International Journal of Internet, Broadcasting and Communication Vol.15 No.4 74-83 (2023)
http://dx.doi.org/10.7236/IJIBC.2023.15.4.74

IJIBC 23-4-8

Demosaicing based Image Compression with Channel-wise Decoder

Indra Imanuel † and Suk-Ho Lee††

†*Doctoral Degree Candidate, Dept. Computer Engineering, Dongseo University, Korea*
††*Professor, Dept. Artificial Intelligence Appliance, Dongseo University, Korea*
E-mail petrasuk@gmail.com

Abstract

In this paper, we propose an image compression scheme which uses a demosaicking network and a channel-wise decoder in the decoding network. For the demosaicing network, we use as the input a colored mosaiced pattern rather than the well-known Bayer pattern. The use of a colored mosaiced pattern results in the mosaiced image containing a greater amount of information pertaining to the original image. Therefore, it contributes to result in a better color reconstruction. The channel-wise decoder is composed of multiple decoders where each decoder is responsible for each channel in the color image, i.e., the R, G, and B channels. The encoder and decoder are both implemented by wavelet based auto-encoders for better performance. Experimental results verify that the separated channel-wise decoders and the colored mosaic pattern produce a better reconstructed color image than a single decoder. When combining the colored CFA with the multi-decoder, the PSNR metric exhibits an increase of over 2dB for three-times compression and approximately 0.6dB for twelve-times compression compared to the Bayer CFA with a single decoder. Therefore, the compression rate is also increased with the proposed method than with the method using a single decoder on the Bayer patterned mosaic image.

Keywords: *Autoencoder, Image Compression, Mosaicing, Demosaicing*

1. Introduction

In recent years, there have been many studies attempting to apply deep learning to image compression. Research on image compression using deep learning can be broadly divided into two categories. One is to compress the image by transforming it into a low dimensional latent representation through methods that reduce the spatial information of the image. The other method involves compressing the transformed representation further into a special binary format using entropy encoding such as run-length encoding etc.

Manuscript Received: August. 23, 2023 / Revised: August. 29, 2023 / Accepted: September. 5, 2023
Corresponding Author: petrasuk@gmail.com
Tel: +82-51-320-1744, Fax: +82-51-327-8955
Professor, Department of Computer Engineering, General graduate school, Dongseo University, Korea

The autoencoder has been a common choice to compress the input image into a compressed latent representation. Early autoencoders found utility in a range of tasks, including feature learning, dimensionality reduction, and noise reduction, as researchers recognized their capability to capture essential data patterns[1][2][3][4][5].

Recently, researchers began adapting autoencoders for the purpose of lossy image compression[6][7]. They achieved this by training autoencoders to transform high-resolution images into a lower-dimensional latent space and subsequently decoding them into slightly lower-resolution images. This innovative approach allowed for compression while preserving satisfactory image quality.

However, until now, autoencoders for compression purposes have been used to reconstruct the input RGB image itself, but have not been combined with demosaicing methods for compression purposes.

Demosaicing can be regarded itself as a kind of compression method as it tries to reconstruct a colored image back from a single channel color filter array(CFA) image. The objective of demosaicing is to reconstruct the color image as accurately as possible to match the true color image. Most demosaicing networks are implemented with the well-known U-Net structure. The U-Net shares similarities with the autoencoder, with the key distinction being the presence of skip connections in the U-Net, which are absent in the autoencoder. When considering demosaicing for compression purposes, skip connection cannot be incorporated in the neural network. This is because we can utilize only the low-dimensional representation as the input to the decoder. The exclusion of the skip connection leads to a decline in the quality of the reconstructed image. Consequently, to mitigate this decline, special techniques must be employed.

