

RESEARCH

Open Access



Image restoration based on sparse representation using feature classification learning

Minhui Chang^{*} and Lei Zhang

^{*} Correspondence: changminhui@ycu.edu.cn
School of mathematics and
information technology, Yuncheng
University, Yuncheng 044000, China

Abstract

In the image inpainting method based on sparse representation, the adaptability of over-complete dictionary has a great influence on the result of image restoration. If the over-complete dictionary cannot effectively reflect the differences between different local features, it may result in the loss of texture details, resulting in blurred or over-smooth phenomenon in restored images. In view of these problems, we propose an image restoration method based on sparse representation using feature classification learning. Firstly, we perform singular value decomposition on the local gradient vector. According to the relationship between the main orientation and the secondary orientation, we classify all the local patches into three categories: smooth patch, edge patch and texture patch. Secondly, we use K-Singular Value Decomposition method to learn over-complete dictionaries that adapt to different features. Finally, we use Orthogonal Matching Pursuit method to calculate the sparse coding of target patches with different local features on their corresponding over-complete dictionaries, and use the over-complete dictionary and corresponding sparse coding to restore the damaged pixels. A series of experiments on various restoration tasks show the superior performance of the proposed method.

Keywords: Image restoration, Sparse representation, Feature classification learning, Sparse coding

1 Introduction

Image restoration, also called image inpainting, originates from the restoration of damaged works of art [1, 2]. Its main purpose is to achieve automatic restoration of damaged images by computers [3]. For a damaged image with scratches, text overlays, holes, and so on, the computer automatically restores the damaged region according to certain principles or algorithms, so that the obtained image is more natural, and people who are not familiar with the original image cannot notice the restoration trace [4]. So far, image restoration technology has been applied in many fields, such as the restoration of old photographs and precious documents, digital protection of cultural relics [5], special effects of film and television production, and so on.



© The Author(s). 2020 **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

At present, the main image restoration methods can be summarized into three categories. The first is based on Partial Differential Equations [6]. Its basic idea is to smoothly spread the information around the damaged region to the inside of the damaged region according to certain principles, to achieve the purpose of image restoration. The representative methods include BSCB model [7], TV (Total Variation) model [8], CDD (Curvature Driven Diffusion) model [9], and so on. These methods can achieve good restoration effect for small-scale damaged areas, but for large-scale damaged areas, there will be over-smooth phenomenon.

The second is based on Texture Synthesis, also known as patch-based method. Its basic idea is to search for the exemplar patch similar to the damaged patch in the undamaged region according to certain principles, and then use it to restore the damaged patch [10]. The representative methods include exemplar-based method [11], the nonlocal-means method [12], and the non-parametric sampling method [13]. These methods can achieve better results for large-scale damaged areas, but once the mismatch between exemplar patch and target patch occurs, it is easy to introduce unpredictable objects into the image, making the image unable to meet the requirements of visual consistency [14].

In recent years, with the continuous improvement of sparse representation theory, researchers have applied sparse representation to image restoration, and gradually formed the third kind of method, that is, the sparse-representation-based method [15–18]. Its basic idea is to calculate the sparse coding of the damaged patch on the over-complete dictionary, and then reconstruct the damaged patch using the sparse coding and the over-complete dictionary. Since the sparse representation method does not need to search for similar exemplar patches during the restoration process, it will not cause mismatch between the target patch and the exemplar patch. Therefore, this kind of method has quickly become a research hotspot in the field of image restoration. Aharon *et al* [19] proposed a K-Singular Value Decomposition (K-SVD) algorithm, which updates the dictionary atoms and corresponding sparse coefficients simultaneously in the iterative process to obtain an over-complete dictionary that adapts to different types of images. Elad [20] used K-SVD algorithm to restore the missing pixels in the face images and achieved good results. Starck *et al* [21] proposed a Morphological Component Analysis (MCA) method, which assumes that an image can be represented as a linear combination of different components, and different morphological components can be sparsely represented using different dictionaries. Elad *et al* [22] used MCA method to decompose the image into cartoon layer and texture layer, and restored the cartoon layer and texture layer separately. Finally, the two parts were merged to get the final restoration results. Since the method required continuous sparse decomposition and reconstruction of each image layer, the computational complexity was high. Shen *et al* [23] directly used the undamaged patches to form the over-complete dictionary, and used the dictionary to reconstruct the damaged patch. Although the construction method of over-complete dictionary is relatively simple, this method may cause the loss of texture details and over smooth phenomenon for texture-rich images. Using the similarity between the damaged patch and its neighborhood, Xu *et al* [24] proposed the patch sparsity, and used it to improve the restoration order, which can achieve better results for large-scale damaged areas. However, due to the lack of consideration of the local characteristics of different image patches, it is easy to produce

discontinuity in the edge structure. Shi *et al* [25] proposed to constrain the dictionary atom selection by adding double regular terms, but only using the consistency of known information cannot guarantee the similarity of unknown information, so there may be some problems such as loss of texture details. Newson *et al* [26] proposed a non-local patch-based image inpainting method. They used PatchMatch algorithm to search for nearest neighbors, used a weighted mean scheme to improve the comparison of textured patches, and used initialization and a multi-scale scheme to achieve more satisfactory solutions. Li *et al* [27] utilized operator splitting method, obtained a relaxed minimization problem with two variables, then used a universal framework called iterative coupled inpainting algorithm to restore damaged images.

By studying the classical restoration methods based on sparse representation, we find the adaptability of over-complete dictionary have a great influence on the result of image restoration. If the over-complete dictionary cannot effectively reflect the differences between different local features, this may lead to the loss of texture details, resulting in blurred or over-smooth phenomenon in restored images. In the proposed method, we perform singular value decomposition on the local gradient vector of the image patch, estimate the local main orientation according to the calculated singular values, and judge the local feature of image patch. The image patches are divided into three categories: smooth patches, edge patches and texture patches. Then, in order to make the over-complete dictionary adapt to different features, we use image patches with the same local features as a group of training samples, and use K-SVD method to learn dictionaries. Therefore, we can get three over-complete dictionaries in the proposed method. Finally, we use Orthogonal Matching Pursuit (OMP) method to calculate the sparse coding of target patches with different local features on their corresponding over-complete dictionaries, and use the over-complete dictionary and corresponding sparse coding to restore the damaged pixels.

The rest of this paper is organized as follows: In Section 2, we introduce the sparse representation model of images. In Section 3, we give a detailed description of the proposed method, including the method of classifying image patches, the method of learning dictionaries, and the method of restoring damaged image patches using the obtained over-complete dictionaries. The experiments and discussions are performed in Section 4. Finally, we conclude this work in Section 5.

