Manifold Modeling in Embedded Space: An Interpretable Alternative to Deep Image Prior

Tatsuya Yokota, *Senior Member, IEEE*, Hidekata Hontani, *Member, IEEE*, Qibin Zhao, *Senior Member, IEEE*, and Andrzej Cichocki, *Fellow, IEEE*

***Abstract*— Deep image prior (DIP), which uses a deep convolutional network (ConvNet) structure as an image prior, has attracted wide attention in computer vision and machine learning. DIP empirically shows the effectiveness of the ConvNet structures for various image restoration applications. However, why the DIP works so well is still unknown. In addition, the reason why the convolution operation is useful in image reconstruction, or image enhancement is not very clear. This study tackles this ambiguity of ConvNet/DIP by proposing an interpretable approach that divides the convolution into "delay embedding" and "transformation" (i.e., encoder–decoder). Our approach is a simple, but essential, image/tensor modeling method that is closely related to self-similarity. The proposed method is called manifold modeling in embedded space (MMES) since it is implemented using a denoising autoencoder in combination with a multiway delay-embedding transform. In spite of its simplicity, MMES can obtain quite similar results to DIP on image/tensor completion, super-resolution, deconvolution, and denoising. In addition, MMES is proven to be competitive with DIP, as shown in our experiments. These results can also facilitate interpretation/characterization of DIP from the perspective of a "low-dimensional patch-manifold prior."**

***Index Terms*— Autoencoder (AE), convolutional neural network (CNN), deblurring, deconvolution, delay embedding, denoising AE (DAE), Hankelization, image inpainting, manifold model, super-resolution, tensor completion.**

I. Introduction

THE most important piece of information for image/tensor restoration would be the "prior." The prior usually converts the optimization problems from ill-posed to well-posed and/or enhances the robustness to specific noises and outliers. Many priors were studied in computer science problems, including low-rank models [26], [27], [49], [60], smoothness [21], [51], sparseness [59], nonnegativity [6], [35], and statistical independence [29]. In computer vision and machine learning, the Markov random fields (MRFs) [18], [37], the total variation (TV) [23], [66], low-rank representation [28], [30], [40], [54], [67], [69], [73], [82], [83], [85], and nonlocal similarity [4], [9] priors are often used for image modeling.

Recently, the deep image prior (DIP) [61], [62] has attracted attention in computer vision and machine learning. Ulyanov *et al.* [61], [62] have reported a very interesting phenomenon of the fully convolutional generator network [convolutional network (ConvNet)]. They claimed that the structure of a generator network is sufficient to capture a great deal of low-level image statistics prior to any learning. Fig. 1(a) shows a conceptual illustration of the method of DIP in an image inpainting task. The method of DIP optimizes untrained (i.e., randomly initialized) weight parameters of ConvNet for minimizing the squares' loss between its output and an observed image (e.g., a noisy/incomplete image) and to stop the optimization before overfitting (to noise). Then, the ConvNet, which learned from only a single corrupted image, reconstructs its restorative one. This implies that the structure of ConvNet itself plays a key role as a prior/regularizer in image restoration tasks.

Ulyanov *et al.* [61] explained the reason why a high-capacity ConvNet can be used as a prior with the following statement: the network resists "bad" solutions and descends much more quickly toward naturally looking images, and its phenomenon of "impedance of ConvNet" was confirmed by several toy experiments. However, it is not fully convinced by the above explanation because it does not say how the impedance is produced. One of the key questions is why must it be ConvNet? From a more practical perspective, it is important to consider what are "priors in DIP" while using simple and clear words (e.g., smoothness, sparseness, and low-rank).

Here, we discuss the problem of insufficient understanding of DIP. Suppose that we have data that are corrupted by noise, blur, or low resolution, and we want to recover them. Think about which model/prior to using from MRF, TV regularization, low-rank approximation, BM3D, and DIP. It is clear that it is best to choose the prior that best matches the given data, but it is impossible to choose without an understanding of each prior. At this time, it may be very difficult to select a method whose prior cannot be explained in words, such as DIP.

This study attempts to make the "DIP" more interpretable or explainable. For this purpose, the convolution operation was divided into "embedding" and "transformation" (see Fig. 2). Here, "embedding" stands for delay/shift-embedding (i.e., Hankelization), which is a

Manuscript received February 16, 2020; revised July 28, 2020; accepted November 5, 2020. Date of publication December 4, 2020; date of current version March 1, 2022. This work was supported in part by Japan Science and Technology Agency (JST) ACT-I under Grant JPMJPR18UU, in part by the Hori Sciences and Arts Foundation, in part by JSPS KAKENHI under Grant 20H04249 and Grant 20H04208, and in part by the MES Russian Federation under Grant 14.756.31.0001. *(Corresponding author: Tatsuya Yokota.)*

Tatsuya Yokota is with the Nagoya Institute of Technology, Nagoya 466-8555, Japan, and also with the RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan (e-mail: t.yokota@nitech.ac.jp).

Hidekata Hontani is with the Nagoya Institute of Technology, Nagoya 466-8555, Japan.

Qibin Zhao is with the RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan, and also with the School of Automation, Guangdong University of Technology, Guangzhou 510006, China.

Andrzej Cichocki is with the Skolkovo Institute of Science and Technology, 143026 Moscow, Russia, also with the RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan, and also with the School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou, China.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TNNLS.2020.3037923.

Digital Object Identifier 10.1109/TNNLS.2020.3037923

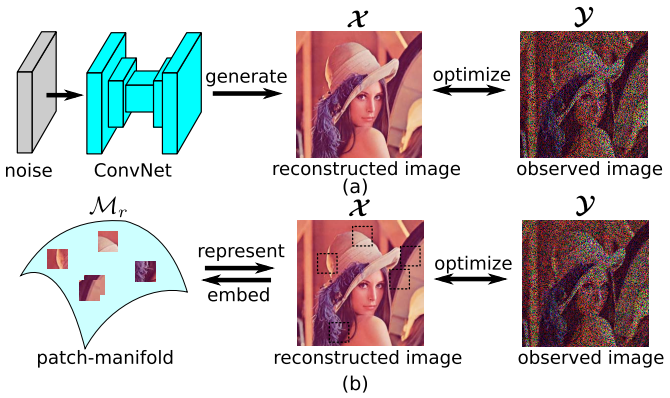


Fig. 1. Conceptual illustrations of DIP and the proposed manifold modeling in the embedded space. A case for the image inpainting task. In both models, we optimize some measure of similarity (distance or divergence) between the observed image and the reconstructed (generated) image for selective pixels. (a) DIP. (b) MMES (proposed).

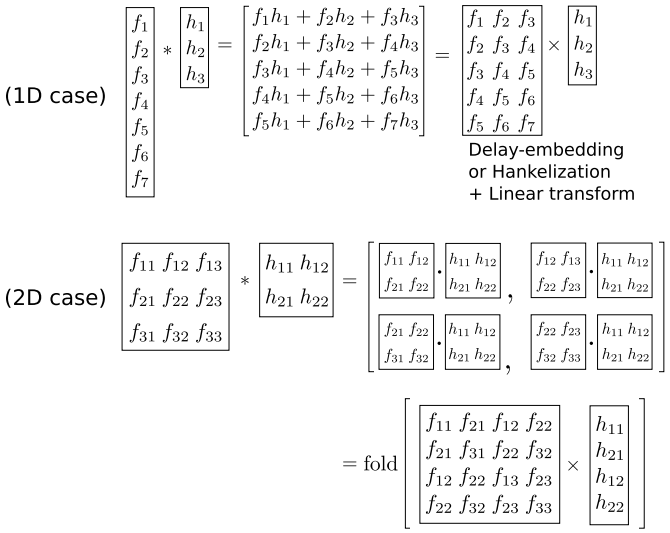


Fig. 2. Decomposition of the 1-D and 2-D convolutions: a valid convolution can be divided into a delay embedding/Hankelization and a linear transformation. The 1-D valid convolution of $\boldsymbol{f}$ with the kernel $\boldsymbol{h} = [h_1, h_2, h_3]$ can be represented by a matrix–vector product of the Hankel matrix and $\boldsymbol{h}$. In a similar way, a 2-D valid convolution can be represented by the matrix–vector product of the block Hankel matrix and an unfolded (vectorized) kernel.

copy/duplication operation of the image patches by the sliding window of the patch size (τ, τ). The embedding/Hankelization performs preprocessing to capture the delay/shift-invariant feature (e.g., nonlocal similarity) of the signals/images. The "transformation" is a simple linear transformation used in the convolution operation (see Fig. 2). Furthermore, we consider the extension of the linear transformation to nonlinear transformation similar to nonlinearity used in the ConvNets.

This study considers the following novel network structures: embedding $\mathcal{H}$ (linear), encoding ϕ_r (nonlinear), decoding ψ_r (nonlinear), and backward embedding $\mathcal{H}^\dagger$ (linear) (see Fig. 3). Note that its encoder–decoder part (ϕ_r, ψ_r) can be considered as a simple multilayer perceptron along with the filter domain (i.e., manifold learning), and it is sandwiched between forward and backward embedding $(\mathcal{H}, \mathcal{H}^\dagger)$. Hence, the proposed network can be characterized by manifold modeling in embedded space (MMES). The proposed MMES was designed to be as simple as possible while having an

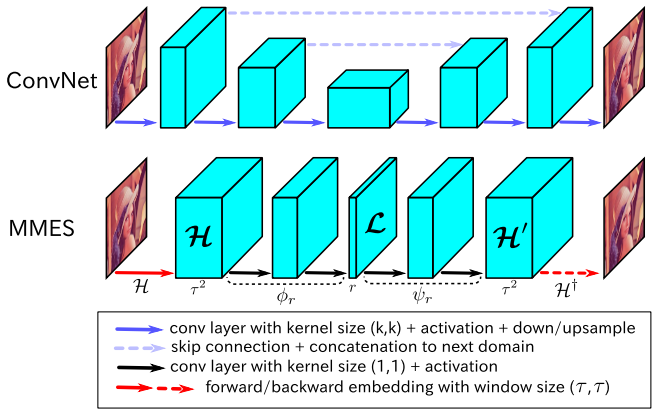


Fig. 3. Comparison of a typical AE ConvNet and the proposed MMES network.

essential ConvNet structure. The parameters τ and r in MMES correspond to the kernel size and the filter size in ConvNet.

Fig. 3 shows the network structures of ConvNet and the proposed MMES. When the horizontal dimension of the hidden tensor was set $\mathcal{L}$ with r, each τ^2-dimensional fiber in $\mathcal{H}$, which is a vectorization of each (τ, τ)-patch of an input image, is encoded into the r-dimensional space. Note that the volume of the hidden tensor $\mathcal{L}$ appears to be larger than the input/output image; however, the representation ability of $\mathcal{L}$ is much lower than that of the input/output image space. This is because the first/last tensor $(\mathcal{H}, \mathcal{H}')$ must have a Hankel structure (i.e., its degree of freedom, or the number of free parameters, is the same as that of the original image although the size of the Hankel tensor is larger than that of the original image), and the hidden tensor's dimensions $\mathcal{L}$ have been reduced from $\mathcal{H}$. Here, if we assume that $r \leq \tau^2$, then its low-dimensionality cannot be provided without the existence of similar (τ, τ)-patches (i.e., self-similarity) in the image. This would provide some "impedance" that passes self-similar patches while resisting others. Each fiber of the hidden tensor $\mathcal{L}$ represents a coordinate on the patch manifold of the image.

It should be noted that the MMES network is a special case for deep neural networks. In fact, the proposed MMES can be considered as a new kind of autoencoder (AE), in which the convolution operations have been replaced by Hankelization in preprocessing and postprocessing. Compared with ConvNet, the forward and backward embedding operations can be implemented by convolution and transposed convolution with one-hot-filters (see Fig. 6 for details). Note that the encoder–decoder part can be implemented by multiple convolution layers with the kernel size (1, 1) and nonlinear activations. The proposed model does not use the convolution explicitly; however, it performs linear transformations and a nonlinear activation in the "filter-domain" (i.e., the horizontal axis of the tensors in Fig. 3).

The contributions in this study can be summarized as follows: 1) a new and simple approach for image/tensor modeling is proposed, which extends and simplifies the ConvNet as a combination of delay embedding and multilayer perceptron; 2) the effectiveness of the proposed method and its similarity to the DIP are demonstrated by extensive experiments; and 3) most importantly, there is a prospect for interpreting/characterizing the DIP as a "low-dimensional patch-manifold prior."

