THE PUBLISHING HOUSE
OF THE ROMANIAN ACADEMY

PROCEEDINGS OF THE ROMANIAN ACADEMY, Series A,
Volume 27, Number 1/2026, pp. 89–100
DOI:

IMAGE INPAINTING APPROACH FOR DUNHUANG MURALS BASED ON DYNAMIC RADIUS GUIDANCE AND IMPROVED FAST MARCHING METHOD ARCHITECTURE

Guozheng LIU[1], Yinglei GAO[2], Wanlu REN[1], Xuechong ZHANG [1], Xiaotong LIU[1]

[1] Shandong Jianzhu University, School of Information and Electrical Engineering, Jinan 250101, China
[2] Dalhousie University, Faculty of Agriculture, Truro, NS B2N 5E3, Canada
Corresponding author: Yinglei GAO, E-mail: yn441332@dal.ca

Abstract. Limited sample data due to the difficulty in digital acquisition of Dunhuang murals renders existing deep learning-based inpainting approaches prone to overfitting, causing color deviation and texture distortion in inpainted images. To address these challenges, this paper proposes DR-IFMM, an inpainting approach based on a dynamic radius strategy and an improved Fast Marching Method (IFMM) architecture. DR-IFMM addresses these challenges by decomposing the inpainting process into several key stages. First, it adaptively computes two optimal radii based on the pixel density of damaged regions within the neighborhood of the target pixel, dynamically optimizing the inpainting of irregular and large-area defects. Second, the weight calculation rules of the FMM algorithm are refined to improve the accuracy of boundary and texture line inpainting. Finally, an image recomposition strategy integrates global structure and local details, yielding coherent textures and fine details. Experimental results on the Dunhuang mural dataset demonstrate that DR-IFMM outperforms competing approaches in terms of SSIM, PSNR, and LPIPS, effectively recovering the original appearance of damaged murals. This validates the practical value of DR-IFMM in Dunhuang mural inpainting, contributing to the digital preservation and inheritance of cultural heritage.

Keywords*:* Dunhuang murals, image inpainting, dynamic radius, IFMM, image recomposition.

1. INTRODUCTION

The Dunhuang murals are an important part of world cultural heritage. They are not only outstanding representatives of ancient Chinese art but also precious witnesses of the multi-civilization exchange along the Silk Road [1-3]. However, many murals have suffered damage such as paint peeling, cracks, and mildew due to erosion by wind and rain, natural aging, and human destruction. These damages pose challenges for the preservation and research of the murals [4]. Traditional physical inpainting may cause secondary damage to the murals. They also find it difficult to meet the demands of large-scale and high-precision protection [5]. Therefore, the use of digital inpainting technology for virtual inpainting and permanent archiving of the Dunhuang murals has become a key technological support for the preservation and inheritance of cultural heritage. This approach holds profound cultural and academic significance [6].

The goal of image inpainting is to reasonably fill the missing areas of the damaged region [7]. Traditional image inpainting approaches usually use local image information or statistical features to inpaint the missing pixels. The total variation (TV) model [8] is a diffusion-based image inpainting approach. It introduces the Euler-Lagrange equation to gradually diffuse image information from the edge of the missing region into the interior. However, the TV model uses a shortest-path approach for image inpainting, which results in visible traces in the inpainted damaged areas. Li *et al*. proposed an improved K-means textile cultural relic image inpainting algorithm based on the Criminisi algorithm [9]. The approach calibrates the edges of pattern designs and reduces the search area for matching blocks. This results in an inpainted structure that more closely resembles the ground-truth image. Furthermore, Zhang *et al*. proposed an improved Criminisi [10]. This approach uses Bezier curves to fit the edges of the missing areas. This approach solves the problem in the original Criminisi where the failure to recognize curved textures causes the confidence term to approach zero. It improves the quality of image inpainting. There are many traditional

image inpainting algorithms of this type. For example, Gaona *et al*. performed image inpainting by reconstructing multi-scale mathematical morphology [11]. Min *et al*. proposed a simplified fast image inpainting algorithm [12]. Bornemann *et al*. developed a fast non-iterative image inpainting method [13]. However, when dealing with irregular damage and large missing areas, such approaches cannot capture the global information of the image. This leads to missing texture structures and details in the inpainted image, causing a significant difference from the original image.

The rapid development of the computer industry has led to the gradual application of deep learning approaches in the field of image inpainting [14-16]. These approaches train models using large datasets, capturing global image information to address the challenges of complex texture and structural reconstruction. Suvorov *et al*. proposed the LaMa model [17]. The network architecture of this model uses fast Fourier convolutions, which capture the receptive field of the entire image. This improves the quality of inpainting and reduces the number of model parameters. Li *et al*. proposed the Multilevel Interactive Siamese Filtering (MISF), comprising the kernel prediction branch and the semantic image filtering branch [18]. The approach leverages effective semantic and image-level filling for high-fidelity inpainting. Xu *et al*. introduced a mural inpainting approach combining CSWin-Transformer and gated convolutions [19]. This approach connects convolutional and Transformer networks in parallel to resolve issues inherent in single-network models, such as high computational complexity, texture blurring, and insufficient feature extraction. Zhang *et al*. proposed the MMGInpainting [20]. This approach accepts both text and image modalities, guiding the inpainting process with a fusion of both. In addition, Li *et al*. devised a recurrent feature reasoning network [21]. This network achieved promising results in image inpainting. Compared to traditional inpainting approaches, deep learning approaches excel at capturing texture and structural information [22]. However, this advantage comes with drawbacks, including more complex model structures and longer training times. These factors impose higher demands on computer performance, which in turn limits the environments in which these advanced inpainting approaches can be deployed.

Achieving high-quality image inpainting in the context of wavelet transform is an important topic. In traditional image inpainting algorithms, Yang introduced the dual-tree complex wavelet transform (CWT) to the field of image inpainting [23]. This approach leverages the approximate translation invariance, good directional selectivity, and relatively low redundancy of the CWT. Consequently, it achieves a favorable balance between computational efficiency and inpainting quality. With the development of deep learning approaches, wavelet transform has been further applied to network feature extraction and loss function construction. Huang *et al*. proposed a two-stage inpainting approach, combining the information of frequency and spatial domain [24]. The approach using Stationary Wavelet Transform achieved good results in highly structured and textured image inpainting. Cao *et al*. proposed a novel image inpainting algorithm WFIL-NET which is based on wavelet downsampling and frequency integrated learning module [25]. The approach uses a wavelet downsampling module in the encoder to effectively fill large and irregular missing areas and significantly enhance the visual quality of inpainted images. However, wavelet transform based image inpainting approaches also face challenges. Traditional iterative algorithms suffer from high computational complexity. They are time-consuming when processing large-sized images, which limits their practical applicability and efficiency.

