Electronic Research Archive

http://www.aimspress.com/journal/ERA

ERA, 34(1): 477–508.
DOI: 10.3934/era.2026023
Received: 22 October 2025
Revised: 01 December 2025
Accepted: 16 December 2025
Published: 16 January 2026

Research article

Tensor ring decomposition with data-driven for color image completion

Yingpin Chen[1,2], Peiqi Zhuo[1,2], Hualin Zhang[1,2,*], Yuan Liao[1,2], Zhixiang Chen[1,2,*], Ronghuan Zhang[1,2] and Yujuan Xu[1,2]

[1] School of Physics and Information Engineering, Minnan Normal University, Zhangzhou 363000, China
[2] Key Laboratory of Light Field Manipulation and System Integration Applications in Fujian Province, Minnan Normal University, Zhangzhou 363000, China

*** Correspondence:** Email: zhl0225@mnnu.edu.cn, zxchenphd@163.com.

Abstract: Tensor ring (TR) decomposition has demonstrated remarkable capability in capturing low-rank tensor structures, achieving significant success in color image completion tasks. However, relying solely on the tensor low-rank prior cannot fully capture the details of a color image. Thus, traditional model-driven TR decomposition models often fail to recover satisfactory local detail. A natural idea is to introduce a deep neural network with end-to-end training to improve the detail quality of image reconstruction. Thus, we proposed a novel hybrid model that integrates model-driven TR decomposition with data-driven deep learning regularization. The proposed framework introduces an energy functional regularization term based on the FFDNET architecture, which enhances the robustness of rank selection while preserving global low-rank and local details. In particular, we assumed that the unfolding matrices of the TR factors exhibited low-rank properties. Thus, nuclear-norm regularization constraints were incorporated on the TR factors to enhance their global low-rank characteristics. Additionally, a deep prior regularization term derived from the FFDNET network was introduced to preserve the local details of the target tensor. We further developed an efficient alternating direction method for the multiplier algorithm to address the associated optimization problem. Extensive experiments on color images have demonstrated that the proposed method outperforms denoising approaches, yielding satisfactory results. The code for implementing the proposed method is available at https://github.com/110500617/TRDD.

Keywords: tensor ring decomposition; deep learning; FFDNET; ADMM; data-driven

1. Introduction

Three-dimensional (3D) color images contain richer information and higher redundancy than two-dimensional (2D) grayscale images, making them more suitable for describing real scenes. However, the influence of sensor errors, interference during signal transmission, and damage to storage media inevitably introduces noise (Gaussian or missing data) into color images [1]. Missing data in color images not only affects the image quality but also hinders subsequent applications. Therefore, color image completion is an important research topic in image processing.

The core idea of image recovery is to utilize various image characteristics to achieve denoising. As such, numerous image recovery methods have been proposed. These can be approximately divided into two major categories: Methods based on 2D image processing [2–4] of each channel and those based on 3D tensor processing across channels [5–10]. The 2D image processing methods first decompose the tensors and rearrange them into matrices, then recover them channel by channel. Common 2D methods include the total variation (TV) [2–4], sparse representation methods [11,12], and low-rank-based methods [13–15]. For example, Thanh et al. [3] proposed an adaptive TV (ATV) regularization model for removing salt-and-pepper noise in digital images, yielding edge-preserving restored images, with well-preserved main structures and no artifacts. To suppress the staircase effect of the TV model, Yang et al. [16] introduced high-order TV into the anisotropic TV model and used the Lp pseudo-norm to describe the sparsity of low- and high-order TVs. Unlike the TV-based local smoothness constraint, methods based on sparse representation assume that a set of fixed or learnable transforms sparsely represent the image to be recovered. For example, Chen et al. [11] introduced the stationary wavelet transform for removing salt-and-pepper noise from images. Low-rank-based methods utilize the low-rank prior in each channel. Representative works include: WNNM [13], NCSR [14], and LSSC [15]. Scholars have gradually realized that relying solely on a single regularization constraint is insufficient to achieve satisfactory image reconstruction performance. An effective solution is to impose multiple prior constraints on 2D image reconstruction tasks simultaneously. For example, Wang et al. [17] proposed combining structured smoothness constraints with image low-rank constraints for image inpainting, achieving impressive reconstruction quality. These 2D methods effectively remove low-level noise. However, they do not achieve satisfactory denoising when noise levels are high, as the redundant information in the channels has not been fully utilized. Conversely, methods based on 3D tensor processing effectively exploit the data redundancy between channels and achieve better image reconstruction performance.

A tensor, as a multi-dimensional generalization of vectors and matrices, represents the complex interactions within high-dimensional data. A 3D image is a stack of several 2D images that contain three dimensions: One color dimension and two spatial dimensions. Therefore, it can be modeled as third-order tensor data. Popular tensor ranks for image completion include the CANDECOMP/PARAFAC (CP) [18–20] rank, Tucker [21–26] rank, Tubal tensor [27,28], fiber rank [29], and so on. The CP rank is defined as the minimum value of the tensor decomposition rank. However, the calculation of the CP rank is typically nondeterministic and polynomial hard. The Tucker rank is a vector that comprises the ranks of each mode unfolding matrix of the tensor. It is easy to calculate, and its convex relaxation is easier to handle than that of the CP rank. For instance, Goldfarb and Qin [30] introduced the matrix nuclear norm and low-rank matrix factors to estimate the Tucker rank. However, the calculation of the Tucker rank requires unfolding the tensor into matrices along all modes. It should be emphasized that the unfolding operator in Tucker decomposition is an unbalanced matricization unfolding and will destroy the original structure of the tensor. Therefore, the Tucker rank cannot well express the global low-rankness of the tensor [31]. Moreover, to avoid destroying the tensor structure caused by matrix flattening of third-order tensors, the tensor tubal rank minimization method [27,28], based on tensor product and tensor singular value decomposition, has been proposed and has garnered considerable attention. However, the tubal rank considers only the correlations of the tensor along the mode-3 direction and lacks the flexibility to handle correlations across dimensions. To address this limitation, Zheng et al. [29] proposed the tensor fiber rank, which comprehensively explores the correlations of data across dimensions along different directions.

Furthermore, tensor network decompositions [32] have garnered widespread attention due to their remarkable ability to capture the correlations between various modes of tensors. The tensor train (TT) [33] model is a kind of tensor network decomposition, which decomposes an N-th-order tensor into the multilinear product of 2 matrices and N-2 third-order tensors. Subsequently, Zhao et al. [34] proposed the tensor ring (TR) decomposition, which represents higher-order tensors as sequences of cyclic contractions of third-order tensors. TR decomposition can be regarded as an extension of the TT model to define a high-dimensional tensor with circular multilinear products over a series of third-order factor tensors (also known as TR factors). TR decomposition can better approximate higher-order tensors than Tucker and CP decomposition because each tensor factor can be circularly shifted and treated equally under trace operations. Based on the definition and properties of TR decomposition, the TR form is more general and effective than other tensor decomposition forms [35]. First, the number of stored variables is much lower than that in Tucker decomposition, where the number of variables increases exponentially relative to the original tensor order. Therefore, the number of variables that TR needs to estimate is significantly less, and an estimation based on TR is usually more accurate than decomposition by Tucker. Second, a characteristic of TR is that each tensor factor can undergo a cyclic shift and equivalent processing under trace operations [35], whereas other tensor decompositions cannot maintain this advantage. Third, the decomposition format of TR is tensor-to-tensor, which preserves the original tensor's structure. That is, the TR model does not expand the higher-order tensor into a matrix. Therefore, it utilizes the correlations between the dimensions of the original image, retaining the rich information contained within it. Given the advantages of TR decomposition, it was applied to image completion [36–40] and denoising [41,42] as soon as it was proposed, yielding excellent experimental results.

Although low-rank tensor recovery methods have achieved great success in practical applications, preserving the rich details in multi-dimensional image recovery remains a bottleneck. Many scholars have introduced additional priors, such as local smoothness and non-local similarity, to enhance the recovery effectiveness. Additionally, data-driven prior regularization has helped preserve detailed image information. Various deep learning (DL)-based denoising techniques utilize a large number of natural images for pre-training and achieve satisfactory image detail reconstruction quality. Scholars have gradually realized that denoising deep neural networks can be used as plug-and-play implicit regularization terms to complement further image completion models that are based on the global tensor low-rank prior. For example, Zhao et al. [43] introduced DL into the tensor-kernel norm minimization model and achieved satisfactory experimental results in recovering the underlying tensors. Zhang et al. [44] combined a data-driven prior with parallel matrix factorization, simultaneously adopting depth priors and low-rank constraints, to enhance the model's ability to restore image details. Inspired by these works, we propose a data-driven tensor ring decomposition (TR-DD) completion model. We first discuss the relationship between the processed tensor and each

TR factor, and strictly prove that the upper bound of the rank of each mode unfolding matrix of the tensor is constrained by the rank of the TR factors. On this basis, the proposed work imposes the nuclear norm regularization on TR factors to explore the low-rank structure of image tensors. Then, the data-driven, fast, and flexible denoising convolutional neural network (FFDNET) [45] is utilized to capture the detailed information within the images that is difficult to recover under low-rank constraints. Accordingly, we propose a model that can be solved using the alternating direction method of multipliers (ADMM) algorithm [46]. Finally, the proposed model is compared with other models.

The contributions of this study are as follows:

1) We impose the nuclear norm regularization on TR factors to further explore the low-rank prior of the processed color images and further improve the detail preserving capacity by introducing the data-driven FFDNET as an implicit regularization term. In this way, the advantages of model-driven and data-driven methods are effectively combined.

2) An efficient algorithm based on ADMM is proposed to solve this model effectively. In the ADMM framework, the complex problem is decomposed into several simpler subproblems. Different subproblems of the optimization problem can be designed and optimized independently of each other. The experimental results show that the ADMM enables the image completion results to obtain satisfactory convergent solutions.

3) The proposed method was applied to color-image completion and compared with some state-of-the-art tensor completion methods. The experimental results demonstrate that this method is effective not only in restoring the missing values of the image but also in capturing texture information that is difficult to recover under low-rank constraints.

The remainder of this article is organized as follows: In Section 2, we provide preparatory knowledge on TR decomposition and the necessary mathematical explanations for the proposed methods. In Section 3, we present the data-driven TR image reconstruction method and provide the ADMM algorithms for each model stage. In Section 4, we present a series of experiments that verify the effectiveness of the proposed method, summarize the full text, and discuss future research directions.

2. Preliminaries

For the convenience of subsequent discussions, in this section, we provide explanations of the relevant symbols, including the definition and analytical expression of the general solution for the TR decomposition.

2.1. Symbol explanations

In this study, scalars, vectors, matrices, and tensors are denoted by x, $\boldsymbol{x}$, $\boldsymbol{X}$ and $\mathcal{X}$, respectively. For a K-th order tensor, $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_K}$, $\mathcal{X}(i_1, i_2, \cdots, i_K)$ denotes the $(i_1, i_2, \cdots, i_K)$-th element of $\mathcal{X}$. The l_2 norm of $\mathcal{X}$ is defined as $\|\mathcal{X}\|_2 = \sqrt{\sum_{i_1, i_2, \cdots, i_K} \left|\mathcal{X}(i_1, i_2, \cdots, i_K)\right|^2}$. For convenience, we set the tensor sequence $\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(K)}$ as $\mathcal{G}^{(1:K)}$.

2.2. TR decomposition model

In TR decomposition, high-order tensors are expressed as a sequence of third-order tensors connected cyclically through matrix multiplication [34]. Let $\mathcal{R}$ be a K-th order tensor of size $I_1 \times I_2 \times \ldots \times I_K$, denoted $\mathcal{R} \in \mathbb{R}^{I_1 \times I_2 \times .. \times I_K}$ The TR decomposition factorizes $\mathcal{R}$ into a sequence of core $\mathcal{G}^{(k)} \in \mathbb{R}^{R_k \times I_k \times R_{k+1}}, k = 1,2,\ldots,K$, which can be denoted in the form of elements, that is

$$\mathcal{R}(i_1,i_2,\cdots,i_K) = \mathrm{Tr}\left\{\boldsymbol{G}^{(1)}(i_1)\boldsymbol{G}^{(2)}(i_2)\cdots\boldsymbol{G}^{(K)}(i_K)\right\} = \mathrm{Tr}\left\{\prod_{k=1}^{K}\boldsymbol{G}^{(k)}(i_k)\right\}, \tag{1}$$

where the symbol Tr represents the trace of a matrix. $\mathcal{R}(i_1,i_2,\cdots,i_K)$ denotes the $(i_1,i_2,\cdots,i_K)$-th element of the tensor $\mathcal{R}$ and $\boldsymbol{G}^{(k)}(i_k)$ denotes the i_k-th lateral slice matrix of the latent tensor $\mathcal{G}^{(k)}$ of size $R_k \times R_{k+1}$. Note that for any two adjacent latent tensors $\mathcal{G}^{(k)}$ and $\mathcal{G}^{(k+1)}$, an identical modulus R_{k+1} exists on their corresponding modes. The size of the last latent tensor $\mathcal{G}^{(K)}$ is $R_K \times I_K \times R_1$, that is, $R_{K+1} = R_1$, thus ensuring that the product of these matrices is a square matrix. $\boldsymbol{r} = [R_1, R_2, \cdots, R_K]^{\mathrm{T}}$ is known as the rank of the TR decomposition. For simplicity, we set $R_1 = R_2 = \cdots = R_K$ as a constant number. In Eq (1), $\mathcal{R}(i_1,i_2,\cdots,i_K)$ is equivalent to the trace of the sequential product of the matrix $\left\{\boldsymbol{G}^{(k)}(i_k)\right\}$. To further describe this concept, Eq (1) can be written in index form as

$$\mathcal{R}(i_1,i_2,\cdots,i_K) = \sum_{r_1,r_2,\ldots,r_K=1}^{R_1,R_2,\ldots,R_K}\prod_{k=1}^{K}\mathcal{G}^{(k)}(r_k,i_k,r_{k+1}), \tag{2}$$

where $\forall k \in \{1,\ldots,K\}, 1 \le r_k \le R_k, 1 \le i_k \le I_k$, k denotes the index of the tensor mode; r_k denotes the index of the latent mode; and i_k denotes the index of the data mode. Moreover, Eq (2) can also be expressed in tensor form as

$$\mathcal{R} = \sum_{r_1,r_2,\ldots,r_K=1}^{R_1,R_2,\ldots,R_K}\boldsymbol{g}^{(1)}(r_1,r_2)\circ\boldsymbol{g}^{(2)}(r_2,r_3)\circ\cdots\circ\boldsymbol{g}^{(K)}(r_K,r_1), \tag{3}$$

where $\circ$ denotes the outer product of vectors and $\boldsymbol{g}^{(k)}(r_k,r_{k+1}) \in \mathbb{R}^{I_k \times 1}$ denotes the (r_k,r_{k+1})-th mode-2 fiber of tensor $\mathcal{G}^{(k)}$. Additionally, Eq (3) indicates that the tensor can be decomposed into the sum of outer products of rank-1 tensors. The number of parameters in the TR representation is $\mathcal{O}\left(KIR^2\right)$, which is linearly related to the tensor order K.