II. Related Works

Note that the idea of a low-dimensional patch manifold has been proposed by Osher *et al.* [45] and Peyre [50]. Peyre [50] first formulated the patch manifold model for natural images and applied it to dictionary learning and performing a manifold pursuit. Osher *et al*. [45] formulated the regularization function to minimize the dimension of patch manifold and solved the Laplace–Beltrami equation by using the point integral method. In contrast to these studies, this study decreased the dimensions of the patch manifold by using AE, as shown in Fig. 3.

A related technique, low-rank tensor modeling in embedded space, was recently studied by [75]. However, the modeling approaches here are quite different since it considered the multilinear approach, while we consider the nonlinear (manifold) approach. Thus, our study can be interpreted as a manifold version of [75] from the perspective of tensor completion methods. Note that Yokota *et al.* [75] applied their model to only the tensor completion task. Moreover, in this study, we investigated not only the tensor completion but also super-resolution, deconvolution, and denoising tasks.

Another related work is devoted to group sparse representation (GSR) [80]. The GSR is roughly characterized as a combination of similar patch grouping and sparse modeling, which is similar to the combination of embedding and manifold modeling. However, the computational cost of similar patch grouping is higher than embedding, and this task is naturally included in manifold learning.

The main difference between the abovementioned studies and our investigation is the motivation: essential and simple image modeling can translate the ConvNet/DIP. The proposed MMES has many connections with ConvNet/DIP, such as embedding, nonlinear mapping, and training with noise.

From the perspective of DIP, there are several related works. First, the deep geometric prior [68] uses the properties of a multilayer perceptron for a shape reconstruction problem, which efficiently learns a smooth function from 2-D space to 3-D space. It helps us understand DIP from the perspective of manifold learning. For example, it can be used for gray-scale image reconstruction if an image is regarded as a point cloud in 3-D space (i, j, X_{ij}). However, this may not provide a good image reconstruction, such as DIP. The reason for this is that it smoothly interpolates a point cloud with a surface, such as a Voronoi interpolation. In addition, it cannot exploit a property of self-similarity for a natural image.

Second, a deep decoder [24] reconstructs natural images from noises by non-ConvNets. These non-ConvNets have linear channel/color transformations, ReLU, channel/color normalization, and upsampling layers. In contrast, the DIP employs an overparameterized network, while the deep decoder uses an underparameterized network, and it shows its ability in image reconstruction. Although the deep decoder is a non-ConvNet, Hackel and Hand [24] have emphasized the closed relationship between the convolutional layers in DIP and the upsampling layers in the deep decoder. As a result, Hackel and Hand [24] have claimed: "if there is no upsampling layer, then there is no notion of locality in the resultant image," for the deep decoder. This implies that the "locality" is the essence of the image reconstruction model, and the convolution/upsampling layer provides it. Furthermore, the deep decoder has a close relationship with the proposed MMES. Note that the MMES essentially has a decoder and an inverse multiway-delay embedding transform (MDT) [see (6)], and the encoder is used for satisfying the Hankel structure. The decoder and the inverse MDT in the proposed MMES correspond to the linear operation and the upsampling layer in the deep decoder, respectively. Note that the concept of underparameterization is also similar to the MMES.

Convolutional sparse coding (CSC) [47], [48] also has close relations with the proposed MMES. The CSC model represents the image by multiplication of a convolutional dictionary matrix with a sparse vector. Papyan *et al.* [47] analyzed convolutional neural networks (CNNs) using the CSC model. In fact, the CSC model can be reformulated by using backward embedding of patches generated by sparse coding (see the Appendix), i.e., each patch is a linear combination of few atoms of the redundant local dictionary. Its sparse coding corresponds to our AE in terms of manifold learning. The main difference between CSC and MMES is that we used the Hankel constraint to generate patches. In our MMES, overlapped patches generated by AE are dependent on each other by the Hankel constraint; however, the overlapped patches generated by sparse coding in the CSC model are independent. In other words, the main difference is in the principles of generative patches. The patches in MMES directly represent local patches in images, while the patches in CSC are "bases" of patches in images.

From this, the essence of the image model is the "locality," and its locality is provided by "convolution," "upsampling," or "delay embedding." This is why image restoration from a single image with deep ConvNets has received wide attention, which is otherwise known as zero-shot learning, internal learning, or self-supervised learning [2], [5], [33], [34], [36], [56], [70].

Furthermore, recently, two generative models, SinGAN [53] and InGAN [55], have been proposed, and they were trained on a single image. The key concept of both works is to impose the constraint for local patches of the image to be natural. From the perspective of the constraint for local patches of the image, the MMES has a closed relationship with these works. However, our study explicitly imposes a low-dimensional manifold constraint for the local patches rather than adversarial training with patch discriminators.

In addition, supervised learning using ConvNet plays an important role in image restoration tasks [12], [13], [31], [36], [81]. The basic idea of the supervised image restoration method is to learn a nonlinear map from the corrupted image $\boldsymbol{x}$ to its original image $\boldsymbol{y}$ by using a large number of pairs $\{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N_{\text{train}}}$. The dimensionality of image input/output (e.g., 128^2 and 256^2) is very high; however, it has been reported that the learning image restoration ConvNet can work well with a relatively small number of training samples (e.g., 91 in SRCNN [12], 500 in FBPConvNet [31], and 400 in DnCNN [81]). After all, surprisingly, it has been demonstrated that ConvNet works well with even zero training data in the study of DIP [61], [62]. This makes us focus on the structure of ConvNet in image restoration tasks.

Regarding embedding, it is often studied that a large number of data can be embedded into latent space for representation learning, such as in [39] and [44]. The motivation of delay

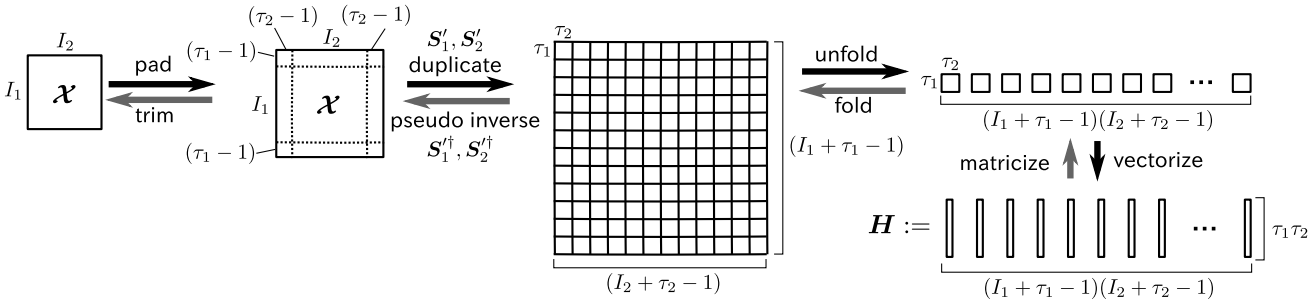


Fig. 4. Flow of the multiway-delay-embedding operation ($N = 2$).

embedding in this study is similar to the above, but we consider a case for which we do not have a large amount of data but only a single image with noise, missing pixels, or low resolution. Thus, we embed local patches of a single image rather than a large set of training samples of images. In our case of delay embedding, overlapped local patches are strongly dependent on each other, while training samples are independent in general embedding.

III. Manifold Modeling in Embedded Space

In this section, we explain, in detail, the proposed method based on the concept of MMES. In addition, this section systematically derives the MMES structure from it. Conceptually, the proposed tensor reconstruction method can be formulated by

$$\begin{aligned} &\min_{\boldsymbol{\mathcal{X}}} \ \|\boldsymbol{\mathcal{Y}} - \mathcal{F}(\boldsymbol{\mathcal{X}})\|_F^2 \\ &\text{s.t. } \mathcal{H}(\boldsymbol{\mathcal{X}}) = [\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T] =: \boldsymbol{H} \\ &\quad \boldsymbol{h}_t \in \mathcal{M}_r \quad \text{for } t = 1, 2, \ldots, T \end{aligned} \tag{1}$$

where $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is an observed corrupted tensor, $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is an estimated tensor, $\mathcal{F} : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}^{J_1 \times J_2 \times \cdots \times J_N}$ is a linear operator, which represents the observation system, $\mathcal{H} : \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N} \to \mathbb{R}^{D \times T}$ is padding, and a Hankelization operator with a sliding window of size $(\tau_1, \tau_2, \ldots, \tau_N)$. In addition, each vector $\boldsymbol{h}_t$ is imposed, so it can be a point in a r-dimensional manifold $\mathcal{M}_r$ that is embedded in the D-dimensional Euclidean space, which results in $r \leq D$. For simplicity, $D := \prod_n \tau_n$ and $T := \prod_n (I_n + \tau_n - 1)$. A linear operator $\mathcal{F}$ can be different for different tasks. For the tensor completion task, $\mathcal{F} = P_\Omega$ is a projection operator for the support set Ω so that the missing elements are set to be zero. For the super-resolution task, $\mathcal{F}$ is a downsampling operator for the images/tensors. For the deconvolution task, $\mathcal{F}$ is a convolution operator with some blur kernels. For the denoising task, $\mathcal{F}$ is an identity map. Fig. 1(b) shows the concept of the proposed manifold modeling for image inpainting/completion (i.e., $N = 2$). The distance was minimized between the observation $\boldsymbol{\mathcal{Y}}$ and the reconstruction $\boldsymbol{\mathcal{X}}$ with its support Ω. In addition, all patches in $\boldsymbol{\mathcal{X}}$ should be included in some restricted manifold $\mathcal{M}_r$. In other words, $\boldsymbol{\mathcal{X}}$ is represented by the patch manifold, and the property of the patch manifold can be image priors. For example, the low dimensionality of the patch manifold restricts the nonlocal similarity of the images/tensors, and it would be related to the "impedance" in DIP. In addition, $\boldsymbol{\mathcal{X}}$ was modeled indirectly by designing the properties of the patch manifold $\mathcal{M}_r$.

A. Multiway-Delay Embedding for Tensors

MDT is a multiway generalization of Hankelization that was proposed by [75].

In [75], MDT is defined using the multilinear tensor product with multiple duplication matrices and tensor reshaping. The same operation was used; however, a padding operation was added. Thus, multiway-delay embedding used in this study is defined by

$$\mathcal{H}(\boldsymbol{\mathcal{X}}) := \text{unfold}_{(D,T)}(\text{pad}_{\boldsymbol{\tau}}(\boldsymbol{\mathcal{X}}) \times_1 \boldsymbol{S}_1 \ldots \times_N \boldsymbol{S}_N) \tag{2}$$

where $\text{pad}_{\boldsymbol{\tau}} : \mathbb{R}^{I_1 \times \cdots \times I_N} \to \mathbb{R}^{(I_1 + 2(\tau_1 - 1)) \times \cdots \times (I_N + 2(\tau_N - 1))}$ is a N-dimensional reflection padding operator of tensors, $\boldsymbol{S}_n \in \mathbb{R}^{\tau_n (I_n + \tau_n - 1) \times (I_n + 2(\tau_n - 1))}$ is a duplication matrix (see Figs. 4 and 5), and $\text{unfold}_{(D,T)} : \mathbb{R}^{\tau_1 (I_1 + \tau_1 - 1) \times \cdots \times \tau_N (I_N + \tau_N - 1)} \to \mathbb{R}^{D \times T}$ is an unfolding operator, which outputs a matrix from an input N-th order tensor.

For example, the proposed Hankelization with reflection padding of $\boldsymbol{x} = [x_1, x_2, \ldots, x_7]^T$ with $\tau = 3$ is given by

$$\begin{aligned} &[x_1, x_2, x_3, x_4, x_5, x_6, x_7]^T \\ &\xrightarrow{\text{pad}_3} [x_3, x_2, x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_6, x_5]^T \\ &\xrightarrow{\text{Hankelize}} \begin{pmatrix} x_3 & x_2 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 \\ x_2 & x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_6 \\ x_1 & x_2 & x_3 & x_4 & x_5 & x_6 & x_7 & x_6 & x_5 \end{pmatrix}. \end{aligned} \tag{3}$$

Fig. 4 shows an example of the proposed multiway-delay embedding in the case of second-order tensors. The overlapped patch grid is constructed by the multilinear tensor product with $\boldsymbol{S}_n$. Finally, all the patches were split, lined up, and vectorized.

