Signal Processing 189 (2021) 108278

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Detail-enhanced image inpainting based on discrete wavelet transforms

Bin Li[a,b], Bowei Zheng[a], Haodong Li[a,*], Yanran Li[b]

[a] Guangdong Key Laboratory of Intelligent Information Processing and Shenzhen Key Laboratory of Media Security, College of Electronics and Information Engineering, Shenzhen University, Shenzhen 518060, China
[b] Shenzhen Institute of Artificial Intelligence and Robotic for Society, Shenzhen 518000, China

ARTICLE INFO

Article history:
Received 7 March 2021
Revised 29 July 2021
Accepted 2 August 2021
Available online 4 August 2021

Keywords:
Deep learning
Image inpainting
Discrete wavelet transform

ABSTRACT

Deep-learning-based method has made great breakthroughs in image inpainting by generating visually plausible contents with reasonable semantic meaning. However, existing deep learning methods still suffer from distorted structures or blurry textures. To mitigate this problem, completing semantic structure and enhancing textural details should be considered simultaneously. To this end, we propose a two-parallel-branch completion network, where the first branch fills semantic content in spatial domain, and the second branch helps to generate high-frequency details in wavelet domain. To reconstruct an inpainted image, the output of the first branch is also decomposed by discrete wavelet transform, and the resulting low-frequency wavelet subband is used jointly with the output of the second branch. In addition, for improving the network capability in semantic understanding, a multi-level fusion module (MLFM) is designed in the first branch to enlarge the receptive field. Furthermore, drawing lessons from some traditional exemplar-based inpainting methods, we develop a free-form spatially discounted mask (SD-mask) to assign different importance priorities for the missing pixels based on their positions, enabling our method to handle missing regions with arbitrary shapes. Extensive experiments on several public datasets demonstrate that the proposed approach outperforms current state-of-the-art ones. The codes are public available at https://github.com/media-sec-lab/DWT_Inpainting.

1. Introduction

Image inpainting [1] is a kind of delicate image processing technique that reconstructs the lost or deteriorated parts within images so as to improve the visual quality. This technology can be used in many applications, such as image editing, old photo restoration, etc. As a kind of imaging inverse problems, image inpainting can be performed by using some model-based image restoration methods [2,3]. And, in the past two decades, great progress has been achieved in image inpainting through various types of tailored approaches, for example, diffusion-based ones [1,4,5], exemplar-based ones [6–8], and deep-learning-based ones [9–15]. Different from the conventional approaches that try to propagate known information or find similar or patches within the defective image to fill the missing parts, deep-learning-based approaches learn high-level deep feature representation from training data and complete the missing regions with reasonable structures and textures. As a consequence, deep inpainting approaches can achieve amazing visual effects. Typically, an encoder-decoder structure [16] based on Convolutional Neural Network (CNN) and a Generative Adversarial Network (GAN) mechanism [17] are working together to perform the deep inpainting task. In specific, a generation network constructed in an encoder-decoder structure equipped with a well-designed training loss function is used for missing area completion, while a discriminator is employed for providing adversarial loss to ensure the inpainted images have indistinguishable visual appearance compared to pristine images.

In general, both reasonable semantic contents and fine details are needed to be synthesized when performing inpainting. Some methods [9–13] solve these two goals in a single network, and some [14,15] use two serial networks, i.e., a coarse network and a refinement network, for dealing with coarse and fine contents, respectively. However, they still suffer from distorted structures and/or blurry textures, implying that a different inpainting architecture may be needed. In this paper, we address the above mentioned problem by designing a deep inpainting architecture based on a two-parallel-branch completion network, namely a content network and a texture network. The content branch fills seman-

* Corresponding author.
E-mail addresses: libin@szu.edu.cn (B. Li), 1800261056@email.szu.edu.cn (B. Zheng), lihaodong@szu.edu.cn (H. Li), lyran@szu.edu.cn (Y. Li).

https://doi.org/10.1016/j.sigpro.2021.108278
0165-1684/

tic content in spatial domain, and the texture branch generates high-frequency details in wavelet domain. In specific, the content branch with a U-net structure takes an image with missing parts as input and outputs the spatially inpainted image. A multi-level fusion module based on dilated gated convolution [15] is proposed to expand the network receptive field so as to improve its capability in semantic understanding. On the other hand, the texture branch takes the discrete wavelet transform (DWT) high-frequency subbands as input and processes the high-frequency part of the inpainted image in wavelet domain. In this way, the first branch can focus more on the semantic contents while the second one can learn better image textural details. To synchronize the outputs of these two branches, the low-frequency wavelet subband of the spatially inpainted image from the content branch and the high-frequency wavelet subbands from the texture branch jointly reconstruct an inpainted image through inverse discrete wavelet transform (IDWT). Furthermore, drawing lessons from some traditional exemplar-based inpainting methods where the missing pixels closer to the known areas have higher inpainting priorities, we develop a spatially discounted mask (SD-mask) for arbitrary missing shape in the loss function to evaluate the roles of missing pixels with different importance. Hence, the missing pixels on the boundaries have higher impact for the loss so that they can be recovered better to make the boundary less abrupt.

The contributions of this paper are summarized as follows.

- We propose a two-parallel-branch network to complete image structure and fill high-frequency details based on DWT, which can produce reasonable and sharp image contents.
- We design a multi-level fusion module based on dilated gated convolution to expand the receptive field of the content branch, enabling the network to learn image semantic contents at different scales.
- We develop the form of spatially discounted mask to evaluate the roles of missing pixels with different importance, which can be applied to missing areas with arbitrary shapes.

2. Related works

The existing inpainting works can be categorized into two types. The first type, developed with the traditional paradigms, adopts the known information within the given image to fill lost contents at image pixel/patch level. The second type, exploiting the outstanding learning capability of CNN, predicts and fills the missing contents at feature level.

