

# Supplemental Material for A Comparison of Image Denoising Methods

Zhaoming Kong, Fangxi Deng, Haomin Zhuang, Jun Yu,  
Lifang He, *Member, IEEE*, and Xiaowei Yang

**Abstract**—In the supplemental material, more experimental results and discussions are provided.

## 1 IMAGE DENOISING USING MODIFIED SVD

In this section, we first introduce the motivation and details of the modified SVD (M-SVD) method presented in our paper, and then discuss the relationship between SVD and its tensor counterpart.

### 1.1 The M-SVD Method

The M-SVD method also follows the grouping-collaborative filtering-aggregation paradigm. For simplicity, our analysis is based on the (sRGB) image denoising task. Let us consider the local group  $\mathcal{G} \in \mathbb{R}^{ps \times ps \times 3 \times K}$  of  $K$  patches, the original SVD algorithm operates directly on the matrix representation  $\mathbf{G}_{(4)}$  of  $\mathcal{G}$  via

$$\mathbf{G}_{(4)} = \mathbf{U}\mathbf{S}\mathbf{V}^T \quad (1)$$

where  $\mathbf{S}$  is singular value matrix,  $\mathbf{U}$  and  $\mathbf{V}$  are group and patch level transform matrices, respectively. A more effective way to obtain  $\mathbf{U}$  is to perform SVD on the opponent color space [1], [2] represented by  $\mathbf{G}_{opp(3)}$ , with  $\mathcal{G}_{opp} = \mathcal{G}(:, :, 1, :) + \mathcal{G}(:, :, 2, :) + \mathcal{G}(:, :, 3, :)$ . Then the coefficient matrix  $\mathbf{C}$  of  $\mathbf{G}_{(4)}$  can be calculated by

$$\mathbf{C} = \mathbf{U}^T \mathbf{G}_{(4)} \mathbf{V} \quad (2)$$

After the truncation of  $\mathbf{C}$  via  $\mathbf{C}_{trun} = \text{hard-threshold}(\mathbf{C})$ , and the inverse transform of Eq. (2), the estimated clean group  $\hat{\mathbf{G}}_c = \mathbf{U}\mathbf{C}_{trun}\mathbf{V}^T$  is obtained. The method is briefed in Algorithm 1.

### 1.2 The Relationship between SVD and HOSVD

The relationship between SVD and HOSVD is derived based on the following Theorems.

**Theorem 1.** Given an  $N$ -th order tensor  $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ , if  $\mathbf{U}_i, \mathbf{V}_i$  and  $\mathbf{S}_i$  are the singular vectors and singular value matrix of  $\mathbf{G}_{(i)}$  obtained by  $\mathbf{G}_{(i)} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T$ , then  $\mathbf{U}_1 \otimes \cdots \otimes \mathbf{U}_N$ ,

- Z. Kong, H. Zhuang and X. Yang are with the School of Software Engineering, South China University of Technology, Guangzhou, China (e-mail: kong.zm@mail.scut.edu.cn; semzm@mail.scut.edu.cn; xwyang@scut.edu.cn).
- F. Deng is with Tencent Technology (email: fangxideng@outlook.com).
- J. Yu and L. He are with the Department of Computer Science and Engineering, Lehigh University, PA, USA (email: juy220@lehigh.edu; lih319@lehigh.edu).

---

#### Algorithm 1 Modified SVD (M-SVD)

**Input:** Noisy image  $\mathcal{A}$ , patch size  $ps$ , number of similar patches  $K$  and search window size  $\Omega_{SR}$ .

**Output:** Estimated clean image  $\hat{\mathcal{A}}_c$ .

**Step 1** (Grouping): For every reference patch of  $\mathcal{A}$ , stack  $K$  similar patches in a group  $\mathcal{G}$  within  $\Omega_{SR}$ .

**Step 2** (Collaborative filtering):

(1) Obtain the group and patch level transform  $\mathbf{U}$  and  $\mathbf{V}$  by performing SVD on  $\mathbf{G}_{opp(3)}$  and  $\mathbf{G}_{(4)}^T$ , respectively.

(2) Apply the hard-threshold technique to  $\mathbf{C} = \mathbf{U}^T \mathbf{G}_{(4)} \mathbf{V}$  via  $\mathbf{C}_{trun} = \text{hard-threshold}(\mathbf{C})$ .

(3) Take the inverse transform to obtain estimated clean group via  $\hat{\mathbf{G}}_c = \mathbf{U}\mathbf{C}_{trun}\mathbf{V}^T$ .

**Step 3** (Aggregation): Averagely write back all image patches in  $\hat{\mathbf{G}}_c$  to their original locations.

---

$\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_N$  and  $\mathbf{S}_1 \otimes \cdots \otimes \mathbf{S}_N$  are singular vectors and singular value matrices of  $\mathbf{G}_{(1)} \otimes \cdots \otimes \mathbf{G}_{(N)}$ . More specifically

$$\begin{aligned} & \mathbf{G}_{(1)} \otimes \cdots \otimes \mathbf{G}_{(N)} \\ &= (\mathbf{U}_1 \mathbf{S}_1 \mathbf{V}_1^T) \otimes \cdots \otimes (\mathbf{U}_N \mathbf{S}_N \mathbf{V}_N^T) \\ &= (\mathbf{U}_1 \otimes \cdots \otimes \mathbf{U}_N)(\mathbf{S}_1 \otimes \cdots \otimes \mathbf{S}_N)(\mathbf{V}_1 \otimes \cdots \otimes \mathbf{V}_N)^T \end{aligned} \quad (3)$$

**Proof.** Theorem 1 is an extension of the results in [3] when  $N > 2$ , and based on mathematical induction, we assume that Eq. (3) holds for  $N - 1$ -th order tensor, we prove that it is also true for  $N$ -th order tensor. Notice that

$$\mathbf{G}_{(1)} \otimes \cdots \otimes \mathbf{G}_{(N)} = (\mathbf{G}_{(1)} \otimes \cdots \otimes \mathbf{G}_{(N-1)}) \otimes \mathbf{G}_{(N)} \quad (4)$$

Let  $\tilde{\mathbf{G}} = \mathbf{G}_{(1)} \otimes \cdots \otimes \mathbf{G}_{(N-1)}$ , then Eq. (4) can be reduced to the case when  $N = 2$ . Therefore Eq. (3) also holds for  $N$ -th order tensor.

**Theorem 2.** Given an  $N$ -th order tensor  $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ , its full HOSVD is

$$\mathcal{G} = \mathcal{C} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times \cdots \times_N \mathbf{U}_N \quad (5)$$

where  $\mathcal{C} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$  and  $\mathbf{U}_i \in \mathbb{R}^{I_i \times I_i}$  represents core tensor and the  $i$ -th mode projection matrix, respectively. If  $\mathbf{S}_i$  is the singular value matrix of  $\mathbf{G}_{(i)}$ , then

$$\mathbf{C}_{(i)} \mathbf{C}_{(i)}^T = \mathbf{S}_i \mathbf{S}_i^T \quad (6)$$

**Proof.** According to [4], [5], Eq. (5) can be rewritten as

$$\mathbf{G}_{(i)} = \mathbf{U}_i \mathbf{C}_{(i)} (\mathbf{U}_N \otimes \cdots \otimes \mathbf{U}_{i+1} \otimes \mathbf{U}_{i-1} \otimes \cdots \otimes \mathbf{U}_1)^T \quad (7)$$

For simplicity, let  $\hat{\mathbf{U}}_i = \mathbf{U}_N \otimes \cdots \otimes \mathbf{U}_{i+1} \otimes \mathbf{U}_{i-1} \otimes \cdots \otimes \mathbf{U}_1$ , then based on Eq. (3), we have

$$\begin{aligned} \hat{\mathbf{U}}_i^T \hat{\mathbf{U}}_i &= (\mathbf{U}_N^T \mathbf{U}_N) \otimes \cdots \otimes (\mathbf{U}_{i+1}^T \mathbf{U}_{i+1}) \otimes \\ &\quad \otimes (\mathbf{U}_{i-1}^T \mathbf{U}_{i-1}) \otimes \cdots \otimes (\mathbf{U}_1^T \mathbf{U}_1) = \mathbf{I} \end{aligned} \quad (8)$$

Therefore,  $\hat{\mathbf{U}}_i$  is also orthogonal. Given that  $\mathbf{G}_{(i)} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T = \mathbf{U}_i \mathbf{C}_{(i)} \hat{\mathbf{U}}_i^T$ , we have

$$\mathbf{G}_{(i)} \mathbf{G}_{(i)}^T = \mathbf{U}_i \mathbf{C}_{(i)} \mathbf{C}_{(i)}^T \mathbf{U}_i^T = \mathbf{U}_i \mathbf{S}_i \mathbf{S}_i^T \mathbf{U}_i^T \quad (9)$$

which indicates  $\mathbf{C}_{(i)} \mathbf{C}_{(i)}^T = \mathbf{S}_i \mathbf{S}_i^T$  since  $\mathbf{U}_i$  is orthogonal.

**Theorem 3.** Given an  $N$ -th order tensor  $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$  and its core tensor  $\mathcal{C}$  of the same size, with  $\mathbf{G}_{(i)} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T$  via SVD, then the  $j$ -th column of  $\mathbf{V}_i$  can be represented as

$$\mathbf{V}_i(:, j) = \frac{\hat{\mathbf{U}}_i \mathbf{C}_{(i)}(j, :)^T}{\|\mathbf{C}_{(i)}(j, :)^T\|}, \quad 1 \leq j \leq I_{i_nz}. \quad (10)$$

where  $I_{i_nz}$  is the number of non-zero singular values of  $\mathbf{S}_i$ , and  $\hat{\mathbf{U}}_i = \mathbf{U}_N \otimes \cdots \otimes \mathbf{U}_{i+1} \otimes \mathbf{U}_{i-1} \otimes \cdots \otimes \mathbf{U}_1$ .

**Proof.** Since

$$\mathbf{G}_{(i)} = \mathbf{U}_i \mathbf{S}_i \mathbf{V}_i^T = \mathbf{U}_i \mathbf{C}_{(i)} \hat{\mathbf{U}}_i^T \quad (11)$$

Then left-multiplying Eq. (11) by the orthogonal matrix  $\mathbf{U}_i^T$ , we have

$$\mathbf{S}_i \mathbf{V}_i^T = \mathbf{C}_{(i)} \hat{\mathbf{U}}_i^T \quad (12)$$

then

$$\sigma_{ij} \mathbf{V}_i(:, j) = \hat{\mathbf{U}}_i \mathbf{C}_{(i)}(j, :)^T \quad (13)$$

where  $\sigma_{ij}$  is the  $j$ -th non-zero singular value of  $\mathbf{S}_i$ . According to Eq. (6), we have  $\mathbf{C}_{(i)}(j, :) \mathbf{C}_{(i)}(j, :)^T = \sigma_{ij}^2$ , which indicates

$$\|\mathbf{C}_{(i)}(j, :)\| = \sigma_{ij} \quad (14)$$

substitute Eq. (14) into Eq. (13), we have

$$\mathbf{V}_i(:, j) = \frac{\hat{\mathbf{U}}_i \mathbf{C}_{(i)}(j, :)^T}{\|\mathbf{C}_{(i)}(j, :)\|} \quad (15)$$

For simplicity, consider the case of third-order tensor group  $\mathcal{G} \in \mathbb{R}^{I_1 \times I_2 \times K}$  with  $K$  patches of size  $I_1 \times I_2$ , its full HOSVD is given by

$$\mathcal{G} = \mathcal{C} \times_1 \mathbf{U}_1 \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3 \quad (16)$$

The full SVD of the matrix representation  $\mathbf{G}_{(3)}$  is

$$\mathbf{G}_{(3)} = \mathbf{U}_3 \mathbf{S}_3 \mathbf{V}_3^T \quad (17)$$

Theorem 2 indicates that truncating the columns of the group-level transform  $\mathbf{U}_3$  is equivalent to shrinking the rows of  $\mathbf{C}_{(3)}$  and also the singular value matrix  $\mathbf{S}_3$  of  $\mathbf{G}_{(3)}$ . Theorem 3 demonstrates how the patch-level transform  $\mathbf{V}_3$  of  $\mathbf{G}_{(3)}$  can be derived from the first and second mode projection matrices  $\mathbf{U}_1$  and  $\mathbf{U}_2$  of  $\mathcal{G}$ .