2 Image sparse representation model

In the sparse representation theory, using an over-complete dictionary D , the signal x can be represented as a linear combination of dictionary atoms [28]:

$$x = D\alpha \quad (1)$$

where $D = [d_1, d_2, \dots, d_L] \in \mathbb{R}^{n \times L}$ ($L > n$) is the over-complete dictionary, $x \in \mathbb{R}^n$ is the signal, and $\alpha = [\alpha_1, \alpha_2, \dots, \alpha_L]^T$ is the sparse representation coefficient, it contains only a handful of non-zero elements, most of which are zeros.

In the image sparse representation model, it is assumed that there exists an over-complete dictionary, and the image can be represented as a linear combination of dictionary atoms. Therefore, the sparse representation model of an image is:

$$\hat{\alpha} = \arg \min \|\alpha\|_0 \quad s.t. \quad x \approx D\alpha \quad (2)$$

where $\|\cdot\|_0$ represents l_0 norm. When the condition $x \approx D\alpha$ is satisfied, we need to find the representation coefficient α which includes the least non-zero elements.

In the specific calculation process, the constraint item is often converted into a penalty item, and Eq. (2) is transformed into an unconstrained optimization problem, i.e.:

$$\hat{\alpha} = \arg \min \|D\alpha - x\|_2^2 + \lambda \|\alpha\|_0 \quad (3)$$

where $\|D\alpha - x\|_2^2$ represents the reconstruction error, and $\|\alpha\|_0$ represents the sparsity.

3 Proposed method

In recent years, because the method based on sparse representation can effectively avoid the mismatch between the exemplar patch and the target patch, researchers are conducting extensive and in-depth research on it. In the method, over-complete dictionary is one of the most important factors, and its performance has a great influence on the restoration results. DCT (Discrete Redundant Transform) dictionary is one of the most commonly used dictionaries. It can achieve good restoration effect for smooth images, but for texture-rich images, it will lead to the loss of texture details and the restoration effect need to be improved [29]. Shen *et al* [23] directly used the undamaged patches to form an over-complete dictionary, although the construction of the dictionary is relatively simple, the dictionary has limited ability to restore image.

In the proposed method, in order to further enhance the adaptability of the over-complete dictionary, we classify the image patches into three categories according to the local features, and then use the K-SVD algorithm to train dictionaries separately, so that these dictionaries can better adapt to the image patches with different features. Finally, we use OMP method to calculate the sparse coding of target patches on their corresponding over-complete dictionaries, and use the over-complete dictionary and corresponding sparse coding to restore the damaged pixels. In this way, the adaptability of the over-complete dictionary can be effectively improved, and better restoration effect can be obtained.

3.1 Image patch classification

As we know, for smooth image patches, the gray values of the pixels are relatively close, while for non-smooth patches, the difference between the gray values of pixels is relatively large. Therefore, the variance of an image patch can effectively reflect its smoothness.

The local variance of an image patch is defined as:

$$v = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (4)$$

where n is the number of pixels in the image patch, \bar{x} is the average of all the pixels, it is defined as:

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i \quad (5)$$

In the proposed method, we first used the variance to classify image patches into smooth patches and non-smooth patches. We set a threshold. If the variance of an image patch is less than the threshold, it means that the gray values of all pixels are relatively close, and we consider it a smooth patch. Otherwise, it means that the difference between pixels is relatively large, and we consider it a non-smooth patch.

In addition, we noticed that in non-smooth patches, there are two kinds of image patches with different local features: edge patches and texture patches. However, the ability of local variance to discriminate these two types of image patches is limited. Therefore, we cannot distinguish edge patches from texture patches using the local variance.

How can we effectively distinguish edge patches from texture patches? By observing these two kinds of image patches, we find that the local main orientations of edge patches and texture patches are quite different. As shown in Fig. 1, (a) and (b) are edge patches, (c) and (d) are texture patches, they are all 8×8 image patches. In order to display clearly, we magnified them by 10 times in equal proportion. As can be seen from Fig. 1, in edge patches, the consistency of main orientations is clear and obvious, while in texture patches, the consistency of main orientations is relatively blurred.

Based on the above analysis, in the proposed method, we use the principal component analysis method to estimate the local main orientation of the image patch. According to the relationship between the main orientation and the secondary orientation, the edge patch and the texture patch can be distinguished. If the difference between the main orientation and the secondary orientation is relative large, that is, the consistency of main orientations is clear and obvious, we consider it an edge patch. Otherwise, the difference between the main orientation and the secondary orientation is relative small, that is, the consistency of main orientations is blurred, and we consider it a texture patch.

For an image patch $f(x, y)$, its local main orientation should be orthogonal to the mean of gradient vectors $\vec{g}_i = \nabla f(x_i, y_i)$ of all the pixels in the image patch. Therefore, the estimation of the main orientation can be transformed into finding a unit vector \vec{a} to minimize the inner product between it and \vec{g}_i , that is, we need to solve the following problems [30]:

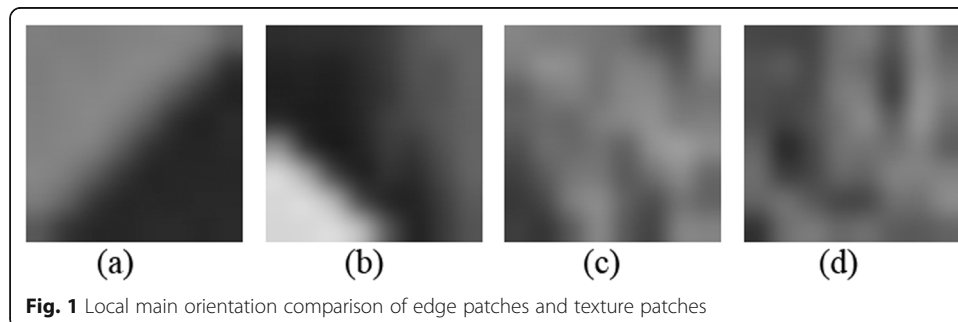


Fig. 1 Local main orientation comparison of edge patches and texture patches

$$\vec{a} = \arg \min \sum_{i=1}^n \left(\vec{a}^T \vec{g}_i \right)^2 = \arg \min \vec{a}^T C \vec{a} \quad s.t. \quad \|\vec{a}\| = 1 \quad (6)$$

where

$$C = \begin{bmatrix} \sum_{i=1}^n g_i^x g_i^x & \sum_{i=1}^n g_i^x g_i^y \\ \sum_{i=1}^n g_i^y g_i^x & \sum_{i=1}^n g_i^y g_i^y \end{bmatrix} \quad (7)$$

where g_i^x is the component of \vec{g}_i in the horizontal direction, g_i^y is the component of \vec{g}_i in the vertical direction.

It can be seen from Eq. (7) that the unit vector \vec{a} that minimizes Eq. (6) is the eigenvector corresponding to the minimum eigenvalue of the matrix C .