In this paper, we advocate for the use of a colored mosaic pattern instead of the well-known Bayer pattern, as it preserves more information. Additionally, we incorporate a channel-wise decoding approach within the compression network. Channel-wise decoders excel in color reconstruction because each decoder can specialize in reconstructing a particular color, allowing them to consider the unique characteristics of each color more effectively. Experimental results verify that the proposed method can better reconstruct the color image, and therefore, can achieve a higher compression ratio than the normal Bayer pattern and single decoder based method.

2. Preliminaries

The following techniques are related to the proposed method.

2.1 Autoencoder based Compression Framework

In the realm of deep learning, an auto-encoder refers to a network structure designed to produce the input data as its output. The concept of employing an auto-encoder as a compression tool is illustrated in Fig. 1.

Here, the encoder compresses the input RGB image into a lower-dimensional latent representation, ideally having a smaller size than the original color image. This compact latent representation is then transmitted to a remote receiver, where the decoder reverses the process, reconstructing the color image. Loss occurs during this compression because the low-dimensional latent representation is a compressed version of the original data. The extent of information loss hinges on the degree of compression applied. Consequently, a tradeoff exists between the compression rate and the quality of the decoded color image. For example, in [10], an encoder-decoder network is proposed which is characterized by a novel scaled-additive framework, designed to accommodate variable compression rates, whereas in [11], an autoencoders is proposed which incorporates a hyperprior that captures spatial dependencies in the latent representation. Furthermore, in [12], a technique is proposed which can navigate the rate distortion trade-off for an image compression auto-encoder. In all those autoencoder based compression frameworks the aim of the auto-encoder is to reconstruct the input image again at the output and focus to optimize on the entropy encoding of the latent representation. Compared to these works, we propose to obtain the latent representation of the mosaiced image that can reconstruct the demosaiced image.

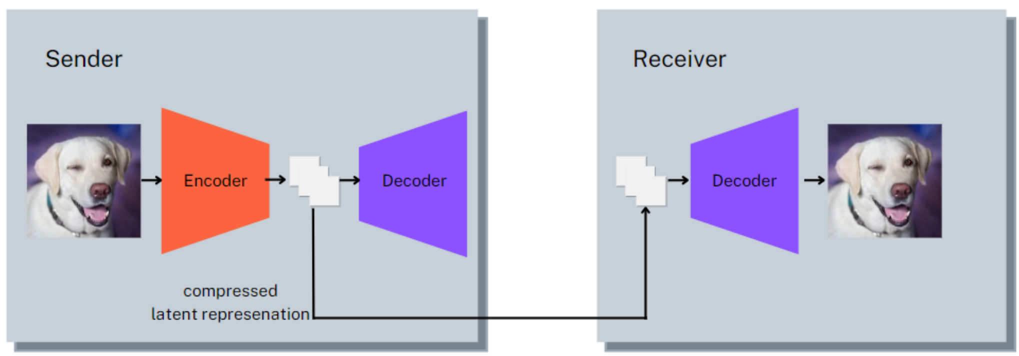

Figure 1. Diagram of the Auto-Encoder based Compression Framework

2.2 Mosaicing as a Tool for Compression

Originally, demosaicing refers to the procedure of reconstructing an image captured through a Color Filter Array (CFA) placed over a digital camera's image sensor. Its purpose is to recover the missing colors within the CFA image. This mosaicing process, applied to the original color image, effectively condenses the information to a third of its original size—a form of 3:1 compression. Demosaicing acts as a decoder in this context, converting the mosaic image back into a fully reconstructed color image. However, when utilizing the mosaicing-demosaicing method for compression and decompression, we don't employ actual color filter arrays for mosaicing. Instead, we mimic the mosaicing process, constructing the mosaic image through simulation. This simulated process can be seen as a compression technique, reducing the image size at the cost of some information loss.

Figure 2 shows the diagram of the concept of using mosaicing and demosaicing as a compression and decompression method. The mosaicing is accomplished through a program that takes the original color image as its input. The demosaicing, which is a challenging and ill-posed task, can be effectively performed by a neural network, as depicted in Figure 3.