## 2 DATASETS

In this section, we provide a short description of datasets in Table 1 and briefly introduce the proposed datasets. More information is available in [9], [10], [11], [13], [14], [16], [17], [19], [20], [22], [24], [26], [27].

TABLE 1  
Popular datasets for synthetic and real-world experiments. 'GT': ground-truth, '√': Available, '-': Not available, 'F': number of frames.

Applications	Name	Experiments	GT	Data size	# Data
Grayscale/Color Image	Standard set [6]	Synthetic	√	512 × 512 / 256 × 256	11
	BSD [7]	Synthetic	√	481 × 321 × 3	500
	Kodak [8]	Synthetic	√	512 × 512 × 3	24
	RENOIR [9]	Real-world	√	3684 × 2760 × 3	120
	Nam-CC15 [10]	Real-world	√	512 × 512 × 3	15
	Nam-CC60 [10]	Real-world	√	500 × 500 × 3	60
	PolyU [11]	Real-world	√	512 × 512 × 3	100
	DnD [12]	Real-world	-	512 × 512 × 3	1000
	SIDD [13]	Real-world	-	256 × 256 × 3	1280
	HighISO [14]	Real-world	√	512 × 512 × 3	100
Video	Our IOCI	Real-world	√	1024 × 1024 × 3	848
	Set8 [15]	Synthetic	√	960 × 540 × 3 × F	8
	DAVIS [16]	Synthetic	√	854 × 480 × 3 × F	30
	CRVD [17]	Real-world	√	1920 × 1080 × 3 × F	61
	PVDD [18]	Real-world	-	-	200
MSI/HSI	Our IOCV	Real-world	√	512 × 512 × 3 × F	39
	CAVE [19]	Synthetic	√	512 × 512 × 31	32
	ICVL [20]	Synthetic	√	1392 × 1300 × 31	201
	Indian Pines [21]	Synthetic	√	145 × 145 × 224	1
	Urban [21]	Real-world	-	307 × 307 × 210	1
MRI	HHD [22]	Real-world	-	1392 × 1040 × 31	77
	Real-HSI [23]	Real-world	√	696 × 520 × 34	59
	BrainWeb [24]	Synthetic	√	181 × 217 × 181	3
	fastMRI [25]	Synthetic	√	320 × 320 × 40	50
OASIS	OASIS [26]	Real-world	-	256 × 256 × 128	2

## 2.1 Image and Video Denoising Datasets

In this paper, we focus on the denoising task of images and videos in the standard RGB (sRGB) color space.

### 2.1.1 Image Denoising Datasets

The information of popular image denoising datasets are summarized below.

- **BSD** [7]. The BSD dataset is originally used for the image segmentation task, which is collected from the Corel dataset images from 30 human subjects. The BSD dataset contains both grayscale and color images of size  $481 \times 321$  for training and testing.
- **Kodak** [8]. The Kodak gallery dataset consists of 24 uncompressed indoor and outdoor images published by the Kodak Corporation. The dataset is widely used for synthetic experiments.
- **RENOIR** [9]. The RENOIR dataset is the first trial to construct real-world image dataset with noisy and 'ground-truth' image pairs. Three cameras are used to take photos of indoor scenes with different ISO values. The low-noise reference images are obtained with low light sensitivity (ISO = 100) and long exposure time.
- **Nam** [10]. The Nam dataset collects images of 11 indoor scenes with printed pictures and some real objects. Three cameras are used across 3 ISO factors (1600, 3200, 6400). Each ground-truth image is generated by sampling the same scene for approximately 500 times and computing their mean value.
- **DnD** [12]. The DnD dataset utilizes the Tobit regression to estimate the parameters of the noise process by accessing only two images. Careful post-processing step

is conducted to correct the misalignment and remove residual low-frequency bias. It contains 50 pairs of real-world noisy and noise-free images, which are further cropped into 1000 smaller patches. The 'ground-truth' images of this dataset are currently not available, but objective results of denoising methods can be obtained by submitting filtered images online.

- **SIDD** [13]. The SIDD dataset captures images of 10 scenes using five smartphones with 15 ISO levels under 3 light brightness levels (low, normal and, high). Each scene is captured multiple times with different cameras settings and lighting conditions. SIDD provides 1280 smaller patches for testing.
- **PolyU** [11]. The PolyU dataset uses five different cameras and includes a broader variety of indoor scenes. More camera settings are included with 6 ISO values (800, 1600, 3200, 6400, 12800, 25600) and carefully adjusted shutter speed and aperture. Besides, three volunteers are invited to manually remove outlier images that show clear misalignment or different illuminations. Each 'ground-truth' image is also obtained by averaging more than 500 successive noisy observations.
- **HighISO** [14]. The HighISO dataset contains 110 paired JPEG images of 28 indoor scenes. Four different cameras are used with  $\text{ISO} \geq 3200$ . The clean image is obtained by shooting the same scene for 100 times and their average is treated as the noise-free image.

The abundance of real-world color image datasets largely results from the relatively low cost of generating clean reference images from noisy observations. A common strategy adopted by a variety of datasets [11], [13], [14], [17], [27] is called image averaging, which captures the same and unchanged scene for many times and computes their mean value to obtain the corresponding noise-free image. The rationale of this simple strategy is that for each pixel, the random noise is assumed to be larger or smaller than 0, and thus can be greatly suppressed by sampling the same pixel for many times [11].

### 2.1.2 Video Denoising Datasets

Unlike image denoising datasets, the number of video denoising datasets are relatively small. The information of related video denoising datasets are summarized below.

- **Set8** [15]. The Set8 dataset is composed of 4 color sequences from the Derf's Test Media collection and 4 color sequences captured with a GoPro camera. The sequences of Set8 have been downsampled to a resolution of  $960 \times 540$  and the maximum number of frames of each sequence is 85.
- **DAVIS** [16], [28], [29]. The DAVIS datasets are originally used for the object segmentation tasks and challenges. In this paper, we select the DAVIS-2017-test-dev, DAVIS-2017-test-challenge and DAVIS-2019-unsupervised datasets for synthetic video denoising experiments. The three selected datasets all contain 30 sequences with a resolution of  $480 \times 854$ .
- **CRVD** [17]. The CRVD dataset contains 55 video sequences captured with 11 different indoor scenes under 5 different ISO levels ranging from 1600 to 25600. The motion is simulated by manually moving controllable

objects. For each motion,  $N$  noisy images are captured and their mean image is regarded as the noise-free video frame. Finally, all the noise-free frames are grouped together according to their temporal order to generate the clean video.

- **PVDD** [18]. The PVDD dataset contains 200 noisy-clean video pairs of both indoor and outdoor scenes with real motions and rich dynamics. The video sequences are captured with a NIKON Z7 II mirrorless camera, and each video clip lasts 8s to 15s. The ground-truth videos are produced by carefully adjusting the camera settings such as aperture, ISO and focal length. The dataset is currently not available.

Compared to the passion for collecting real-world image datasets, fewer efforts have been made to produce realistic video pairs for denoising. The image averaging strategy can not be directly applied to videos, and the difficulty lies mainly in continuously capturing noisy-clean video pairs for dynamic scenes. Besides, the strategy of generating clean dynamic videos using low ISO and high exposure time may cause motion blur [17].

## 2.2 MSI/HSI and MRI Denoising Datasets

In this section, we briefly introduce the MSI/HSI and MRI datasets listed in Table 1. Briefly, the emergence of MSI and MRI reflects the development of new imaging techniques. It is noteworthy that compared to image and video data, collecting MSI/HSI and MRI data are of greater difficulty and also more expensive.

### 2.2.1 MSI/HSI Denoising Datasets

MSI/HSI cameras are mostly spectroscopic devices, which need to be carefully calibrated in order to obtain a reliable spectral information [30]. The information of datasets are summarized below.

- **CAVE** [19]. The representative CAVE dataset is constructed with a generalized assorted pixel (GAP) camera, it includes 32 scenes of a wide variety of real-world materials and objects. Each image has the size of  $512 \times 512$  in space and includes full spectral resolution reflectance data ranging from  $400 \text{ nm}$  to  $700 \text{ nm}$  at  $10 \text{ nm}$  steps, which leads to 31 bands.
- **ICVL** [20]. The ICVL dataset is acquired using a Specim PS Kappa DX4 camera, and it includes more than 200 different images collected at  $1392 \times 1300$  spatial resolution, and the data are downsampled to 31 spectral channels from  $400 \text{ nm}$  to  $700 \text{ nm}$  at  $10 \text{ nm}$  increments.
- **Real-HSI** [23]. The Real-HSI dataset contains paired real HSI data of size  $696 \times 520 \times 34$ , captured by a SOC710-VP hyperspectral camera from Surface Optics Corporation. 34 bands around  $400 \text{ nm}$  to  $700 \text{ nm}$  are selected in visible spectral range. The dynamic range is 12 bit with the spectral range between 0 and 4096.
- **HHD** [22]. The real-world HHD dataset from Harvard includes 75 images of size  $1040 \times 1392 \times 31$  collected under daylight, artificial and mixed illumination. The images were captured using a commercial hyperspectral camera (Nuance FX, CRI Inc) with 31 wavelength bands ranging from  $420 \text{ nm}$  to  $720 \text{ nm}$  at  $10 \text{ nm}$  steps.

### 2.2.2 MRI Denoising Datasets

MRI techniques use a strong magnetic field and radio waves to create detailed images of the organs and tissues within the body, and it plays an important role in disease diagnosis. In our paper, we use three datasets from different modalities (3D brain and knee MRI) for comparison. The information of datasets are summarized below.

- **Brainweb** [24]. The Brainweb dataset contains a set of MRI data volumes produced by an MRI simulator, which are often used as ground-truth MR images for different validation purpose. Full 3D data volumes have been simulated using three sequences (T1-, T2-, and proton-density- (PD-) weighted), a variety of slice thicknesses and levels of intensity non-uniformity.
- **fastMRI** [25]. The fastMRI Initiative dataset consists of PD-weighted knee MRI scans with and without fat suppression with in-plane size  $320 \times 320$  and the number of slices varying from 27 to 45. Similar to [31], we select 50 volumes from the dataset for evaluation.
- **OASIS** [26]. The realistic OASIS brain image dataset contains selected 3D T1-weighted (T1w) data of size  $256 \times 256 \times 128$ , acquired by an MP-RAGE volumetric sequence on a Siemens 1.5T Vision scanner, with repetition time (TR) 9.7 msec, echo time (TE) 4.0 msec, flip angle 10 degrees, inversion time (TI) 10 msec, duration time of 200 ms, and voxel resolution of  $1 \times 1 \times 1.25 \text{ mm}^3$ .

The acquisition of MRI data is often expensive and time-consuming. To the best of our knowledge, there is no appropriate paired real MRI denoising dataset.

### 2.3 The Proposed Dataset

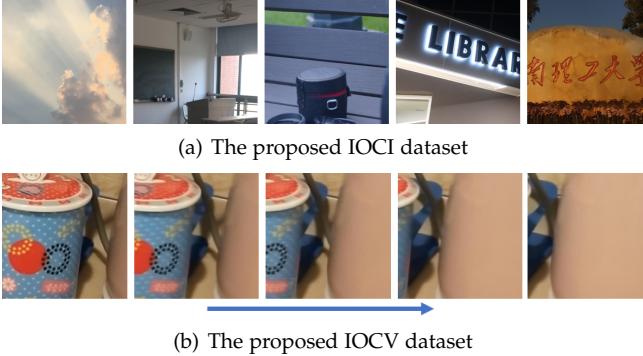


Fig. 1. Illustration of the proposed IOCI and IOCV dataset.

In this subsection, we briefly introduce the motivation and details regarding the setup and protocol followed by our indoor-outdoor color image (IOCI) and video (IOCV) datasets, some examples are shown in Fig. 1.

