A Deep Learning Image Inpainting Method based on Stationary Wavelet Transform

Yuhan Huang
Capital Normal University
Nianzhe Chen
Capital Normal University
Jiacheng Lu
Capital Normal University
Hui Ding (dhui@cnu.edu.cn)
Capital Normal University

Research Article

Keywords: Image inpainting, Stationary wavelet transform, Contextual attention, Loss function

Posted Date: December 30th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-2411942/v1

Additional Declarations: No competing interests reported.

A Deep Learning Image Inpainting Method based on Stationary Wavelet Transform

Yuhan Huang[1], Nianzhe Chen[1], Jiacheng Lu[1] and Hui Ding[1,2*]

[1]School of Information Engineering, Capital Normal University, 105 North Road of West Ring 3, Beijing, 100048, China.
[2]Beijing Advanced Innovation Center for Imaging Technology, Beijing, 100048, China.

*Corresponding author(s). E-mail(s): dhui@cnu.edu.cn;
Contributing authors: yhan_h@163.com; chennz915@163.com; jchengl@foxmail.com;

Abstract

The development of deep learning has greatly improved the image inpainting performance in the past decades. To inpaint images with specific tasks usually require different network models. For instance, the highly structured images need to recover the structural consistency, and the textured ones restore the local high-frequency details. However, it is still challenging to realize an effective algorithm taking account of the global structure and texture details separately. Herein, we proposed a two-stage inpainting method, by combining the information of frequency and spatial domain networks. Stationary wavelet transform (SWT) with good time-frequency characteristics was applied to obtain the sub-band images as the basic input for frequency domain inpainting. Contextual attention layer (CAL) modules were optionally introduced in the network model to adapt to various inpainting tasks. We also tested and discussed the impacts of some commonly used loss functions, including normal L1 loss, normal GAN loss, weighted L1 loss and WGAN-GP loss, on highly structured and textured images.

Keywords: Image inpainting, Stationary wavelet transform, Contextual attention, Loss function

1 Introduction

Image inpainting is a conservation technique to reconstruct the missing or damaged parts of images. The recent development of deep learning has greatly improved the performance of image inpainting, making it widely used in many fields such as life, culture, medicine, satellite, security and so on. Most of the currently explored inpainting algorithms focus on the extraction of spatial features and thus work better for highly structured images [1–3], such as natural, facial, and architectural images. However, the inpainting effect on the textured images is still not ideal to date.

Some algorithms adopted the combination of frequency and spatial domains to inpaint textured images [4–8]. The frequency domain images, as obtained by discrete fourier transform (DFT) or discrete wavelet transform (DWT), can be used as information for network training. The DFT-extracted frequency components can represent the information in the global context but in the absence of the temporal locality of the signals. In contrast, DWT is a multiresolution time-frequency-dependent transform and extracts the high-frequency components in three directions of horizontal, vertical, and diagonal. It can be used to learn more high-frequency details with the advantage of the modeling frequency domain. As an alternative implementation of DWT, stationary wavelet transform (SWT) can model the sub-band images with the same size as the original ones, and the sufficient coefficients can make the transform translation-invariant and data-redundant, and also avoid the downsampling-caused Gibbs phenomenon.

In order to take account of both the global structure and texture details simultaneously, we propose a two-stage image inpainting algorithm based on SWT. Firstly, the high and low subband images obtained by SWT are taken as the network input in the first stage to restore texture details. Secondly, the inpainted features are transformed back to the spatial domain by inverse stationary wavelet transform (ISWT) and used as the prior information for the next stage. In addition, the Contextual Attention Layer (CAL) [9] are introduced into the second stage as an optional module. It can be set up to enabled or disabled in the network model to adapt to different types of inpainting tasks, including the highly structured and textured images. Our contributions are summarized as follows:

- In order to inpaint the high-frequency details of images, we apply the SWT with good time-frequency characteristics to model the frequency domain information, which is used as the input of frequency domain inpainting stage.
- In order to meet the different modeling requirements for different highly structured images and textured images, we construct a spatial image inpainting model with an optional CAL module to adapt to different types of images.
- We discuss the impact of the commonly used loss functions, including normal L1 loss, normal GAN loss, weighted L1 loss and WGAN-GP loss, on the performance of our model for two different types of images.

2 Related Work

Traditional image inpainting algorithms [10–14] inpaint defects by diffusing edge pixels into missing areas or using shallow texture and structure of known patches to fill defect areas. These methods are less effective when the missing area is large or the texture is complex. In contrast, image inpainting algorithms based on deep learning can directly extract deep semantic features from underlying images through networks, and has better generalization ability and better inpainting effect. Isola et al. [1] proposed the Pix2pix algorithm, replacing the encoder-decoder network with U-Net. Its skip connections supplemented the underlying semantic information to high layers. PatchGAN was also proposed to judge image local parts to better restore local high-frequency details. Liu et al. [2] proposed Partial Convolutions (PConv). It only convolves the pixels in the known area as effective pixels, and realizes the different treatment of pixels inside and outside the damaged area, thereby effectively solving the problem of visual artifacts. Liu et al. [15] proposed a probabilistic diverse GAN (PD-GAN), it adopted both soft and hard spatially probabilistic diversity normalization (SPDNorm) to control the probability of producing diverse results, their method can produce diverse prediction and generate high-quality reconstruction content. Guo et al. [16] proposed a two-stream network, which models the structure-constrained texture synthesis and texture-guided structure reconstruction. They designed a Bi-GFF module to combine the structure and texture information and developed a CFA module to refine inpainting. Peng et al. [17] proposed a multiple-solution method which used hierarchical VQ-VAE to generate diverse and high-quality images. They designed a structural attention module to combine texture and structure and two feature losses to improve structure coherence and texture realism. Their method shows the superiority in both quality and diversity. Wang et al. [18] proposed a DSNet which contained VMC and RCN modules. The VMC module dynamically selects sampling locations based on the information in feature maps for flexible learning and the RCN module can adaptively learn weights. In addition, there are algorithms that combines frequency domain and spatial domain [4–8], which propose to use DFT or DWT to model frequency domain information, they can restore clearer texture details. Therefore, in this paper, we also use the combination of frequency domain and spatial domain to realize the network.

In deep learning, there are many researches on image inpainting algorithms utilizing attention mechanism [9, 19–26], and these methods are inspired by the idea of patch matching in traditional methods. Yu et al. [9] proposed CAL, whose role is to obtain local features from farther known regions and use them to predict more likely and effective content for missing regions. CAL can be flexibly added to the network to improve algorithm performance [3] , so we want to discuss its effectiveness in our algorithm for inpainting different types of images.

In previous image inpainting algorithms, the combination and improvement of some loss functions [3, 9, 27–29] are often proposed to enhance the inpainting

performance. Quan et al. [3] used the weighted L1 loss, which is a weighted sum of the inpainted region loss and the known region loss. By setting the loss weight of the inpainted area to be larger, the training process pays more attention to the compensation effect of the inpainted area. Yu et al. [9] used the WGAN-GP loss [28] as the loss for the adversarial network. It added a gradient penalty term on the basis of WGAN [29] to further stabilize the training, so we discuss the impact of the above two loss functions on the performance of our model.