The Moore–Penrose pseudoinverse of $\mathcal{H}$ is given by

$$\mathcal{H}^\dagger(\boldsymbol{H}) = \text{trim}_{\boldsymbol{\tau}}\big(\text{fold}_{(D,T)}(\boldsymbol{H}) \times_1 \boldsymbol{S}_1^\dagger \cdots \times_N \boldsymbol{S}_N^\dagger\big) \tag{4}$$

where $\boldsymbol{S}_n^\dagger := (\boldsymbol{S}_n^T \boldsymbol{S}_n)^{-1} \boldsymbol{S}_n^T$ is the pseudoinverse of $\boldsymbol{S}_n$ $\text{fold}_{(D,T)} := \text{unfold}_{(D,T)}^{-1}$ and $\text{trim}_{\boldsymbol{\tau}} = \text{pad}_{\boldsymbol{\tau}}^\dagger$ is a trimming operator for removing the $(\tau_n - 1)$ elements at the start and the end for each mode. Note that $\mathcal{H}^\dagger \circ \mathcal{H}$ is an identity map; however, $\mathcal{H} \circ \mathcal{H}^\dagger$ is not since it is a projection.

1) Delay Embedding Using Convolution: Delay embedding and its pseudoinverse can be implemented using a convolution, in which the kernels are one-hot-tensor windows of size $(\tau_1, \tau_2, \ldots, \tau_N)$. The one-hot-tensor windows can be provided by folding a D-dimensional identity matrix $\boldsymbol{I}_D \in \mathbb{R}^{D \times D}$ into $\boldsymbol{\mathcal{I}}_D \in \mathbb{R}^{\tau_1 \times \cdots \times \tau_N \times D}$. Fig. 6 demonstrates the calculation flow for multiway delay embedding, which uses the convolution for the case where $N = 2$. The multilinear tensor product was replaced with a convolution using the one-hot-tensor windows.

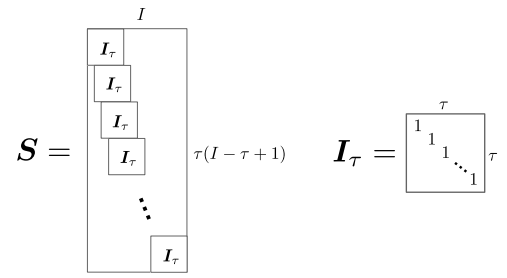


Fig. 5. Duplication matrix $\boldsymbol{S} \in \mathbb{R}^{\tau(I-\tau+1)\times I}$. In the case where there are I columns, it consists of the $(I-\tau+1)$ identity matrices with the size (τ, τ).

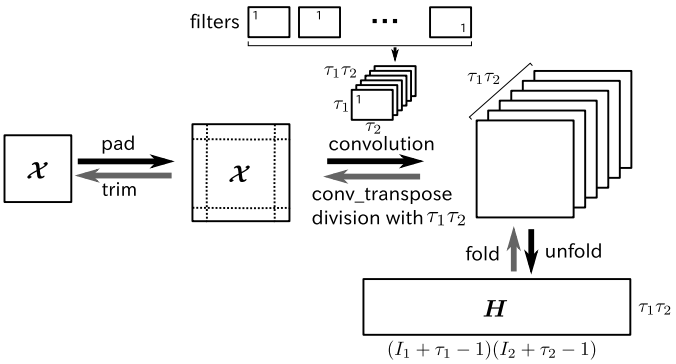


Fig. 6. Multiway-delay embedding using 2-D-convolution ($N = 2$). Functionality is the same as the flow shown in Fig. 4; only the implementation is different.

The pseudoinverse convolution with padding was provided with its adjoint operation. This is called the "transposed convolution" in some neural network libraries, and there is trimming and simple scaling with D^{-1}. Note that this implementation can perform the equivalent function of MDT and its inverse defined in (2) and (4), respectively. It allows us to implement the proposed method easily by using neural network libraries.

B. Definition of Low-Dimensional Manifold

This study considered AE when defining the r-dimensional manifold $\mathcal{M}_r$ in the $(\prod_n \tau_n)$-dimensional Euclidean space as follows:

$$\mathcal{M}_r := \{\hat{\psi}_r(\boldsymbol{l}) \mid \boldsymbol{l} \in \mathbb{R}^r\}$$
$$(\hat{\psi}_r, \hat{\phi}_r) := \underset{(\psi_r,\phi_r)}{\operatorname{argmin}} \sum_{t=1}^{T} \|\boldsymbol{h}_t - \psi_r\phi_r(\boldsymbol{h}_t)\|_2^2 \tag{5}$$

where $\phi_r : \mathbb{R}^D \to \mathbb{R}^r$ is an encoder, $\psi_r : \mathbb{R}^r \to \mathbb{R}^D$ is a decoder, and $\hat{\psi}_r\hat{\phi}_r : \mathbb{R}^D \to \mathbb{R}^D$ is an AE that was constructed from $\{\boldsymbol{h}_t\}_{t=1}^T$. Note that, in general, the use of AE models is a widely accepted approach for manifold learning [25]. The properties of the manifold $\mathcal{M}_r$ are determined by the properties of ϕ_r and ψ_r. By employing multilayer perceptrons (neural networks) for ϕ_r and ψ_r, the encoder–decoder may provide a smooth manifold.

C. Problem Formulation

In this section, the conceptual formulation (1) and the AE guided manifold constraint were combined to derive a practical optimization problem. First, a tensor $\boldsymbol{\mathcal{X}}$ was redefined as an output of the generator

$$\begin{aligned}\boldsymbol{\mathcal{X}} &:= \mathcal{H}^\dagger[\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_T], \quad \text{where } \boldsymbol{h}_t \in \mathcal{M}_r \\ &= \mathcal{H}^\dagger[\hat{\psi}_r(\boldsymbol{l}_1), \hat{\psi}_r(\boldsymbol{l}_2), \ldots, \hat{\psi}_r(\boldsymbol{l}_T)]\end{aligned} \tag{6}$$

where $\boldsymbol{l}_t \in \mathbb{R}^r$ and $\mathcal{H}^\dagger$ is the pseudoinverse of $\mathcal{H}$. At this moment, $\boldsymbol{\mathcal{X}}$ is a function of $\{\boldsymbol{l}_t\}_{t=1}^T$; however, the Hankel structure of the matrix $\boldsymbol{H}$ cannot always be guaranteed under the unconstrained condition of $\boldsymbol{l}_t$. By guaranteeing the Hankel structure of the matrix $\boldsymbol{H}$, it was further transformed as follows:

$$\begin{aligned}\boldsymbol{\mathcal{X}} &:= \mathcal{H}^\dagger[\hat{\psi}_r\hat{\phi}_r(\boldsymbol{g}_1), \hat{\psi}_r\hat{\phi}_r(\boldsymbol{g}_2), \ldots, \hat{\psi}_r\hat{\phi}_r(\boldsymbol{g}_T)], \\ &= \mathcal{H}^\dagger\mathcal{A}_r[\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_T] \\ &= \mathcal{H}^\dagger\mathcal{A}_r\mathcal{H}(\boldsymbol{\mathcal{Z}})\end{aligned} \tag{7}$$

where $\mathcal{A}_r : \mathbb{R}^{D\times T} \to \mathbb{R}^{D\times T}$ was inserted as an operator, which autoencodes each column of the input matrix with $(\hat{\psi}_r, \hat{\phi}_r)$ and $[\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_T]$ as a matrix. This has a Hankel structure, and it was transformed by the Hankelization with some input tensors $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$. Note that the tensor $\boldsymbol{\mathcal{Z}}$ is the most compact representation for the Hankel matrix $[\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_T]$. Since $\mathcal{A}_r$ is an AE, which enforces the outputs to be as close as possible similar to the inputs by using low-dimensional representations, $\mathcal{A}_r\mathcal{H}(\boldsymbol{\mathcal{Z}})$ is estimated to be a Hankel matrix.

Therefore, the constraint of $\boldsymbol{\mathcal{X}}$ is in the original optimization problem (1), and it is replaced with (7)

$$\begin{aligned}&\min_{\boldsymbol{\mathcal{Z}}} \ \|\boldsymbol{\mathcal{Y}} - \mathcal{F}(\boldsymbol{\mathcal{X}})\|_F^2, \\ &\text{s.t. } \boldsymbol{\mathcal{X}} = \mathcal{H}^\dagger\mathcal{A}_r\mathcal{H}(\boldsymbol{\mathcal{Z}}).\end{aligned} \tag{8}$$

Note that the optimization parameter $\boldsymbol{\mathcal{X}}$ in (1) is replaced with $\boldsymbol{\mathcal{Z}}$ in (8). This is because $\boldsymbol{\mathcal{X}}$ is a function of $\boldsymbol{\mathcal{Z}}$. By substituting the constraint into the objective function, the problem (8) is transformed as $\text{minimize}_{\boldsymbol{\mathcal{Z}}} \|\boldsymbol{\mathcal{Y}} - \mathcal{F}(\mathcal{H}^\dagger\mathcal{A}_r\mathcal{H}(\boldsymbol{\mathcal{Z}}))\|_F^2$, where $\mathcal{A}_r$ is an AE, which defines the manifold $\mathcal{M}_r$. Note that the problems (1) and (8) are slightly different because the original constraint imposes $\boldsymbol{H}$ to be strictly Hankel in contrast to the modified constraint that imposes $\mathcal{A}_r\mathcal{H}(\boldsymbol{\mathcal{Z}})$ to be a Hankel. This difference is due to the difficulty of the constraints of the "Hankel structure" and a "low-dimensional representation" for the matrix $\boldsymbol{H}$.

In this study, the AE/manifold is learned from the observed tensor $\boldsymbol{\mathcal{Y}}$; thus, the optimization problem is formulated as

$$\begin{aligned}\min_{\boldsymbol{\mathcal{Z}},\mathcal{A}_r} \ &\underbrace{\|\boldsymbol{\mathcal{Y}} - \mathcal{F}(\mathcal{H}^\dagger\mathcal{A}_r\mathcal{H}(\boldsymbol{\mathcal{Z}}))\|_F^2}_{=:\mathcal{L}_{\text{rec}}} \\ &+ \lambda \underbrace{\|\mathcal{H}(\boldsymbol{\mathcal{Z}}) - \mathcal{A}_r\mathcal{H}(\boldsymbol{\mathcal{Z}})\|_F^2}_{=:\mathcal{L}_{\text{AE}}}\end{aligned} \tag{9}$$

where the first and second terms are referred by a reconstruction loss and an autoencoding loss, and $\lambda > 0$ is a tradeoff parameter for balancing both losses.

Finally, it should be noted that (7) describes the MMES network shown in Fig. 3. This includes $\mathcal{H}$, $\hat{\phi}_r$, $\hat{\psi}_r$, and $\mathcal{H}^\dagger$, which, respectively, correspond to forward embedding, encoding, decoding, and backward embedding. The encoder and decoder can be defined by the multilayer perceptrons (i.e., repetition of the linear transformation and nonlinear activation).

D. Design of the Autoencoder

This section discusses how to design the neural network architecture of the AE for restricting the manifold $\mathcal{M}_r$. The

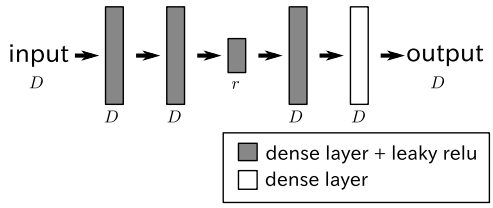


Fig. 7. Example of the architecture of the AE.

simplest way is controlling the value of r, and it directly restricts the dimensionality of the latent space. There are many other possibilities: Tikhonov regularization [20], dropout [16], denoising AE (DAE) [41], [65], variational AE [10], adversarial AE [7], [42], and alpha-GAN [52], among others. All methods have some perspective and promise; however, the cost is not low. This study selected an attractive and fundamental method: the "DAE" [65]. The DAE is attractive because it has a strong relationship with the Tikhonov regularization [3], and it decreases the entropy of the data [57]. Furthermore, learning with noise was also employed for the DIP.

Finally, an AE was designed that controls the dimension r and the standard deviation σ of the additive zero-mean Gaussian noise. Fig. 7 provides an illustration of the examples of architecture for the AE that was used in this study. The sizes of the hidden variables affect the representation ability for image reconstruction.

E. Optimization

The optimization problem (9) consists of two terms: a reconstruction loss and an autoencoding loss. The hyperparameter λ was set to balance both losses. In essence, λ should be large because the autoencoding loss should be zero. However, a very large λ prohibits minimization of the reconstruction loss, which may lead to local optima. Therefore, the value of λ was gradually adjusted in the optimization process.

Algorithm 1 shows an optimization algorithm for the tensor reconstruction and/or enhancement. A strategy of DAE was employed for AE learning. Adaptation of λ is an example, and it can be modified appropriately with data and tasks. Here, the tradeoff parameter λ is adjusted for keeping $\mathcal{L}_{\text{rec}} > \mathcal{L}_{\text{AE}}$; however, there is no large gap between both losses. By exploiting the convolutional structure of $\mathcal{H}$ and $\mathcal{H}^\dagger$, the calculation flow of $\mathcal{L}_{\text{rec}}$ and $\mathcal{L}_{\text{AE}}$ can easily be implemented using neural network libraries, such as `TensorFlow`. This study employed the `Adam` [32] optimizer for updating $(\boldsymbol{\mathcal{Z}}, \mathcal{A}_r)$.