For an image patch $f(x, y)$, we convert the gradient vectors of all pixels into $n \times 2$ matrices G :

$$G = \begin{bmatrix} \nabla f(x_1, y_1)^T \\ \nabla f(x_2, y_2)^T \\ \vdots \\ \nabla f(x_n, y_n)^T \end{bmatrix} \quad (8)$$

We perform singular value decomposition on the matrix G :

$$G = U \Lambda V^T \quad (9)$$

By decomposition, we obtain two singular values: s_1 and s_2 . s_1 represents the energy in the main orientation, and s_2 represents the energy orthogonal to the main orientation. Therefore, according to the difference between s_1 and s_2 , edge patches and texture patches are distinguished.

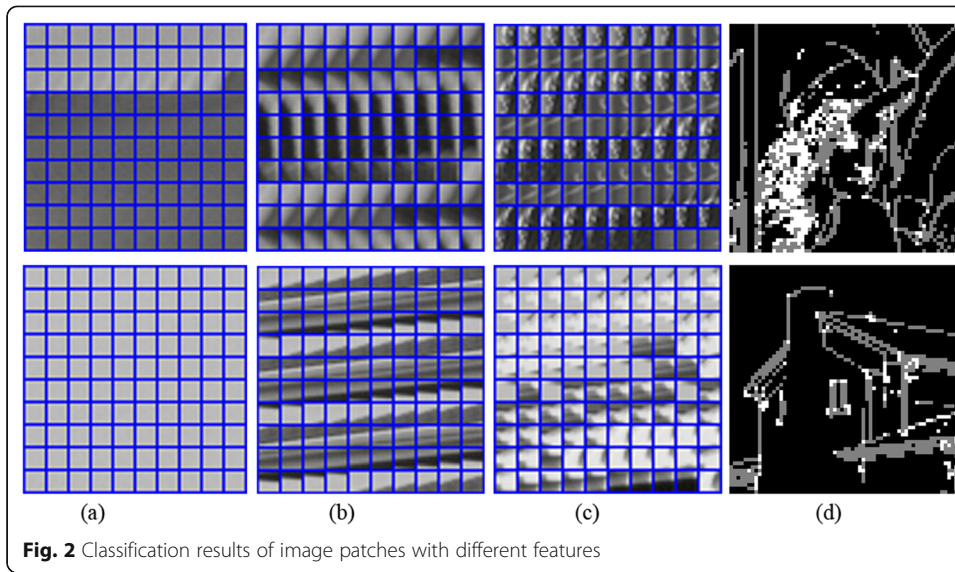
Let:

$$r = \frac{s_1 - s_2}{s_1 + s_2} \quad (10)$$

We set a threshold β . If $r > \beta$, it means that the difference between s_1 and s_2 is relative large, that is, the consistency of main orientations is clear and obvious, we judge the current patch as an edge patch. Otherwise, it means that the difference between s_1 and s_2 is relative small, that is, the consistency of main orientations is blurred, we judge the current patch as a texture patch.

For example, we use the proposed classification method to divide the image patches of two images (standard test image Lena and House) into three categories, and the results are shown in Fig. 2, where (a) shows the obtained smooth patches, (b) shows the obtained edge patches, and (c) shows the obtained texture patches. It should be mentioned that there are many image patches for each type. In order to facilitate display, we only show 100 image patches for each type. Besides, we also show the classification results of the whole image in (d), where black indicates smooth patches, gray indicates edge patches, and white indicates texture patches.

From the Fig. 2 we can see that using the above methods, we can effectively classify the image patches into three categories: smooth patches, edge patches and texture patches. In the image patches in (a), that is, the first column in Fig. 2, the gray values



of the pixels are relatively close, so they are judged as smooth patches. In the image patches in (b), that is, the second column in Fig. 2, the consistency of main orientations is clear and obvious, so they are judged as edge patches. In the image patches in (c), that is, the third column in Fig. 2, although the gray values of pixels are quite different, the consistency of the main orientations is not obvious, so they are judged as texture patches. Besides, it can be seen from the classification results of the entire image, as shown in (d), the black image patches (i.e. the smooth patches) are located in the smooth area of the image, the gray image patches (i.e. the edge patches) are located in the edge area of the image, and the white image patches (i.e. the texture patches) are located in the texture area of the image. Therefore, these results can verify the effectiveness of the classification method used in this paper.

3.2 Dictionary learning

After classifying image patches according to local features, we take image patches with the same features as a group and learn the over-complete dictionary respectively, so that the obtained over-complete dictionaries can be adapted to image patches with different local features.

At present, K-SVD algorithm is the most commonly used method for learning dictionaries. Given a set of signal samples $X = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^{n \times N}$, we need to find an over-complete dictionary D , so that X can be sparsely represented on the dictionary D , i.e.:

$$\min_{D, \alpha} \|X - D\alpha\|^2 \quad s.t. \quad \|\alpha_i\|_0 \leq T_0 \quad (11)$$

where $D = \{d_1, d_2, \dots, d_K\} \in \mathbb{R}^{n \times K}$, $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_N\} \in \mathbb{R}^{K \times N}$, α_i is the sparse representation coefficient of each sample x_i , T_0 is the sparsity.

The difference between Eq. (11) and Eq. (2) is that the dictionary in Eq. (2) is known, the coefficients are unknown, while in Eq. (11), the dictionary and the coefficients are unknown. Therefore, the process of solving Eq. (11) consists of two stages: sparse

coding and dictionary updating, the dictionary atoms and coefficients are updated simultaneously by iteration.

Suppose $X_s = \{x_i | i \in \Omega_s\}$ represents a set of training samples randomly extracted from smooth feature patches, $X_e = \{x_i | i \in \Omega_e\}$ represents a set of training samples randomly extracted from edge feature patches, and $X_t = \{x_i | i \in \Omega_t\}$ represents a set of training samples randomly extracted from texture feature patches. Using K-SVD algorithm to learn over-complete dictionaries adapted to different local features can be described as follows:

$$\min_{D_s, \alpha_s} \|X_s - D_s \alpha_s\|^2 \quad s.t. \quad \|\alpha_i\|_0 \leq T_0, i \in \Omega_s \quad (12)$$

$$\min_{D_e, \alpha_e} \|X_e - D_e \alpha_e\|^2 \quad s.t. \quad \|\alpha_i\|_0 \leq T_0, i \in \Omega_e \quad (13)$$

$$\min_{D_t, \alpha_t} \|X_t - D_t \alpha_t\|^2 \quad s.t. \quad \|\alpha_i\|_0 \leq T_0, i \in \Omega_t \quad (14)$$

In the learning process of Eq. (12), (13), and (14), each type of over-complete dictionary requires an initial value. In some methods, the DCT dictionary is used as the initial value. In proposed method, we randomly select image patches from Ω_s, Ω_e , and Ω_t , respectively, to form the initial dictionaries D_s , D_e , and D_t , so as to accelerate the convergence speed of the algorithm. This learning process is iterative and consists of two stages: sparse coding and dictionary updating. In sparse coding stage, we use the OMP algorithm to calculate the representation coefficients of an image patch on the current over-complete dictionary. In dictionary updating stage, dictionary atoms and representation coefficients are updated at the same time.