The inpainting of Dunhuang mural images is particularly challenging when performed under small sample conditions. The paper proposes a mural image inpainting approach based on a dynamic radius strategy and the improved Fast Marching Method architecture. The approach does not require a large dataset for training. It demonstrates strong capabilities in inpainting damaged images without prior knowledge.

The main contributions of this paper are summarized as follows:

- An approach based on a dynamic radius strategy and the IFMM architecture is proposed for mural image inpainting, named improvement of fast marching method using dynamic radius (DR-IFMM). This approach focuses on damaged Dunhuang mural images to address the small sample image inpainting problem.
- In the proposed DR-IFMM, a dynamic radius strategy is introduced. This strategy extracts detail information from the neighborhood of the pixels to be inpainted. Two optimal radii are calculated, which are then used as parameters to generate two preliminary inpainted images. The weight calculation rules of the FMM algorithm are further improved by prioritizing the directional weight factor and selecting pixels highly correlated with the target pixel in texture direction for weight

calculation. Additionally, an image fusion method is employed to average the pixels of the two images, thereby preserving the global and detail features of the image.

- The method was evaluated on the Dunhuang mural image dataset. The results demonstrate that DR-IFMM achieves superior performance on the PSNR, SSIM, and LPIPS metrics.

The remainder of the paper is organized as follows. Section 2 introduces the proposed DR-IFMM approach, Section 3 explores experimental results, and Section 4 summarizes the study conclusions.

2. APPROACH

The DR-IFMM is an extension and improvement of the Fast Marching Method (FMM) [26]. A dynamic radius strategy is introduced. The weight calculation approach of the FMM algorithm is improved. This emphasizes the inpainting parameters and focuses on the directional weight factor. The proposed approach processes mural images in three stages to achieve the inpainting of damaged areas, highlighting their edge details. This process enhances the inpainting of mural texture and structural information. The image inpainting process is illustrated in Fig.1.

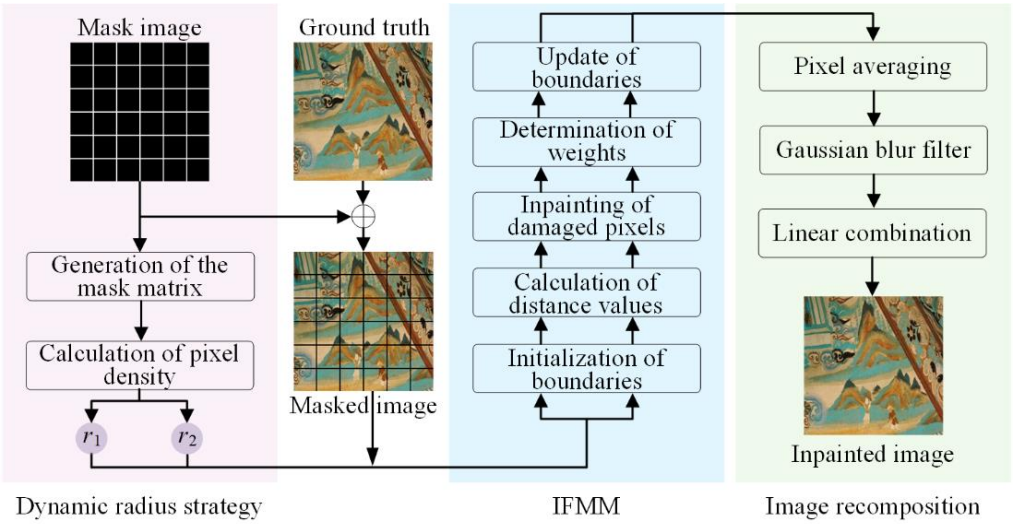

Fig. 1 – Inpainting process.

2.1. Dynamic radius strategy

The FMM algorithm can effectively inpaint image regions that are small and regularly shaped. However, when the damaged region is irregular and large, it fails to produce reasonable inpainting results using estimated radii. Consequently, a fixed inpainting radius cannot effectively address the inpainting of extensively damaged mural images. This paper introduces a dynamic radius strategy to overcome this limitation. This strategy determines two optimal radii based on the pixel density values in the neighborhood of the pixels to be inpainted. These radii are then used as parameters within the FMM for the inpainting process.

Calculating the pixel density values of the damaged region requires a binary mask matrix $\boldsymbol{M_c(x, y)}$. This matrix is generated for each pixel $M(x, y)$ within the mask image M. The matrix $\boldsymbol{M_c(x, y)}$ is used to identify the damaged region pixels. The generation rule for $\boldsymbol{M_c(x, y)}$ is shown in Eq. (1).

$$\boldsymbol{M_c}(x, y) = \begin{cases} 1, \text{if } M_i(x, y) = 255 \\ 0, \text{else} \end{cases} \tag{1}$$

In this case, x and y represent the horizontal and vertical coordinates of the image respectively. Each (x, y) corresponds to a pixel coordinate in the image. In the generated binary mask matrix, 1 represents damaged region pixels, and 0 represents normal region pixels. Let pixel P be any pixel on the boundary of the damaged region to be inpainted, and r is the radius of a circular neighborhood centered at pixel P, as shown

in Fig. 2. To optimize the computational process, a square window of side length (2*r*+1), centered at pixel *P*, is used to calculate the pixel density within the damaged region. The pixel density of the damaged region for pixel *P* at different values of *r* is calculated using Eq. (2).

$$D_r = \frac{\mu_r}{(2r+1)^2}, r \in \{1,2,3,4,5,6\} \tag{2}$$

In this case, μ_r represents the number of damaged pixels within a rectangular area of size (2*r*+1) × (2*r*+1). This corresponds to the number of elements in the mask matrix $\boldsymbol{M_c}$ **(*x*, *y*)** that equal 1 within the rectangular region. The two values of *r* that minimize the pixel density value D_r of the damaged region are selected as the inpainting radii for the pixel *P*.

