Rethinking Learning-based Demosaicing, Denoising, and Super-Resolution Pipeline

Guocheng Qian[1*], Yuanhao Wang[1*], Jinjin Gu[2], Chao Dong[3,4], Wolfgang Heidrich[1], Bernard Ghanem[1], Jimmy S. Ren[5,6]

Abstract—Imaging is usually a mixture problem of incomplete color sampling, noise degradation, and limited resolution. This mixture problem is typically solved by a sequential solution that applies demosaicing (DM), denoising (DN), and super-resolution (SR) sequentially in a fixed and predefined pipeline (execution order of tasks), DM→DN→SR. The most recent work on image processing focuses on developing more sophisticated architectures to achieve higher image quality. Little attention has been paid to the design of the pipeline, and it is still not clear how significant the pipeline is to image quality. In this work, we comprehensively study the effects of pipelines on the mixture problem of learning-based DN, DM, and SR, in both sequential and joint solutions. On the one hand, in sequential solutions, we find that the pipeline has a non-trivial effect on the resulted image quality. Our suggested pipeline DN→SR→DM yields consistently better performance than other sequential pipelines in various experimental settings and benchmarks. On the other hand, in joint solutions, we propose an end-to-end Trinity Pixel Enhancement NETwork (TENet) that achieves the state-of-the-art performance for the mixture problem. We further present a novel and simple method that can integrate a certain pipeline into a given end-to-end network by providing intermediate supervision using a detachable head. Extensive experiments show that an end-to-end network with the proposed pipeline can attain only a consistent but insignificant improvement. Our work indicates that the investigation of pipelines is applicable in sequential solutions, but is not very necessary in end-to-end networks.

Index Terms—Image Demosaicing, Image Denoising, Image Super-resolution, ISP, Deep Learning

1 Introduction

Obtaining high-quality, high-resolution images has attracted increasing attention. Acquiring such images is difficult in practice due to hardware limitations, especially for mobile devices. First, most digital cameras capture images using a single image sensor overlaid with a color filter array (*e.g.* Bayer pattern), which causes incomplete color sampling, *i.e.* resulting in mosaic images instead of RGB images. Second, images taken directly from the image sensor are inevitably noisy. Third, typical mobile devices are equipped with limited pixel numbers and lenses with fixed and short focal lengths, which makes imaging of distant or small objects challenging and limits image resolution. The real-shot image captured by an iPhone X shown in Fig. 1 shows unnatural colorization, noise, and loss of detail due to these limitations. Demosaicing (DM) [1], denoising (DN) [2] and super-resolution (SR) [3] are the three fundamental tasks that have been studied and included in image processing pipelines (ISPs[1]) to resolve the hardware limitations mentioned above and to improve image quality.

Deep learning technologies [4], [5], [6] have recently led to breakthrough progress in DN, DM, and SR algorithms, and have spawned commercial products using learning-based image processing such as modern mobile phones (iPhone, Google Pixel, *etc.*). Despite the achievement of deep learning in each task, *imaging is usually a mixture problem of incomplete color sampling, noise degradation, and resolution limitation*. The combination of DN, DM, and SR is more common and more complicated than any single problem in practical application.

Previous methods handle the mixture problem through a sequential solution that performs DM, DN, and SR independently in a predefined and fixed order: DM→DN→SR [7], *i.e.* firstly DM, followed by DN, and then SR. Recent methods instead show a trend in performing DN and SR in mosaic space before DM [8], [9], [10]. However, these works do not consider the important but under-explored mixture problem of DN, DM, and SR. Furthermore, it is not clear how significant the execution order of tasks (*i.e.* pipeline) is to the performance of this mixture problem.

In this paper, we analyze the characteristics of DN, DM, and SR and the behaviors of their interactions. We find that issues caused by interactions between tasks occur when the corresponding algorithms are applied sequentially to solve the mixture problem. For example, superresolving a demosaiced image will magnify artifacts (*e.g.* moiré) introduced by the DM algorithm (see Fig. 4). *We propose a novel image processing pipeline: DN→SR→DM, for sequential solutions*. We find that the proposed pipeline can alleviate problems caused by task interactions to a great extent. Extensive experiments of learning-based DN, DM, and SR show that our pipeline can *consistently* improve image quality of sequential solutions, regardless of architecture, dataset, and SR factor (see Sec. 6).

We further study the effect of pipelines in joint solutions (end-to-end networks) for the mixture problem. We first

- [1]*KAUST,* [2]*The University of Sydney,* [3]*Shanghai AI Laboratory,* [4]*Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences* [5]*SenseTime Research,* [6]*Qing Yuan Research Institute, Shanghai Jiao Tong University,*
- **Equal contribution.*

1. ISP can be the abbreviation for image processing pipeline or image signal processor. We use these terms interchangeably.

propose a **T**rinity **E**nhancement **Net**work (TENet++[2]) to address the mixture problem. We then present a simple yet effective way that enforces an end-to-end network to follow a certain pipeline by providing intermediate supervision. Through experiments on TENet++ and other architectures [11], [12], [13], we notice marginal but consistent improvements after inserting the proposed pipeline. Our studies suggest that the investigation of pipelines in end-to-end networks can improve the performance but is not very necessary considering the insignificant improvement.

Contributions: **(1)** We are the first to propose and analyze the mixture problem of learning-based denoising, demosaicing, and super-resolution. **(2)** We suggest a new pipeline: DN→SR→DM for solving the mixture problem of DN, DM, and SR. Extensive experiments show that the proposed pipeline can consistently improve performance for sequential solutions. **(3)** We propose an end-to-end network named Trinity Pixel Enhancement Network (TENet++) that achieves SOTA performance for joint DN, DM, and SR. **(4)** We show how to make an end-to-end network follow a certain pipeline. We indicate an insignificant effect of pipelines on end-to-end networks. **(5)** We notice that there is a lack of full-color sampled datasets in the literature. We contribute a new real-world dataset, namely *PixelShift200*, which consists of red, green, and blue channels without the need for color interpolation. We demonstrate the benefits of PixelShift200 in training and evaluating raw image processing tasks. Code, models, and our contributed PixelShift200 dataset are available at https://github.com/guochengqian/TENet.

2 Related Work

Demosaicing. Digital cameras take subsampled color measurements at alternating pixel locations. The resulting images of the subsampled measurements are named mosaic images. The mosaic images are then interpolated to create full-color images with per-pixel red, green, and blue information by a so-called demosaicing (DM) process. Early DM methods are model-based [14], [15], [16], which focus on the construction of filters (*e.g.* edge-aware interpolation) and image priors (*e.g.* chrominance continuity). Model-based methods are still commonly used in camera systems and software; *e.g.* the image processing library DCRaw utilizes [15]. Pioneering works also explored data-driven methods [17], [18] that learn a mapping from a raw image to an RGB image. Recently, deep learning has achieved overwhelming performance in DM. [4] presented DemosaicNet, a deep convolutional neural network (CNN)-based DM algorithm that outperforms the previous methods by a large margin. Following DemosaicNet, many works [19], [20] design different architectures to improve the demosaicing quality.

Denoising. Noise is inevitable during the imaging process. Early denoising (DN) methods exploited image priors, such as content variance [21], self-similarity [22], and sparse representation [23] for image denoising. The most recent denoisers are entirely data-driven, consisting of CNNs trained to recover noisy images to noise-free targets [5], [24]. Despite the effectiveness of these learning-based denoisers on synthesized benchmarks [25], they generalize poorly to real-shot images due to their oversimplified assumption that noise is additive, white, and Gaussian [26]. While the noise pattern of a color image is complex because of nonlinear image processing (DM, color mapping, and compression), the noise patterns on raw images are well studied. [27] characterized how sensor noise primarily comes from two sources: Poisson noise (shot noise) and Gaussian noise (read noise). To improve the generalization ability of deep denoisers, [9], [28] proposed denoising on raw images using Poisson-Gaussian noise, which outperformed previous methods on the real-world image denoising dataset DND [29]. In this paper, we find that denoising RAW images directly yields higher quality, regardless of the network architecture.

Super-resolution. Due to the limited sensor size, image resolution is usually not as high as desired. Image SR aims to recover a high-resolution (HR) image from its low-resolution (LR) version. Previously, example-based SR methods [30], [31] that exploit the self-similarity property provided state-of-the-art performance. Recently, learning-based methods [6], [32] developed the CNN-based SR algorithms SRCNN and FSRCNN, outperforming example-based methods. After these seminal works, many learning-based SR methods have emerged [11], [33]. However, most of them focus on color image SR. Only a few works have paid attention to the SR of raw images [10], [12], [34].

ISP and Mixture Problem. Image processing is always accompanied by a mixture problem of DN, DM and SR. An ISP is embedded in a modern camera to perform all these tasks. Most ISPs solve tasks independently and sequentially through the predefined pipeline DM→DN→SR [7]. Although some previous works proposed new pipelines, such as performing DN before DM [8], [9], less attention has been paid to the execution order of joint DN, DM, and SR, especially since many ISP methods in the deep learning era are now learning-based and leverage end-to-end algorithms [10], [12], [13], [35], [36]. These end-to-end solutions map a raw image to a desired RGB image directly, without focusing on the pipeline. In this work, we diverge from the common architecture engineering in the area of image processing and rethink the mixture problem of DM, DN and SR from a holistic perspective, and more especially, the execution order (pipeline) of tasks.

