https://doi.org/10.2352/EI.2025.37.10.IPAS-235

CIS Image Reconstruction Network for Real Time Performance on Mobile Device

Seokhyeon Lee, Yoojeong Seo, Kundong Kim, Sung-su Kim, and Yitae Kim; S.LSI Division, Samsung Electronics; Hwaseong-si, Gyeonggi-do, Republic of Korea

Abstract

With the emergence of 200 mega pixel QxQ Bayer pattern image sensors, the remosaic technology that rearranges color filter arrays (CFAs) into Bayer patterns has become increasingly important. However, the limitations of the remosaic algorithm in the sensor often result in artifacts that degrade the details and textures of the images. In this paper, we propose a deep learning-based artifact correction method to enhance image quality within a mobile environment while minimizing shutter lag. We generated a dataset for training by utilizing a high-performance remosaic algorithm and trained a lightweight U-Net based network. The proposed network effectively removes these artifacts, thereby improving the overall image quality. Additionally, it only takes about 15 ms to process a 4000x3000 image on a Galaxy S22 Ultra, making it suitable for real-time applications.

Introduction

In recent years, smartphones have increasingly adopted high-resolution image sensors that offer crop zooms, known as in-sensor zooms or lossless zooms. By utilizing only 12.5MP located in the center of the 200MP full pixel array, it is possible to achieve the same level of zooming effect as x4. Although it offers high-quality zoom performance without telephoto lenses, it is prone to artifacts such as false colors and unnatural texts as shown in Figure 1.

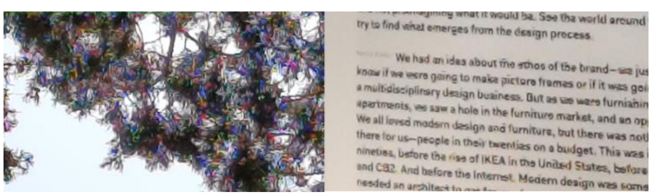

Figure 1. *Example of QxQ sensor remosaic artifacts: (left) false color, (right) unnatural text*

When cropping a 12.5MP image from a 200MP sensor with a QxQ Bayer pattern, the output does not follow a Bayer pattern. Since most image signal processors are designed to process Bayer-patterned images, the sensor must apply an in-sensor remosaicing algorithm to convert the QxQ Bayer pattern into a Bayer pattern. This step is essential to ensure compatibility with standard image processing pipelines. Additionally, the sensor's Bayer pattern output should achieve 30 frames per second to meet real-time processing requirements. To accomplish this, the remosaicing algorithm must be highly efficient and designed for low-power operation to optimize battery life and thermal performance. However, these performance constraints can lead to interpolation errors in certain areas, especially where there are sharp color transitions or unclear directional patterns. As a result, artifacts may appear in the image, affecting overall visual quality.

To enhance image quality, an alternative approach is to disable the in-sensor hardware remosaicing and utilize software-based remosaicing instead. As illustrated in Figure 2, this method employs the sensor's built-in remosaic algorithm for real-time processing in preview mode, where maintaining a high frame rate is essential. However, during image capture, a software remosaicing algorithm is applied to achieve superior image quality. This approach eliminates the need to maintain 30fps, thereby enabling the implementation of a high-performance software remosaicing algorithm. Such an algorithm allows for more precise interpolation, improved color accuracy, and enhanced overall image quality.

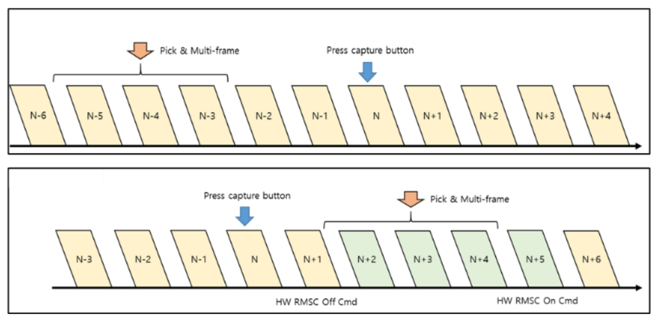

Figure 2. *Changes in image buffer (zero-shutter-lag buffer) usage depending on hardware remosaicing on/off: (top) default scenario, (bottom) scenario with software remosaicing*

A significant limitation of this method is that it cannot utilize the zero-shutter-lag (ZSL) buffer [1] with in the image processing pipeline. The ZSL buffer stores frames captured immediately prior to the user pressing the shutter button, compensating for the delay between the capture moment and the actual image acquisition. This is particularly critical in crop zoom scenarios, where capturing fast-moving subjects requires minimizing shutter lag between the desired and actual capture moments. When employing software-based remosaicing, frames stored in the ZSL buffer remain in Bayer format and cannot be directly utilized for software remosaicing. Consequently, only frames stored after the shutter button is pressed can be processed, potentially introducing a delay and affecting real-time responsiveness in dynamic imaging scenarios.