The optimization of the algorithm's runtime involves the generation of a preprocessing matrix $\boldsymbol{Q(x, y)}$ from the mask matrix $\boldsymbol{M_c(x, y)}$. This is done after a value for *r* (*r* ∈[1,6]) has been selected. The preprocessing matrix represents the set of all elements from (0,0) to (*x*, *y*) in the mask matrix. The number of elements *N* in the preprocessing matrix is calculated. This structure enables the rapid calculation of the number of pixels within any rectangular submatrix. As shown in Fig. 2, the pixel count of rectangle *ABCD* is calculated. Based on the preprocessing matrix for each point, the cumulative element counts from the (0, 0) to the corresponding points are calculated as N_A, N_B, N_C, and N_D. These values are substituted into Eq. (3) to determine N_{sum}, the total number of pixels within the rectangle *ABCD*. When only considering the number of elements with values of 1 or 0 in the preprocessing matrix, the number of damaged region pixels or normal region pixels within rectangle *ABCD* can also be calculated.

$$N_{sum} = N_C - N_D - N_B + N_A \tag{3}$$

2.2. Improved Fast Marching Method

The core of the inpainting algorithm is based on the improved Fast Marching Method (IFMM) to calculate the distance *T*. Following the principles of gradient propagation and distance prioritization, the damaged region is then inpainted using known pixels.

As shown in Fig. 2, the original image is initially divided into a known region *A* and a damaged region Ω. The inpainting process begins at any pixel *P* on the boundary ∂Ω, from which the entire damaged region Ω is gradually inpainted. The pixels on the boundary are considered the starting points, with their distance *T* set to 0. The calculation of *T* is based on the Eikonal equation.

$$|\nabla T(x,y)| = 1 \text{ on } \Omega, \text{with} T(x,y) = 0 \text{ on } \partial\Omega \tag{4}$$

In this context, *T*(*x*, *y*) denotes the shortest distance from each pixel to the boundary ∂Ω, which dictates the inpainting order. $\nabla T(x,y)$is the gradient of *T*(*x*, *y*), representing the rate of change of the distance. A value of "1" indicates that the rate of change of the distance is constant for each pixel, while "0" indicates that the pixel is located on the initial boundary. The inpainting order is determined based on the distance *T*, in the sequence from closest to furthest, as shown in Fig. 2.

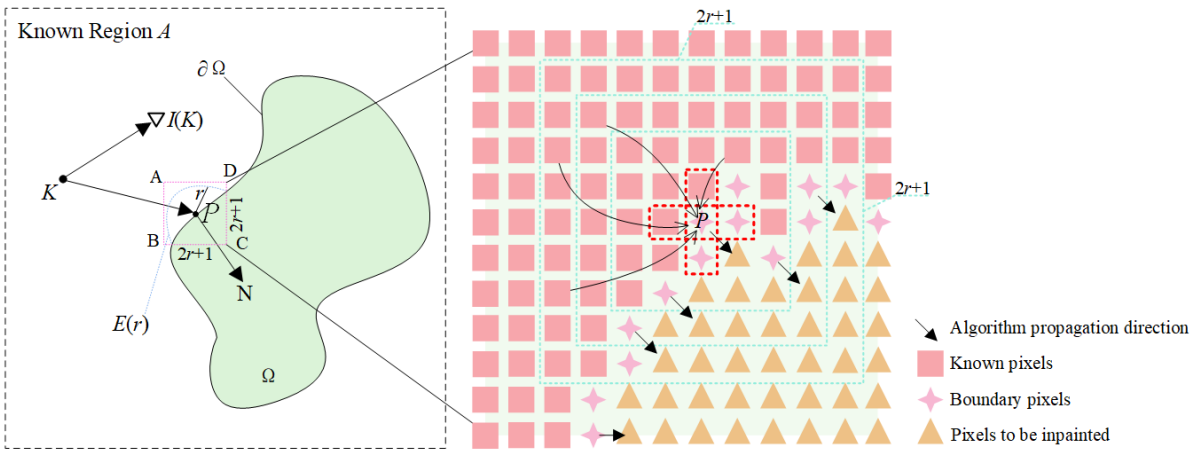

Fig. 2 – The principle of IFMM algorithm for inpainting.

Let any pixel to be inpainted be denoted as P. A known pixel K within a neighborhood of radius r is used to calculate the pixel approximation for P. This approximation value, derived from pixel K, is denoted as $I_K(P)$.

$$I_K(P) = I(K) + \nabla I(K)(P - K) \tag{5}$$

In this case, $I(K)$ represents the pixel value of the known point pixel K. The image gradient at K is denoted as$\nabla I(K)$, which represents the rate of change of the pixel values. (P-K) represents the displacement vector from the known pixel K to the pixel P to be inpainted. $\nabla I(K)(P - K)$denotes the change in pixel values along the gradient direction over the displacement vector (P-K). The value of $I(P)$ is calculated by inpainting all known pixel points within the repair radius.

$$I(P) = \frac{\sum_{K \in E_r(P)} w(P,K)\,[I(K) + \nabla I(K)(P - K)]}{\sum_{K \in E_r(P)} w(P,K)} \tag{6}$$

In this context, $I(P)$ represents the value of the pixel to be inpainted. $E_r(P)$ refers to the neighborhood centered at point P with a radius of r. $w(P,K)$ is the weight function. It is used to weight the influence of different neighborhood pixels on point P.

$$w(P,K) = w_1(P,K) + w_2(P,K) \tag{7}$$

$$w_1(P,K) = dir(P,K) \cdot dst(P,K) \cdot lev(P,K) \tag{8}$$

$$w_2(P,K) = dst(P,K) \cdot lev(P,K) \tag{9}$$

In this context, $dir(P,K)$ represents the directional weight factor. It ensures that information propagates faster in the normal direction, or gradient direction. Pixels aligned with the edge direction contribute more to the inpainting process, as shown in Eq. (10). $dst(P,K)$ is the geometric distance weight factor. It weakens the influence of known pixels that are farther from the pixel to be inpainted, as shown in Eq. (11). $lev(P,K)$ is the contour line weight factor. It ensures that pixels closer to the contour lines contribute more to the inpainting process. This helps avoid abrupt changes and maintain smooth contours, as shown in Eq. (12).

$$dir(P,K) = \frac{P - K}{\|P - K\|} \cdot N(P) \tag{10}$$

In this case, $\|P\text{-}K\|$ represents the distance between pixel P and pixel K. $N(P)$ is the normal of the pixel to be inpainted, which refers to the gradient direction.

$$dst(P,K) = \frac{d_0^2}{\|P - K\|^2} \tag{11}$$

In this case, d_0 is the reference distance. It represents the range of influence that neighboring pixels have on the central pixel during the inpainting process. It is typically set to 1. $\|P\text{-}K\|$ is the distance between pixel P and pixel K. The greater the distance, the smaller the weight.

$$lev(P,K) = \frac{T_0}{1 + |T(P) - T(K)|} \tag{12}$$

In this case, $T(P)$ and $T(K)$ represent the distances of the pixel to be inpainted and the known pixel, respectively. T_0 is a reference distance constant. It is typically set to 1. The greater the difference between the distances, the greater the structural difference between the two pixels. As a result, the weight is smaller.