The traditional inpainting works include diffusion-based [1,4,5], sparsity-based [3,18,19], and exemplar-based approaches [6–8]. The basic idea of diffusion-based inpainting is to propagate the known information into the missing area. For example, Bertalmio et al. [1] proposed to spreads the information around the missing area iteratively from the border to the inner along isophote direction. Such methods can fill small missing areas, such as the cracks in old photos, but they are not good for large missing areas. In sparsity-based inpainting, a given image is sparsely represented by a redundant system, which is constructed by a set of transforms. In this way, the missing area can be inferred by updating the sparse representation. Many types of transforms have been utilized, especially framelets [3,19,20]. Similar as diffusion-based inpainting, sparsity-based inpainting also brings blurring effect for large missing areas. The key of exemplar-based inpainting is to find patches similar to the missing content in the known area, and then paste them to fill in the missing content [6]. However, it takes a long time to fill in an image when the similar patches are found based on an exhaustive search. To reduce the time consumption, Barnes et al. [8] designed the PatchMatch algorithm to search closest patches and fill the missing content, while the obtained patches are often not as accurate as that from an exhaustive search. A common shortcoming of the traditional inpainting approaches is that the missing contents are usually inferred from only the known information within the given image, hence no novel semantic contents can be produced, such as a nose in a face image.

In recent years, the performance of inpainting has been significantly improved by deep-learning-based models with well-designed loss functions. By learning hierarchical image representations, inpainting networks can fill missing areas with much better visual quality and even create new objects. Pathak et al. [9] pioneered an encoder-decoder network architecture, where the encoder network was fed with an image having missing parts and produced a latent feature representation of that image, and the decoder network took the latent feature and produced missing parts. Both reconstruction loss and adversarial loss were used for making prediction plausible. Iizuka et al. [10] proposed an architecture by considering global and local consistency with a global and a local context discriminator for providing adversarial loss. The overall visual consistency as well as textural details can be improved for a wide variety of scenes. Yu et al. [14] proposed a contextual attention layer to learn compatible features from known background patches for producing missing patches. The use of partial convolutions [11] or gated convolutions [15] can let the learned features be conditioned on valid pixels close to the deteriorated area. As most inpainting methods only output a corresponding predicted result for a given image, which is not flexible enough, Zheng et al. [12] proposed an inpainting structure that can generate different types of inpainting results, increasing the diversity of image inpainting. To enhance the quality of inpainted images, Zeng et al. [13] proposed to combine low-level feature with high-level feature. In addition to directly performing inpainting in spatial domain, we notice that the recent deep-learning-based methods [21,22] tried to complete missing areas by using wavelet information. In [21], an input image is processed by an encoder-decoder network equipped with hierarchical DWT and IDWT. In [22], a multi-frequency probabilistic inference model is used to ensure the distributions of inpainted images in latent spaces are close to those of ground-truth images. Although these methods operate in wavelet domain, they predict the missing information in different wavelet subbands with a single-branch completion network. In this way, the networks must learn to predict image semantic contents and textural details simultaneously, which might not be an optimized option. To mitigate this drawback, in this paper we develop a two-branch completion network that predicts image contents and textural details separately.

3. Methodology

As depicted in Fig. 1, the proposed image inpainting framework consists of a completion network G and a discriminator network D, and each network is comprised of two parts. G consists of a content branch G_{con} and a texture branch G_{txt}, while D consists of a global discriminator D_{glb} and a local discriminator D_{loc}. In the training phase, both G and D work together, while in the deployment phase, only G is employed for image completion.

Given a damaged image I_m, which can be considered as a ground-truth image I_{gt} deteriorated by a binary missing area mask M (1 for missing pixel and 0 for known pixel), i.e., $I_m = I_{gt} \odot (1 - M)$ ($\odot$ denotes element-wise multiplication), a completion image I_{com} can be obtained by G:

$$I_{com} = \mathrm{G}(I_m) \odot M + I_m. \tag{1}$$

To ensure the visual quality of I_{com}, some loss functions are designed for I_{com} and the intermediate feature maps of G, where the SD-mask is employed to assign different importance for the missing pixels based on their positions. Meanwhile, D takes both I_{com}