3 Proposed Method

Since the image inpainting algorithm [4] that uses both frequency domain and spatial domain information has a good performance on the public image inpainting dataset. Therefore, we construct the network model by copying and improving this algorithm. The framework of our network model is shown in Figure 1. It is divided into two stages, including the frequency domain inpainting network (NetF) and the spatial domain inpainting network (NetS). NetF takes the subband images constructed by SWT as network input. The network uses simple CNNs. The inpainted subband images are transformed back to the spatial domain by ISWT. The structure of NetS is GAN. It takes the first stage output as prior conditions, which provides more complete basic information. In addition, an optional CAL module is applied to adapt to different types of images inpainting tasks.

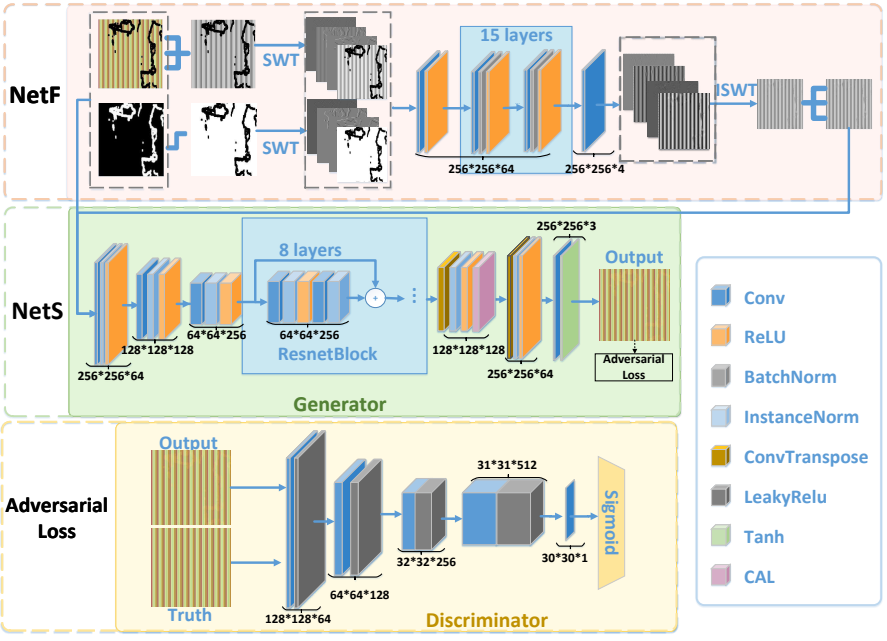

Fig. 1 The framework of our baseline network model.

3.1 NetF based on SWT

3.1.1 The Details of SWT

In the process of signal analysis, the signal is not stationary, but changes over time. The wavelet transform has frequency and time windows that can change shape. It can adapt to the change of signal waveform, which consider the sharp and stable time of waveform change at the same time and has the ability to analyze the signal locally. Compared with other frequency domain modeling methods, it can reflect the local characteristics of time and frequency more flexibly.

The wavelet transform will downsample when generating the subband image, causing its size to become a quarter of the original image. In the inverse wavelet transform, interpolation is needed to restore the original size. This transformation method cannot preserve translation invariance. But SWT can solve this problem. In the process of decomposition, SWT inserts 0 in each layer filter, which ensures that the image size unchanged. This will provide enough coefficients to make the transformation have translation invariance and data redundancy, avoiding the Gibbs phenomenon caused by downsampling.

The advantages of SWT inspired this study to utilize it as the input to the frequency-domain inpainting network. Haar wavelet is adopted to implement 2-D Stationary Wavelet Transform (SWT) for its simplicity. The low-pass and high-pass filters used in SWT upsample as a factor of 2. If S(x,y) represents spatial domain image, it is iteratively decomposed into four subband images including LL,LH,HL, and HH. The process of one level's SWT, defined as:

$$LL, LH, HL, HH = (f^p \uparrow 2 \otimes S(x, y)) \tag{1}$$

where f denotes filters, $\uparrow 2$ represents the standard up-sampling operator with factor 2. Due to the invertibility of SWT, the reconstruction of one level's subband images can be correspondingly implemented by completely inverse operation via ISWT without any information loss.

3.1.2 NetF Achievement

In the Frequency Domain Inpainting Network (NetF shown in Figure 1), the inpainting process of frequency domain images is defined as:

$$I^1_{pred} = ISWT\left(f\left(SWT\left(I_{input}, M\right);\ \theta\right)\right) \tag{2}$$

where I^1_{pred} represents inpainted images at this stage, I_{input} denotes input damaged images, M is the binary representation of mask images, f means the function equivalent to the frequency domain inpainting network.

We adopt DnCNNs [4] network to realize feature learning of frequency domain images. The details of the network structure are shown in Table 1. In the table, C represents the number of channels, and W and H represent the width and height of the input image, respectively. This network combines

ReLU and batch normalization (BN), which improves the model's ability to learn nonlinear mapping relationships and makes model training more stable.

Table 1 Detailed Description of DnCNNs Network Structure

Layer	Stride,Padding	Activation	Output size
Input	-	-	1*C*H*W
3*3 Conv	1,1	ReLU	1*64*H*W
3*3 Conv×15	1,1	BN+ReLU	1*64*H*W
3*3 Conv	1,1	-	1*C/2*H*W

The inputs to the network are the masked image, the mask and the subband images constructed by the SWT. Since the frequency domain transformation of the RGB images multiplies the network input channels, and we found in our experiments that this stage network can not correctly use and restore color information. To reduce the time cost of model training at this stage, we convert the RGB image inpainting problem into a grayscale image inpainting problem, the conversion is defined as:

$$\mathrm{GRAY} = 0.299\mathrm{R} + 0.587\mathrm{G} + 0.114\mathrm{B} \tag{3}$$

3.2 NetS based on Optional CAL

3.2.1 The Details of CAL

In deep learning, the inpainting network of spatial image usually uses dilated convolution, residual blocks or encoder-decoder structure to expand the receptive field. But these methods are still difficult to extract image local features from distant regions. Since the capture and learning of distant local features can help missing regions recover more appropriate content [9], we add an optional CAL [9] to the network at this stage to help borrow feature information when appropriate.