After the learning process, we obtain three dictionaries adapted to different characteristics: smooth dictionary D_s , edge dictionary D_e , and texture dictionary D_t . In Fig. 3, we show three dictionaries learned from the standard test image Lena using the above method, where (a) is smooth dictionary D_s , (b) is edge dictionary D_e , and (c) is texture dictionary D_t .

3.3 Image patch restoration

After getting three over-complete dictionaries, we can use them to restore the image patches of different features. Suppose x is the current damaged image patch, and M is the binary mask, it uses one to indicate the pixels that need to be restored, and uses zero to indicate the already existing pixels. The sparse coding of x on the corresponding over-complete dictionary can be described as follows:

$$\begin{cases} \hat{\alpha} = \arg \min \|\hat{x}' - D' \alpha\|_2^2 + \lambda \|\alpha\|_0 \\ \hat{x}' = \overline{M}x, D' = \overline{M}D \end{cases} \quad (15)$$

In Eq. (15), since the image patch x includes damaged pixels, in order to restore the damaged pixels, we need to extract effective information from x to form the \hat{x}' . Accordingly, the effective information of each atom is extracted from the dictionary D to form the D' .

In the proposed method, image patches with different features are sparsely coded in different dictionaries. Therefore, the sparse coding of smooth patch, edge patch and texture patch can be described as follows:

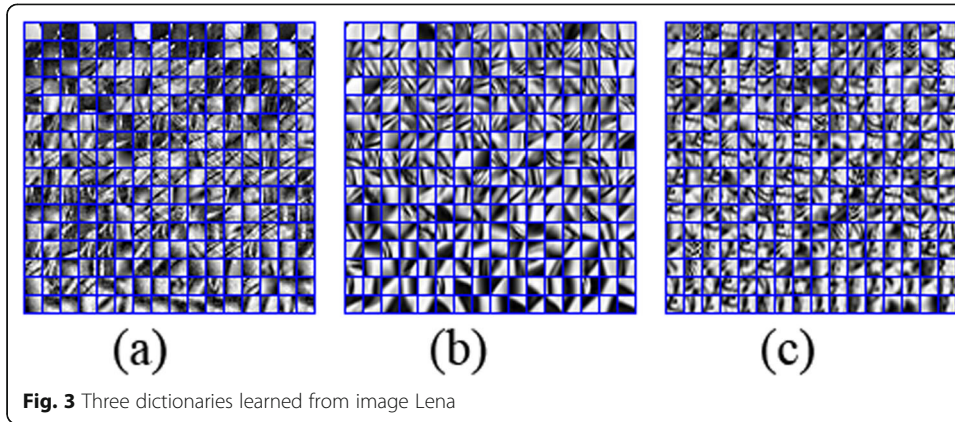


Fig. 3 Three dictionaries learned from image Lena

$$\begin{cases} \hat{\alpha}_s = \arg \min \|x'_s - D'_s \alpha_s\|_2^2 + \lambda \|\alpha_s\|_0 \\ x'_s = \overline{M}x_s, D'_s = \overline{M}D_s \end{cases} \quad (16)$$

$$\begin{cases} \hat{\alpha}_e = \arg \min \|x'_e - D'_e \alpha_e\|_2^2 + \lambda \|\alpha_e\|_0 \\ x'_e = \overline{M}x_e, D'_e = \overline{M}D_e \end{cases} \quad (17)$$

$$\begin{cases} \hat{\alpha}_t = \arg \min \|x'_t - D'_t \alpha_t\|_2^2 + \lambda \|\alpha_t\|_0 \\ x'_t = \overline{M}x_t, D'_t = \overline{M}D_t \end{cases} \quad (18)$$

For Eqs (16), (17) and (18), we use the OMP algorithm to solve them and get the corresponding sparse representation coefficients: α_s, α_e , and α_t .

Finally, we use over-complete dictionaries: D_s, D_e, D_t and corresponding representation coefficients: $\alpha_s, \alpha_e, \alpha_t$ to restore the damaged image patches. The restoration processes of smooth patch, edge patch and texture patch are described as follows:

$$\hat{x}_i^s = \begin{cases} x_i^s, & i \notin M \\ D_s \alpha_s, & i \in M \end{cases} \quad (19)$$

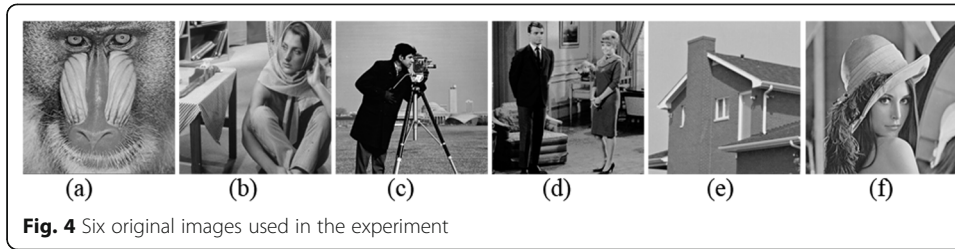
$$\hat{x}_i^e = \begin{cases} x_i^e, & i \notin M \\ D_e \alpha_e, & i \in M \end{cases} \quad (20)$$

$$\hat{x}_i^t = \begin{cases} x_i^t, & i \notin M \\ D_t \alpha_t, & i \in M \end{cases} \quad (21)$$

where i indicates the location of pixels in an image patch. If the current pixel is undamaged, it remains unchanged. Otherwise, it is restored using the dictionary and the corresponding coefficient.

4 Experimental results and discussion

In order to verify the feasibility and effectiveness of the proposed method, we selected six standard test images in the experiment, as shown in Fig. 4, where (a) is Baboon, (b) is Barbara, (c) is Cameraman, (d) is Couple, (e) is House, and (f) is Lena. We performed simulation experiments in three aspects: removing text coverage, removing scratches, and filling holes. For better comparison and analysis, we restored each damaged image using BSCB method [7], TV method [8], DCT method, K-SVD method [19], NLPB



(Non-Local Patch-Based Image Inpainting, NLPB for short) method [26], and the proposed method, respectively.

The experimental environment is Matlab 2014, and the computer is configured as an Intel (R) Core (TM) i5-6200U 2.3GHz processor, 4G Memory. The main parameters are set as follows: The size of the image patch is 8×8 , and the size of all the over-complete dictionary is 64×256 . The iteration times of the BSCB method and the TV method are set to 2000 and 1000, respectively.

In order to objectively and quantitatively compare the restored images, we compare the results of each method from two aspects of Peak Signal-to-Noise Ratio (PSNR for short) and Structure Similarity (SSIM for short). On the one hand, PSNR is analyzed and calculated based on the difference between the gray values of two images, and it is often used to measure the degree of signal distortion. It can reflect the difference of corresponding pixels between the restored image and the original image, so we use it to measure the restoration effect of each method.