This paper proposed a method that combines existing approaches to achieve higher image quality than hardware remosaic while avoiding shutter lag as shown in Figure 3. In preview mode, the system utilizes hardware remosaic to ensure real-time processing. However, during image capture, instead of performing a full software-based remosaic, a U-Net based lightweight network is employed to effectively remove only the artifacts. This approach allows for improved image quality while minimizing computational overhead. While the image quality may not match that of a software-based remosaic, it eliminates shutter lag and achieves significantly better image quality than hardware remosaic alone.

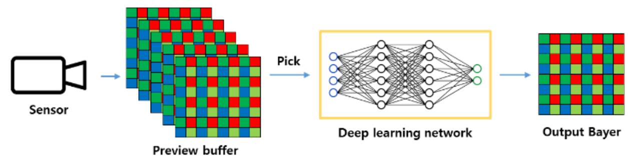

Figure 3. *The proposed image processing pipeline during image capture*

Method

Dataset Construction

To train a model for artifact removal, it is essential to have paired datasets consisting of Bayer images processed with hardware remosaicing and clean Bayer images without artifacts. Instead of directly capturing a dataset using a QxQ Bayer sensor, we leveraged the publicly available RIASE dataset [2], which consists of high-quality images captured with a DSLR. However, since the RAISE dataset follows a conventional Bayer pattern, it must be converted into a QxQ Bayer patter to align with our target sensor characteristics.

To achieve this transformation, we first applied a demosaicing algorithm to the original Bayer images in the RAISE dataset generating RGB images. Subsequently, we performed QxQ Bayer sampling on these RGB images to reconstruct them into the QxQ bayer pattern. Once dataset was converted into this format, we applied both hardware-based remosaicing and software-based remosaicing to corresponding dataset pairs as shown in Figure 4. This dataset enables the model to learn the mapping between hardware-remosaiced Bayer images which contain artifacts, and their clean Bayer counterparts.

Training Software Remosaic

For the generation of GT, we trained a software remosaicing model based on a U-Net [3]. Similar to the process illustrated in Figure 4, we first applied a demosaicing algorithm to the RAISE dataset and then performed QxQ pattern sampling to generate the QxQ Bayer input dataset. The original Bayer images from the RAISE dataset were used as the corresponding GT. Since the purpose of software remosaicing for GT generation is to produce high-quality Bayer images with minimal artifacts, we did not employ a lightweight model optimized for efficiency. Instead, we prioritized reconstruction accuracy to ensure that the generated GT closely matched the original Bayer images, thereby providing high-fidelity supervision for training the artifact removal model.

Initially, we considered using the Bayer images from the RAISE dataset as ground truth (GT) directly without applying software remosaicing. However, the QxQ pattern sampling process introduces significant data loss, leading to a noticeable degradation in image quality when remosaicing is applied to the QxQ Bayer data. As a result, the Bayer images generated through this process exhibit lower quality compared to the original Bayer images in the RAISE dataset.

Given these limitations, using the original RAISE dataset as GT would require the model not only to remove artifacts but also to reconstruct the information lost during the QxQ sampling process. This significantly increases the complexity of the task, as the model would need to perform both artifact removal and data restoration. Such an approach demands a more complex and computationally expensive model, which is unsuitable for resource-constrained environments. Since our primary objective is to develop a lightweight model optimized for efficient execution on mobile devices, we concluded that this approach would be impractical.

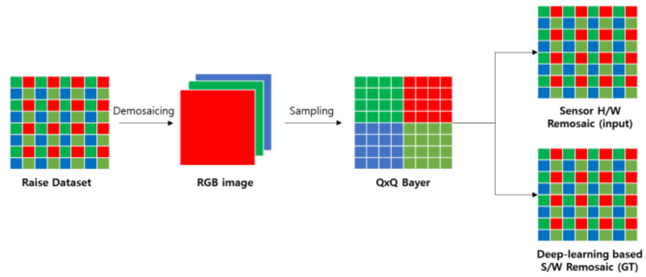

Figure 4. *The process of creating a training dataset*

Deep Learning Model

The proposed model adopts a lightweight structure utilizing residual blocks for efficient feature extraction and artifact removal. Similar to U-Net, which reduces the spatial dimension to extract hierarchical features, our model employs convolutional layers for down-sampling. After the initial down-sampling stage, the features are processed through multiple residual blocks, followed by transposed convolution layers to restore the original resolution.

Since most regions of the image are free from artifacts, the input and output should remain identical in these areas. To enforce this property, we employed a residual learning approach by adding the input directly to the network output as seen in Figure 5.a. This allows the model to focus primarily on learning the residuals, ensuring that artifact-free regions are preserved while selectively correcting regions affected by artifacts.