3 Methodology

3.1 A New Pipeline for DN, DM and SR

We propose a new image processing pipeline, DN→SR→DM, that significantly improves the image quality of sequential solutions for the joint problem of DM, DN and SR. For a given noisy LR raw image M_n^{LR}, our pipeline obtains the final HR color image I^{HR} from M_n^{LR} using a composite function as follows:

$$I^{HR} = \mathcal{C}_M(\mathcal{S}_M(\mathcal{D}_M(M_n^{LR}))), \tag{1}$$

where $\mathcal{C}_M$ is the demosaicing function ($\mathcal{C}$ denotes "**c**olorize"), $\mathcal{S}_M$ the **S**R function for mosaic images, and $\mathcal{D}_M$ the **d**enoising function for mosaic images. M and subscript $_M$ stand for "mosaic", while subscript $_n$ indicates noisy. We first perform DN on the noisy raw mosaic image

2. We add ++ after TENet to avoid confusion with the outdated architecture used in previous arXiv version of this paper

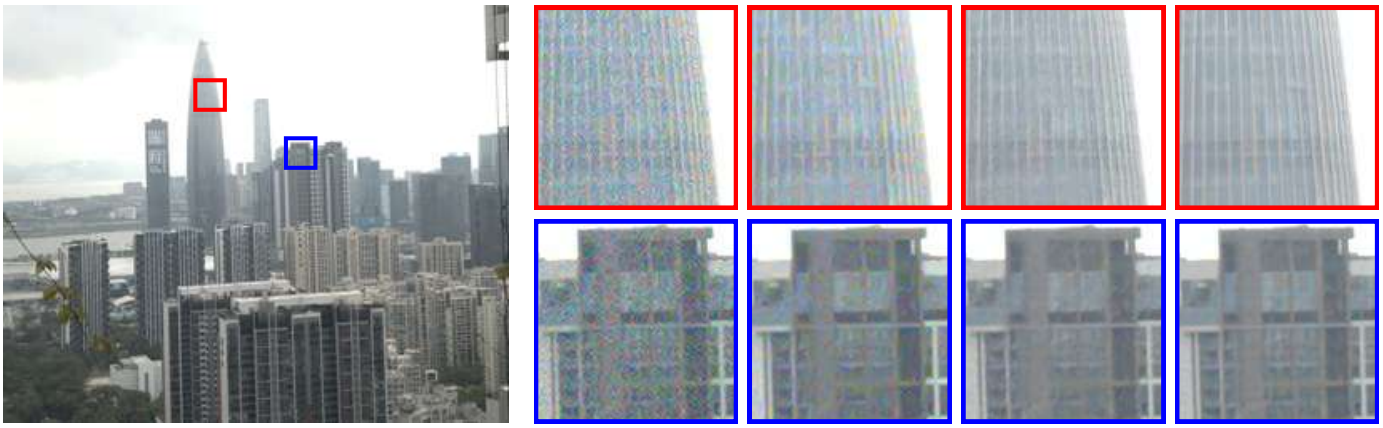


Fig. 1: **Qualitative comparisons for joint DM, DN, and SR (×2) on a real raw image captured by an iPhone X.** Our TENet++ delivers a more visually appearing result compared to popular software DCRaw and Camera Raw and state-of-the-art JDnDmSR [13], producing less color distortions and more fine-grained details. The output of DCRaw and Camera Raw is superresolved by a SR model implemented by the same 6 RRDB blocks as TENet++ for a fair comparison (Sec. 5).

to obtain its noise-free version, $M^{LR} = D_M(M_n^{LR})$. We then adopt S_M to superresolve the LR mosaic image and obtain a HR mosaic image, $M^{HR} = \mathcal{S}_M(M^{LR})$. Finally, we use DM to interpolate M^{HR} to a full-color HR image, $I^{HR} = \mathcal{C}_M(M^{HR})$.

Why perform DN at the first stage? DN is usually performed after DM in a typical ISP. We propose DN first for three reasons: (1) the noise model for raw images has been well studied (Gaussian-Poisson distribution). The quality of DN is higher for raw images than for color images. (2) The existence of noise adds complexity to subsequent tasks. Noise has a high possibility of hiding color information and destroying textures, depending on the noise level. Processing a noisy image will result in unwanted artifacts in most cases. For example, Fig. 4 (DM→DN→SR) showcases that demosaicing a noisy image is prone to moiré. (3) Image processing prior to DN will degrade the noise pattern and complicate denoising. For example, SR will destroy the noise distribution and make removal of noise from the super-resolved image extremely difficult. Fig. 4 (DM→SR→DN) shows such an example, where obvious noise appears.

Why perform SR before DM? Previous ISPs usually first demosaic a raw image into a color image and then perform SR. We suggest super-resolving the raw image to a higher resolution before conducting DM. In other words, super-resolution in our suggested pipeline is performed on mosaic images instead of RGB images. Our proposed pipeline has at least two advantages: (1) demosaicing a higher resolution raw image yields fewer artifacts than demosaicing a lower resolution image. A DM algorithm usually introduces conspicuous artifacts (zippering, color moiré, and blurring) in the high-frequency texture regions, especially when the input resolution is low. These artifacts are alleviated when DM is applied to an image with higher resolution. (2) The artifacts caused by super-resolving the defects of a demosaiced image can be avoided in our pipeline. As shown in Fig. 4, DN→SR→DM that performs SR before DM alleviates color distortion and moiré compared to its counterpart DN→DM→SR.

3.2 Inserting Our Pipeline into An End-to-end Network

Despite the effectiveness of the proposed pipeline, simply performing multiple tasks sequentially and independently, as shown in Equation 1 reduces performance. For example, DN will introduce blurring in subsequent tasks. An important reason for this performance drop is that no appropriate model can perfectly handle the intermediate state. The intermediate state refers to the temporal result after previous processing and usually involves complex task-related defects that affect subsequent tasks. With the advent of deep learning-based methods, we can address complicated multitask problems in an end-to-end manner, *i.e.* a "joint solution". Although the joint solution has shown impressive performance in a variety of tasks [37], [38], [39], it is still underexplored for joint DN, DM, and SR. The most recent works [10], [12], [13], [40] focused on such a mixture problem. However, most of them simply treat the whole network as a black box, without considering the pipeline inside. Their methods just learn a mapping from the noisy LR raw image to the HR color image, with the final target (the output of a camera ISP) serving as supervision. We denote this type of one-stage end-to-end black-box network as **E2ENet**, whose architecture is illustrated in Fig. 2a. We denote E2ENet's pipeline as DN+SR+DM.

We show how to make an end-to-end network follow a certain pipeline to solve the mixture problem instead of just learning a one-stage mapping. With the joint solution, we can *simplify the sequential pipeline DN→SR→DM as DN+SR→DM*. Compared to E2ENet (DN+SR+DM), we assign a specific task to each component of the network. Our network performs joint DN and SR in the first stage, followed by DM in the final stage. We achieve this pipeline by providing intermediate supervision when training an end-to-end network. We denote the mapping function of joint DN and SR as $\mathcal{F}_M$, and the DM mapping as $\mathcal{C}_M$. $\mathcal{F}_M$ and $\mathcal{C}_M$ can be trained jointly. The l_1-norm loss for the final output is calculated by:

$$\mathcal{L}_{joint} = \|\mathcal{C}_M(\mathcal{F}_M(M_n^{LR})) - I_{gt}^{HR}\|, \tag{2}$$