**IOCI.** Images captured by different cameras under various conditions may show distinct noise patterns, as illustrated in Fig. 2 and Fig. 3. In our dataset, 13 camera devices<sup>1</sup> of different types and manufacturers are used. The cameras are carefully mounted on a sturdy tripod, and a remote control software is employed to reduce human interference. Besides, it is noticed that photos taken outdoor also pose great challenge on denoising, since the source of light may

be more complicated and it is not easy to find sufficient similar patches. Therefore, the images in our dataset are captured based on more than 160 indoor and outdoor scenes with different backgrounds and lighting conditions. Instead of using pre-defined camera settings such as ISO, shutter speed and aperture, we mostly resort to the cameras' 'auto mode', which emulate real scenarios. In uncontrollable and dark environments, we adopt short exposures and high ISO values to reduce misalignment and produce over 100 images with higher noise levels. In addition, images that display obvious misalignment and illumination differences are discarded. Considering the large size of original images, around 4 subimages of size  $1024 \times 1024 \times 3$  are selected for each original image.

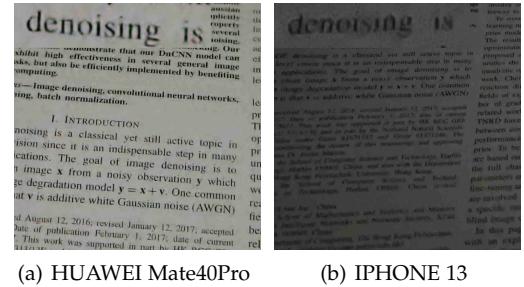


Fig. 2. Images captured by two popular smartphone cameras of the same objects in the dark environment.

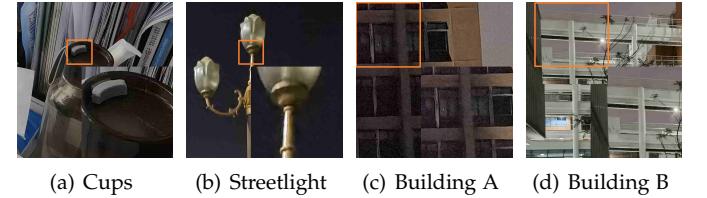


Fig. 3. Images captured by HUAWEI Mate40Pro of different scenes.

**IOCV.** The carefully-designed frame-by-frame strategy of the CRVD dataset is able to produce high-quality clean reference videos, but it also requires a lot of human effort, and the manually-created motions may not be continuous. Therefore, we adopt a video-by-video strategy, and instead of manually moving controllable objects, we propose to move cameras automatically. The procedure of generating mean videos as ground-truth is illustrated in Fig. 4. Specifically, we choose light-weight cameras to avoid clear shaking and misalignment, and fix the cameras to a rotatable tripod ball-head placed on top of a professional motorized slider. The slider and the tripod ball-head could be set to move and rotate at different speeds, which simulate the movement of observed objects in more than one directions. Both the slider and cameras are controlled remotely to reduce human interference. Noisy sequences are captured repeatedly to generate the corresponding ground-truth reference video by their mean value.

**Image averaging.** The image averaging strategy is simple and effective, thus it is widely adopted in several pioneering works [10], [11], which acquire the mean image by averaging more than 500 noisy images of the same scene. Ideally, the quality of the mean image will grow as the

<sup>1</sup>Datasets will be updated once new camera devices are available.

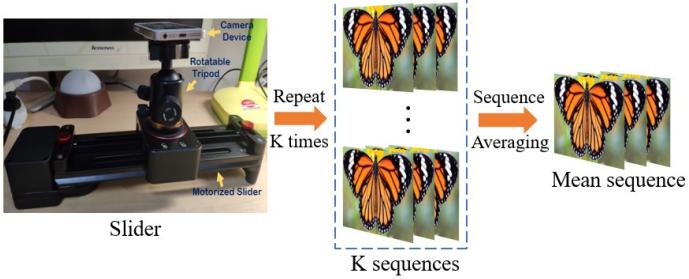


Fig. 4. The procedure of generating mean video sequences with a motorized slider. The camera is fixed to a rotatable tripod ball-head placed on top of the slider.

number of the noisy images increases. However, sampling a large number of noisy images is time-consuming, and it will cause irreversible damage to the camera devices and requires tremendous human efforts for preparation and post-processing, which may eventually lead to degraded results such as misalignment and blurry effects. We notice that huge progress of imaging devices and systems has been made to improve image quality and capture more details even under poor lighting conditions, as can be seen from Fig. 5. Therefore, we realize that the number of sampled



Fig. 5. Images captured by HUAWEI Mate40Pro in dark environments.

noisy images  $N_{noisy}$  may vary according to different camera devices and environments.

In our paper,  $N_{noisy}$  is selected empirically, and to demonstrate how  $N_{noisy}$  may be decided, we perform a toy experiment. Specifically, we fix the camera device (IPHONE 13) and capture images of the same static scene early in the morning, late in the afternoon and during midnight to simulate three different lighting conditions. The sampled noisy images are illustrated in Fig. 6. It can be observed that noise exists in different conditions, and when there is insufficient light, the image is more severely corrupted. For each lighting condition, the corresponding ground-truth image is obtained by averaging over 1000 images, then we vary the value  $N_{noisy}$  to generate different mean images and compare them with the ground-truth image. Fig. 7 shows the relative error and SSIM value, we can see that in low light environments, increasing  $N_{noisy}$  can help produce a higher quality noise-free reference image. In bright sunlight, however, the benefit of choosing a large  $N_{noisy}$  is not obvious. In practice, each indoor and outdoor ground-truth is obtained by averaging at least 100 and 30 noisy images of the same scene, respectively. Similarly, noisy video sequences are captured repeatedly for at least 30 times to generate the corresponding ground-truth reference video by their mean value. Since the data collection process often lasts dozens of seconds, to reduce misalignment and blurry

effects, most outdoor images are captured on the hottest and windless days during summer in Guangzhou, China.

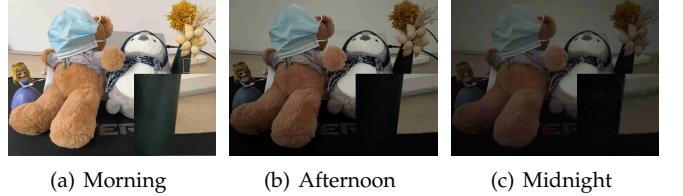


Fig. 6. Images of IPHONE13 under different lighting conditions.

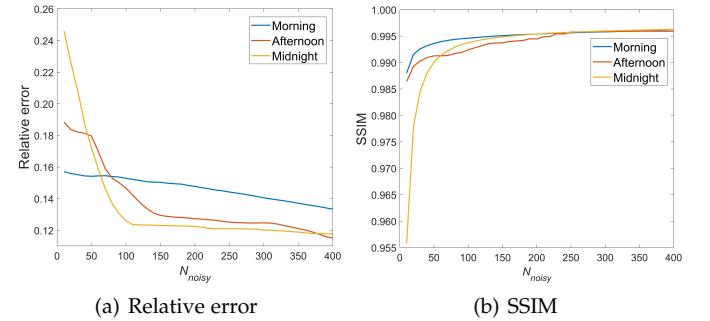


Fig. 7. Relative error and SSIM of different mean images compared to the ground-truth. The camera device is IPHONE 13.

### 3 EXPERIMENTS

#### 3.1 Real-world Image Denoising

Results of compared methods according to different according to alphabetical order are listed in Table 2. In addition, PSNR values of compared methods on both the validation and test sets of SIDD are given in Table 3. It is noticed that the results on the validation and test sets are almost the same, which indicates that their noise patterns are similar. Besides, since images of the training/validation and test sets are captured by the same cameras, they may also share similar features, as illustrated in Fig. 8. This interesting observation may help explain the outstanding performance of supervised methods.

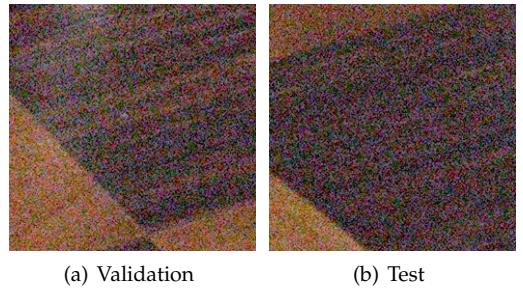


Fig. 8. Sample images from the validation and test sets of SIDD.

To better understand the performance of compared methods in real-world cases, more denoised images are presented in Fig. 9, Fig. 10 and Fig. 11. From Fig. 9, it can be seen that many DNN models are able to better suppress severe noise and thus provide more visually pleasant outputs. Meanwhile, they may also produce over-smooth effects, as

TABLE 2

Average PSNR, SSIM values on real-world color image datasets. The average time is calculated based on the PolyU dataset. '-' means the results are not available. 'C: CANON', 'F: FUJIFILM', 'H: HUAWEI', 'I: IPHONE', 'N: NIKON', 'O: OPPO', 'S: SONY', 'X: XIAOMI'.