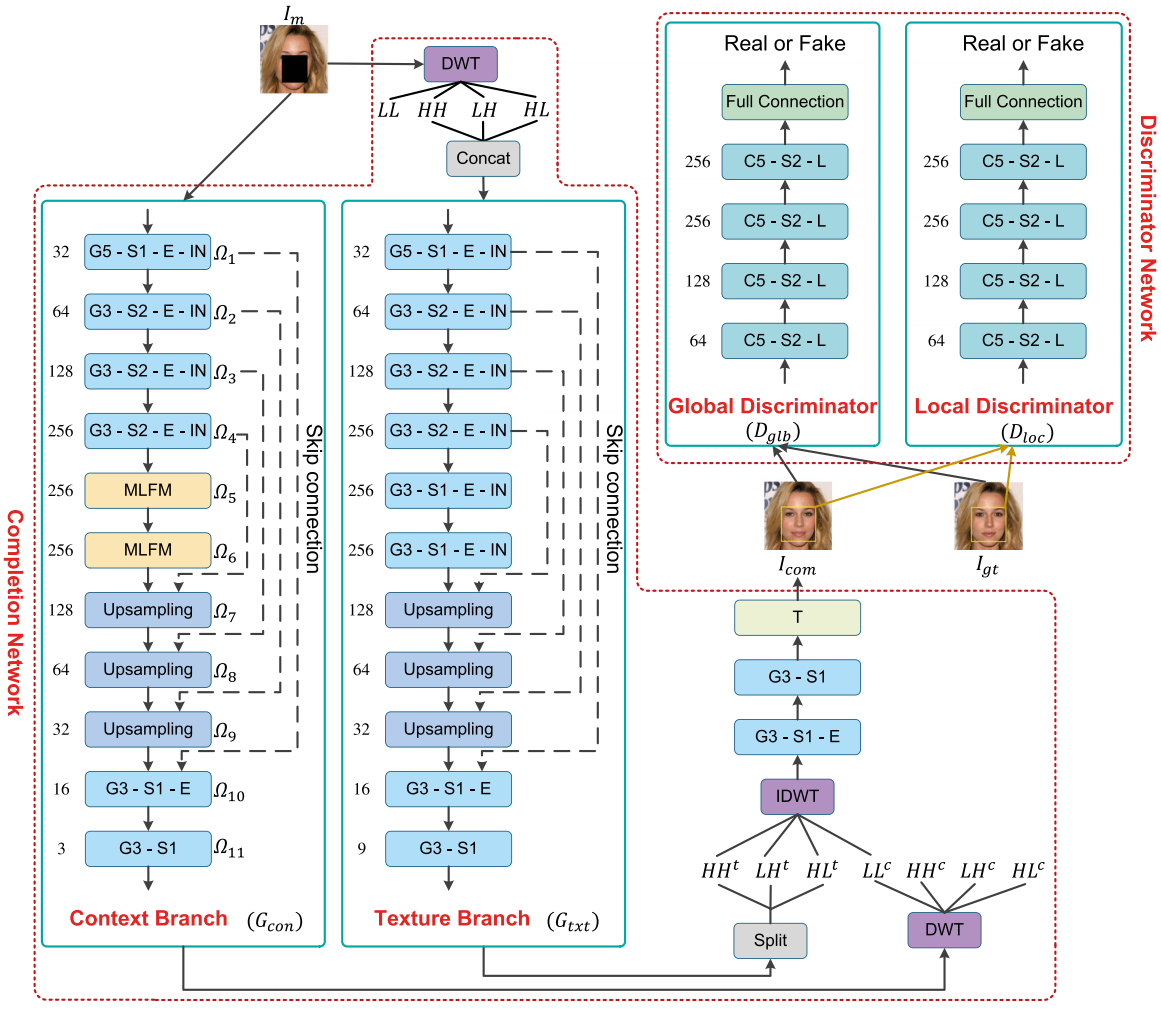

Fig. 1. The framework of the proposed method. In the figure, "Ga - Sb - E - IN" indicates $a \times a$ gated convolution - stride b - ELU activation function - Instance Normalization. "Ca - Sb - L" indicates $a \times a$ vanilla convolution - stride b - LeakyRelu activation function. "T" indicates Tanh activation function. The dilation rate of all convolutions is 1 except MLFM. Ω_i ($i \in \{1, 2, \ldots, 11\}$) denotes the output feature of the ith operation group. "Upsampling" denotes the proposed upsampling block. Daubechies2 (db2) is chosen as the wavelet basis in our method.

and I_{gt} as input, and provides adversarial training loss for G, so that the obtained I_{com} could be indistinguishable from I_{gt} in the perspective of D.

In the following subsections, we first elaborate the proposed completion network equipped with discrete wavelet transform (DWT) and the discriminator network, and then introduce the SD-mask as well as the training loss function.

3.1. Completion network

In the completion network, the two parallel branches G_{con} and G_{txt} process image contents and image textural details, respectively. G_{con} takes I_m as input to predict the semantic contents of the missing area in spatial domain. On the other hand, I_m is decomposed by DWT and processed by G_{txt} in the wavelet domain. In specific, only the three high-frequency subbands (HL, LH, HH) are fed to G_{txt} to obtain the corresponding texture-filling outputs (HL^t, LH^t, HH^t). To synchronize the outputs of two network branches, the output of the G_{con} is also decomposed by DWT and the inpainted low-frequency subband (LL^c) is employed together with the three inpainted high-frequency subbands (HL^t, LH^t, HH^t) to perform inverse DWT (IDWT). The resultant spatial domain inpainted image is processed with two convolution layers in order to better synchronize/fuse the outcomes of the two different network branches.

Please note that only one-level DWT is performed in the completion network. We have investigated the effect of the decomposition level of DWT on the inpainting performance. The experimental results showed that two-level DWT slightly underperformed one-level DWT, while three-level DWT resulted in poorer performance. It may be due to the fact that the more levels of DWT are used, the more learnable parameters in the network, and the easier for the network to get overfitting if there is no other well-defined regularization term in the loss function. Hence, we use one-level DWT for simplicity.

To highlight important information of the input feature through the dynamic learnable mechanism, the generalized gated convolution [15] is utilized in both branch networks, which can be expressed as:

$$Y_j = \phi\left(\sum_i W_i^f * X_i + b\right) \odot \sigma\left(\sum_i W_i^g * X_i + c\right), \tag{2}$$

where X_i and Y_j are respectively the ith input feature map and the jth output feature map in a gated convolution block, W_i^f and W_i^g are convolution kernels, and b and c are the bias. σ is the Sigmoid function for gating, and ϕ is the activation function for the learned

