Deep Demosaicing using ResNet-Bottleneck Architecture

Divakar Verma, Manish Kumar and Srinivas Eregala

November 16, 2019

Deep Demosaicing using ResNet-Bottleneck Architecture

Divakar Verma, Manish Kumar, and Srinivas Eregala

Samsung R&D Institute Bengaluru, India

Abstract. Demosaicing is a fundamental step in a camera pipeline to construct a full RGB image from the bayer data captured by a camera sensor. The conventional signal processing algorithms fail to perform well on complex-pattern images giving rise to several artefacts like Moire, color and Zipper artefacts. The proposed deep learning based model removes such artefacts and generates visually superior quality images. The model performs well on both the sRGB (standard RGB color space) and the linear datasets without any need of retraining. It is based on Convolutional Neural Networks (CNNs) and uses a residual architecture with multiple 'Residual Bottleneck Blocks' each having 3 CNN layers. The use of 1x1 kernels allowed to increase the number of filters (width) of the model and hence, learned the inter-channel dependencies in a better way. The proposed network outperforms the state-of-the-art demosaicing methods on both sRGB and linear datasets.

Keywords: Demosaicing, RGB, bayer, Moire artefacts, CNN, Residual Bottleneck architecture

1 Introduction

De-mosaicing is the first and the foremost step of any camera ISP (Image Signal Processing) pipeline. Color image sensor can only capture one color at any pixel location in a fixed bayer pattern forming a mosaic/bayer image. An interpolation method is needed to fill the missing colors at each pixel location in the mosaiced image and this process is known as De-mosaicing. A common challenge faced for demosaicing is the unavailability of the actual ground truth images where each pixel contains the actual R (red), G (green) and B (blue) components. It is not feasible to capture all the color components at any given pixel location. So, the common approach is to take high quality images and treat them as the ground truth. These images are then mosaiced into bayer images which goes as an input to the demosaicing algorithm.

Traditional interpolation algorithms take advantage of correlation between R, G and B components of bayer image. Since G component has double sampling frequency, interpolation of G is done first, followed by R and B. Interpolation is done along both horizontal and vertical direction and combined using various metrics. In MSG [1] algorithm, authors improved the interpolation accuracy by using Multi-Scale color Gradients to adaptively combine color-difference-estimates from different directions. In ARI (Adaptive Residual Interpolation) [2], authors used R as a guided filter to interpolate G at R&B (guided upsampling) and vice versa to interpolate R&B at G. Due to inherent sensor noise, interpolation based algorithm sometimes fails to demosaic the complicated patterns near the edges, leading to moire, zippering and other color artefacts. To remove the moire artefact from images, camera uses low pass filter but that reduces the sharpness of the image. To address these challenges, deep learning algorithms have been proposed which show significant improvement over traditional interpolation based methods.

1.1 Related Work

Numerous deep learning architectures have been proposed for demosaicing and with the advancements in the processing power, the networks are becoming deeper and deeper. The authors of 'A Multilayer Neural Network for Image Demosaicing' [3] had proposed a 3 layered deep network which achieved a PSNR (Peak signal-to-noise ratio) of 36.71 on 19 Kodak images and showed initial promise that deep learning network could prove to be useful for demosaicing. Gharbi et al [4] uses a 15 layered network with a residual learning approach. It was able to outperform all the interpolation based demosaicing methods and deep learning based networks by achieving 41.2 PSNR on Kodak dataset [5]. Runjie Tan et al [6] uses a two stage network which is similar to interpolation algorithms such as MSG and AHD [7]. The Green channel is used as a guide for interpolation of Red and Blue channels. First, the demosaicing kernels are learned using the L2 loss [8] on Green channel and then in the second stage, the loss is calculated on all channels. Thus, the Green channel guides the interpolation of the final RGB channels. On Kodak-24 image dataset, it achieved a PSNR of 42.04 and on McMaster (McM) [9] dataset, it achieved a PSNR of 39.98. The network proposed in DMCNN-VD [10] is even deeper and consists of 20 convolutional layers. It also uses a residual learning approach and achieved a PSNR of 42.27 on the Kodak-24 dataset.

However, the above mentioned deep learning networks do not generalize well on all kind of images and hence, will require a re-training for the specific kind of images. The proposed deep learning architecture addresses these issues and outperforms the state-of-the-art deep learning based demosaicing network on both linear and sRGB datasets. For the first time, a bottleneck residual network for demosaicing has been proposed which can generalize across different types of datasets. The proposed network is a fully convolutional neural network and uses multiple residual blocks.