Assuming that $f(x, y)$ and $\hat{f}(x, y)$ are the original pixels and the restored pixels respectively, for an image $M \times N$, the PSNR is defined as follows:

$$PSNR = 10 \cdot \log_{10} \left(\frac{255^2}{MSE} \right) \quad (22)$$

where MSE is the Mean Square Error, and it is defined as:

$$MSE = \frac{1}{M} \times \frac{1}{N} \times \sum_{i=1}^M \sum_{j=1}^N [f(x, y) - \hat{f}(x, y)]^2 \quad (23)$$

On the other hand, SSIM measures the similarity of two images in structure information from three aspects: luminance, contrast and structure. The SSIM is defined as follows:

$$SSIM(f, \hat{f}) = [l(f, \hat{f})]^\alpha \times [c(f, \hat{f})]^\beta \times [s(f, \hat{f})]^\gamma \quad (24)$$

where $\alpha > 0, \beta > 0, \gamma > 0$, are used to adjust the weight of the three parts respectively. $l(f, \hat{f})$ is luminance function, it is defined as:

$$l(f, \hat{f}) = \frac{2u_f u_{\hat{f}} + C_1}{u_f^2 + u_{\hat{f}}^2 + C_1}, C_1 = (K_1 L)^2 \quad (25)$$

where L is the dynamic change range of the pixel, for the gray image, its value is 255. $K_1 \ll 1$ is a constant, u_f and $u_{\hat{f}}$ are the average luminance of the two images.

$c(f, \hat{f})$ is the contrast function, it is defined as:

$$c(f, \hat{f}) = \frac{2\sigma_f\sigma_{\hat{f}} + C_2}{\sigma_f^2 + \sigma_{\hat{f}}^2 + C_2}, C_2 = (K_2L)^2 \quad (26)$$

where σ_f and $\sigma_{\hat{f}}$ are the standard deviation of the two images respectively, and $K_2 \ll 1$ is a constant.

$s(f, \hat{f})$ is the structure function, it is defined as:

$$s(f, \hat{f}) = \frac{\sigma_{f\hat{f}} + C_3}{\sigma_f\sigma_{\hat{f}} + C_3} \quad (27)$$

where $C_3 = C_2/2$.

4.1 Removing text coverage

In some shared pictures or documents, special marks or text are often added, which will affect the use of pictures or documents to a certain extent. Therefore, we often need to remove these marks or text from images. In this group of experiments, we artificially generate some texts and add them to each image, obtaining the damaged images we are going to restore, as shown in Fig. 5.

We use each method to remove the covered text from these images. Finally, we respectively calculate the PSNR and SSIM between the original images and the restored images to quantitatively compare the performance of each method. The PSNR of each method is shown in Table 1, where the maximum PSNR value of each image is indicated in bold.

The SSIM of each method is shown in Table 2, where the maximum SSIM value of each image is indicated in bold. It should be noted that because the value of SSIM is between 0 and 1, we reserve five decimal places for better differentiation.

In Table 1, BSCB method, DCT method and NLPB method did not obtain the highest PSNR value on all images. TV method obtained the highest PSNR value on Camera-man and Couple, and K-SVD method obtained the highest PSNR value on Baboon and Barbara. The proposed method obtained the highest PSNR value on three images (Couple, House and Lena). In Table 2, the proposed method obtained the highest SSIM value on all images. It can be seen from the data in Table 1 and Table 2 that our method has achieved better restoration effects compared with other methods. The reason is that our method classifies the sample patches and learns the corresponding over-complete dictionaries according to different classes, so that the over-complete dictionaries can adapt to different local features. Thus the proposed method can better restore the texture or structure information of the damaged image.

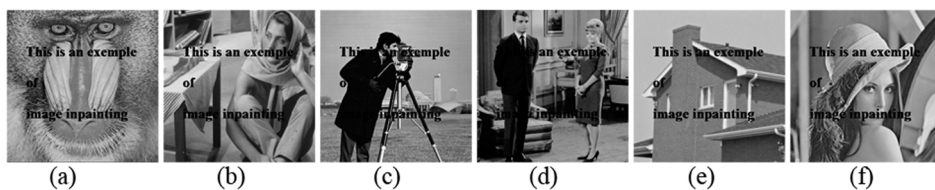


Fig. 5 Six images damaged by text

Table 1 The PSNR of each method for removing text coverage (Unit: dB)

	BSCB	TV	DCT	K-SVD	NLPB	Proposed method
Baboon	43.17	44.30	44.57	44.89	43.81	44.76
Barbara	45.92	47.14	47.34	48.25	47.39	48.17
Cameraman	45.34	46.91	45.72	45.59	46.87	45.63
Couple	45.12	46.71	46.67	46.53	46.33	46.71
House	45.59	48.30	47.43	49.21	49.76	50.11
Lena	44.76	46.17	46.91	47.13	46.44	47.19

4.2 Removing scratches

In the old photos or precious historical documents, some scratches often appear due to long-term use or limitations of the preservation environment. These scratches have a great influence on the historical value of photos or materials. Therefore, we often need to remove scratches from images. In this group of experiments, we artificially generate broken scratches and add them to each image, obtaining the damaged images we are going to restore, as shown in Fig. 6.

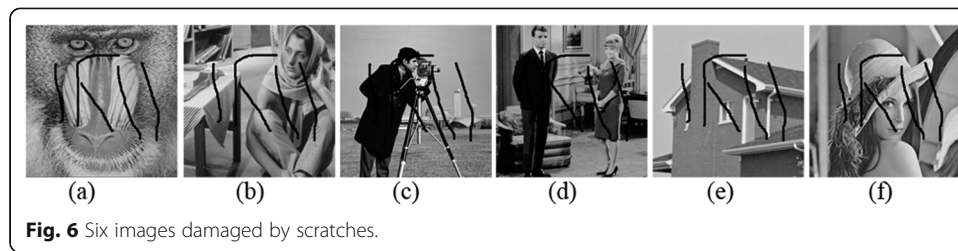
We use each method to remove the scratches from the images. Finally, we respectively calculate the PSNR and SSIM between the original images and the restored images to quantitatively compare the performance of each method. The PSNR of each method is shown in Table 3, where the maximum PSNR value of each image is indicated in bold.

The SSIM of each method is shown in Table 4, where the maximum SSIM value of each image is indicated in bold. It should be noted that because the value of SSIM is between 0 and 1, we reserve five decimal places for better differentiation.

