Event Camera Demosaicing via Swin Transformer and Pixel-focus Loss

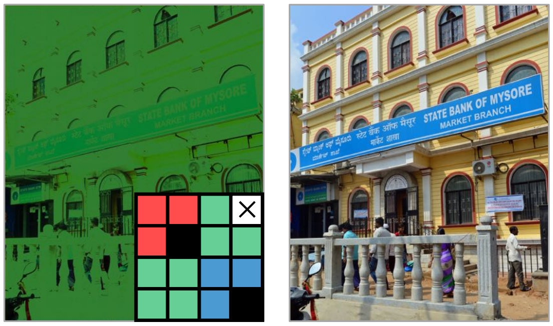
Yunfan Lu, Yijie Xu, Wenzong Ma, Weiyu Guo, Hui Xiong
AI Thrust, Hong Kong University of Science and Technology (Guangzhou)
{ylu066,yxu409,wma423,wguo395}@connect.hkust-gz.edu.cn, xionghui@ust.hk

Abstract

Recent research has highlighted improvements in high-quality imaging guided by event cameras, with most of these efforts concentrating on the RGB domain. However, these advancements frequently neglect the unique challenges introduced by the inherent flaws in the sensor design of event cameras in the RAW domain. Specifically, this sensor design results in the partial loss of pixel values, posing new challenges for RAW domain processes like demosaicing. The challenge intensifies as most research in the RAW domain is based on the premise that each pixel contains a value, making the straightforward adaptation of these methods to event camera demosaicing problematic. To end this, we present a Swin-Transformer-based backbone and a pixel-focus loss function for demosaicing with missing pixel values in RAW domain processing. Our core motivation is to refine a general and widely applicable foundational model from the RGB domain for RAW domain processing, thereby broadening the model's applicability within the entire imaging process. Our method harnesses multi-scale processing and space-to-depth techniques to ensure efficiency and reduce computing complexity. We also proposed the Pixel-focus Loss function for network fine-tuning to improve network convergence based on our discovery of a long-tailed distribution in training loss. Our method has undergone validation on the MIPI Demosaic Challenge dataset, with subsequent analytical experimentation confirming its efficacy. All code and trained models are released here: https://github.com/yunfanLu/ev-demosaic.

1. Introduction

The event camera [8, 53, 59], with its low latency ($< 100\mu s$), high dynamic range ($> 120dB$), high temporal resolution ($> 1000fps$), and efficient power consumption, has garnered significant interest for enhancing computational imaging in applications, *e.g.*, video frame interpolation [28, 35, 46], super-resolution [15, 29], deblurring [14, 44, 52], and high dynamic range [32, 38]. These works are realized in the RGB domain, based on the premise that the camera sensor can simultaneously and seamlessly deliver RGB images and events. These RGB images are obtained by RAW image processing approaches. Specifically, RAW domain approaches convert RAW images, where each pixel contains only one type of color information with noise, into full-color images with all three RGB color information with high-quality [13]. Filling in the missing color information is known as demosaicing [31, 53], a core component of RAW domain image process.


(a) Raw Image and Sensor design

(b) RGB Color Image

Figure 1. Contemporary design of an actual event camera sensor (Hybridevs sensor), featuring red, green, and blue pixels for outputting RGB RAW signals. Black pixels in the lower right corner of the green and red areas are designated for event signal output, and white pixels do not emit any signals. The demosaicing task aims to convert a RAW image with RGB signals and black holes (a) into a full-color image with three RGB channels (b).

However, the transformation of RAW to RGB faces significant challenges due to the existing event sensor chip design technology. Specifically, event cameras produce RAW images where specific pixels are absent, as illustrated in Fig. 1. These missing pixels emit event signals, not RGB signals, resulting in incomplete pixel values in the RAW output. This absence of pixel values poses challenges for traditional RAW domain processing approaches, such as demosaicing, because the pixels emitting event signals cause the RAW image to lose **a quarter of** its red and blue color information. To effectively address the challenge of enhancing downstream RGB images/videos with event guidance, it is imperative to transform incomplete RAW images into high-quality RGB full-color images without losing information. Moreover, minimizing accumulated errors in this transformation process is essential for improving the quality of inputs for downstream tasks [12].

RAW image process methods [1, 10, 17, 27, 31, 37, 43, 57] are predicated on traditional sensor technology, wherein each pixel is capable of capturing a color signal. These methods can be divided into model-based [10, 31, 43, 57] and learning-based methods [1, 17, 37]. Learning-based approaches have attracted substantial interest due to the powerful fitting capabilities and robust generalizability of neural networks. Inspired by the evolution of network architectures, these methods primarily aim to integrate and innovate neural network designs for RAW to RGB conversion mapping. Many convolutional neural networks (CNNs) [1, 17, 37] were designed as the backbone for tasks such as demosaicing and denoising of RAW images. For instance, PyNet [12] has designed a multi-scale, multi-resolution CNN network architecture for processing RAW into RGB. More recently, benefits from the enhanced contextual modeling abilities and broader receptive fields from Vision Transformers (ViT) [7]. Many Transformer-based methods are proposed for image processing [16, 26, 42, 49, 51, 55]. For example, RSTCANet [51] employs the Swin-Transformer [9, 22] to the demosaicing, incorporating global residual connections. However, RSTCANet [51] stacks Swin-Transformer layers [22, 24, 25] without downsample and multi-scale leads to higher computational complexity while failing to provide the network with a sufficiently large field of view.

Based on these considerations, we employ the Swin-Transformer-based backbone and a pixel-focus loss function for event camera demosaicing. Our motivation is threefold: **(1) Scalability:** The Swin Transformer is a widely used and powerful foundational model in the RGB domain. Adapting it to the RAW domain could bridge foundational modeling across RAW and RGB imaging tasks. **(2) Efficiency:** RAW domain methods are upstream of RGB domain processes and underpin all computer vision tasks. Therefore, RAW domain methods need to be sufficiently efficient to support downstream applications in the real world. **(3) Training Effectivity:** A long-tail distribution of training loss was identified for the demosaicing task. Consequently, the pixel-focus loss was designed to facilitate a two-stage training process, enhancing the network's performance.

To achieve scalability, we refine the standard operators from the Swin-transformer [9, 22] while avoiding customizations to enhance its portability. To ensure efficiency, we initially employ the space-to-depth [5] method to reduce the network's resolution and design a network structure akin to U-Net [39], achieving multi-scale and multi-resolution capabilities. This structure allows for a broader field of view with fewer layers. For effective training, we devised a two-stage loss function to fine-tune the network after completing the first training phase with *Charbonnier* loss [18].