The structure of the CAL module is shown in Figure 2. Feature maps are separated into foreground (missing) and background (known), and they are split into many patches. Cosine similarity is used to measure the similarity between background and foreground patches, and all background patch attention scores are obtained by softmax. Then, the transposed convolutions with the background patches as the convolution kernels on the attention scores get the prediction of the missing patch. For the feature matching, we select the 3×3 patches as the filter [9], and the cosine similarity is defined as:

$$S_{i,j} = \left\langle \frac{F_i}{\|F_i\|}, \frac{\overline{F_j}}{\|\overline{F_j}\|} \right\rangle \tag{4}$$

where F denotes the known area, and $\overline{F}$ represents the missing area. i means the i-th patch, and j means the j-th patch. $S_{i,j}$ denotes the similarity

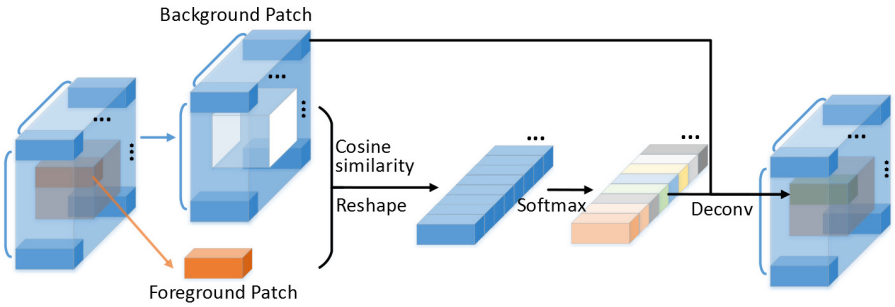

Fig. 2 The structure of the CAL module.

of F_i and $\overline{F_j}$. $S_{i,j}$ can get the attention score by softmax calculation:

$$\lambda_{i,j} = \frac{exp\,(S_{i,j})}{\sum_{j=1}^{M} exp\,(S_{i,j})} \tag{5}$$

where M represents the number of patches in the missing area. $\lambda_{i,j}$ denotes the attention score.

3.2.2 NetS Achievement

The network at this stage adopts GAN [4], which consists of a generator and a discriminator, and the structure is shown as NetS in Figure 1. The network converts the grayscale image inpainting problem in the first stage into an RGB image inpainting problem.

The generator structure details are shown in Table 2, where IN represents instance normalization. It uses a downsampling-upsampling structure to reduce the computational complexity of feature learning and increase the receptive field. The middle residual network directly passes the low-level feature to the high-level. The existing identity mapping solves the problem of network degradation, and the convergence speed is faster. A CAL is added before the second transposed convolution as an optional module to match distant local features. The last layer of convolution uses Tanh activation, which limits the output range between -1 and 1, and outputs the compensated image.

The discriminator structure details are shown in Table 3, where LReLU means Leaky ReLU and SN denotes spectral normalization. It adopts a structure similar to PatchGAN, which maps the input into an N*N matrix through convolution. And each element in the matrix represents the evaluation value of a receptive field. It is used for judging the input image. Compared with the traditional discriminator that only gives one evaluation value, PatchGAN can comprehensively score multiple local information. The activation in the discriminator is Leaky ReLU, which preserves the linearity and ensures that the information less than 0 is not completely lost.

In addition, the residual block and the discriminator use spectral normalization, which solves the problem that the better the discriminator is trained,

Table 2 Detailed Description of Generator Network Structure

Module	Layer	Stride, Padding	Activation	Output size
-	Input	-	-	1*C*H*W
Encoder	7*7 Conv	1,3	IN+ReLU	1*64*H*W
	4*4 Conv	2,1	IN+ReLU	1*128*H/2*W/2
	4*4 Conv	2,1	IN+ReLU	1*256*H/4*W/4
Residual blocks×8	3*3 Conv	1,1	IN+ReLU	1*256*H/4*W/4
	3*3 Conv	1,1	IN	1*256*H/4*W/4
Decoder	4*4 Conv	2,1	IN+ReLU	1*128*H/2*W/2
	CAL	-	-	-
	4*4 Conv	2,1	IN+ReLU	1*64*H*W
	7*7 Conv	1,3	Tanh	1*C/3*H*W

Table 3 Detailed Description of Discriminator Network Structure

Layer	Stride, Padding	Activation	Output size
Input	-	-	1*C*H*W
4*4 Conv	2,1	SN+LReLU	1*64*H/2*W/2
4*4 Conv	2,1	SN+LReLU	1*128*H/4*W/4
4*4 Conv	2,1	SN+LReLU	1*256*H/8*W/8
4*4 Conv	1,1	SN+LReLU	1*512*(H/8-1)*(W/8-1)
4*4 Conv	1,1	SN	1*1*(H/8-2)*(W/8-2)
Output	-	Sigmoid	-

the more serious the generator gradient disappears. SN enables the discriminator to have Lipschitz continuity, which is a stronger smoothness condition that limits the severity of function changes and makes model training more stable.

3.3 Training Loss

3.3.1 Frequency Domain Inpainting Loss

In the frequency domain inpainting model training, we use the L2 loss function to minimize the distance between the ground-truth and the inpainted images, defined as:

$$L_F = \left\| I_{gt}^{f} - I_{pred}^{1f} \right\|_2^2 \tag{6}$$

where I_{gt}^{f} means the ground-truth subband image, I_{pred}^{1f} represents the inpainted subband image, L_F denotes the loss of NetF.

3.3.2 Spatial Domain Inpainting Loss

In the past image inpainting algorithms, the loss functions of generative adversarial networks are often combined and improved to achieve better results. In the training of our spatial inpainting network, we use a weighted sum of two losses, including the reconstruction loss and the adversarial loss. Choosing appropriate loss functions is important for training the network. Therefore, for the reconstruction loss, we compare the difference between the normal L1 loss and the improved weighted L1 loss [3], and for the adversarial loss we compare

the effect of the normal GAN loss and the newly proposed WGAN-GP loss [28].

The Reconstruction Loss. The normal L1 loss is defined as:

$$L_{recon} = \|I_{gt} - I_{pred}\|_1 \tag{7}$$

where I_{pred} denotes the inpainted image, I_{gt} means the ground-truth image. Normal L1 loss treats reconstruction regions and known regions equally, in contrast, weighted L1 loss takes pixel reconstruction regions as key regions for model training. That is, the loss of the reconstructed area is calculated separately from the known area, and the weight of the loss of the reconstructed area is increased. In this way, the network is more inclined to the correctness of the reconstructed region, which can theoretically improve the network repair performance. The reconstruction area loss L_{hole} and the known area loss L_{known} are defined as follows:

$$L_{hole} = \frac{1}{sum\,(M)} \|(I_{pred} - I_{gt}) \odot M\|_1 \tag{8}$$

$$L_{known} = \frac{1}{sum\,(1-M)} \|(I_{pred} - I_{gt}) \odot (1-M)\|_1 \tag{9}$$

where M is the mask (the missing region is 1). I_{pred} denotes the inpainted image, I_{gt} means the ground-truth image. When using weighted L1 loss as reconstruction loss in training, the reconstruction loss defined as:

$$L_{recon} = \lambda_{hole} L_{hole} + \lambda_{known} L_{known} \tag{10}$$

where λ_{hole} is set to 6, λ_{known} is set to 1.