2 Proposed Deep CNN Architecture

The proposed bottleneck residual network architecture for demosaicing generalizes well and generates superior quality images with minimal artefacts. The

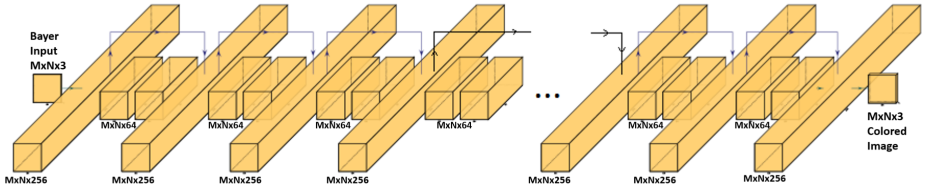


Fig. 1: Proposed Deep Learning model for demosaicing with 10 residual blocks

network is inspired from Residual Network (ResNet) architecture [11]. The proposed network is able to handle the complicated patterns in the image and gives much better visual quality.

The proposed network is based on CNNs and uses a residual architecture with each residual block having a bottleneck structure [12]. The network has 10 such residual blocks each having 3 convolutional layers. The network has a varying width of 256 and 64 channels. The bottleneck structure allows faster learning and at the same time learns more number of features.

The input to the network is a bayer image which is split into 3 channels - Red(R), Green(G) and Blue(B), with each channel having interleaved zeros. The starting convolutional layer in the network uses 3x3 filters and converts the dimensions of the 3-channeled bayer input to 256 channels that goes as an input to the first residual block. Each residual block has 3 CNN layers. The first layer uses 1x1 filters to change the dimension of 256-channeled input from the previous residual block and convert it to a 64-channeled output. This output is then passed through a ReLU activation layer. The second CNN layer operates on a reduced dimensional output of 64 channels from the previous layer. This layer uses a filter size of 3x3 which helps the model to learn important features and interchannel relationships. The output from this layer is 64-channeled and is passed through a ReLU activation layer. The third CNN layer uses 1x1 filters to restore the dimensions from 64 to 256 channels. Using a skip connection, the output from the third CNN layer is added with the original input (256-channeled) of the given residual block. This output now goes as an input to the next residual block. After the 10th (last) residual block, the final convolutional layer of the network uses 3x3 filters and converts the output having 256 channels into a 3-channeled color image. This is the final output of the network and has the same dimensions as of the input bayer image. Fig. 1. shows the proposed network architecture. The network uses an L2 loss function between the ground truth and the output of the model.

Fig. 2 shows few possibilities of different input bayer images possible for the network. The input is generated from the ground truth RGB image by mosaicing it in a bayer fashion. The basic form is shown in (a) which is a single channeled image with all the three color components interleaved in the same plane. This form is generally not preferred as an input to the network because it adds an additional burden on the network to learn the relationship between

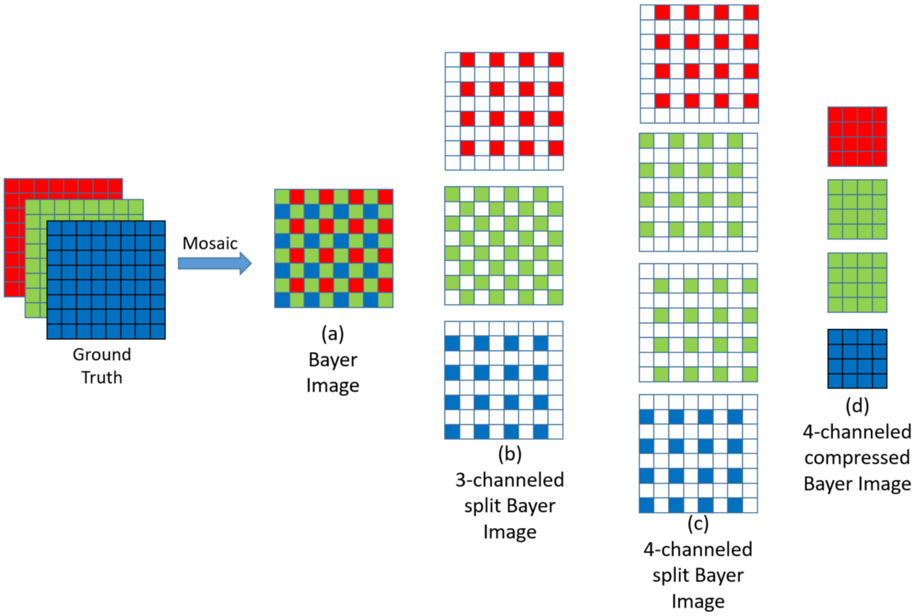

Mosaic
Ground Truth
(a)
Bayer Image
(b)
3-channeled split Bayer Image
(c)
4-channeled split Bayer Image
(d)
4-channeled compressed Bayer Image