Definition 1 (Reverse mode-k unfolding) [34]: The reverse-mode-k expansion matrix of $\mathcal{R}$ is denoted by $\boldsymbol{R}_{[k]}$ of size $I_k \times \prod_{j\neq k} I_j$, and its elements are defined as [34]

$$\boldsymbol{R}_{[k]}\left(i_k, \overline{i_{k+1}\cdots i_K i_1\cdots i_{k-1}}\right) = \mathcal{R}\left(i_1, i_2, \cdots, i_K\right). \tag{4}$$

Definition 2 (Standard mode-k unfolding): The mode-k unfolding matrix is denoted by $\boldsymbol{R}_{(k)}$ of size $I_k \times \prod_{j\neq k} I_j$, which is defined as

$$\boldsymbol{R}_{(k)}\left(i_k, \overline{i_1\cdots i_{k-1} i_{k+1}\cdots i_K}\right) = \mathcal{R}\left(i_1, i_2, \cdots, i_K\right), \tag{5}$$

where $\overline{i_1\cdots i_{k-1} i_{k+1}\cdots i_K}$ is defined as [47]

$$\overline{i_1\cdots i_{k-1} i_{k+1}\cdots i_K} = \begin{cases} \overline{i_2 i_3\cdots i_K} = i_2 + \left(i_3 - 1\right) I_2 + \left(i_4 - 1\right) I_2 I_3 + \cdots + \left(i_K - 1\right) I_2 I_3 \cdots I_{K-1}, \ k = 1; \\ \overline{i_1\cdots i_{k-1} i_{k+1}\cdots i_K} = i_1 + \left(i_2 - 1\right) I_1 + \left(i_3 - 1\right) I_1 I_2 + \cdots + \left(i_{k-1} - 1\right) \prod_{s=1}^{k-2} I_s + \sum_{s=k+1}^{K} \left(i_s - 1\right) \prod_{\substack{j=1 \\ j\neq k}}^{s-1} I_j, \ k \neq 1 \end{cases}. \tag{6}$$

Definition 3 (adjacent core factor merging sub-chains) [34]: Let $\mathcal{R} = \mathbf{TR}\left(\mathcal{G}^{(1)}, \mathcal{G}^{(2)}, \ldots, \mathcal{G}^{(K)}\right)$ be a TR decomposition. The TR cores $\mathcal{G}^{(k)}$ and $\mathcal{G}^{(k+1)}$ have the same mode size R_{k+1} and can be merged into a single sub-chain $\mathcal{G}^{(k,k+1)} \in \mathbb{R}^{R_k \times I_k I_{k+1} \times R_{k+2}}$ using a multilinear product. The lateral slice matrix of the merged core is expressed as

$$\boldsymbol{G}^{(k,k+1)}\left(\overline{i_k i_{k+1}}\right) = \boldsymbol{G}^{(k)}\left(i_k\right) \boldsymbol{G}^{(k+1)}\left(i_{k+1}\right). \tag{7}$$

Similarly, the sub-chain tensor obtained by combining all cores $\mathcal{G}^{(k)}$ but except the k-th core (that is, $\mathcal{G}^{(k+1)}, \ldots, \mathcal{G}^{(K)}, \mathcal{G}^{(1)}, \ldots, \mathcal{G}^{(k-1)}$) is denoted as $\mathcal{G}^{\neq k} \in \mathbb{R}^{R_{k+1} \times \prod_{j=1, j\neq k}^{K} I_j \times R_k}$, and its lateral slice matrix is defined as [34]

$$\boldsymbol{G}^{\neq k}\left(\overline{i_{k+1}\ldots i_K i_1 \ldots i_{k-1}}\right) = \prod_{j=k+1}^{K} \boldsymbol{G}^{(j)}\left(i_j\right) \prod_{t=1}^{k-1} \boldsymbol{G}^{(t)}\left(i_t\right). \tag{8}$$

2.3. Least squares algorithm for TR decomposition

In this section, we present a TR decomposition algorithm using alternating least squares (ALS). The ALS algorithm has been widely applied in various tensor decomposition models, including CP [48] and Tucker [49]. The main concept of ALS is to optimize one core while the other cores are fixed, and this process is repeated continuously until a certain convergence criterion is met. Given a fourth-order tensor $\mathcal{R}$, the goal is to optimize the core with a given TR rank $\boldsymbol{r}$, that is

$$\min_{\mathcal{G}^{(1:K)}} \left\| \mathcal{R} - \mathbf{TR}\left(\mathcal{G}^{(1:K)}\right) \right\|_2 . \tag{9}$$

Theorem 1: Given a TR decomposition $\mathcal{R} = \mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)$, its reverse-mode-$k$ unfolding matrix can be written as

$$\boldsymbol{R}_{[k]} = \boldsymbol{G}_{(2)}^{(k)} \left(\boldsymbol{G}_{[2]}^{\neq k}\right)^{\mathrm{T}} , \tag{10}$$

where $\mathcal{G}^{\neq k}$ is a sub-chain obtained by merging the $K-1$ cores, as defined in Eq (8).

Proof. According to the definition of TR in Eq (2), we obtain:

$$\begin{aligned}
\mathcal{R}(i_1,i_2,\cdots,i_K) &= \sum_{r_1,r_2,\ldots,r_K=1}^{R_1,R_2,\ldots,R_K} \mathcal{G}^{(1)}(r_1,i_1,r_2)\mathcal{G}^{(2)}(r_2,i_2,r_3)\cdots\mathcal{G}^{(K)}(r_K,i_K,r_1) \\
&= \sum_{r_k,r_{k+1}=1}^{R_k,R_{k+1}} \{\mathcal{G}^{(k)}(r_k,i_k,r_{k+1}) \sum_{\substack{r_1,\ldots,r_{k-1},\\ r_{k+2},\ldots,r_K=1}}^{\substack{R_1,\ldots,R_{k-1},\\ R_{k+2},\ldots,R_K}} \mathcal{G}^{(k+1)}(r_{k+1},i_{k+1},r_{k+2})\cdots\mathcal{G}^{(K)}(r_K,i_K,r_1)\mathcal{G}^{(1)}(r_1,i_1,r_2)\cdots\mathcal{G}^{(k-1)}(r_{k-1},i_{k-1},r_k)\} \\
&= \sum_{r_k,r_{k+1}=1}^{R_k,R_{k+1}} \{\mathcal{G}^{(k)}(r_k,i_k,r_{k+1})\mathcal{G}^{\neq k}\left(r_{k+1},\overline{i_{k+1}\ldots i_K i_1\ldots i_{k-1}},r_k\right)\}.
\end{aligned} \tag{11}$$

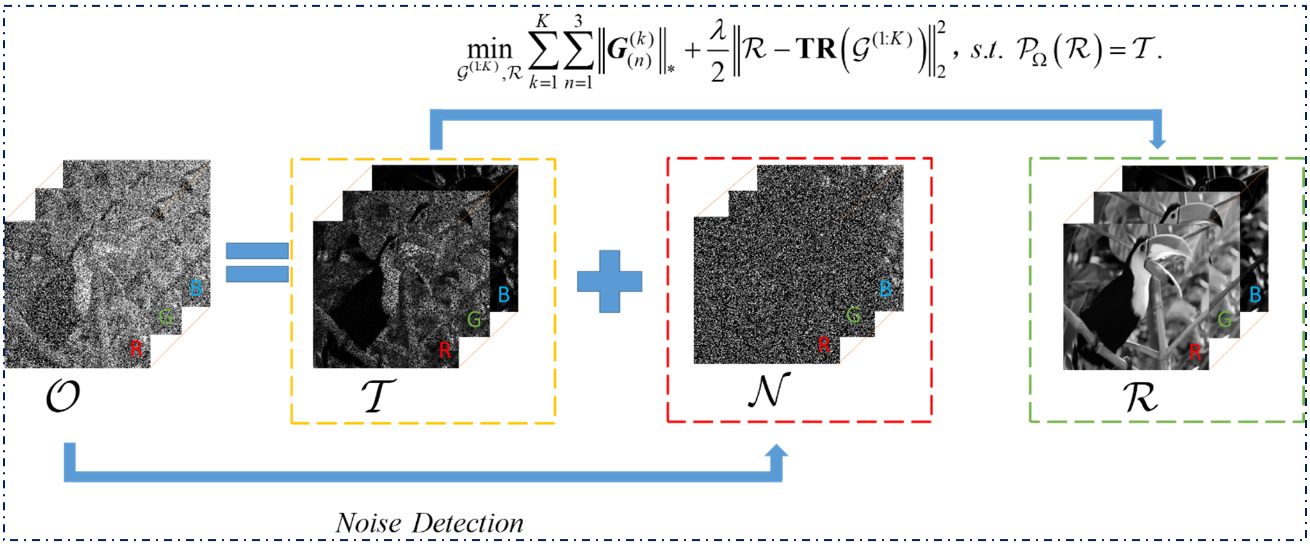
Therefore, the reverse-mode-k unfolding matrix of $\mathcal{R}$ can be expressed as:

$$\boldsymbol{R}_{[k]}\left(i_k,\overline{i_{k+1}\cdots i_K i_1\cdots i_{k-1}}\right) = \boldsymbol{G}_{(2)}^{(k)}(i_k,:)\left(\boldsymbol{G}_{[2]}^{\neq k}(\overline{i_{k+1}\cdots i_K i_1\cdots i_{k-1}},:)\right)^{\mathrm{T}} . \tag{12}$$

Then, Eq (10) holds. Therefore, the upper bound of the rank of each mode unfolding matrix of the tensor is constrained by the rank of the TR factors. That is, $rank\left(\boldsymbol{R}_{(k)}\right) = rank\left(\boldsymbol{R}_{[k]}\right) \leq rank(\boldsymbol{G}_{(2)}^{(k)})$.

3. The proposed method

In this section, we provide a detailed description of the proposed model-driven TR decomposition method. As shown in Figure 1, an image corrupted by impulsive noise (e.g., salt-and-pepper noise) is regarded as the sum of a missing data tensor and a noise tensor. Among them, the noise tensor is estimated after determining the locations of noise occurrences through noise detection. On this basis, we introduce the TR decomposition model to estimate the missing data. Subsequently, to restore the detailed textures of difficult-to-recover images, the FFDNET data-driven priors are introduced to explore the deep prior information of images, thereby enhancing the quality of the restored images. In this way, the advantages of model-driven TR decomposition and data-driven deep neural networks in image reconstruction are fully integrated.



$$\min_{\mathcal{G}^{(1:K)},\mathcal{R}} \sum_{k=1}^{K}\sum_{n=1}^{3}\left\|\boldsymbol{G}_{(n)}^{(k)}\right\|_* + \frac{\lambda}{2}\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_2^2,\ s.t.\ \mathcal{P}_\Omega(\mathcal{R})=\mathcal{T}.$$

Figure 1. Overview of the proposed method.

3.1. Tensor ring decomposition with data-driven model

The classical decomposition completion model can be described as follows:

$$\min_{\mathcal{R}} \sum_{k=1}^{K}\left\|\boldsymbol{R}_{(k)}\right\|_*,\ s.t.\ \mathcal{P}_\Omega(\mathcal{R})=\mathcal{T}, \tag{13}$$

where Ω is the index set; $\mathcal{P}_\Omega$ is a projection operator that preserves the entries in the index set while setting the entries in the complement of the index set to zero; $\mathcal{T}$ is the observed tensor, and $\|\cdot\|_*$ represents the nuclear norm matrix. Classical decomposition generally requires mode-k matricization, which destroys the intrinsic structure of an image to some extent. The low-rank structure of the internal tensor cannot be guaranteed, and the number of parameters is large. To address these limitations, we abandon the use of the classical tensor decomposition completion model in favor of the TR model. In this study, a low rank is imposed on each TR factor, whereby Eq (13) is expressed as

$$\min_{\mathcal{G}^{(1:K)},\mathcal{R}} \sum_{k=1}^{K}\left\|\boldsymbol{R}_{(k)}\right\|_* + \frac{\lambda}{2}\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_2^2, s.t.\ \mathcal{P}_\Omega(\mathcal{R})=\mathcal{T}, \tag{14}$$

where $\mathcal{G}^{(k)}$ denotes the k-th TR factor and **TR** denotes TR decomposition. λ is a balance parameter. Theorem 1 expounds on the relationship between the ranks of the tensor and TR. Thus, the upper bound rank of the mode-k unfolding matrix of the tensor $\mathcal{R}$ is determined to be the rank of the mode-2 unfolding matrix of corresponding core tensor $\mathcal{G}^{(k)}$. Therefore, the above model can be written as

$$\min_{\mathcal{G}^{(1:K)},\mathcal{R}} \sum_{k=1}^{K} \left\| \boldsymbol{G}_{(2)}^{(k)} \right\|_* + \frac{\lambda}{2} \left\| \mathcal{R} - \mathbf{TR}\left(\mathcal{G}^{(1:K)}\right) \right\|_2^2, \ s.t. \ \mathcal{P}_\Omega(\mathcal{R}) = \mathcal{T}. \tag{15}$$

In Eq (15), a tensor nuclear norm constraint is imposed on the mode-2 unfolding of each TR factor, to avoid destroying the image structure and reduce the computational complexity to a certain extent. In addition, we consider the imposition of low-rank constraints on the unfolding of the other two modes of the TR factor, that is, the TR factor is unfolded along modes 1 and 3, which can be expressed as $\sum_{k=1}^{K} \left\| \boldsymbol{G}_{(1)}^{(k)} \right\|_* + \sum_{k=1}^{K} \left\| \boldsymbol{G}_{(3)}^{(k)} \right\|_*$. When the model is optimized, the nuclear norm of the mode-n unfolding rank of the tensor and the fitting error of the approximate tensor are minimized simultaneously. This method ensures that the initial TR rank is used as the upper bound for the actual tensor rank, thereby enhancing the model's robustness when selecting the rank. The TR low-rank factor completion model is described as

$$\min_{\mathcal{G}^{(1:K)},\mathcal{R}} \sum_{k=1}^{K} \sum_{n=1}^{3} \left\| \boldsymbol{G}_{(n)}^{(k)} \right\|_* + \frac{\lambda}{2} \left\| \mathcal{R} - \mathbf{TR}\left(\mathcal{G}^{(1:K)}\right) \right\|_2^2, \ s.t. \ \mathcal{P}_\Omega(\mathcal{R}) = \mathcal{T}. \tag{16}$$

To recover the detailed image texture in the difficult-to-recover low-rank constraint terms in Eq (16), we introduced the FFDNET data-driven approach before exploring the deep prior information of an image, thereby improving the quality of the recovered image. As shown in Figure 1, the low-rank constraint of the TR decomposition is first utilized to retrieve the missing image, and then FFDNET is employed to recover the image's details. Therefore, the model can be rewritten as

$$\min_{\mathcal{G}^{(1:K)},\mathcal{R}} \sum_{k=1}^{K} \sum_{n=1}^{3} \left\| \boldsymbol{G}_{(n)}^{(k)} \right\|_* + \frac{\lambda}{2} \left\| \mathcal{R} - \mathbf{TR}\left(\mathcal{G}^{(1:K)}\right) \right\|_2^2 + \delta\psi(\mathcal{R}), \ s.t. \ \mathcal{P}_\Omega(\mathcal{R}) = \mathcal{T}, \tag{17}$$

where δ denotes the regularization parameter used to balance the low-rank model-driven $\sum_{k=1}^{K} \sum_{n=1}^{3} \left\| \boldsymbol{G}_{(n)}^{(k)} \right\|_* + \frac{\lambda}{2} \left\| \mathcal{R} - \mathbf{TR}\left(\mathcal{G}^{(1:K)}\right) \right\|_2^2$ and data-driven deep prior implemented by FFDNET. ψ is the implicit regularization term implemented by FFDNET, which plays the role of data-driven denoising by using a deep convolutional neural network with large-scale end-to-end training with massive data.