The weight function in the traditional FMM inpainting algorithm calculates the directional weight factor, geometric distance weight factor, and contour line weight factor for all pixels in the neighborhood. This type of fuzzy calculation introduces irrelevant information. As a result, it not only increases the algorithm's runtime but also leads to inaccurate inpainting of the damaged region's boundary and texture.

Complex textures in the neighborhood of a pixel to be inpainted significantly impact the inpainting outcome, necessitating a higher priority for the directional weight factor. In this paper, the directional weight factor is recalculated based on the original weight function. This enhances the algorithm's ability to inpaint textures. Neighboring pixels are selected, and those with a higher correlation in texture direction to the pixel to be inpainted are used in the weight calculation. Pixels with a stronger correlation are processed by Eq. (8), while the others are handled by Eq. (9). To select the most influential pixels, the gradient magnitude and

gradient direction values of each known pixel in the neighborhood are calculated, as shown in Eqs. (13) and (14).

$$G_a(x,y) = [[I(x+1,y) - I(x-1,y)]^2 + [I(x,y+1) - I(x,y-1)]^2]^{\frac{1}{2}} \tag{13}$$

$$G_d(x,y) = \arctan\frac{I(x+1,y) - I(x-1,y)}{I(x,y+1) - I(x,y-1)} \tag{14}$$

In this case, $G_a(x, y)$ represents the gradient magnitude, reflecting the strength of the gradient at each pixel. $G_d(x, y)$ denotes the gradient direction, which indicates the flow of the gradient. $I(x+1, y)$, $I(x-1, y)$, $I(x, y+1)$, and $I(x, y-1)$ are the neighboring pixels of $I(x, y)$. For pixel points located on the boundary of the damaged region $\partial\Omega$, as shown for pixel P in Fig. 3, the value of the desired pixel point is used to replace the missing pixel. $I(x, y)$ is used to replace the missing $I(x+1, y)$. Eqs. (13) and (14) are modified to Eqs. (15) and (16).

$$G_a(x,y) = [[I(x,y) - I(x-1,y)]^2 + [I(x,y+1) - I(x,y-1)]^2]^{\frac{1}{2}} \tag{15}$$

$$G_d(x,y) = \arctan\frac{I(x,y) - I(x-1,y)}{I(x,y+1) - I(x,y-1)} \tag{16}$$

In this manner, all known pixels are grouped into a gradient matrix. Based on the concept of uniform grouping, the obtained gradient matrix is discretized. The interval $[0, \pi]$ is uniformly divided into L sub-intervals. If $G_d(x, y)$ satisfies Eq. (17), the pixel is defined as belonging to the l-th sub-interval, where $l \in [1, L]$. Fig. 3 shows two examples for $L = 8$ and $L = 16$.

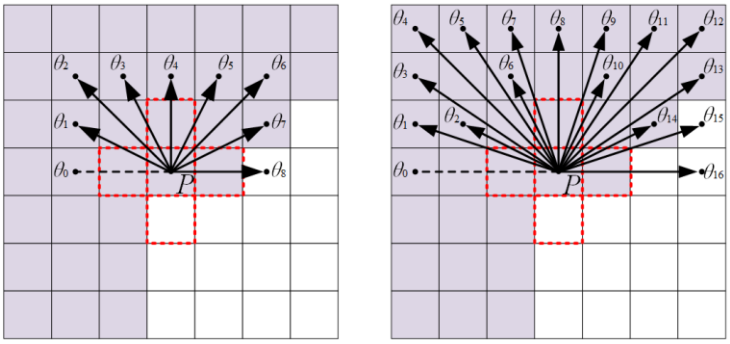

Fig. 3 – Calculation of IFMM direction weight.

Each subinterval represents a gradient direction. The direction with the largest sum of gradient magnitudes is selected as the gradient direction for the pixel to be inpainted. The selected pixel is then substituted into Eq. (8) to calculate w_1. The IFMM weight calculation approach reduces the number of pixels involved in the calculation. It also emphasizes the importance of complex textures in the inpainting process. This ensures the correct inpainting of the damaged region's boundary and texture.

$$G_d(x,y) \in \left(\frac{2(l-1)-1}{2L}\pi, \frac{2l+1}{2L}\pi\right) \tag{17}$$

After the boundary pixels have been inpainted, further inpainting of the damaged region pixels is required. The update of distance T is shown in Eq. (18).

$$\max(D^{-x}T, -D^{+x}T, 0)^2 + \max(D^{-y}T, -D^{+y}T, 0)^2 = 1 \tag{18}$$

In this case, $D^{-x}T(i, j)=T(i, j)-T(i-1, j)$ and $D^{+x}T(i, j)=T(i+1, j)-T(i, j)$ represent the finite differences in the horizontal direction. They calculate the distance variation from the neighboring pixels on both sides to the current pixel. Similarly, $D^{-y}T$ and $D^{-y}T$ represent the finite differences in the vertical direction. This approach employs the "upwind" difference scheme. The difference is computed in the direction of the fastest gradient change to update the distance values. This ensures that the distance values are updated only along the direction of information propagation, maintaining the monotonicity of the inpainting direction.

2.3. Image recomposition

From the above approach, it can be seen that the inpainted images $F_{r_1}(x, y)$ and $F_{r_2}(x, y)$ are obtained based on the repair radii r_1 and r_2. The two images are then averaged pixel-wise. This results in the preliminary inpainted image $F_r(x, y)$.

$$F_r(x, y) = \frac{F_{r_1}(x, y) + F_{r_2}(x, y)}{2} \tag{19}$$

In order to reduce the difference between $F_{r_1}(x, y)$ and $F_{r_2}(x, y)$ and further smooth the image, a 3×3 Gaussian low-pass blur filter is applied to $F_r(x, y)$. This results in the blurred image $F_{blur}(x, y)$. The final output image $F_{out}(x, y)$ is obtained using a linear combination. The calculation approach is shown in Eq. (20).

$$F_{out}(x, y) = (1 - \alpha)F_{blur}(x, y) + \alpha F_r(x, y), \alpha \in [0,1] \tag{20}$$

In this case, α is a constant. It is used to adjust the balance between $F_{blur}(x, y)$ and $F_r(x, y)$. The final output image $F_{out}(x, y)$ retains more details.

3. RESULTS AND ANALYSIS

3.1. Image dataset

The mural dataset is provided by the Dunhuang Mural Research Institute. The mural data are available at https://ip.e-dunhuang.com/. The images are uniformly cropped to a size of 256 px × 256 px. Subsequently, we applied a series of data augmentation techniques, namely scaling, rotation (90°, 180°, 270°), mirroring, and flipping. This expands the dataset to a total of 2000 images.