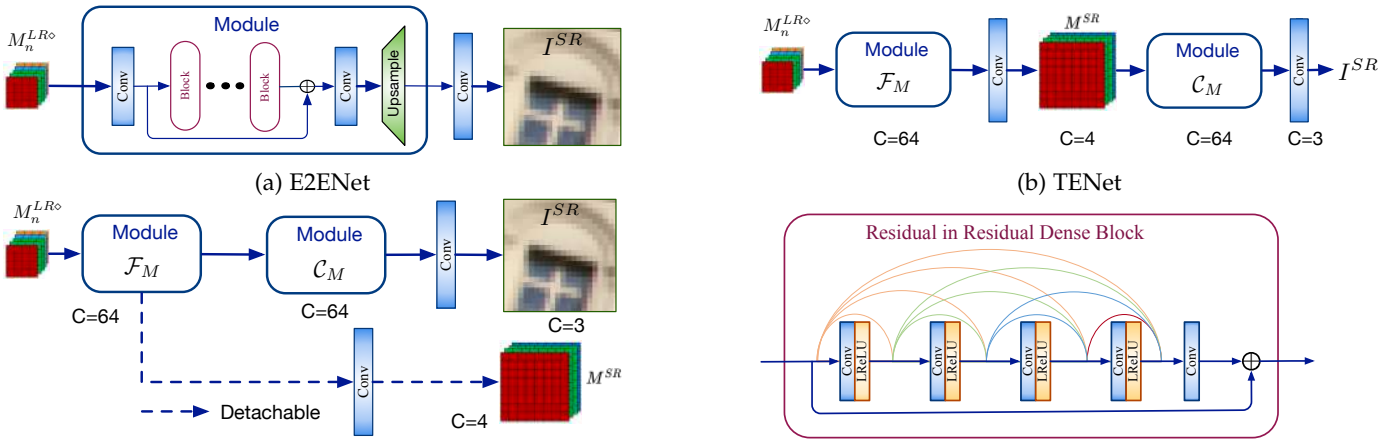


Fig. 2: **Architecture of our TENet++ (c).** (a) E2ENet is a one-stage end-to-end network that learns a mapping from the noisy LR raw image to the HR color image directly using a single module. (b) Our naive version of Trinity Enhancement Network, denoted as TENet, consists of two main components: a joint denoising and super-resolution module $\mathcal{F}_M$ and a demosaicing module $\mathcal{C}_M$. Each module shares the same architecture as (a), which composes convolutional layers to extract features and an upsampling layer to interpolate features. This two-component design makes the network follow a certain pipeline (supersolve raw image before demosaicking) for the joint DN, DM, and SR problem, and facilitates optimization by providing intermediate supervision, compared to (a). However, TENet suffers from the bottleneck issue (channel size is dropped to $C = 4$ in the middle of the network). (c) Our proposed TENet++ where a detachable convolution layer is adopted after $\mathcal{F}_M$ for reconstructing the high-resolution raw image (M^{SR}). This detached layer is activated during training, thus eschewing TENet++ from the bottleneck issue, and is detached in inference. (d) The default block (RRDB [33]) used in TENet++.

where I_{gt}^{HR} represents the ground-truth HR color image of the input LR noisy raw image M_n^{LR}. We further construct an intermediate output M^{HR} (the superresolved mosaic image) and propose an intermediate loss, $\mathcal{L}_{SR}$, as follows:

$$\mathcal{L}_{SR} = \|\mathcal{F}_M(M_n^{LR}) - M_{gt}^{HR}\|, \tag{3}$$

where M_{gt}^{HR} represents the ground-truth HR noise-free mosaic image and the output of $\mathcal{F}_M(M_n^{LR})$ is M^{SR}. The $\mathcal{L}_{SR}$ loss makes the first part of the network focus on joint DN and SR, and the second part on DM. The $\mathcal{L}_{joint}$ loss controls the fidelity of the final output. The final objective function is the sum of two loss terms:

$$\mathcal{L} = \mathcal{L}_{joint} + \mathcal{L}_{SR}, \tag{4}$$

While DN→SR→DM outperforms other pipelines in sequential solutions, DN+SR→DM is the best overall in joint solutions (despite the marginal improvements). Note here adding additional denoising supervision is not beneficial to the performance as shown in our experiment. This demonstrates that the essence of our proposed pipeline is to perform DN and SR in mosaic space, not RGB space, which are our core arguments for both pipelines (see Sec. 3.1).

3.3 Trinity of Pixel Enhancement Network

The naive solution to achieve the DN+SR→DM pipeline is to concatenate two subnetworks $\mathcal{F}_M$ and $\mathcal{C}_M$ in the network backbone and actually produce M^{SR} in the middle of the network, as shown in Fig. 2b. This is the architecture that we used in the preprint version of our work and is denoted TENet. Unfortunately, this solution will face performance drops due to a bottleneck issue. The bottleneck arises as the channel size C is decreased from the latent space (*e.g.* $C = 64$) to the raw image space ($C = 4$) to yield the SR raw image. To solve this issue, we present **T**rinity of Pixel **E**nhancement **N**etwork (**TENet++**). TENet++ leverages an attachable branch to provide additional supervision during training. The attachable branch is implemented by a single convolutional layer to map the feature from the latent space to the raw image space. The architecture of TENet++ is illustrated in Fig. 2c. The noisy LR mosaic image M_n^{LR} with size $H \times W$ is reshaped to a four-channel image (red, green, green, blue) with size $\frac{H}{2} \times \frac{W}{2} \times 4$. The noise variance for each channel is concatenated into the reshaped raw image. The eight-channel input is denoted $M_n^{LR\diamond}$, which is passed to the TENet++ backbone. TENet++ consists of two components in its backbone: a joint denoising and super-resolution module $\mathcal{F}_M$ and a demosaicing module $\mathcal{C}_M$. $\mathcal{F}_M$ and $\mathcal{C}_M$ share the same structure as the module used in E2ENet (detailed in Fig. 2a). Both $\mathcal{F}_M$ and $\mathcal{C}_M$ are composed of a convolution layer to transform features, $N/2$ convolutional blocks to extract features, and a convolution layer with an upsampling layer to interpolate features. Note N is the total number of blocks in TENet++ and is set to 12 by default. The upsampling ratio of $\mathcal{F}_M$ is the SR ratio (2 by default), while the upsampling ratio of $\mathcal{C}_M$ equals 2 since $\mathcal{C}_M$ is the demosaicing module to interpolate colors. We employ the Residual in Residual Dense Block (RRDB) proposed in ESRGAN [33] (see Fig. 2d) to implement the blocks used in each module by default. A pixel shuffle layer [41] is used to upsample the feature maps for DM and SR. A detachable layer is attached to $\mathcal{F}_M$ to produce the in-

termediate output M^{SR} for additional training supervision, and can be removed during testing. In our experiment, the number of RRDB modules for both $\mathcal{F}_M$ and $\mathcal{C}_M$ is set to 6.

Compared to E2ENet, TENet++ has two major differences: (1) the upsampling layer for SR is moved forward to the middle of the network (end of $\mathcal{F}_M$) to yield superresolved raw images; (2) intermediate supervision is provided. From a theoretical perspective, we hypothesize that reasonable intermediate supervision (superresolved raw image in our case) yields a limited solution space with good local minima, thus leading to an eased optimization. We show the effectiveness of TENet++ over E2ENet through extensive experiments in Sec. 5.

4 PixelShift200 Dataset

4.1 Motivation of PixelShift200

Previous learning-based DM algorithms train their networks on incompletely color-sampled datasets such as DIV2K [42] and ImageNet [43], where they take color images demosaiced from incomplete color samples (Bayer images) as Ground Truth and synthesize the mosaic images as input [9], [28], [44]. However, this scheme has three main issues: (1) the color images are interpolated by the camera ISP, which introduces DM artifacts caused by incomplete color sampling. These artifacts will also be learned if a DM model is trained on them. (2) The DM model trained on such synthesized dataset only learns an "average" DM algorithm used in the camera's ISP. And (3) the synthesized raw images only have a depth of 8-bit and therefore suffer from information loss, compared to normal 14-bit real raw images. Thus, real-world, high-resolution, uncompressed image datasets with full-color sampling are needed.

We contribute a novel real-world dataset *PixelShift200*, which contains 200 4K-resolution full-color sampled images. The color information in red, green, and blue in 14-bit for each pixel is known in our dataset without any domosaicing. PixelShift200 was collected using the pixel shift technique [45] embedded in the camera we use (see Sec. 4.2). This technique takes four samples of the same image at the same time, and physically controls the camera sensor to precisely move one pixel horizontally or vertically at each sampling. The four samples are then combined to directly obtain all the color information for each pixel. Refer to Fig. 3 for an example of the pixel shift process. The pixel shift technique ensures that the sampled images follow the distribution of natural images.

Due to full-color sampling, our collected images in PixelShift200 are almost free of artifacts compared to the images interpolated from mosaic inputs. Fig. 3 compares a color image obtained by the pixel shift technique with the output of the well-known raw processing software, Adobe Camera Raw (version 12.3). The pixel shift combines the four raw images into a single full-color sampled image, while Camera Raw interpolates the first sample using the built-in demosaicing algorithm. It is worth noting that the pixel shift technique generates much less aliasing (see the letter "K" in the first row) and fewer moirés (see the barcode in the second row). In Sec. 6.2, we demonstrate training raw image processing networks on our PixelShift200 dataset will produce better image quality than training the same network on the incompletely color sampled dataset (*e.g.* DIV2K [42]). We highlight that, as far as we are aware, we are the first to collect such a full-color sampled dataset. PixelShift200 is useful for training raw image processing methods and can also be used as a unique benchmark for demosaicing-related tasks.

4.2 PixelShift200 Collection Procedure

We collected PixelShift200 dataset with a Sony ILCE-7RM3 digital camera, which includes the pixel shift technique in its camera system [45]. To avoid serious noise, we mounted a lens with fixed focal length and aperture (Zeiss FE $50\ mm/1.4$) with low photosensitivity (ISO 100 or less). To reduce motion parallax, we controlled the depth of the scene field to a small range and held the camera with a heavy tripod. PixelShift200 consists of 180 4K resolution images for training and 20 1K resolution images for testing. The testing set is selected to cover a wide range of scenes. As data augmentation, the training samples were cropped into 9444 overlapping patches of size 512×512.