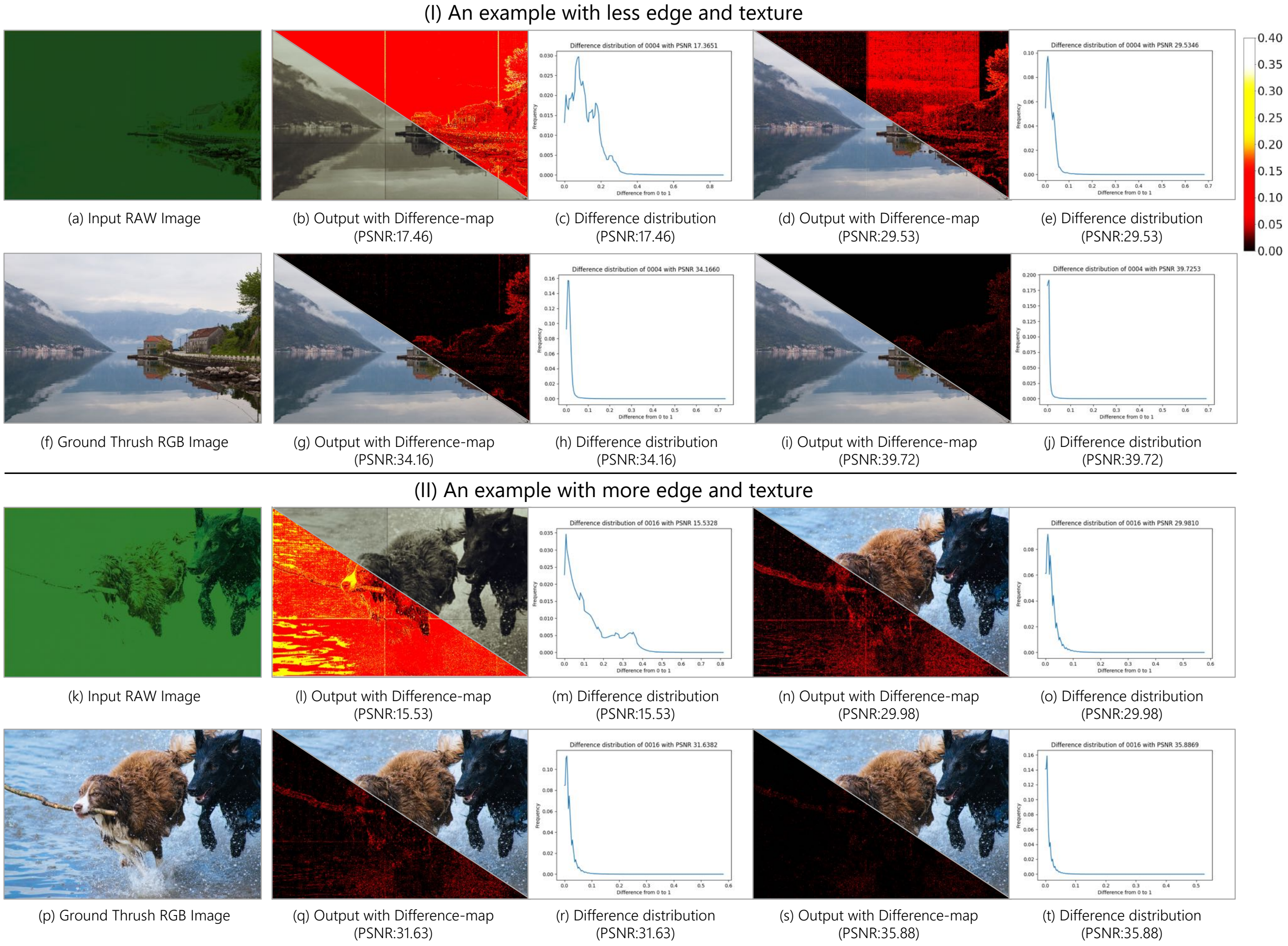
Our method underwent testing on the MIPI Demosaic Challenge dataset [34, 53] of the CVPR 2024 Workshop, demonstrating its applicability and performance. Subsequently, we conducted additional analytical experiments to evaluate its robustness and adaptability. These tests solidified the method's effectiveness in various scenarios, clearly illustrating the superiority of our approach in addressing the intricacies of demosaicing in the RAW domain. Furthermore, we believe this work will inspire applications in the RAW domain and catalyze enhancements across multiple RAW-based tasks, fostering a new wave of innovation.

2. Related Works

Modern digital cameras capture light, producing images with individual color channels (*e.g.*, red, green, or blue) for each pixel [30]. To compensate for the absence of color information, demosaicing is devised to reconstruct a full-color image from a single-channel RAW image [20]. In addition, owing to the wave-particle duality of light and the instability induced by dark currents in electronic devices, noise is a pervasive issue in the pixels of RAW images [3]. Consequently, the processes of denoising and demosaicing frequently occur concurrently. Our paper focuses on the RAW domain processing of event cameras, prioritizing the demosaicing task due to its unique characteristics from the event camera sensor design.

2.1. Camera RAW Image Demosaicing

Demosaicing approaches can be categorized into two main groups: **(1)** model-based methodologies [10, 31, 43, 57], which rely on mathematical models and spatial-spectral image priors for image reconstruction and **(2)** learning-based methodologies [1, 12, 17, 23, 24, 37, 45, 50], which leverage process mappings learned from extensive datasets of ground-truth images and corresponding mosaic counterparts. These techniques use different neural networks, *e.g.*, CNNs and Transformers, to learn complex mappings between mosaic images and their corresponding full-color images. While CNN architectures have been widely used in learning-based demosaicing methods, they are limited in fixed-size receptive fields of convolution kernels and global context awareness compared with Transformer [51]. As a result, recent advancements in Transformer architectures, particularly the Swin-Transformer [51], have shown promise in addressing these challenges and improving the performance of learning-based demosaicing approaches.



(a) Input RAW Image

(b) Output with Difference-map (PSNR:17.46)

(c) Difference distribution (PSNR:17.46)

(d) Output with Difference-map (PSNR:29.53)

(e) Difference distribution (PSNR:29.53)

(f) Ground Thrush RGB Image

(g) Output with Difference-map (PSNR:34.16)

(h) Difference distribution (PSNR:34.16)

(i) Output with Difference-map (PSNR:39.72)

(j) Difference distribution (PSNR:39.72)

(k) Input RAW Image

(l) Output with Difference-map (PSNR:15.53)

(m) Difference distribution (PSNR:15.53)

(n) Output with Difference-map (PSNR:29.98)

(o) Difference distribution (PSNR:29.98)

(p) Ground Thrush RGB Image

(q) Output with Difference-map (PSNR:31.63)

(r) Difference distribution (PSNR:31.63)

(s) Output with Difference-map (PSNR:35.88)

(t) Difference distribution (PSNR:35.88)

Figure 2. Visual results of two images at different stages of training. Example (I) displays an image with less edge and texture, featuring extensive areas of sky and lake, while example (II) presents an image rich in edge and texture, including animal fur and splashing water. For these two examples, four groups of reconstruction results are shown under varying PSNR values, along with different maps and difference distributions. Here, ”difference” refers to the absolute value of discrepancies compared to the ground truth. As PSNR increases, the differences exhibit a long-tailed distribution.

