation set includes 45 Quad Bayer samples. The resolutions of these samples are 1200×1800. The RGB thumbnails of these 85 raw images are also provided for the convenience of visualization.

In experiments, we use all the 270 training samples to train our model and evaluate our model on the 45 validation samples. Because the distribution of the noise model is unknown, no additional datasets are used for training to prevent overfitting.

4.2 Implementation Details

We trained three models for three noise levels (0 dB, 24 dB, and 42 dB), respectively. The training sample is the cropped raw data with a size of 256×256. The hyper-parameter ϵ of Charbonnier loss is set to 10^{-3}. We use Adam to optimize the network. The initial learning rate is 2×10^{-4}, decaying by 1/10 every 235,200 iterations. The batch size of training is 2 and the average training time of one model is about 36 h in one NVIDIA-2080 Ti. When validating and testing, original Quad Bayer data without any cropping and the output of the model is the corresponding RGGB Bayer map that has the same resolution as the input.

4.3 Evaluation Metrics

The official metrics used by Quad Joint Remosaic and Denoise @MIPI-challenge are Peak Signal To Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), Learned perceptual image patch similarity (LIPIS), and KL divergence (KLD). The final results and rankings are evaluated by the M_4 score:

$$M_4 = PSNR \times SSIM \times 2^{1-LPIPS-KLD} \tag{2}$$

In the development and validation phase, we evaluate our model on the validation set. Since the ground-truth RGGB Bayer maps of the validation set are

not provided, we can only compute the accurate M_4 scores through the official website. However, the submission number is limited, we cannot compute the accurate M_4 for all experimental results. For the convenience of development and validation, we use PSNR to evaluate some of the experiments.

4.4 Ablation Study

In this section, we analyze the effects of the data augmentation methods and the proposed two-stage training strategy.

Ablation Study of Data Augmentations. To analyze the importances of the data augmentation methods, we first train three models without data augmentations for three noise levels and then train three models with the data augmentations described in Sect. 3.1. During training, we randomly select one augmentation from vertical flips, horizontal flips, and transpositions with a 25% probability, and do not apply any augmentations with a 25% probability. The ablation results on validation set are presented in Table 1. Table 1 shows that using data augmentations can improve the scores of PSNR, SSIM, and LPIPS.

Table 1. The ablation results of data augmentations

Model	PSNR	SSIM	LPIPS	KLD	M4
Without aug	36.24	0.954	0.127	0.00496	64.18
With aug	**36.65**	**0.956**	**0.121**	**0.00628**	**65.15**

Ablation Study of Two-Stage Training Strategy. To analyze the importance of the data augmentation methods, we first train three models for the three noise levels without the second fine-tuning stage. When the PSNR values converge on the validation set, we fine tune these three models on base of the first stage. We use PSNR to evaluate the improvement of the second stage. For 0 dB, the PSNR value increases from 40.91 to 41.17. For 24 dB, the PSNR value increases from 36.22 to 36.43. For 42 dB, the PSNR value increases from 32.27 to 32.36. These results show that fine-tuning the model with RGB images can further improve the visual quality of the reconstructed images. Compared to the raw data, RGB images contain richer color information since the channel number is three times that of raw images.

4.5 Model Complexity and Runtime

The total number of trainable parameters is 31,788,571. When validating and testing, the resolution of the input is 1200×1800 and the average test time in one NVIDIA-2080 Ti is about 0.7 s. Although we use a single model for jointly remosaicing and denoising, our model is not light-weighted, but can be further optimized to adapt real-time application.

Table 2. The results and rankings of Quad Joint Remosaic and Denoise @MIPI-challenge

Rank	Team	PSNR	SSIM	LPIPS	KLD	M4
1	op-summer-po	**37.93**	**0.965**	0.104	0.019	**68.03**
2	**JHC-SJTU (ours)**	37.64	0.96	0.1	**0.0068**	67.99
3	IMEC-IPI & NPU	37.76	0.96	0.1	0.014	67.95
4	BITSpectral	37.2	0.96	0.11	0.03	66
5	HITZST01	37.2	0.96	0.11	0.06	64.82
6	MegNR	36.08	0.95	**0.095**	0.023	64.1

Fig. 4. The qualitative results of normal brightness images on Quad Joint Remosaic and Denoise @MIPI-challenge validation set.