3.2. ADMM-based solver for the TR-DD model

In this section, the ADMM is used to solve Eq (17). First, an auxiliary variable $\mathcal{Q}$ is introduced, such that the model can be written as

$$\begin{gathered} \min_{[\mathcal{Q}],\mathcal{G}^{(1:K)},\mathcal{R}} \sum_{k=1}^{K} \sum_{n=1}^{3} \left\| \boldsymbol{Q}_{(n)}^{(k,n)} \right\|_* + \frac{\lambda}{2} \left\| \mathcal{R} - \mathbf{TR}\left(\mathcal{G}^{(1:K)}\right) \right\|_2^2 + \delta\psi(\mathcal{R}), \\ s.t. \ \boldsymbol{Q}_{(n)}^{(k,n)} = \boldsymbol{G}_{(n)}^{(k)}, k = 1, \ldots, K, n = 1,2,3, \mathcal{P}_\Omega(\mathcal{R}) = \mathcal{T}, \end{gathered} \tag{18}$$

where $[\mathcal{Q}] = \left\{\mathcal{Q}^{(k,n)}\right\}_{k=1,n=1}^{K,3}$ is an auxiliary variable of $\mathcal{G}^{(k)}$. Subsequently, the model is simplified and optimized as

$$
\begin{aligned}
& L\left(\mathcal{G}^{(1:K)},\mathcal{R},[\mathcal{Q}],[\mathcal{L}]\right) \\
& = \min_{\mathcal{G}^{(1:K)},\mathcal{R},[\mathcal{Q}],[\mathcal{L}]} \sum_{k=1}^{K}\sum_{n=1}^{3}\left(\left\|\boldsymbol{Q}_{(n)}^{(k,n)}\right\|_{*}+\left\langle\mathcal{L}^{(k,n)},\mathcal{Q}^{(k,n)}-\mathcal{G}^{(k)}\right\rangle+\frac{\mu}{2}\left\|\mathcal{Q}^{(k,n)}-\mathcal{G}^{(k)}\right\|_{2}^{2}\right)+\frac{\lambda}{2}\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_{F}^{2}+\delta\psi(\mathcal{R}) \quad (19) \\
& s.t.\ \mathcal{P}_{\Omega}(\mathcal{R})=\mathcal{T},
\end{aligned}
$$

where $[\mathcal{L}]=\left\{\mathcal{L}^{(k,n)}\right\}_{k=1,n=1}^{K,3}$ denotes the Lagrange multiplier and $\mu>0$ denotes the quadratic penalty parameter. As $k=1,2,\ldots,K$, $n=1,2,3$, $\mathcal{G}^{(k)}$, $\mathcal{Q}^{(k,n)}$, and $\mathcal{L}^{(k,n)}$ are independent of each other. They can be solved using ADMM.

1) The $\mathcal{G}^{(k)}$ subproblem is expressed as

$$
J_{\mathcal{G}^{(k)}}=\min_{\mathcal{G}^{(k)}}\sum_{n=1}^{3}\frac{\mu}{2}\left\|\mathcal{Q}^{(k,n)}-\mathcal{G}^{(k)}+\frac{1}{\mu}\mathcal{L}^{(k,n)}\right\|_{2}^{2}+\frac{\lambda}{2}\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_{2}^{2}+\delta\psi(\mathcal{R})+C_{\mathcal{G}^{(k)}}, \quad (20)
$$

where $C_{\mathcal{G}^{(k)}}$ is the other part of the Lagrange function, unrelated to $\mathcal{G}^{(k)}$, that is, the constant term. According to Theorem 1, Eq (20) can be simplified first and then written in matrix form, as

$$
J_{\boldsymbol{G}_{(2)}^{(k)}}=\min_{\boldsymbol{G}_{(2)}^{(k)}}\sum_{n=1}^{3}\frac{\mu}{2}\left\|\boldsymbol{Q}_{(2)}^{(k,n)}-\boldsymbol{G}_{(2)}^{(k)}+\frac{1}{\mu}\boldsymbol{L}_{(2)}^{(k,n)}\right\|_{2}^{2}+\frac{\lambda}{2}\left\|\boldsymbol{R}_{[k]}-\boldsymbol{G}_{(2)}^{(k)}\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\right\|_{2}^{2}+C_{\mathcal{G}^{(k)}}. \quad (21)
$$

To simplify and optimize Eq (21), both sides of the equation are transposed simultaneously as

$$
J_{\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}}=\min_{\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}}\sum_{n=1}^{3}\frac{\mu}{2}\left\|\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}-\left(\boldsymbol{Q}_{(2)}^{(k,n)}\right)^{\mathrm{T}}-\frac{1}{\mu}\left(\boldsymbol{L}_{(2)}^{(k,n)}\right)^{\mathrm{T}}\right\|_{2}^{2}+\frac{\lambda}{2}\left\|\boldsymbol{G}_{[2]}^{(\neq k)}\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}-\left(\boldsymbol{R}_{[k]}\right)^{\mathrm{T}}\right\|_{2}^{2}+C_{\mathcal{G}^{(k)}}. \quad (22)
$$

Letting $\dfrac{\partial\left(J_{\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}}\right)}{\partial\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}}=\mathbf{0}$, we obtain

$$
\sum_{n=1}^{3}\mu\left[\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}-\left(\boldsymbol{Q}_{(2)}^{(k,n)}\right)^{\mathrm{T}}-\frac{1}{\mu}\left(\boldsymbol{L}_{(2)}^{(k,n)}\right)^{\mathrm{T}}\right]+\lambda\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\left[\boldsymbol{G}_{[2]}^{(\neq k)}\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}-\left(\boldsymbol{R}_{[k]}\right)^{\mathrm{T}}\right]=\mathbf{0}. \quad (23)
$$

Subsequently, to optimize Eq (23), such terms are combined to yield

$$
\sum_{n=1}^{3}\mu\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}+\lambda\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\boldsymbol{G}_{[2]}^{(\neq k)}\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{\mathrm{T}}=\mu\sum_{n=1}^{3}\left[\left(\boldsymbol{Q}_{(2)}^{(k,n)}\right)^{\mathrm{T}}+\frac{1}{\mu}\left(\boldsymbol{L}_{(2)}^{(k,n)}\right)^{\mathrm{T}}\right]+\lambda\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)\left(\boldsymbol{R}_{[k]}\right)^{\mathrm{T}}. \quad (24)
$$

Thereafter, transposing both sides of Eq (24) yields

$$\sum_{n=1}^{3}\mu \boldsymbol{G}_{(2)}^{(k)}+\lambda \boldsymbol{G}_{(2)}^{(k)}\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\boldsymbol{G}_{[2]}^{(\neq k)}=\mu\sum_{n=1}^{3}\left[\boldsymbol{Q}_{(2)}^{(k,n)}+\frac{1}{\mu}\boldsymbol{L}_{(2)}^{(k,n)}\right]+\lambda \boldsymbol{R}_{[k]}\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}. \tag{25}$$

Then, we have

$$\left(\boldsymbol{G}_{(2)}^{(k)}\right)^{+}=\left[\mu\sum_{n=1}^{3}\left[\boldsymbol{Q}_{(2)}^{(k,n)}+\frac{1}{\mu}\boldsymbol{L}_{(2)}^{(k,n)}\right]+\lambda \boldsymbol{R}_{[k]}\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\right]\left[\sum_{n=1}^{3}\mu \boldsymbol{I}+\lambda\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\boldsymbol{G}_{[2]}^{(\neq k)}\right]^{-1}, \tag{26}$$

where $\boldsymbol{I}$ denotes the identity matrix.

Finally, the matrix form is restacked into a tensor form to obtain

$$\left(\mathcal{G}^{(k)}\right)^{+}=\mathbf{fold}_2\left\{\left[\mu\sum_{n=1}^{3}\left[\boldsymbol{Q}_{(2)}^{(k,n)}+\frac{1}{\mu}\boldsymbol{L}_{(2)}^{(k,n)}\right]+\lambda \boldsymbol{R}_{[k]}\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\right]\left[\sum_{n=1}^{3}\mu \boldsymbol{I}+\lambda\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\boldsymbol{G}_{[2]}^{(\neq k)}\right]^{-1}\right\}, \tag{27}$$

where $\mathbf{fold}_2$ is a mode-2 fold operator that transforms the mode-2 unfolding matrix into its tensor form.

2) The $\mathcal{Q}^{(k,n)}$ subproblem is expressed as

$$J_{\mathcal{Q}^{(k,n)}}=\min_{\mathcal{Q}^{(k,n)}}\sum_{n=1}^{3}\left(\frac{\mu}{2}\left\|\mathcal{Q}^{(k,n)}-\mathcal{G}^{(k)}+\frac{1}{\mu}\mathcal{L}^{(k,n)}\right\|_2^2+\left\|\boldsymbol{Q}_{(n)}^{(k,n)}\right\|_*+C_{\mathcal{Q}^{(k,n)}}\right). \tag{28}$$

The objective function in Eq (28) contains three subproblems ($\mathcal{Q}^{(k,1)}$, $\mathcal{Q}^{(k,2)}$, $\mathcal{Q}^{(k,3)}$), and each can be described as

$$J_{\mathcal{Q}^{(k,n)}}=\min_{\mathcal{Q}^{(k,n)}}\frac{\mu}{2}\left\|\mathcal{Q}^{(k,n)}-\mathcal{G}^{(k)}+\frac{1}{\mu}\mathcal{L}^{(k,n)}\right\|_2^2+\left\|\boldsymbol{Q}_{(n)}^{(k,n)}\right\|_*+C_{\mathcal{Q}^{(k,n)}}. \tag{29}$$

First, rewriting Eq (29) in matrix form as

$$J_{\boldsymbol{Q}^{(k,n)}}=\min_{\mathcal{Q}^{(k,n)}}\frac{\mu}{2}\left\|\boldsymbol{Q}_{(n)}^{(k,n)}-\boldsymbol{G}_{(n)}^{(k)}+\frac{1}{\mu}\boldsymbol{L}_{(n)}^{(k,n)}\right\|_2^2+\left\|\boldsymbol{Q}_{(n)}^{(k,n)}\right\|_*+C_{\mathcal{Q}^{(k,n)}}. \tag{30}$$

A singular value threshold is used to solve Eq (30), and its solution is:

$$\left(\mathcal{Q}^{(k,n)}\right)^{+}=\mathbf{fold}_n\left(\mathbf{D}_{\frac{1}{\mu}}\left(\boldsymbol{G}_{(n)}^{(k)}-\frac{1}{\mu}\boldsymbol{L}_{(n)}^{(k,n)}\right)\right), \tag{31}$$

where $\mathbf{D}_{\beta}(\boldsymbol{M})=\mathbf{D}_{\beta}(\boldsymbol{USV}^{\mathrm{H}})=\boldsymbol{U}(\max(\boldsymbol{S}-\beta,0))\boldsymbol{V}^{\mathrm{H}}$ denotes the singular value threshold operation. β is a thresholding value. $\mathbf{fold}_n$ is a mode-$n(n=1,2,3)$ fold operator that transforms the mode-$n(n=1,2,3)$ unfolding matrix into its tensor form.

3) The $\mathcal{R}$ subproblem is expressed as

$$\mathcal{R}=\arg\min_{\mathcal{R}}\frac{\lambda}{2}\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_2^2+\delta\psi(\mathcal{R})+C_{\mathcal{R}}, \tag{32}$$

and Eq (32) is simplified to:

$$\mathcal{R}=\arg\min_{\mathcal{R}}\frac{\lambda}{2\delta}\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_2^2+\psi(\mathcal{R})+C_{\mathcal{R}}. \tag{33}$$

Let $\phi=\sqrt{\dfrac{\delta}{\lambda}}$, then Eq (33) can be written as:

$$\mathcal{R}=\arg\min_{\mathcal{R}}=\frac{1}{2\phi^2}\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_2^2+\psi(\mathcal{R})+C_{\mathcal{R}}. \tag{34}$$

The forward–backward splitting algorithm is used to optimize and solve Eq (34), dividing the process into two steps.

To update the forward subproblem, $\frac{1}{2\phi^2}\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_2^2$ is determined based on the observed values of the corresponding index. Subsequently, the updated TR factor is used to approximate the missing terms, yielding

$$\left(\mathcal{R}^{\left(\frac{1}{2}\right)}\right)^{+}=\mathcal{P}_{\Omega}(\mathcal{T})+\mathcal{P}_{\Omega^c}\left(\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right), \tag{35}$$

where $\left(\mathcal{R}^{\left(\frac{1}{2}\right)}\right)^{+}$ denotes the update of the forward subproblem of $\mathcal{R}$, and Ω^c denotes the complement of Ω, representing the missing index values, subsequently, according to the updated $\left(\mathcal{R}^{\left(\frac{1}{2}\right)}\right)^{+}$ of forward term.

The updated forward $\left(\mathcal{R}^{\left(\frac{1}{2}\right)}\right)^{+}$ problem uses a plug-and-play framework, specifically FFDNet. The restored image is calculated as

$$\mathcal{R}^{+}=\mathcal{P}_{\Omega^{c}}\left(\mathbf{FFDNet}\left(\left(\mathcal{R}^{\left(\frac{1}{2}\right)}\right)^{+},\phi\right)\right)+\mathcal{T}, \tag{36}$$

where **FFDNet** is a denoising neural network driven by the data $\mathcal{R}$, and ϕ is a model hyperparameter.