In order to evaluate the image inpainting quality of murals under different levels of damage, we generate corresponding mask images. The size of the mask image is the same as the mural image, both being 256 px × 256 px. The scratches take the following forms: cross lines, sparse grid lines, medium-density grid lines, high-density grid lines, and random lines. Except for the random lines, the scratch line width is set to 2 pixels. The random lines consist of 20 straight segments with a width of 3 pixels, automatically generated, and several curves with a width of 6 pixels, hand-drawn. The mask image is used to generate damaged rock painting images with varying levels of damage for subsequent experimental validation.

3.2. Contrast experiment

This paper uses three image quality evaluation metrics: Structural Similarity Index (SSIM), Peak Signal to Noise Ratio (PSNR), and Learned Perceptual Image Patch Similarity (LPIPS) [27]. A higher PSNR and SSIM value indicates better inpainting performance, while a lower LPIPS value indicates superior inpainting quality.

The SSIM measures image similarity in terms of luminance, contrast, and structure. It is formulated as Eq. (21).

$$SSIM(I, I_O) = l(I, I_O) \cdot c(I, I_O) \cdot s(I, I_O) \tag{21}$$

In this context, I and I_O represent the original image and the inpainted image. The terms $l(I, I_O)$, $c(I, I_O)$ and $s(I, I_O)$ denote the brightness, contrast, and structural similarities, respectively. These components are defined by Eqs. (22), (23), and (24).

$$l(I, I_O) = \frac{2\mu_I\mu_{I_O} + C_1}{\mu_I^2 + \mu_{I_O}^2 + C_1} \tag{22}$$

$$c(I, I_O) = \frac{2\sigma_I\sigma_{I_O} + C_2}{\sigma_I^2 + \sigma_{I_O}^2 + C_2} \tag{23}$$

$$s(I, I_O) = \frac{\sigma_{II_O} + C_3}{\sigma_I\sigma_{I_O} + C_3} \tag{24}$$

In this context, μ_I and μ_{I_O} represent the average gray levels of images I and I_O. The standard deviations of the gray levels for images I and I_O are denoted as σ_x and σ_{I_O}. The covariance between images I and I_O are represented by σ_{II_O}. Constants C_1, C_2, and C_3 are introduced to be much smaller than 1. These constants prevent the denominator from becoming 0.

PSNR is primarily used to measure the error between the inpainted image and the original image. It is defined by Eq. (25).

$$PSNR(dB) = 10\lg\left(\frac{MAX_I^2}{MSE}\right) \tag{25}$$

In this context, MAX_I represents the maximum possible value of image pixels, which is 255 for 8-bit images. MSE denotes the mean squared error of the image.

LPIPS is a deep learning based perceptual similarity metric for images. It employs a pre-trained deep convolutional neural network to extract image features and compute distances in the feature space. The core formula is defined by Eq. (26).

$$LPIPS(I, I_O) = \sum_{l=1}^{L} w_l \|\phi_l(I) - \phi_l(I_O)\|_2^2 \tag{26}$$

In this context, I and I_O represent the original image and the inpainted image. $\phi_l(\cdot)$ denotes the l-th perceptual feature extracted by a pre-trained convolutional neural network. $\|\cdot\|_2$ represents the L_2 norm, which is used to measure the distance between image features. w_l denotes the weight assigned to the given perceptual feature, which is used to weight the contribution of different features to the overall similarity. L is the total number of feature layers. In the actual computation process, both images are input into a pre-trained feature extraction network, such as VGG-16. Multi-layer feature maps are extracted from specific layers of the network. The features are then normalized. Subsequently, the differences between the normalized feature maps for each layer are calculated using the corresponding weight w_l. Finally, the results from all selected layers are weighted and summed to obtain the LPIPS value.

The experimental environment is Python3.7 and Cuda10.0, Windows10 professional 64-bit operating system, equipped with Intel(R) Core (TM) i5-8400 CPU @ 2.80GHz (16 GB memory) RAM and NVIDIA GeForce RTX3080 GPU. The image inpainting process of DR-IFMM is divided into three steps. In the first stage, the damaged image and the mask image are input. The damaged image remains unchanged, while the mask image is converted into a binary mask matrix $\boldsymbol{M_c(x, y)}$ using Eq. (1). For each pixel to be inpainted in $\boldsymbol{M_c(x, y)}$, two dynamic radii are generated for image inpainting according to Eqs (2) and (3). In the second stage, the inpainting order of the damaged pixels is determined according to Eq. (4). Taking an arbitrary point P on the boundary of the damaged region that is repaired first as an example, the gradient direction and magnitude of each known pixel within the dynamic radius of P are calculated using Eqs. (13) to (16). Eq. (17) discretizes the gradient direction into L subintervals. The direction with the highest total magnitude is selected as the gradient direction for the pixel P to be inpainted. All pixel points within the chosen gradient direction are substituted into Eq. (8) to calculate the weight w_1. The remaining known pixel points are substituted into Eq. (9) to calculate the weight w_2. Finally, the weights of the known pixel points in the neighborhood of point P are obtained using Eq. (7). These weights are then sequentially substituted into Eq. (6) to obtain the pixel value for point P. The damaged region pixels are traversed according to the inpainting order until the inpainting is completed. In the third stage, each pixel to be inpainted has two dynamic radii. Consequently, two preliminary inpainted images are obtained. The two images are then processed using pixel averaging fusion and Gaussian low pass filtering, as described in Eqs. (19) and (20). This process ultimately generates the final inpainted image.

Fig. 4 shows various mural images randomly selected for our experiments. These images display four types of line damage: cross lines, sparse grid lines, medium-density grid lines, and high-density grid lines. The inpainting results of eight approaches are presented: MSMM [11], SFIIM [12], COTR [13], ICriminisi [9], LaMa [17], MISF [18], RFR [21], and DR-IFMM. The original images are also shown. As seen in Fig. 4, all eight approaches are able to inpaint the damaged Dunhuang mural images.