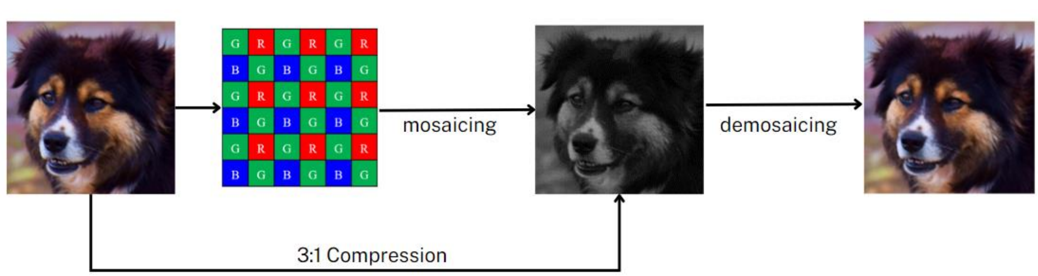

Figure 2. Diagram of the concept of demosaicing based compression

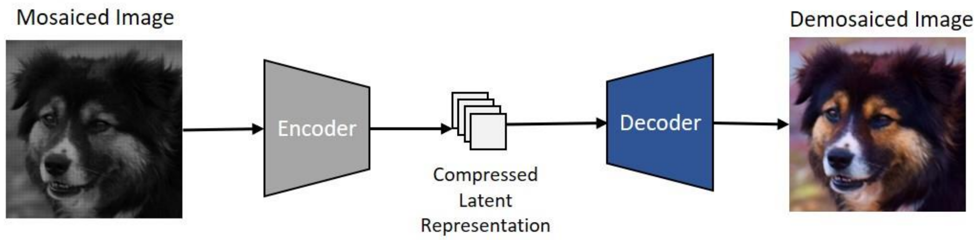

Figure 3. Usage of the auto-encoder for the demosaicing process

3. Proposed Method

3.1 Mosaicing with Colored Mosaic Pattern

In doing the compression, we simulate the mosaicing process on the original color image, to get the 3:1 compressed mosaic image. In other words, we first do image mosaicing with a certain mosaic pattern, and then do channel-wise summing to produce a single-channel mosaic image. We refer to this process as function $M(\cdot)$, i.e., given the original color image $\boldsymbol{x}$, the mosaiced image becomes $M(\boldsymbol{x})$. The mosaiced image $M(\boldsymbol{x})$ is then fed into the wavelet autoencoder. Having a single-channel mosaiced input image by itself is already a 3x image compression. It will also make it easier for the encoder to compress the image into smaller size.

The wavelet encoder will then compress the input image into a compressed latent representation and send it to the wavelet decoders. For example, an input image of size 128x128x1 can be compressed into a 32x32x4 latent representation. In this case, we get a compression ratio of 4:1. Together with the 3:1 compression ratio of the mosaicing process, the total compression ratio then becomes 12:1. We found out that if we use a colored mosaic pattern rather than the conventional Bayer mosaic pattern, the reconstruction becomes better. This is due to the fact that the colored mosaic pattern preserves more information than the Bayer mosaic pattern.

Let denote by $c[\mathbf{k}]$ the action of the mosaic pattern at the pixel position $\mathbf{k}$:

$$\boldsymbol{c}[\mathbf{k}] = [c_R[\mathbf{k}], c_G[\mathbf{k}], c_B[\mathbf{k}]]. \tag{1}$$