Algorithm 1. The TR-DD method.

```
Input: the tensor $\mathcal{T}$ to be completed and the index set $\Omega$; initialize the TR rank $\{R_k\}_{k=1}^{K}$.
Initialization: the random sample $\mathcal{G}^{(k)}$ follows the normal distribution with a mean of zero and standard deviation as 1; $\mathcal{L}^{(k,n)}=\mathbf{0}$; $\mathcal{Q}^{(k,n)}=\mathbf{0}$; $\lambda=5$; $\mu^{0}=1,\ \mu_{\max}=100$; $\rho=1.5$.
For $t=1:t_{\max}$ do
    If $\frac{\left\|\mathcal{R}^{+}-\mathcal{R}\right\|_2}{\left\|\mathcal{R}^{+}\right\|_2}<10^{-5}$
        Break
    Else
    $\mathcal{G}_k^{+}\leftarrow\mathbf{fold}_2\left\{\left[\mu\sum_{n=1}^{3}\left[\boldsymbol{Q}_{(2)}^{(k,n)}+\frac{1}{\mu}\boldsymbol{L}_{(2)}^{(k,n)}\right]+\lambda\boldsymbol{R}_{[k]}\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\right]\left[\sum_{n=1}^{3}\mu\boldsymbol{I}+\lambda\left(\boldsymbol{G}_{[2]}^{(\neq k)}\right)^{\mathrm{T}}\boldsymbol{G}_{[2]}^{(\neq k)}\right]^{-1}\right\}.$
    $\left(\mathcal{Q}^{(k,n)}\right)^{+}\leftarrow\mathbf{fold}_n\left(\mathbf{D}_{\frac{1}{\mu}}\left(\boldsymbol{G}_{(n)}^{(k)}-\frac{1}{\mu}\boldsymbol{L}_{(n)}^{(k,n)}\right)\right).$
    The $\mathcal{R}$ subproblem update is divided into two steps
        Step 1: $\left(\mathcal{R}^{\left(\frac{1}{2}\right)}\right)^{+}\leftarrow\mathcal{P}_{\Omega}(\mathcal{T})+\mathcal{P}_{\Omega^{c}}\left(\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right),$
        Step 2: $\mathcal{R}^{+}\leftarrow\mathcal{P}_{\Omega^{c}}\left(\mathbf{FFDNet}\left(\left(\mathcal{R}^{\left(\frac{1}{2}\right)}\right)^{+},\phi\right)\right)+\mathcal{T}.$
    $\left(\mathcal{L}^{(k,n)}\right)^{+}\leftarrow\left[\mathcal{L}^{(k,n)}+\mu\left(\mathcal{Q}^{(k,n)}-\mathcal{G}^{(k)}\right)\right].$
    $\mu^{+}=\min\left(\rho\mu,\mu_{\max}\right)$
    End If
End For
Return $\mathcal{R}$
```

4) The $\mathcal{L}^{(k,n)}$ subproblem can be written as follows

$$\mathcal{L}^{(k,n)} = \underset{\mathcal{L}^{(k,n)}}{\arg\min} \frac{\mu}{2} \left\| \mathcal{Q}^{(k,n)} - \mathcal{G}_k + \frac{1}{\mu} \mathcal{L}^{(k,n)} \right\|_2^2 + C_{\mathcal{L}}, \tag{37}$$

where the Lagrange multiplier $\mathcal{L}^{(k,n)}$ is updated to

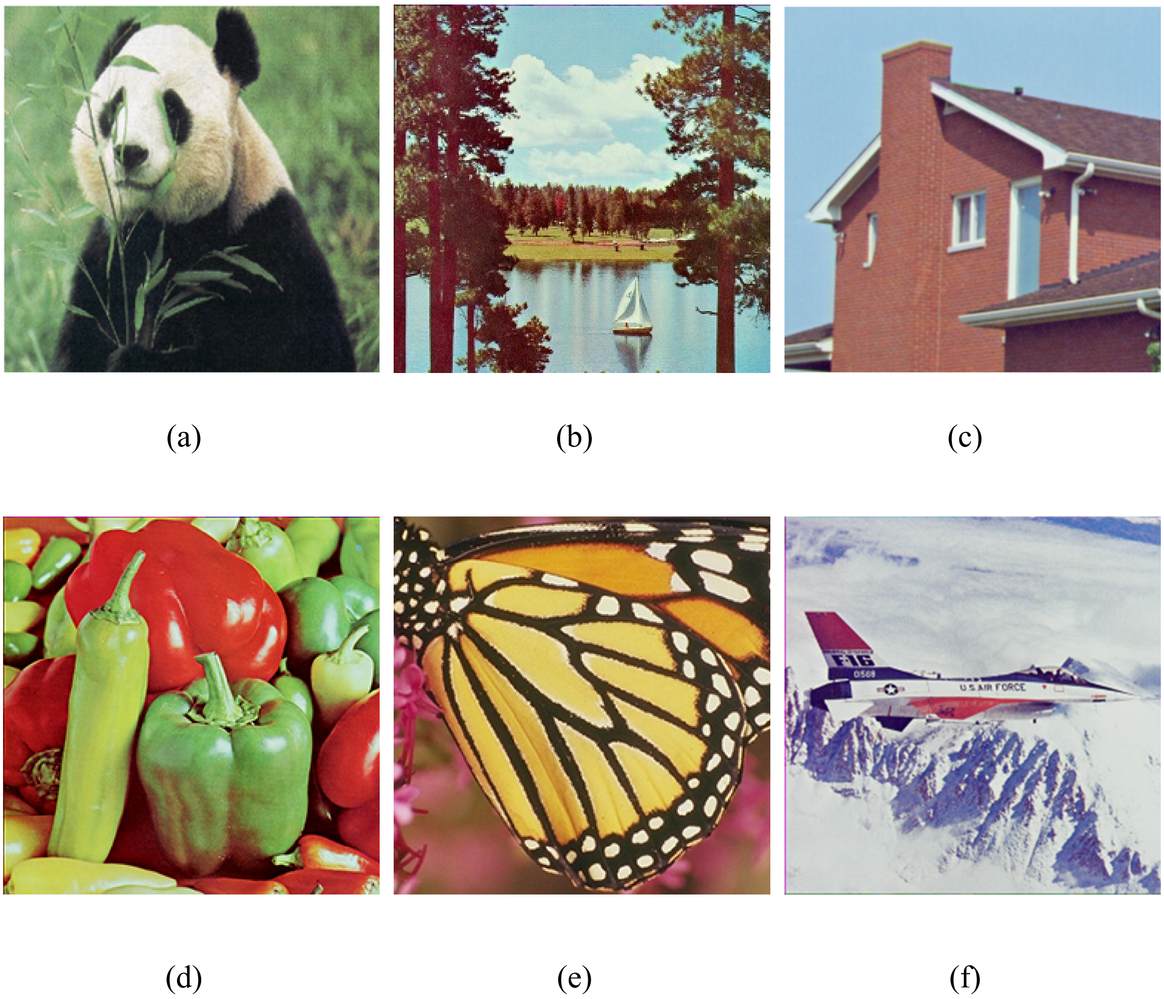
$$\left(\mathcal{L}^{(k,n)}\right)^+ = \mathcal{L}^{(k,n)} + \mu\left(\mathcal{Q}^{(k,n)} - \mathcal{G}^{(k)}\right). \tag{38}$$

Because the μ parameter limits the penalty term of the Lagrangian function, μ is updated using $\mu^+ = \min\left(\rho\mu, \mu_{\max}\right)$ and ρ is one of the tuning hyperparameters.

In summary, the ADMM-based solution is updated iteratively. In addition, we consider two optimization stopping conditions for the model settings: 1) The maximum number of iterations $k_{\max}$ and 2) the difference between two iterations (i.e., $\frac{\left\|\mathcal{R}^+ - \mathcal{R}\right\|_2}{\left\|\mathcal{R}^+\right\|_2} < 10^{-5}$). Finally, the proposed method is summarized in Algorithm 1.

4. Experiments

In this section, we verify the performance of the proposed TR-DD model and compare it with tensor decomposition models based on different tensor ranks, namely the TMac [50] model based on Tucker rank, the transformer neural network (TNN) [51] model based on tube rank, 3DlogTNN [29] model based on fiber rank, and tensor ring with low-rank factor (TRLRF) [52] and tensor ring decomposition-based gradient factor regularization (TR-GFR) [36] models based on the TR rank. We first conduct color image completion experiments on the color images with different missing rates and then applied the model to a color image graffiti removal application. Finally, the parameter sensitivity of the proposed model is analyzed. To visually present the experimental results, the peak signal-to-noise ratio and structural similarity index are used to evaluate model performance. To test the model's completion ability for different images, six 256 × 256 test images are selected, as shown in Figure 2.



(a) (b) (c)

(d) (e) (f)

Figure 2. Test pictures. (a) Panda; (b) sailboat; (c) house; (d) peppers; (e) butterfly; and (f) airplane.

4.1. Color image completion

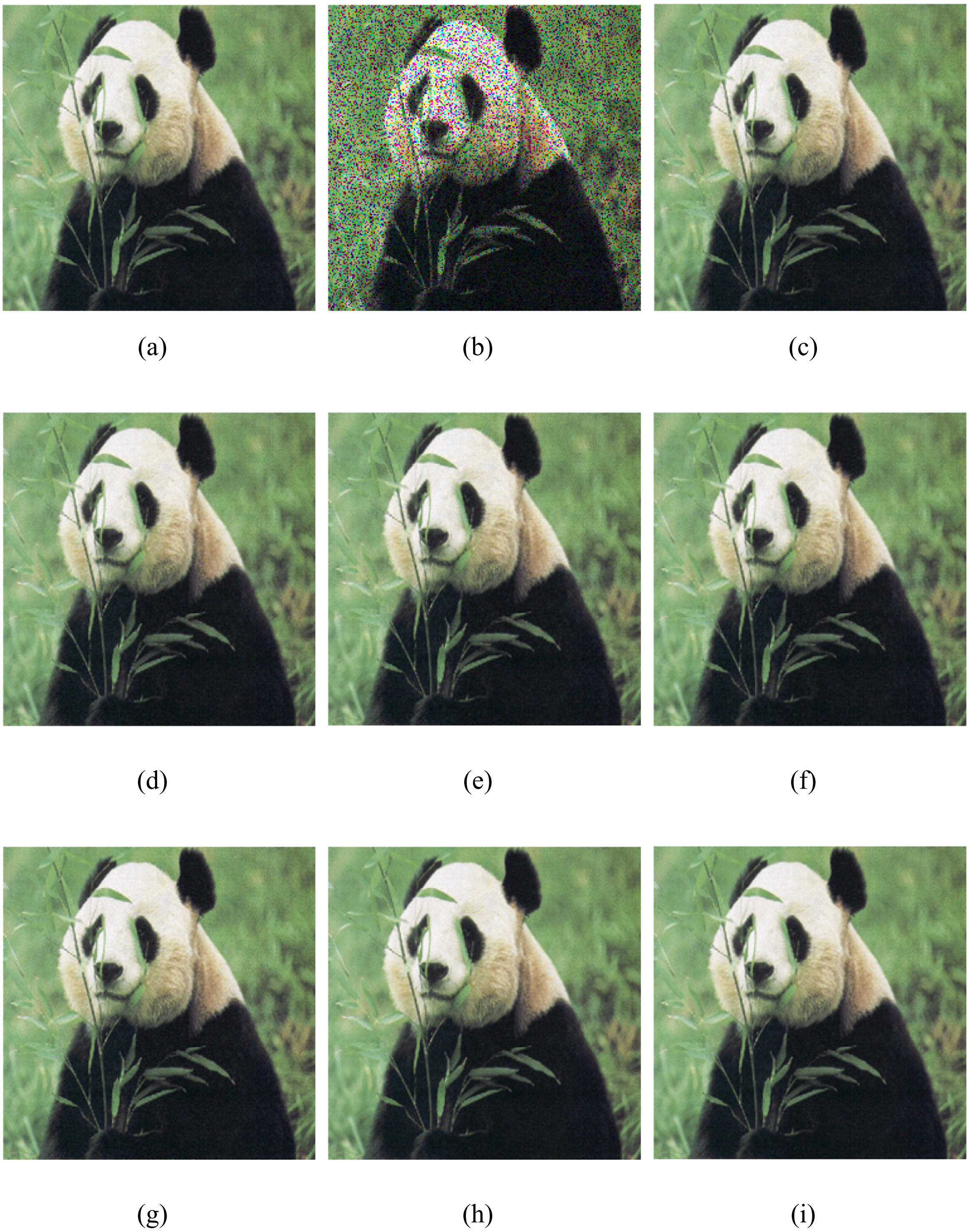
The proposed method is experimentally evaluated under different scenarios. The experiments are conducted on a desktop computer equipped with 32 GB of RAM and an Intel Xeon(R) CPU (clock frequency: 3.5 GHz), utilizing MATLAB 2017b. To evaluate the performance of the proposed model, partial image data are randomly deleted from six test images. Color image completion is performed at a missing rate of 20%–80%. The proportion of missing data to the total data is referred to as the missing rate. The experimental results are summarized in Tables 1 and 2, where bold values indicate the highest values.

Table 1. PSNR values of color image completion calculated by different methods.