The Adversarial Loss. The adversarial loss adopts a minimum optimization strategy. During training, The generator (G) tries to inpaint the masks image as close as possible to the ground-truth, and the discriminator (D) tries to distinguish between the ground-truth and the inpainted image as much as possible. Through the iterative game between G and D, the image inpainting effect is optimized. The objective function of normal GAN loss is expressed as follows:

$$\begin{aligned} L_{adv}\,(G, D) = \mathbb{E}_{I_{gt} \sim \mathbb{P}_{gt}} \left[log\,(D\,(I_{gt}))\right] + \\ \mathbb{E}_{I_{pred} \sim \mathbb{P}_{pred}} \left[log\left(1 - D\left(G\left(I^1_{pred}\right)\right)\right)\right] \end{aligned} \tag{11}$$

where I_{pred} denotes the inpainted image, I^1_{pred} represents the input to G, I_{gt} means the ground-truth. $\mathbb{P}_{gt}$ is the data distribution of I_{gt}, $\mathbb{P}_{pred}$ is the data distribution of I_{pred}.

Normal GAN loss is the simplest adversarial loss, it has the problem of unstable training. In contrast, the WGAN [29] can make training more stable without balancing the training levels of G and D. However, WGAN directly adopts weight clipping to satisfy Lipschitz constraint, which wastes the fitting ability of deep neural networks and easily causes gradients to vanish or explode. WGAN-GP loss is proposed for the problems existing in WGAN. It uses another truncation strategy, gradient penalty.

The objective function of WGAN-GP loss is expressed as follows:

$$L_{adv}(G,D) = \underbrace{\mathbb{E}_{G(z)\sim\mathbb{P}_g}[D(G(z))] - \mathbb{E}_{x\sim\mathbb{P}_r}[D(x)]}_{D\ loss} + \underbrace{\lambda\mathbb{E}_{\hat{x}\sim\mathbb{P}_{\hat{x}}}\left[\left(\|\bigtriangledown_{\hat{x}}D(\hat{x})\odot M\|_2 - 1\right)^2\right]}_{Gradient\ penalty} \tag{12}$$

$$\hat{x} = tG(z) + (1-t)x, t \sim U[0,1] \tag{13}$$

where z is the input to G, $\mathbb{P}_g$ is the distribution of $G(z)$. x is the input to D, $\mathbb{P}_r$ is the data distribution of x. $\hat{x}$ is sampled from the straight line between points sampled from distribution $\mathbb{P}_g$ and $\mathbb{P}_r$. M means the binary representation of the mask, it represents the gradient penalty by only applied to predict regions. The value of λ is set to 10. When applying the WGAN-GP loss, the sigmoid layer in D is removed.

According to the above, the total loss function of NetS is expressed as:

$$L_{NetS} = \lambda_{recon}L_{recon} + \lambda_{adv}L_{adv} \tag{14}$$

where λ_{recon} is set to 1, λ_{adv} is set to 0.1 in the experiments.

4 Experiments

Based on publicly available datasets of different types of images, we discuss the role of the optional module CAL and compare the effectiveness of loss functions commonly used in image inpainting.

4.1 Experimental setup

4.1.1 Datasets

To discuss the inpainting of highly structured images and textured images respectively, we select the highly structured CelebA-HQ dataset [30] and the textured DTD dataset [31] to validate our algorithm. They are frequently used in other image inpainting papers.

- CelebA-HQ dataset: It is a high-quality version of part of CelebA dataset. It is a collection of face images, consisting of 30,000 images of 1024*1024. We randomly divided the dataset into 28,000 training set, 1,000 validation set and 1,000 test set.
- DTD dataset: It includes 5640 real-world texture images, obtained from Google and Flickr websites. The resolution of the images varies from 300*300 to 640*640. We randomly divided the dataset into 5170 training, 235 validation, and 235 testing sets.

For the mask, we used the irregular masks shared by Liu et al. [2]. They divided the masks into six types according to their proportions. In our experiments, we use four of them, 10-20%, 20-30%, 30-40%, and 40-50%.

4.1.2 Implementation Details

Our proposed model is implemented with PyTorch 1.7.1. We unified the dataset image size to 256×256 during training and validation and used the original dataset images during testing. The mask size changes with the dataset image size. In the first stage, we initialize the weights by using He initialization. And we use Adam optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ at the first stage, $\beta_1 = 0.5$ and $\beta_2 = 0.999$ at the second stage. To train the network, we decayed the learning rate 0.001 at the first stage, and the learning rate of G and D at the second stage are set to 0.0001 and 0.00001 respectively. Our full model runs on hardware with GPU GV102.

4.1.3 Evaluation Metrics

For the quantitative evaluation, we adopt several common metrics in the image inpainting task: MAE, MSE, PSNR, SSIM and FID. The first four metrics are based on the low-level pixel values, the last one metric is related to the high-level visual perception. If T is the ground truth image and F is the inpainted image, then MAE represents the mean of the absolute errors between T and F, and MSE denotes the mean of the sum of squared errors between T and F. PSNR is the ratio of peak signal energy to average energy of image distortion, which can be represented by MSE.

SSIM [32] measures the structural similarity of T and F from three perspectives: lightness (L), contrast (C), and structure (S). FID [33] reflects the similarity from the aspect of visual feature statistics by measuring the distance between the feature vectors of T and F. Their formulas are defined as follows:

$$SSIM\,(T,F) = L\,(T,F) \times C\,(T,F) \times S\,(T,F) \tag{15}$$

$$FID\,(T,F) = Tr\left(\sum T + \sum F - 2\left(\sum T \sum F\right)^{\frac{1}{2}}\right) + \left\|\mu_T - \mu_F\right\|_2^2 \tag{16}$$

where Tr represents trace of matrix, $\sum$ represents the covariance. μ_T and μ_F is the mean pixel values of image T and F.

4.2 Discussion on the optional CAL and loss functions

We analyzed the application value of the optional module CAL and different loss functions on the textured DTD dataset and the highly structured CelebA-HQ dataset, as shown in Figure 3 and Figure 4. They are the training and validation curves of different conditions on two dataset, where CAL indicates that the CAL module is enabled, WL1 and L1 respectively denote weighted L1