As can be seen from Table 3, BSCB method and method B did not obtain the highest PSNR value on all images. TV method obtained the highest PSNR value on Couple. The proposed method obtained the highest PSNR value on Barbara. K-SVD method obtained the highest PSNR value on Baboon and Lena, and NLPB method obtained the highest PSNR value on Cameraman and House. It can be seen from Table 4 that, BSCB method, TV method and DCT method did not obtain the highest SSIM value on all images. K-SVD method obtained the highest SSIM value on Lena. NLPB method obtained the highest SSIM value on Cameraman. However, the proposed method obtained the highest SSIM value on four images (Baboon, Barbara, Couple and House). This shows that although our method does not obtain the highest PSNR value on some images, it can obtain the most similar structural information on most images, and can effectively restore the texture or edge structure of damaged images.

Table 2 The SSIM of each method for removing text coverage

	BSCB	TV	DCT	K-SVD	NLPB	Proposed method
Baboon	0.97555	0.98609	0.98811	0.98882	0.98466	0.98918
Barbara	0.98398	0.99249	0.99375	0.99494	0.99337	0.99528
Cameraman	0.98559	0.99226	0.99166	0.99436	0.99253	0.99502
Couple	0.98407	0.99180	0.99268	0.99424	0.99130	0.99450
House	0.98791	0.99525	0.99427	0.99570	0.99543	0.99681
Lena	0.98283	0.99183	0.99184	0.99447	0.99186	0.99462



4.3 Filling holes

The long-term preservation or repeated use of photos or documents will cause some pixels to be lost, forming some voids in the photos or documents. Therefore, we often need to restore the holes in images. In this group of experiments, we artificially generate some holes and add them to each image, obtaining the damaged images we are going to restore, as shown in Fig. 7.

We use each method to remove the holes from the images. Finally, we respectively calculate the PSNR and SSIM between the original images and the restored images to quantitatively compare the performance of each method. The PSNR of each method is shown in Table 5, where the maximum PSNR value of each image is indicated in bold.

The SSIM of each method is shown in Table 6, where the maximum SSIM value of each image is indicated in bold. It should be noted that because the value of SSIM is between 0 and 1, we reserve five decimal places for better differentiation.

In Table 5, BSCB method and K-SVD method did not obtain the highest PSNR value on all images. DCT method obtained the highest PSNR value on Baboon. NLPB method obtained the highest PSNR value on Barbara. TV method obtained the highest PSNR value on Cameraman and Couple. The proposed method obtained the highest PSNR value on House and Lena. In Table 6, BSCB method, TV method, DCT method, and NLPB method did not obtain the highest SSIM value on all images. K-SVD method obtained the highest SSIM value on Baboon, Barbara and Lena. The proposed method obtained the highest SSIM value on Cameraman, Couple and House. It can be seen from the data in Table 5 and Table 6 that compared with other method, the proposed method can obtain better results, retain better texture or structure information, and make the repaired image more consistent with the requirements of visual consistency.

4.4 Discussion

Based on the data in Table 1-6, we calculated the average values of PSNR and SSIM for each method, as shown in Table 7.

Table 3 The PSNR of each method for removing scratches (Unit: dB)

	BSCB	TV	DCT	K-SVD	NLPB	Proposed method
Baboon	43.56	43.82	43.87	44.36	43.43	44.08
Barbara	44.79	46.44	46.03	47.06	46.66	47.42
Cameraman	47.52	47.74	46.94	45.85	48.14	46.37
Couple	46.06	47.18	46.01	46.24	46.39	46.79
House	46.93	47.85	46.33	47.49	49.49	49.08
Lena	46.18	46.84	45.83	47.03	46.82	46.62

Table 4 The SSIM of each method for removing scratches

	BSCB	TV	DCT	K-SVD	NLPB	Proposed method
Baboon	0.97895	0.98435	0.98444	0.98474	0.98329	0.98661
Barbara	0.98765	0.99214	0.99129	0.99346	0.99322	0.99467
Cameraman	0.98690	0.99225	0.99234	0.99261	0.99406	0.99367
Couple	0.98716	0.99133	0.99122	0.99370	0.99033	0.99456
House	0.98886	0.99395	0.99322	0.99470	0.99618	0.99667
Lena	0.98833	0.99150	0.98993	0.99328	0.99161	0.99246

For a more intuitive comparison, we display the data in Table 7 graphically, as shown in Fig. 8 and Fig. 9.

It can be seen from Table 7, Fig. 8 and Fig. 9 that the PSNR and SSIM of the first three methods (BSCB method, TV method, and DCT method) are smaller than those of the latter three methods (K-SVD method, NLPB method, and proposed method). The first three methods are original and classic image restoration methods. The latter three methods are improved image restoration methods proposed by researchers. In particular, the proposed method achieves the highest PSNR and SSIM among all methods.

The PSNR and SSIM of BSCB method are the lowest, which means that its restoration effect is worse than other methods. The reason is that it spreads the effective information smoothly to the inside of the damaged area according to the direction of the isophotes, which will lead to the loss of texture details. However, as the most classical method of image restoration, it provides the idea of automatic image restoration, which makes image restoration become a research hotspot of computer vision.

The restoration effect of TV method is better than BSCB method, the reason is that by solving the Partial Differential Equation, the effective information is anisotropically diffused and the restoration effect is improved to a certain extent.

As a simple method based on sparse representation, the PSNR and SSIM of DCT method are smaller than those of K-SVD method. The reason is that although the construction of DCT dictionary is relatively simple, its adaptability is poor and it cannot adapt to different features of images.

Compared with the first three methods, the PSNR and SSIM of the latter three methods are higher. The reason is that these three methods have improved the classic method from different aspects and improved the effect of image restoration. The K-SVD method has achieved higher PSNR values and obtained a better restoration effect. The reason is that it uses the sample patches extracted from the original image to learn the over-complete dictionary, which can improve the ability of the over-complete

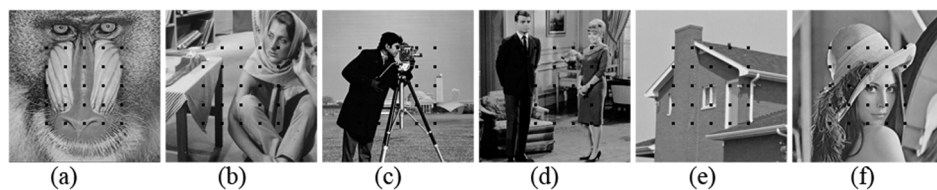
**Fig. 7** Six images damaged by holes

Table 5 The PSNR of each method for removing scratches (Unit: dB)

	BSCB	TV	DCT	K-SVD	NLPB	Proposed method
Baboon	50.04	52.15	52.18	51.95	51.28	52.15
Barbara	50.63	52.72	54.81	54.24	55.17	54.71
Cameraman	52.07	54.58	52.25	53.34	53.81	53.22
Couple	51.66	54.10	52.36	53.98	52.38	53.87
House	50.73	53.98	52.72	54.14	55.61	56.41
Lena	51.71	52.30	52.82	54.12	52.79	54.20

dictionary to represent the natural image, so that it can adapt to the images of different features and improve the reconstruction effect.