Image	Method	Missing rate						
		20%	30%	40%	50%	60%	70%	80%
Panda	TMac	41.710	38.786	36.147	33.949	31.583	29.073	25.562
	TNN	42.344	38.973	36.188	33.759	31.167	28.487	25.581
	TRLRF	42.143	39.285	36.577	34.450	32.031	29.323	26.784
	TR-GFR	42.423	40.168	37.885	35.980	33.584	31.824	29.367
	3DLogTNN	**46.217**	**43.073**	**40.454**	**38.001**	35.758	33.461	30.847
	TMac_DP3	43.458	41.021	39.022	37.437	35.596	33.835	31.743
	TR-DD	43.665	41.221	39.251	37.709	**35.790**	**34.016**	**31.768**
Sailboat	TMac	34.584	31.881	29.294	27.011	24.907	23.114	20.902
	TNN	39.984	34.76	30.77	28.007	25.387	23.16	20.825
	TRLRF	40.086	35.881	32.790	30.107	26.505	24.201	21.801
	TR-GFR	41.755	35.636	33.095	30.703	28.173	25.982	23.87
	3DLogTNN	**44.161**	**39.794**	**35.426**	**32.616**	**30.178**	27.533	24.796
	TMac_DP3	40.202	36.770	34.070	31.815	29.659	27.591	25.244
	TR-DD	42.068	37.982	34.876	32.491	30.157	**27.873**	**25.505**
House	TMac	43.741	40.617	38.283	35.955	33.729	30.614	26.558
	TNN	43.681	40.533	37.879	35.395	32.776	29.755	26.425
	TRLRF	41.501	39.108	38.864	36.877	34.658	31.440	27.517
	TR-GFR	**46.306**	**43.742**	**41.711**	**39.588**	**37.643**	**35.079**	32.239
	3DLogTNN	45.903	43.289	40.787	38.62	36.22	34.238	31.239
	TMac_DP3	41.919	39.765	37.188	36.237	35.110	34.333	32.817
	TR-DD	42.736	40.556	39.027	37.073	35.854	34.705	**33.124**
Peppers	TMac	42.616	38.221	35.339	32.746	30.345	27.864	25.175
	TNN	42.006	37.313	33.622	31.074	28.683	26.242	23.750
	TRLRF	43.606	39.578	36.100	33.567	31.138	28.635	25.728
	TR-GFR	43.644	39.900	37.82	35.034	33.337	31.093	28.890
	3DLogTNN	43.133	39.651	37.098	35.169	33.092	30.953	28.677
	TMac_DP3	43.793	40.374	38.200	36.369	34.895	33.211	31.412
	TR-DD	**44.485**	**40.826**	**38.590**	**36.760**	**35.264**	**33.663**	**31.692**
Butterfly	TMac	38.521	35.603	32.699	28.949	25.535	22.008	18.546
	TNN	42.553	36.811	32.227	28.114	24.522	21.310	18.115
	TRLRF	43.064	39.238	34.313	31.012	27.028	24.102	20.785
	TR-GFR	43.569	40.246	36.537	33.137	30.364	27.232	23.439
	3DLogTNN	**49.718**	**45.296**	**41.887**	**38.315**	34.802	31.141	26.861
	TMac_DP3	44.310	41.365	38.752	36.521	34.343	31.929	28.862
	TR-DD	45.552	42.240	39.673	37.465	**35.054**	**32.520**	**29.437**
Airplane	TMac	36.135	33.486	31.01	28.638	26.31	24.11	21.116
	TNN	37.025	33.807	30.964	28.522	26.313	24.144	21.884
	TRLRF	40.251	36.642	33.684	30.778	27.996	26.012	23.014
	TR-GFR	**41.441**	**38.212**	35.088	32.126	30.191	27.585	24.772
	3DLogTNN	41.371	37.804	34.892	32.393	29.946	27.856	25.299
	TMac_DP3	40.235	37.298	35.159	32.981	30.936	29.032	26.781
	TR-DD	40.846	37.810	**35.778**	**33.802**	**31.724**	**29.432**	**27.215**

Table 2. SSIM values for color image completion calculated by different methods.

Image	Method	Missing rate						
		20%	30%	40%	50%	60%	70%	80%
Panda	TMac	0.980	0.964	0.937	0.902	0.851	0.777	0.636
	TNN	0.986	0.972	0.951	0.919	0.869	0.790	0.671
	TRLRF	0.986	0.971	0.949	0.921	0.877	0.799	0.696
	TR-GFR	0.987	0.978	0.964	0.946	0.912	0.878	0.811
	3DLogTNN	**0.994**	**0.988**	**0.981**	**0.969**	**0.951**	**0.925**	**0.877**
	TMac_DP3	0.986	0.977	0.964	0.950	0.928	0.902	0.861
	TR-DD	0.987	0.978	0.967	0.953	0.932	0.909	0.868
Sailboat	TMac	0.952	0.918	0.868	0.804	0.730	0.649	0.502
	TNN	0.990	0.968	0.926	0.871	0.791	0.694	0.500
	TRLRF	0.990	0.971	0.930	0.905	0.816	0.726	0.604
	TR-GFR	0.993	0.977	0.959	0.933	0.885	0.820	0.766
	3DLogTNN	**0.998**	**0.994**	**0.985**	**0.957**	**0.948**	0.892	0.802
	TMac_DP3	0.990	0.981	0.969	0.951	0.926	0.892	0.832
	TR-DD	0.995	0.984	0.972	0.956	0.933	**0.897**	**0.845**
House	TMac	0.983	0.967	0.947	0.917	0.88	0.816	0.697
	TNN	0.982	0.965	0.932	0.895	0.843	0.785	0.689
	TRLRF	0.982	0.969	0.969	0.956	0.926	0.848	0.739
	TR-GFR	**0.992**	**0.986**	**0.978**	**0.968**	**0.953**	**0.922**	0.877
	3DLogTNN	0.992	0.986	0.978	0.966	0.949	0.921	0.870
	TMac_DP3	0.988	0.979	0.967	0.953	0.938	0.918	0.894
	TR-DD	0.989	0.980	0.970	0.957	0.943	0.920	**0.900**
Peppers	TMac	0.966	0.940	0.900	0.850	0.787	0.685	0.535
	TNN	0.958	0.928	0.898	0.844	0.760	0.650	0.495
	TRLRF	0.978	0.953	0.927	0.892	0.828	0.710	0.601
	TR-GFR	0.987	0.978	0.962	0.944	0.915	0.865	0.778
	3DLogTNN	0.989	0.981	0.971	0.956	0.932	0.898	0.832
	TMac_DP3	0.989	0.983	0.975	0.966	0.952	0.934	0.900
	TR-DD	**0.990**	**0.983**	**0.977**	**0.967**	**0.955**	**0.942**	**0.906**
Butterfly	TMac	0.972	0.952	0.916	0.85	0.761	0.636	0.471
	TNN	0.992	0.974	0.915	0.842	0.752	0.629	0.465
	TRLRF	0.992	0.982	0.952	0.911	0.825	0.723	0.625
	TR-GFR	0.995	0.989	0.977	0.955	0.926	0.864	0.763
	3DLogTNN	**0.999**	**0.998**	**0.996**	**0.992**	0.972	0.956	0.875
	TMac_DP3	0.996	0.993	0.989	0.984	0.976	0.964	0.939
	TR-DD	0.996	0.994	0.990	0.986	**0.979**	**0.969**	**0.948**
Airplane	TMac	0.955	0.924	0.883	0.826	0.742	0.674	0.524
	TNN	0.977	0.955	0.880	0.820	0.749	0.677	0.604
	TRLRF	0.985	0.967	0.940	0.902	0.844	0.774	0.652
	TR-GFR	**0.994**	**0.986**	0.972	0.954	0.910	0.861	0.774
	3DLogTNN	0.994	0.988	0.974	0.964	0.945	0.876	0.815
	TMac_DP3	0.992	0.983	0.973	0.960	0.940	0.915	0.868
	TR-DD	0.994	0.984	**0.975**	**0.964**	**0.947**	**0.922**	**0.882**



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3. “Panda” image with a 20% missing rate. (a) Original image; (b) noisy image; (c) TMac; (d) TNN; (e) TRLRF; (f) TR-GFR; (g) 3DlogTNN; (h) TMac_DP3; and (i) TR-DD.

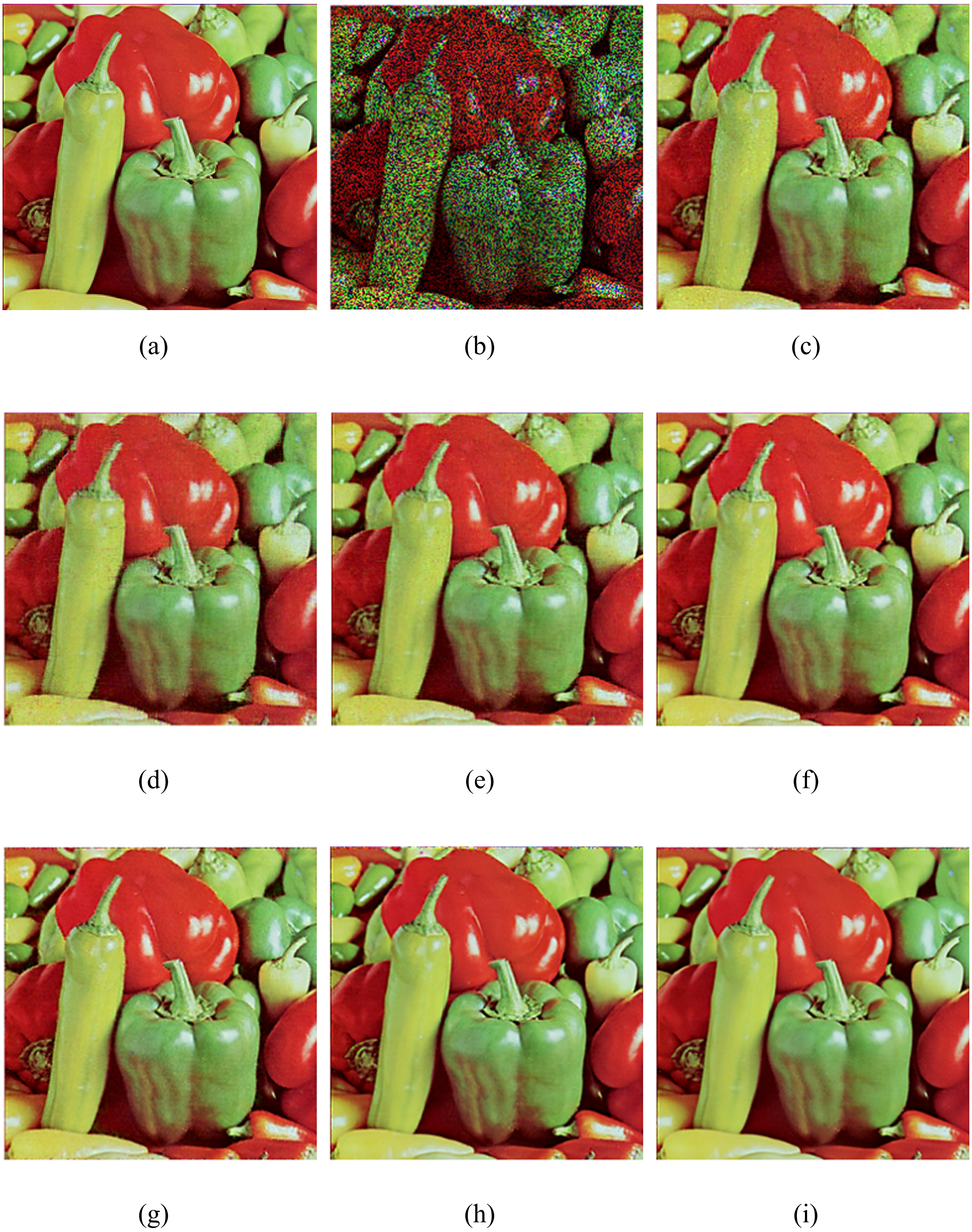
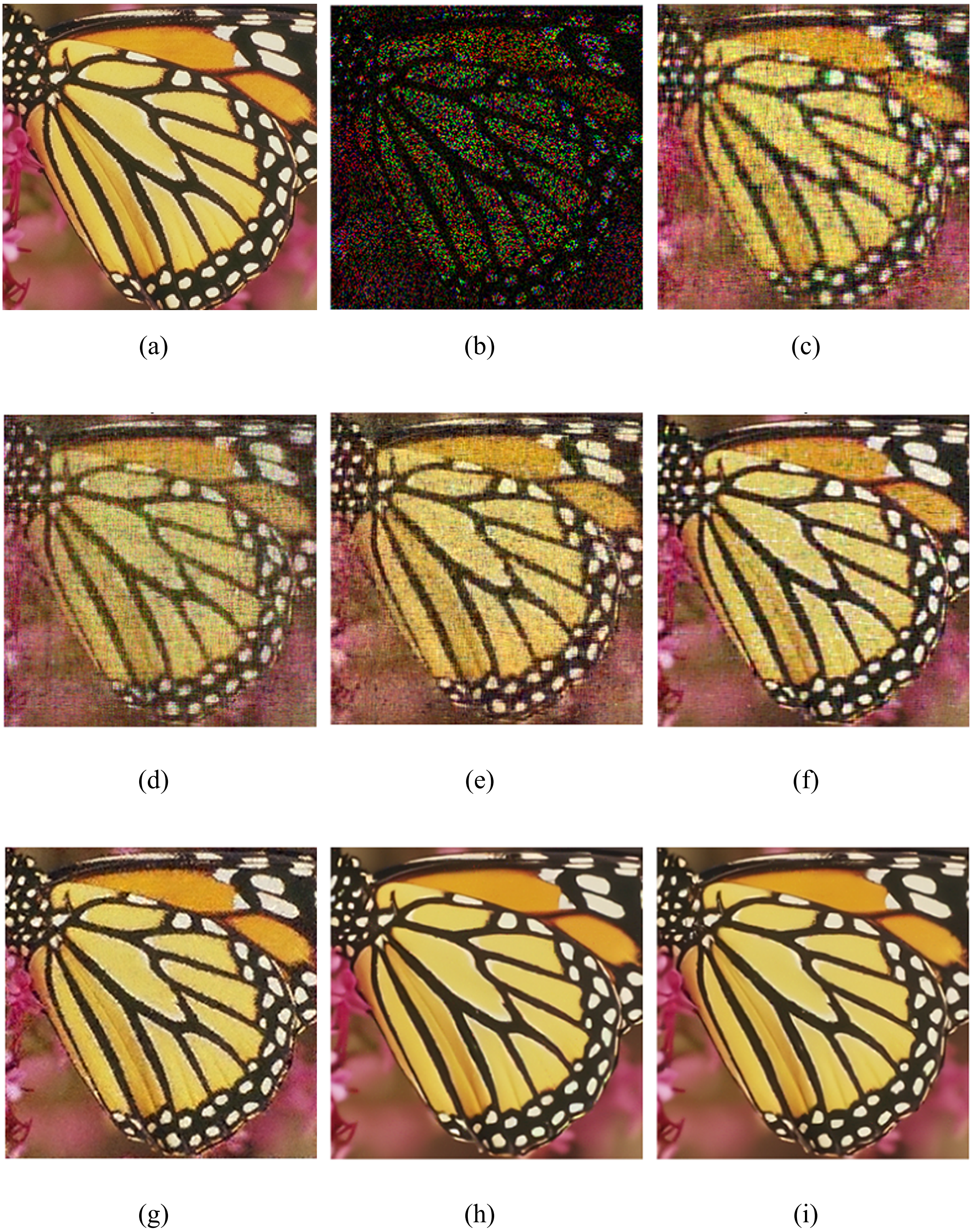


Figure 4. “Peppers” image with a 50% missing rate. (a) Original image; (b) Noisy image; (c) TMac; (d) TNN; (e) TRLRF; (f) TR-GFR; (g) 3DlogTNN; (h) TMac_DP3; and (i) TR-DD.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5. "Butterfly" image with an 80% missing rate. (a) Original image; (b) Noisy image; (c) TMac; (d) TNN; (e) TRLRF; (f) TR-GFR; (g) 3DlogTNN; (h) TMac_DP3; and (i) TR-DD.