Fig. 2: Comparison of different possible forms of the input image to the network

the interleaved color components. For example, the network needs to learn that two alternate pixels belong to the same color channel. So, a common approach is to split the color components into different channels. The proposed architecture uses a 3-channeled bayer image as shown in (b). The interleaved white blocks in the channels are the places where no color component is present and have been initialized with zeroes. The Green channel contains 50% of the color components whereas the Red and Blue contains 25% each. For uniformity, the Green channel can further be split into two channels, as shown in (c), so that each channel contains 25% of the color components. This approach was not adopted because it would have increased the training parameters and made the model more complex. The four channels shown in (c), can be compressed by packing the color pixels together, as shown in (d). This would lead to the loss of spatial information of the pixels and hence make it difficult for the network to learn some important information, like edges, which is of utmost priority for demosaicing. Hence, form (b) was chosen as the input for the proposed network.

The proposed model was trained solely on sRGB dataset and still it is able to generalize well across linear dataset. Due to the limited availability of linear datasets, the model was not trained on the linear dataset. So, to test the model on linear datasets, the images were transformed to sRGB domain and demosaiced using the network already trained on the sRGB dataset. During the experiment, it was found that the model performed equally well for the linear dataset.

Table 1: Comparison of BottleNeck architectures with different widths

	Kodak12	McM	Kodak24	Panasonic	Canon
128-64	43.8	39.28	42.24	42.34	44.41
256-64	**43.86**	**39.29**	**42.3**	**42.42**	**44.43**

To confirm the role of the width of the architecture, the model was tested with a modified version of the architecture having a smaller width of 128 instead of 256. Table 1 shows the results of the experiment on different datasets. Panasonic and Canon are the linear datasets of Microsoft Demosaicing Dataset (MDD) [13] while the rest of them are sRGB datasets. The first row shows the results of the architecture having widths of 128 and 64. The second row shows the results of the proposed model having widths of 256 and 64. It can thus be confirmed that, more number of channels (width) helps the network to learn more number of features required for demosaicing. Hence, increasing the width of the network improves the quality of demosaiced image.

3 Experiments and Results

In all the mentioned experiments, Bayer color filter array was used, as it is the most commonly and widely used color filter array in cameras. The network was trained on Waterloo Exploration Dataset (WED) [14] dataset which contains 4,744 colored images of roughly 600x400 resolution. The dataset was augmented by shifting 1 pixel along horizontal and vertical direction, all four rotations and flipping. Shifting an image by 1 pixel helps to capture all the color components at any given pixel location when mosaicing the ground truth image into bayer image. Rotations and flipping helps to generate different orientations of the same image and helps the network to learn a wide variety of patterns and orientations. Finally, image patches of size 128x128 was cropped from this augmented dataset for training. Total number of training images generated was 735,920.

Table 2: PSNR comparison for sRGB dataset

	Kodak-12	McM	Kodak-24
MSG	NA	NA	41.00
ARI	41.47	37.60	NA
DMCNN-VD	43.45	39.54	42.27
Gharbi	41.2	39.5	NA
Tan	NA	38.98	42.04
Kokkinos [15]	41.5	39.7	NA
MMNet [16]	42.0	**39.7**	NA
Proposed	**43.86**	39.29	**42.30**

Table 2. shows the quantitative comparison on sRGB datasets. Kodak-12 and Kodak-24 are the sets of 12 and 24 Kodak images respectively. Different authors have used different Kodak sets to measure the performance. We have compared our results on both the Kodak datasets. The proposed model outperforms other algorithms on Kodak sets. In case of McMaster (McM) dataset, the results are not far behind. Table 3. shows the quantitative comparison on MDD. The proposed method outperforms other demosaicing algorithms and is the state-of-the-art. Note that the PSNR 42.86 achieved is the weighted average of 200 Panasonic and 57 Canon images.