The NLPB method also obtained better restoration result. The reason is that it improves the patch-based restoration method from many aspects. It uses PatchMatch algorithm to search for nearest neighbors, uses a weighted mean scheme to improve the comparison of textured patches, and uses initialization and a multi-scale scheme to achieve more satisfactory solutions. Based on the above improvements, it has also achieved higher PSNR and SSIM.

Compared with other methods, the proposed method has achieved good restoration effect, and the PSNR and SSIM are the highest of all methods. The reason is that, we use local features to divide the image patches into different classes, and then learn the corresponding over-complete dictionary according to different classes, which can further improve the adaptability of the over-complete dictionary, so the image patch can be better sparsely expressed. Therefore, texture or structure information can be better preserved in the restored image.

In addition, it should be noted that in recent years, with the rapid development of deep learning technology, researchers have introduced deep learning into the field of image inpainting, and proposed some methods based on deep neural network, which provides a new and broad idea for the research of image inpainting [31]. Yu *et al* [32] proposed a coarse-to-fine generative image inpainting method with a novel contextual attention module. The contextual attention module significantly improves image inpainting results by learning feature representations for explicitly matching and attending to relevant background patches. Sagong *et al* [33] proposed a novel network structure called PEPSI. They adopted a structure consisting of a single shared encoding network and a parallel decoding network to reduce the number of convolution operations. Zeng *et al* [34] proposed a Pyramid-context ENcoder Network (PEN-Net) for

Table 6 The SSIM of each method for removing scratches

	BSCB	TV	DCT	K-SVD	NLPB	Proposed method
Baboon	0.99640	0.99771	0.99795	0.99811	0.99746	0.99808
Barbara	0.99700	0.99854	0.99919	0.99935	0.99922	0.99929
Cameraman	0.99588	0.99770	0.99748	0.99826	0.99817	0.99836
Couple	0.99711	0.99854	0.99866	0.99896	0.99813	0.99898
House	0.99695	0.99880	0.99851	0.99900	0.99895	0.99934
Lena	0.99526	0.99762	0.99794	0.99889	0.99841	0.99881

Table 7 Average PSNR and SSIM of each method

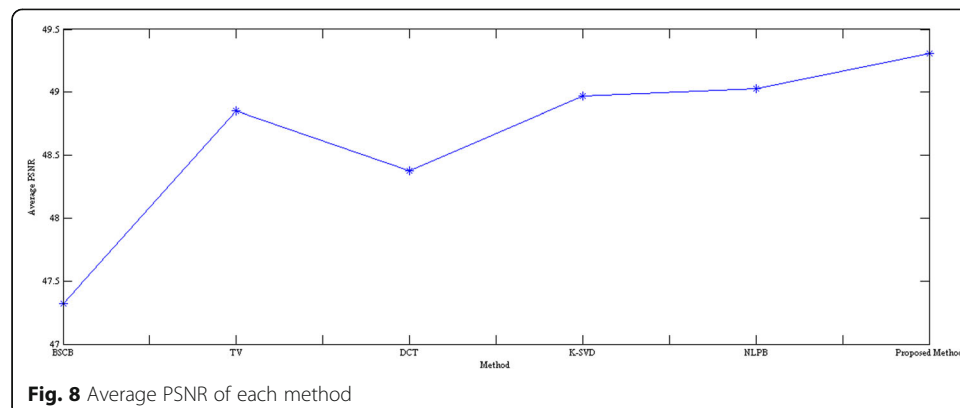
	BSCB	TV	DCT	K-SVD	NLPB	Proposed method
PSNR	47.32	48.85	48.38	48.97	49.03	49.31
SSIM	0.98869	0.99356	0.99358	0.99487	0.99379	0.99538

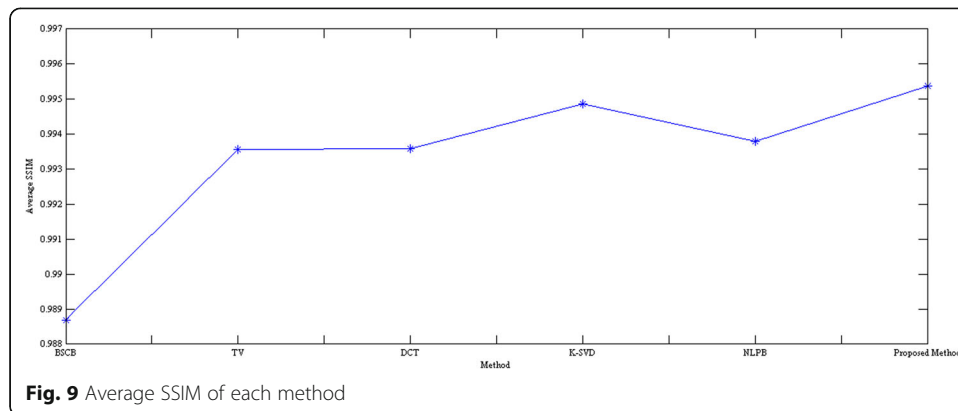
image inpainting by deep generative models. They used a pyramid-context encoder to progressively learn region affinity by attention from a high-level semantic feature map and transfer the learned attention to the previous low-level feature map. Besides, there are seven papers on image inpainting methods based on deep learning at the 2020 CVPR.

Also, it should be mentioned that in this article we did not compare our method with the method based on deep learning. We think that the basic ideas of these two types of methods are different. The method based on sparse representation calculates the sparse coding of the damaged patch on the over-complete dictionary, and then reconstruct the damaged patch using the sparse coding and the over-complete dictionary, while the method based on deep learning uses a large number of real images to train the generative model and the discriminative model, so that the deep network can learn the feature distribution of the real images, and then uses the generative model to automatically generate the image of the damaged region to achieve the purpose of image restoration. However, we have realized that deep neural networks, especially generative adversarial networks, have very powerful capabilities in image restoration. Therefore, we have read a lot of references and started to conduct in-depth research on restoration methods based on deep learning, hoping to further improve the restoration effect of large-scale damaged images.

5 Conclusions

In order to improve the adaptability of the over-complete dictionary and effectively improve the restoration effect of sparse-representation-based method, we proposed a novel method based on feature classification learning. We performed singular value decomposition on the local gradient vector of image patch, and according to the relationship between the main orientation and the secondary orientation, we distinguish edge patches from texture patches effectively. Then, based on sample patches with different

**Fig. 8** Average PSNR of each method



features, we learn over-complete dictionaries that adapt to different features. Finally, we use the obtained over-complete dictionary to sparsely encode and reconstruct the damaged image patches. Simulation results demonstrate the feasibility and effectiveness of our method. In the next study, we will conduct in-depth research on inpainting methods based on generative adversarial networks to further improve the restoration effect for object removal.