That is, we get the intensity value at the position $\mathbf{k}$ of the mosaiced image $M(\boldsymbol{x})$ by

$$M(\boldsymbol{x})[\mathbf{k}] = \boldsymbol{c}[\mathbf{k}]^T \boldsymbol{x}[\mathbf{k}], \tag{2}$$

where $M(\boldsymbol{x})[\mathbf{k}]$ denotes the $\mathbf{k}$-th pixel of the mosaiced image $M(\boldsymbol{x})$ and $\boldsymbol{x}$ denotes the original color image. We propose to use the colored mosaic pattern in Fig. 1, which means we use the following values for $c[\mathbf{k}]$:

$$\begin{aligned}
&\mathrm{c_R}[\mathbf{k}] = 1, c_G[\mathbf{k}] = 0.5, c_B[\mathbf{k}] = 0 && \text{if } \mathbf{k} \in S_1 \\
&\mathrm{c_R}[\mathbf{k}] = 1, \mathrm{c_G}[\mathbf{k}] = 1, \mathrm{c_B}[\mathbf{k}] = 1 && \text{if } \mathbf{k} \in \mathrm{S_2} \\
&\mathrm{c_R}[\mathbf{k}] = 0, \mathrm{c_G}[\mathbf{k}] = 1, \mathrm{c_B}[\mathbf{k}] = 0.5 && \text{if } \mathbf{k} \in \mathrm{S_3} \\
&\mathrm{c_R}[\mathbf{k}] = 0.5, \mathrm{c_G}[\mathbf{k}] = 0, \mathrm{c_B}[\mathbf{k}] = 1 && \text{if } \mathbf{k} \in \mathrm{S_4}
\end{aligned} \tag{3}$$

Compared to the Bayer mosaic pattern, which takes only one of the R,G,B values, the proposed colored mosaic pattern takes more information at each pixel. Due to the more information, the decoding results in a better reconstructed color image.

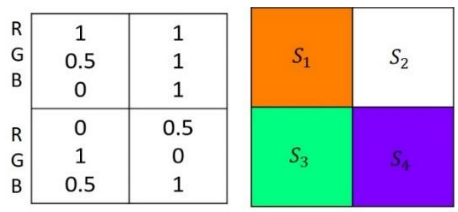

Figure 4. Proposed Colored Mosaic Pattern

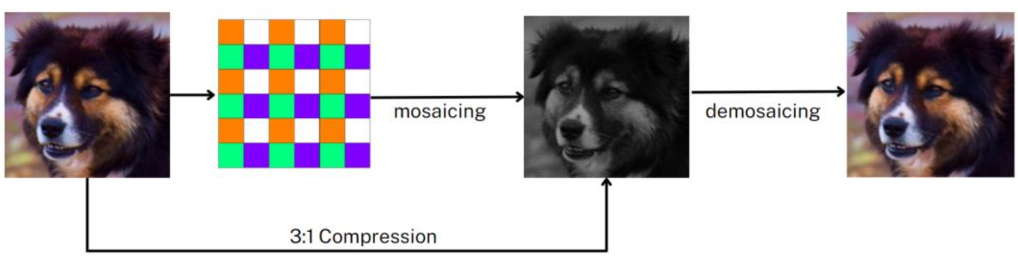

Figure 5. Compression with Proposed Colored Mosaic Pattern

3.2 Usage of a Multi-Decoder

In normal demosaicing networks, a single decoder is used to reconstruct the color image. Instead, we propose to use channel-wise decoders, where each decoder is responsible for reconstructing one of the RGB channels of the image. Assigning a dedicated decoder to each color channel is justified by the ability to handle the unique characteristics of each color independently, leading to improved reconstruction quality. The different decoders are trained by a common reconstruction loss, which is the L_1 norm between the difference of the RGB input image, $\boldsymbol{x}$, and the RGB output image, $\boldsymbol{x}'$:

$$\mathcal{L}_{\text{rec}} = |\boldsymbol{x}' - \boldsymbol{x}| \tag{1}$$

Besides the reconstruction loss between the original RGB color image and the reconstructed RGB image, we also use a constraint loss between $M(\boldsymbol{x})$ and $M(\boldsymbol{x}')$, where $M(\boldsymbol{x}')$ is the mosaic image applied on the output image $\boldsymbol{x}'$. This loss constrains the RGB output image $\boldsymbol{x}'$ to obey the mosaic constraint. This additional constraint aids the decoder in avoiding the generation of a color image that does not correspond to the mosaic input image. The loss for this constraint is

$$\mathcal{L}_{\text{const}} = |M(\boldsymbol{x}') - M(\boldsymbol{x})| \tag{2}$$