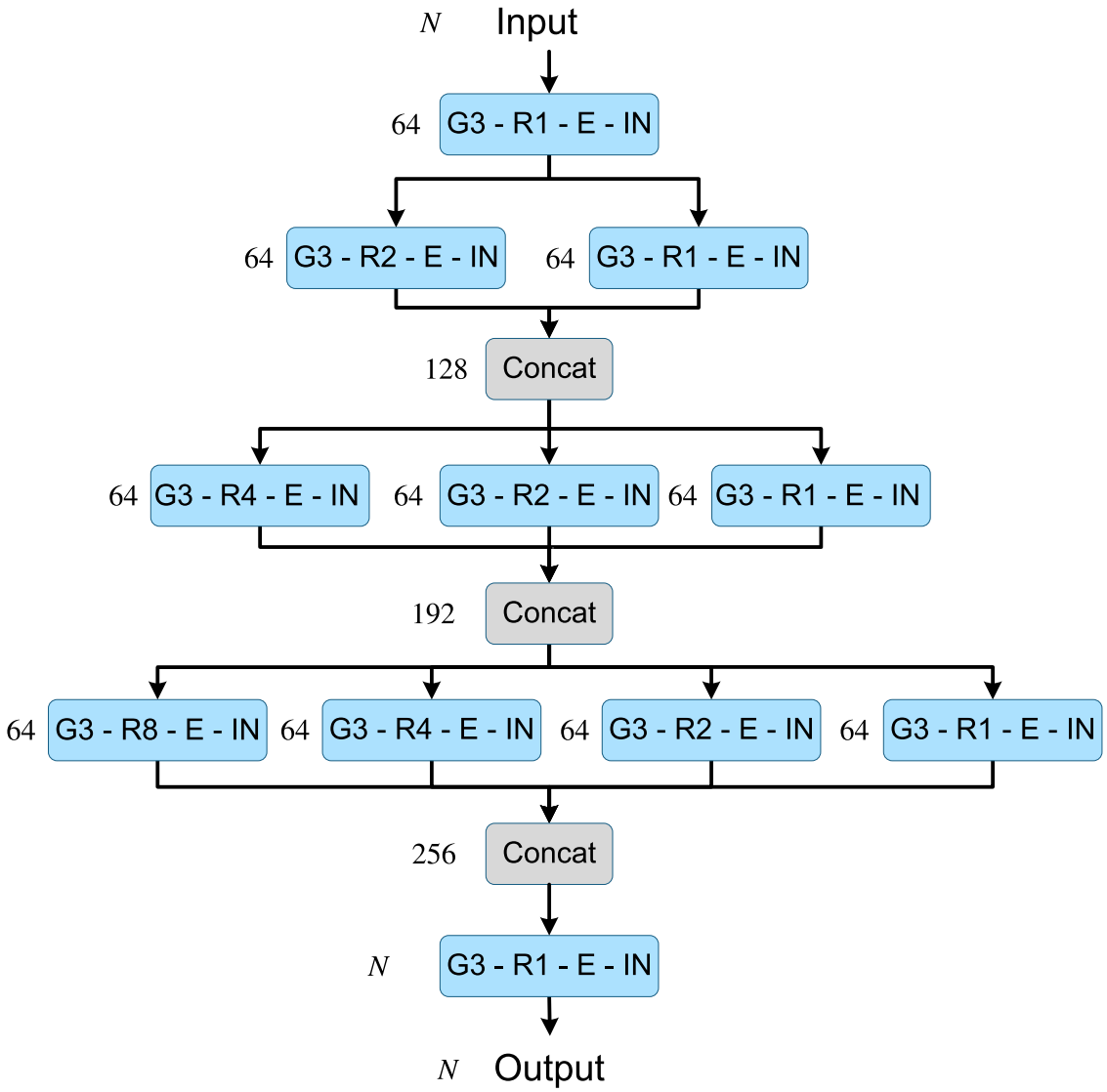

Fig. 2. The structure of the multi-level fusion module (MLFM). "Ga - Rc - E - IN" indicates $a \times a$ gated convolution - dilation rate c - ELU activation function - Instance Normalization. The stride of all convolutions is 1. "N" indicates the number of the output channel of convolution.

features, where Exponential Linear Unit (ELU) [23] is used in our work.

3.1.1. Content branch

The goal of G_{con} is to ensure that the final completion image I_{com} has reasonable image structures and semantic contents. As shown in Fig. 1, G_{con} is built with an encoder-decoder U-net structure [24].

At the encoder side, four convolution groups are first used, where 5×5 gated convolution with the stride of 1 is used in the first group, and 3×3 gated convolutions with the stride of 2 are used in the subsequent three groups. Since the size of receptive field plays an important role in semantic understanding, we design the multi-level fusion module (MLFM) to expand the receptive field and fuse different levels of information behind the four normal convolution groups. As shown in Fig. 2, there are five convolution groups in a MLFM. In the first four groups, the receptive field is gradually expanded by adding more number of convolutions and increasing the dilated rate. At the end of the second to the fourth group, the features with different sizes of receptive field are fused together via concatenation, so that the content network can learn features in different scales. Finally, a 3×3 gated convolution is used to recover the input feature dimension. In total, two MLFMs are adopted in the encoder.

At the decoder side, three upsampling blocks are used to restore the input dimension. As shown in Fig. 3, there are two inputs in each upsampling block. One is from the last output layer while the other is from the skip connection. They are firstly concatenated together and then subjected to two kinds of operations, i.e., a 4×4 deconvolution with the stride of 2 followed by a 3×3 convolution ($deconv + conv$), and a 2 times bilinear interpolation followed by a 3×3 convolution ($bilinear + conv$). The two upsampled features are merged and finally fused by a 3×3 gated convolution. For $deconv + conv$, the deconvolution can enlarge image size and reduce feature dimension, and then the convolution is used to reduce checkerboard artifacts. However, the checkerboard artifacts may not be fully eliminated. For $bilinear + conv$, the bilinear interpolation can enlarge the image size and the convolution can eliminate the checkerboard artifacts, but it produces obscure details compared to deconvolution. Our proposed upsampling block can take the advantages of both deconvolution and interpolation, resulting better results. This has been demonstrated through ablation study (please refer to Section 4.3.3). To obtain the output of