As summarized in Table 1, the performance of the TR-DD completion model is the best, and its peak-to-signal noise ratio (PSNR) is significantly higher than that of the TMac and TR models. Compared with the TMac_DP3 model, the performance of the TR-DD model also improved, which is attributed to the effective retention of the internal structure of the image by the TR decomposition factors. For the TR-GFR model based on TR decomposition, TR-DD achieved good completion results because the deep prior recovered detailed texture that is difficult to capture using low-rank constraints. In the case of missing rates in Figure 3, all the models recovered clear images overall, with salient details. However, when the missing rate increased, as shown in Figure 4, in the "Peppers" image with a 50% missing rate, the TMac, TNN, and TRLFR models fail to recover pixels, resulting in blurred images and missing details. Although the overall recovery effects of the TR-GFR and 3DLogTNN models are good, the recovered images are smoother than those of the TMac_DP3 and TR-DD models using DL. In the "Butterfly" image, which has an 80% missing rate, the TMac_DP3 and TR-DD models can recover the image texture information, as shown in Figure 5. Numerous artifacts appear in the TMac, TNN, and TRLFR models, and the background details are lost. The recovery effect of TR-GFR is poor, and white dots appear in some images due to missing data. Although the overall image recovery of the 3DLogTNN model is good, some details are still missing on the butterfly wings, and numerous noise speckles are present in the images. The images recovered by the TMac_DP3 and TR-DD models not only contain salient information but are also smooth. In general, the performance of the TR-DD model is superior to that of the TMac_DP3 model, yielding satisfactory completion results even for high missing rates.

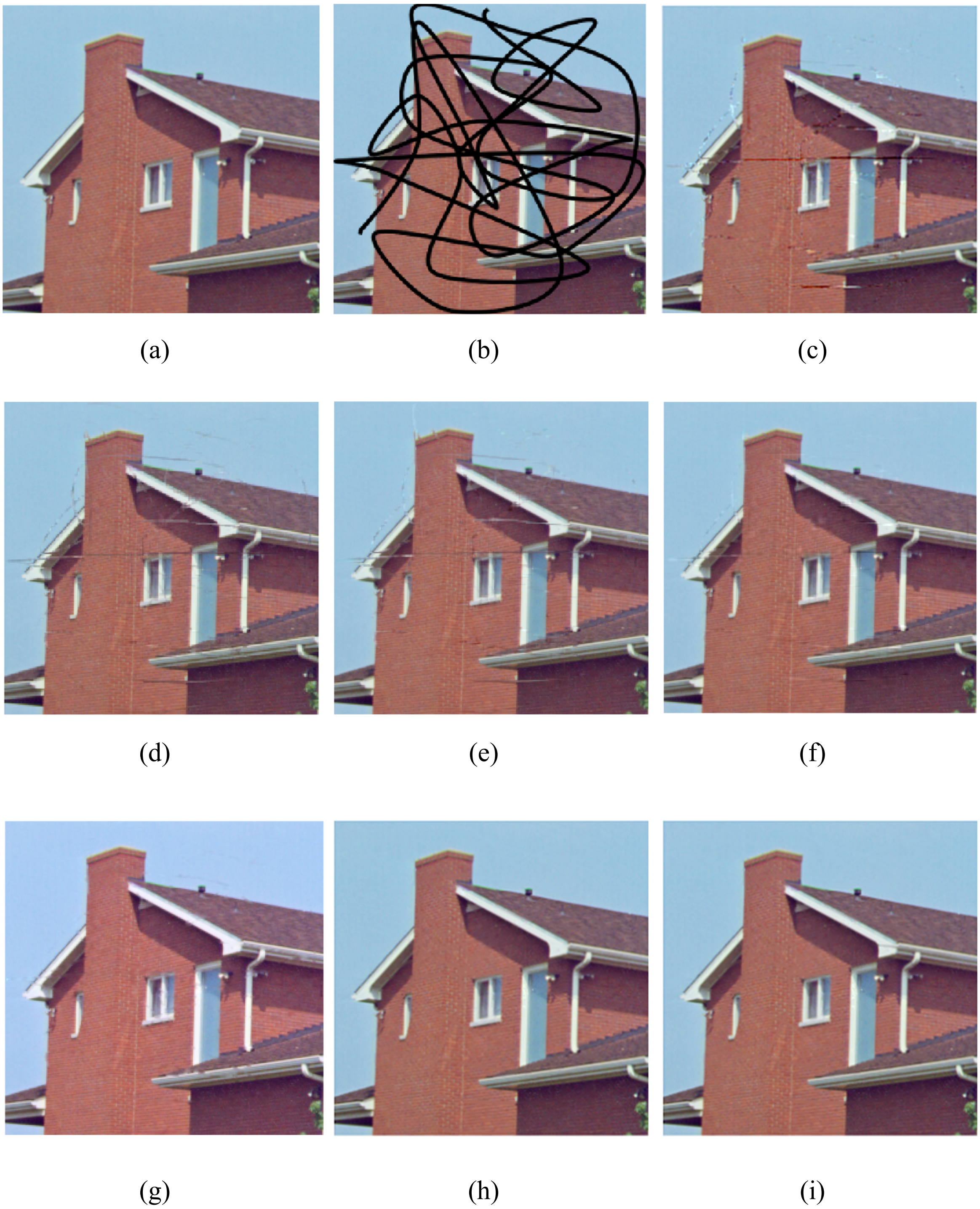
4.2. Doodle color image recovery

To further evaluate the model's performance, we apply it to doodle color image recovery, which involves removing irregular scribbles, grids, and letters. The selected test images consist of a "House", "Sailboat", and "Airplane".

As shown in Figure 6, TMac_DP3, TR-DD, TR-GFR, and 3DlogTNN effectively remove the random graffiti. However, graffiti traces remain in TMac, TNN, and TRLRF. As shown in Figure 7, numerous grid cracks are present in TMac, TNN, and TRLRF, and the damaged pixels cannot be restored. Clear but fewer grid traces are observed in TR-GFR and 3DlogTNN. These two models are better at restoring the pixels damaged by the grid graffiti, although some blurred grid traces are present in TMac_DP3 and TR-DD, compared with the other models. As shown in Figure 8 for letter removal, some artifacts are present in TR-GFR, 3DlogTNN, TMac, TNN, and TRLRF.

4.3. Parameter sensitivity analysis

In this section, we focus on discussing the impact of algorithm-related parameters on algorithm performance, including TR tensor rank (for simplicity, we set all elements of the TR rank vector to a scalar and use this scalar to replace the TR rank vector), δ, λ and ρ.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 6. "House" image after graffiti removal. (a) Original image; (b) Noisy image; (c) TMac; (d) TNN; (e) TRLRF; (f) TR-GFR; (g) 3DlogTNN; (h) TMac_DP3; and (i) TR-DD.

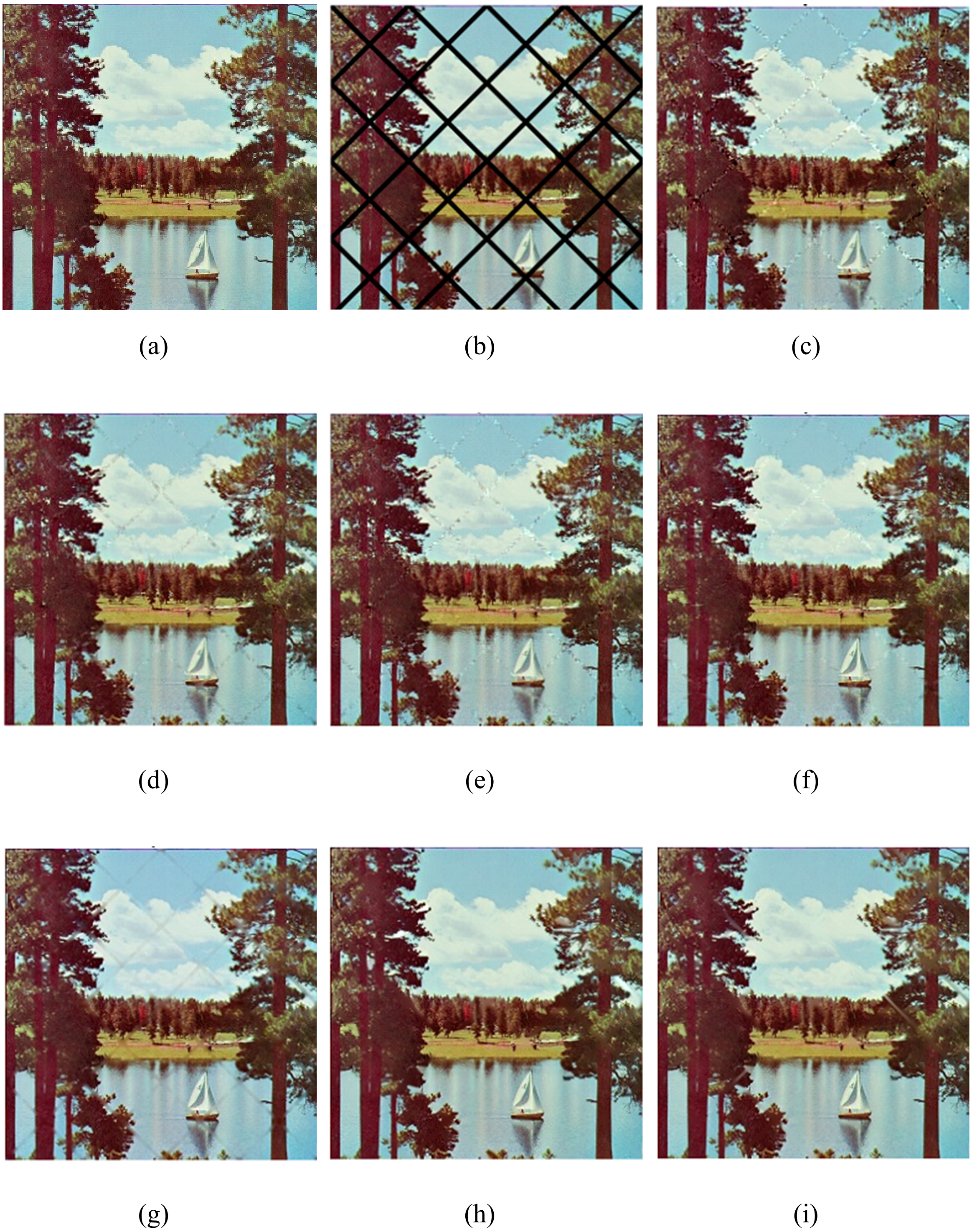


Figure 7. “Sailboat” image after grid removal. (a) Original image; (b) Noisy image; (c) TMac; (d) TNN; (e) TRLRF; (f) TR-GFR; (g) 3DlogTNN; (h) TMac_DP3; and (i) TR-DD.

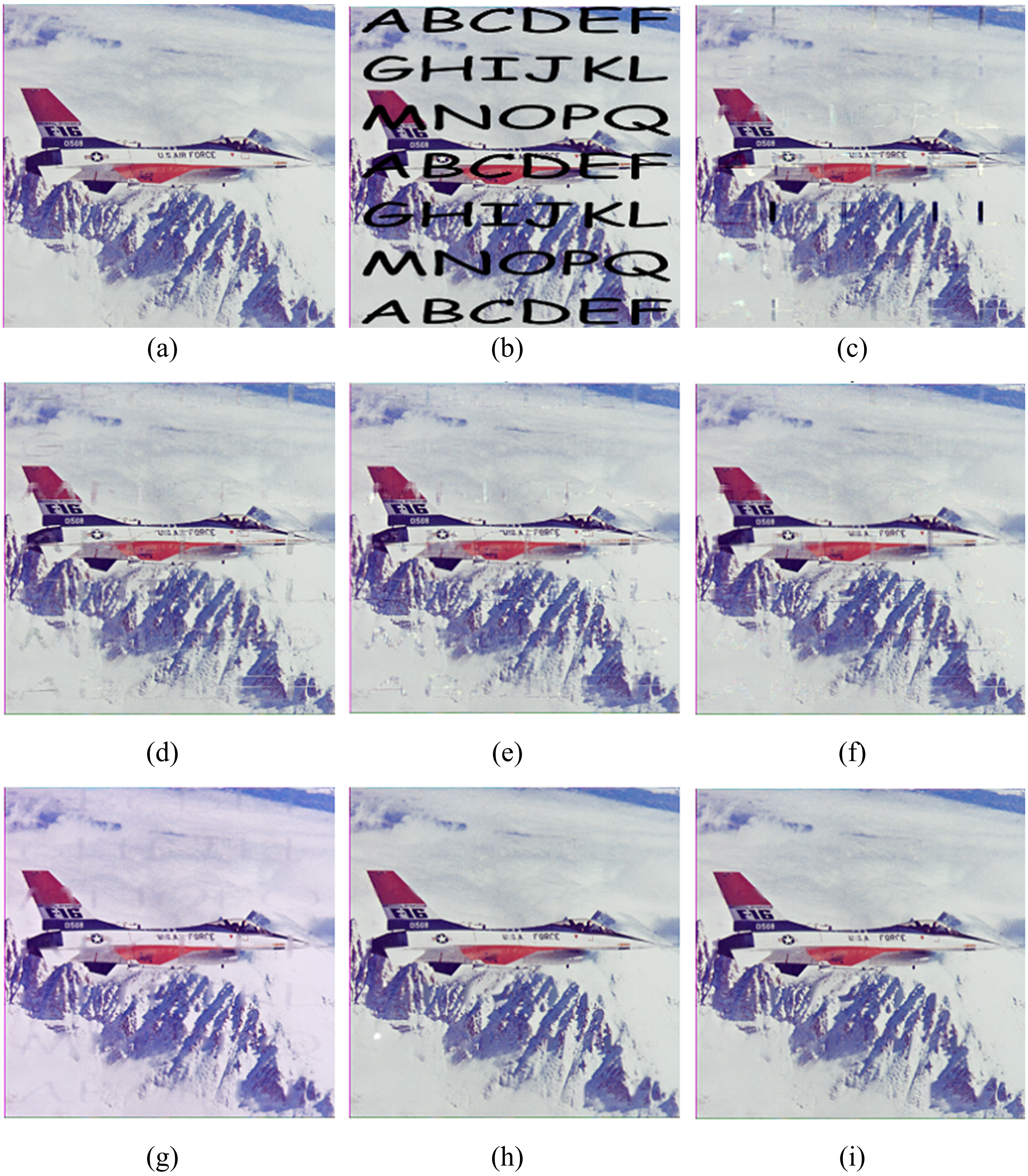


Figure 8. "Airplane" image after letter removal. (a) Original image; (b) Noisy image; (c) TMac_DP3; (d) TRLRF; (e) TMac; (f) TNN; (g) TR-GFR; (h) 3DlogTNN; and (i) TR-DD.