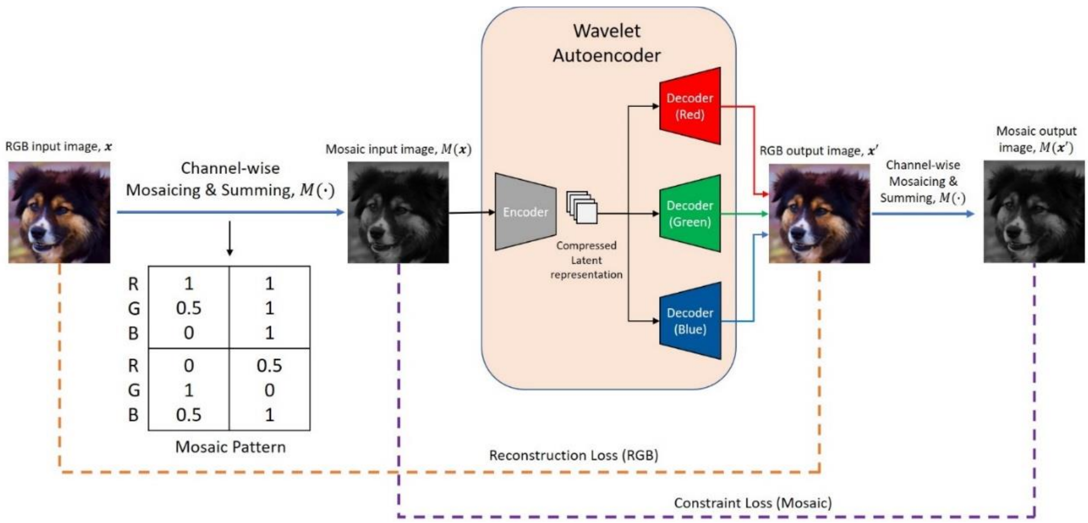

Figure 5. Workflow of the Proposed Method

Figure 5 illustrates the comprehensive workflow of our proposed method. Following the network training, the image undergoes compression through an initial mosaicking step. Subsequently, the encoder compresses it further, yielding a compact latent representation. This compressed representation is then transmitted to the sender, where channel-wise decoders independently decode each color channel. Finally, these channels are concatenated to form the reconstructed color image. Figure 6 shows the decoding step of the proposed method

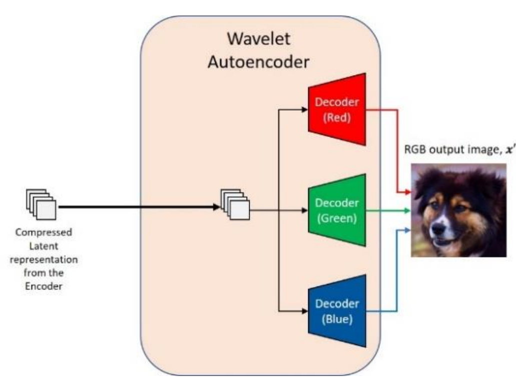

Figure 6. Decoding step of the proposed method

3.3 Architecture of the Proposed Method

Our proposed method achieves good image compression results, capable of compressing a 128x128x3 image into a compact 32x32x4 latent representation, effectively achieving a 12 times reduction in the data size. This accomplishment is made possible through the integration of demosaicing techniques and a wavelet neural network architecture.