Table 3: PSNR comparison for linear (MDD) dataset

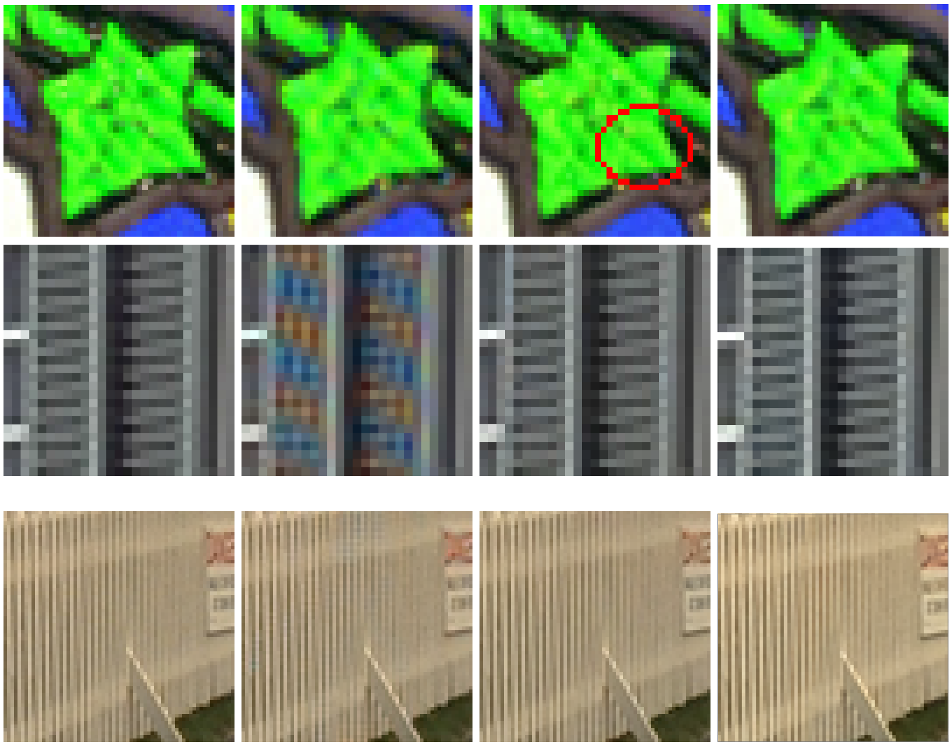
ARI	RTF [17]	DMCNN-VD	Kokkinos	Gharbi	MMNet	Proposed
39.94	39.39	41.35	42.6	42.7	42.8	**42.86**

Table 4. shows the comparison for two networks with widths of 256 and 128 for linear dataset. In the table, the first row (128-64) refers to the bottleneck architecture with widths 128 and 64. Similarly, 256-64 refers to the bottleneck architecture with widths 256 and 64. The prefix 'lin_sRGB' refers to the method where the testing linear images were first converted to sRGB domain, then demosaiced and finally converted back to linear domain to find the PSNR values. The data clearly shows that the network with 256-width outperforms the 128-width network in both linear and sRGB domain demosaicing.

Table 4: PSNR Comparison of bottleneck architecture on linear datasets

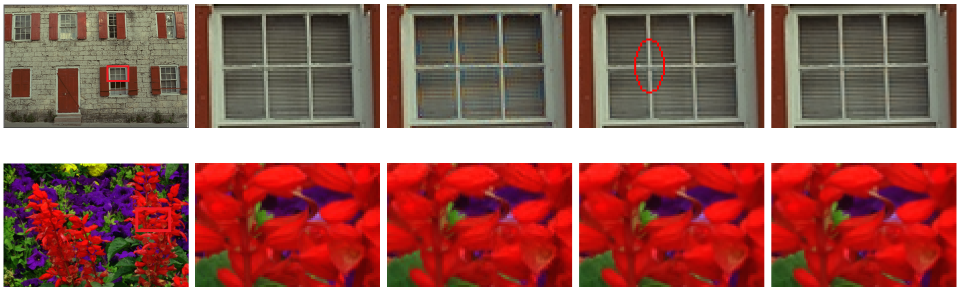
	Panasonic(200)	Canon(57)
128-64	41.92	44.05
128-64 lin_sRGB	42.14	44.41
256-64	41.94	44.07
256-64 lin_sRGB	**42.42**	**44.43**