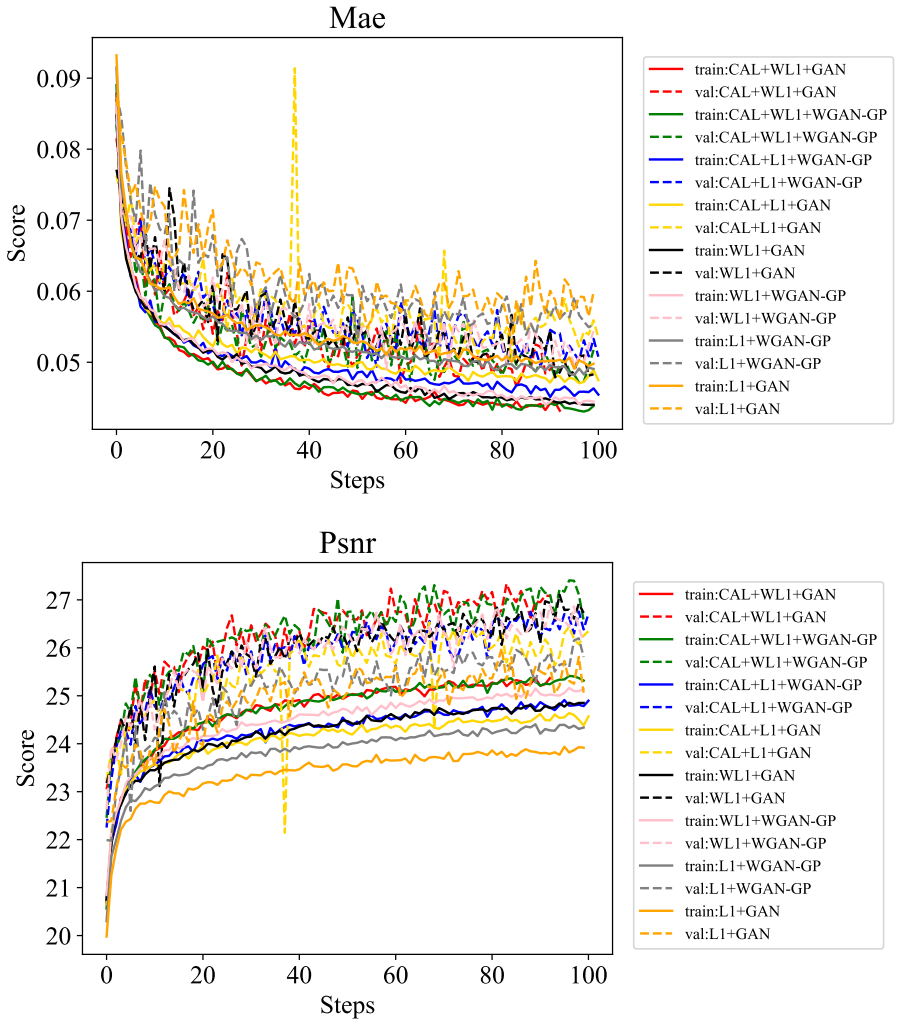

Fig. 3 The training (train) and validation (val) curves under different conditions on the DTD dataset. (The first one is the MAE curves, and the second one is the PSNR curves.)

loss and normal L1 loss, WGAN-GP and GAN respectively represent WGAN-GP loss and normal GAN loss. From Figure 3, it can be observed that enabling the CAL module on the DTD dataset can speed up model convergence and improve model inpainting performance. However, from Figure 4, the model that skips the CAL performs better for the CelebA-HQ dataset. According to the above, the CAL is more suitable for textured images. It means that the long-distance features of texture images have more utilization value.

Figure 3 and Figure 4 also shows the effect of joint constraints of different losses on the training process. For both datasets using the CAL module correctly, the model using WL1 is optimal, and the use of WGAN-GP has little

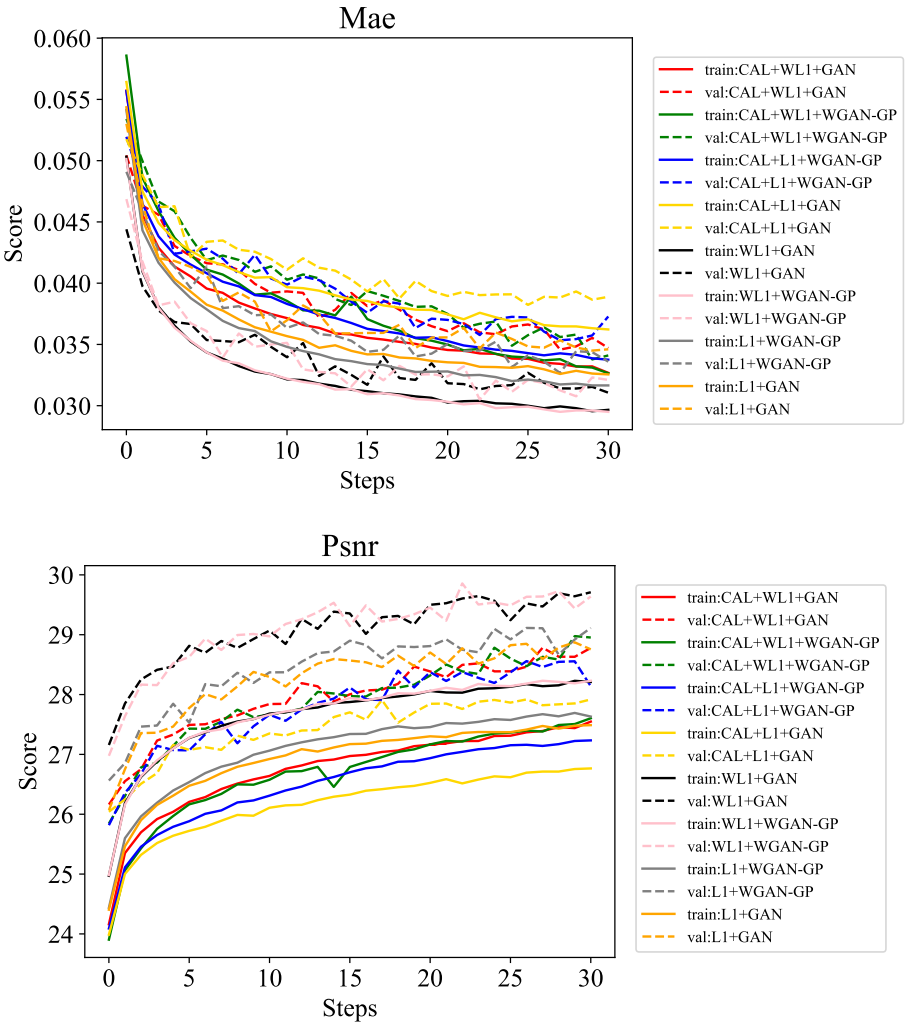

Fig. 4 The training (train) and validation (val) curves under different conditions on the CelebA-HQ dataset. (The first one is the MAE curves, and the second one is the PSNR curves.)

effect on the optimal model curve. WGAN-GP is effective only when the CAL module is incorrect or weighted L1 loss is not used, probably because this gives WGAN-GP room to show its effects. That means WGAN-GP as adversarial loss can improve the minimum performance level of the model.

Table 4 and Table 5 show the evaluation metrics for different training situations (in Figure 3 and Figure 4) on two dataset with different proportion masks (10-50%). ↑ means higher is better, ↓ means lower is better. The values bolded in red represent the best, and the values bolded in black represent the second best. From the tables, we observe the same rules as the curves in Figure

Table 4 The comparisons of different training situations on the textured DTD dataset in Figure 3.