Our model architecture draws inspiration from the U-net wavelet neural network design pioneered by Yang & Fu[1]. In our model, we employ the downsampling segment of the U-net as the encoder architecture, while the upsampling segment serves as the decoder architecture. Unlike [1], where the middle part size shrinks but the channel depth increases, our approach maintains a small middle section with fewer channels. This choice is deliberate, as the middle part of our U-net represents the encoded input image—the latent representation—which we aim to keep as compact as possible. To achieve this, we employ a deep network at the final U-net level to reduce channel dimensions. Notably, we refrain from using skip connections between the downsampling and upsampling portions, a departure from Yang & Fu's approach. This decision is driven by our goal of minimizing data transfer from the encoder to the decoder. Instead, we utilize residual blocks separately within the encoder and decoder. Our wavelet autoencoder architecture is visualized in Figure 7. In our network, we replace the max pooling layer with the wavelet pooling to reduce the image dimension. The encoder for the 3 times compression model uses 1 wavelet pooling layer, and four convolutional layers each consisting of with 1 residual block. We use 3x3 filters for all the convolution operations. The decoder for the 3 times compression model uses 1 inverse wavelet layer, and 11 convolutional layers each consisting of four residual blocks. Meanwhile, for the 12 times compression model, we use the same structure as the 3 times compression model, except that we use 2 wavelet pooling layers and 2 inverse wavelet layers instead of 1.

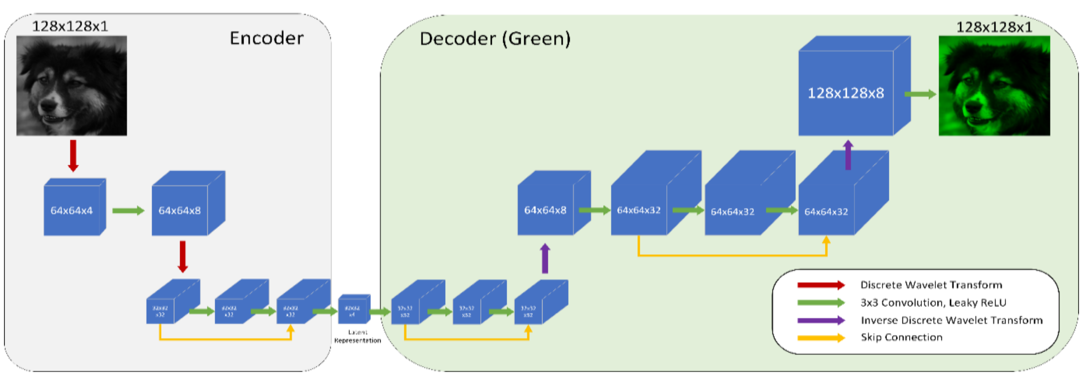

Figure 7. Architecture of the Wavelet Autoencoder Model

4. Experimental Results

For our experiment, we use the CLIC 2022 Dataset [9]. The dataset consists of png images collected from Unsplash. The dataset encompasses 290 training images and 30 test images. Within the training set, we partitioned 264 images for training and reserved 26 images for validation. The image size and orientation vary in the dataset, with the longer side of the image is resized to 2048 px. For consistency during training and testing, we reshape all the image into landscape orientation with a size of 1280x2048 px. We trained the network with the loss functions shown in (1) and (2).

The images are cut into 160 patches of size 128x128 before feeding it into the network, as the image with size 1280x2048 is too big for the network to process. We trained our model for 100 epochs with a learning rate of 0.0001 and used the ADAM optimizer with batch size 8. The model is evaluated both qualitatively and quantitatively. For the quantitative evaluation, we used the PSNR metric.

Figure 8 shows the decoded results with the proposed method for the 3 times and the 12 times compression, respectively. It can be observed that both compression results are visually pleasing, and free from artifacts frequently encountered in standard compression techniques. Figure 9 and Fig. 10 show the enlarged version of the decoded results of the proposed method which uses the proposed CFA with multi-decoder and the method using the conventional Bayer CFA with a single decoder. It can be observed that the proposed method results in less artifacts as can be observed specifically around the letters.