5 Experiments

5.1 Experimental Setup

Data Preprocessing We perform a bicubic downsampling kernel (denoted as $\mathcal{S}_C^{-1}$), a mosaic kernel [9] ($\mathcal{C}_M^{-1}$), and then the Gaussian-Poisson noise model [27] to generate LR noisy raw images M_n^{LR} as input from HR color images I^{HR} in pixelshift200:

$$M_n^{LR} = \mathcal{C}_M^{-1}(\mathcal{S}_C^{-1}(I^{HR})) + n \tag{5}$$

where the noise term n is sampled from:

$$n \sim \mathcal{N}(\mu = 0, \sigma^2 = \lambda_{read} + \lambda_{shot} M^{LR}) \tag{6}$$

λ_{read} and λ_{shot} are the read and shot noise levels of a given raw image. The noise variance n' is given by $(\lambda_{read} + \lambda_{shot} M_n^{LR})$.

In Pixelshift200, we generated the random Gaussian-Poisson noise in both training and testing. Note that the noise was generated on the fly during training and was sampled once and fixed for the testing samples. Noise levels follow the same range as the real-shot denoising benchmark dataset, DND [29]. Random rotation and flipping were used as data augmentation during training. The output of the model after the whole pipeline is the RGB image in the linear color space. The black level subtraction is conducted as the pre-processing step for each raw image and is performed before DN, DM, and SR. The white balance and color mappings were read from the raw images and applied to the final outputs to transform them into standard RGB space (sRGB).

Metric For quantitative experiments, we use PSNR (↑), SSIM (↑), and FreqGain (↓) [4] to measure overall fidelity, overall structure similarity, and fine-grained artifacts. Note that FreqGain is the metric we modify from [4], which was proposed to detect moirés. We revise it to a scalar version by averaging the positive logarithmic values of the frequency gains. The formula of FreqGain is the following:

$$\rho = \text{avg}\left(\text{ReLU}\left(\log\left(\frac{|\mathcal{F}_{\mathcal{O}}(\omega)|^2 + \epsilon}{|\mathcal{F}_{\mathcal{I}}(\omega)|^2 + \epsilon}\right)\right)\right) \tag{7}$$

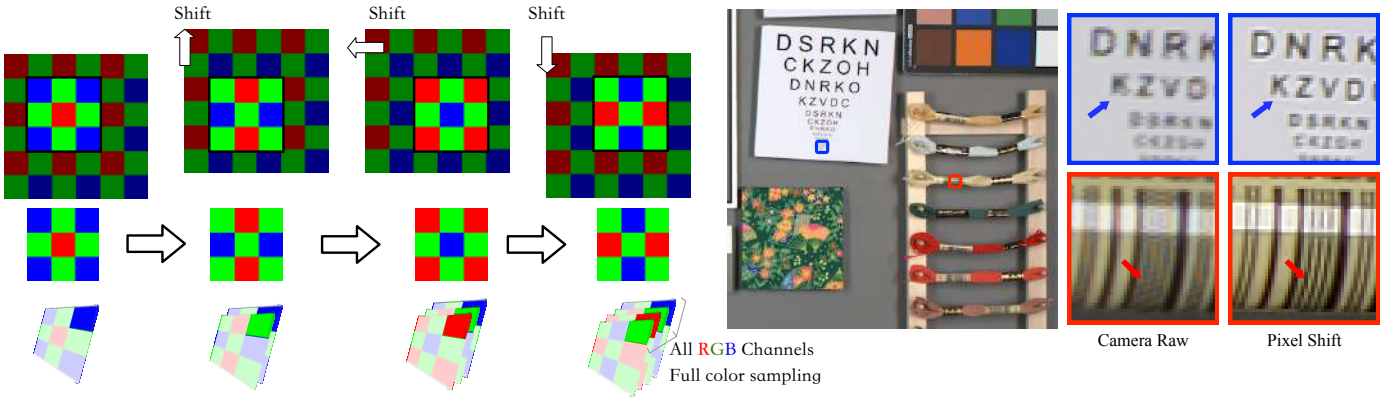


Fig. 3: The pixel shift technique used to create dataset *PixelShift200* (left) and qualitative comparison between the commonly used raw processing software Camera Raw, and the pixel shift. Pixel shift collects artifact-less (less zippering, moiré and chromatic aberration) full color sampled images directly without color interpolation.

where $\mathcal{F}_{\mathcal{I}}(\omega)$ and $\mathcal{F}_{\mathcal{O}}(\omega)$ represent the 2D Fourier transform of the ground truth and the prediction. $\epsilon = 10^{-6}$ is added to avoid dividing by zero. ReLU is used to only consider positive values that represent regions where moiré-like artifacts are likely to appear. Averaged value across frequencies is returned as the quantitative metric.

Network Training We optimized all models using Adam [46] with an initial learning rate $lr = 5 \times 10^{-4}$ on four NVIDIA RTX2080Ti GPU. A cosine annealing learning rate schedule is adopted. All models are trained for 1000 epochs to ensure convergence.

Experimental Setup of Comparison with the State-of-the-art The most closely related works are JDSR [12], RawSR [10], SGNet [40], and JDnDmSR [13], where most of which are black-box end-to-end networks without a specific pipeline (**E2ENet**). Since the architectures and data processing are different, rather than unfairly comparing with these networks, we implemented all possible pipelines (including E2ENet) using the same module as TENet++ (see Fig. 2a for the module structure) and trained all networks on the same PixelShift200 dataset. We also validate our proposed pipeline on different datasets and using models built by different modules.

We compare our proposed pipeline DN→SR→DM with all other possible pipelines in sequential solutions, and our DN+SR→DM pipeline with others in partially and fully joint solutions. A sequential solution applies three separate models sequentially, *e.g.* DN→SR→DM executing DN, SR, and DM sequentially. A partially joint solution sequentially conducts two models, while one is a joint model for two tasks, and another a single-task model. For example, DN+SR→DM performs first a joint DN and SR model DN+SR, and then a DM model. A fully joint solution solves the three tasks together using a single model. The pipeline of a joint solution (*e.g.*, DN+SR→DM) is achieved by providing additional supervision (*e.g.*, denoised super-resolved mosaic image) in an end-to-end network. The way of providing intermediate supervision is mentioned in Sec. 3.2. All models needed are implmented as follows:

- Five single-task models: raw image denoising, raw image SR, demosaicing, color image denoising, and color image SR. All five models are implemented in the same way as $\mathcal{F}_M$ (Fig. 2) by 6 RRDBs.
- Three partially joint models: DN+DM (joint DN and DM), DN+SR (joint raw image DN and SR), and DM+SR (joint DM and SR). While DN+DM and DN+SR are implemented as $\mathcal{F}_M$ (Fig. 2) using 6 RRDBs, DM+SR are implemented as E2ENet using 12 RRDBs.
- Four fully joint models: DN+SR→DM (proposed TENet++), DN+DM+SR (E2ENet), DN→DM+SR (similar architecture as TENet++ where DN supervision is provided instead), and DN+DM→SR (similar architecture as TENet++ where DM supervision is provided instead). All fully joint models are implemented by 12 RRDBs.

TABLE 1: **Comparison of pipelines on PixelShift200 test set**. Gaussian-Poisson noise with ×2 SR are used. **Bold** denotes the best performance. Our proposed pipelines yield the best quantitative results among all possible pipelines.

Type	Pipeline	PSNR	SSIM	FreqGain↓
Sequential	DM→DN→SR (usual)	33.51	0.8379	0.4853
	DM→SR→DN	30.01	0.6773	1.0978
	SR→DM→DN	31.44	0.7270	0.7858
	SR→DN→DM	33.42	0.8059	0.5583
	DN→DM→SR	36.33	0.9256	0.2067
	DN→SR→DM (ours)	**36.61**	**0.9294**	**0.1886**
Partially joint	DN→DM+SR	36.65	0.9299	0.1884
	DN+DM→SR	36.24	0.9259	0.1952
	DN+SR→DM (ours)	**37.04**	**0.9327**	**0.1829**
Fully joint	DN+DM+SR	36.71	0.9292	0.1851
	DN→DM+SR	36.18	0.9245	0.2451
	DN+DM→SR	37.24	0.9341	0.1907
	DN+SR→DM (ours)	**37.36**	**0.9353**	**0.1814**

All models are trained with the ×2 SR factor and the same level of Gaussian-Poisson noise in PixelShift200. We compare our proposed pipelines with other pipelines using these models for a fair comparison.