2.2. Transformer-based Imaging Processes

Transformer have been employed in many imaging processes, *e.g.*, image/video super-resolution [29, 54], deblurring [56]. Remarkably, Swin-Transformer [9, 21, 22] utilizes the shifted window mechanism to capture long-range dependencies in images, enabling effective aggregation of information from distant spatial locations, presenting impressive performance in vision tasks like super-resolution [6, 19], video deblurring [4], and video frame interpolation [9, 26]. Besides, Swin-Transformer's hierarchical architecture partitions input images into smaller patches, which are then processed through multiple transformer blocks, facilitating the learning of both local and global features. For example, [22] utilizes a sequence of residual Swin-Transformer blocks for deep feature extraction, demonstrating leading-edge performance across various image super-resolution tasks. Consequently, inspired by these successful works [4, 9, 22] we also employ Swin-Transformer as a backbone to leverage the demosaicing.

2.3. Image Reconstruction Loss Functions

A series of works prefer Charbonnier Loss [18] as the loss function to train their neural network for image reconstruction [48, 58]. Nevertheless, areas characterized by numerous high-frequency details, *e.g.* edges and textures, merit increased attention compared to low-frequency and smooth regions that are easily recoverable during the training process, as shown in Fig. 2. In [23], an adaptive-threshold edge loss is introduced to tackle this challenge, which adaptively adjusts the edge detection threshold for different im-

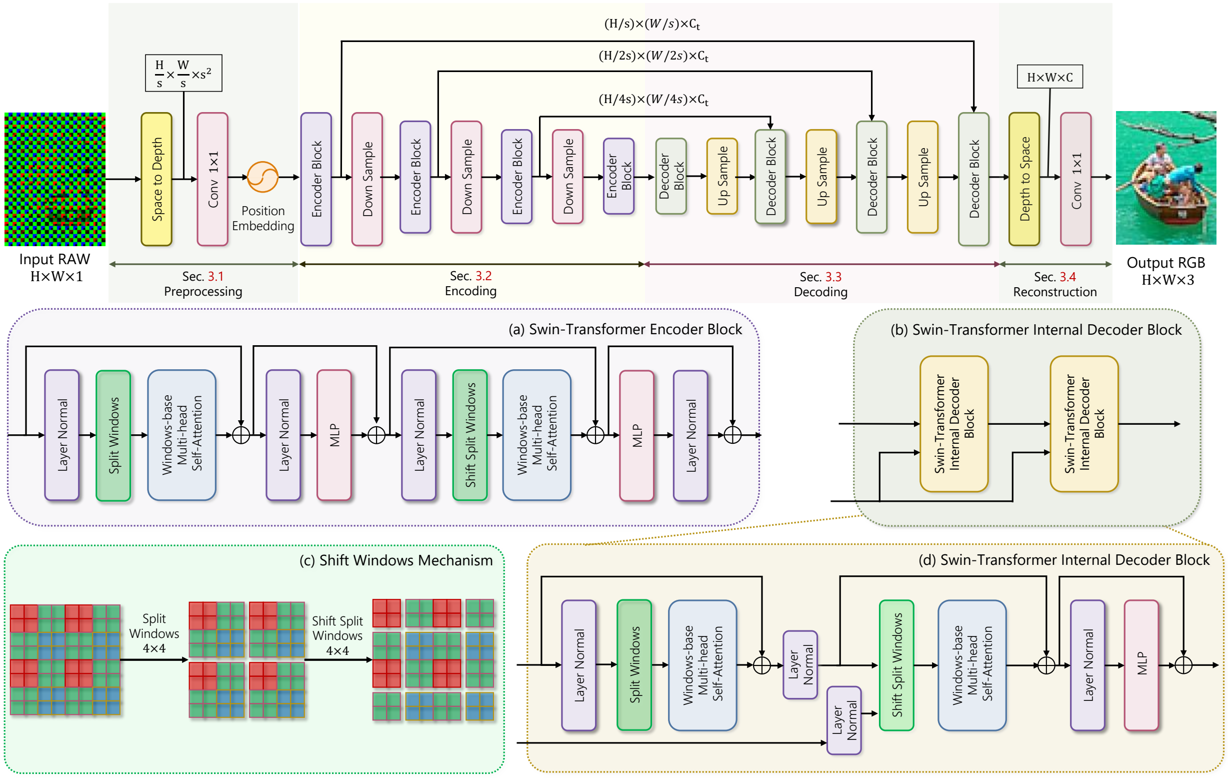


Figure 3. Overview of the event camera demosaicing method. The input RAW image is first preprocessed using space-to-depth and 1×1 convolution operations. The encoder then extracts multi-scale features using Swin Transformer blocks with the shifted window mechanism. The decoder mirrors the encoder's structure and incorporates skip connections to recover spatial details. Finally, the reconstruction module generates the output RGB image. (a) Encoder block architecture. (b) Shifted window mechanism for cross-window interactions. (c) Decoder block architecture.

age patches based on their edge density, allowing the model to focus more on regions with rich edge details during training. However, the loss in [23] needs to divide the image into specific patches according to their edge density, demanding a series of complicated thresholds and cross-entropy loss calculations. Consequently, we offer a facilitated approach called Pixel Focus Loss to optimize the model to capture subtle differences effectively.

3. Methods

This section presents the details of our event camera demosaicing method. As illustrated in Fig. 3, our method leverages the strengths of the Swin-Transformer [22] and the U-Net architecture [39] to take the RAW image with missing pixel values as input and aims to reconstruct a high-quality RGB image. Our framework consists of five key components: (1) preprocessing 3.1 with a space-to-depth operation [40] and a 1×1 convolution, (2) encoding 3.2 with Swin-Transformer and shifted window mechanism, (3) decoding 3.3 with mirrored encoding modules, (4) reconstruction 3.4 with mirrored preprocessing modules, and (5) loss functions 3.5 with exquisite designs.