Fig. 3. and Fig. 4. shows the qualitative comparison on sRGB datasets. In Fig. 3. top row image (green star), it can be observed that DMCNN-VD fails to produce sharp edges inside the marked region. In Fig. 4, a blue-colored artefact can be observed in the marked region when looked closely which is absent in the proposed image . Fig. 5. shows the qualitative results on linear MDD dataset. First two images (a,b) are Ground Truth and the proposed method's output. Next three images (c,d,e) are snapshots taken directly from the DMCNN-VD paper. The authors have increased the saturation and brightness for these images to highlight the chroma artefacts. The proposed model is not fine-tuned using any linear dataset and even then, it is able to match the visual quality of DMCNN-VD-Tr, which is a fine tuned version of DMCNN-VD on MDD dataset using



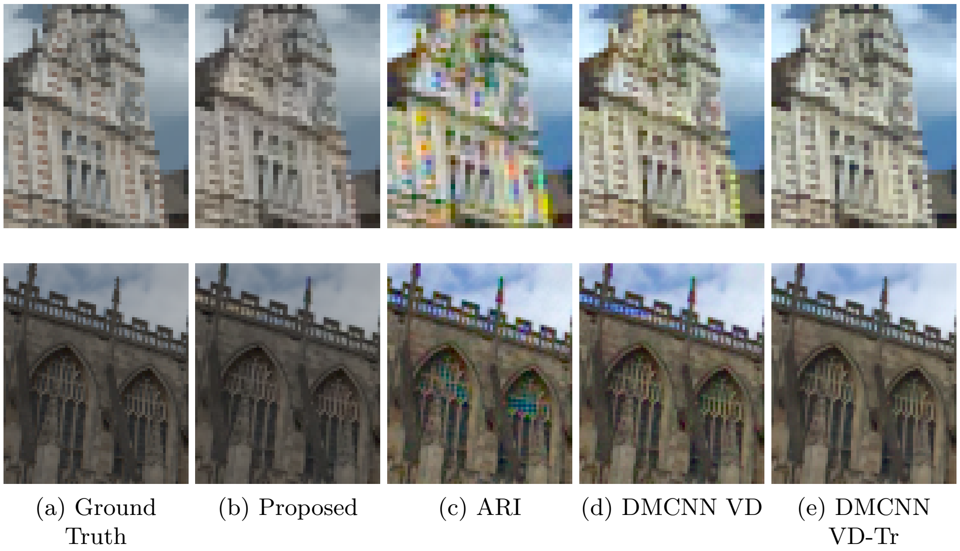
(a) GroundTruth (b) ARI (c) DMCNN-VD (d) Proposed

Fig. 3: Visual comparison with ARI and DMCNN-VD



(a) Original Img (b) GroundTruth (c) ARI (d) Tan (e) Proposed

Fig. 4: Visual comparison with ARI and Tan on Kodak (Top row) and McM (Bottom row) datasets



(a) Ground Truth (b) Proposed (c) ARI (d) DMCNN VD (e) DMCNN VD-Tr

Fig. 5: Visual comparison on Linear Images (MDD dataset)

transfer learning. In the top row, the DMCNN-VD-Tr output appears to have lost the chroma information for the monument but the proposed model preserves the color. The proposed model also outperforms DMCNN model in terms of PSNR metric, as shown in Table 3.

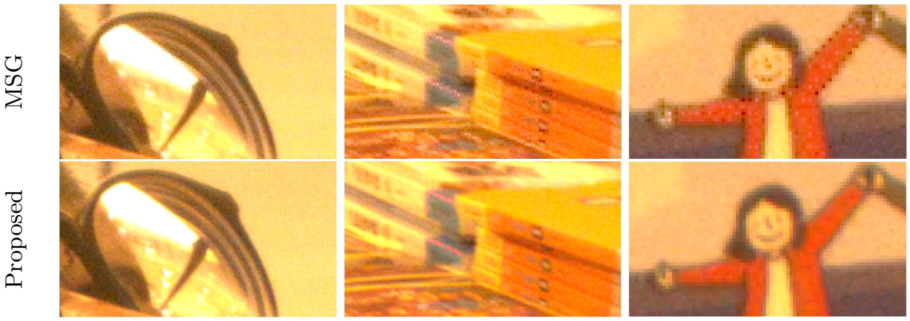


Fig. 6: Demosaicing on raw data captured from a smartphone

The proposed method was also tested on a real-life image dataset. Fig. 6. shows demosaicing algorithms applied on the raw images captured at 12 MP by a smartphone and it can be seen that the proposed model has generalized well. Random noise and zipper artefacts can be clearly seen on MSG demosaiced images. The proposed model minimizes all such artefacts.