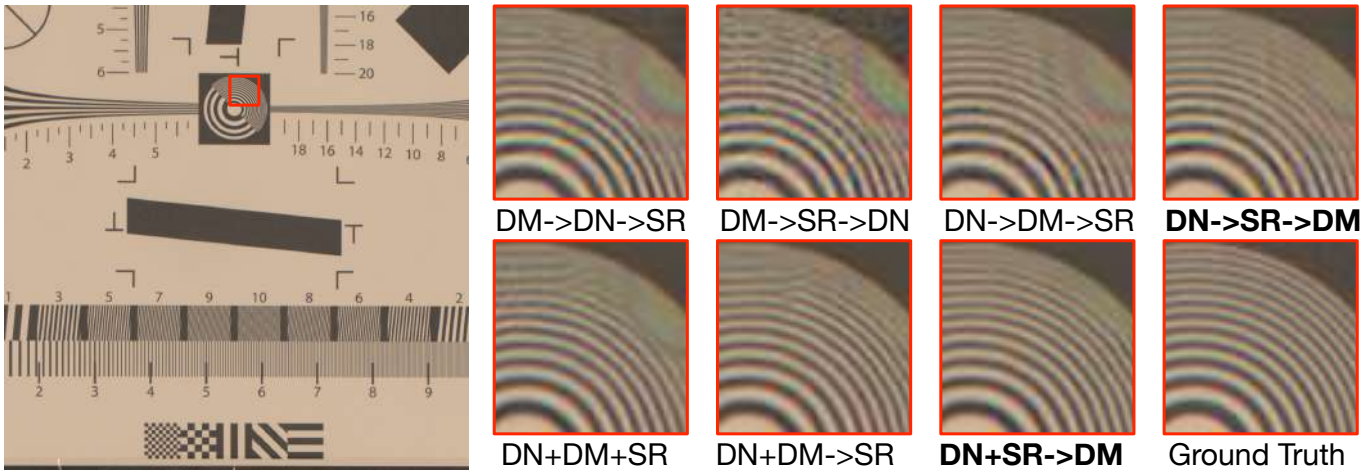


Fig. 4: **Qualitative comparisons of different pipelines on an example from PixelShift200 test set.** The left is the ground truth image, while the right shows the closeups of the output of different pipelines. The input is the low-resolution noisy mosaiced version of the left image. The top row of the right shows the results using different sequential solutions, while the bottom row shows the results of fully joint pipelines and the ground truth. Our proposed pipeline DN→SR→DM yields the highest quality among all sequential pipelines, while DN+SR→DM achieves the best among all the joint pipelines.

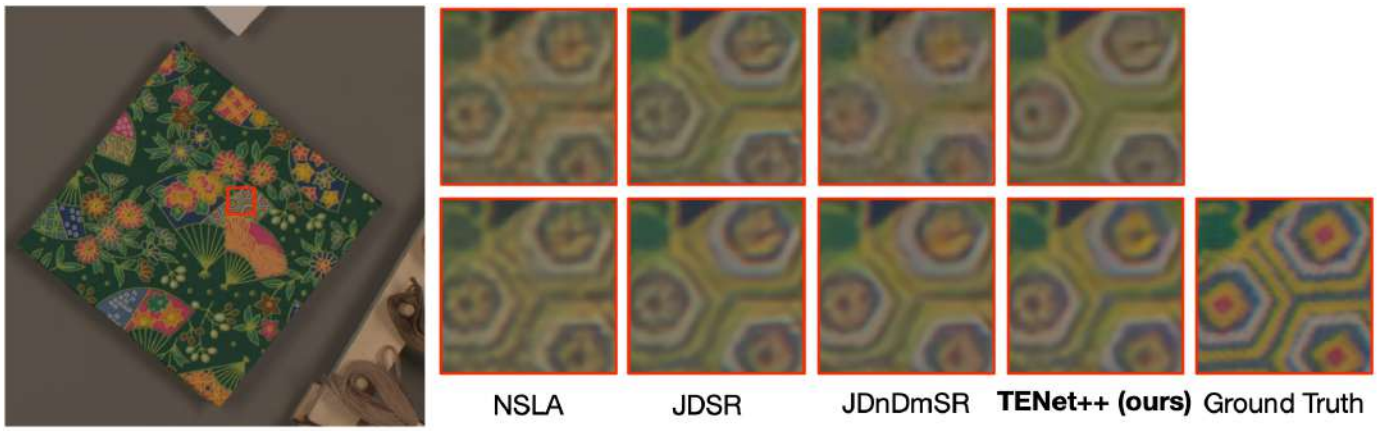


Fig. 5: **Qualitative comparisons between DN+DM+SR (top row) *vs.* DN+SR→DM (bottom row) on different architectures on PixelShift200 test set.** Our pipeline produces results with sharper edges and preserves the color of the objects better.

5.2 Pipeline Comparison Experiments

Proposed pipeline outperforms others in sequential solutions. TABLE 1 shows that *our proposed pipeline DN→SR→DM clearly outperforms all other pipelines under the sequential solution setting.* Surprisingly, the PSNR is 3.10 dB higher for our pipeline than the usual pipeline DM→DN→SR. This improvement is achieved simply by adopting our pipeline as a replacement for the other pipelines. As observed, when DN is not performed in the first stage, the image quality obtained will drop sharply. When DN is fixed as the first task, our proposed pipeline still improves PSNR by 0.28 dB, which reflects that performing SR before DM yields a higher quality than performing DM before SR. These experimental findings confirm our discussion in Sec. 3.1 that DN and SR in mosaic space are suggested.

Proposed pipeline insignificantly outperforms others in joint solutions. Since performing DN in the first stage is the best option, we now mainly study the pipelines where DN is performed first among partially and fully joint solutions. TABLE 1 shows the quantitative comparisons. One can conclude that (1) joint solutions outperform sequential solutions in the same execution order, as expected. For example, DN+SR→DM in both partially and fully joint solutions produces images with better metric values than DN→SR→DM in sequential solutions. (2) *In both partially joint and fully joint solutions, our proposed pipeline DN+SR→DM consistently generates slightly higher PSNR and SSIM than other pipelines*. In particular, PSNR of the proposed pipeline is 0.39 dB higher than any other pipelines in the partially joint solutions. In joint solutions, TENet++ (DN+SR→DM) outperforms the E2ENet counterpart (DN+DM+SR) by 0.65 dB in terms of PSNR. However, we highlight that the improvement of the

TABLE 2: **Ablation on architectures.** We experiment with the other possible architectures constructed by the NLSA [11] block and two SOTA models, JDSR [12] and JDnDmSR [13]. Our proposed pipeline DN+SR→DM consistently improves the performance of all given networks on the task of joint DN, DM and SR.

Architecture	Pipeline	PSNR	SSIM	FreqGain↓
NLSA block [11]	DN+DM+SR	34.63	0.9086	0.2975
	DN+SR→DM (ours)	**36.05**	**0.9270**	**0.1769**
JDSR [12]	DN+DM+SR	36.53	0.9289	0.1957
	DN+SR→DM (ours)	**36.68**	**0.9296**	**0.1947**
JDnDmSR [13]	DN+DM+SR	33.11	0.8782	0.4180
	DN+SR→DM (ours)	**36.91**	**0.9317**	**0.1959**
TENet++ (Ours)	DN+DM+SR	36.71	0.9292	0.1851
	DN+SR→DM (ours)	**37.36**	**0.9353**	**0.1814**

proposed pipeline in joint solutions is less than 1 dB, which is not as significant as sequential solutions. Such a marginal improvement may indicate that the execution order of tasks in an end-to-end solution might be inapplicable.

Qualitative comparisons of different pipelines. The comparison of sequential solutions in Fig. 4 (top row) shows our proposed pipeline DN→SR→DM clearly outperforms other pipelines with significantly fewer color artifacts, validating our suggestion to perform SR before DM. Our pipeline produces less noise and reflects the importance of performing DN at the first stage. In the fully joint solutions (Fig. 4 bottom row), our proposed pipeline again achieves better qualitative results than others, suffering less moiré.

6 Ablation Study

6.1 Ablate Proposed Pipeline

We have demonstrated that our proposed pipeline is quantitatively and qualitatively better than other pipelines in Sec. 5. However, one may wonder: (1) what if a different architecture is used other than the RRDB module and TENet++? (2) What if a different dataset instead of Pixelshift200 is used for training and evaluation? (3) What if a different SR factor is used instead of 2? (4) What if a different noise model is adopted instead of the Gaussian-Poisson noise model? Here *we validate that our proposed pipelines consistently outperform other pipelines in a variety of settings*.

Architecture. We ablate the module-level and network-level architectures in PixelShift200. The module-level architecture ablation study denotes that we use the same architecture as E2ENet for pipeline DN+DM+SR and as TENet++ for pipeline DN+SR→DM where a different module (*e.g.* NLSA) is used instead of the original RRGB. The network-level architecture ablation study means a different architecture rather than TENet++ is used. For the module-level experiment, we leverage the non-local sparse attention (NLSA) module from the state-of-the-art (SOTA) image SR work [11] to build the end-to-end network. For the network-level experiment, we replace TENet++ with the SOTA networks JDSR [12] and JDnDmSR [13] for the joint DN, DM and SR problem. We insert our proposed pipeline into the two models by providing intermediate supervision in a similar way as TENet++ as illustrated in Fig. 2c. TABLE 2 compares the performance of the original pipeline (DN+DM+SR) and the same network using our proposed pipeline DN+SR→DM. Experiments on all three architectures show *our pipeline consistently improves the performance regardless of the architecture designs*. In addition, by comparing results in Tab. 2 with Tab. 1, one can observe that our proposed TENet++ outperforms the network constructed by the SOTA module and the SOTA networks (JDSR [12], JDnDmSR [13]) for joint DN, DM, and SR. Qualitative comparisons of the results of different architectures (columns) fitted with two different pipelines (the top row shows the usual pipeline DN+DM+SR, the bottom row shows our pipeline DN+SR→DM) are presented in Fig. 5. For each network, our pipeline DN+SR→DM in joint solution enhances image sharpness while also preserving the color of the objects to a greater extent. Our TENet++ also yields more visually appealing images than SOTA when equipped with the same pipeline.