Methods/Models	DnD	SIDD	CC15	CC60	PolyU	HighISO	IOCI														Time (s)	
							C-100D	C-600D	C-5DMark4	F-X100T	H-Honor6X	H-Mate40Pro	I-S5	I-6S	I-13	N-D5300	O-R11s	S-A6500	X-M18			
-	# Images	1000	1280	15	60	100	55	25	83	71	30	126	36	67	174	56	39	36	50			
4DHOSVD [32]	-	37.58	34.49	37.52	39.15	38.54	40.27	41.40	42.19	44.49	42.60	39.82	38.88	40.59	40.37	41.75	41.82	40.71	45.58	36.27	123.06	
Bitonic [33]	-	0.929	0.911	0.956	0.973	0.968	0.973	0.977	0.982	0.990	0.976	0.966	0.973	0.965	0.967	0.980	0.977	0.974	0.990	0.961		
CBMSD1 [6]	-	0.936	0.933	0.925	0.931	0.940	0.943	0.959	0.952	0.965	0.964	0.940	0.956	0.947	0.939	0.952	0.954	0.959	0.979	0.941	2.57	
CBMSD2 [6]	-	0.934	0.922	0.957	0.975	0.970	0.974	0.978	0.984	0.992	0.977	0.967	0.974	0.967	0.969	0.983	0.979	0.974	0.990	0.961	2.94	
CMSt-SVD [2]	-	38.04	35.00	37.58	39.21	38.60	40.11	41.58	42.40	44.43	42.53	39.93	38.84	40.56	40.41	41.93	42.29	40.62	45.58	36.24	1.02	
Traditional denoisers	#	0.938	0.925	0.955	0.971	0.969	0.972	0.977	0.982	0.990	0.976	0.966	0.973	0.967	0.970	0.982	0.979	0.973	0.990	0.959		
GID [34]	-	34.32	27.16	37.17	38.41	38.37	39.63	40.86	41.60	44.11	42.31	39.52	38.66	40.12	40.16	41.50	41.09	40.50	44.96	36.11	49.99	
LLRT [35]	-	0.817	0.634	0.946	0.963	0.967	0.967	0.974	0.979	0.989	0.975	0.965	0.972	0.964	0.967	0.979	0.973	0.973	0.989	0.960		
LSCD [36]	-	35.45	30.74	37.77	39.76	38.28	39.59	41.60	42.24	42.68	42.22	37.91	38.80	38.01	39.87	42.02	41.76	38.69	45.17	35.71	285.62	
MCWNNM [37]	-	37.38	29.54	37.02	38.54	38.26	39.89	41.47	42.07	44.22	42.48	39.46	38.78	40.18	41.33	41.74	40.71	45.38	35.84	237.57		
M-SVD	-	0.929	0.888	0.950	0.967	0.965	0.970	0.977	0.979	0.988	0.976	0.961	0.971	0.957	0.963	0.976	0.975	0.973	0.990	0.952		
NLHCC [38]	-	37.45	33.51	37.63	39.36	38.57	40.27	41.81	42.55	44.39	42.68	39.99	38.93	40.78	40.52	41.72	42.18	40.73	45.71	36.37	104.24	
NLHCC [38]	-	0.924	0.867	0.954	0.971	0.967	0.971	0.979	0.983	0.989	0.976	0.967	0.973	0.973	0.969	0.977	0.974	0.990	0.961			
TWSC [39]	-	37.96	-	37.90	39.66	38.62	40.62	41.65	42.52	44.94	42.26	38.71	38.81	38.77	40.12	41.71	42.23	40.65	45.49	35.40	350.30	
WTR1 [40]	-	-	37.80	37.32	38.20	38.19	-	-	-	-	-	-	-	-	-	-	-	-	-	600.33		
AINDNet [41]	-	39.77	39.08	36.14	37.19	37.33	38.00	38.54	39.33	39.49	38.50	36.53	36.24	36.93	36.83	37.27	38.11	37.44	40.17	34.65	0.33	
AP-BSN [42]	-	0.959	0.953	0.933	0.949	0.954	0.946	0.975	0.976	0.979	0.966	0.954	0.965	0.958	0.953	0.964	0.961	0.968	0.981	0.953		
Blind2Unblind [43]	-	37.29	35.97	35.44	36.76	36.99	38.26	40.23	41.29	42.72	41.40	37.87	37.62	37.47	38.29	40.83	40.86	39.55	43.73	34.37	3.56	
BRDNet [44]	-	36.51	38.18	38.25	39.03	40.71	41.72	43.11	41.17	39.35	38.27	39.39	39.04	40.88	40.90	39.76	43.78	35.75	0.74			
C2N [45]	-	33.80	-	37.27	39.16	38.04	39.64	39.85	40.95	43.54	40.27	39.48	38.66	38.19	40.07	41.92	41.34	39.66	42.76	36.08	14.66	
CBDNet [46]	-	37.28	-	37.02	-	37.69	38.86	41.12	41.95	42.61	41.33	38.73	38.15	38.95	38.67	40.50	40.78	40.05	44.64	35.53	88.00	
CFV-SID [47]	-	0.924	0.867	0.954	0.971	0.967	0.971	0.979	0.983	0.989	0.976	0.967	0.973	0.969	0.977	0.984	0.983	0.937	0.937			
CycleISP [48]	-	39.57	39.42	35.40	36.67	37.61	37.70	41.26	41.84	42.64	41.59	38.47	38.01	38.79	38.67	40.61	40.03	39.86	44.75	35.54	0.26	
DANet [49]	-	39.55	39.43	37.17	38.63	41.49	38.92	41.59	42.59	42.83	41.70	37.91	38.17	39.23	38.74	41.58	41.54	39.58	44.03	36.21	0.02	
DIDN [53]	-	0.953	0.956	0.953	0.969	0.977	0.960	0.979	0.981	0.980	0.971	0.964	0.960	0.951	0.968	0.963	0.965	0.981	0.950			
DnCNN [54]	-	37.90	37.73	37.47	39.32	38.51	40.01	40.81	41.91	44.16	41.57	39.92	38.72	39.34	40.22	42.03	42.12	40.26	43.97	36.30	3.23	
DRUNet [55]	-	39.63	39.35	36.63	-	37.70	36.93	40.90	40.86	41.22	39.72	33.67	36.40	33.16	33.16	32.07	33.33	36.92	38.40	35.22	38.56	29.89
DudeNet [56]	-	0.810	0.645	0.933	0.953	0.973	0.955	0.972	0.977	0.981	0.979	0.967	0.954	0.961	0.948	0.966	0.957	0.969	0.980	0.952		
FCCF [14]	-	36.30	-	39.02	40.89	38.87	41.20	42.03	43.07	45.15	42.80	40.14	39.04	40.61	40.66	42.31	43.05	40.88	45.47	36.72	11.20	
FFDNet [57]	-	0.885	-	0.968	0.982	0.971	0.978	0.980	0.993	0.993	0.974	0.966	0.968	0.970	0.973	0.964	0.969	0.983	0.974	0.989		
DNN methods	IDR [58]	-	-	34.26	36.18	36.28	36.44	39.33	40.31	40.94	37.38	37.39	38.30	38.15	38.77	38.27	38.58	40.03	40.32	39.73	44.07	35.28
InvDN [59]	-	39.57	39.28	34.55	36.33	35.94	38.01	38.95	40.09	42.03	40.02	33.32	37.00	36.64	37.66	40.06	40.98	39.10	40.74	34.15	0.74	
IRCNN [60]	-	0.952	0.955	0.937	0.953	0.947	0.952	0.965	0.973	0.978	0.970	0.960	0.950	0.959	0.966	0.963	0.966	0.969	0.973	0.969		
MIRNet [61]	-	34.79	35.55	37.26	37.25	37.49	39.66	41.24	41.85	43.89	41.78	39.91	38.63	38.01	38.72	39.44	39.77	39.32	40.45	34.82	2.15	
MPRNet [62]	-	0.954	0.958	0.935	-	0.954	0.946	0.974	0.974	0.978	0.969	0.952	0.966	0.971	0.974	0.976	0.977	0.972	0.980	0.981		
NAFNet [63]	-	38.34	40.15	34.39	-	36.38	37.88	40.12	40.23	40.66	40.01	36.13	35.87	36.66	36.79	36.53	39.93	39.32	40.45	34.82	0.35	
NBNNet [64]	-	39.89	39.64	35.89	-	37.37	38.32	40.95	41.91	42.92	41.17	38.02	37.87	37.79	38.45	40.51	40.45	39.73	43.96	35.37	0.12	
Neigh2Neigh [65]	-	-	34.47	35.65	37.10	36.59	40.22	40.41	41.29	41.41	42.48	38.71	-	-	-	-	-	-	-	2.48		
NLNet [66]	-	-	37.56	39.32	38.58	38.58	40.17	41.43	44.53	42.19	39.96	38.89	40.06	40.61	42.12	42.16	40.63	44.87	36.34			
Noise2Noise [67]	-	-	0.955	0.974	0.968	0.973	0.976	0.981	0.991	0.974	0.969	0.973	0.973	0.974	0.978	0.973	0.988	0.982	0.962			
PD-Denoising [68]	-	38.40	33.99	35.85	37.13	37.13	37.12	40.85	41.38	41.13	41.05	37.72	37.57	39.03	38.01	39.18	39.64	38.80	43.73	35.05	0.61	
Restormer [71]	-	40.03	40.02	36.33	-	37.66	38.29	41.10	41.84	42.49	41.47	38.42	38.08	39.05	38.77	40.13	40.53	39.56	44.19	35.65	0.77	
RIDNet [72]	-	39.26	38.70	36.84	38.11	38.57	38.60	41.09	42.15	42.39	41.59	38.88	38.21	38.67	39.02	40.91	41.01	40.07	44.40	35.76	0.11	
SADNet [73]	-	0.953	0.950	0.941	0.961	0.970	0.965	0.976	0.980	0.992	0.975	0.965	0.970	0.965	0.963	0.965	0.979	0.974	0.988	0.962		
SASL [74]	-	39.59	-	37.76	38.11	39.22	38.90	40.82	42.02	44.25	41.85	39.53	38.33	39.83	39.59	39.81	41.64	40.70	44.61	36.36		
SCUNet [75]	-	38.93	35.31	37.44	-	39.57	40.51	41.79	43.13	44.73	42.55	39.84	38.78	40.63	40.20	41.85	43.15	40.88	45.32	36.40	1.53	
PNGAN [69]	-	0.953	0.959	0.963	-	0.974	0.974	0.979	0.992	0.977	0.977	0.964	0.964	0.965	0.966	0.967	0.967	0.969	0.974	0.988	0.962	
R2R [70]	-	-	34.78	36.32	38.11	38.44	-	-	-	-	-	-	-	-	-	-	-	-	-	4481.98		
Restormer [71]	-	-	0.844	0.937	0.961	0.965	0.958	-</														

TABLE 3  
Average PSNR value of compared methods on the validation and test data of the SIDD datasets.

PSNR	Traditional denoisers							DNN methods										
	Bitonic	CBM3D2	CMSt-SVD	NLHCC	AINDNet	CycleISP	DANet	DeamNet	DIDN	InvDN	MIRNet	MPRNet	NAFNet	NBNNet	PNGAN	Restormer	RIDNet	VDIR
Validation	36.46	34.80	34.26	35.18	39.06	39.52	39.54	39.47	40.04	39.32	39.68	39.71	40.21	39.76	40.07	39.93	38.78	39.37
Test	36.68	34.74	34.38	35.31	39.08	39.35	39.78	39.35	39.78	39.28	39.55	39.57	40.15	39.64	39.81	40.02	38.70	39.26
Difference	0.22	0.06	0.12	0.13	0.02	0.17	0.24	0.12	0.26	0.04	0.13	0.14	0.06	0.12	0.26	0.14	0.08	0.11

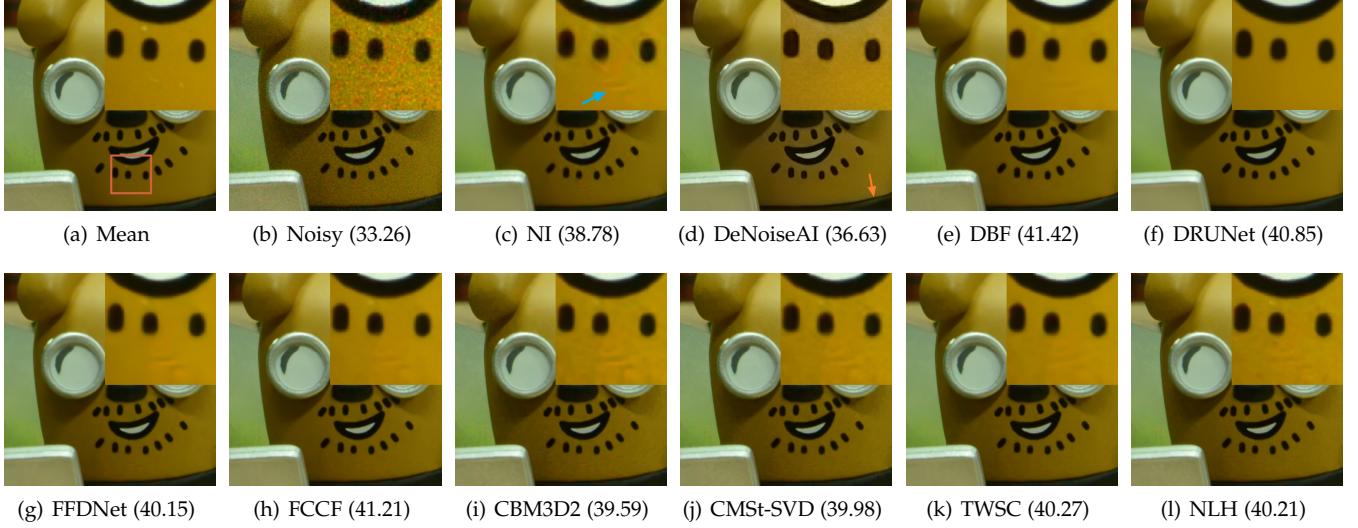


Fig. 9. Visual evaluations of compared methods (PSNR) on CC15 dataset.

illustrated in Fig. 10 and Fig. 11. The powerful commercial software DeNoiseAI improves the visual quality in terms of both noise removal and detail recovery, but it also produces some color artifacts. An interesting example presented in Fig. 10 shows that DeNoiseAI tends to leave unwanted artifacts along the edges, which are barely noticeable in noisy observation and results of other compared methods. This may be caused by its unique image sharpening technique, which amplifies the influence of noise.

### 3.2 Video Denoising

Table 4 reports denoising results of compared methods on the set8, DAVIS, CRVD and IOCV datasets. FloRNN shows state-of-the-art performance in both synthetic and real-world experiments. Visual evaluation of compared methods on the IOCV dataset is presented in Fig. 12. Due to the limitations of hardware equipments and environment, the mean videos generated with our video averaging strategy also inevitably exhibit some noise, flickering, staircase effects, motion blur and misalignment. Furthermore, we notice that in this case most compared methods struggle to find a balance between noise removal and detail preservation.