To maintain a lightweight design suitable for real-time applications, we fixed the number of feature channels in all convolutional layers to 32. Additionally, to enhance the performance of the model, we integrated ideas from ResUNet [4], incorporating residual blocks that utilize multiple convolutional layers with different kernel sizes, ReLU activation functions, and identity mapping. This architectural choice enables the model to effectively capture both local and global features while maintaining computational efficiency as illustrated in Figure 5.b. Our model was trained using L1 loss, which is widely used for image-to-image regression tasks. Also, the model is trained using the ADAM optimizer with $\beta1 = 0.9$, $\beta2 = 0.99$, and a learning rate of 10^{-4}. The training images were cropped to 768 x 768 with a batch size of 16.

For the experimental environment, we deployed our model on a Samsung Galaxy S22 Ultra, which is equipped with a Qualcomm Snapdragon 8 Gen 1 processor. The system was configured to process Bayer images efficiently on the device. To achieve fast inference speed, we utilized the Snapdragon Neural Processing Engine (SNPE) SDK [5], specifically leveraging its Quantization Toolkit for post-training quantization. The Quantization Toolkit supports 8-bit and 16-bit fixed-point precision post-training quantization and enables inference acceleration using dedicated DSP hardware instead of the CPU. For optimal performance, the proposed model was quantized with 16-bit activation precision and 8-bit weight precision. All image quality evaluations were conducted using the quantized model to ensure that the reported performance reflects real-world deployment conditions.

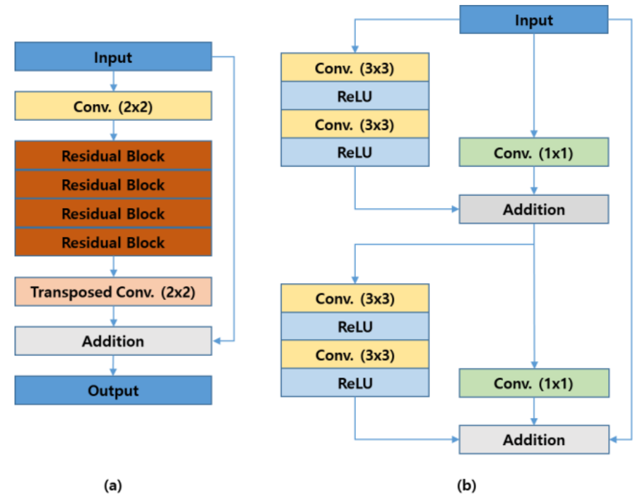

Figure 5. *(a) Block diagram of the proposed model architecture, (b) Residual block in the proposed model*

Result

Quantitative evaluation

Table 1. Image quality results of baseline, proposed model, and GT images

Metric	Baseline	Proposed	SWRMSC (GT)
Siemens Star MTF10	3568	3535	3458
Siemens Star MTF25	2939	2945	2937
Siemens Star MTF50	2157	2120	2177
Dead Leaves HC MTF10	2694	2773	3000
Dead Leaves LC MTF 10	2240	2322	2737
Slanted Edge 60	16.55	16.31	17.7
Slanted Edge 80	22.65	22.67	23.73
Visual Noise	96.45	95.96	92.67
SNR	40.55	40.5	38.61

For the evaluation of image quality, we utilized the TE42 [6] test chart from Image Engineering, which is widely used for assessing camera and imaging system performance. The dataset was captured using a 200 MP sensor under a controlled lighting environment of 1000 lux with a D65 halogen light source. The TE42 chart was positioned to occupy half (1/2) of the sensor's field of view to ensure a detailed assessment of image quality characteristics.

The TE42 test chart enables comprehensive evaluation of multiple image quality aspects, including resolution, texture reproduction, dynamic range, and color accuracy. [7-10] Since remosaicing artifacts predominantly appear in high-frequency regions, we hypothesized that artifact correction could potentially lead to a reduction in overall image sharpness. To validate this, we focused our analysis on MTF measurements, which provide an objective metric for evaluating resolution and sharpness. The detailed results are presented in Table 1.

When comparing the Siemens Star MTF results of the proposed model with the baseline, we observed that the performance remains almost identical. However, in the Dead Leaves MTF measurement, our model achieves an approximately 3% improvement, indicating enhanced representation of random texture patterns. Since the flat regions remain unchanged, the SNR values are also almost same.

Qualitative evaluation

We found that conventional quantitative evaluation metrics alone were insufficient for accurately assessing the level of artifacts. Therefore, in addition to quantitative analysis, we conducted a qualitative evaluation to further examine the visual impact of our model's artifact removal performance. As shown in Figure 6, the proposed model demonstrates notable improvements in false color suppression and text readability, leading to a morwe visually accurate reconstruction. However, it appears that the model does not fully recover the fine details lost due to limitations of hardware remosaicing, suggesting that some high-frequency information remains irretrievable after the remosaicing process.