3.1. Preprocessing

The preprocessing module aims to transform the input RAW image into a suitable representation for the subsequent encoding stages while reducing the computational complexity. Given the input RAW image $I_{RAW} \in \mathbb{R}^{H \times W \times 1}$, we apply a space-to-depth operation [5] with a factor of s to reduce the spatial resolution to $(H/s) \times (W/s)$ and increase the channel dimension to s^2. This operation effectively reduces the model complexity, as the computational cost is linear with respect to the number of channels and quadratic concerning the spatial resolution [22]. Subsequently, a 1×1 convolution is employed to generate the feature $F_0 \in \mathbb{R}^{(H/s) \times (W/s) \times C}$ from the RAW image. To incorporate positional information, we add positional embeddings $E_{pos} \in \mathbb{R}^{(H/s) \times (W/s) \times C}$ to F_0. The positional embeddings [47] are computed using a sinusoidal function:

$$\begin{cases} E_{pos}(2i) & = \sin\left(p/10000^{2i/C}\right) \\ E_{pos}(2i+1) & = \cos\left(p/10000^{2i/C}\right), \end{cases} \tag{1}$$

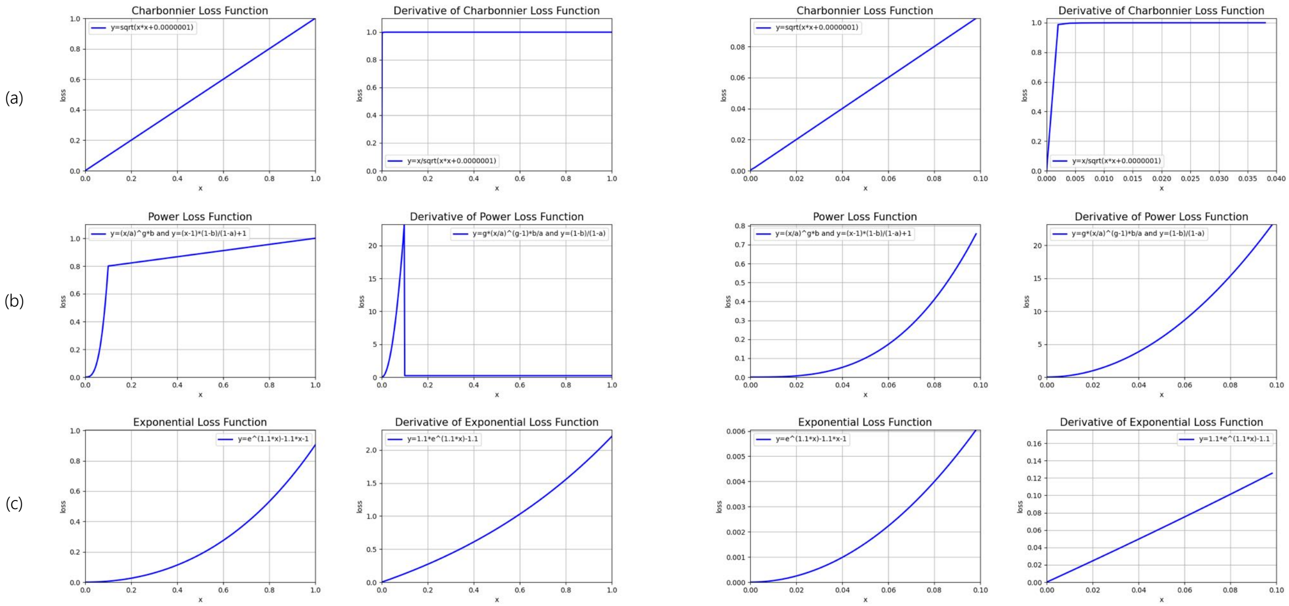


Figure 4. Loss functions visualization. (a) (b) and (c) refer to Charbonnier and pixel-focus loss with the power and the exponential function, respectively. The line charts loss functions within the 0-1 range and their gradients. It also provides a magnified view of the 0 to 0.1 interval to observe the characteristics of different loss functions better when dealing with long-tail distributions.

where p represents the position index and i is the dimension index. The resulting preprocessed feature representation is obtained as $F_0^{'} = F_0 + E_{pos}$.

3.2. Encoding

The encoding module aims to extract multi-scale features and capture long-range dependencies. We adopt a U-Net-like architecture [39] with the Swin-Transformer [22] as the backbone. The encoding module consists of four stages, each containing a Swin-Transformer block followed by a down-sample layer. The Swin-Transformer block comprises a Layer Normalization (LN) layer [2], a Window-based Multi-head Self-Attention (W-MSA) module [24], and a Multi-Layer Perceptron (MLP). The W-MSA module, illustrated in Fig. 3 (a), performs self-attention within local windows of varying sizes, allowing the model to capture multi-scale features and structural details at different granularities. The Shifted Window mechanism, illustrated in Fig. 3 (c), is employed in alternating Swin-Transformer blocks to facilitate cross-window interactions and enhance the model's representational power. After each Swin-Transformer block, a down-sample layer is applied to half the resolution of feature maps. This multi-scale architecture enables the model to process information at multiple scales while progressively reducing the spatial resolution. Each downsampling operation reduces the computational complexity by a factor of 4 while quadrupling the receptive field, enabling the model to capture a large view field.

3.3. Decoding

The decoding module aims to gradually upsample the feature maps and recover the spatial resolution of the output image. It follows a symmetric structure to the encoding module, consisting of four stages. Each decoding stage contains a decoder block, as depicted in Fig. 3 (b), followed by an up-sample layer. The decoder block comprises an LN layer, a W-MSA module, and an MLP, as illustrated in Fig. 3 (d). The W-MSA module in the decoder block operates similarly to its counterpart in the encoder block, capturing local dependencies within windows. An up-sample layer is employed after each decoder block to increase the spatial resolution. Furthermore, skip connections are introduced between corresponding encoder and decoder stages to facilitate the flow of information and aid in recovering fine-grained details. The multi-scale architecture of the decoding module enables the model to gradually refine the reconstructed image while incorporating features from different scales, leading to improved demosaicing performance.

3.4. Reconstruction

The reconstruction module aims to generate the final output RGB image from the upsampled feature maps produced by the decoding module. To achieve this, we apply a depth-to-space operation [41] to recover the resolution. This step is crucial for maintaining image quality and minimizing distortions introduced during preprocessing. The depth-to-space operation rearranges the features and increases the