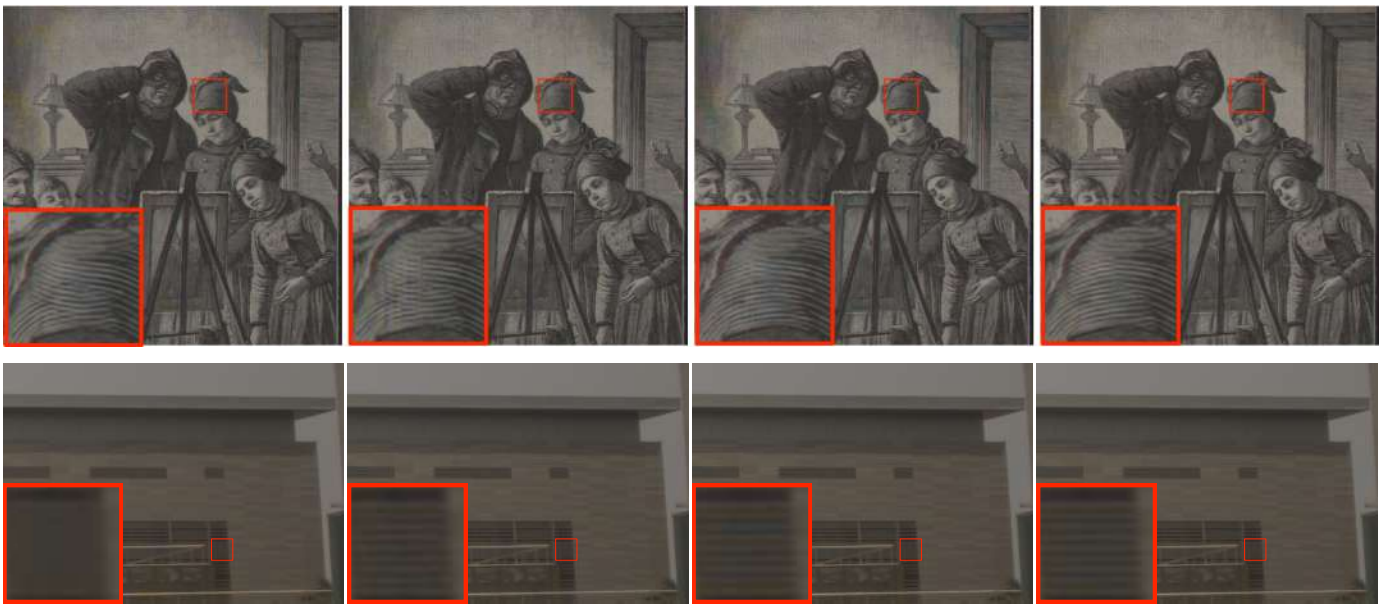
Dataset. We also experiment with different pipelines (sequential and joint solutions) on other datasets instead of PixelShift200. We train models with different pipelines on DIV2K 800 training images [42], where the mosaic images are synthesized from color images using the same unprocessing technique in [9]. Gaussian-Poisson noise and $\times 2$ SR are used. The evaluation on three widely used benchmarks, the DIV2K test set, Urban100 [49], and CBSD68 [25] is provided in TABLE 3. As observed, *our proposed pipelines improves the network's performance across all the widely-used benchmarks in both sequential and joint solutions*. Despite the consistent improvement, the PSNR gain in joint solution is less than 0.2 dB in all benchmarks, which again shows that shuffling the pipeline in an end-to-end network is not necessarily applicable.

SR factor. We also experiment with a different factor ($\times 4$) of super-resolution to validate the benefit of the proposed pipeline. TABLE 4 shows that *our proposed pipeline outperforms other pipelines under the $\times 4$ SR factor* in both sequential and joint solutions.

Noise model. Our previous experiments are conducted under the Gaussian-Poisson noise modeling assumption. Here, we further validate our pipeline under a different assumption of the noise model. We study the widely used Gaussian noise. The noise level (sigma) is set to 10. We train models on DIV2K [42] and evaluate on the DIV2K test set [42], Urban100 [49] and CBSD68 [25]. *TABLE 5 shows that our proposed pipeline DN+SR→DM is only able to marginally outperform the vanilla pipeline DN+DM+SR under the Gaussian noise setting* in joint solutions. In Fig.7, we further show the qualitative results of our TENet++ compared to the previous methods with a pipeline of DN+DM→SR. It is worth noting that our method achieves the closest qualitative performance to the Ground Truth.

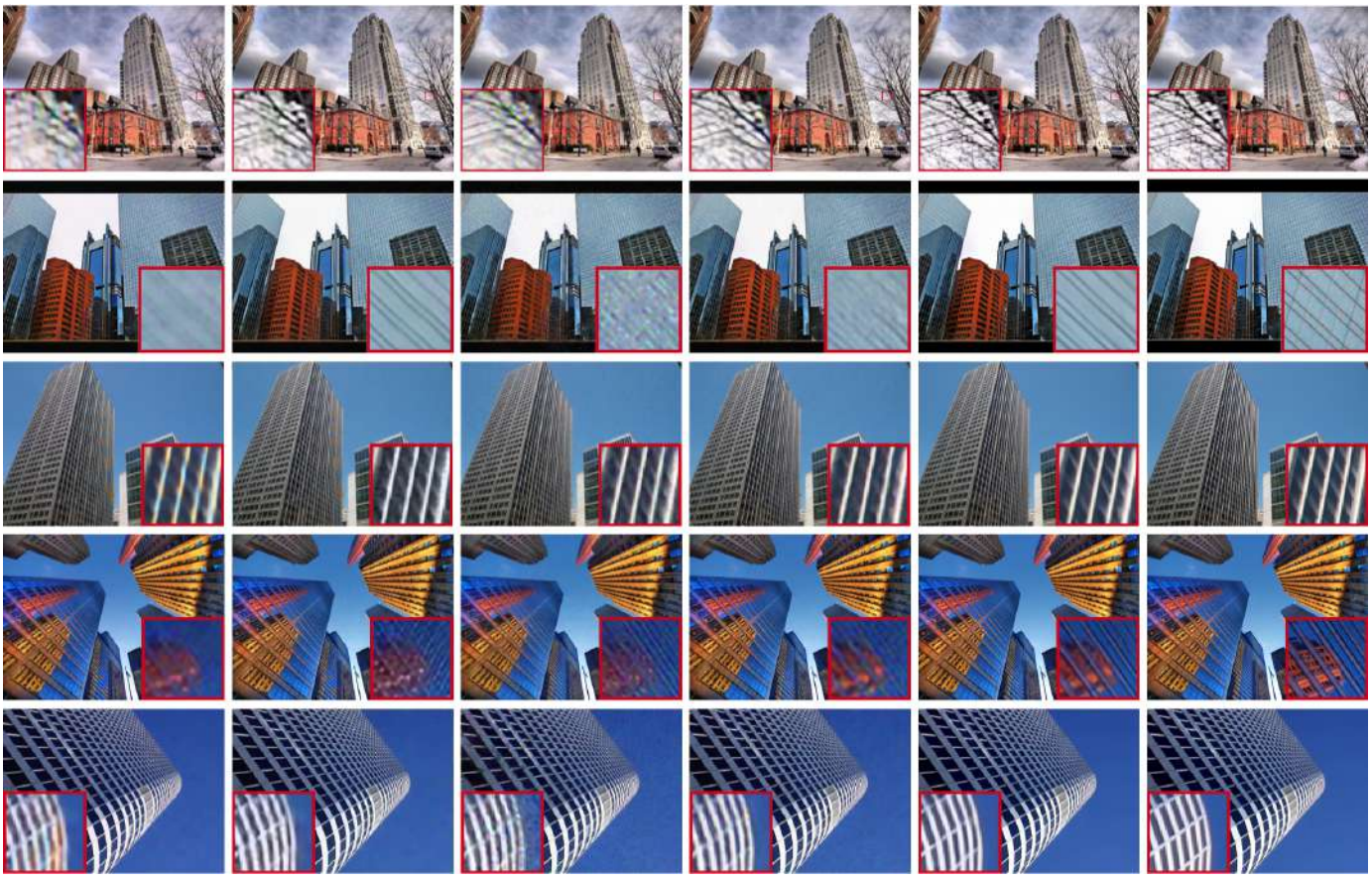
6.2 Ablate proposed Dataset PixelShift200

We evaluate two identical models (TENet++) trained on two distinct datasets, our PixelShift200 and the incompletely color-sampled dataset DIV2K [42]. The real-shot raw images are used as input. *PixelShift200 helps the model suffer less moiré and color artifacts*, as shown in Fig. 6 (column 3 *vs.* column 4). The improved qualitative performance is attributed to



JDSR [12] JDnDmSR [13] TENet++ (DIV2K) TENet++ (PixelShift200)

Fig. 6: **Qualitative comparisons on the real-shot images**. We compare the SOTA methods JDSR [12], JDnDmSR [13] and our proposed TENet++ trained on DIV2K, and TENet++ trained on PixelShift200. Images were captured using a Sony ILCE-7RM3 (top row) and iPhone XS Max (bottom row).