F. Computational Complexity

Here, we discuss the computational complexity of the proposed MMES and ConvNet used in DIP. Let us assume that the total number of pixels is T and delay-embedding dimension is D, and the computational complexity of delay embedding $\mathcal{H}$ or its inverse $\mathcal{H}^\dagger$ is $\mathcal{O}(DT)$. Then, the computational complexity of the AE is $\mathcal{O}(D^2T)$. On the other hand, the computational complexity of ConvNet is $\mathcal{O}(kc^2\ T)$, where k is the size of the convolutional kernel and c is the number of channels/filters for input/output feature maps. For example, when we consider (8, 8)-patches, the embedded dimension is $D = 64$. Since DIP uses the (3, 3)-kernel and the number of channels/filters is 128, e.g., $k = 9$ and $c = 128$, in the abovementioned case, we have $D^2\ T < kc^2T$. Thus, the computational complexity of MMES is slightly smaller than that of the original DIP in most cases.

Algorithm 1 Optimization Algorithm for Tensor Reconstruction

input: $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{J_1 \times \cdots \times J_N}$ (corrupted tensor), $\mathcal{F}$, $\boldsymbol{\tau}$, r, σ;
initialize: $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, auto-encoder $\mathcal{A}_r$, $\lambda = 5.0$;
repeat
  $\boldsymbol{H} \leftarrow \mathcal{H}(\boldsymbol{\mathcal{Z}}) \in \mathbb{R}^{D \times T}$ with $\boldsymbol{\tau}$;
  generate noise $\boldsymbol{E} \in \mathbb{R}^{D \times T}$ with σ;
  $\mathcal{L}_{\text{AE}} \leftarrow \|\boldsymbol{H} - \mathcal{A}_r(\boldsymbol{H} + \boldsymbol{E})\|_F^2$;
  $\mathcal{L}_{\text{rec}} \leftarrow \frac{1}{D}\|\boldsymbol{\mathcal{Y}} - \mathcal{F}(\mathcal{H}^\dagger \mathcal{A}_r(\boldsymbol{H} + \boldsymbol{E}))\|_F^2$;
  update $(\boldsymbol{\mathcal{Z}}, \mathcal{A}_r)$ by `Adam` for $\mathcal{L}_{\text{rec}} + \lambda\mathcal{L}_{\text{AE}}$;
  if $\mathcal{L}_{\text{rec}} < \mathcal{L}_{\text{AE}}$ **then** $\lambda \leftarrow 1.1\lambda$; **else** $\lambda \leftarrow 0.99\lambda$;
until converge
output: $\widehat{\boldsymbol{\mathcal{X}}} = \mathcal{H}^\dagger \mathcal{A}_r \mathcal{H}(\boldsymbol{\mathcal{Z}}) \in \mathbb{R}^{I_1 \times \cdots \times I_N}$ (reconstructed tensor);

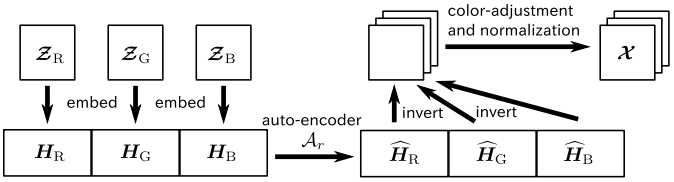


Fig. 8. Generator network in a case of RGB color-image recovery.

G. Special Setting for Color-Image Recovery

In the case of the multichannel or color-image recovery, a special setting for the generator network was used. This is because the spatial pattern of individual channels is similar, and the patch manifold can be shared. Fig. 8 shows an illustration of the AE shared version of MMES in the case of color-image recovery. In this case, three channels were inputted, and each channel input was embedded independently. Then, three block Hankel matrices were concatenated and autoencoded simultaneously. The inverted three images were stacked as a color-image (third-order tensor). Finally, the multichannel image is color-transformed. The last color-transform can be implemented by a convolution layer with a kernel size (1, 1), and it is also optimized as the parameters. It should be noted that the three input channels are not necessary for corresponding to RGB; however, it would be optimized with compact color-representation.

IV. Experiments

This section shows the extensive experimental results that demonstrate the similarity and some slight differences between the DIP and MMES. First, toy examples with a time-series signal and a gray-scale image were recovered by the proposed method to show its basic behaviors. Second, hyperparameter sensitivity was demonstrated to get a sense for adjusting the parameters and show the effects of DAE. Third, the phenomenon of noise impedance in MMES was demonstrated in comparison to the DIP. Finally, the results are presented in comparison to the DIP and other selective methods for color-image inpainting, super-resolution, deconvolution, and denoising tasks.

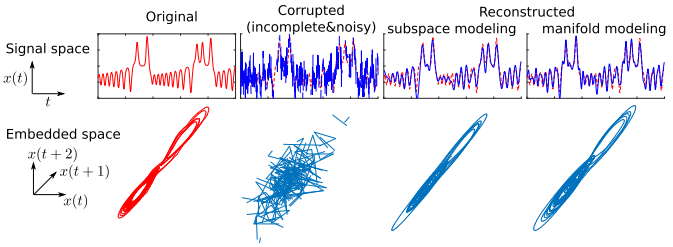


Fig. 9. Time-series signal recovery of the subspace and manifold models in the embedded space.

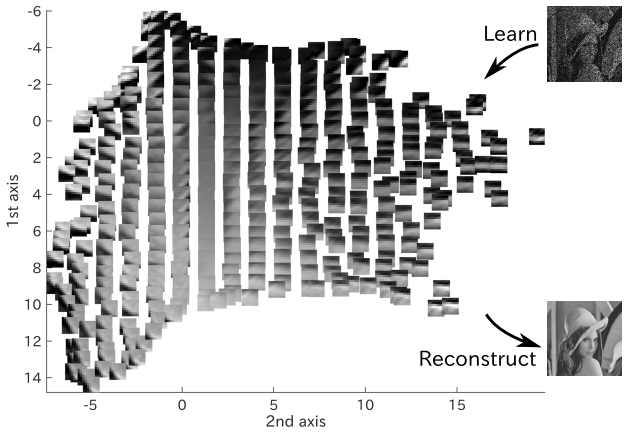


Fig. 10. 2-D (8, 8)-patch manifold that was learned from a gray-scale image of "Lena" with 50% of missing pixels.

A. Toy Examples

In this section, the proposed method was applied to a toy example of signal recovery. Fig. 9 shows the results of this experiment. A 1-D time-series signal was generated from the Lorentz system. This signal was corrupted by additive Gaussian noise, random missing, and three block occlusions. The corrupted signal was recovered by the subspace modeling [75] and the proposed manifold modeling in the embedded space. The window size of the delay embedding was $\tau = 64$, the lowest dimension of the AE was $r = 3$, and the additive noise standard deviation was set to $\sigma = 0.05$. The structure of the Lorentz attractor was caught more effectively through manifold modeling than the subspace modeling.

Fig. 10 visualizes a 2-D $(8, 8)$-patch manifold that was learned by the proposed method from a 50% missing gray-scale image of "Lena." For this figure, this study set the parameters as follows: $\boldsymbol{\tau} = [8, 8]$, $r = 2$, and $\sigma = 0.05$. Similar patches were located near each other, and a smooth change of the patterns was observed. This implies that the relationship between the nonlocal similarity-based methods [4], [9], [22], [80] and the manifold modeling (i.e., DAE) plays a key role for "patch grouping" for the proposed method. The difference from the nonlocal similarity-based approach is that manifold modeling is "global" rather than "nonlocal." In other words, the MMES finds similar patches for the target patch from the whole image area rather than its neighborhood area.

1) Optimization Behavior: This experiment recovered 50% of the missing gray-scale image of "Lena." The optimization algorithm was stopped after 20 000 iterations. The learning rate was set as 0.01, and the learning rate was decayed by 0.98 for every 100 iterations. λ was adapted by the algorithm 1 for every ten iterations. Fig. 11 shows the optimization behavior of the reconstructed image, the reconstruction loss $\mathcal{L}_{\text{rec}}$, the autoencoding loss $\mathcal{L}_{\text{DAE}}$, and the tradeoff coefficient λ. By using the tradeoff adjustment, the reconstruction loss and the autoencoding loss were intersected around 1500 iterations. In addition, both losses jointly decreased after the intersection point.

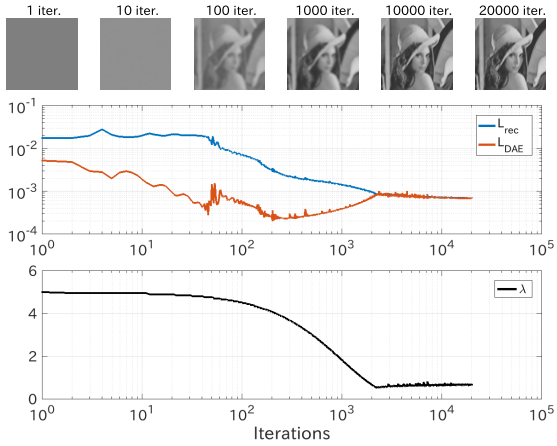


Fig. 11. Optimization behavior of MMES during reconstruction of "Lena" with 50% missing pixels.

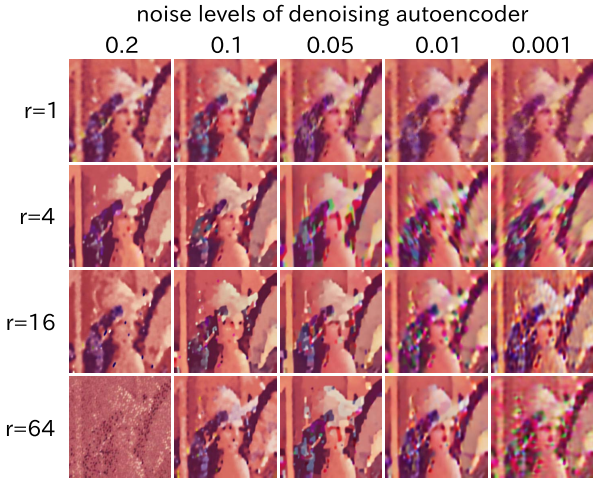


Fig. 12. Reconstructing the color image of "Lena" with 99% pixels missing for the various values of r and noise level.

B. Hyperparameter Sensitivity

The sensitivity of the MMES was evaluated with three hyperparameters: r, σ, and $\boldsymbol{\tau}$. First, the patch size was fixed as $(8, 8)$, and the dimension r and the noise standard deviation σ varied. Fig. 12 shows the reconstruction results of a 99% missing image of "Lena" by the proposed method with different settings for (r, σ). The proposed method with very low dimensions $(r = 1)$ provided blurred results, whereas the proposed method with very high dimensions $(r = 64)$ provided results that had many peaks. Furthermore, an appropriate noise level $(\sigma = 0.05)$ can provide sharp and clean results. As a reference, Fig. 13 shows the difference of the DIP optimized with and without noise. From these results, the effects of learning with noise can be confirmed.

Next, the noise level was fixed as $\sigma = 0.05$, and the patch size varied with some values of r. Fig. 14 shows the results with various patch-size settings for recovering a 99% missing

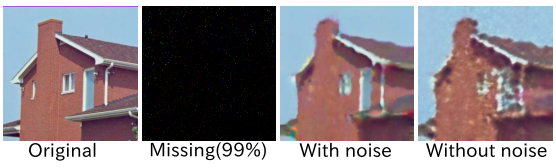


Fig. 13. Reconstruction of the 'home' image by training with/without noise in the DIP with 99% missing pixels.

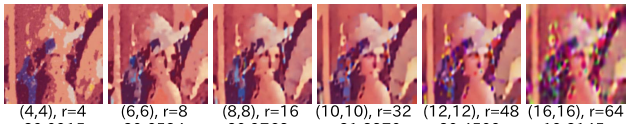


Fig. 14. Reconstruction of the "Lena" image for various patch sizes $\boldsymbol{\tau}$.

image. The patch sizes $\boldsymbol{\tau}$ of (8, 8) or (10, 10) were appropriate for this case. The patch size is very important because it depends on the variety of patch patterns. If the patch size is too large, then the patch variations might expand, and the structure of the patch manifold is complicated. In contrast, if the patch size is too small, then the information obtained from the embedded matrix $\boldsymbol{H}$ is limited, and reconstruction becomes difficult in the highly missing cases. The same problem might occur for all the patch-based image reconstruction methods [4], [9], [22], [80]. However, good patch sizes would be unique for different images and types/levels of corruption. In addition, the estimation of a good patch size is an open problem. A multiscale approach [74] may partially help with this issue; however, the patch size is still fixed or tuned as a hyperparameter.