4.3.1. Influence of tensor rank

In this section, we utilize TRLRF and TR-DD models to investigate the effect of TR rank on PSNR value and calculation time. In this section, the test images are "pepper" and "sailboat", and the missing rate is 60%. As shown in Table 3, the PSNR values of the two models increase gradually with the increase in TR rank, and TR-DD is superior to TRLRF in most cases. However, with the increase in TR rank, the time cost increases rapidly. As shown in Table 3, when the TR rank falls between 12 and 16, the model's performance increases gradually; however, the calculation time also increases significantly. When the TR rank is 20, although the time is doubled, the effect decreases. Therefore, to ensure good performance within the acceptable time cost range, the TR rank should be selected as 12~16. Experiments demonstrate that the proposed TR-DD model is robust to rank selection when the TR rank is sufficiently large. The main reason for this robustness is the constraint of minimizing the kernel norm of the TR factor proposed in this paper. Even if the rank setting is relatively large, the kernel norm minimization constraint can ensure that the TR factor maintains a low rank structure.

Table 3. Tensor completion results of TRLRF and TR-DD when the deletion rate is 60% and the TR ranks are different.

Image		TR-rank									
		4		8		12		16		20	
		Method		Method		Method		Method		Method	
		TRLRF	TR-DD	TRLRF	TR-DD	TRLRF	TR-DD	TRLRF	TR-DD	TRLRF	TR-DD
Peppers	PSNR (dB)	26.067	25.646	30.767	30.865	32.915	33.014	33.879	35.264	33.968	**33.991**
	Time (s)	**5.39**	9.814	9.61	18.184	21.06	35.001	33.06	56.754	58.90	95.022
Sailboat	PSNR (dB)	24.776	25.276	28.256	29.290	29.650	31.192	30.055	31.829	30.159	**31.700**
	Time (s)	**7.27**	9.879	11.80	21.294	18.94	36.056	33.06	60.325	68.60	87.336

4.3.2. Influence of hyperparameter δ

In this section, the influence of the hyperparameter ϕ on the TR-DD model is discussed. In $\phi = \sqrt{\frac{\delta}{\lambda}}$, ϕ can be adjusted through δ, and the test image is the “House” image.

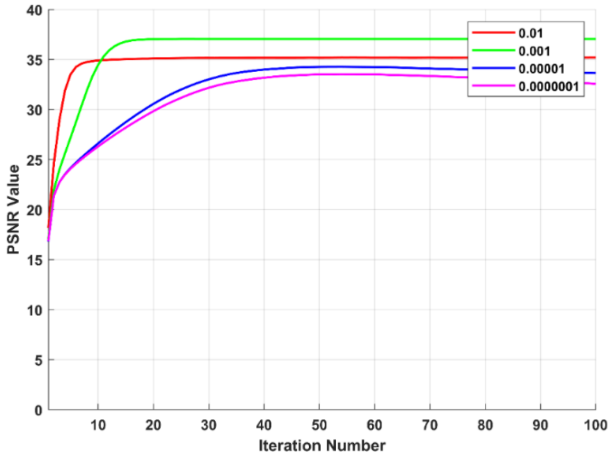
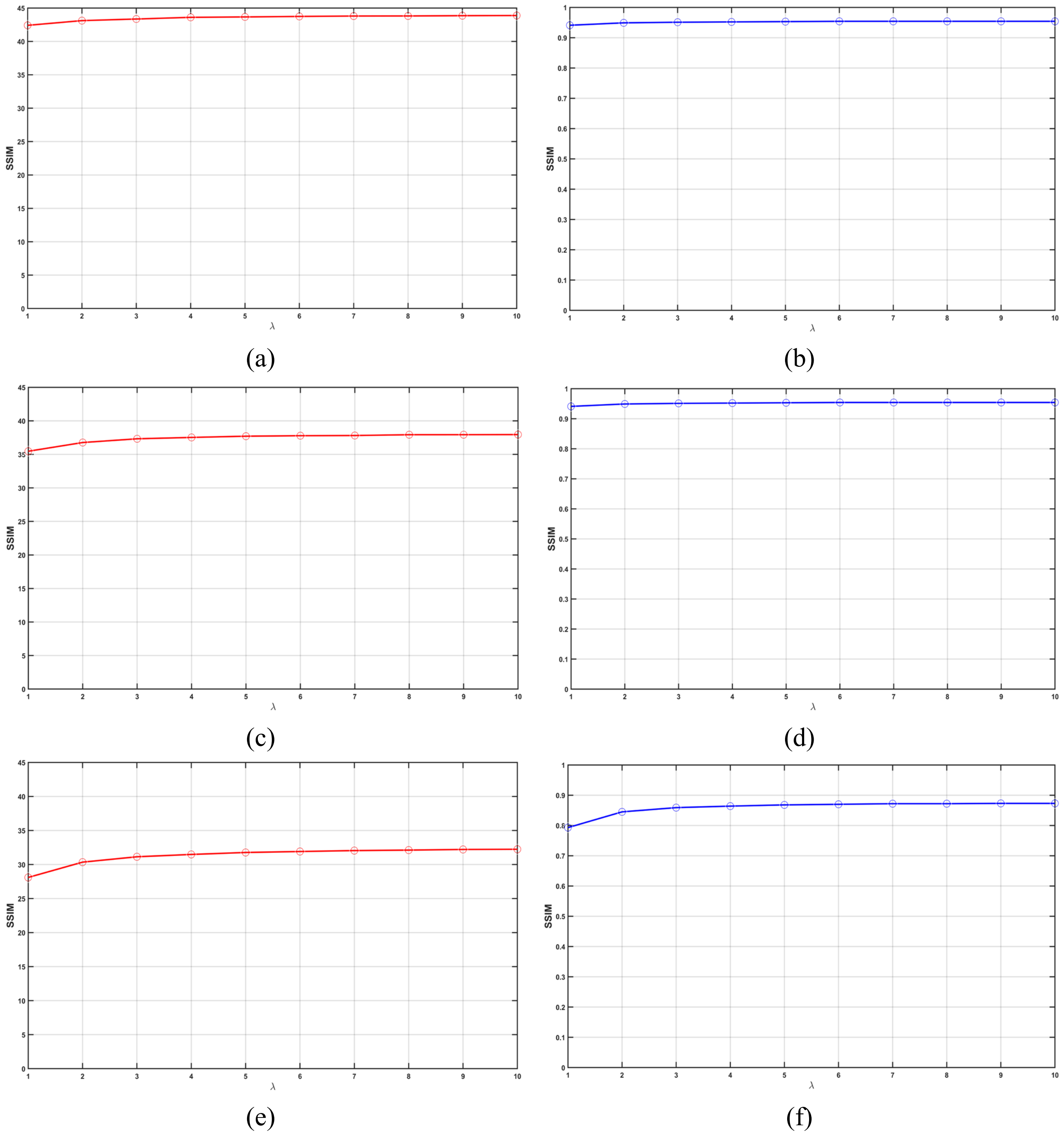


Figure 9. Hyperparameter δ sensitivity curve.

As shown in Figure 9, when other parameters remain unchanged, the completion effect first increases and then decreases with an increase in δ. This is because the FFDNET network can restore image detail, making the edges more straightforward. However, as the value of hyperparameter δ increases, color artifacts appear in the image, making it appear unrealistic and overly smooth, thus counteracting the completion effect. Thus, the selection of an appropriate hyperparameter ϕ is important.

4.3.3. Influence of hyperparameter λ



(a) (b)

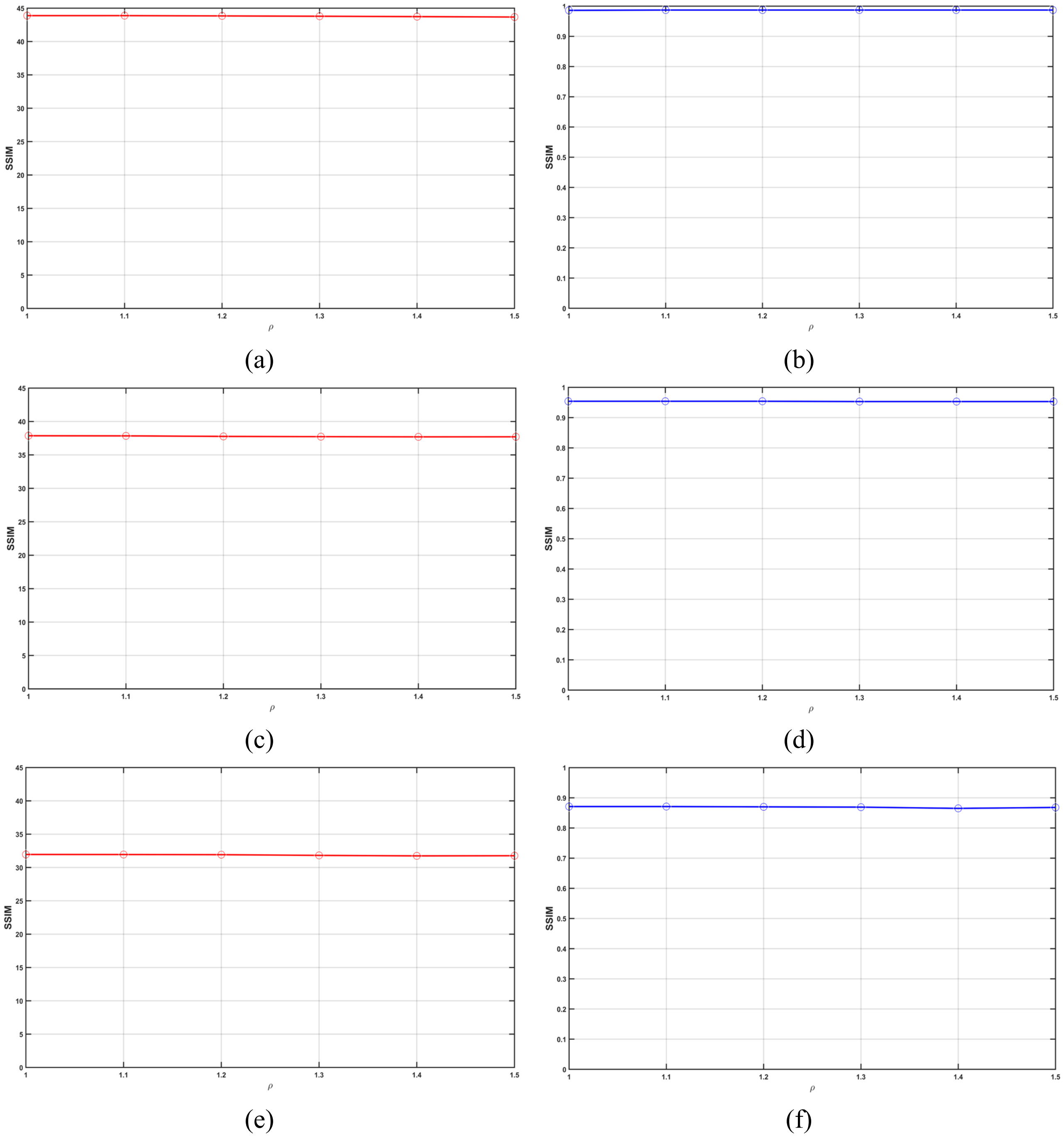
(c) (d)

(e) (f)

Figure 10. Hyperparameter λ sensitivity curve. (a) PSNR curve at MR = 20%; (b) SSIM curve at MR = 20%; (c) PSNR curve at MR = 50%; (d) SSIM curve at MR = 50%; (e) PSNR curve at MR = 80%; and (f) SSIM curve at MR = 80%.

In this section, we analyze the influence of the parameter λ on the TR-DD model. λ is a regularization parameter that adjusts the weight of $\left\|\mathcal{R}-\mathbf{TR}\left(\mathcal{G}^{(1:K)}\right)\right\|_2^2$. We carry out experiments on the color image "Panda" with different Missing Rate (MR), and the value of λ is selected from 1 to 10 one by one. Figure 10 shows the curves of PSNR and SSIM values obtained for different MR values at 20%, 50%, and 80%. As shown in Figure 10, when other parameters remain constant, the values of PSNR and SSIM tend to stabilize after increasing slightly with the gradual increase in the value. Therefore, the TR-DD model exhibits obvious stability and is less affected by the parameter.

4.3.4. Influence of hyperparameter ρ



(a) (b)

(c) (d)

(e) (f)

Figure 11. Hyperparameter ρ sensitivity curve. (a) PSNR curve at MR = 20%; (b) SSIM curve at MR = 20%; (c) PSNR curve at MR = 50%; (d) SSIM curve at MR = 50%; (e) PSNR curve at MR = 80%; and (f) SSIM curve at MR = 80%.

In this section, we analyze the effect of parameter ρ on the TR-DD model. The value of ρ is selected one by one from 1 to 1.5. Figure 11 shows the curves of PSNR and SSIM values obtained for different MR values at 20%, 50%, and 80%. It can be seen from Figure 11 that when other parameters remain unchanged, the values of PSNR and SSIM do not fluctuate significantly with the gradual increase in the value and remain in a stable state. Therefore, we can infer that the TR-DD model is less sensitive to the parameter and exhibits superior stability.

4.4. Computational efficiency comparison

In this section, we compare the computational efficiency of seven methods. Table 4 presents a comparison of the calculation times for seven models under different MR choices on the "Panda" test image. It can be seen that the insufficient calculation efficiency is the main limitation of this model. In the future, we will focus on optimizing the work to improve the model's running speed further.

Table 4. The cost times of color image completion calculated by different methods.

Image	Method	Missing rate						
		20%	30%	40%	50%	60%	70%	80%
Panda	TMac	6.276	6.757	5.835	6.028	5.565	5.255	5.717
	TNN	6.346	5.740	**5.551**	**5.602**	**5.371**	**5.245**	**5.067**
	TRLRF	32.567	36.723	37.521	39.922	39.963	40.824	40.311
	TR-GFR	38.130	38.917	39.622	37.827	38.186	39.502	39.856
	3DLogTNN	**5.453**	**5.643**	6.693	7.067	7.739	8.030	8.807
	TMac_DP3	28.316	28.440	29.153	28.436	28.507	27.443	29.475
	TR-DD	7.971	13.324	28.162	41.945	47.643	47.803	47.919

5. Conclusions

We introduced energy functional regularization based on DL within a TR decomposition framework. A new model combining model- and data-driven approaches was proposed. First, regularization techniques were introduced to improve the robustness of the TR rank, thereby enhancing the global low-rank property of the target tensor and its local detail recovery ability. A data-driven DL network was then used to strengthen the robustness of the model parameter selection and simultaneously obtain detailed image texture that was difficult to capture using low-rank constraints. Subsequently, an efficient ADMM algorithm was used to solve the new model. Finally, image reconstruction experiments were conducted with different missing color image rates to compare the performance of each method. The experimental results demonstrated that the proposed model outperforms advanced image-completion models. In the future, a better network that can serve as a DL prior for tensor completion would improve image completion performance. Additionally, combining the non-local similarity of images to enhance reconstruction performance is also a promising research direction.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This work was supported by the Natural Science Foundation of Fujian Province (2024J01820, 2024J01821, 2024J01822), the Principal Foundation of Minnan Normal University (KJ19019), the Education and Teaching Reform Research Project of Minnan Normal University (JG202425), Postgraduate Education and Teaching Reform Research Project (YJG202526), the High-Level Science Research Project of Minnan Normal University (GJ19019), Research Project on Education and Teaching of Undergraduate Colleges and Universities in Fujian Province (FBJY20230083), Guangdong Province Natural Science Foundation (2024A1515011766), National Independent Innovation Demonstration Zone System (Fuzhou, Xiamen, Quanzhou) Innovation Platform Project (3502ZCQXT2024006), Major Postgraduate Education Reform Project of Undergraduate Colleges and Universities in Fujian Province (FBJY20250180), and State Key laboratory Major Special Project of Jilin Province Science and Technology Development Plan (SKL202402024).