ADMM [47] Condat [48] FlexISP [35] DemosaicNet [4] TENet++ (ours) Ground Truth

Fig. 7: **The qualitative comparison of different methods on the noisy Urban100 [49] test images.** The noise model is the additive Gaussian noise (sigma=10) and the SR factor is 2. Our TENet achieves a close performance to the Ground Truth.

TABLE 3: **Ablation on datasets.** Models are trained on DIV2K [42], and tested on DIV2K test set, Urban100 [49], and CBSD68 [25] with Gaussian-Poisson noise and $\times 2$ SR. Our proposed pipeline outperforms other pipelines in both sequential and joint solutions regardless of datasets.

Type	Pipeline	DIV2K [42]			Urban100 [49]			CBSD68 [25]		
		PSNR	SSIM	FreqGain↓	PSNR	SSIM	FreqGain↓	PSNR	SSIM	FreqGain↓
Sequential	DM→DN→SR (usual)	20.02	0.6296	0.8705	19.32	0.6393	1.7710	20.95	0.6399	0.8073
	DM→SR→DN	19.60	0.5684	1.1938	18.96	0.5838	1.9813	20.55	0.5989	1.0690
	SR→DM→DN	19.88	0.5950	0.7514	19.25	0.6103	1.5520	20.83	0.6195	0.6139
	SR→DN→DM	20.07	0.6309	0.6012	19.41	0.6379	1.4749	21.00	0.6446	0.4723
	DN→DM→SR	20.30	**0.7660**	0.4046	19.52	**0.7326**	1.5211	21.15	**0.7097**	0.4599
	DN→SR→DM (ours)	**20.30**	0.7602	**0.2748**	**19.55**	0.7282	**1.3483**	**21.17**	0.7079	**0.2837**
Partially joint	DN→DM+SR	20.30	0.7617	0.2875	19.59	0.7337	1.3684	21.15	0.7040	0.3260
	DN+DM→SR	20.30	**0.7664**	0.4217	19.55	0.7366	1.5405	21.15	0.7101	0.4661
	DN+SR→DM (ours)	**20.34**	0.7630	**0.2693**	**19.67**	**0.7391**	**1.3379**	**21.22**	**0.7109**	**0.2776**
Fully joint	DN+DM+SR	20.30	0.7563	0.3167	19.59	0.7295	1.3980	21.16	0.7019	0.3571
	DN+SR→DM (ours)	**20.37**	**0.7677**	**0.2766**	**19.72**	**0.7467**	**1.2965**	**21.24**	**0.7118**	**0.3216**

TABLE 4: **Ablation on SR factor.** SR factor is 4. The proposed pipeline again outperforms other pipelines in both sequential and partially joint solutions.

Type	Pipeline	PSNR	SSIM	FreqGain↓
Sequential	DM→DN→SR (usual)	31.49	0.8347	0.2473
	DM→SR→DN	28.05	0.6766	0.7773
	SR→DM→DN	28.70	0.6561	0.5657
	SR→DN→DM	29.25	0.6752	0.5200
	DN→DM→SR	32.41	0.8654	0.1650
	DN→SR→DM (ours)	**32.99**	**0.8752**	**0.1506**
Partially joint	DN→DM+SR	32.95	0.8739	0.1593
	DN+DM→SR	32.32	0.8665	**0.1542**
	DN+SR→DM (ours)	**33.06**	**0.8770**	0.1547
Fully joint	DN+DM+SR	33.48	0.8797	0.1656
	DN→DM+SR	33.21	0.8765	0.1917
	DN+DM→SR	33.45	0.8796	0.1893
	DN+SR→DM (ours)	**33.54**	**0.8810**	**0.1655**

TABLE 5: **Ablation on noise model.** We experiment a different noise model, the additive Gaussian noise (sigma 10), with $\times 2$ SR. Models are trained on DIV2K [42]. Our proposed pipeline DN+SR→DM outperforms DN+DM+SR.

Dataset	Pipeline	PSNR	SSIM	FreqGain↓
DIV2K [42]	DN+DM+SR	29.74	0.8396	0.3113
	DN+SR→DM (ours)	**29.81**	**0.8410**	**0.2978**
Urban100 [49]	DN+DM+SR	26.81	0.8287	1.2128
	DN+SR→DM (ours)	**26.96**	**0.8327**	**1.1960**
CBSD68 [25]	DN+DM+SR	27.52	0.7766	0.3674
	DN+SR→DM (ours)	**27.56**	**0.7775**	**0.3616**

the full color-sampling and natural image distribution characteristics of the proposed PixelShift200.

7 Real-Shot Experiments

We compare TENet++ with the raw image processing library, DCRaw, and popular commercial software, Camera Raw, on a raw image shot with an iPhone X (see Fig. 1). The SR model implemented using the Fig. 2a network structure by 6 RRDBs (refer to Sec. 5.2 for details) is used to super-resolve the demoisaiced outputs of DCRaw and Camera Raw. The proposed TENet++ yields clean results with rich detail. We also provide more real-shot comparisons between and our TENet++ and SOTA methods when equipped with the same pipeline (DN+SR→DM) as TENet++ in Fig. 6. All models are trained on DIV2K for a fair comparison. Compared to JDSR [12], our TENet++ successfully reconstructs the high-frequency texture. TENet++ also produces far fewer artifacts, such as moirés (refer to the scarf texture in the top row) and color aliasing (refer to the steel railing in the bottom row), than JDnDmSR [13].

8 Conclusion

We presented intermediate supervision that enforces a certain pipeline in an end-to-end network. We performed a comprehensive study in the effect of pipelines on the task of learning-based denoising (DN), demosaicing (DM), and super-resolution (SR) in both sequential and joint solutions. We found that the effect of the pipeline is significant in sequential solutions, while it is marginal in joint solutions, and thus shuffling the execution order of tasks is not very necessary for an end-to-end network. We also contributed PixelShift200, a full-color sampled dataset, for training and evaluating raw image processing-related tasks.

9 Limitation and Future work

First, the proposed PixelShift200 only includes static objects and has a limited size (200 unique samples). It will be more beneficial to the community if more samples could be collected. Second, this work only considers single-frame image processing. With increasing interest in the use of multiple frames [50], we believe that it is promising to study end-to-end networks for multi-frame DN, DM, and SR, which have greater practical values but are rather under-explored.

Acknowledgement The authors thank the reviewers of ICCP 2022 for valuable suggestions and Dr. Silvio Giancola for proofreading the rebuttals. This work was supported by the KAUST Office of Sponsored Research (OSR) through the Visual Computing Center (VCC) funding.

REFERENCES

[1] R. Kimmel, "Demosaicing: image reconstruction from color ccd samples," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 8 9, pp. 1221–8, 1999.

[2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[3] M. Irani and S. Peleg, "Improving resolution by image registration," *CVGIP Graph. Model. Image Process.*, vol. 53, pp. 231–239, 1991.

[4] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, "Deep joint demosaicking and denoising," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 191, 2016.

[5] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[6] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2016.

[7] J. Nakamura, "Image sensors and signal processing for digital still cameras," 2005.

[8] P. Chatterjee, N. Joshi, S. B. Kang, and Y. Matsushita, "Noise suppression in low-light images through joint denoising and demosaicing," *CVPR 2011*, pp. 321–328, 2011.

[9] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. Barron, "Unprocessing images for learned raw denoising," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 11 028–11 037, 2019.

[10] X. Xu, Y. Ma, and W. Sun, "Towards real scene super-resolution with raw images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[11] Y. Mei, Y. Fan, and Y. Zhou, "Image super-resolution with non-local sparse attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 3517–3526.

[12] R. Zhou, R. Achanta, and S. Süsstrunk, "Deep residual network for joint demosaicing and super-resolution," in *Color and Imaging Conference*, vol. 2018, no. 1. Society for Imaging Science and Technology, 2018, pp. 75–80.

[13] W. Xing and K. Egiazarian, "End-to-end learning for joint image demosaicing, denoising and super-resolution," *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3507–3516, 2021.

[14] H. S. Malvar, L.-w. He, and R. Cutler, "High-quality linear interpolation for demosaicing of bayer-patterned color images," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 3. IEEE, 2004, pp. iii–485.