C. Noise Impedance of MMES

This section reproduces the demonstration of the noise impedance in [61] with the proposed MMES. Four target color images, a natural image, a natural image with noise $\sim\mathcal{N}(0, 20^2)$, a pixel-shuffled image, and an image of uniform noise, were prepared and are displayed in Fig. 15(a). They were reconstructed using DIP and MMES. In both methods, the mean square error between the target and the reconstructed images was recorded for each iteration. The learning rate was set as 0.01, and the learning rate was decayed by 0.98 for every 100 iterations. In MMES, the tradeoff parameter λ was dynamically adjusted so that the autoencoding loss $\mathcal{L}_{\mathrm{DAE}}$ does not exceed some small value.

Fig. 15(b) and (c) shows the optimization behavior of DIP and MMES when reconstructing the four target images. First, a natural image was reconstructed rapidly for both DIP and MMES. The optimization of the noisy image for DIP and MMES was slower than the noise-free image, and both curves were similar. The curves of the shuffled and uniform images of MMES were also similar to DIP. This implies that the MMES has noise impedance.

D. Comparisons

This section displays the experimental results of the performance comparison for four tasks: tensor completion, super-resolution, deconvolution, and denoising.

1) Color-Image Completion, for an Extremely High Ratio of Missing Pixels: This section compares the performance of the proposed method with several selected unsupervised tensor completion/image inpainting methods. This includes the low-rank tensor completion (HaLRTC) [40], parallel low-rank matrix factorization (TMac) [72], tubal nuclear norm regularization (tSVD) [84], Tucker decomposition with a rank increment (Tucker inc.) [75], low-rank and total-variation (LRTV) regularization[1] [76], [77], smooth PARAFAC tensor completion (SPC)[2] [79], GSR[3] [80], multiway delay embedding based on Tucker modeling (MDT-Tucker)[4] [75], and DIP[5] [61].

TABLE I

PARAMETER SETTINGS FOR THE MMES IN THE IMAGE COMPLETION EXPERIMENTS

(τ, r)	airplane	baboon	barbara	facade	house	lena	peppers	saiboat
50 %	(16,4)	(10,4)	(6,4)	(10,4)	(16,4)	(6,4)	(6,4)	(6,4)
70 %	(16,4)	(10,4)	(6,4)	(16,4)	(16,4)	(6,4)	(16,4)	(6,4)
90 %	(16,4)	(4,8)	(6,4)	(16,4)	(16,4)	(8,4)	(16,4)	(4,4)
95 %	(16,4)	(4,6)	(6,4)	(16,4)	(16,4)	(6,8)	(16,4)	(6,8)
99 %	(8,32)	(4,4)	(6,4)	(4,1)	(8,16)	(10,32)	(8,8)	(6,4)

For these experiments, the hyperparameters of all the methods were manually tuned to obtain the best peak-signal-to-noise ratio (PSNR) and for the structural similarity (SSIM). For DIP, it is impossible to investigate all the network structures since there were various kernel sizes, filter sizes, and depths. Instead, a "default architecture" was employed [61], and the details are available in the Supplementary Material.[6] For this investigation of DIP, the best number of intermediate iterations was employed for each image based on the value of the PSNR during the optimization. For the proposed MMES method, this study handly tuned the patch size $\boldsymbol{\tau}$ and the dimension r for each image. Table I shows the parameter settings of $\boldsymbol{\tau} = [\tau, \tau]$, and r for MMES. The noise level of the DAE was set as $\sigma = 0.05$ for all the images. For the AE, the same architecture shown in Fig. 7 was employed. The initial learning rate of Adam optimizer was 0.01. In addition, the learning rate was decayed by 0.98 for every 100 iterations. The optimization was stopped after 20 000 iterations for each image.

Fig. 16(a) shows the eight test images and averages of the PSNR and SSIM for the various missing ratios {50%, 70%, 90%, 95%, 99%} and the selective competitive methods. The proposed method is quite competitive with DIP. Fig. 17 shows the illustration of the results. As a result, 99% of the randomly selected voxels were removed from the 3-D (256, 256, 3)-tensors; furthermore, the tensors were recovered by various methods. Low-rank priors (HaLRTC, TMac, tSVD, and Tucker) were unable to recover a highly incomplete image. In piecewise smoothness prior (LRTV), the oversmoothed images were reconstructed since the essential image properties could not be captured. There was somewhat of a jump from them by the SPC (i.e., smooth prior of basis functions for low-rank tensor decomposition). The MDT-Tucker further

[1]For LRTV, software was downloaded from https://sites.google.com/site/yokotatsuya/home/software/lrtv_pds

[2]For SPC, software was downloaded from https://sites.google.com/site/yokotatsuya/home/software/smooth-parafac-decomposition-for-tensor-completion

[3]For GSR, each color channel was recovered, independently, using software, which was downloaded from https://github.com/jianzhangcs/GSR

[4]For MDT-Tucker, software was downloaded from https://sites.google.com/site/yokotatsuya/home/software/mdt-tucker-decomposition-for-tensor-completion

[5]For DIP, this was implemented in Python with `TensorFlow`.

[6]https://dmitryulyanov.github.io/deep_image_prior

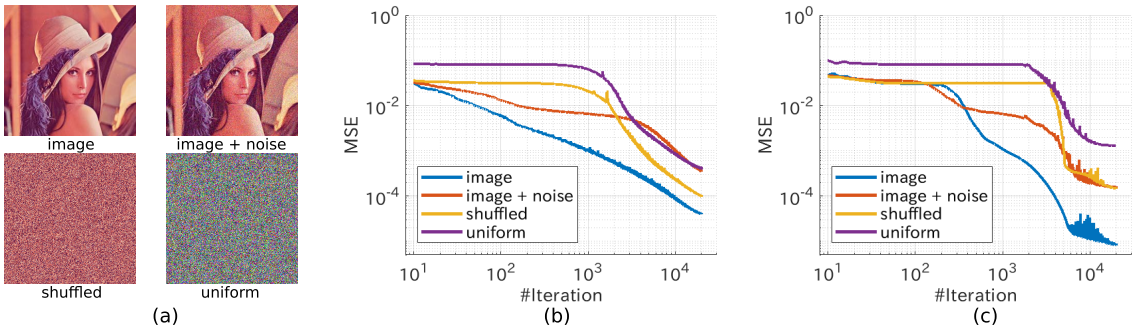


Fig. 15. Optimization behavior of DIP and MMES for the different target images. (a) Four target images. (b) and (c) Convergence behavior of DIP and proposed MMES. The natural image is optimized faster than the noisy images for DIP and MMES.

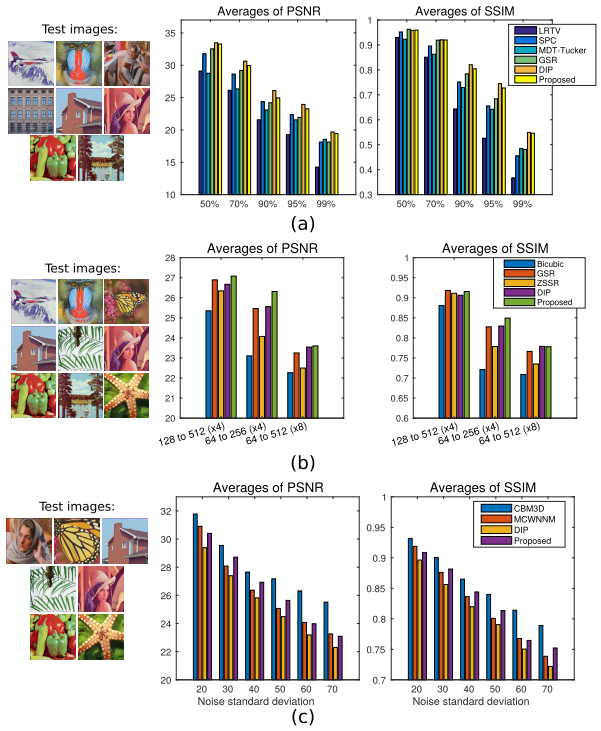


Fig. 16. Comparison of averages of PSNR and SSIM. (a) Eight color images were used for task completion with various missing rates (from 50% to 99% missing pixels). (b) Nine color images were used for the super-resolution task with three settings. (c) Seven color images were used for the denoising task with various noise levels. (a) Completion task. (b) Superresolution task. (c) Denoising task.

improves it by exploiting the shift-invariant multilinear basis. GSR recovered the global pattern of the images; however, the details are insufficient. Finally, the images reconstructed by the DIP and MMES were of significantly high quality for both global and local patterns of the images.

2) Volumetric/3-D Image/Tensor Completion: This section presents the results of the MR-image/3-D-tensor completion problem. The size of the MR image is (109, 91, 91). This study randomly removed 50%, 70%, and 90% of the voxels of the original MR-image and recovered the missing MR-images by the proposed method and DIP. For DIP, this study implemented the 3-D version of the default architecture in the `TensorFlow`; however, the number of filters of shallow layers was slightly reduced because of the GPU memory constraint. For the proposed method, the 3-D patch size was set as $\tau = [4, 4, 4]$, the lowest dimension was $r = 6$, and the noise level was $\sigma = 0.05$. The same architecture that is shown in Fig. 7 was employed.

Fig. 18 shows the reconstruction results and the behavior of the PSNR with the final value of the PSNR/SSIM in this experiment. From the values of the PSNR and SSIM, the proposed MMES outperforms the DIP in the cases of a low missing rate. In addition, it is quite competitive for highly missing cases. Degradation of the DIP might have occurred due to the insufficiency of the filter sizes since there are many filter sizes that are required for 3-D ConvNet than 2-D ConvNet. Moreover, the computational run time that is required by the MMES is significantly reduced compared with DIP.

3) Color-Image Super-Resolution: This section compares the performance of the proposed method with several unsupervised image super-resolution methods. This includes bicubic interpolation, GSR[7] [80], ZSSR[8] and DIP [61].

In these experiments, DIP was conducted with the best number of iterations from $\{1000, 2000, 3000, \ldots, 9000\}$. For four times (x4) upscaling in MMES, the following parameters were set: $\tau = 6$, $r = 32$, and $\sigma = 0.1$. For eight times (x8) upscaling in MMES, the following parameters were set: $\tau = 6$, $r = 16$, and $\sigma = 0.1$. For all the images in MMES, the architecture of the AE consists of three hidden layers with sizes of $[8\tau^2, r, 8\tau^2]$. It was assumed that the same Lanczos2 kernel was used for the downsampling system for all the super-resolution methods.

Fig. 16(b) shows the nine test images and the averages of the PSNR and SSIM for the three super-resolution settings. This study used three (256, 256, 3) color images and six (512, 512, 3) color images. Super-resolution methods recovered the four or eight times downscaled images. According to this quantitative evaluation, bicubic interpolation was clearly worse than the others were. In essence, GSR, DIP, and MMES were very competitive. In particular, DIP was slightly better than GSR, and the proposed MMES was slightly better than DIP.

Fig. 19 shows the selected high-resolution images that were reconstructed by the four super-resolution methods. In general, the bicubic method reconstructed the blurred images and these were visually worse than the others were. The GSR

[7]For GSR, each color channel was recovered, independently, using software that was downloaded from https://github.com/jianzhangcs/GSR. We slightly modified the code by applying it to the super-resolution task.

[8]For ZSSR, software was downloaded from https://github.com/assafshocher/ZSSR. This study set the same Lanczos2 kernel for the super-resolution task.