Masks	Models	MAE↓ (%)	MSE↓ (%)	SSIM↑	PSNR↑	FID↓
10-20 %	L1+GAN	1.2864	0.2034	0.9237	28.9934	43.8651
	L1+WGAN-GP	1.2925	0.2032	0.9254	28.8028	67.7349
	WL1+GAN	1.1605	0.1573	0.9327	30.0189	51.4741
	WL1+WGAN-GP	1.1930	0.1708	0.9312	29.7181	54.5805
	CAL+L1+GAN	1.2548	0.1823	0.9286	29.3348	**33.0890**
	CAL+L1+WGAN-GP	1.1931	0.1738	0.9318	29.7022	44.2949
	CAL+WL1+GAN	**1.1154**	**0.1489**	**0.9366**	**30.4303**	**39.1003**
	CAL+WL1+WGAN-GP	**1.1307**	**0.1494**	**0.9368**	**30.3113**	44.5382
20-30 %	L1+GAN	2.1859	0.3915	0.8678	26.1551	72.2455
	L1+WGAN-GP	2.1039	0.3528	0.8729	26.2897	98.1306
	WL1+GAN	1.9720	0.3029	0.8826	27.1626	84.4447
	WL1+WGAN-GP	1.9692	0.3068	0.8807	27.0478	88.3887
	CAL+L1+GAN	2.0975	0.3427	0.8777	26.5804	**55.4511**
	CAL+L1+WGAN-GP	2.0372	0.3383	0.8818	26.7667	76.6929
	CAL+WL1+GAN	**1.8871**	**0.2835**	**0.8893**	**27.5550**	**67.2282**
	CAL+WL1+WGAN-GP	**1.8751**	**0.2719**	**0.8908**	**27.6457**	74.3146
30-40 %	L1+GAN	3.0508	0.5780	0.8067	24.3200	103.3478
	L1+WGAN-GP	3.0448	0.5686	0.8138	24.2690	133.7379
	WL1+GAN	2.7604	0.4601	0.8274	25.2770	115.9347
	WL1+WGAN-GP	2.8460	0.4893	0.8242	24.9473	122.6283
	CAL+L1+GAN	2.9326	0.5161	0.8213	24.7256	**83.4982**
	CAL+L1+WGAN-GP	2.8645	0.5100	0.8267	24.8400	112.6386
	CAL+WL1+GAN	**2.6601**	**0.4411**	**0.8366**	**25.6854**	**98.4709**
	CAL+WL1+WGAN-GP	**2.7113**	**0.4407**	**0.8383**	**25.4415**	104.1821
40-50 %	L1+GAN	4.0007	0.7891	0.7451	22.8922	131.0941
	L1+WGAN-GP	4.0022	0.8074	0.7560	22.8074	157.8199
	WL1+GAN	3.6476	0.6486	0.7722	23.6807	142.2222
	WL1+WGAN-GP	3.7574	0.6988	0.7686	23.4030	150.7337
	CAL+L1+GAN	3.9055	0.7402	0.7611	23.0193	**108.2128**
	CAL+L1+WGAN-GP	3.8315	0.7299	0.7681	23.1786	141.7997
	CAL+WL1+GAN	**3.5189**	**0.6183**	**0.7831**	**24.0142**	**123.9327**
	CAL+WL1+WGAN-GP	**3.5626**	**0.6288**	**0.7862**	**23.9417**	129.1315

3 and Figure 4, but we take the optimal models in the metric tables to participate in the subsequent algorithm comparisons. That is, the CAL+WL1+GAN model of the DTD dataset is the best, and the WL1+WGAN-GP model of the CelebA-HQ dataset is the best. The above optimal methods are determined according to the number of optimal metrics, and when the number of optimal metrics is the same, the situation of suboptimal metrics is considered.

4.3 Performance Evaluation

We compare the metrics of our baseline (SWT), our optimal method, and other three state-of-the-art image inpainting methods on DTD and CelebA-HQ dataset. Three comparison algorithms include Pix2pix [1], PConv [2] and DFT [4] methods. They are shown in Table 6 and Table 7, respectively. The optimal methods have optimal values on the four metrics of MAE, MSE, PSNR, SSIM, which indicates that the images they inpaint are most similar to ground-truth at low pixel values. However, their poor performance on the FID metric

Table 5 The comparisons of different training situations on the highly structured CelebA-HQ dataset in Figure 4.

Masks	Models	MAE↓ (%)	MSE↓ (%)	SSIM↑	PSNR↑	FID↓
10-20 %	L1+GAN	0.6649	0.0633	0.9647	33.2510	**4.4347**
	L1+WGAN-GP	0.6585	0.0614	0.9662	33.3510	5.4739
	WL1+GAN	**0.6253**	**0.0559**	**0.9681**	**33.8290**	**4.9616**
	WL1+WGAN-GP	**0.6166**	**0.0554**	**0.9685**	**33.9261**	5.2724
	CAL+L1+GAN	0.7970	0.0841	0.9560	31.8122	5.9134
	CAL+L1+WGAN-GP	0.7116	0.0711	0.9625	32.6180	5.9179
	CAL+WL1+GAN	0.6971	0.0665	0.9626	32.8518	5.4596
	CAL+WL1+WGAN-GP	0.6989	0.0653	0.9632	32.9137	5.8113
20-30 %	L1+GAN	1.1699	0.1335	0.9356	29.7419	**7.7946**
	L1+WGAN-GP	1.1504	0.1279	0.9393	29.9133	10.0889
	WL1+GAN	**1.1013**	**0.1180**	**0.9421**	**30.3246**	**9.3039**
	WL1+WGAN-GP	**1.0900**	**0.1175**	**0.9428**	**30.3553**	9.9635
	CAL+L1+GAN	1.3884	0.1696	0.9216	28.4819	10.4847
	CAL+L1+WGAN-GP	1.2388	0.1454	0.9337	29.2735	11.1279
	CAL+WL1+GAN	1.2176	0.1388	0.9333	29.4753	9.8763
	CAL+WL1+WGAN-GP	1.2171	0.1360	0.9347	29.5453	10.5719
30-40 %	L1+GAN	1.7643	0.2327	0.9008	27.1604	**11.2610**
	L1+WGAN-GP	1.7326	0.2240	0.9079	27.3340	14.8881
	WL1+GAN	**1.6600**	**0.2049**	**0.9115**	**27.7390**	**13.8859**
	WL1+WGAN-GP	**1.6484**	**0.2068**	**0.9126**	**27.7406**	15.0056
	CAL+L1+GAN	2.0697	0.2913	0.8823	26.0393	15.6793
	CAL+L1+WGAN-GP	1.8542	0.2496	0.9007	26.7799	16.6646
	CAL+WL1+GAN	1.8172	0.2378	0.9004	26.9961	14.3135
	CAL+WL1+WGAN-GP	1.8034	0.2293	0.9029	27.1316	15.6672
40-50 %	L1+GAN	2.5153	0.3815	0.8598	24.9384	**15.5680**
	L1+WGAN-GP	2.4611	0.3656	0.8724	25.1331	21.2383
	WL1+GAN	**2.3601**	**0.3355**	**0.8765**	**25.5238**	**19.6511**
	WL1+WGAN-GP	**2.3488**	**0.3414**	**0.8784**	**25.4944**	20.9365
	CAL+L1+GAN	2.9074	0.4698	0.8379	23.9336	23.9017
	CAL+L1+WGAN-GP	2.6190	0.4032	0.8635	24.6479	24.0297
	CAL+WL1+GAN	2.5617	0.3873	0.8630	24.8496	20.2044
	CAL+WL1+WGAN-GP	2.5315	0.3687	0.8668	25.0390	21.8003