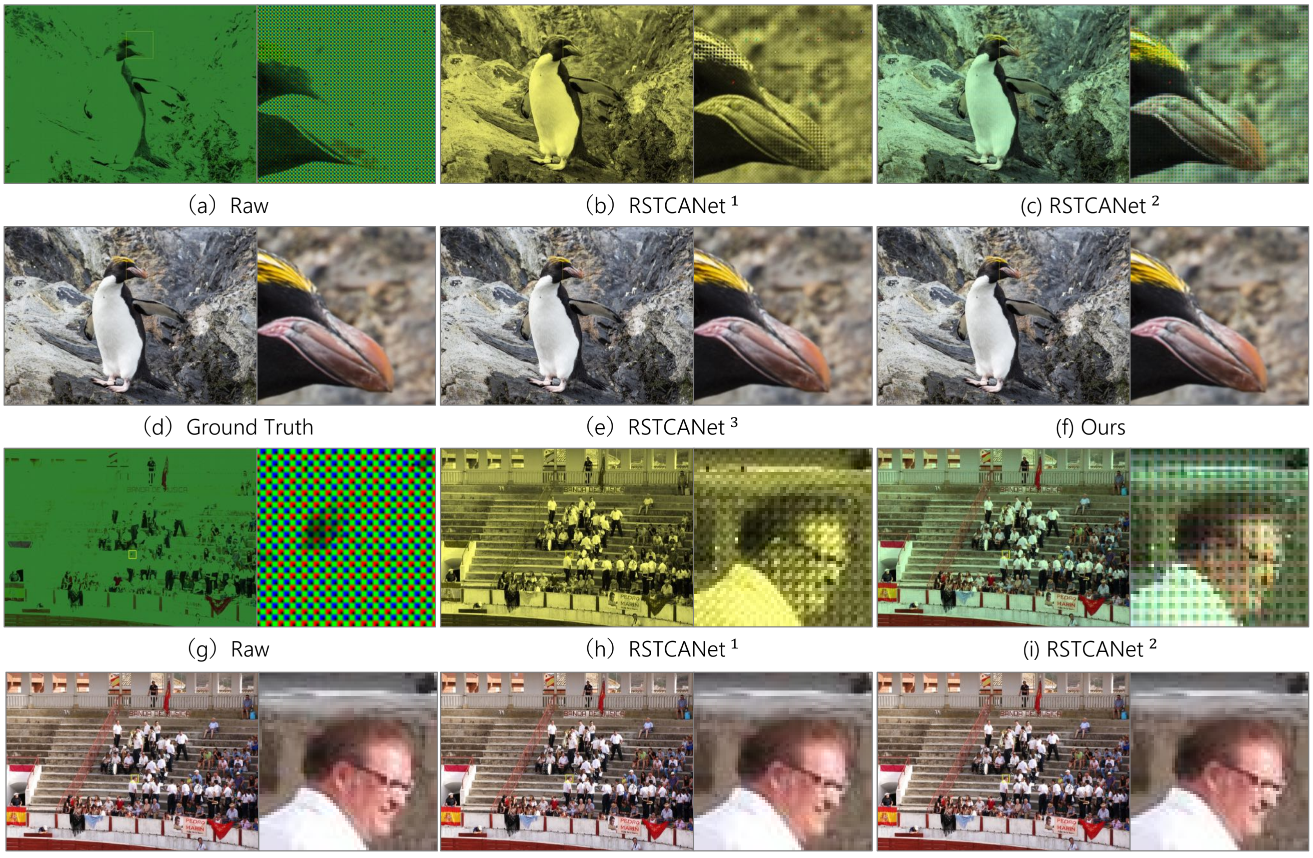


Figure 5. Visualized results of our method and compared method - RSTCANet [51]. Comparison methods 1, 2, and 3, respectively represent the processing directly on the original RAW, processing after converting the original RAW into Bayer Pattern, and the results after fine-tuning RSTCANet [51].

spatial resolution by a factor of s, restoring the original spatial resolution of the input image. Finally, we employ a 1×1 convolution to map the high-dimensional features to the three RGB channels with shape $H \times W \times 3$.

3.5. Loss function

To train our demosaicing network, we employ a two-stage training approach. In the first stage, we use the Charbonnier loss [18] for pre-training. The Charbonnier loss is a commonly used loss function in image processing as shown:

$$\mathcal{L}_{Charbonnier} = \frac{1}{N}\sum_{i=1}^{N}\sqrt{\left(I_{RGB}^{(i)} - I_{GT}^{(i)}\right)^2 + \epsilon^2}, \quad (2)$$

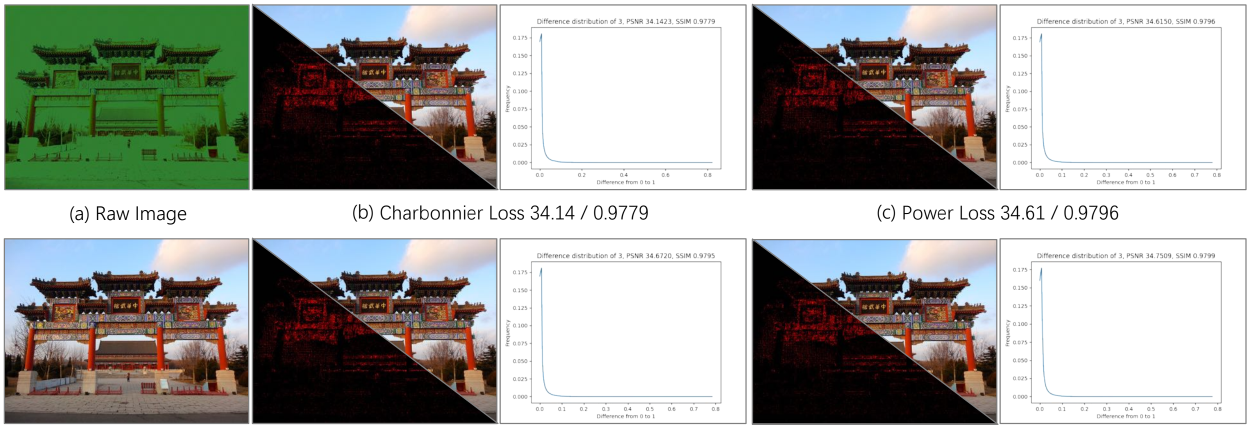
where $I_{RGB}^{(i)}$ and $I_{GT}^{(i)}$ represent the i-th reconstructed RGB image pixel and it's corresponding in the ground-truth image, respectively, N is the total number pixels of images, and ϵ is a small constant (*e.g.*, $1e-3$) added to improve the robustness of the loss function to outliers [18]. Evident from Fig. 2 (I), the Charbonnier loss effectively reduces the difference with larger values during the pre-training stage.

Upon closer examination of the Difference Distribution in Fig. 2 (II), we observe that it comprises two main components: high frequency, *e.g.*, edge areas, at low difference values and low frequency, *e.g.*, smooth areas, at high difference values. This observation suggests that while the model efficiently learns to restore smooth blocks such as backgrounds, it has yet to handle edge areas fully. This shortfall is primarily due to edge differences, highlighting the importance of increasing the gradient magnitude for edge.

In response to this need, we explored two forms of Pixel Focus Loss ($\mathcal{L}_{pf}$) to capture edge-related differences better. One version of the Pixel Focus Loss is described as a piecewise function:

$$\mathcal{L}_{pf}^{p} = \begin{cases} (d/a)^g \cdot b & , 0 < d < a \\ (d-1)(1-b)/(1-a)+1 & , a \leq d \leq 1, \end{cases} \quad (3)$$

where d represents the difference value, a is a threshold parameter, and b and g are scaling factors that control the gradient magnitude. We also introduced another version of



(a) Raw Image (b) Charbonnier Loss 34.14 / 0.9779 (c) Power Loss 34.61 / 0.9796

(d) GT Color Image (e) Exponential Loss (λ = 1.0) 34.67 / 0.9795 (f) Exponential Loss (λ = 1.1) 34.75 / 0.9799

Figure 6. Visualization for output of fin-tuned models with different loss functions. The number values represent PSNR/SSIM.