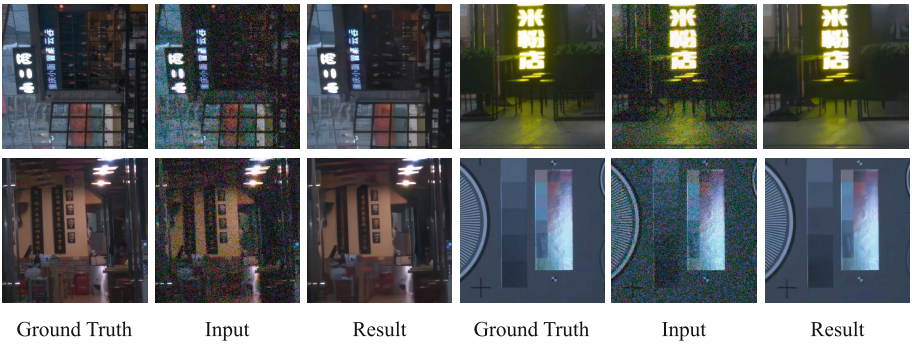

Fig. 5. The qualitative results of low brightness images on Quad Joint Remosaic and Denoise @MIPI-challenge validation set.

4.6 Challenge Submission

After validating the proposed solution, we submit the trained model in Quad Joint Remosaic and Denoise @MIPI-challenge. The dataset in the final test phase includes 15 Quad Bayer files and the resolution of each test sample is 1200×1800. The ranking of our solution is the second as shown in Table 2. Table 2 shows that

Fig. 6. The qualitative results of text region reconstruction on Quad Joint Remosaic and Denoise @MIPI-challenge validation set.

Fig. 7. The qualitative results of text region reconstruction on Quad Joint Remosaic and Denoise @MIPI-challenge test set.

our solution achieves the best KLD score. We present some qualitative results in Fig. 4, Fig. 5, Fig. 6, and Fig. 7. Figure 4 shows the representative reconstruction results of normal brightness images on Quad Joint Remosaic and Denoise @MIPI-challenge validation set, Fig. 5 shows the representative results of low brightness images, Fig. 6 shows the representative results of text regions, and Fig. 7 shows the results on the test set. The noise level of these presented examples is 42 dB.

4.7 Limitations

In experiments, we find that our model has limitations in two scenes. The first scene is the low brightness image as shown in Fig. 8(a)–(c). The second scene is

the texture region containing the high noise level as shown in Fig. 8(d), which suffers color errors in the reconstruction result.

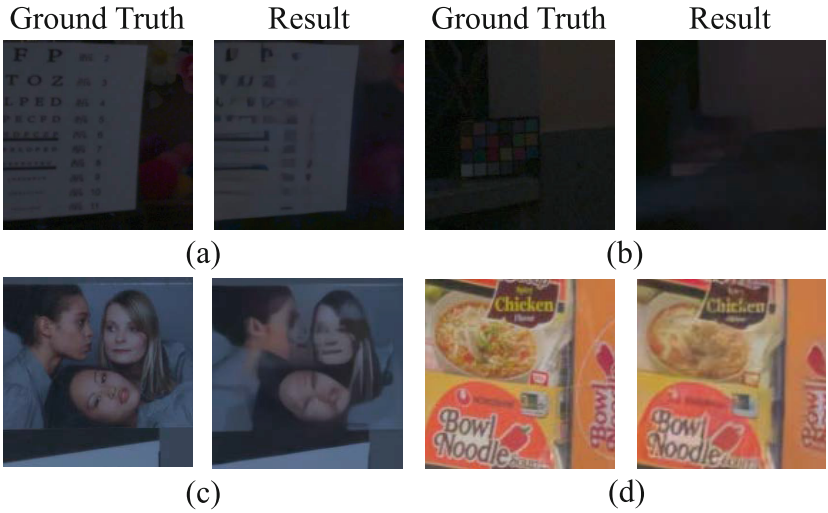

Fig. 8. The failure examples on Quad Joint Remosaic and Denoise @MIPI-challenge validation set.

5 Conclusions

The paper proposes a novel solution of jointly remosaicing and denoising for the camera raw data in the Quad Bayer pattern. We use a DNNs-based multi-scale model to remove noises while remosaicing the Quad Bayer map to the RGGB Bayer map. An effective data pre-processing method is proposed to augment and rearrange the original Quad Bayer data. To make full use of the color information, we propose a two-stage training strategy to fine-tune the model with the corresponding RGB images. The experimental results show that the data pre-processing method and the two-stage training strategy can significantly benefit the network training. We submit our solution to Quad Joint Remosaic and Denoise @MIPI-challenge, and achieve the second rank and the best KLD scores on the final test set. Our solution still has three limitations at this stage: (1) the reconstruction quality of low brightness images needs to be improved, (2) the reconstruction quality is not good enough for high-level noises, and (3) our model has a slightly large number of parameters, which needs to be further optimized for real-time applications. We will solve these problems in the future.