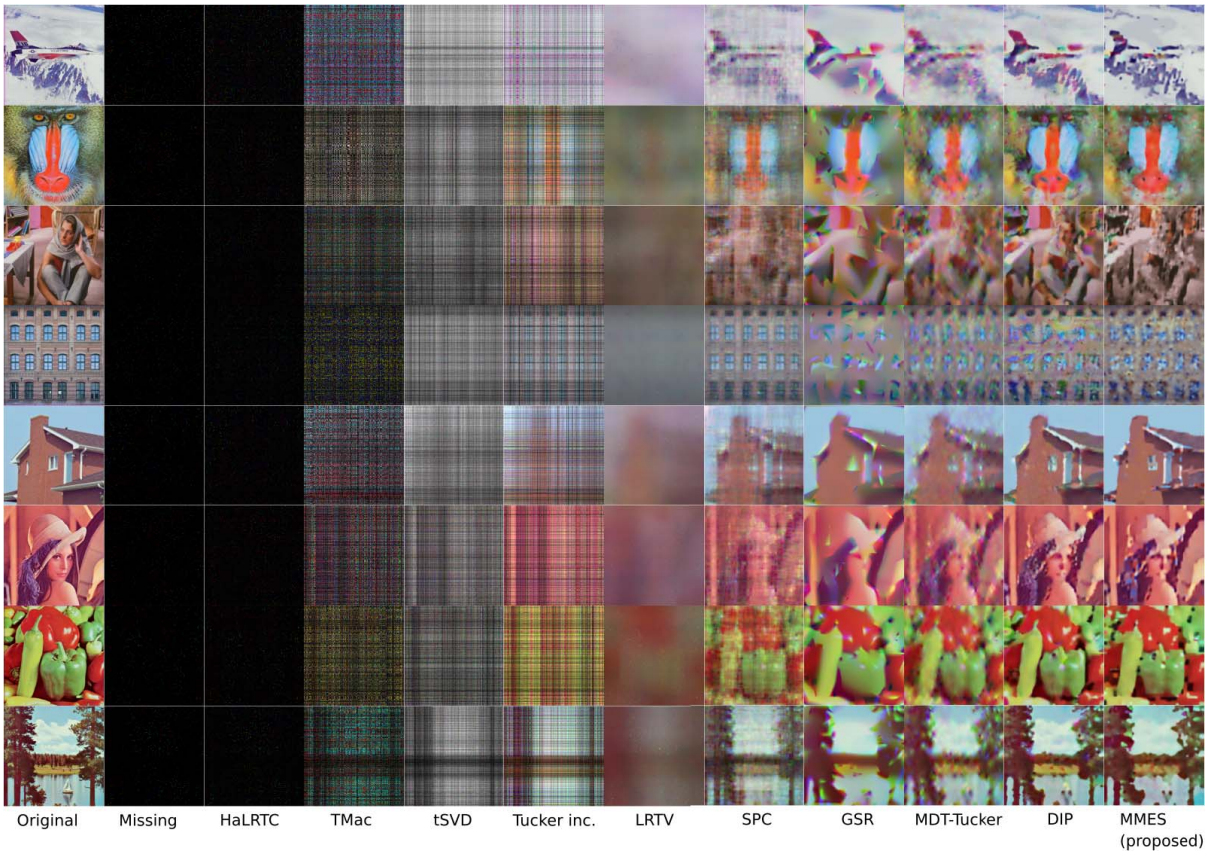


Fig. 17. Completion results from images with 99% randomly distributed missing pixels by HaLRTC [40], TMac [72], tSVD [84], Tucker inc. [75], LRTV [76], SPC [79], GSR [80], MDT-Tucker [75], DIP [61], and the proposed MMES.

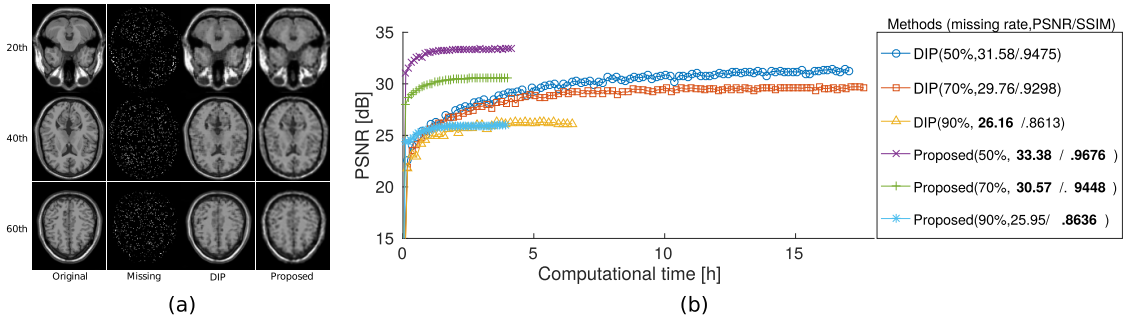


Fig. 18. Results of the MRI completion. (a) Illustration of the MRI reconstructed from a 90% missing tensor and (b) optimization behaviors of PSNR with the final values of PSNR/SSIM by the DIP and the proposed MMES.

results had smooth outlines in all the images; however, these were slightly blurred. As demonstrated, ZSSR was weak for very low-resolution images. DIP reconstructed visually sharp images; however, these images had jagged artifacts along the diagonal lines. The proposed MMES reconstructed the sharp and smooth outlines.

4) Color-Image Deconvolution: This section compares the proposed method with DIP for the image deconvolution/deblurring task. Three (256, 256, 3) color images were prepared and blurred using three different Gaussian filters. For DIP, this study chose the best early stopping time from $\{1000, 2000, \ldots, 10\,000\}$ iterations. For MMES, the fixed AE structure was employed as $[32\tau^2, r, 32\tau^2]$, and the parameters were $\tau = 4$, $r = 16$, and $\sigma = 0.01$ for all the nine cases. Fig. 20 shows the reconstructed deblurring images by DIP and MMES with these PSNR and SSIM values. It can be observed that the methods are similar to qualitatively and quantitatively.

5) Color-Image Denoising: This section compares the performance of the proposed method with that of several selected unsupervised image denoising methods: CBM3D[9] [8], MCWNNM[10] [71], and DIP [61]. This study synthetically generated noisy images by additive Gaussian noise with various standard deviations ranging from {20, 30, 40, 50, 60, 70}.

In these experiments, both DIP and MMES were conducted with the best number of iterations from {100, 200, 300,…, 9900} with the same early stopping strategy. For all the images in MMES, the architecture of the AE consists of three hidden layers with sizes of $[8\tau^2, r, 8\tau^2]$, and the parameters were set as $\tau = 6$, $r = 36$, and $\sigma = 0.05$. The tradeoff parameter λ

[9]For CBM3D, software was downloaded from http://www.cs.tut.fi/~foi/GCF-BM3D

[10]For MCWNNM, software was downloaded from https://github.com/csjunxu/MCWNNM_ICCV2017

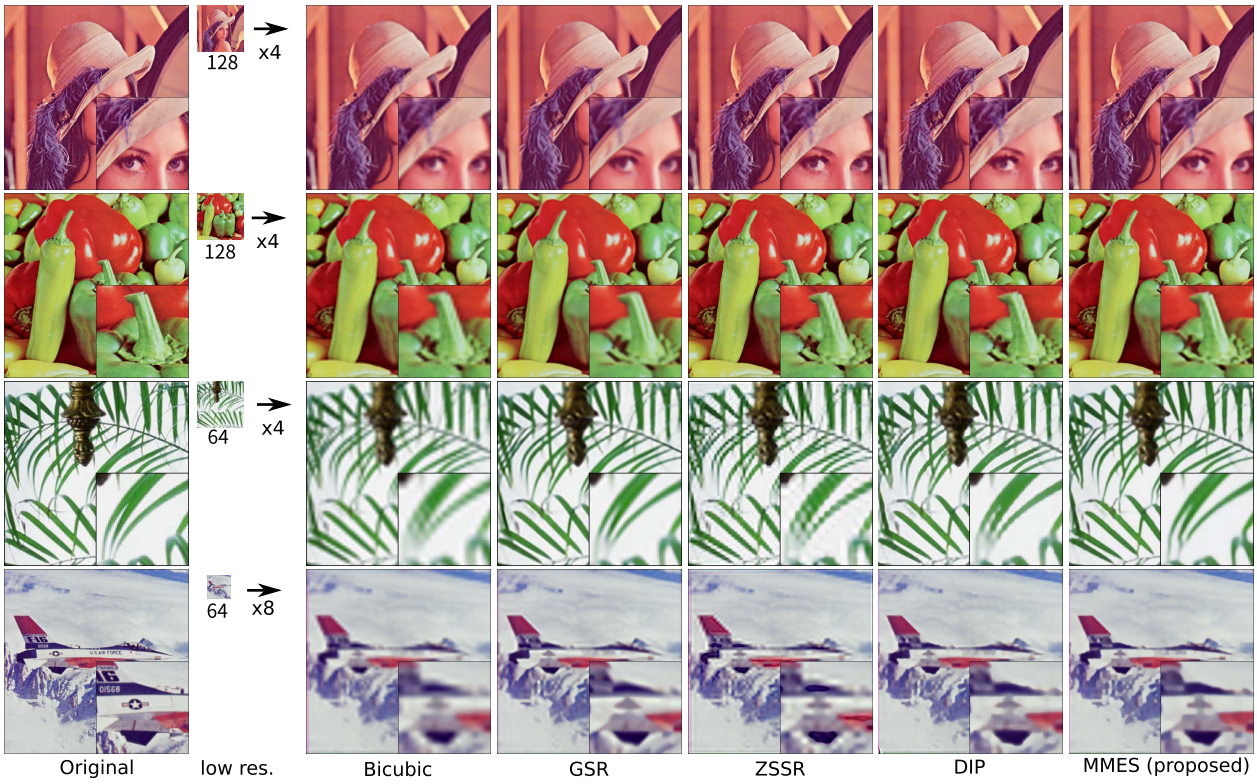


Fig. 19. Comparison of performance of super-resolution with state-of-the-art methods. The first and second lines "Lena" and "Peppers" were upscaled from (128, 128, 3) to (512, 512, 3). The third line "Leaves" was upscaled from (64, 64, 3) to (256, 256, 3). The fourth line "Airplane" was upscaled from (64, 64, 3) to (512, 512, 3).

Fig. 20. Comparison of the proposed approach with DIP for the deconvolution/deblurring task. Three color images were blurred by three Gaussian windows for different sizes. These were recovered by the DIP and the proposed MMES. The MMES provides consistently better performance.

was controlled to keep the low AE loss $\mathcal{L}_{\text{DAE}}$ and smoothly minimize the reconstruction loss $\mathcal{L}_{\text{rec}}$.

Fig. 16(c) shows the seven test images and the averages of the PSNR and SSIM for the various noise levels. According to this quantitative evaluation, CBM3D was the best for PSNR and SSIM, and DIP was slightly worse than the other methods. This study revealed that MCWNNM and MMES were competitive.

Fig. 21 shows that the selected images were reconstructed by four denoising methods. MCWNNM reconstructed the natural smooth images; however, it tends to remove too many signal components. In contrast, DIP and MMES tend to leave noise components. CBM3D provided a good balance by removing the noise while keeping the signal components.

From these results, DIP and MMES were still similar for the denoising problem. Both methods have the same limitation regarding the difficulty of the early stopping strategy, which is equivalent to the difficulty of tuning the hyperparameters for the denoising problem. This includes noise estimation and rank estimation.

V. Interpretation of MMES Toward Explaining DIP

It is well known that there is no mathematical definition of interpretability in machine learning, and there is no unique

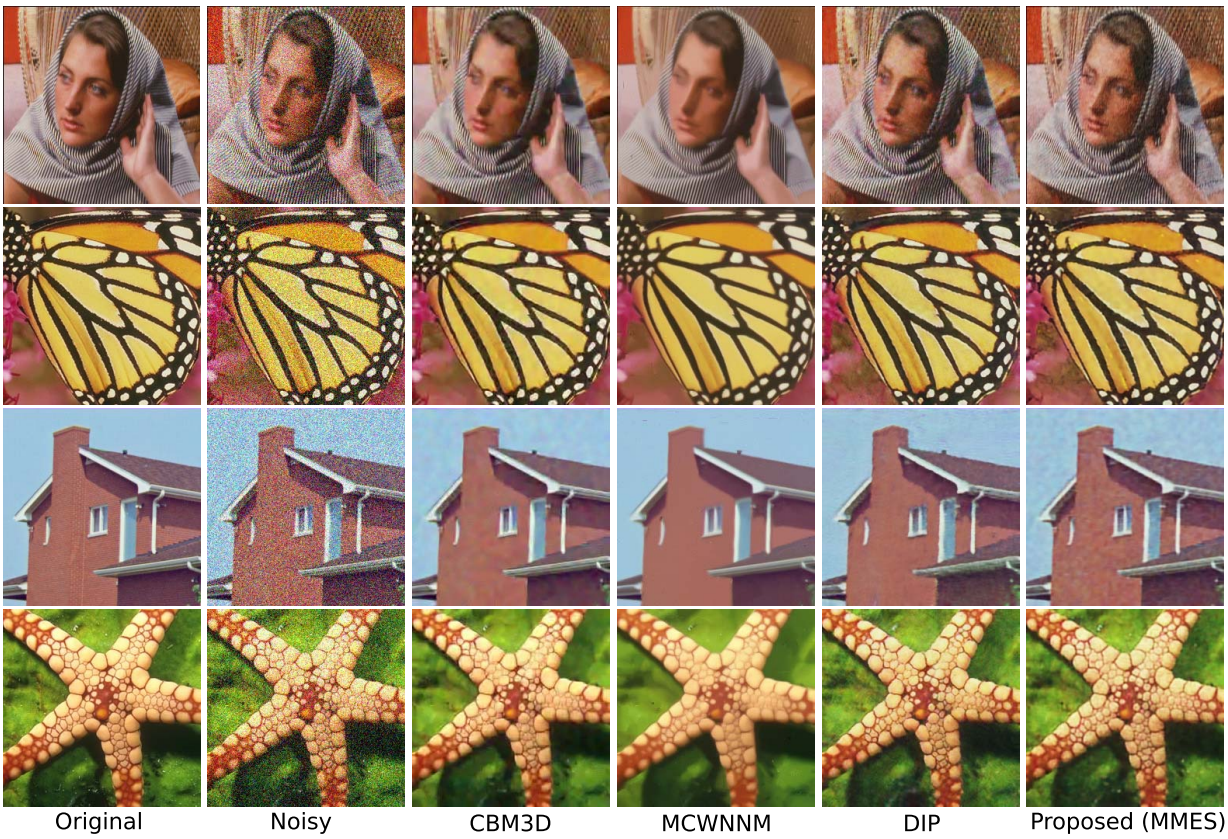


Fig. 21. Denoising results for cases where the noise standard deviation is 40.

definition for interpretation. We understand interpretability as a degree to which a human can consistently predict the model's results or performance. The higher the interpretability of a deep learning model, the easier it is for someone to comprehend why a certain performance, prediction, or expected output can be achieved. It is believed that a model is more interpretable than another model if its performance or behavior is easier for a human to comprehend than the performance of the other models.