### 3.3 MSI/HSI Denoising

Apart from the CAVE [19] dataset, the ICVL [20] dataset is also commonly used for synthetic experiments. Considering the large size of the ICVL dataset, we select the first 20 image data and evaluate two representative methods, namely MSt-SVD and QRNN3D. The average PSNR and SSIM values are reported in Table 5. Comparing the results listed in Table 5 of the supplemental material and Table 7 in the

paper, we notice that QRNN3D performs much better on the ICVL dataset than on the CAVE dataset. We notice that QRNN3D is trained with the ICVL dataset, and like many other supervised DNN methods, the pretrained model of QRNN3D does not guarantee good denoising performance on other datasets.

TABLE 5  
Average PSNR/SSIM and computational time (minutes) of MSt-SVD and QRNN3D under i.i.d Gaussian noise  $\sigma \in \{10, 20, 40\}$  on the first 20 MSI data of ICVL dataset.

Methods	$\sigma = 10$	$\sigma = 20$	$\sigma = 40$	Time (m)
MSt-SVD	46.58/0.9868	42.98/0.9746	39.32/0.9525	10.1
QRNN3D	<b>47.37/0.9908</b>	<b>45.08/0.9842</b>	<b>42.38/0.9731</b>	<b>0.16</b>

### 3.4 MRI Denoising

Fig. 13 illustrates denoised results of compared methods on the fastMRI dataset at different Rician noise levels. It can be seen that at low noise level ( $\sigma = 3\%$ ), all compared methods are able to effectively suppress noise and restore details and structures, however, when the MR image is severely corrupted ( $\sigma = 19\%$ ), the denoising performance is far from satisfactory. Nevertheless, the predefined transforms of BM4D renders certain robustness, making BM4D less affected by the increase of the noise level.

### 3.5 Discussion

In this section, we present more observations and analysis based on experimental results.

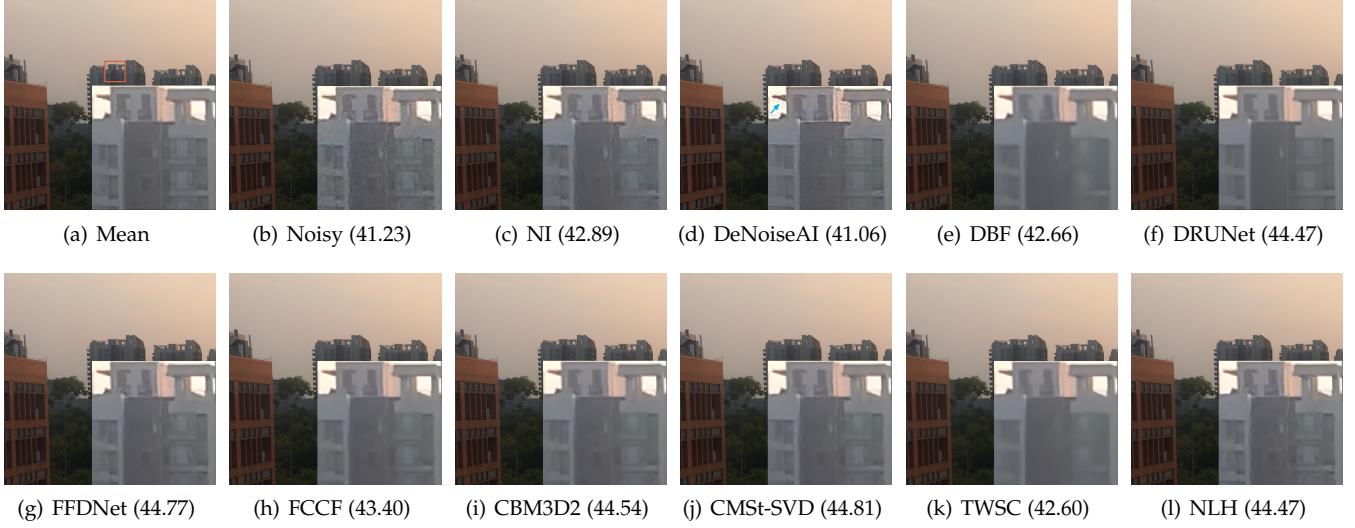


Fig. 10. Visual evaluation of compared methods (PSNR) on the IOCI dataset. The camera device is IPHONE 5S.

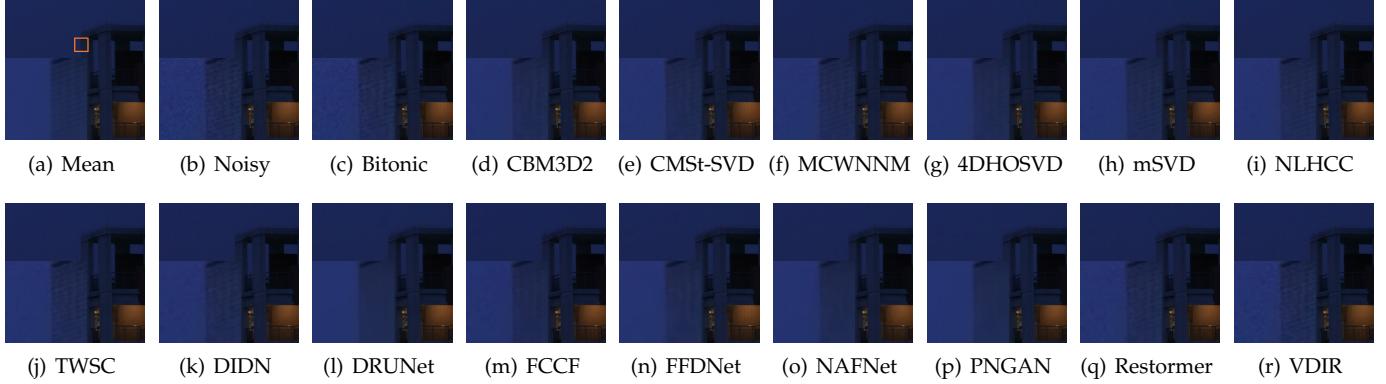


Fig. 11. Visual evaluation of compared methods on the IOCI dataset. The camera device is SONYA6500.

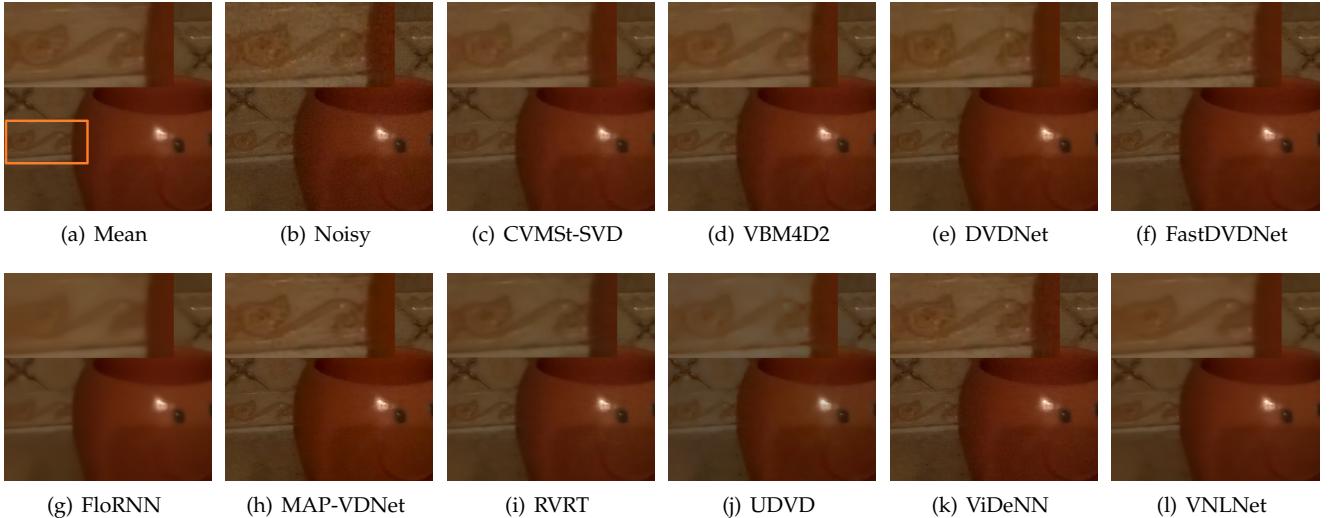


Fig. 12. Visual evaluation of video denoising on the IOCV dataset. The camera device is OPPO R11s.

TABLE 4

Average PSNR and SSIM values of compared methods on four video datasets (sRGB space). The average computational time (s) is calculated based on the Set8 dataset with noise level  $\sigma = 50$ . '-' means the results are not available due to high computational cost or other issues.