Table 1. Comparison of our method with the RSTCANet method [51] on the MIPI-Challenge Demosaic dataset. RSTCANet 1, 2, and 3 denote different processing strategies.

Case	Methods	Params (M)	PSNR	SSIM
#1	RSTCANet[1]	7.19	13.2477	0.3590
#2	RSTCANet[2]	7.19	15.7721	0.4234
#3	RSTCANet[3]	7.19	36.2691	0.9659
#4	Our-`Small`	6.22	37.3272	0.9738
#5	Our-`Large`	9.40	**37.4117**	**0.9756**

Pixel Focus Loss defined as:

$$\mathcal{L}_{pf}^{e} = e^{\lambda d} - \lambda d - 1, \tag{4}$$

where λ serves as a hyperparameter, both versions address the issue, providing a comprehensive approach to fine-tuning the model's ability to capture edge areas. We utilize these two Pixel Focus Loss to fine-tune the pre-trained model obtained from the first stage, enhancing the demosaicing performance by emphasizing gradients for edge-related differences. In the experiments section, we delve into the impact of different forms of Pixel Focus Loss and the effects of varying hyperparameters on the experimental results, providing a detailed analysis of our findings. Combining the two-stage training approach, utilizing the Charbonnier loss for pre-training and the proposed Pixel Focus loss for fine-tuning, enables our network to learn high-quality RGB image reconstruction.

4. Experiments

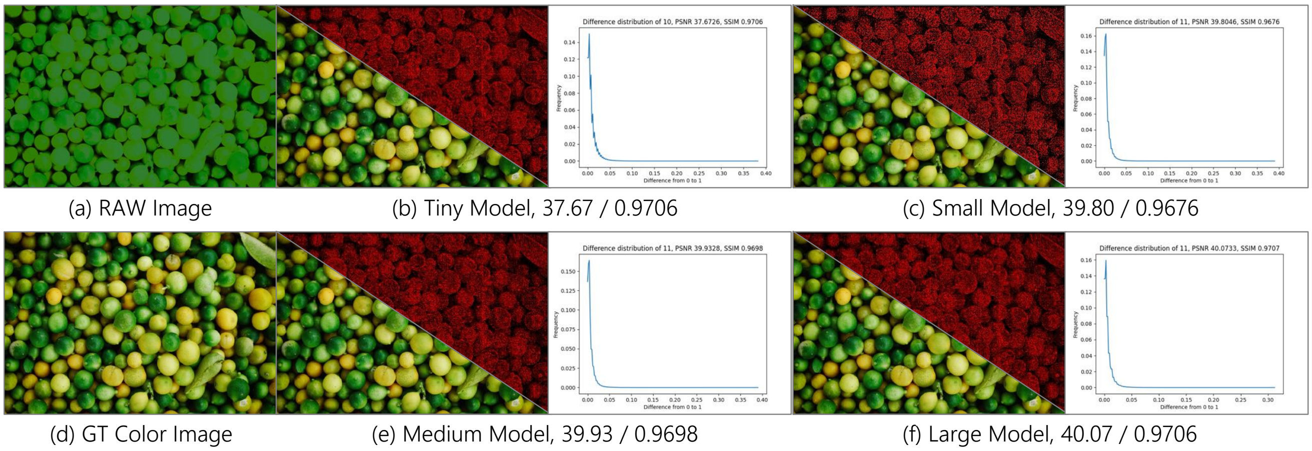
Dataset: Our experiments are based on the Demosaic for HybridEVs Camera dataset at MIPI-Challenge 2024 [34], comprising 900 RAW-Color image pairs with around 2000 × 1500 resolution. In this dataset, 800 RAW-Color pairs are designated for training. 50 color images are allocated for validation and another 50 for testing. Note that the validation and test sets were not released during the competition phase. Consequently, we adapted our approach by utilizing 760 images from the training set and designated 40 images for testing. Within the validation set, 26 pairs of color images are available for local quantitative testing.

Implementation Details: Our experiments were conducted using PyTorch [36] on a server with an Intel(R) Xeon(R) Platinum 8378A CPU and one NVIDIA A800 GPU. The training batch size is one. Each training iteration employs random crop augmentation with patches sized at 640×640. In the first training phase, we utilized the Charbonnier Loss [18], training for 500 epochs with a learning rate starting from $1e-4$ and decreasing to 0 following a cosine function. In the second training phase, we applied the pixel-focus Loss, training for 200 epochs with a learning rate initiating from $1e - 5$ and diminishing to 0. To achieve faster training speeds, we employed mixed precision techniques [33] facilitated by PyTorch.

Evaluation: PSNR and SSIM [11] were utilized as quantitative evaluation metrics. Qualitative results are demonstrated through the visualization of difference maps and difference distributions. Given the high performance of demosaicing methods, directly observing differences between images can be challenging; difference map visualization facilitates addressing this issue.

4.1. Comparison Experiments:

We benchmark our method against the publicly available RSTCANet [51], which is a pioneer in applying the Transformer architecture to demosaicing tasks. In contrast to our method, RSTCANet [51] is tailored exclusively for demosaicing under the Bayer pattern and, consequently, is not inherently equipped to address scenarios with missing pixel



(a) RAW Image (b) Tiny Model, 37.67 / 0.9706 (c) Small Model, 39.80 / 0.9676

(d) GT Color Image (e) Medium Model, 39.93 / 0.9698 (f) Large Model, 40.07 / 0.9706

Figure 7. Visualization of results from model different size. The number values represent PSNR and SSIM, respectively.

Table 2. Ablation for the model size. Params denote the model's parameters, measured in millions. Depth indicates the count of Transformer layers within each block

Case	Model Size	Params (M)	Depth	PSNR	SSIIM
#1	`Tiny`	4.62	2	36.3044	0.9696
#2	`Small`	6.22	4	37.3272	0.9738
#3	`Medium`	7.81	6	37.3798	0.9751
#4	`Large`	9.40	8	**37.4117**	**0.9756**

Table 3. Ablation for the Loss Function.

Case	Loss Function	PSNR	SSIM
#1	w/o	37.4117	0.9756
#2	$\mathcal{L}_{Charbonnier}$	37.4402	0.9758
#3	$\mathcal{L}^{p}_{pf}$	37.8164	0.9767
#4	$\mathcal{L}^{e}_{pf}$ with $\lambda = 1$	37.8360	0.9765
#5	$\mathcal{L}^{e}_{pf}$ with $\lambda = 1.1$	**37.9656**	**0.9770**

values. RSTCANet also falls short in addressing the missing values associated with defects in RAW images. Consequently, employing the RSTCANet method often results in images plagued with noise, as depicted in Fig. 5.