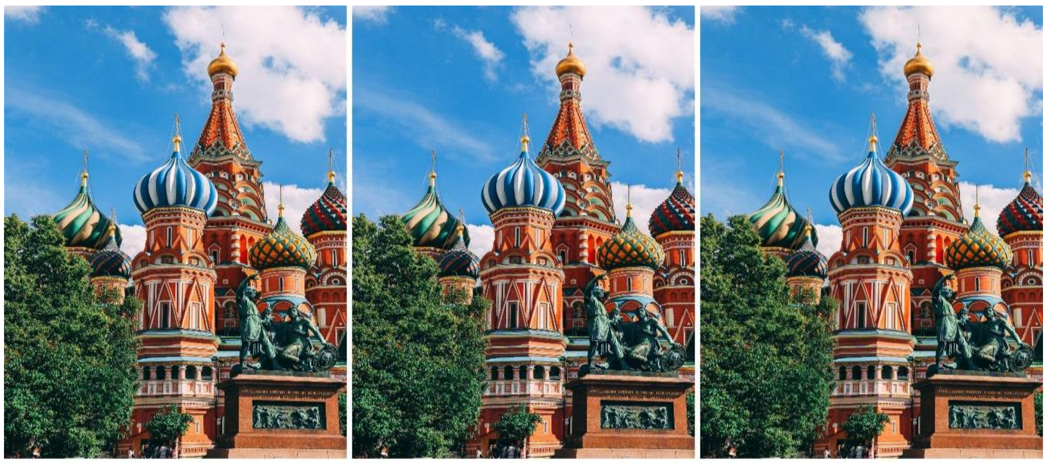

Figure 8. Qualitative Comparison on the CLIC 2022 testset. The 3x compression encodes the input image into 64x64x4 latent representation, while the 12x compression encodes into 32x32x4

The quantitative PSNR comparisons are summarized in Table 1 and Table 2. Here, we also compare the results with the JPEG standard, where we compressed the images to 6:1 and 24:1 with the JPEG. Actually, in our method, we didn't use any optimal entropy encoding, but for the comparison with the JPEG standard, we used the simplest form of entropy encoding, i.e., the lossless Huffman coding. The lossless Huffman coding further reduces the size by a factor of 2, therefore, the 3:1 compression results in a 6:1 compression and the 12:1 compression results in a 24:1 compression. However, it should be noted that we did not propose an optimal full compression framework, but only how to use of mosaiced images in compression, which is why we not compare with optimal compression frameworks. However, the proposed method can be incorporated into any kind of compression framework. Additionally, it's important to highlight that we have not employed entropy encoding at this stage. The compression rate can be further improved when utilized in conjunction with optimal entropy encoding. The proposed method using the proposed colored CFA with multi-decoder has the highest PSNR value in the 24:1 compression case, whereas the JPEG standard highest PSNR value in the 6:1 compression case. It can be seen that the JPEG standard results in many blocky artifacts when the compression rate increases, while the proposed method the reconstructed image has less blocky artifacts but instead shows some smoothing artifacts.

Figure 9. Enlarged Qualitative Comparison of the proposed method with the method using the conventional Bayer CFA with a single decoder for the 3:1 compression rate. Together with the lossless Hoffman encoding, the compression rate becomes 6:1.

Figure 10. Enlarged Qualitative Comparison of the proposed method with the method using the conventional Bayer CFA with a single decoder for the 12:1 compression rate. Together with the lossless Hoffman encoding, the compression rate becomes 24:1.

Table 1. Quantitative comparison for the 6x compression

Method	PSNR(dB)
Colored CFA, Multi-Decoder, 6x Compression	34.742
Colored CFA, Single-Decoder, 6x Compression	33.343
Bayer CFA, Multi-Decoder, 6x Compression	32.456
Bayer CFA, Single-Decoder, 6x Compression	32.082
JPEG Standard, 6x Compression	37.420

Table 2. Quantitative comparison for the 24x compression

Method	PSNR(dB)
Colored CFA, Multi-Decoder, 24x Compression	29.664
Colored CFA, Single-Decoder, 24x Compression	29.426
Bayer CFA, Multi-Decoder, 24x Compression	29.084
Bayer CFA, Single-Decoder, 24x Compression	29.006
JPEG Standard, 24x Compression	27.813