Dataset	Noise level	Traditional_denoisers					DNN methods								
		CVMSst-SVD	RTA-LSM	VBM4D1	VBM4D2	DVDNet	FastDVDNet	FloRNN	MAP-VDNet	MMNet	RFR	RVRT	UDVD	ViDeNN	VNLNet
Set8	10	36.25	36.36	35.69	35.96	36.04	36.38	37.55	-	37.28	-	<b>37.58</b>	32.30	34.80	37.28
		0.9453	0.9474	0.9342	0.9405	0.9487	0.9513	0.9617	-	0.9595	-	<b>0.9626</b>	0.9081	0.9114	0.9584
	20	32.50	32.87	31.56	32.10	33.42	33.35	34.65	-	34.11	30.95	<b>34.85</b>	32.24	28.34	34.02
		0.8891	0.8977	0.8485	0.8731	0.9151	0.9159	0.9347	-	0.9270	0.8321	<b>0.9383</b>	0.8937	0.7619	0.9243
	30	30.38	30.84	29.20	29.90	31.70	31.59	32.94	-	32.27	30.18	<b>33.31</b>	31.02	-	-
		0.8369	0.8542	0.7626	0.8081	0.8825	0.8848	0.9100	-	0.8973	0.8299	<b>0.9173</b>	0.8653	-	-
	40	28.92	29.41	27.56	28.37	30.45	30.36	31.71	-	30.98	28.89	<b>32.21</b>	29.46	-	30.72
		0.7871	0.8170	0.6820	0.7482	0.8528	0.8569	0.8869	-	0.8702	0.7891	<b>0.8981</b>	0.8221	-	0.8596
	50	27.80	28.34	26.29	27.20	29.45	29.42	30.75	26.48	29.99	26.57	<b>31.32</b>	27.89	-	-
		0.7392	0.7860	0.6086	0.6935	0.8264	0.8317	0.8653	0.7220	0.8452	0.6461	<b>0.8800</b>	0.7654	-	-
DAVIS-2017-test-dev	10	38.49	-	37.57	37.97	37.70	38.33	38.92	-	38.25	-	<b>39.07</b>	33.65	-	38.68
		0.9611	-	0.9475	0.9555	0.9600	0.9631	0.9680	-	0.9639	-	<b>0.9691</b>	0.9204	-	0.9653
	20	34.73	-	33.33	34.03	35.34	35.52	36.43	-	35.63	33.09	<b>36.69</b>	33.95	-	35.84
		0.9147	-	0.8709	0.9000	0.9361	0.9364	0.9493	-	0.9384	0.8765	<b>0.9520</b>	0.9154	-	0.9398
	30	32.53	-	30.82	31.71	33.84	33.88	35.06	-	34.06	31.93	<b>35.47</b>	32.85	-	-
		0.8665	-	0.7866	0.8415	0.9140	0.9134	0.9341	-	0.9166	0.8564	<b>0.9394</b>	0.8918	-	-
	40	30.96	-	29.02	30.06	32.70	32.71	34.03	-	32.93	30.22	<b>34.58</b>	31.29	-	33.01
		0.8182	-	0.7036	0.7838	0.8931	0.8926	0.9195	-	0.8969	0.7997	<b>0.9280</b>	0.8545	-	0.8959
	50	29.72	-	27.61	28.78	31.76	31.79	33.18	28.35	32.04	27.22	<b>33.85</b>	29.65	-	-
		0.7701	-	0.6270	0.7287	0.8730	0.8735	0.9054	0.7911	0.8788	0.6350	<b>0.9170</b>	0.8033	-	-
DAVIS-2017-test-challenge	10	38.90	-	37.69	38.02	37.80	38.55	39.12	-	38.47	-	<b>39.29</b>	33.95	36.30	38.96
		0.9644	-	0.9476	0.9542	0.9609	0.9648	0.9690	-	0.9656	-	<b>0.9701</b>	0.9295	0.9254	0.9669
	20	35.15	-	33.53	34.13	35.46	35.77	36.56	-	35.85	33.25	<b>36.83</b>	34.34	-	36.06
		0.9244	-	0.8737	0.9002	0.9391	0.9415	0.9515	-	0.9425	0.8787	<b>0.9541</b>	0.9246	-	0.9441
	30	32.91	-	31.02	31.82	33.99	34.14	35.18	-	34.26	31.75	<b>35.56</b>	33.23	-	-
		0.8811	-	0.7919	0.8440	0.9201	0.9214	0.9380	-	0.9230	0.8546	<b>0.9426</b>	0.9034	-	-
	40	31.29	-	29.22	30.16	32.85	32.96	34.17	-	33.11	29.87	<b>34.65</b>	31.66	-	33.20
		0.8357	-	0.7106	0.7883	0.9025	0.9033	0.9257	-	0.9055	0.8058	<b>0.9325</b>	0.8682	-	0.9062
	50	30.00	-	27.80	28.87	31.89	32.03	33.33	28.60	32.20	27.62	<b>33.91</b>	29.99	-	-
		0.7890	-	0.6346	0.7351	0.8856	0.8867	0.9138	0.8091	0.8894	0.7201	<b>0.9229</b>	0.8172	-	-
DAVIS-2019-unsupervised	10	39.45	-	38.05	38.41	37.72	38.96	39.56	-	39.00	-	<b>39.76</b>	33.94	36.93	39.51
		0.9696	-	0.9527	0.9591	0.9628	0.9692	0.9722	-	0.9700	-	<b>0.9738</b>	0.9330	0.9385	0.9712
	20	35.55	-	33.72	34.37	35.48	36.15	37.00	-	36.24	33.90	<b>37.29</b>	34.64	31.78	36.57
		0.9327	-	0.8811	0.9082	0.9435	0.9483	0.9565	-	0.9492	0.9008	<b>0.9594</b>	0.9325	0.8625	0.9509
	30	33.21	-	31.12	31.96	34.04	34.47	35.61	-	34.55	32.62	<b>36.00</b>	33.49	-	-
		0.8908	-	0.7991	0.8528	0.9265	0.9298	0.9445	-	0.9306	0.8821	<b>0.9490</b>	0.9115	-	-
	40	31.51	-	29.25	30.24	32.91	33.23	34.58	-	33.32	30.82	<b>35.07</b>	31.83	-	33.56
		0.8459	-	0.7173	0.7972	0.9104	0.9126	0.9336	-	0.9136	0.8281	<b>0.9399</b>	0.8760	-	0.9149
	50	30.17	-	27.79	28.90	31.95	32.25	33.74	29.33	32.36	27.73	<b>34.32</b>	30.08	-	-
		0.7997	-	0.6408	0.7439	0.8949	0.8964	0.9232	0.8472	0.8975	0.6578	<b>0.9312</b>	0.8233	-	-
CRVD	-	36.66	-	34.66	34.14	34.50	35.84	36.66	-	-	31.30	<b>36.94</b>	-	32.31	36.11
	-	0.9463	-	0.9224	0.9079	0.9493	0.9306	<b>0.9605</b>	-	-	0.7785	0.9559	-	0.8449	0.9449
IOCV	-	38.22	-	38.65	<b>38.76</b>	38.53	37.57	38.64	35.52	-	31.46	38.50	35.02	36.13	<b>38.76</b>
	-	0.9736	-	0.9763	<b>0.9765</b>	0.9754	0.9699	0.9743	0.9313	-	0.8346	0.9672	0.9660	0.9506	<b>0.9765</b>
Implementation	-	MEX	MATLAB	MEX	MEX	Python	Python	Python	Python	Python	Python	Python	Python	Python	Python
Time (s)	-	888.0	~1h	920.9	2076.0	818.8	<b>6.6</b>	53.9	77.2	30.1	37.8	93.3	163.4	19.0	217.1

### 3.5.1 The Cost of Pre-Training

The strength of many DNN methods lies mainly in their outstanding capability of exploiting extra information to capture noise patterns and customize camera-aware noise models. Despite all the bright side and advantages of DNN frameworks, the training process can be time-consuming. To investigate the cost of pre-training, we use DANet trained with the PolyU dataset as an example. Specifically, for each camera of the PolyU dataset, we randomly select 80% of clean-noisy pairs for training, and the rest 20% for testing. The training dataset is augmented via random flipping and rotation. We use the code provided by the authors to train the DANet model for up to 60 epochs, and each epoch costs about 9 minutes run on the GPU device. From Fig. 14, it is observed that DANet slightly outperforms CBM3D2 on the

test set after 30 training epochs.

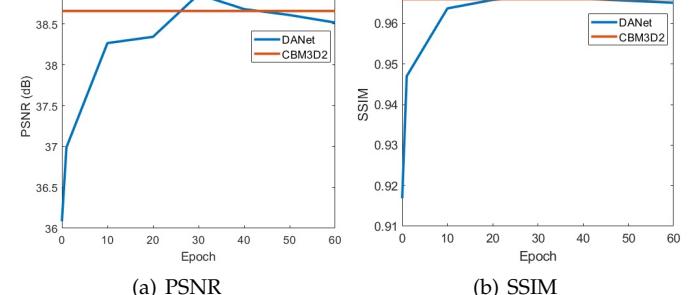


Fig. 14. PSNR and SSIM values of DANet on the PolyU test set w.r.t different training epochs.

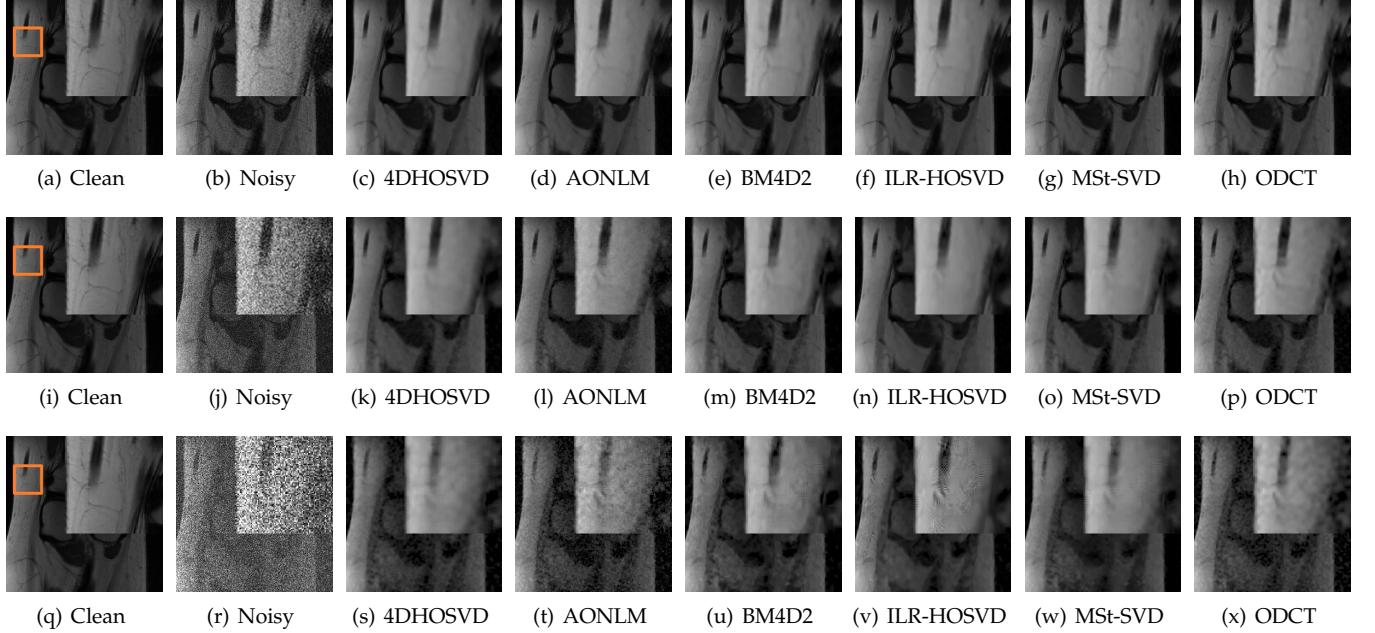


Fig. 13. Visual evaluation of the fastMRI dataset at different Rician noise levels. From the first row to the third row:  $\sigma = 3\%$ ,  $\sigma = 11\%$  and  $\sigma = 19\%$ .

In real-world applications, it may be impractical to train a model for each camera setting due to insufficient training data and high computational burdens. Recently, some few-shot domain transfer schemes [50] are proposed to enable model reuse and also reduce the load of re-training, thus it is interesting to further investigate their effectiveness on more camera settings.

### 3.5.2 The Robustness of BM3D

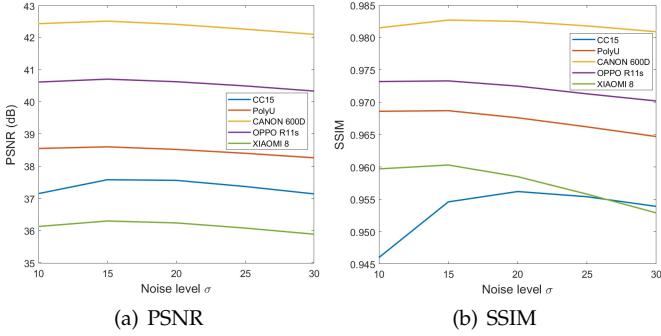


Fig. 15. PSNR and SSIM change of CBM3D1 with noise level  $\sigma \in [10, 30]$  on five real-world color image datasets.

Recently, there is some criticism [50] of BM3D that to achieve competitive denoising performance, it may need to extensively traverse different Gaussian noise levels  $\sigma$ , which is potentially time-consuming and impractical without ground-truth. Such criticism may be unfair for BM3D for two reasons. First, compared to nearly all other methods, BM3D does not involve any model training process which can be influenced by the noise distributions and patterns of specific datasets. Besides, in our experiments, we discover that the predefined transforms of BM3D are not very sensitive to the choice of the input noise level parameter  $\sigma$  within a reasonable range. Fig. 15 compares the PSNR and

SSIM values of CBM3D1 with  $\sigma \in [10, 30]$  on five datasets. It is noticed that choosing a noise level  $\sigma$  between 10 and 20 could yield similarly good performance for CBM3D, making it an exceedingly robust denoiser. Furthermore, from the perspective of blind denoising [46], [81], [82], [83], CBM3D could be viewed as a competitive blind denoiser with input noise level fixed to be  $\sigma = 15$ .

### 3.5.3 Denoising with Resizing

To boost the denoising performance of traditional patch-based denoisers, many methods [1], [32], [35], [37], [39], [40], [84] propose to iteratively combine the filtered image with the corresponding noisy observation in different ways [85], [86], [87], [88]. A simple approach is to add back the noisy observation  $\mathcal{Y}$  to the denoised image  $\hat{\mathcal{X}}$  via

$$\hat{\mathcal{Y}} = \lambda \mathcal{Y} + (1 - \lambda) \hat{\mathcal{X}} \quad (18)$$

where  $\lambda \in [0, 1]$  is a relaxation parameter reflecting the importance of  $\mathcal{Y}$ . The new noisy image  $\hat{\mathcal{Y}}$  is then fed into the denoiser for final estimate. The iterative denoising strategies may significantly increase the computational burden, and the results reported in our paper show that a simple one-stage implementation such as MSt-SVD and BMXD1 could produce competitive performance at moderate noise levels. It is therefore interesting to investigate how these effective methods can be modified to further handle severe noise with traditional denoisers. Recently, Zontak et al. [89] show that in the down-sampled noisy image, patches tend to be noiseless and share similar patterns with underlying clean patches. Therefore, instead of directly filtering the large-size noisy observation, an alternative is to first handle down-sized image, and then upscale the denoised image back to its original size with some effective image super-resolution algorithms [90], [91]. This idea is similar to the encoder-decoder scheme of DNN methods. For simplicity we use MATLAB's built-in `imresize` function and applies it to the efficient CMSt-SVD. This special implementation is termed

as CMSt-SVD\_R. The visual effects of CMSt-SVD with and without the image resizing strategy are compared in Fig. 16. As can be seen, the resizing strategy could effectively reduce color artifacts at the cost of slight over-smooth effects, thus it may be more effective and suitable for very noisy images with fewer fine details and textures.