4 Conclusion and Future Work

In this paper, a novel approach for demosaicing has been proposed. The proposed method is the state-of-the-art and confirms the ability to generalize well, across different types of datasets. Most of the computational photography techniques and computer vision algorithms rely on edge detection. Images with artefacts on the edges such as zippering and chroma are likely to give poor segmentation results, thus, further affecting the processed image. Therefore, it is crucial to solve such issues at the very start of the Image Processing Pipeline. With a superior quality at the initial steps of the camera pipeline, it is expected that further processing blocks will perform better and the final output will be much more appealing and free from artefacts. Also, camera image enhancement solutions such as low-light imaging and super resolution, rely heavily upon per pixel quality. It is expected that the proposed method, which has minimal artefacts, will directly benefit these solutions.

The future work involves exploring the effects of demosaicing algorithms on the computational photography solutions like HDR and Super-Resolution and evaluate the extent to which the proposed demosaicing algorithm improves these solutions. Along with that, the next focus will be to explore the capability of the proposed network to handle simultaneous demosaicing and denoising. Demosaicing and denoising is a tightly coupled problem, solving one greatly affects the other. A wide research is going on to address both of them simultaneously and many deep learning architectures have been proposed. Additionally, it will be explored if such a network can be compressed and optimized for an on-device ISP pipeline without significant loss in performance.

References

1. Pekkucuksen, I.; Altunbasak, Y. Multiscale gradients-based color filter array interpolation. IEEE Trans. Image Process. 2013, 22, 157–165.
2. Y. Monno, D. Kiku, M. Tanaka, and M. Okutomi, “Adaptive residual interpolation for color image demosaicking,” in Proceedings of IEEE ICIP 2015, 2015, pp. 3861–3865.
3. Wang, Y.Q., 2014, October. A multilayer neural network for image demosaicking. In 2014 IEEE International Conference on Image Processing (ICIP) (pp. 1852-1856). IEEE.
4. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Transactions on Graphics (TOG), 35(6), p.191 (2016).
5. Kodak Dataset. http://r0k.us/graphics/kodak
6. Tan, R., Zhang, K., Zuo, W., Zhang, L.: Color image demosaicking via deep residual learning. In IEEE Int. Conf. Multimedia and Expo (ICME) (2017).
7. Hirakawa, K., Parks, T.W.: Adaptive homogeneity-directed demosaicing algorithm. IEEE Transactions on Image Processing, 14(3), pp.360-369 (2005).
8. Janocha, K., Czarnecki, W.M.: On loss functions for deep neural networks in classification. arXiv preprint arXiv:1702.05659 (2017).
9. Zhang, L., Wu, X., Buades, A., Li, X.: Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. Journal of Electronic imaging, 20(2), p.023016 (2011).

10. N.-S. Syu, Y.-S. Chen, and Y.-Y. Chuang: Learning deep convolutional networks for demosaicing. arXiv preprint arXiv:1802.03769, 2018.
11. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.
12. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In European conference on computer vision (pp. 630-645). Springer, Cham (2016).
13. Syu, N.S., Chen, Y.S., Chuang, Y.Y.: Learning deep convolutional networks for demosaicing. arXiv preprint arXiv:1802.03769 (2018).
14. Ma, K., Duanmu, Z., Wu, Q., Wang, Z., Yong, H., Li, H., Zhang, L.: Waterloo exploration database: New challenges for image quality assessment models. IEEE Transactions on Image Processing, 26(2), pp.1004-1016 (2016).
15. Kokkinos, F., Lefkimmiatis, S.: Deep image demosaicking using a cascade of convolutional residual denoising networks. In Proceedings of the European Conference on Computer Vision (ECCV) (pp. 303-319) (2018).
16. Kokkinos, F., Lefkimmiatis, S.: Iterative Joint Image Demosaicking and Denoising using a Residual Denoising Network. IEEE Transactions on Image Processing (2019).
17. Khashabi, D., Nowozin, S., Jancsary, J., Fitzgibbon, A.W.: Joint demosaicing and denoising via learned nonparametric random fields. IEEE Transactions on Image Processing, 23(12), pp.4968-4981 (2014).