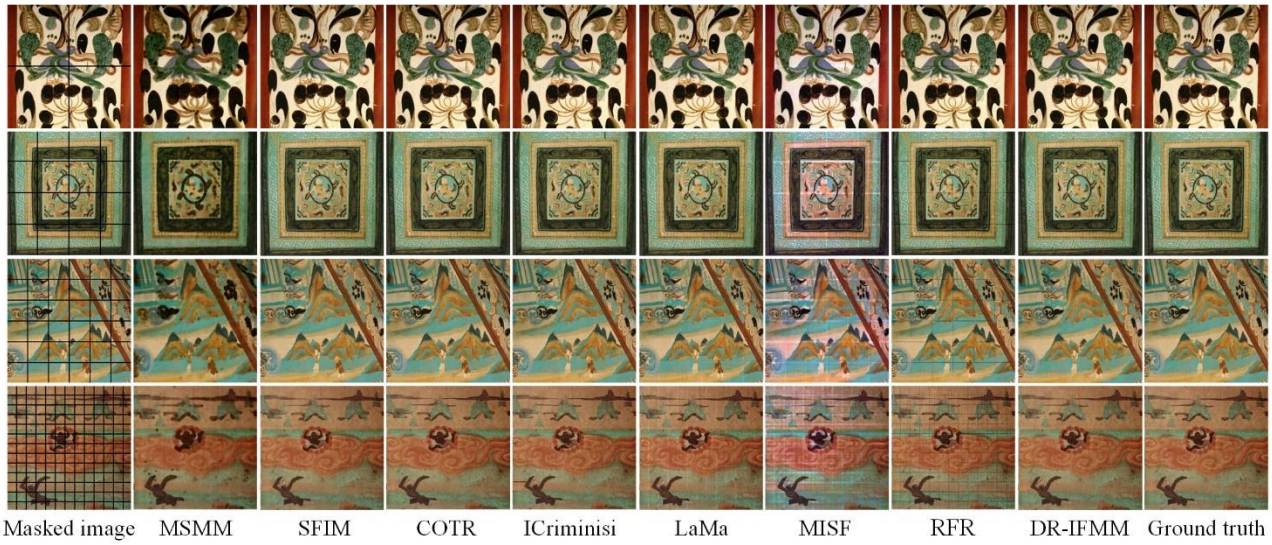

Fig. 4 – Inpainting results of Dunhuang murals.

As shown in Fig. 4, each approach shows different advantages and disadvantages in the mural image inpainting task. MSMM reconstructs the damaged area through morphological operations, which can largely inpaint the color and contrast. However, the inpainted result exhibits distortion, blurring, and loss of details. SFIIM leverages local information from known regions to inpaint damaged areas, achieving relatively accurate color inpainting. Nevertheless, it tends to blur some image details. COTR excels at preserving edge clarity but often introduces spatial distortion within the inpainted regions, leading to misalignments in the position and orientation of details. ICriminisi employs a sample-driven strategy for inpainting, which effectively inpaints color and texture, but leaves noticeable artifacts in the transition zones. LaMa utilizes fast Fourier convolution to capture global receptive fields, demonstrating remarkable performance on complex damages. However, in the case of high-density grid line damage, inpainting traces are still visible. MISF inpaints missing information via a multi-level interactive twin network filtering technique but suffers from noticeable color distortion. RFR uses a recurrent feature reasoning network to process large damaged areas, effectively inpainting texture and color, though inpainting traces are evident. In contrast, the approach proposed in this paper is able to effectively inpaint color, texture, and structural details under various damage conditions, demonstrating outstanding inpainting performance.

In order to further verify the effectiveness of the image inpainting approaches, SSIM, PSNR, and LPIPS are used to evaluate the inpainting capabilities of the eight approaches. Table 1 presents the evaluation results of the inpainting quality for four types of line scratches: cross lines, sparse grid lines, medium-density grid lines, and high-density grid lines, under regular scratch damage. As shown in Table 1, SSIM, PSNR, and LPIPS exhibit significant differences in repairing scratches of different line types. As the severity of image scratch damage increases, the difficulty of image inpainting also rises. The inpainting results for high-density grid line damage show that the SSIM, PSNR, and LPIPS values of the inpainted image are 9.62×10^{-1}, 3.81×10^{1}dB, and 2.14×10^{-2}, respectively, which are the best among the eight approaches. For other scratch types, except for cross-line damage where LaMa achieves the highest PSNR and LPIPS, our approach yields the best results across all three metrics. Combined with the analysis in Fig. 4, it can be concluded that the DR-IFMM outperforms MSMM, SFIIM, COTR, ICriminisi, LaMa, MISF, and RFR in image inpainting.

As shown in Table 1, under simple scratch damage, the SSIM, PSNR, and LPIPS evaluation metrics for LaMa, MISF, and RFR deep learning approaches are similar to those of the DR-IFMM. However, the proposed approach yields slightly higher numerical results for repairing different scratch damages, outperforming the deep learning approaches. This can be attributed to the following reasons. As damage complexity increases, deep learning approaches can still perform well in inpainting tasks but rely heavily on large-scale training datasets. The Dunhuang murals, being a world cultural heritage, are susceptible to damage during digital collection, which makes image acquisition difficult. Due to the lack of a large training dataset, the deep learning approaches show slightly worse results compared to the DR-IFMM algorithm. The

advantage of the proposed approach lies in its ability to achieve good inpainting results without model training or the need for powerful hardware. It can effectively address image inpainting without prior knowledge. Therefore, the proposed approach offers certain advantages in the inpainting of the Dunhuang murals.

Table 1

Quality evaluation of inpainted images under regular scratch conditions

Approach	Cross lines			Sparse grid lines			Medium-density grid lines			High-density grid lines		
	SSIM	PSNR/dB	LPIPS	SSIM	PSNR/dB	LPIPS	SSIM	PSNR/dB	LPIPS	SSIM	PSNR/dB	LPIPS
MSMM [11]	4.78×10^{-1}	1.54×10^{1}	1.69×10^{-1}	4.20×10^{-1}	1.81×10^{1}	2.15×10^{-1}	5.52×10^{-1}	1.97×10^{1}	1.61×10^{-1}	6.43×10^{-1}	2.37×10^{1}	1.83×10^{-1}
SFIIM [12]	9.96×10^{-1}	4.08×10^{1}	1.79×10^{-3}	9.84×10^{-1}	3.94×10^{1}	4.68×10^{-3}	9.78×10^{-1}	3.75×10^{1}	1.12×10^{-2}	9.52×10^{-1}	3.70×10^{1}	2.59×10^{-2}
COTR [13]	9.96×10^{-1}	4.13×10^{1}	1.27×10^{-3}	9.84×10^{-1}	3.92×10^{1}	4.71×10^{-3}	9.78×10^{-1}	3.74×10^{1}	8.59×10^{-3}	9.46×10^{-1}	3.67×10^{1}	2.57×10^{-2}
ICriminisi [9]	9.92×10^{-1}	3.76×10^{1}	3.89×10^{-3}	9.73×10^{-1}	3.55×10^{1}	1.06×10^{-2}	9.61×10^{-1}	3.46×10^{1}	1.83×10^{-2}	9.02×10^{-1}	3.29×10^{1}	5.06×10^{-2}
LaMa [17]	9.96×10^{-1}	**4.19×10^{1}**	**1.14×10^{-3}**	9.83×10^{-1}	3.91×10^{1}	4.10×10^{-3}	9.69×10^{-1}	3.67×10^{1}	1.23×10^{-2}	9.23×10^{-1}	3.54×10^{1}	5.03×10^{-1}
MISF [18]	9.57×10^{-1}	3.13×10^{1}	1.23×10^{-2}	9.14×10^{-1}	2.83×10^{1}	4.94×10^{-2}	8.66×10^{-1}	2.85×10^{1}	6.12×10^{-2}	7.67×10^{-1}	2.79×10^{1}	1.56×10^{-1}
RFR [21]	9.92×10^{-1}	3.80×10^{1}	4.00×10^{-3}	9.69×10^{-1}	3.56×10^{1}	1.97×10^{-2}	9.25×10^{-1}	3.23×10^{1}	6.77×10^{-2}	8.14×10^{-1}	3.00×10^{1}	2.30×10^{-1}
DR-IFMM	**9.97×10^{-1}**	4.18×10^{1}	1.25×10^{-3}	**9.86×10^{-1}**	**4.01×10^{1}**	**3.99×10^{-3}**	**9.82×10^{-1}**	**3.83×10^{1}**	**9.33×10^{-3}**	**9.62×10^{-1}**	**3.81×10^{1}**	**2.14×10^{-2}**