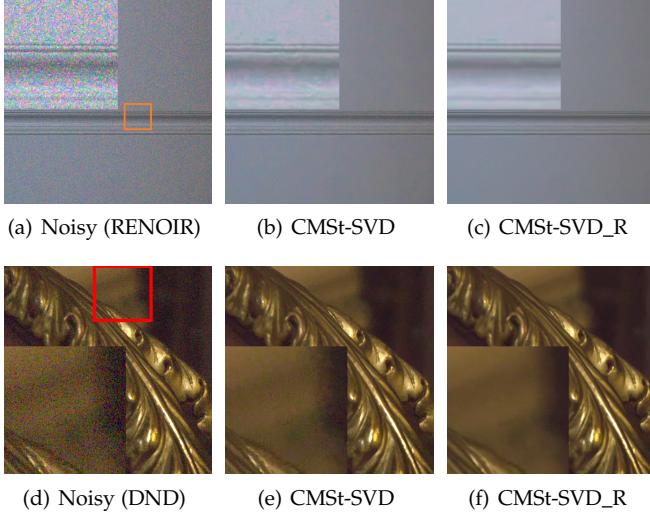


Fig. 16. Visual evaluation of CMSt-SVD with and without the image resizing strategy on RENOIR (first row) and DND (second row) datasets.

### 3.5.4 A Closer Look at the Effectiveness of Higher-order SVD (HOSVD) for Denoising

Our real-world experiments of MRI denoising show that the SVD-based methods could produce very competitive performance compared with HOSVD-based approaches, which are also prone to over-smooth textures. With such observation, we further investigate and discuss the effectiveness of HOSVD for denoising. Our analysis in this section is carried out with the results of [3], [4], [5], [92], [93], [94].

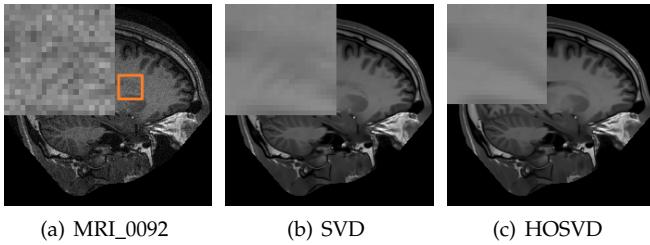


Fig. 17. Comparison of SVD and HOSVD for real MRI denoising.

**The risk of truncated HOSVD for denoising.** To filter out noise, HOSVD applies hard-thresholding technique to shrink core tensor  $\mathcal{C}$  or truncates the projection matrices  $\mathbf{U}$  and  $\mathbf{V}$ , which is similar to the low-rank assumption of SVD. It is noticed that HOSVD also needs to reshape image patches into vectors, and by learning projection matrices of different modes, it may better capture more high-dimensional information. However, the HOSVD transform does not guarantee better detail preservation, as illustrated in Fig. 17. To understand the over-smooth effects of truncated HOSVD (T-HOSVD), we use a noise-free group  $\mathcal{G} \in \mathbb{R}^{3 \times 3 \times 3}$  consisting

of three same patches with  $\mathcal{G}(:,:,1) = \mathcal{G}(:,:,2) = \mathcal{G}(:,:,3)$ , and the third-mode unfolding of  $\mathcal{G}$  is

$$\mathbf{G}_{(3)} = \begin{pmatrix} 3 & 1 & 5 & 6 & 4 & 8 & 9 & 2 & 6 \\ 3 & 1 & 5 & 6 & 4 & 8 & 9 & 2 & 6 \\ 3 & 1 & 5 & 6 & 4 & 8 & 9 & 2 & 6 \end{pmatrix} \quad (19)$$

where every row of  $\mathbf{G}_{(3)}$  corresponds to a vectorized image patch. In this case, the low-rank SVD method would not suffer from loss of information, since  $\mathbf{G}_{(3)}$  has only one non-zero singular value. For HOSVD, According to Eq. (16), the core tensor  $\mathcal{C}$  is

$$\mathbf{C}_{(3)} = \begin{pmatrix} -27.98 & 0 & 0 & 0 & -5.38 & 0 & 0 & 0 & 2.07 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad (20)$$

Assume that the pre-defined multi-rank  $r_1 = r_2 = 2$  is applied to truncate  $\mathbf{U}_1$  and  $\mathbf{U}_2$  with  $\mathbf{U} = \mathbf{U}(:, 1 : 2)$ , then after the inverse transform of Eq. (16), the corresponding denoised group  $\hat{\mathcal{G}}_c$  is

$$\begin{pmatrix} 3.09 & 1.92 & 4.54 & 5.95 & 3.46 & 8.26 & 9.00 & 2.03 & 5.98 \\ 3.09 & 1.92 & 4.54 & 5.95 & 3.46 & 8.26 & 9.00 & 2.03 & 5.98 \\ 3.09 & 1.92 & 4.54 & 5.95 & 3.46 & 8.26 & 9.00 & 2.03 & 5.98 \end{pmatrix} \quad (21)$$

Comparing Eq. (19) and Eq. (21), the loss of information is  $\|\mathbf{G}_{(3)} - \hat{\mathbf{G}}_{c(3)}\|_F^2 \approx 2.07^2 \approx 4.28$ . Also, from Eq. (20) it is noticed that the hard-thresholding technique will yield the same loss if the threshold parameter  $\tau$  is set to  $2.07 < \tau < 5.38$ . Fig. 18 illustrates the over-smooth effects of T-HOSVD when applied to a noise-free image with all the patches in a group are forced to be the same.

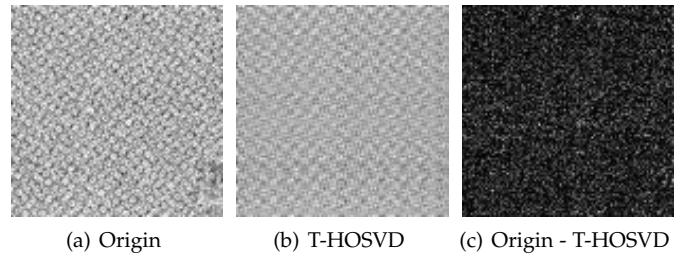


Fig. 18. Illustration of over-smooth effects produced by truncated HOSVD (T-HOSVD) when all the patches in a group are the same.

In real-world applications, the difference among nonlocal similar patches is not negligible, and the use of tensor decomposition techniques may help filter out more redundant information and extract latent features, as is observed in the MRI denoising results illustrated in Fig. 13. Also, it is noticed from Fig. 18 that certain structural shapes and textures are preserved. However, despite recent development of the tensor theory [95], the choice of the best multi-rank and threshold parameter  $\tau$  for the tensor truncation strategy still remains a challenge.

### 3.5.5 More Image Denoising Problems And Applications

The advancement of imaging techniques has brought challenges and also opened up fantastic probabilities to the realm of image denoising. In addition to the limited applications investigated in the paper, image denoising plays an important role in many different fields such as computed tomography [96], fluorescence microscopy [97] and seismic

exploration [98]. Besides, to verify the value of image denoising in downstream tasks, some works have recently been proposed to bridge the gap between denoising and other image processing tasks. For example, Zhao et al. [99] utilize hierarchical network for joint denoising and deblurring of night images. Guo et al. [100] handle denoising and demosaicing with green channel prior. Liu et al. [101] connect denoising and high-level tasks (classification and semantic segmentation) by utilizing joint loss to guide the training of denoising network.

Despite the achievements in recent years, the state-of-the-art approaches still suffer from limitations that restrict their application to certain real-world scenarios. For example, from Fig. 10 and Fig. 11, it is noticed that many image denoising methods show only limited enhancement and may even degrade the image quality with over-smoothness and annoying artifacts. As image size grows rapidly with the development of new imaging technologies, a challenging problem is to decide if it is worth applying filtering to an image at hand [102], and if denoising is desired, which of the denoisers should be used.

## REFERENCES

- [1] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Color image denoising via sparse 3d collaborative filtering with grouping constraint in luminance-chrominance space," in *Proc. IEEE Int. Conf. Image Process.*, 2007, pp. 313–316.
- [2] Z. Kong and X. Yang, "Color image and multispectral image denoising using block diagonal representation," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4247–4259, 2019.
- [3] J. Kamm and J. G. Nagy, "Kronecker product and svd approximations in image restoration," *Linear Algebra Appl.*, vol. 284, no. 1-3, pp. 177–192, 1998.
- [4] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.
- [5] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.
- [6] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Trans. Image Process.*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [7] 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 *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, July 2001, pp. 416–423.
- [8] Kodak, "Kodak gallery dataset," [Online]. Available: <http://r0k.us/graphics/kodak>.
- [9] J. Anaya and A. Barbu, "Renoir—a dataset for real low-light image noise reduction," *J. Vis. Commun. Image Represent.*, vol. 51, pp. 144–154, 2018.
- [10] S. Nam, Y. Hwang, Y. Matsushita, and S. Joo Kim, "A holistic approach to cross-channel image noise modeling and its application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1683–1691.
- [11] J. Xu, H. Li, Z. Liang, D. Zhang, and L. Zhang, "Real-world noisy image denoising: A new benchmark," *arXiv preprint arXiv:1804.02603*, 2018.
- [12] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1586–1595.
- [13] A. Abdelhamed, S. Lin, and M. S. Brown, "A high-quality denoising dataset for smartphone cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1692–1700.
- [14] H. Yue, J. Liu, J. Yang, T. Q. Nguyen, and F. Wu, "High iso jpeg image denoising by deep fusion of collaborative and convolutional filtering," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4339–4353, 2019.
- [15] Derf, "Derfs test media collection," [Online]. Available: <https://media.xiph.org/video/derf>.
- [16] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 724–732.
- [17] H. Yue, C. Cao, L. Liao, R. Chu, and J. Yang, "Supervised raw video denoising with a benchmark dataset on dynamic scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2301–2310.
- [18] X. Xu, Y. Yu, N. Jiang, J. Lu, B. Yu, and J. Jia, "Pvdd: A practical video denoising dataset with real-world dynamic scenes," *arXiv preprint arXiv:2207.01356*, 2022.
- [19] F. Yasuma, T. Mitsunaga, D. Iso, and S. Nayar, "Generalized Assorted Pixel Camera: Post-Capture Control of Resolution, Dynamic Range and Spectrum," Tech. Rep., Nov 2008.
- [20] B. Arad and O. Ben-Shahar, "Sparse recovery of hyperspectral signal from natural rgb images," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 19–34.
- [21] M. F. Baumgardner, L. L. Biehl, and D. A. Landgrebe, "220 band aviris hyperspectral image data set: June 12, 1992 indian pine test site 3," Sep 2015. [Online]. Available: <https://purr.purdue.edu/publications/1947/1>
- [22] A. Chakrabarti and T. Zickler, "Statistics of real-world hyperspectral images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 193–200.
- [23] T. Zhang, Y. Fu, and C. Li, "Hyperspectral image denoising with realistic data," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2248–2257.
- [24] C. A. Cocosco, V. Kollokian, R. K.-S. Kwan, G. B. Pike, and A. C. Evans, "Brainweb: Online interface to a 3d mri simulated brain database," in *NeuroImage*, 1997.
- [25] J. Zbontar, F. Knoll, A. Sriram, T. Murrell, Z. Huang, M. J. Muckley, A. Defazio, R. Stern, P. Johnson, M. Bruno *et al.*, "fastmri: An open dataset and benchmarks for accelerated mri," *arXiv preprint arXiv:1811.08839*, 2018.
- [26] D. S. Marcus, T. H. Wang, J. Parker, J. G. Csernansky, J. C. Morris, and R. L. Buckner, "Open access series of imaging studies (oasis): cross-sectional mri data in young, middle aged, nondemented, and demented older adults," *J. Cogn. Neurosci.*, vol. 19, no. 9, pp. 1498–1507, 2007.
- [27] Z. Kong and X. Yang, "A brief review of real-world color image denoising," *arXiv preprint arXiv:1809.03298*, 2018.
- [28] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.
- [29] S. Caelles, J. Pont-Tuset, F. Perazzi, A. Montes, K.-K. Maninis, and L. Van Gool, "The 2019 davis challenge on vos: Unsupervised multi-object segmentation," *arXiv preprint arXiv:1905.00737*, 2019.
- [30] J. M. Amigo and S. Grassi, "Configuration of hyperspectral and multispectral imaging systems," in *Data Handling in Science and Technology*, 2020, vol. 32, pp. 17–34.
- [31] A. S. J. T. Morteza Ghahremani, Mohammad Khateri, "Adversarial distortion learning for medical image denoising," *arXiv:2204.14100*, 2022.
- [32] A. Rajwade, A. Rangarajan, and A. Banerjee, "Image denoising using the higher order singular value decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 4, pp. 849–862, 2012.
- [33] G. Treece, "Real image denoising with a locally-adaptive bitonic filter," *IEEE Trans. Image Process.*, vol. 31, pp. 3151–3165, 2022.
- [34] J. Xu, L. Zhang, and D. Zhang, "External prior guided internal prior learning for real-world noisy image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 6, pp. 2996–3010, 2018.
- [35] Y. Chang, L. Yan, and S. Zhong, "Hyper-laplacian regularized unidirectional low-rank tensor recovery for multispectral image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4260–4268.
- [36] M. Rizkinia, T. Baba, K. Shirai, and M. Okuda, "Local spectral component decomposition for multi-channel image denoising," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3208–3218, 2016.
- [37] J. Xu, L. Zhang, D. Zhang, and X. Feng, "Multi-channel weighted nuclear norm minimization for real color image denoising," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1096–1104.
- [38] Y. Hou, J. Xu, M. Liu, G. Liu, L. Liu, F. Zhu, and L. Shao, "Nlh: A blind pixel-level non-local method for real-world image denoising," *IEEE Trans. Image Process.*, vol. 29, pp. 5121–5135, 2020.