indicates the lack of image diversity. Nevertheless, for DTD dataset, the FID metric of the optimal method is still superior to the three comparison algorithms. Furthermore, from the definition of WL1, the application of WL1 will make the network pay more attention to low-level pixels, so this metric result is acceptable. Figure 5 and Figure 6 show the comparison of inpainted results between our method and three other state-of-the-art image inpainting methods on DTD and CelebA-HQ dataset. From Figure 5, we can find that our method can build the clearest and reasonable texture. From Figure 6, it can be seen that our method can restore the most complete structure and color. In short, although the optimal methods has shortcomings in the FID metric, the visual effects of inpainted image are still optimal.

Table 6 Algorithms comparison on DTD dataset.

Metrics	Masks	Pix2pix	PConv	DFT	SWT	CAL+WL1+GAN
MAE↓(%)	10-20%	1.9484	1.5431	1.3619	**1.1784**	**1.1154**
	20-30%	3.2734	2.6433	2.2652	**2.0563**	**1.8871**
	30-40%	4.6348	3.7928	3.2512	**2.8961**	**2.6601**
	40-50%	6.0480	5.0926	4.2556	**3.8807**	**3.5189**
MSE↓(%)	10-20%	0.4467	0.2892	0.2317	**0.1758**	**0.1489**
	20-30%	0.8100	0.5381	0.4043	**0.3487**	**0.2835**
	30-40%	1.2364	0.8412	0.6493	**0.5372**	**0.4411**
	40-50%	1.6761	1.1931	0.8869	**0.7419**	**0.6183**
SSIM↑	10-20%	0.8887	0.9061	0.9298	**0.9315**	**0.9366**
	20-30%	0.8099	0.8425	0.8732	**0.8751**	**0.8893**
	30-40%	0.7331	0.7691	0.8174	**0.8199**	**0.8366**
	40-50%	0.6552	0.6902	0.7544	**0.7588**	**0.7831**
PSNR↑	10-20%	25.6367	27.3283	28.9578	**29.5936**	**30.4303**
	20-30%	22.6726	24.6597	26.0266	**26.7036**	**27.5550**
	30-40%	20.7376	22.6839	23.9774	**24.6367**	**25.6854**
	40-50%	19.2174	20.9238	22.4964	**23.0351**	**24.0142**
FID↓	10-20%	63.3749	44.6316	42.3652	**30.5995**	**39.1003**
	20-30%	106.1846	75.5827	70.3736	**59.2970**	**67.2282**
	30-40%	130.9867	106.9886	101.5031	**89.2330**	**98.4709**
	40-50%	165.8571	137.3557	128.7030	**122.5976**	**123.9327**

Table 7 Algorithms comparison on CelebA-HQ dataset.

Metrics	Masks	Pix2pix	PConv	DFT	SWT	WL1+WGAN-GP
MAE↓(%)	10-20%	1.1843	0.7349	0.6884	**0.6649**	**0.6166**
	20-30%	2.1676	1.2757	1.1860	**1.1699**	**1.0900**
	30-40%	3.3828	1.8692	**1.7524**	1.7643	**1.6484**
	40-50%	4.9941	2.5983	**2.4874**	2.5153	**2.3488**
MSE↓(%)	10-20%	0.1751	0.0714	0.0658	**0.0633**	**0.0554**
	20-30%	0.3850	0.1462	0.1378	**0.1335**	**0.1175**
	30-40%	0.7069	0.2413	**0.2279**	0.2327	**0.2068**
	40-50%	1.2190	0.3845	**0.3754**	0.3815	**0.3414**
SSIM↑	10-20%	0.9301	0.9560	0.9636	**0.9647**	**0.9685**
	20-30%	0.8800	0.9230	0.9345	**0.9356**	**0.9428**
	30-40%	0.8271	0.8875	**0.9014**	0.9008	**0.9126**
	40-50%	0.7673	0.8465	**0.8615**	0.8598	**0.8784**
PSNR↑	10-20%	28.5599	32.4523	32.8730	**33.2510**	**33.9261**
	20-30%	24.9911	29.2466	29.6561	**29.7419**	**30.3553**
	30-40%	22.3196	26.9153	**27.2547**	27.1604	**27.7406**
	40-50%	19.8954	24.8673	**25.0496**	24.9384	**25.4944**
FID↓	10-20%	10.2569	5.9819	**4.9132**	**4.4347**	5.2724
	20-30%	16.1473	8.8809	**8.0420**	**7.7946**	9.9635
	30-40%	23.1082	12.7015	**11.5212**	**11.2610**	15.0056
	40-50%	31.8632	**16.1474**	16.3228	**15.5680**	20.9365

5 Conclusion

Since structural and textural features cannot be optimized simultaneously, different types of images usually require different inpainting algorithms. In view of this, we proposed a two-stage image inpainting algorithm based on SWT and optional module CAL, to explore its application potential to various