A. From the Perspective of Dimensionality Reduction/Manifold Learning

Manifold learning and the associated AE can be viewed as the generalized nonlinear version of the principal component analysis (PCA). In fact, manifold learning solves the key problem of dimensionality reduction very efficiently. In other words, manifold learning (modeling) is an approach to nonlinear dimensionality reduction. Manifold modeling for this task is based on the idea that the dimensionality of many data sets is only artificially high. Although the patches of the images (data points) consist of several tens of pixels, they may be represented as a function of only a few or a limited number of underlying parameters. In other words, the patches are samples from a low-dimensional manifold that is embedded in a high-dimensional space. Manifold learning algorithms attempt to uncover these parameters in order to find a low-dimensional representation of the images.

With the MMES approach, this study applied original embedding via multiway delay embedding transform (MDT or Hankelization). The proposed algorithm is based on the optimization of the cost function, and it works toward extracting the low-dimensional manifold that is used to describe the high-dimensional data. The manifold is described mathematically by (5), and the cost function is formulated by (9).

B. Regarding Our Attempt to Interpret "Noise Impedance in DIP" via MMES

As mentioned in Section I, Ulyanov *et al.* [61] reported an important phenomenon of noise impedance for the ConvNet structures. The experiments in this study demonstrated that the MMES has a noise impedance that is shown in Fig. 15. This subsection provides a discussion of the noise impedance in DIP through the MMES.

Let us consider the sparse-land model [14], [15]. The noise-free images were distributed along with the low-dimensional manifolds in the high-dimensional Euclidean space and the images perturbed by the noises thicken the manifolds (i.e., make the manifolds' dimensions higher). By assuming that the image patches are sampled from the low-dimensional manifold, such as the sparse-land model, it is difficult to put noisy patches on the low-dimensional manifold. Let us consider fitting the network for noisy images. In such a case, the fastest way for decreasing the squared error (loss function) is to learn "similar patches," which frequently appears in a large set of image patches. Note that finding similar image patches for denoising is a well-known problem that has been solved (e.g., by the BM3D algorithm). In contrast, the proposed AE automatically maps similar patches into close points for a low-dimensional manifold. When similar patches have some noise, the low-dimensional representation tries to

keep the common components of the similar patches while reducing the noise components. This has been proven by Alain and Bengio [1] so that a (denoising) AE maps the input image patches toward higher density portions in the image space. In other words, a (denoising) AE has a force to reconstruct the low-dimensional patch manifold. As a result, this is a rough explanation for the noise impedance phenomenon. Although the proposed MMES and DIP are not completely equivalent, there are many analogies and similarities. It is believed that the proposed MMES model and the associated learning algorithm provide some new insights for the DIP.

VI. Conclusion and Discussion

A beautiful manifold representation of the complicated signals in the embedded space was originally discovered in a study that performed dynamical system analysis (i.e., chaos analysis) for time-series signals [46]. Afterward, many signal processing and computer vision applications have been studied; however, most methods have considered only a linear approximation because of the difficulty of nonlinear modeling [11], [38], [43], [58], [63]. Currently, the study of nonlinear/manifold modeling has significantly progressed with deep learning, and it was successfully applied in this study. We were able to apply this nonlinear system analysis not only for the time-series signals but also for the natural color images and tensors (this is an extension of the delay embedding to multiway delay embedding). To the best of our knowledge, this is the first study that applies Hankelization with AE into the general tensor for data reconstruction for a wide spectrum of applications: denoising, super-resolution, deblurring, inpainting, and so on.

MMES is a novel and simple image/tensor reconstruction model based on the low-dimensional patch-manifold prior, and it has many connections to ConvNet. We believe that it helps us understand how ConvNet/DIP work through MMES while supporting its use to DIP for various applications, such as tensor/image reconstruction or enhancement [17], [19], [64], [78].

Finally, we discussed the connections between different research areas, such as the dynamical system analysis, deep learning, and tensor modeling. The proposed method is a prototype, and it can be further improved by incorporating other methods, such as regularizations, multiscale extensions, and adversarial training.

Appendix
Relation to Convolutional Sparse Coding

Here, we discuss the similarities and differences between the CSC and the MMES. First, we remind the CSC model [48] can be described as follows:

$$\boldsymbol{x}_{\mathrm{CSC}} = \sum_{i=1}^{N} \boldsymbol{R}_i^T \boldsymbol{D}_L \boldsymbol{\alpha}_i \tag{10}$$

where $\boldsymbol{x}_{\mathrm{CSC}} \in \mathbb{R}^N$ is a reconstructed signal by the CSC model, $\boldsymbol{D}_L \in \mathbb{R}^{n\times m}$ is a dictionary matrix, $\boldsymbol{\alpha}_i \in \mathbb{R}^m$ is a sparse coefficient vector, and $\boldsymbol{R}_i^T \in \{0,1\}^{N\times n}$ is a binary matrix for performing the shift operation. In fact, (10) can be transformed to the following equivalent formulation:

$$\boldsymbol{x}_{\mathrm{CSC}} = \mathcal{H}^{\dagger}([\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_N]) \tag{11}$$

where $\boldsymbol{s}_i = \boldsymbol{D}_L\boldsymbol{\alpha}_i \in \mathbb{R}^n$, and $\mathcal{H}^{\dagger}$ is an operator of inverse delay embedding. On the other hand, MMES can be represented as

$$\boldsymbol{x}_{\mathrm{MMES}} = \mathcal{H}^{\dagger}([\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_N]) \tag{12}$$

where $\boldsymbol{h}_i = \psi_r(\boldsymbol{l}_i) \in \mathbb{R}^n$ is the output of the AE, where $\boldsymbol{l}_i \in \mathbb{R}^r$ is a latent variable. Note that the dimensions of $\boldsymbol{s}_i$ and $\boldsymbol{h}_i$ correspond to each other. In this regard, CSC and MMES have similar structures in terms of inverse delay embedding of a matrix.

The fundamental difference between CSC and MMES is based on the following two points.

1) Sparse coding of $\boldsymbol{s}_i$ is used in CSC, while AE for $\boldsymbol{h}_i$ is employed in MMES.
2) In CSC, the Hankel constraint is not imposed on $\boldsymbol{S} = [\boldsymbol{s}_1, \ldots, \boldsymbol{s}_N]$, while, in MMES, we impose the Hankel constraint on $\boldsymbol{H} = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N]$.

First, sparse coding and AE are quite different although both models can be commonly used for encoding data. Each $\boldsymbol{s}_i \in \mathbb{R}^n$ is represented by $\boldsymbol{\alpha}_i \in \mathbb{R}^m$ in sparse coding, and $\boldsymbol{h}_i \in \mathbb{R}^n$ is represented by $\boldsymbol{l}_i \in \mathbb{R}^r$ in the AE. In sparse coding, various data are encoded into sparse vectors using a redundant dictionary so that $n < m$ in appearance. By contrast, in the AE, $r < n$ because various data are directly encoded into low-dimensional vectors by a differentiable nonlinear map. Second, the Hankel constraint of the latent matrix is employed only in MMES. Since the Hankel constraint enforces the consistency of overlapped patches, $\boldsymbol{h}_i$ directly represents the 'patch' in the image, but $\boldsymbol{s}_i$ represents the "basis" of it. In other words, the Hankel constraint is necessary to understand the proposed method as manifold modeling "in embedded space."

Acknowledgment

Please visit https://github.com/yokotatsuya/MMES for python code.

References

[1] G. Alain and Y. Bengio, "What regularized auto-encoders learn from the data-generating distribution," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 3563–3593, 2014.
[2] J. Batson and L. Royer, "Noise2Self: Blind denoising by self-supervision," in *Proc. ICML*, 2019, pp. 524–533.
[3] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Comput.*, vol. 7, no. 1, pp. 108–116, Jan. 1995.
[4] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 2, Jun. 2005, pp. 60–65.
[5] S. Cha, T. Park, and T. Moon, "GAN2GAN: Generative noise learning for iterative blind denoising with single noisy images," 2019, *arXiv:1905.10488*. [Online]. Available: http://arxiv.org/abs/1905.10488
[6] A. Cichocki, R. Zdunek, A. H. Phan, and S. Amari, *Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-Way Data Analysis and Blind Source Separation*. Hoboken, NJ, USA: Wiley, 2009.
[7] A. Creswell and A. A. Bharath, "Denoising adversarial autoencoders," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 4, pp. 968–984, Apr. 2019.
[8] 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. ICIP*, vol. 1, 2007, p. I-313.
[9] 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, Aug. 2007.
[10] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," in *Proc. ICLR*, 2014, pp. 1–14.
[11] T. Ding, M. Sznaier, and O. I. Camps, "A rank minimization approach to video inpainting," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[12] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. ECCV*. Cham, Switzerland: Springer, 2014, pp. 184–199.

[13] 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, Feb. 2016.

[14] M. Elad and M. Aharon, "Image denoising via learned dictionaries and sparse representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Jun. 2006, pp. 895–900.

[15] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[16] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. ICML*, 2016, pp. 1050–1059.

[17] Y. Gandelsman, A. Shocher, and M. Irani, "Double-DIP: Unsupervised image decomposition via coupled deep-image-priors," in *Proc. CVPR*, Jun. 2019, pp. 11026–11035.

[18] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[19] K. Gong, C. Catana, J. Qi, and Q. Li, "PET image reconstruction using deep image prior," *IEEE Trans. Med. Imag.*, vol. 38, no. 7, pp. 1655–1665, Jul. 2019.

[20] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[21] W. E. L. Grimson, *From Images to Surfaces: A Computational Study of the Human Early Visual System*. Cambridge, MA, USA: MIT Press, 1981.

[22] S. Gu, L. Zhang, W. Zuo, and X. Feng, "Weighted nuclear norm minimization with application to image denoising," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2862–2869.

[23] F. Guichard and F. Malgouyres, "Total variation based interpolation," in *Proc. EUSIPCO*, 1998, pp. 1–4.

[24] R. Heckel and P. Hand, "Deep decoder: Concise image representations from untrained non-convolutional networks," 2018, *arXiv:1810.03982*. [Online]. Available: http://arxiv.org/abs/1810.03982

[25] G. E. Hinton, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, Jul. 2006.

[26] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Math. Phys.*, vol. 6, nos. 1–4, pp. 164–189, Apr. 1927.

[27] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *J. Educ. Psychol.*, vol. 24, no. 6, p. 417, 1933.

[28] W. Hu, D. Tao, W. Zhang, Y. Xie, and Y. Yang, "The twist tensor nuclear norm for video completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 12, pp. 2961–2973, Dec. 2017.

[29] A. Hyvarinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, vol. 46. Hoboken, NJ, USA: Wiley, 2004.

[30] H. Ji, C. Liu, Z. Shen, and Y. Xu, "Robust video denoising using low rank matrix completion," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 1791–1798.