- [39] J. Xu, L. Zhang, and D. Zhang, "A trilateral weighted sparse coding scheme for real-world image denoising," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 20–36.
- [40] Y. Wu, L. Fang, and S. Li, "Weighted tensor rank-1 decomposition for nonlocal image denoising," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2719–2730, 2018.
- [41] Y. Kim, J. W. Soh, G. Y. Park, and N. I. Cho, "Transfer learning from synthetic to real-noise denoising with adaptive instance normalization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3482–3492.
- [42] W. Lee, S. Son, and K. M. Lee, "Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17725–17734.
- [43] Z. Wang, J. Liu, G. Li, and H. Han, "Blind2unblind: Self-supervised image denoising with visible blind spots," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2027–2036.
- [44] C. Tian, Y. Xu, and W. Zuo, "Image denoising using deep cnn with batch renormalization," *Neural Networks*, vol. 121, pp. 461–473, 2020.
- [45] G. Jang, W. Lee, S. Son, and K. M. Lee, "C2n: Practical generative noise modeling for real-world denoising," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 2350–2359.
- [46] S. Guo, Z. Yan, K. Zhang, W. Zuo, and L. Zhang, "Toward convolutional blind denoising of real photographs," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1712–1722.
- [47] R. Neshatavar, M. Yavartanoo, S. Son, and K. M. Lee, "Cvf-sid: Cyclic multi-variate function for self-supervised image denoising by disentangling noise from image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 17583–17591.
- [48] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Cycleisp: Real image restoration via improved data synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2696–2705.
- [49] Z. Yue, Q. Zhao, L. Zhang, and D. Meng, "Dual adversarial network: Toward real-world noise removal and noise generation," in *Proc. Eur. Conf. Comput. Vis.*, 2020.
- [50] C. Chen, Z. Xiong, X. Tian, Z.-J. Zha, and F. Wu, "Real-world image denoising with deep boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019.
- [51] H. Zheng, H. Yong, and L. Zhang, "Deep convolutional dictionary learning for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 630–641.
- [52] C. Ren, X. He, C. Wang, and Z. Zhao, "Adaptive consistency prior based deep network for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 8596–8606.
- [53] S. Yu, B. Park, and J. Jeong, "Deep iterative down-up cnn for image denoising," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019.
- [54] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [55] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, "Plug-and-play image restoration with deep denoiser prior," *arXiv preprint arXiv:2008.13751*, 2020.
- [56] C. Tian, Y. Xu, W. Zuo, B. Du, C.-W. Lin, and D. Zhang, "Designing and training of a dual cnn for image denoising," *Knowledge-Based Systems*, vol. 226, p. 106949, 2021.
- [57] K. Zhang, W. Zuo, and L. Zhang, "Ffdnet: Toward a fast and flexible solution for cnn-based image denoising," *IEEE Trans. Image Process.*, vol. 27, no. 9, pp. 4608–4622, 2018.
- [58] Y. Zhang, D. Li, K. L. Law, X. Wang, H. Qin, and H. Li, "Idr: Self-supervised image denoising via iterative data refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 2098–2107.
- [59] Y. Liu, Z. Qin, S. Anwar, P. Ji, D. Kim, S. Caldwell, and T. Gedeon, "Invertible denoising network: A light solution for real noise removal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13365–13374.
- [60] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3929–3938.
- [61] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, "Learning enriched features for real image restoration and enhancement," in *Proc. Eur. Conf. Comput. Vis.*, 2020.
- [62] —, "Multi-stage progressive image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14 821–14 831.
- [63] L. Chen, X. Chu, X. Zhang, and J. Sun, "Simple baselines for image restoration," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2022, pp. 17–33.
- [64] S. Cheng, Y. Wang, H. Huang, D. Liu, H. Fan, and S. Liu, "Nbnet: Noise basis learning for image denoising with subspace projection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4896–4906.
- [65] T. Huang, S. Li, X. Jia, H. Lu, and J. Liu, "Neighbor2neighbor: Self-supervised denoising from single noisy images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 14 781–14 790.
- [66] S. Lefkimiatis, "Non-local color image denoising with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3587–3596.
- [67] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2noise: Learning image restoration without clean data," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2965–2974.
- [68] Y. Zhou, J. Jiao, H. Huang, Y. Wang, J. Wang, H. Shi, and T. Huang, "When awgn-based denoiser meets real noises," in *Proc. AAAI*, vol. 34, no. 07, 2020, pp. 13 074–13 081.
- [69] Y. Cai, X. Hu, H. Wang, Y. Zhang, H. Pfister, and D. Wei, "Learning to generate realistic noisy images via pixel-level noise-aware adversarial training," *Proc. Advances Neural Inf. Process. Syst.*, vol. 34, pp. 3259–3270, 2021.
- [70] T. Pang, H. Zheng, Y. Quan, and H. Ji, "Recorrupted-to-recorrupted: unsupervised deep learning for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2043–2052.
- [71] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Restormer: Efficient transformer for high-resolution image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 5728–5739.
- [72] S. Anwar and N. Barnes, "Real image denoising with feature attention," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 3155–3164.
- [73] M. Chang, Q. Li, H. Feng, and Z. Xu, "Spatial-adaptive network for single image denoising," *arXiv preprint arXiv:2001.10291*, 2020.
- [74] J. Li, Z. Zhang, X. Liu, C. Feng, X. Wang, L. Lei, and W. Zuo, "Spatially adaptive self-supervised learning for real-world image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2023.
- [75] K. Zhang, Y. Li, J. Liang, J. Cao, Y. Zhang, H. Tang, R. Timofte, and L. Van Gool, "Practical blind denoising via swin-conv-unet and data synthesis," *arXiv preprint*, 2022.
- [76] Y. Quan, M. Chen, T. Pang, and H. Ji, "Self2self with dropout: Learning self-supervised denoising from single image," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 1890–1898.
- [77] S. Lefkimiatis, "Universal denoising networks: a novel cnn architecture for image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3204–3213.
- [78] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general u-shaped transformer for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, June 2022, pp. 17 683–17 693.
- [79] J. W. Soh and N. I. Cho, "Variational deep image restoration," *IEEE Trans. Image Process.*, vol. 31, pp. 4363–4376, 2022.
- [80] Z. Yue, H. Yong, Q. Zhao, D. Meng, and L. Zhang, "Variational denoising network: Toward blind noise modeling and removal," in *Proc. Advances Neural Inf. Process. Syst.*, 2019, pp. 1690–1701.
- [81] T. Rabie, "Robust estimation approach for blind denoising," *IEEE Trans. Image Process.*, vol. 14, no. 11, pp. 1755–1765, 2005.
- [82] M. Lebrun, M. Colom, and J.-M. Morel, "Multiscale image blind denoising," *IEEE Trans. Image Process.*, vol. 24, no. 10, pp. 3149–3161, 2015.
- [83] A. Majumdar, "Blind denoising autoencoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 312–317, 2018.
- [84] X. Zhang, Z. Xu, N. Jia, W. Yang, Q. Feng, W. Chen, and Y. Feng, "Denoising of 3d magnetic resonance images by using higher-order singular value decomposition," *Med. Image Anal.*, vol. 19, no. 1, pp. 75–86, 2015.
- [85] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction wiener filter," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 14, no. 4, pp. 1218–1234, 2006.

- [86] E. Luo, S. Pan, and T. Nguyen, "Generalized non-local means for iterative denoising," in *Proc. 20th Eur. Signal Process. Conf.*, 2012, pp. 260–264.
- [87] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.
- [88] T. Tirer and R. Giryen, "Image restoration by iterative denoising and backward projections," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1220–1234, 2018.
- [89] M. Zontak, I. Mosseri, and M. Irani, "Separating signal from noise using patch recurrence across scales," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1195–1202.
- [90] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2015.
- [91] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2020.
- [92] L. De Lathauwer, B. De Moor, and J. Vandewalle, "On the best rank-1 and rank-(r<sub>1</sub>, r<sub>2</sub>,..., r<sub>n</sub>) approximation of higher-order tensors," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1324–1342, 2000.
- [93] A. Cichocki, D. Mandic, L. De Lathauwer, G. Zhou, Q. Zhao, C. Caiafa, and H. A. Phan, "Tensor decompositions for signal processing applications: From two-way to multiway component analysis," *IEEE Sig. Process. Mag.*, vol. 32, no. 2, pp. 145–163, 2015.
- [94] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2466–2479, 2017.
- [95] A. Zhang and D. Xia, "Tensor svd: Statistical and computational limits," *IEEE Trans. Inf. Theory*, vol. 64, no. 11, pp. 7311–7338, 2018.
- [96] M. Diwakar and M. Kumar, "A review on ct image noise and its denoising," *Biomed. Signal Process. Control.*, vol. 42, pp. 73–88, 2018.
- [97] Y. Zhang, Y. Zhu, E. Nichols, Q. Wang, S. Zhang, C. Smith, and S. Howard, "A poisson-gaussian denoising dataset with real fluorescence microscopy images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11710–11718.
- [98] L. Kuruguntla, V. C. Doddha, and K. Elumalai, "Study of parameters in dictionary learning method for seismic denoising," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–13, 2021.
- [99] Y. Zhao, Y. Xu, Q. Yan, D. Yang, X. Wang, and L.-M. Po, "D2hnet: Joint denoising and deblurring with hierarchical network for robust night image restoration," in *Proc. Eur. Conf. Comput. Vis.*, pp. 91–110.
- [100] S. Guo, Z. Liang, and L. Zhang, "Joint denoising and demosaicking with green channel prior for real-world burst images," *IEEE Trans. Image Process.*, vol. 30, pp. 6930–6942, 2021.
- [101] D. Liu, B. Wen, J. Jiao, X. Liu, Z. Wang, and T. S. Huang, "Connecting image denoising and high-level vision tasks via deep learning," *IEEE Trans. Image Process.*, vol. 29, pp. 3695–3706, 2020.
- [102] S. Abramov, V. Lukin, O. Rubel, and K. Egiazarian, "Prediction of performance of 2d dct-based filter and adaptive selection of its parameters," *Electronic Imaging*, vol. 2020, no. 9, 2020.