Acknowledgements. This work was supported by the National Key R&D Program of China 2021YFE0206700, NSFC 61831015.

References

1. Abdelhamed, A., Lin, S., Brown, M.S.: A high-quality denoising dataset for smartphone cameras. In: IEEE/CVF Conference on Computer Vision & Pattern Recognition (2018)
2. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: an algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. Sig. Process. **54**(11), 4311–4322 (2006). https://doi.org/10.1109/TSP.2006.881199
3. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International Conference on Machine Learning (2017)
4. Buades, A., Coll, B., Morel, J.M.: A non-local algorithm for image denoising. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005 (2005)
5. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: can plain neural networks compete with BM3D? In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
6. Charbonnier, P., Blanc-Feraud, L., Aubert, G., Barlaud, M.: Two deterministic half-quadratic regularization algorithms for computed imaging. In: Proceedings of 1st International Conference on Image Processing, vol. 2, pp. 168–172 (1994). https://doi.org/10.1109/ICIP.1994.413553
7. Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3291–3300 (2018)
8. Chen, J., Chen, J., Chao, H., Ming, Y.: Image blind denoising with generative adversarial network based noise modeling. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
9. Cheng, S., Wang, Y., Huang, H., Liu, D., Fan, H., Liu, S.: NBNet: noise basis learning for image denoising with subspace projection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 4896–4906 (2021)
10. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Trans. Image Process. **16**(8), 2080–2095 (2007)
11. Dai, L., Liu, X., Li, C., Chen, J.: AWNet: attentive wavelet network for image ISP. In: Bartoli, A., Fusiello, A. (eds.) ECCV 2020. LNCS, vol. 12537, pp. 185–201. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-67070-2_11
12. Alleysson, D., Süsstrunk, S., Hérault, J.: Linear demosaicing inspired by the human visual system. IEEE Trans. Image Process. **14**(4), 439–449 (2005)
13. Dong, W., Xin, L., Lei, Z., Shi, G.: Sparsity-based image denoising via dictionary learning and structural clustering. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2011)
14. Ehret, T., Davy, A., Arias, P., Facciolo, G.: Joint demosaicking and denoising by fine-tuning of bursts of raw images. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019)
15. Foi, A., Katkovnik, V., Egiazarian, K.: Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images. IEEE Trans. Image Process. **16**, 1395–1411 (2007)
16. Foi, A., Trimeche, M., Katkovnik, V., Egiazarian, K.: Practical Poissonian-Gaussian noise modeling and fitting for single-image raw-data. IEEE Trans. Image Process. **17**(10), 1737–1754 (2008)

17. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Trans. Graph. **35**(6), 191 (2016)
18. Gu, S., Lei, Z., Zuo, W., Feng, X.: Weighted nuclear norm minimization with application to image denoising. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
19. Heide, F., et al.: FlexISP: a flexible camera image processing framework. In: International Conference on Computer Graphics and Interactive Techniques (2014)
20. Hirakawa, K., Parks, T.W.: Adaptive homogeneity-directed demosaicing algorithm. IEEE Trans. Image Process. **14**(3), 360 (2005)
21. Ignatov, A., Gool, L.V., Timofte, R.: Replacing mobile camera ISP with a single deep learning model. In: Computer Vision and Pattern Recognition (2020)
22. Kai, Z., Zuo, W., Chen, Y., Meng, D., Lei, Z.: Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising. IEEE Trans. Image Process. **26**(7), 3142–3155 (2016)
23. Kim, I., Song, S., Chang, S., Lim, S., Guo, K.: Deep image demosaicing for submicron image sensors. J. Imaging Sci. Technol. **63**(6), 060410-1–060410-12 (2019)
24. Kim, Y., Soh, J.W., Gu, Y.P., Cho, N.I.: Transfer learning from synthetic to real-noise denoising with adaptive instance normalization. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
25. Klatzer, T., Hammernik, K., Knobelreiter, P., Pock, T.: Learning joint demosaicing and denoising based on sequential energy minimization. In: 2016 IEEE International Conference on Computational Photography (ICCP) (2016)
26. Kokkinos, F., Lefkimmiatis, S.: Deep image demosaicking using a cascade of convolutional residual denoising networks. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) Computer Vision – ECCV 2018. LNCS, vol. 11218, pp. 317–333. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01264-9_19
27. Liang, Z., Cai, J., Cao, Z., Zhang, L.: CameraNet: a two-stage framework for effective camera ISP learning. IEEE Trans. Image Process. **30**, 2248–2262 (2019)
28. Liu, J., et al.: Learning raw image denoising with bayer pattern unification and bayer preserving augmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (2019)
29. Liu, L., Jia, X., Liu, J., Tian, Q.: Joint demosaicing and denoising with self guidance. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
30. Liu, X., Shi, K., Wang, Z., Chen, J.: Exploit camera raw data for video super-resolution via hidden Markov model inference. IEEE Trans. Image Process. **30**, 2127–2140 (2021)
31. Liu, Y., et al.: Invertible denoising network: a light solution for real noise removal (2021)
32. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440 (2015)
33. Mairal, J., Bach, F., Ponce, J., Sapiro, G., Zisserman, A.: Non-local sparse models for image restoration. In: 2009 IEEE 12th International Conference on Computer Vision (ICCV) (2010)
34. Malvar, H.S., He, L.W., Cutler, R.: High-quality linear interpolation for demosaicing of bayer-patterned color images. In: 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (2004)
35. Mantiuk, R., Kim, K.J., Rempel, A.G., Heidrich, W.: HDR-VDP-2: a calibrated visual metric for visibility and quality predictions in all luminance conditions. ACM Trans. Graph. **30**(4), 1–14 (2011)