Under conventional conditions, mural image damage consists of irregular scratches or random disruptions. Therefore, two types of damaged images were created in this study. These images were used to test the inpainting capability of the DR-IFMM. The performance of the DR-IFMM was compared with that of seven other approaches: MSMM, SFIIM, COTR, ICriminisi, LaMa, MISF, and RFR. The results are shown in Fig. 5.

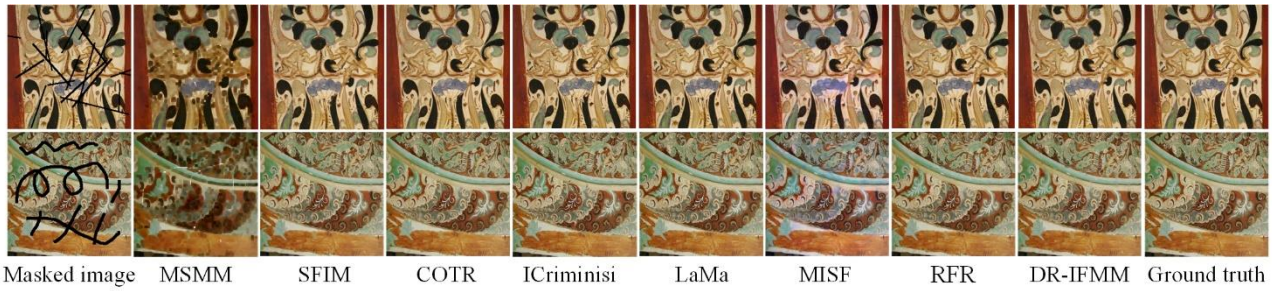

Fig. 5 – Inpainting results under random scratch damage.

As shown in Fig. 5, the first row contains 20 randomly generated straight line segments. Each segment has a width of 3 pixels. The second row contains manually drawn curved segments. Each curved segment has a width of 6 pixels. These segments present higher inpainting difficulty and complexity. By comparing the inpainting details, it is observed that the MSMM fails to inpaint the damaged regions. The other approaches are able to complete the inpainting task. The SFIIM, COTR, and ICriminisi approaches inpaint the color and detail information of the image. However, minor inpainting traces remain in the inpainted images. The LaMa, MISF, and RFR approaches effectively inpaint the missing texture and structural information. However, due to limitations in the dataset, the inpainted images show color distortion and missing structural information. The proposed approach produces inpainting results that are visually closest to the original image, demonstrating strong image inpainting capabilities.

Table 2
Quality evaluation of inpainted images under regular scratch conditions

Approach	Random straight-line scratches			Random hand-drawn scratches		
	SSIM	PSNR/dB	LPIPS	SSIM	PSNR/dB	LPIPS
MSMM [11]	3.89×10^{-1}	1.52×10^{1}	2.16×10^{-1}	3.13×10^{-1}	1.57×10^{1}	3.16×10^{-1}
SFIIM [12]	9.65×10^{-1}	3.17×10^{1}	2.30×10^{-2}	9.65×10^{-1}	3.33×10^{1}	1.95×10^{-2}
COTR [13]	9.67×10^{-1}	3.20×10^{1}	1.60×10^{-2}	9.65×10^{-1}	3.32×10^{1}	1.81×10^{-2}
ICriminisi [9]	9.45×10^{-1}	2.92×10^{1}	3.18×10^{-2}	9.42×10^{-1}	3.00×10^{1}	2.68×10^{-2}
LaMa [17]	9.69×10^{-1}	3.25×10^{1}	1.97×10^{-2}	9.68×10^{-1}	3.38×10^{1}	1.83×10^{-2}
MISF [18]	9.07×10^{-1}	2.81×10^{1}	3.44×10^{-2}	8.83×10^{-1}	2.80×10^{1}	5.52×10^{-2}
RFR [21]	9.67×10^{-1}	3.19×10^{1}	1.58×10^{-2}	$\mathbf{9.69\times10^{-1}}$	3.38×10^{1}	1.40×10^{-2}
DR-IFMM	$\mathbf{9.70\times10^{-1}}$	$\mathbf{3.33\times10^{1}}$	$\mathbf{1.21\times10^{-2}}$	$\mathbf{9.69\times10^{-1}}$	$\mathbf{3.41\times10^{1}}$	$\mathbf{1.10\times10^{-2}}$

The SSIM, PSNR, and LPIPS evaluation metrics are used to verify the inpainting performance of eight image inpainting approaches on randomly damaged images. The results are shown in Table 2. The DR-IFMM algorithm outperforms other comparison approaches. Although it is very close to the LaMa and RFR approaches, the proposed algorithm still demonstrates the best inpainting performance.

In summary, the proposed approach performs best in inpainting regular scratches. When facing complex damage, the inpainting difficulty increases significantly. In this case, the DR-IFMM shows similar performance in detail inpainting compared to deep learning approaches, such as LaMa. However, the DR-IFMM does not rely on large-scale datasets or model training. It effectively addresses the challenge of inpainting images in small-sample datasets. Moreover, it demonstrates strong inpainting capability for damaged images even in the absence of prior knowledge.