4.2. Ablation and Analytical Experiments:

Ablation for the model depth and size: The ablation study indicates a progressive improvement in image reconstruction quality with increasing model depth, as shown in Tab. 2 and Fig. 7. While the initial increase from a `Tiny` to a `Small` model shows a substantial rise in quality metrics, the growth tapers as the depth extends to `Medium` and `Large`. This pattern suggests a diminishing return on enhancing PSNR and SSIM values with deeper networks, implying an optimal balance between depth and performance.
Ablation for the loss function: The Tab. 3 presented delineates an ablation study examining refining loss functions, particularly in mitigating the challenges posed by long-tail distributions encountered during the latter part of the initial training phase. A secondary training phase spanning 200 epochs was implemented to counteract this issue, utilizing a suite of four variant loss functions. The benchmark was set using the Charbonnier loss function. Contrastive analyses were conducted with a power loss function and two exponential loss functions with λ weights set at 1 and 1.1, as delineated in the second and fourth rows of Tab. 3. Notably, excessively high λ values, such as 2, have been observed to induce instability within the network. Evaluating the impact on PSNR and SSIM scores reveals that alterations to the loss function can significantly affect model performance. The exponential loss functions, particularly with a λ value of 1.1, surpassed the performance of the baseline Charbonnier function. These insights imply that meticulous adjustment of the loss function parameters can effectively overcome optimization hurdles in the advanced stages of training, thereby improving the fidelity of the reconstructed images, as shown in Fig. 6.

5. Conclusion

This paper employs Swin-Transformer and U-Net architecture tailored for the demosaicing task within the CVPR 2024 MIPI-Challenge. A pixel focus loss was designed for a two-stage training to facilitate efficient training. Our model demonstrates advantages over transformer-based methods for event camera demosaicing. The model and the proposed loss function hold the potential to inspire future research in the field of demosaicing and beyond.

Acknowledgment: This work was partially supported by Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2023A03J0008), Education Bureau of Guangzhou Municipality, Guangdong Science and Technology Department, and the Foshan HKUST Projects (FSUST21-FYTRI01A).

References

[1] SM A Sharif, Rizwan Ali Naqvi, and Mithun Biswas. Beyond joint demosaicking and denoising: An image processing pipeline for a pixel-bin image sensor. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 233–242, 2021. 2

[2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 5

[3] Tim Brooks, Ben Mildenhall, Tianfan Xue, Jiawen Chen, Dillon Sharlet, and Jonathan T Barron. Unprocessing images for learned raw denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11036–11045, 2019. 2

[4] Mingdeng Cao, Yanbo Fan, Yong Zhang, Jue Wang, and Yujiu Yang. Vdtr: Video deblurring with transformer. *IEEE Transactions on Circuits and Systems for Video Technology*, 33(1):160–171, 2022. 3

[5] Xinjing Cheng, Peng Wang, and Ruigang Yang. Learning depth with convolutional spatial propagation network. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2361–2379, 2019. 2, 4

[6] Haram Choi, Jeongmin Lee, and Jihoon Yang. N-gram in swin transformers for efficient lightweight image super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2071–2081, 2023. 3

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[8] Guillermo Gallego, Tobi Delbrück, Garrick Orchard, Chiara Bartolozzi, Brian Taba, Andrea Censi, Stefan Leutenegger, Andrew J Davison, Jörg Conradt, Kostas Daniilidis, et al. Event-based vision: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 44(1):154–180, 2020. 1

[9] Zhicheng Geng, Luming Liang, Tianyu Ding, and Ilya Zharkov. Rstt: Real-time spatial temporal transformer for space-time video super-resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17441–17451, 2022. 2, 3

[10] Keigo Hirakawa and Thomas W Parks. Adaptive homogeneity-directed demosaicing algorithm. *Ieee transactions on image processing*, 14(3):360–369, 2005. 2

[11] Alain Hore and Djemel Ziou. Image quality metrics: Psnr vs. ssim. In *2010 20th international conference on pattern recognition*, pages 2366–2369. IEEE, 2010. 7

[12] Andrey Ignatov, Luc Van Gool, and Radu Timofte. Replacing mobile camera isp with a single deep learning model. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 536–537, 2020. 2

[13] Andrey Ignatov, Grigory Malivenko, Radu Timofte, Yu Tseng, Yu-Syuan Xu, Po-Hsiang Yu, Cheng-Ming Chiang, Hsien-Kai Kuo, Min-Hung Chen, Chia-Ming Cheng, et al. Pynet-v2 mobile: Efficient on-device photo processing with neural networks. In *2022 26th International Conference on Pattern Recognition (ICPR)*, pages 677–684. IEEE, 2022. 1

[14] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3320–3329, 2020. 1

[15] Yongcheng Jing, Yiding Yang, Xinchao Wang, Mingli Song, and Dacheng Tao. Turning frequency to resolution: Video super-resolution via event cameras. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7772–7781, 2021. 1

[16] Jungwoo Kim and Min H Kim. Joint demosaicing and deghosting of time-varying exposures for single-shot hdr imaging. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12292–12301, 2023. 2

[17] Filippos Kokkinos and Stamatios Lefkimmiatis. Deep image demosaicking using a cascade of convolutional residual denoising networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 303–319, 2018. 2

[18] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and Ming-Hsuan Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018. 2, 3, 6, 7

[19] Bingchen Li, Xin Li, Yiting Lu, Sen Liu, Ruoyu Feng, and Zhibo Chen. Hst: Hierarchical swin transformer for compressed image super-resolution. In *European conference on computer vision*, pages 651–668. Springer, 2022. 3

[20] Xin Li, Bahadir Gunturk, and Lei Zhang. Image demosaicing: A systematic survey. In *Visual Communications and Image Processing 2008*, pages 489–503. SPIE, 2008. 2

[21] Yong Li, Naipeng Miao, Liangdi Ma, Feng Shuang, and Xingwen Huang. Transformer for object detection: Review and benchmark. *Engineering Applications of Artificial Intelligence*, 126:107021, 2023. 3

[22] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1833–1844, 2021. 2, 3, 4, 5

[23] Lin Liu, Xu Jia, Jianzhuang Liu, and Qi Tian. Joint demosaicing and denoising with self guidance. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2240–2249, 2020. 2, 3, 4

[24] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021. 2, 5

[25] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3202–3211, 2022. 2

[26] Liying Lu, Ruizheng Wu, Huaijia Lin, Jiangbo Lu, and Jiaya Jia. Video frame interpolation with transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3532–3542, 2022. 2, 3