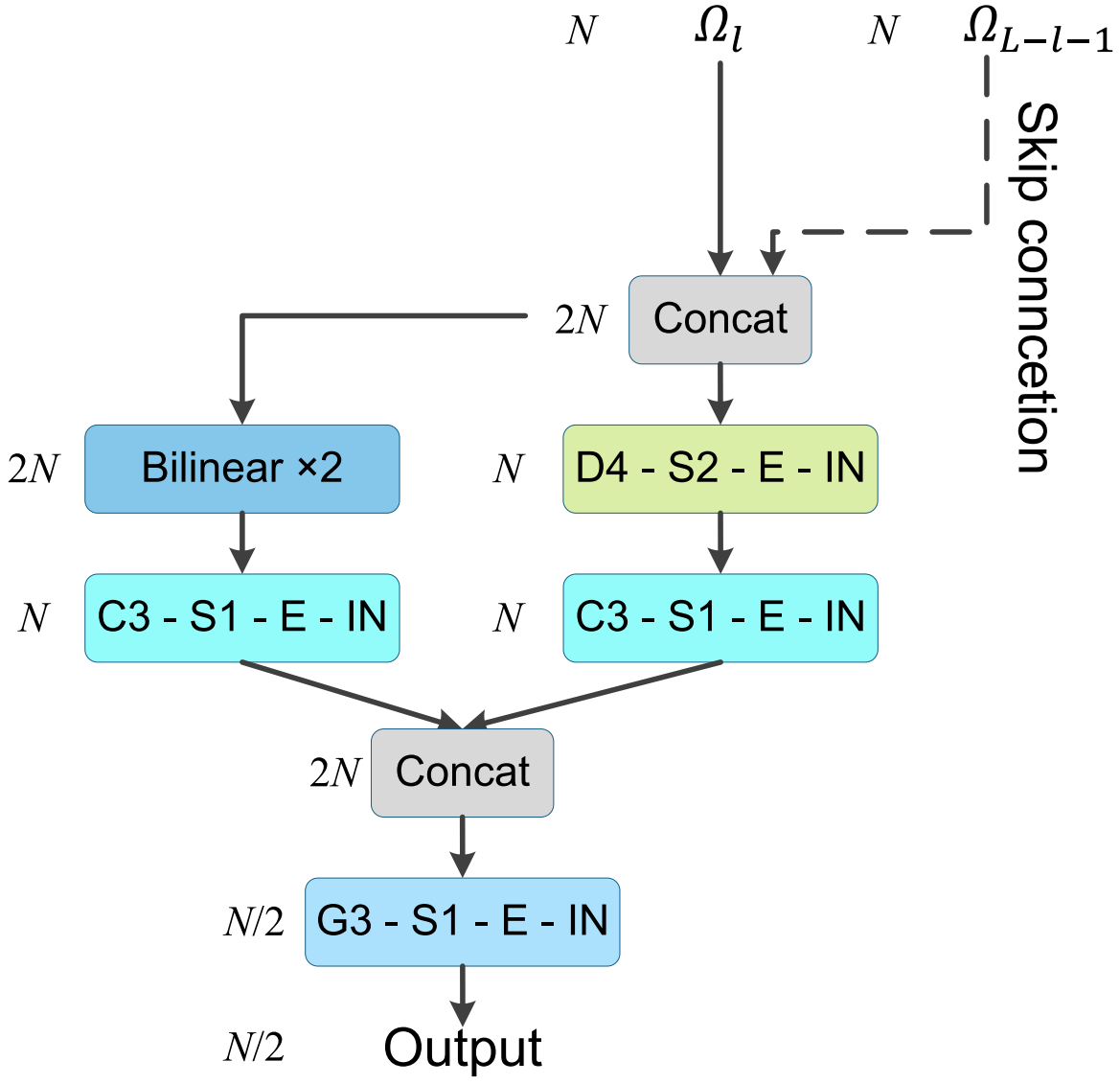

Fig. 3. The structure of the upsampling block. "Da - Sb - E - IN" indicates $a \times a$ deconvolution - stride b - ELU activation function - Instance Normalization. The dilation rate of all convolutions is 1. "N" indicates the number of the output channel of convolution. Ω_l denotes the output feature of the lth operation group of G_{con} or G_{txt} in Fig. 1. Take G_{con} for example, there are 11 operation groups, hence $L = 11$ and $l \in \{6, 7, 8, 9\}$.

G_{con}, two groups of 3×3 gated convolutions are applied after three upsampling blocks.

3.1.2. Texture branch

The goal of G_{txt} is to fill the missing area of wavelet high-frequency subbands, thus it takes the concatenation of three wavelet high-frequency subbands with missing area as input. The network structure of G_{txt} is similar to that of G_{con}, except that MLFM is not employed. This is because unlike image structure which may require a large receptive field for completion, image texture is usually locally correlated and only needs a small receptive field.

3.2. Discriminator network

Similar to Yu et al. [14], a global discriminator (D_{glb}) and a local discriminator (D_{loc}) are used to distinguish ground-truth images (I_{gt}) from completed ones (I_{com}), thus providing adversarial information for training the completion network. The D_{glb} takes the full images as input, while the D_{loc} takes local image patches containing the missing parts as input. In our work, we use the full image of size 256×256, and use the image patch of size 128×128. The detailed network structures are depicted in Fig. 1.

3.3. Spatially discounted mask

In [14], Yu et al. presented a *spatially discounted mask* (SD-mask) in the content reconstruction loss function, expecting that the boundary missing pixels should have larger weighting values in the loss to make them easier to converge. This is consistent with the idea of the traditional exemplar-based method [6] where the boundary pixels in the missing area have more known information than the inner pixels so they should have greater confidence.

Given a binary mask M, the SD-mask M_{sd} in Yu et al. [14] is obtained as:

$$m'_{i,j} = \begin{cases} 0, & m_{i,j} = 0, \\ \gamma^{l_{i,j}}, & m_{i,j} = 1, \end{cases} \tag{3}$$

where $m_{i,j} \in M$ and $m'_{i,j} \in M_{sd}$, $l_{i,j}$ is the distance between the (i, j)th missing pixel and its nearest pixel with known value, and γ is set as 0.99. Note that implementation in Yu et al. [14] only

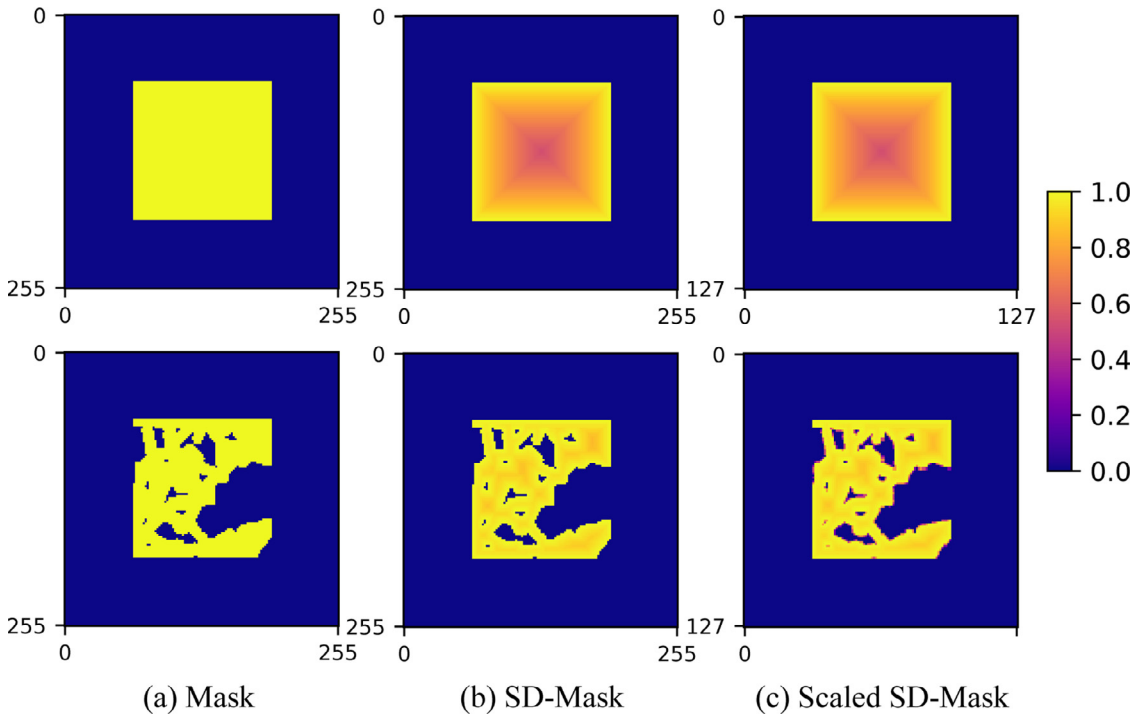

Fig. 4. Results of our *spatially discounted mask* (SD-mask).

considers a missing area of rectangular shape, which is inapplicable for a missing area of arbitrary shape. Besides, when the size of a feature map is reduced, how to process the corresponding SD-mask is unclear.