Conflict of interest

The authors declare there is no conflict of interest.

References

1. H. Jiao, Q. Liu, K. Liang, L. An, Low-rank quaternion tensor completion by truncated nuclear norm regularization for color video recovery, *Neurocomputing*, **661** (2026), 131905. https://doi.org/10.1016/j.neucom.2025.131905
2. Y. Chen, Z. Peng, M. Li, F. Yu, F. Lin, Seismic signal denoising using total generalized variation with overlapping group sparsity in the accelerated ADMM framework, *J. Geophys. Eng.*, **16** (2019), 30–51. https://doi.org/10.1093/jge/gxy003
3. D. N. H. Thanh, L. T. Thanh, N. N. Hien, S. Prasath, Adaptive total variation L1 regularization for salt and pepper image denoising, *Optik*, **208** (2020), 163677. https://doi.org/10.1016/j.ijleo.2019.163677
4. L. Wang, Y. Chen, F. Lin, Y. Chen, F. Yu, Z. Cai, Impulse noise denoising using total variation with overlapping group sparsity and Lp-pseudo-norm shrinkage, *Appl. Sci.*, **8** (2018), 2317. https://doi.org/10.3390/app8112317
5. M. Ding, X. Fu, X. L. Zhao, Bilinear hyperspectral unmixing via tensor decomposition, in *2023 31st European Signal Processing Conference (EUSIPCO)*, IEEE, (2023), 640–644. https://doi.org/10.23919/EUSIPCO58844.2023.10289962
6. M. Ding, X. Fu, X. L. Zhao, Fast and structured block-term tensor decomposition for hyperspectral unmixing, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **16** (2023), 1691–1709. https://doi.org/10.1109/jstars.2023.3238653
7. Z. L. Han, T. Z. Huang, X. L. Zhao, H. Zhang, W. H. Wu, Nested fully-connected tensor network decomposition for multi-dimensional visual data recovery, *IEEE Trans. Circuits Syst. Video Technol.*, **34** (2024), 10092–10106. https://doi.org/10.1109/TCSVT.2024.3401134
8. Y. Luo, X. Zhao, Z. Li, M. K. Ng, D. Meng, Low-rank tensor function representation for multi-dimensional data recovery, *IEEE Trans. Pattern Anal. Mach. Intell.*, **46** (2023), 3351–3369. https://doi.org/10.1109/TPAMI.2023.3341688

9. S. Liu, J. Leng, X. L. Zhao, H. Zeng, Y. Wang, J. H. Yang, Learnable spatial-spectral transform-based tensor nuclear norm for multi-dimensional visual data recovery, *IEEE Trans. Circuits Syst. Video Technol.*, **34** (2023), 3633–3646. https://doi.org/10.1109/TCSVT.2023.3316279
10. Z. L. Han, T. Z. Huang, X. L. Zhao, H. Zhang, Y. Y. Liu, Multi-dimensional data recovery via feature-based fully-connected tensor network decomposition, *IEEE Trans. Big Data*, **10** (2023), 386–399. https://doi.org/10.1109/TBDATA.2023.3342611
11. Y. Chen, Y. Huang, L. Wang, H. Huang, J. Song, C. Yu, et al., Salt and pepper noise removal method based on stationary Framelet transform with non-convex sparsity regularization, *IET Image Process.*, **16** (2022), 1846–1865. https://doi.org/10.1049/ipr2.12451
12. Y. Chen, H. Xie, J. Zhang, H. Yang, H. Wang, L. Wang, Infrared image deblurring with Framelet regularization in the background of salt-and-pepper noise, *J. Mod. Opt.*, **71** (2024), 1–15. https://doi.org/10.1080/09500340.2024.2383277
13. S. Gu, L. Zhang, W. Zuo, X. Feng, Weighted nuclear norm minimization with application to image denoising, in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, IEEE, (2014), 2862–2869. https://doi.org/10.1109/CVPR.2014.366
14. W. Dong, L. Zhang, G. Shi, X. Li, Nonlocally centralized sparse representation for image restoration, *IEEE Trans. Image Process.*, **22** (2012), 1620–1630. https://doi.org/10.1109/TIP.2012.2235847
15. J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, Non-local sparse models for image restoration, in *2009 IEEE 12th International Conference on Computer Vision*, IEEE, (2009), 2272–2279. https://doi.org/10.1109/ICCV.2009.5459452
16. J. Yang, Y. Chen, Z. Chen, Infrared image deblurring via high-order total variation and Lp-pseudonorm shrinkage, *Appl. Sci.*, **10** (2020), 2533. https://doi.org/10.3390/app10072533
17. H. Wang, Y. Cen, Z. He, Z. He, R. Zhao, F. Zhang, Reweighted low-rank matrix analysis with structural smoothness for image denoising, *IEEE Trans. Image Process.*, **27** (2017), 1777–1792. https://doi.org/10.1109/TIP.2017.2781425
18. L. Chiantini, G. Ottaviani, On generic identifiability of 3-tensors of small rank, *SIAM J. Matrix Anal. Appl.*, **33** (2012), 1018–1037. https://doi.org/10.1137/110829180
19. F. L. Hitchcock, The expression of a tensor or a polyadic as a sum of products, *J. Math. Phys.*, **6** (1927), 164–189. https://doi.org/10.1002/sapm192761164
20. X. Liu, S. Bourennane, C. Fossati, Denoising of hyperspectral images using the PARAFAC Model and statistical performance analysis, *IEEE Trans. Geosci. Remote Sens.*, **50** (2012), 3717–3724. https://doi.org/10.1109/tgrs.2012.2187063
21. N. Renard, S. Bourennane, J. Blanc-Talon, Denoising and dimensionality reduction using multilinear tools for hyperspectral images, *IEEE Geosci. Remote Sens. Lett.*, **5** (2008), 138–142. https://doi.org/10.1109/lgrs.2008.915736
22. Y. Chen, W. He, N. Yokoya, T. Z. Huang, Hyperspectral image restoration using weighted group sparsity-regularized low-rank tensor decomposition, *IEEE Trans. Cybern.*, **50** (2019), 3556–3570. https://doi.org/10.1109/tcyb.2019.2936042
23. X. Bai, F. Xu, L. Zhou, Y. Xing, L. Bai, J. Zhou, Nonlocal similarity based nonnegative tucker decomposition for hyperspectral image denoising, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **11** (2018), 701–712. https://doi.org/10.1109/jstars.2018.2791718

24. Y. Wang, J. Peng, Q. Zhao, Y. Leung, X. L. Zhao, D. Meng, Hyperspectral image restoration via total variation regularized low-rank tensor decomposition, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **11** (2017), 1227–1243. https://doi.org/10.1109/jstars.2017.2779539
25. Y. Chen, T. Z. Huang, X. L. Zhao, Destriping of multispectral remote sensing image using low-rank tensor decomposition, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, **11** (2018), 4950–4967. https://doi.org/10.1109/jstars.2018.2877722
26. W. Gong, Z. Huang, L. Yan, Enhanced low-rank and sparse tucker decomposition for image completion, in *2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE, (2024), 2425–2429. https://doi.org/10.1109/ICASSP48485.2024.10448445
27. M. E. Kilmer, K. Braman, N. Hao, R. C. Hoover, Third-order tensors as operators on matrices: A theoretical and computational framework with applications in imaging, *SIAM J. Matrix Anal. Appl.*, **34** (2013), 148–172. https://doi.org/10.1137/110837711
28. M. E. Kilmer, C. D. Martin, Factorization strategies for third-order tensors, *Linear Algebra Appl.*, **435** (2011), 641–658. https://doi.org/10.1016/j.laa.2010.09.020
29. Y. B. Zheng, T. Z. Huang, X. L. Zhao, T. X. Jiang, T. Y. Ji, Mixed noise removal in hyperspectral image via low-fibered-rank regularization, *IEEE Trans. Geosci. Remote Sens.*, **58** (2019), 734–749. https://doi.org/10.1109/TGRS.2019.2940534
30. D. Goldfarb, Z. Qin, Robust low-rank tensor recovery: Models and algorithms, *SIAM J. Matrix Anal. Appl.*, **35** (2014), 225–253. https://doi.org/10.1137/130905010
31. Y. Hu, Z. Tang, T. X. Jiang, X. L. Zhao, G. Liu, Degradation accordant plug-and-play for low-rank tensor recovery, *Pattern Recognit.*, **172** (2026), 112612. https://doi.org/10.1016/j.patcog.2025.112612
32. Y. B. Zheng, X. L. Zhao, H. C. Li, C. Li, T. Z. Huang, Q. Zhao, Tensor network decomposition for data recovery: Recent advancements and future prospects, *Neural Networks*, **191** (2025), 107808. https://doi.org/10.1016/j.neunet.2025.107808
33. J. A. Bengua, H. N. Phien, H. D. Tuan, M. N. Do, Efficient tensor completion for color image and video recovery: Low-rank tensor train, *IEEE Trans. Image Process.*, **26** (2017), 2466–2479. https://doi.org/10.1109/TIP.2017.2672439
34. Q. Zhao, G. Zhou, S. Xie, L. Zhang, A. Cichocki, Tensor ring decomposition, preprint, arXiv:160605535.
35. J. Miao, K. I. Kou, H. Cai, L. Liu, Quaternion tensor left ring decomposition and application for color image inpainting, *J. Sci. Comput.*, **101** (2024), 1. https://doi.org/10.1007/s10915-024-02624-z
36. P. L. Wu, X. L. Zhao, M. Ding, Y. B. Zheng, L. B. Cui, T. Z. Huang, Tensor ring decomposition-based model with interpretable gradient factors regularization for tensor completion, *Knowledge-Based Syst.*, **259** (2023), 110094. https://doi.org/10.1016/j.knosys.2022.110094
37. Y. B. Zheng, T. Z. Huang, X. L. Zhao, Q. Zhao, T. X. Jiang, Fully-connected tensor network decomposition and its application to higher-order tensor completion, in *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI Press, (2021), 11071–11078. https://doi.org/10.1609/aaai.v35i12.17321
38. M. Ding, T. Z. Huang, X. L. Zhao, T. H. Ma, Tensor completion via nonconvex tensor ring rank minimization with guaranteed convergence, *Signal Process.*, **194** (2022), 108425. https://doi.org/10.1016/j.sigpro.2021.108425

39. W. Wang, V. Aggarwal, S. Aeron, Efficient low rank tensor ring completion, in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, (2017), 5698–5706. https://doi.org/10.1109/ICCV.2017.607
40. L. Yuan, J. Cao, X. Zhao, Q. Wu, Q. Zhao, Higher-dimension tensor completion via Low-rank tensor ring decomposition, in *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, IEEE, (2018), 1071–1076. https://doi.org/10.23919/APSIPA.2018.8659708
41. Y. Chen, W. He, N. Yokoya, T. Z. Huang, X. L. Zhao, Nonlocal tensor-ring decomposition for hyperspectral image denoising, *IEEE Trans. Geosci. Remote Sens.*, **58** (2019), 1348–1362. https://doi.org/10.1109/tgrs.2019.2946050
42. D. Li, D. Chu, X. Guan, W. He, H. Shen, Adaptive regularized low-rank tensor decomposition for hyperspectral image denoising, *IEEE Trans. Geosci. Remote Sens.*, **62** (2024), 1–17. https://doi.org/10.1109/TGRS.2024.3385536
43. X. L. Zhao, W. H. Xu, T. X. Jiang, Y. Wang, M. K. Ng, Deep plug-and-play prior for low-rank tensor completion, *Neurocomputing*, **400** (2020), 137–149. https://doi.org/10.1016/j.neucom.2020.03.018
44. J. Zhang, Z. Li, L. Wang, Y. Chen, Salt and pepper denoising method for colour images based on tensor low-rank prior and implicit regularization, *IET Image Process.*, **17** (2023), 886–900. https://doi.org/10.1049/ipr2.12680
45. K. Zhang, W. Zuo, L. Zhang, FFDNet: Toward a fast and flexible solution for CNN-based image denoising, *IEEE Trans. Image Process.*, **27** (2018), 4608–4622. https://doi.org/10.1109/TIP.2018.2839891
46. B. Stephen, P. Neal, C. Eric, P. Borja, E. Jonathan, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.*, **3** (2011), 1–122.
47. Y. Chen, Y. Liao, Y. He, X. He, Q. Yu, T. Chen, et al., Non-local similar block matching and hybrid low-rank tensor network for colour image inpainting, *Digital Signal Process.*, **162** (2025), 105169. https://doi.org/10.1016/j.dsp.2025.105169
48. T. G. Kolda, B. W. Bader, Tensor decompositions and applications, *SIAM Rev.*, **51** (2009), 455–500. https://doi.org/10.1137/07070111x
49. S. Holtz, T. Rohwedder, R. Schneider, The alternating linear scheme for tensor optimization in the tensor train format, *SIAM J. Sci. Comput.*, **34** (2012), A683–A713. https://doi.org/10.1137/100818893
50. Y. Xu, R. Hao, W. Yin, Z. Su, Parallel matrix factorization for low-rank tensor completion, *Inverse Probl. Imaging*, **9** (2015), 601–624. https://doi.org/10.3934/ipi.2015.9.601
51. Z. Zhang, S. Aeron, Exact tensor completion using t-SVD, *IEEE Trans. Signal Process.*, **65** (2016), 1511–1526. https://doi.org/10.1109/TSP.2016.2639466
52. L. Yuan, C. Li, D. Mandic, J. Cao, Q. Zhao, Tensor ring decomposition with rank minimization on latent space: An efficient approach for tensor completion, in *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI Press, (2019), 9151–9158. https://doi.org/10.1609/aaai.v33i01.33019151