5. Conclusion

In conclusion, our research has delved into the intersection of demosaicing and compression techniques in the context of color image reconstruction. When demosaicing is considered for compression purposes, the incorporation of skip connections into the neural network becomes impractical. This constraint arises since only the low-dimensional representation can be delivered to the receiver as the input to the decoder. The omission of skip connections introduces a trade-off, resulting in a degradation in the quality of the reconstructed image. To address this decline, we have proposed to use a colored mosaic pattern other than the well-known Bayer mosaic pattern, which offers the advantage of preserving more information. Furthermore, we introduce a channel-wise decoding approach within the compression network. This novel approach enables each decoder to focus on the reconstruction of a specific color, harnessing the unique characteristics of each hue more effectively. Empirical evidence from our experiments validates the effectiveness of our proposed method. It demonstrates superior color image reconstruction capabilities, leading to a significantly higher compression ratio when compared to conventional methods employing the Bayer pattern and single decoder approaches. Our findings underscore the potential of leveraging demosaicing techniques for image compression and inspire future explorations in this promising interdisciplinary domain.

Acknowledgement

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea under Grant NRF-2022R1I1A3065211

References

[1] G. E. Hinton, R.R. Salakhutdinov, (2006) "Reducing the Dimensionality of Data with Neural Networks," *Science*, Vol. 313, Issue 5786, pp. 504–507, DOI:https://doi:10.1126/science.1127647

[2] P. Vincent, H. L., (2010) "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion," *Journal of Machine Learning Research*, Vol. 11, pp. 3371–3408

[3] N. Japkowicz, S.J. Hanson, M.A. Gluck, (2000) "Nonlinear Autoassociation Is Not Equivalent to PCA". *Neural Computation*, Vol.12, No. 3, pp. 531–545, DOI:https://doi:10.1162/089976600300015691

[4] M.A. Kramer, (1991) "Nonlinear principal component analysis using autoassociative neural networks," *AIChE Journal*. Vol. 37, No. 2, pp. 233–243, DOI:https://doi:10.1002/aic.690370209.

[5] M. A. Kramer (1992) "Autoassociative neural networks," Computers & Chemical Engineering, Vol. 16, No. 4, pp. 313–328, DOI:https://doi:10.1016/0098-1354(92)80051-A

[6] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool (2018) "Conditional probability models for deep image compression," *arXiv:1801.04260*, DOI:https://doi.org/10.48550/arXiv.1801.04260

[7] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, (2017) "Learning convolutional networks for content-weighted image Compression," *arXiv:1703.10553*, DOI:https://doi.org/10.48550/arXiv.1703.10553

[8] H.-H. Yang and Y. Fu, "Wavelet U-Net and the Chromatic Adaptation Transform for Single Image Dehazing", *in Proc. 2019 IEEE International Conference on Image Processing (ICIP)*, 22-25 September 2019. DOI:https://doi.org/10.1109/ICIP.2019.8803391

[9] CLIC 2022, the 5th Workshop and Challenge on Learned Image Compression, *http://compression.cc/*, accessed August, 10, 2022

[10] G. Toderici, D. Vincent, N. Johnston, S.J. Hwang, D. Minnen, J. Shor, M. Covell, "Full Resolution Image Compression with Recurrent Neural Networks," *Proc. 2017 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 21-26 July 2017, pp.5306-5314. DOI:https://doi.org/10.1109/CVPR.2017.577

[11] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, "Variational image compression with a scale hyperprior," *Proc. 2018 International Conference on Learning Representations(ICLR)*

[12] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, L. V. Gool, "Conditional Probability Models for Deep Image Compression," *Proc. 2018 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 18-23 June 2018, pp. 4394-4402. DOI:https://doi.org/10.1109/CVPR.2018.00462