[31] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, "Deep convolutional neural network for inverse problems in imaging," *IEEE Trans. Image Process.*, vol. 26, no. 9, pp. 4509–4522, Sep. 2017.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[33] A. Krull, T.-O. Buchholz, and F. Jug, "Noise2Void-learning denoising from single noisy images," in *Proc. CVPR*, Jun. 2019, pp. 2129–2137.

[34] S. Laine, T. Karras, J. Lehtinen, and T. Aila, "High-quality self-supervised deep image denoising," 2019, *arXiv:1901.10277*. [Online]. Available: http://arxiv.org/abs/1901.10277

[35] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, p. 788, 1999.

[36] J. Lehtinen *et al.*, "Noise2noise: Learning image restoration without clean data," in *Proc. ICML*, 2018, pp. 2971–2980.

[37] S. Z. Li, "Markov random field models in computer vision," in *Proc. ECCV*. Berlin, Germany: Springer, 1994, pp. 361–370.

[38] Y. Li, K. J. R. Liu, and J. Razavilar, "A parameter estimation scheme for damped sinusoidal signals based on low-rank hankel approximation," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 481–486, Feb. 1997.

[39] Z. Li, J. Tang, and T. Mei, "Deep collaborative embedding for social image understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2070–2083, Sep. 2019.

[40] J. Liu, P. Musialski, P. Wonka, and J. Ye, "Tensor completion for estimating missing values in visual data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 208–220, Jan. 2013.

[41] A. Majumdar, "Blind denoising autoencoder," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 1, pp. 312–317, Jan. 2019.

[42] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*. [Online]. Available: http://arxiv.org/abs/1511.05644

[43] I. Markovsky, "Structured low-rank approximation and its applications," *Automatica*, vol. 44, no. 4, pp. 891–909, Apr. 2008.

[44] M. Nickel and D. Kiela, "Poincaré embeddings for learning hierarchical representations," in *Proc. NIPS*, 2017, pp. 6338–6347.

[45] S. Osher, Z. Shi, and W. Zhu, "Low dimensional manifold model for image processing," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1669–1690, Jan. 2017.

[46] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw, "Geometry from a time series," *Phys. Rev. Lett.*, vol. 45, no. 9, p. 712, Sep. 1980.

[47] V. Papyan, Y. Romano, and M. Elad, "Convolutional neural networks analyzed via convolutional sparse coding," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2887–2938, 2017.

[48] V. Papyan, Y. Romano, M. Elad, and J. Sulam, "Convolutional dictionary learning via local processing," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5296–5304.

[49] K. Pearson, "LIII. On lines and planes of closest fit to systems of points in space," *London, Edinburgh, Dublin Phil. Mag. J. Sci.*, vol. 2, no. 11, pp. 559–572, Nov. 1901.

[50] G. Peyré, "Manifold models for signals and images," *Comput. Vis. Image Understand.*, vol. 113, no. 2, pp. 249–260, Feb. 2009.

[51] T. Poggio, V. Torre, and C. Koch, "Computational vision and regularization theory," *Nature*, vol. 317, p. 26, Jan. 1985.

[52] M. Rosca, B. Lakshminarayanan, D. Warde-Farley, and S. Mohamed, "Variational approaches for auto-encoding generative adversarial networks," 2017, *arXiv:1706.04987*. [Online]. Available: http://arxiv.org/abs/1706.04987

[53] T. R. Shaham, T. Dekel, and T. Michaeli, "SinGAN: Learning a generative model from a single natural image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4570–4580.

[54] Q. Shi, Y.-M. Cheung, Q. Zhao, and H. Lu, "Feature extraction for incomplete data via low-rank tensor decomposition with feature regularization," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1803–1817, Jun. 2019.

[55] A. Shocher, S. Bagon, P. Isola, and M. Irani, "InGAN: Capturing and retargeting the 'DNA' of a natural image," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4492–4501.

[56] A. Shocher, N. Cohen, and M. Irani, "Zero-shot super-resolution using deep internal learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3118–3126.

[57] S. Sonoda and N. Murata, "Transportation analysis of denoising autoencoders: A novel method for analyzing deep neural networks," 2017, *arXiv:1712.04145*. [Online]. Available: http://arxiv.org/abs/1712.04145

[58] M. Szummer and R. W. Picard, "Temporal texture modeling," in *Proc. ICIP*, vol. 3, 1996, pp. 823–826.

[59] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc., B, Methodol.*, vol. 58, no. 1, pp. 267–288, Jan. 1996.

[60] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, no. 3, pp. 279–311, Sep. 1966.

[61] V. Lempitsky, A. Vedaldi, and D. Ulyanov, "Deep image prior," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9446–9454.

[62] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *Int. J. Comput. Vis.*, vol. 128, no. 7, pp. 1867–1888, Jul. 2020.

[63] P. Van Overschee and B. De Moor, "Subspace algorithms for the stochastic identification problem," in *Proc. 30th IEEE Conf. Decis. Control*, Dec. 1991, pp. 1321–1326.

[64] D. Van Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis, "Compressed sensing with deep image prior and learned regularization," 2018, *arXiv:1806.06438*. [Online]. Available: http://arxiv.org/abs/1806.06438

[65] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 1096–1103.

[66] C. R. Vogel and M. E. Oman, "Fast, robust total variation-based reconstruction of noisy, blurred images," *IEEE Trans. Image Process.*, vol. 7, no. 6, pp. 813–824, Jun. 1998.

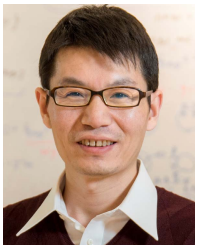
[67] W. Wang, V. Aggarwal, and S. Aeron, "Efficient low rank tensor ring completion," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5697–5705.
[68] F. Williams, T. Schneider, C. Silva, D. Zorin, J. Bruna, and D. Panozzo, "Deep geometric prior for surface reconstruction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10130–10139.
[69] Y. Wu, H. Tan, Y. Li, J. Zhang, and X. Chen, "A fused CP factorization method for incomplete tensors," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 3, pp. 751–764, Mar. 2019.
[70] J. Xu *et al.*, "Noisy-as-clean: Learning self-supervised denoising from the corrupted image," 2019, *arXiv:1906.06878*. [Online]. Available: http://arxiv.org/abs/1906.06878
[71] 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. (ICCV)*, Oct. 2017, pp. 1096–1104.
[72] Y. Xu *et al.*, "Parallel matrix factorization for low-rank tensor completion," *Inverse Problems Imag.*, vol. 9, no. 2, pp. 601–624, 2015.
[73] J. Xue, Y. Zhao, W. Liao, J. C.-W. Chan, and S. G. Kong, "Enhanced sparsity prior model for low-rank tensor completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 11, pp. 4567–4581, Nov. 2020.
[74] N. Yair and T. Michaeli, "Multi-scale weighted nuclear norm image restoration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3165–3174.
[75] T. Yokota, B. Erem, S. Guler, S. K. Warfield, and H. Hontani, "Missing slice recovery for tensors using a low-rank model in embedded space," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8251–8259.
[76] T. Yokota and H. Hontani, "Simultaneous visual data completion and denoising based on tensor rank and total variation minimization and its primal-dual splitting algorithm," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3732–3740.
[77] T. Yokota and H. Hontani, "Simultaneous tensor completion and denoising by noise inequality constrained convex optimization," *IEEE Access*, vol. 7, pp. 15669–15682, 2019.
[78] T. Yokota, K. Kawai, M. Sakata, Y. Kimura, and H. Hontani, "Dynamic PET image reconstruction using nonnegative matrix factorization incorporated with deep image prior," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3126–3135.
[79] T. Yokota, Q. Zhao, and A. Cichocki, "Smooth PARAFAC decomposition for tensor completion," *IEEE Trans. Signal Process.*, vol. 64, no. 20, pp. 5423–5436, Oct. 2016.
[80] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.
[81] 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, Jul. 2017.
[82] L. Zhang, W. Wei, Q. Shi, C. Shen, A. van den Hengel, and Y. Zhang, "Accurate tensor completion via adaptive low-rank representation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 10, pp. 4170–4184, Oct. 2020.
[83] X. Zhang, "A nonconvex relaxation approach to low-rank tensor completion," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 6, pp. 1659–1671, Jun. 2019.
[84] Z. Zhang, G. Ely, S. Aeron, N. Hao, and M. Kilmer, "Novel methods for multilinear data completion and de-noising based on tensor-SVD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3842–3849.
[85] Q. Zhao, L. Zhang, and A. Cichocki, "Bayesian CP factorization of incomplete tensors with automatic rank determination," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1751–1763, Sep. 2015.

Tatsuya Yokota (Senior Member, IEEE) received the Ph.D. degree in engineering from the Tokyo Institute of Technology, Tokyo, Japan, in 2014.

From 2011 to 2014, he was a Junior Research Associate with the Laboratory for Advanced Brain Signal Processing (ABSP), RIKEN Brain Science Institute (BSI), Wako, Japan. From 2014 to 2016, he was a Research Scientist with the Laboratory for ABSP and a Visiting Research Scientist with the TOYOTA Collaboration Center, RIKEN BSI. He is currently an Assistant Professor with the Department of Computer Science, Nagoya Institute of Technology, Nagoya, Japan, and a Visiting Scientist with the Tensor Learning Team, RIKEN Center for Advanced Intelligence Project, Tokyo, Japan. His research interests include matrix/tensor factorizations, signal/image processing, pattern recognition, and machine learning.

Hidekata Hontani (Member, IEEE) received the B.E., M.E., and Ph.D. degrees from The University of Tokyo, Tokyo, Japan, in 1991, 1993, and 2000, respectively.

From 1993 to1996, he was with Toshiba Corporation, Tokyo. From 1996 to 2000, he was a Research Associate with The University of Tokyo. From 2000 to 2005, he was an associate professor at Yamagata University. In 2005, he joined the Nagoya Institute of Technology, Nagoya, Japan, where he is currently a Professor. In 2010, he was a Researcher with the University of California at Los Angeles (UCLA), Los Angeles, CA, USA. His research activities include image processing, pattern recognition, and signal processing.



Qibin Zhao (Senior Member, IEEE) received the Ph.D. degree in computer science from Shanghai Jiao Tong University, Shanghai, China, in 2009.

He was a Research Scientist with the RIKEN Brain Science Institute, Wako, Japan, from 2009 to 2017. He is currently the Leader of the Tensor Learning Team, RIKEN Center for Advanced Intelligence Project (AIP), Tokyo, Japan, a Visiting Professor with the Saitama Institute of Technology, Fukaya, Japan, and the Guangdong University of Technology, Guangzhou, China, and a Visiting Associate Professor with the Tokyo University of Agriculture and Technology, Fuchu, Japan. He has more than 120 publications in international journals and conferences and two monographs. His research interests include machine learning, tensor factorization and tensor networks, computer vision, and brain signal processing.

Dr. Zhao is also an Editorial Board Member of *Science China Technological Sciences*. He also serves as the Area Chair of NeurIPS 2020, ICLR 2021, AISTATS 2021, IJCAI 2021, and ICML 2021.

Andrzej Cichocki (Fellow, IEEE) received the M.Sc. (Hons.), Ph.D., and Dr.Sc. (Habilitation) degrees in electrical engineering from the Warsaw University of Technology, Warsaw, Poland.

He was an Alexander-von-Humboldt Research Fellow and a Guest Professor with the University of Erlangen-Nuremberg, Erlangen, Germany. From 1995 to 2018, he was the Team Leader and the Head of the Laboratory for Advanced Brain Signal Processing, RIKEN Brain Science Institute, Wako, Japan. He is currently a Professor with the Skolkovo Institute of Science and Technology (Skoltech), Moscow, Russia. He is also a Visiting/Adjunct Professor with the Tokyo University of Agriculture and Technology (TUAT), Fuchu, Japan, Hangzhou Dianzi University (HDU), Hangzhou, China, Nicolaus Copernicus University (UMK), Toruń, Poland, and the Institute of Systems Research (IBS), Polish Academy of Sciences, Warsaw. He is the author of more than 500 peer-review articles and six monographs in English (two of them are translated to Chinese). His publications currently report over 46 000 citations according to Google Scholar, with an H-index of 97. His research focuses on deep learning, tensor decompositions, tensor networks for big data analytics, multiway blind source separation, and brain–computer interface and their biomedical applications.

Prof. Cichocki is among the three most cited Polish Computer Scientists. He has served as the Founding Editor-in-Chief of *Computational Intelligence and Neuroscience*. He has also served as an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, the IEEE TRANSACTIONS ON CYBERNETICS, and the *Journal of Neuroscience Methods*.