We address the problem by computing the SD-mask as:

$$m''_{i,j} = \begin{cases} 0, & m_{i,j} = 0, \\ \gamma^{t_{i,j}-1}, & m_{i,j} = 1, \end{cases} \tag{4}$$

where $m_{i,j} \in M$ and $m''_{i,j} \in M_{sd}$, γ is set as 0.99, and $t_{i,j}$ is the times of erosion by performing erode operation on M with a 3×3 kernel with all one values to turn $m_{i,j} = 1$ into $m_{i,j} = 0$. In this way, there is no need to compute the distance for each missing pixel, greatly improving the implementation efficiency. For a feature map with reduced resize, the corresponding SD-mask can be resized from the full-size SD-mask without performing the operation of (4) again. We use $M_{sd}^{(1/2^i)}$ to denote a resized SD-mask when the area of M_{sd} is reduced to its $1/2^{2i}$. An example of rectangular and arbitrary shape missing areas and their corresponding SD-masks are given in Fig. 4.

3.4. Loss function

The overall loss function of G is defined by

$$L_G = \omega_1 L_{con} + \omega_2 L_{dwt} + \omega_3 L_{per} + \omega_4 L_{sty} + \omega_5 L^G_{adv}, \tag{5}$$

where L_{con}, L_{dwt}, L_{per}, L_{sty}, and L^G_{adv} are the content loss, DWT loss, perceptual loss [25], style loss [26], and generator adversarial loss [27], respectively. The corresponding weighting factors are empirically set as $\omega_1 = 10.0$, $\omega_2 = 10.0$, $\omega_3 = 1.0$, $\omega_4 = 1000$, and $\omega_5 = 0.001$, so that these losses are in the similar magnitude.

The loss function of D is defined by $L_D = L^D_{adv}$, where L^D_{adv} is discriminator adversarial loss.

3.4.1. Content loss

We use L-1 norm to evaluate the pixel content loss between I_{com} and I_{gt} in the missing area covered by the SD-mask:

$$L_{con} = ||(I_{com} - I_{gt}) \odot M_{sd}||_1. \tag{6}$$

Compared to L-2 norm, the non-smooth nature of L-1 norm can ensure that the details of the inpainted images are not blurred.

3.4.2. DWT loss

We use DWT loss to ensure the high-frequency wavelet subbands generated by the texture branch and the low-frequency wavelet subband generated by the content branch are close to their ground-truth. Since the size of the DWT subband is reduced compared to the input image size, the corresponding SD-mask is also reduced by performing bilinear resizing. The DWT loss is defined as:

$$\begin{aligned} L_{dwt} = & ||(HL^t - HL^{gt}) \odot M_{sd}^{(1/2)}||_1 \\ & + ||(LH^t - LH^{gt}) \odot M_{sd}^{(1/2)}||_1 \\ & + ||(HH^t - HH^{gt}) \odot M_{sd}^{(1/2)}||_1 \\ & + ||(LL^c - LL^{gt}) \odot M_{sd}^{(1/2)}||_1. \end{aligned} \tag{7}$$

3.4.3. Perceptual loss

We use perceptual loss [25] to ensure the similarity between I_{gt} and I_{com} in different levels of feature feature representations. A VGG19 [28] model pretrained on ImageNet dataset [29] is used for computing perceptual loss, that is

$$L_{per} = \Sigma_{i=0}^{4} ||(\varphi^{Q_i}_{I_{gt}} - \varphi^{Q_i}_{I_{com}}) \odot M_{sd}^{(1/2^i)}||_1, \tag{8}$$

where φ^{Q_0}, φ^{Q_1}, φ^{Q_2}, φ^{Q_3}, and φ^{Q_4} denote the feature maps output by the layers of *relu*1_1, *relu*2_1, *relu*3_1, *relu*4_1, and *relu*5_1 in VGG19, respectively.

3.4.4. Style loss

The main purpose of style loss is to ensure the style similarity between I_{gt} and I_{com}. To compute the style loss, Gram matrix ($\mathcal{G}[\cdot]$) is firstly applied to a feature map x, namely

$$\mathcal{G}[x] = \frac{f(x) * f^T(x)}{H_x W_x C_x}, \tag{9}$$

Table 1
Training, testing, and validation splits of the datasets.

Dataset	Training	Testing	Validation
CelebA-HQ	21,000	6000	3000
DTD	3947	1132	564
Facade	424	122	60
PSV	13,410	100	1490
Places2	1,808,460	50,000	36,500

where H_x, W_x and C_x are the height, width and channel of x, and $f(x)$ reshapes the $H_x \times W_x \times C_x$ feature map x to the size of $(H_x W_x) \times C_x$ while $f^T(x)$ means the transpose of $f(x)$. The value of style loss is given by the summation of L-1 norms on the Gram matrix results:

$$L_{sty} = \Sigma_{i=0}^{3} ||\mathcal{G}[\varphi_{I_{gt}}^{P_i} \odot M_{sd}^{(1/2^i)}] - \mathcal{G}[\varphi_{I_{com}}^{P_i} \odot M_{sd}^{(1/2^i)}]||_1, \tag{10}$$

where φ^{P_0}, φ^{P_1}, φ^{P_2}, and φ^{P_3} denote the feature maps output by the layers of *relu*2_2, *relu*3_4, *relu*4_4 and *relu*5_2 in VGG19, respectively.