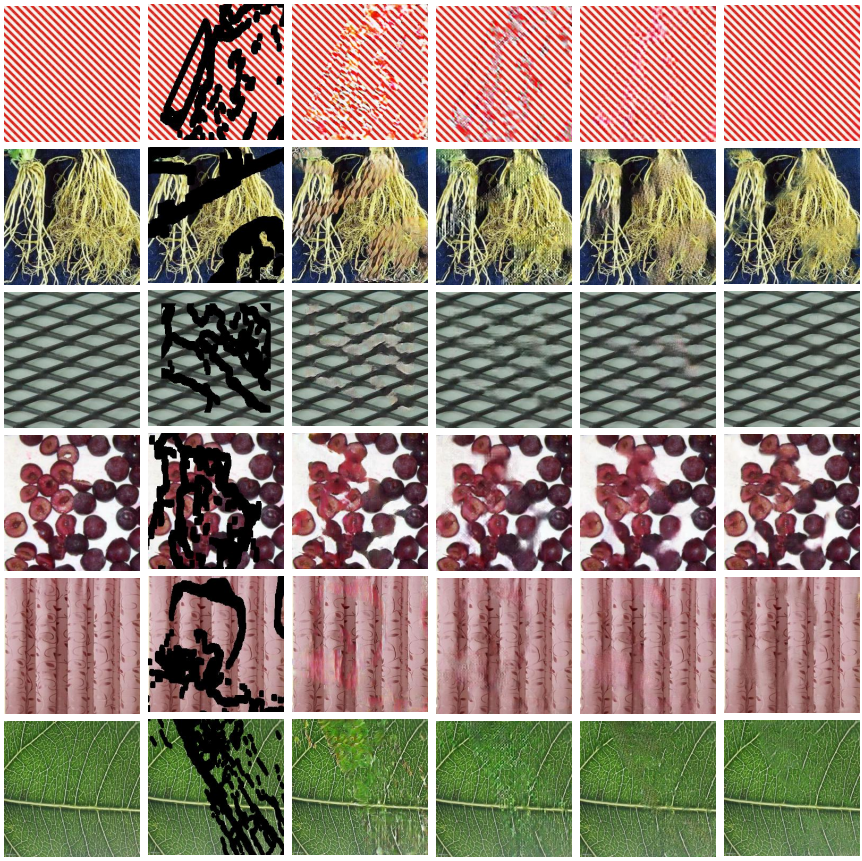

Fig. 5 Qualitative comparisons of our baseline method with Pix2pix, PConv and DFT on DTD dataset with irregular masks.

types of images. Through experiments, it is found that textured (highly structured) images tend to enable (disable) the CAL module, demonstrating that the extraction of long-distance features is helpful for the learning of textured features. Generative adversarial network training is generally constrained by a combination of reconstruction loss and adversarial loss. For each, we compare the normal and the commonly used state-of-the-art loss. Normal loss includes L1 loss and GAN loss, and advanced loss includes weighted L1 loss and WGAN-GP loss. The weighted L1 loss has a large performance improvement for the model, and the role of the WGAN-GP loss is less. In the future, we plan to apply these image inpainting algorithms to medical images or special types of images and explore the impact of frequency domain methods on algorithm performance.

Acknowledgments This work was supported by the National Natural Science Foundation of China (61876112).

Fig. 6 Qualitative comparisons of our baseline method with Pix2pix, PConv and DFT on CelebA-HQ dataset with irregular masks.

Declarations

- Funding: This work was supported by the National Natural Science Foundation of China (61876112).
- Competing interests: The authors declare that they have no competing interests.
- Ethics approval: Not applicable
- Consent to participate: Not applicable
- Consent for publication: Not applicable
- Availability of data and materials: All data generated or analysed during this study are included in this published article.
- Code availability: The code will be sorted out and exposed later.
- Authors' contributions: All authors contributed to the study conception and design. Material preparation and data collection were performed by all authors. Main manuscript writing, main experiments and result analysis

were performed by Yuhan Huang and Hui Ding. Partial experiments and tables were performed by Nianzhe Chen and Jiacheng Lu. All authors read and approved the final manuscript.

References

[1] Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)

[2] Liu, G., Reda, F.A., Shih, K.J., Wang, T.C., Tao, A., Catanzaro, B.: Image inpainting for irregular holes using partial convolutions. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 85–100 (2018)

[3] Quan, W., Zhang, R., Zhang, Y., Li, Z., Wang, J., Yan, D.M.: Image inpainting with local and global refinement. IEEE Transactions on Image Processing **31**, 2405–2420 (2022)

[4] Roy, H., Chaudhury, S., Yamasaki, T., Hashimoto, T.: Image inpainting using frequency-domain priors. Journal of Electronic Imaging **30**(2), 023016 (2021)

[5] Wang, C., Wang, J., Zhu, Q., Yin, B.: Generative image inpainting based on wavelet transform attention model. In: 2020 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1–5 (2020)

[6] He, X., Cui, X., Li, Q.: Image inpainting based on inside–outside attention and wavelet decomposition. IEEE Access **8**, 62343–62355 (2020)

[7] Yu, Y., Zhan, F., Lu, S., Pan, J., Ma, F., Xie, X., Miao, C.: Wavefill: A wavelet-based generation network for image inpainting. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14114–14123 (2021)

[8] Li, B., Zheng, B., Li, H., Li, Y.: Detail-enhanced image inpainting based on discrete wavelet transforms. Signal Processing **189**, 108278 (2021)

[9] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.S.: Generative image inpainting with contextual attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5505–5514 (2018)

[10] Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques, pp. 417–424 (2000)

[11] Shen, J., Chan, T.F.: Mathematical models for local nontexture inpaintings. SIAM Journal on Applied Mathematics **62**(3), 1019–1043 (2002)

[12] Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, pp. 341–346 (2001)

[13] Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Transactions on image processing **13**(9), 1200–1212 (2004)

[14] Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. **28**(3), 24 (2009)

[15] Liu, H., Wan, Z., Huang, W., Song, Y., Han, X., Liao, J.: Pd-gan: Probabilistic diverse gan for image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9371–9381 (2021)

[16] Guo, X., Yang, H., Huang, D.: Image inpainting via conditional texture and structure dual generation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 14134–14143 (2021)

[17] Peng, J., Liu, D., Xu, S., Li, H.: Generating diverse structure for image inpainting with hierarchical vq-vae. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 10775–10784 (2021)

[18] Wang, N., Zhang, Y., Zhang, L.: Dynamic selection network for image inpainting. IEEE Transactions on Image Processing **30**, 1784–1798 (2021)

[19] Yang, C., Lu, X., Lin, Z., Shechtman, E., Wang, O., Li, H.: High-resolution image inpainting using multi-scale neural patch synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6721–6729 (2017)

[20] Song, Y., Yang, C., Lin, Z., Liu, X., Huang, Q., Li, H., Kuo, C.C.J.: Contextual-based image inpainting: Infer, match, and translate. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 3–19 (2018)

[21] Yan, Z., Li, X., Li, M., Zuo, W., Shan, S.: Shift-net: Image inpainting via deep feature rearrangement. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 1–17 (2018)

[22] Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder

network for high-quality image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1486–1494 (2019)

[23] Liu, H., Jiang, B., Song, Y., Huang, W., Yang, C.: Rethinking image inpainting via a mutual encoder-decoder with feature equalizations. In: European Conference on Computer Vision, pp. 725–741 (2020)

[24] Li, J., Wang, N., Zhang, L., Du, B., Tao, D.: Recurrent feature reasoning for image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7760–7768 (2020)

[25] Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: European Conference on Computer Vision, pp. 1–17 (2020)

[26] Yi, Z., Tang, Q., Azizi, S., Jang, D., Xu, Z.: Contextual residual aggregation for ultra high-resolution image inpainting. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 7508–7517 (2020)

[27] Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European Conference on Computer Vision, pp. 694–711 (2016)

[28] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)

[29] Arjovsky, M., Chintala, S., , Bottou, L.: Wasserstein gan. Preprint at https://arxiv.org/abs/1701.07875 (2017)

[30] Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. Preprint at https://arxiv.org/abs/1710.10196v3 (2017)

[31] Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3606–3613 (2014)

[32] Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)

[33] Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: Gans trained by a two time-scale update rule converge to a local

nash equilibrium. Advances in neural information processing systems **30** (2017)