4. CONCLUSION

This paper proposes an image inpainting approach based on DR-IFMM to address the challenge of insufficient small sample data in the inpainting of Dunhuang murals. This approach effectively tackles the issue of data scarcity and achieves high-quality inpainting of murals with diverse types of damage. In the initial stage, two optimal inpainting radii are determined based on the pixel density of the damaged region. These radii are then sequentially input into the improved weighting calculation of the IFMM algorithm. The two generated images are subsequently recombined to form the final inpainted image. Compared to approaches such as MSMM, SFIM, COTR, and ICriminisi, the DR-IFMM performs the best in terms of SSIM, PSNR, and LPIPS evaluation metrics. While deep learning approaches represented by LaMa, MISF, and RFR show competitive results on some datasets, the DR-IFMM offers advantages in terms of data volume, portability, and computational resource requirements. It proves more powerful in handling the inpainting of damaged images without prior knowledge. Future work will focus on further optimizing the DR-IFMM to enhance its performance on images with larger damaged areas and higher levels of destruction.

ACKNOWLEDGMENTS

This paper was supported by Funded Project under the Open Fund of Key Research Bases (XTYC202401). We gratefully acknowledge the Dunhuang Mural Research Institute for providing data support.

REFERENCES

[1] Cheng Y, Huang J, Zhang Y, Peng N. Artificial intelligence in cultural heritage conservation. Chinese Journal of Nature. 2024; 46(04):261-270.

[2] Liu B, He F, Du S, Zhang K, Wang J. Dunhuang murals contour generation network based on convolution and self-attention fusion. Applied Intelligence.2023; 53:22073–22085.

[3] Tian B, Zhou Y, Shan H, Yang L, Dang J. HASM: restoration of dunhuang murals with spatial structure prediction and line drawing assisted guidance. Journal of University of Electronic Science and Technology of China. 2025; 54(04):582-591.

[4] Guan J, Li H, Cai X, Chen E, Lin J, Ding Z, Ni Y. Progressive generative mural image restoration based on adversarial structure learning. npj Heritage Science. 2025; 13:1-11.

[5] Wu K, Shan H, Wei Y, Tian B, Wang L. Color enhancement of dunhuang murals under the fusion of multi-task decomposition and adaptive color equalization. Journal of University of Electronic Science and Technology of China. 2025; 54(03):384-392.

[6] Liu Z, Liu Y, Hu W. Mural restoration via joint gated attention and residual dense transformer. Infrared and Laser Engineering. 2025; 54(8):1-14.

[7] Luo Y, Bi X. MFNet: Multiscale fusion network for Dongba character image inpainting. Expert Systems with Applications.2025; 294:1-12.

[8] Shen J, Chan T. Mathematical models for local non-texture inpainting. Society for Industrial and Applied Mathematics. 2002; 62(03):1019-1043.

[9] Li Q, Li L, Wang W, Nan P. Image inpainting of damaged textiles based on improved Criminisi algorithm. Laser & Optoelectronics Progress. 2023; 60(16):173-182.

[10] Zhang J, Xin B, Yuan Z, Xu Y. Digitized restoration of textile pattern through edge-guided image inpainting method. Journal of Textile Research. 2024; 45(02):101-111.

[11] Gaona I, Mello-Román J, Noguera J, Legal-Ayala H, Méndez J, Grillo S. Enhanced medical images through multi-scale mathematical morphology by reconstruction. 2023 18th Iberian Conference on Information Systems and Technologies. 2023; 1-5.

[12] Min X, Huang J. Simplified fast image inpainting method. Journal of Computer Applications. 2017; 37(S1):169-172.

[13] Bornemann F, Mrz T. Fast image inpainting based on coherence transport. Journal of Mathematical Imaging and Vision. 2007; 28(03):259-278.

[14] Chen W, Tian Q, Lian L, Zhang X, Wang H. Research progress of image inpainting methods based on deep learning. Computer Engineering and Applications. 2024; 60(22):58-73.

[15] Zhang X, Zhai D, Li T, Zhou Y, Lin Y. Image inpainting based on deep learning: A review. Information Fusion. 2023; 90:74-94.

[16] Meng S, Chhea K, Muy S, Lee J. Deep reinforcement learning and metaheuristic approaches to maximize downlink sum-rate for Internet of Things systems in non-orthogonal multiple access-based space-air-ground integrated networks. Romanian Journal of Information Science and Technology. 2025; 28(04):327-340.

[17] Suvorov R, Logacheva E, Mashikhin A, Remizova A, Ashukha A, Silvestrov A, Kong N, Goka H, Park K, Lempitsky V. Resolution-robust large mask inpainting with fourier convolutions. IEEE/CVF Winter Conference On Applications Of Computer Vision. Waikoloa. 2022; 2149 − 2159.

[18] Li X, Guo Q, Lin D, Li P, Feng W, Wang S. MISF: multi-level interactive siamese filtering for high-fidelity image inpainting. 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022; 1859-1868.

[19] Xu Z, Yang X. Mural image restoration method based on CSWin-Transformer and gate convolution. Computer Engineering and Applications. 2024; 60(21):215-224.

[20] Zhang C, Yang W, Li X, Han H. MMGInpainting: multimodality guided image inpainting based on diffusion models. IEEE Transactions on Multimedia. 2024; 26:8811-8823.

[21] Li J, Wang N, Zhang L, Du B, Tao D. Recurrent feature reasoning for image inpainting. 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020; 7757-7765.

[22] Zhang S, Yang F. Digital mural inpainting model based on improved two-stage generative adversarial network. Electronic Measurement Technology. 2023; 46(11):123-129.

[23] Yang J. Image inpainting using complex 2-D dual-tree wavelet transform. Applied Mathematics-A Journal of Chinese Universities Series B. 2011; 26(01):70-76.

[24] Huang Y, Lu J, Chen N, Ding H, Shang Y. A deep learning image inpainting method based on stationary wavelet transform. Multimedia Systems. 2023; 29(06):3193-3207.

[25] Cao Y, Ma R, Zhao K, An P. WFIL-NET: image inpainting based on wavelet downsampling and frequency integrated learning module. Multimedia Systems. 2024; 31(01):1-13.

[26] Yan Y, Zhang X, Cai X, Lei T, Lian L, Zhang H. Restoration and reuse of clothing patterns and colors based on FMM and network analysis. Journal of Silk. 2024; 61(09):28-38.

[27] Zhang R, Isola P, Efros A, Shechtman E, Wang O. The unreasonable effectiveness of deep features as a perceptual metric. 31st IEEE/CVF Conf on Computer Vision and Pattern Recognition. 2018; 586-595.

Received September 8, 2025