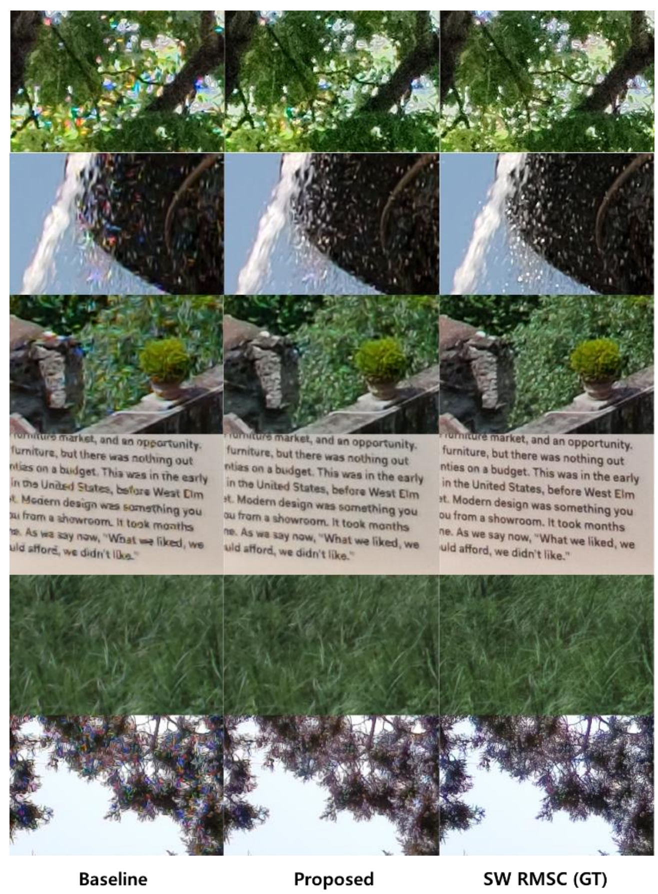

Figure 6. *Qualitative comparison*

Conclusion

High-resolution sensors, such as 200 MP sensors, enable crop zoom functionality. However, artifacts often arise due to the limitations of hardware processing. While software remosaicing can improve image quality, it introduces shutter lag since the camera buffer cannot be used for this scenario. To address this issue, we propose a method that effectively removes artifacts while minimizing shutter lag. We also propose a novel model architecture optimized for mobile environments and constructed a suitable dataset to facilitate this task. This approach preserves the resolution of the original image while selectively eliminating artifacts, ensuring high image quality without compromising sharpness. Furthermore, all experiments were conducted with a mobile environment. We verified that the proposed model runs efficiently on the Samsung Galaxy S22 Ultra, achieving a latency of 15 ms, making it suitable for real-time operation on mobile devices.

References

[1] G., Gagrani, et al., Adaptive buffering rate technology for zero shutter lag (ZSL) camera-inclusive devices, U.S. Patent 15/414,030 filed Jan. 24, 2017 and issued Jul. 26, 2018

[2] D.-T., Dang-Nguyen, C, Pasquini, V., Conotter, G., Boato, RAISE – A Raw Images Dataset for Digital Image Forensics, ACM Multimedia Systems, Portland, Oregon, March 18-20, 2015

[3] O., Ronneberger, P., Fischer, T., Brox: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention, pp. 234–241, 2015

[4] Z., Zhang, Q., Liu, Y., Wang, Road extraction by deep residual u-net. IEEE Geoscience and Remote Sensing Letters 15(5), 749–753, 2018

[5] SNPE. https://www.qualcomm.com/developer/software/neural-processing-sdk-for-ai

[6] TE42. https://www.image-engineering.de/products/charts/all/425-te42.

[7] D., Wueller, & U. B., Kejser, Standardization of image quality analysis - ISO 19264. Proc. IS&T Archiving Conference. 1, 111-116, 2016

[8] F., Cao, et al., Dead leaves model for measuring texture quality on a digital camera, Proc. SPIE, 7537, 75370E, 2010

[9] D., Baxter, et al., Development of the I3A CPIQ spatial metric, Proc. SPIE, 8293, Quality and System IX, 829302, 2012

[10] U., Artmann, Image quality assessment using the dead leaves target: experience with the latest approach and further investigations, Proc. SPIE, 9404, 94040J, 2015

Author Biography

Seokhyeon Lee received her B.S. in electronic engineering from Seoul National University in 2019. Since then, he has worked in Samsung Electronics, Republic of Korea. His work has focused on the machine learning for image sensors such as automating image quality tuning and remosaicing