[15] K. Hirakawa and T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.

[16] L. Zhang and X. Wu, "Color demosaicking via directional linear minimum mean square-error estimation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2167–2178, 2005.

[17] O. Kapah and H. Z. Hel-Or, "Demosaicking using artificial neural networks," in *Applications of Artificial Neural Networks in Image Processing V*, vol. 3962. International Society for Optics and Photonics, 2000, pp. 112–121.

[18] Y.-Q. Wang, "A multilayer neural network for image demosaicking," in *2014 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2014, pp. 1852–1856.

[19] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," in *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. [Online]. Available: https://openreview.net/forum?id=HkeGhoA5FX

[20] Q. Bammey, R. G. V. Gioi, and J. Morel, "An adaptive neural network for unsupervised mosaic consistency analysis in image forensics," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14 182–14 192, 2020.

[21] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D: nonlinear phenomena*, vol. 60, no. 1-4, pp. 259–268, 1992.

[22] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 2. IEEE, 2005, pp. 60–65.

[23] M. Aharon, M. Elad, A. Bruckstein *et al.*, "K-svd: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Transactions on signal processing*, vol. 54, no. 11, p. 4311, 2006.

[24] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with bm3d?" in *2012 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2012, pp. 2392–2399.

[25] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 416–423.

[26] T. Plötz and S. Roth, "Neural nearest neighbors networks," in *NeurIPS*, 2018.

[27] S. W. Hasinoff, "Photon, poisson noise," in *Computer Vision, A Reference Guide*, 2014.

[28] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. Khan, M.-H. Yang, and L. Shao, "Cycleisp: Real image restoration via improved data synthesis," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2693–2702, 2020.

[29] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2750–2759, 2017.

[30] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," *2009 IEEE 12th International Conference on Computer Vision (ICCV)*, pp. 349–356, 2009.

[31] R. Timofte, V. De Smet, and L. Van Gool, "A+: Adjusted anchored neighborhood regression for fast super-resolution," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 111–126.

[32] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European Conference on Computer Vision (ECCV)*. Springer, 2016, pp. 391–407.

[33] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. C. Loy, "Esrgan: Enhanced super-resolution generative adversarial networks," in *European Conference on Computer Vision (ECCV)*. Springer, 2018, pp. 63–79.

[34] X. Liu, K. Shi, Z. Wang, and J. Chen, "Exploit camera raw data for video super- resolution via hidden markov model inference," *IEEE Transactions on Image Processing*, vol. 30, pp. 2127–2140, 2021.

[35] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian *et al.*, "Flexisp: A flexible camera image processing framework," *ACM Transactions on Graphics (TOG)*, vol. 33, no. 6, p. 231, 2014.

[36] E. Schwartz, R. Giryes, and A. M. Bronstein, "Deepisp: Toward learning an end-to-end image processing pipeline," *IEEE Transactions on Image Processing*, vol. 28, pp. 912–923, 2019.

[37] T. Klatzer, K. Hammernik, P. Knobelreiter, and T. Pock, "Learning joint demosaicing and denoising based on sequential energy minimization," in *2016 IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2016, pp. 1–11.

[38] K. Zhang, W. Zuo, and L. Zhang, "Learning a single convolutional super-resolution network for multiple degradations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3262–3271.

[39] Y.-S. Xu, S.-Y. R. Tseng, Y. Tseng, H.-K. Kuo, and Y.-M. Tsai, "Unified dynamic convolutional network for super-resolution with variational degradations," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12 493–12 502, 2020.

[40] L. Liu, X. Jia, J. Liu, and Q. Tian, "Joint demosaicing and denoising with self guidance," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2237–2246, 2020.

[41] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. 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*, 2016, pp. 1874–1883.

[42] E. Agustsson and R. Timofte, "Ntire 2017 challenge on single image super-resolution: Dataset and study," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 126–135.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 2009, pp. 248–255.

[44] Y. Xing, Z. Qian, and Q. Chen, "Invertible image signal processing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 6287–6296.

[45] Sony. (2017) Pixel shift multi shooting. https://support.d-imaging.sony.co.jp/support/ilc/psms/ilce7rm3/en/index.html.
[46] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2015.
[47] H. Tan, X. Zeng, S. Lai, Y. Liu, and M. Zhang, "Joint demosaicing and denoising of noisy bayer images with admm," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 2951–2955.
[48] L. Condat and S. Mosaddegh, "Joint demosaicking and denoising by total variation minimization," in *2012 19th IEEE International Conference on Image Processing*. IEEE, 2012, pp. 2781–2784.
[49] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5197–5206.
[50] B. Wronski, I. Garcia-Dorado, M. Ernst, D. Kelly, M. Krainin, C.-K. Liang, M. Levoy, and P. Milanfar, "Handheld multi-frame super-resolution," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–18, 2019.

Guocheng Qian is currently working towards a doctoral degree in the Department of Computer Science at King Abdullah University of Science and Technology (KAUST). He received his Master's degree from KAUST in 2020 and his BEng degree with first-class honors from Xi'an Jiaotong University (XJTU) in 2018. His research interests are in computer vision and geometric deep learning. He has co-authored five peer-reviewed conference and journal papers in CVPR, NeurIPS, T-PAMI, *etc.*

Yuanhao Wang received the BEng degree from Beijing University of Posts and Telecommunications in 2013, and MEng degree from Tsinghua University in 2016. He is working towards the doctoral degree currently in the department of Electrical and Computer Engineering at King Abdullah University of Science and Technology. His research interests line in compuational imaging and neural radiance field.

Jinjin Gu is currently pursuing a Ph.D. degree in Engineering and IT with the University of Sydney. He received his B.Eng. degree in computer science and engineering from the Chinese University of Hong Kong, Shenzhen, in 2020. His research interests include computer vision, image processing, interpretability of deep learning algorithms, and machine learning applications in industrial.

Chao Dong is currently an associate professor at Shenzhen Institute of Advanced Technology, Chinese Academy of Science. He received his Ph.D. degree from The Chinese University of Hong Kong in 2016. In 2014, he introduced the deep learning method – SRCNN into the super-resolution field. This seminal work was chosen as one of the top ten "Most Popular Articles" of TPAMI in 2016. His team has won several championships in international challenges – NTIRE2018, PIRM2018, NTIRE2019, NTIRE2020 and AIM2020. He worked in SenseTime from 2016 to 2018 as the team leader of Super-Resolution Group. His Google citation has surpassed 16,000. His current research interest focuses on low-level vision problems, such as image/video super-resolution, denoising and enhancement. Email: chao.dong@siat.ac.cn.

Wolfgang Heidrich (Fellow, IEEE) is a Professor of Computer Science and Electrical and Computer Engineering in the King Abdullah University of Science and Technology (KAUST) Visual Computing Center, for which he also served as director from 2012 to 2021. Prof. Heidrich joined KAUST in 2014, after 13 years as a faculty member at the University of British Columbia. He received his Ph.D. from the University of Erlangen in 1999, and then worked as a Research Associate in the Computer Graphics Group of the Max-Planck Institute for Computer Science in Saarbrucken, Germany, before joining UBC in 2000. Prof. Heidrich's research interests lie at the intersection of imaging, optics, computer vision, computer graphics, and inverse problems. His more recent interest is in computational imaging, focusing on hardware-software co-design of the next generation of imaging systems, with applications such as High-Dynamic Range imaging, compact computational cameras, hyperspectral cameras, to name just a few. Prof. Heidrich's work on High Dynamic Range Displays served as the basis for the technology behind Brightside Technologies, which was acquired by Dolby in 2007. Prof. Heidrich is a Fellow of the IEEE and Eurographics, and the recipient of a Humboldt Research Award.

Bernard Ghanem is currently a Professor in the CEMSE division, a theme leader at the Visual Computing Center (VCC), and the Deputy Director of the AI Initiative at King Abdullah University of Science and Technology (KAUST). His research interests lie in computer vision and machine learning with emphasis on topics in video understanding, 3D recognition, and theoretical foundations of deep learning. He received his Bachelor's degree from the American University of Beirut (AUB) in 2005 and his MS/PhD from the University of Illinois at Urbana-Champaign (UIUC) in 2010. His work has received several awards and honors, including six Best Paper Awards for workshops in CVPR, ICCV, and ECCV, a Google Faculty Research Award in 2015 (1st in MENA for Machine Perception), and a Abdul Hameed Shoman Arab Researchers Award for Big Data and Machine Learning in 2020. He has co-authored more than 150 peer reviewed conference and journal papers in his field as well as three issued patents. He serves as an Associate Editor for IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) and has served several times as Area Chair (AC) for CVPR, ICCV, ECCV, ICLR, AAAI, and NeurIPS.

Jimmy S. Ren is currently a senior research director at SenseTime where he leads a team to build high impact computational photography products. He also holds an adjunct faculty position in Qing Yuan Research Institute, Shanghai Jiao Tong University. He received his Ph.D. degree from City University of Hong Kong in 2013. His research interests are computational photography, image processing and computer vision.