36. Mao, X.J., Shen, C., Yang, Y.B.: Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections (2016)
37. Menon, D., Calvagno, G.: Color image demosaicking. Sig. Process. Image Commun. **26**(8–9), 518–533 (2011)
38. Monno, Y., Kiku, D., Tanaka, M., Okutomi, M.: Adaptive residual interpolation for color image demosaicking. In: IEEE International Conference on Image Processing (2015)
39. Pekkucuksen, I., Altunbasak, Y.: Gradient based threshold free color filter array interpolation. In: 2010 17th IEEE International Conference on Image Processing (ICIP) (2010)
40. Plotz, T., Roth, S.: Benchmarking denoising algorithms with real photographs, pp. 2750–2759 (2017)
41. Portilla, J., Strela, V., Wainwright, M.J., Simoncelli, E.P.: Image denoising using scale mixtures of Gaussians in the wavelet domain. IEEE Trans. Image Process. **12**(11), 1338–1351 (2003)
42. Ronneberger, O., Fischer, P., Brox, T.: U-Net: convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) MICCAI 2015. LNCS, vol. 9351, pp. 234–241. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24574-4_28
43. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. Physica D **60**(1–4), 259–268 (1992)
44. Schwartz, E., Giryes, R., Bronstein, A.M.: DeepISP: towards learning an end-to-end image processing pipeline. IEEE Trans. Image Process. **28**(2), 912–923 (2018)
45. Sharif, S., Naqvi, R.A., Biswas, M.: Beyond joint demosaicking and denoising: an image processing pipeline for a pixel-bin image sensor (2021)
46. Shi, G., Yan, Z., Kai, Z., Zuo, W., Lei, Z.: Toward convolutional blind denoising of real photographs (2018)
47. Xin, L., Gunturk, B., Lei, Z.: Image demosaicing: a systematic survey. In: Proceedings of SPIE - The International Society for Optical Engineering, vol. 6822 (2008)
48. Yue, Z., Yong, H., Zhao, Q., Zhang, L., Meng, D.: Variational denoising network: toward blind noise modeling and removal (2019)
49. Yue, Z., Yong, H., Zhao, Q., Meng, D., Zhang, L.: Variational denoising network: toward blind noise modeling and removal. In: Advances in Neural Information Processing Systems, vol. 32 (2019)
50. Yue, Z., Zhao, Q., Zhang, L., Meng, D.: Dual adversarial network: toward real-world noise removal and noise generation. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12355, pp. 41–58. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58607-2_3
51. Zamir, S.W., et al.: Learning enriched features for real image restoration and enhancement. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12370, pp. 492–511. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58595-2_30
52. Zamir, S.W., et al.: Learning enriched features for fast image restoration and enhancement (2022)
53. Zhang, K., Zuo, W., Zhang, L.: FFDNet: toward a fast and flexible solution for CNN based image denoising. IEEE Trans. Image Process. **27**(9), 4608–4622 (2017)
54. Zhu, F., Chen, G., Heng, P.A.: From noise modeling to blind image denoising. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)