3.4.5. Adversarial loss

Both the global and local discriminator networks adopt a loss function from WGAN-GP (Wasserstein GAN-Gradient Penalty) [27]. Denote (x_r, x_g) as a $(real, genreated)$ data pair respectively sampled from ground-truth images (I_{gt}) and inpainted images (I_{com}). The discriminator adversarial loss is set as

$$\begin{aligned} L_{adv}^{D} = & -\mathbb{E}_{x_r}[D_{loc}(\mathcal{C}[x_r])] + \mathbb{E}_{x_g}[D_{loc}(\mathcal{C}[x_g])] \\ & +\lambda \mathbb{E}_{\hat{x}}[(||\nabla_{\hat{x}} D_{loc}(\mathcal{C}[\hat{x}])||_2 - 1)^2] \\ & -\mathbb{E}_{x_r}[D_{glb}(x_r)] + \mathbb{E}_{x_g}[D_{glb}(x_g)] \\ & +\lambda \mathbb{E}_{\hat{x}}[(||\nabla_{\hat{x}} D_{glb}(\hat{x})||_2 - 1)^2], \end{aligned} \tag{11}$$

where $\mathbb{E}$ is the expectation, $\mathcal{C}[\cdot]$ means to crop a local area (e.g., 128×128) from the full size image, $\hat{x} = \epsilon x_r + (1-\epsilon) x_g$ ($\epsilon \sim$ U[0, 1]) is a random sample, and λ is the gradient penalty factor which is set as 10.

The generator adversarial loss is set as

$$L_{adv}^{G} = -\mathbb{E}_{x_g}[D_{loc}(\mathcal{C}[x_g])] - \mathbb{E}_{x_g}[D_{glb}(x_g)]. \tag{12}$$

4. Experiments

4.1. Experimental settings

4.1.1. Datasets

We evaluated the proposed method on five datasets, i.e., CelebA-HQ [30], Describable Textures Dataset (DTD) [31], Facade [32], Paris Street View (PSV) [9] and Places2 [33]. We divided the first three datasets into training, validation, and testing sets with the portions of 70%, 10%, and 20%. The PSV dataset has already been divided into a training set and a testing set, so we randomly selected 90% of the training data for training and used the rest 10% for validation. The Places2 dataset has already been divided into training, validation, and testing sets, and thus we adopted them directly. The details are shown in Table 1.

4.1.2. Comparative study

We compared with five existing image inpainting algorithms, including the Exemplar-based Inpainting (EBIN) [6], ContextEncoder (CE) [9], Contextual Attention (CA) [14], Gated Convolution (GConv) [15], Pluralistic Image Completion (PICNet) [12] and Pyramid-Context Encoder Network (PENet) [13]. Some important technical details of these methods and the proposed one are shown in Table 2. All the methods used for comparisons have publicly available implementation code.[1] For certain datasets, some already-trained models are available with the implementations, and these models were used as-is. For other datasets, we trained models accordingly and selected the best ones by using the same protocols as ours.

4.1.3. Performance metrics

In order to make the comparison more comprehensive, six commonly used quantitative metrics were adopted to evaluate the performance of different methods, i.e., Mean Absolute Loss (MAE), Peak Signal to Noise Ratio (PSNR), Structural SIMilarity (SSIM), Frechet Inception Distance (FID) [36], Learned Perceptual Image Patch Similarity (LPIPS) [37] and Deep Image Structure and Texture Similarity (DISTS) [38]. The first three metrics (MAE, PSNR and SSIM) are directly calculated on spatial images, while the rest (FID, LPIPS and DISTS) are calculated on feature maps output from deep networks.

4.1.4. Implementation details

In the training stage, the model was trained to inpaint a missing area within a 256×256 image. The missing area could locate at any position in a random 128×128 area and be of arbitrary shape. In the testing stage, the model can also be applied to multiple missing areas, whose positions are not limited in a 128×128 area. For the CelebA-HQ dataset, the square images were directly scaled to 256×256. For other datasets, the longest side of an image was randomly cropped as the same as its shortest side, and then the image was scaled to 256×256. We mainly followed [15] to generate masks of free-form for the training images. A slight difference was that the masks were generated in a random 128×128 local image area. Hence, the locations of the 128×128 areas for a batch of images are the same, while the locations would be different for different batches. For the validation images, the missing regions are always in the central 128×128 areas. We used Adam optimizer [39] with $\beta_1 = 0.5$ and $\beta_2 = 0.9$, and set the learning rate as 1×10^{-4} and the batch size as 8. The proposed network was trained to reach a total of 1,000,000 iterations and validation was conducted every 5000 iterations. The model achieved the minimum DISTS distance on the validation set was chosen as the best model. Daubechies2 (db2) was chosen as the wavelet basis in our method. Our model was implemented with the PyTorch framework and operated on a Tesla-P100 GPU (16 GB memory).

4.2. Comparative results

We first compare the visual effects for inpainting a central 128×128 square area with existing works, as shown in Fig. 5. It can be seen that EBIN [6] often generates unreasonable contents in many cases, since traditional inpainting methods have no image semantic understanding ability. We also observe that CE [9] and CA [14] tend to produce distorted structure, while GConv [15] and PICNet [12] can not generate fine-grained texture in complex images (i.e., DTD dataset). Besides, the facial hair details produced by PENet [13] are unsatisfactory. In general, our method shows better inpainting quality in the overall content semantic and structure.

We then conduct quantitative analysis. Specifically, we calculate the six evaluation metrics with the testing datasets, where the inpainted areas are set to the central 128×128 square areas. From Table 3, it is observed that our method achieves the best results

[1] Available at https://github.com/jonzhaocn/exemplar-based-image-inpainting, https://github.com/BoyuanJiang/context_encoder_pytorch, https://github.com/JiahuiYu/generative_inpainting/tree/v1.0.0, https://github.com/JiahuiYu/generative_inpainting, https://github.com/lyndonzheng/Pluralistic-Inpainting, and https://github.com/researchmm/PEN-Net-for-Inpainting. Except for the CE algorithm and EBIN algorithm, the others are official implementations.