Abbreviations

KSVD: K-Singular Value Decomposition OMP: Orthogonal Matching Pursuit TV: Total Variation CDD: Curvature Driven Diffusion MCAM: Morphological Component Analysis PSNR: Peak Signal-to-Noise Ratio SSIM: Structure Similarity NLPB: Non-Local Patch-Based Image Inpainting

Acknowledgements

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

Authors' contributions

All authors take part in the discussion of the work described in this paper. All authors read and approved the final manuscript.

Funding

The research is supported by National Natural Science Foundation of China (Grant: 61703363), Scientific and Technological Innovation Programs of Higher Education Institutions in Shanxi (Grant: 2019L0855, 2020L0572), Scientific Research Project of Yuncheng University (Grant YQ-2017027, XK-2018034, CY-2019025).

Availability of data and materials

Please contact authors for data requests.

Competing interests

The authors declare that they have no competing interests.

Received: 7 February 2020 Accepted: 18 September 2020

Published online: 04 November 2020

References

1. C. Guillemot, O. Le Meur, Image inpainting: Overview and recent advances. *IEEE Signal Process. Mag.* **31**(1), 127–144 (2014)
2. N. Zhang, H. Ji, L. Liu, et al., Exemplar-based image inpainting using angle-aware patch matching. *EURASIP J. Image Video Process.* **70**, 1–13 (2019)
3. X. Dong, J. Dong, G. Sun, et al., Learning-based texture synthesis and automatic inpainting using support vector machines. *IEEE Trans. Ind. Electron.* **66**(6), 4777–4787 (2019)
4. Q. Guo, S. Gao, X. Zhang, et al., Patch-based image inpainting via two-stage low rank approximation. *IEEE Trans. Vis. Comput. Graph.* **24**(6), 2023–2036 (2018)
5. F. Yao, Damaged region filling by improved criminisi image inpainting algorithm for thangka. *Clust. Comput.* **6**, 1–9 (2018)
6. B.V. Rathish Kumar, A. Halim, A linear fourth-order PDE-based gray-scale image inpainting model. *Comput. Appl. Math.* **38**(6), 1–21 (2019)
7. M. Bertalmio, G. Sapiro, V. Caselles, et al. Image inpainting. *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 417–424 (2000).
8. T.F. Chan, J. Shen, Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* **62**(3), 1019–1043 (2002)

9. T.F. Chan, J. Shen, Nontexture inpainting by curvature-driven diffusions. *J. Vis. Commun. Image Represent.* **12**(4), 436–449 (2001)
10. H. Wang, Y. Cai, R. Liang, et al., Exemplar-based image inpainting using structure consistent patch matching. *Neurocomputing* **269**, 401–410 (2017)
11. A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. Image Process.* **13**(9), 1200–1212 (2004)
12. A. Wong, J. Orchard. A Nonlocal-means approach to exemplar-based inpainting. 15th IEEE International Conference on Image Processing. 2600–2603 (2008).
13. A.A. Efros, T.K. Leung, Texture synthesis by non-parametric sampling. *IEEE Int Conference Comput Vision*, 1033–1038 (1999)
14. L. Zhang, M. Chang, A novel image inpainting method for object removal based on structure sparsity. *Int J Performabil Eng.* **14**(11), 2777–2788 (2018)
15. J. Mo, Y. Zhou. The research of image inpainting algorithm using self-adaptive group structure and sparse representation. *Cluster Computing*. 1–9 (2018).
16. J. Zhang, D. Zhao, W. Gao, Group-based sparse representation for image restoration. *IEEE Trans. Image Process.* **23**(8), 3336–3351 (2014)
17. Z. Zha, X. Yuan, B. Wen, et al., Image restoration using joint patch-group-based sparse representation. *IEEE Trans. Image Process.* **29**, 7735–7750 (2020)
18. Z. Zha, X. Yuan, J. Zhou, et al., Image restoration via simultaneous nonlocal self-similarity priors. *IEEE Trans. Image Process.* **29**, 8561–8576 (2020)
19. M. Aharon, M. Elad, A. Bruckstein, K-SVD: An algorithm for designing over-complete dictionaries for sparse representation. *IEEE Trans. Signal Process.* **54**(11), 4311–4322 (2006)
20. M. Elad, *Sparse and Redundant Representations from Theory to Applications in Signal and Image Processing* (Springer, New York, 2010)
21. J. L. Starck, Y. Moudden, J. Bobin, et al. Morphological component analysis. *Proceedings of International Society for Optics and Photonics*. 5914 (2005).
22. M. Elad, J.L. Starck, P. Querre, et al., Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Appl. Comput. Harmon. Anal.* **19**(3), 340–358 (2005)
23. B. Shen, W. Hu, Y. Zhang, et al. Image inpainting via sparse representation. *IEEE International Conference on Acoustics, Speech and Signal Processing*. 697–700 (2009).
24. Z. Xu, J. Sun, Image inpainting by patch propagation using patch sparsity. *IEEE Trans. Image Process.* **19**(5), 1153–1165 (2010)
25. J. G. Shi, C. Qi. Sparse modeling based image inpainting with local similarity constraint. *Proceedings of 2013 IEEE International Conference on Image Processing, Melbourne*. 1371–1375 (2013).
26. A. Newson, A. Almansa, Y. Gousseau, et al. Non-local patch-based image inpainting, *Image Processing On Line*. 7, 373–385 (2017).
27. F. Li, T. Zeng, A universal variational framework for sparsity-based image inpainting. *IEEE Trans. Image Process.* **23**(10), 4242–4254 (2014)
28. M. Ghorai, S. Mandal, B. Chanda. Patch sparsity based image inpainting using local patch statistics and steering kernel descriptor. *Proceedings of the 23rd International Conference on Pattern Recognition, Cancun*. 781–786 (2017).
29. L. Zhang, B. Kang, B. Liu, et al., A new inpainting method for object removal based on patch local feature and sparse representation. *Int J Innov Comput Inform Control* **12**(1), 113–124 (2016)
30. X. G. Feng, P. Milanfar. Multiscale principal components analysis for image local orientation estimation. *Proceedings of the 36th Asilomar Conference on Signals, Systems and Computers*. 1, 478–482 (2002).
31. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio. Generative Adversarial Networks, *Advances in Neural Information Processing Systems*, no. 6, pp. 1–9, 2014.
32. J. Yu, Z. Lin, J. Yang, et al. Generative image inpainting with contextual attention, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 5505–5514 (2018).
33. M. Sagong, Y. Shin, S. Kim, et al. PEPSI: Fast image inpainting with parallel decoding network, 2019 International Conference on Computer Vision and Pattern Recognition. 11360–11368 (2019).
34. Y. Zeng, J. Fu, H. Chao, et al. Learning pyramid-context encoder network for high-quality image inpainting, 2019 International Conference on Computer Vision and Pattern Recognition. 1486–1494 (2019).

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.