[27] Yunfan Lu, Yiqi Lin, Hao Wu, Yunhao Luo, Xu Zheng, and Lin Wang. All one needs to know about priors for deep image restoration and enhancement: A survey. *arXiv preprint arXiv:2206.02070*, 2022. 2

[28] Yunfan Lu, Guoqiang Liang, and Lin Wang. Learning inr for event-guided rolling shutter frame correction, deblur, and interpolation. *arXiv preprint arXiv:2305.15078*, 2023. 1

[29] Yunfan Lu, Zipeng Wang, Minjie Liu, Hongjian Wang, and Lin Wang. Learning spatial-temporal implicit neural representations for event-guided video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1557–1567, 2023. 1, 3

[30] Vyacheslav Lyashenko, Svitlana Sotnik, and Volodymyr Manakov. Modern cad/cam/cae systems: brief overview. 2021. 2

[31] Henrique S Malvar, Li-wei He, and Ross Cutler. High-quality linear interpolation for demosaicing of bayer-patterned color images. In *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages iii–485. IEEE, 2004. 1, 2

[32] Nico Messikommer, Stamatios Georgoulis, Daniel Gehrig, Stepan Tulyakov, Julius Erbach, Alfredo Bochicchio, Yuanyou Li, and Davide Scaramuzza. Multi-bracket high dynamic range imaging with event cameras. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 547–557, 2022. 1

[33] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. *arXiv preprint arXiv:1710.03740*, 2017. 7

[34] MIPI Challenge 2024. Mobile intelligent photography and imaging workshop 2024. https://mipi-challenge.org/MIPI2024/, 2024. 2, 7

[35] Genady Paikin, Yotam Ater, Roy Shaul, and Evgeny Soloveichik. Efi-net: Video frame interpolation from fusion of events and frames. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1291–1301, 2021. 1

[36] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 7

[37] Guocheng Qian, Yuanhao Wang, Jinjin Gu, Chao Dong, Wolfgang Heidrich, Bernard Ghanem, and Jimmy S Ren. Rethinking learning-based demosaicing, denoising, and super-resolution pipeline. In *2022 IEEE International Conference on Computational Photography (ICCP)*, pages 1–12. IEEE, 2022. 2

[38] Henri Rebecq, René Ranftl, Vladlen Koltun, and Davide Scaramuzza. High speed and high dynamic range video with an event camera. *IEEE transactions on pattern analysis and machine intelligence*, 43(6):1964–1980, 2019. 1

[39] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer, 2015. 2, 4, 5

[40] Mehdi SM Sajjadi, Raviteja Vemulapalli, and Matthew Brown. Frame-recurrent video super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6626–6634, 2018. 4

[41] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 5

[42] Zhihao Shi, Xiangyu Xu, Xiaohong Liu, Jun Chen, and Ming-Hsuan Yang. Video frame interpolation transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17482–17491, 2022. 2

[43] Chung-Yen Su. Highly effective iterative demosaicing using weighted-edge and color-difference interpolations. *IEEE Transactions on Consumer Electronics*, 52(2):639–645, 2006. 2

[44] Lei Sun, Christos Sakaridis, Jingyun Liang, Qi Jiang, Kailun Yang, Peng Sun, Yaozu Ye, Kaiwei Wang, and Luc Van Gool. Event-based fusion for motion deblurring with cross-modal attention. In *European conference on computer vision*, pages 412–428. Springer, 2022. 1

[45] Nai-Sheng Syu, Yu-Sheng Chen, and Yung-Yu Chuang. Learning deep convolutional networks for demosaicing. *arXiv preprint arXiv:1802.03769*, 2018. 2

[46] Stepan Tulyakov, Daniel Gehrig, Stamatios Georgoulis, Julius Erbach, Mathias Gehrig, Yuanyou Li, and Davide Scaramuzza. Time lens: Event-based video frame interpolation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16155–16164, 2021. 1

[47] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 4

[48] Yusheng Wang, Yunfan Lu, Ye Gao, Lin Wang, Zhihang Zhong, Yinqiang Zheng, and Atsushi Yamashita. Efficient video deblurring guided by motion magnitude. In *European Conference on Computer Vision*, pages 413–429. Springer, 2022. 3

[49] Wei Wu, Shuming Hu, Pengxiang Xiao, Sibin Deng, Yilin Li, Ying Chen, and Kai Li. Video quality assessment based on swin transformer with spatio-temporal feature fusion and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1846–1854, 2023. 2

[50] Wenzhu Xing and Karen Egiazarian. End-to-end learning for joint image demosaicing, denoising and super-resolution. In

Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 3507–3516, 2021. 2

[51] Wenzhu Xing and Karen Egiazarian. Residual swin transformer channel attention network for image demosaicing. In *2022 10th European Workshop on Visual Information Processing (EUVIP)*, pages 1–6. IEEE, 2022. 2, 6, 7

[52] Fang Xu, Lei Yu, Bishan Wang, Wen Yang, Gui-Song Xia, Xu Jia, Zhendong Qiao, and Jianzhuang Liu. Motion deblurring with real events. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2583–2592, 2021. 1

[53] Wu Yaqi, Fan Zhihao, Chu Xiaofeng, Ren Jimmy S., Li Xiaoming, Yue Zongsheng, Li Chongyi, Zhou Shangcheng, Feng Ruicheng, Dai Yuekun, Yang Peiqing, Loy Chen Change, et al. Mipi 2024 challenge on demosaic for hybridevs camera: Methods and results. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024. 1, 2

[54] Linwei Yue, Huanfeng Shen, Jie Li, Qiangqiang Yuan, Hongyan Zhang, and Liangpei Zhang. Image super-resolution: The techniques, applications, and future. *Signal processing*, 128:389–408, 2016. 3

[55] Haijin Zeng, Kai Feng, Shaoguang Huang, Jiezhang Cao, Yongyong Chen, Hongyan Zhang, Hiep Luong, and Wilfried Philips. Msfa-frequency-aware transformer for hyperspectral images demosaicing. *arXiv preprint arXiv:2303.13404*, 2023. 2

[56] Kaihao Zhang, Wenqi Ren, Wenhan Luo, Wei-Sheng Lai, Björn Stenger, Ming-Hsuan Yang, and Hongdong Li. Deep image deblurring: A survey. *International Journal of Computer Vision*, 130(9):2103–2130, 2022. 3

[57] Lei Zhang and Xiaolin Wu. Color demosaicking via directional linear minimum mean square-error estimation. *IEEE Transactions on Image Processing*, 14(12):2167–2178, 2005. 2

[58] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016. 3

[59] Xu Zheng, Yexin Liu, Yunfan Lu, Tongyan Hua, Tianbo Pan, Weiming Zhang, Dacheng Tao, and Lin Wang. Deep learning for event-based vision: A comprehensive survey and benchmarks. *arXiv preprint arXiv:2302.08890*, 2023. 1