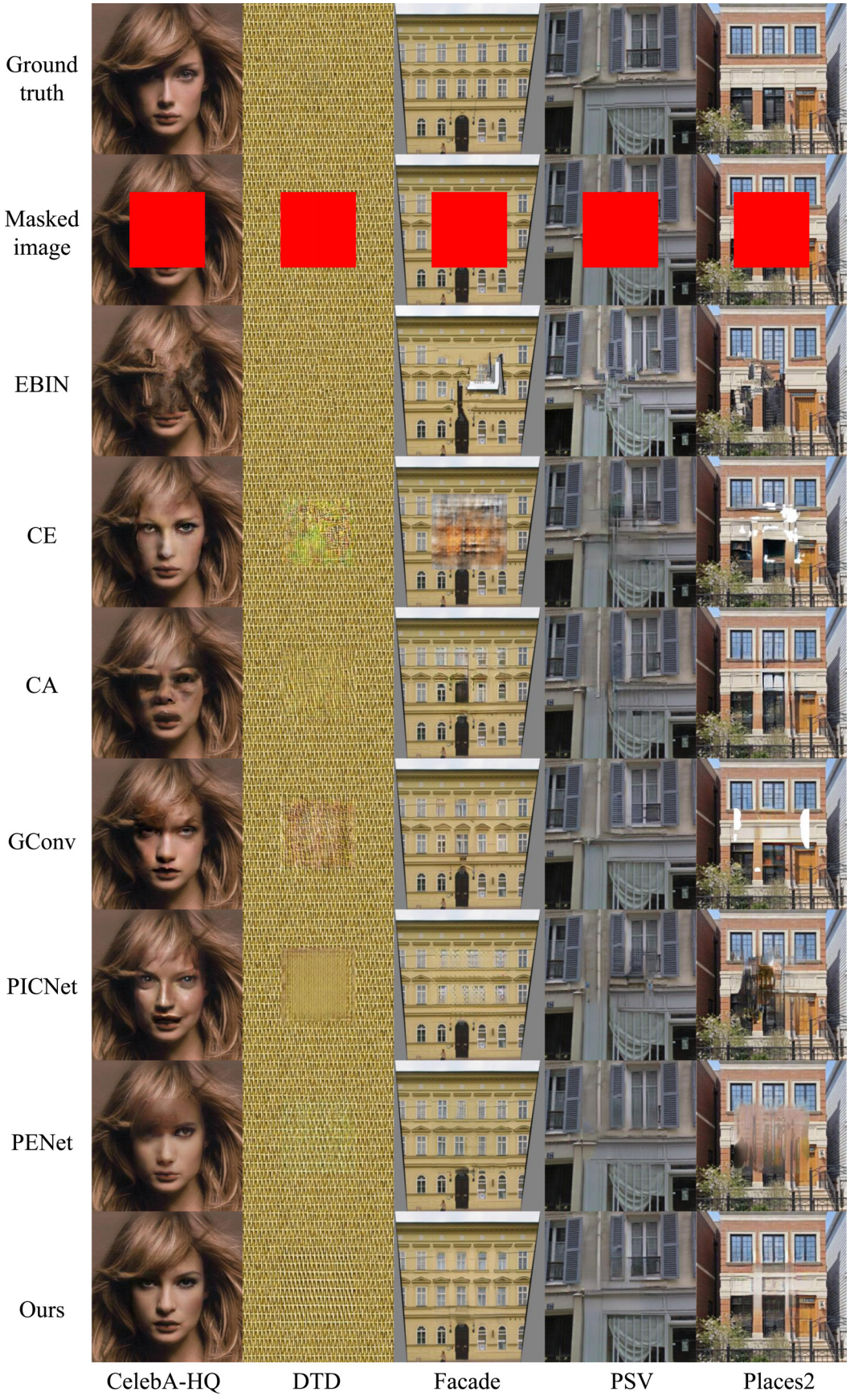

Fig. 5. Example results of different methods.

Table 2
The technical details of different methods.

Method	Mask shape in training stage	Receptive field[2]	Adversarial loss
CE [9]	rectangle	256	Primitive GAN [34]
CA [14]	rectangle	256	WGAN-GP [27]
GConv [15]	free-form + rectangle	256	Hinge Loss [35]
PICNet [12]	free-form	256	Hinge Loss [35]
PENet [13]	rectangle	127	Hinge Loss [35]
Ours	free-form	256(Max), 179(Min)	WGAN-GP [27]

Table 3
Quantitative results of different methods for each testing dataset. The inpainted area is a central 128 × 128 square area. ↑/↓ means higher/lower is better.

Dataset	Method	PSNR↑	SSIM↑	MAE↓	FID↓	LPIPS↓	DISTS↓
CelebA-HQ	EBIN	17.71	0.7443	13.27	92.98	0.2093	0.1709
	CE	26.03	0.8286	4.840	5.334	0.0916	0.0442
	CA	23.99	0.7972	6.014	6.220	0.1034	0.0506
	GConv	25.48	0.8162	5.018	**4.200**	0.0901	0.0426
	PICNet	24.24	0.8067	5.624	5.315	0.0921	0.0428
	PENet	25.49	0.8164	4.804	7.679	0.0967	0.0467
	Ours	**26.17**	**0.8390**	**4.371**	5.168	**0.0735**	**0.0356**
DTD	EBIN	20.18	0.7488	15.06	**32.15**	0.1414	0.0893
	CE	20.40	0.7278	15.21	77.01	0.1995	0.1610
	CA	22.80	0.7676	7.739	33.89	0.1389	0.0788
	GConv	23.13	0.7680	7.402	33.66	0.1356	0.0741
	PICNet	23.03	0.7546	7.495	45.88	0.1595	0.1052
	PENet	**24.19**	0.7678	**6.638**	47.57	0.1577	0.0940
	Ours	23.60	**0.7732**	6.981	32.97	**0.1344**	**0.0689**
Facade	EBIN	10.41	0.0143	63.41	46.88	0.6733	0.3275
	CE	20.04	0.7316	13.91	83.53	0.2091	0.1354
	CA	22.87	0.8082	6.876	32.47	0.1126	0.0503
	GConv	**23.92**	**0.8203**	**6.023**	**24.56**	0.1184	0.0502
	PENet	23.09	0.7933	6.719	37.29	0.1253	0.0545
	PICNet	23.47	0.8000	6.621	37.14	0.1207	0.0549
	Ours	23.15	0.7973	6.788	28.21	**0.1088**	**0.0442**
PSV	EBIN	19.48	0.7430	16.40	63.51	0.1665	0.0979
	CE	21.75	0.7524	14.09	58.49	0.1639	0.0916
	CA	22.95	0.7768	7.336	53.78	0.1444	0.0824
	GConv	24.12	0.7874	6.225	49.11	0.1351	0.0752
	PENet	22.44	0.7605	7.899	56.36	0.1593	0.0939
	PICNet	24.22	0.7700	6.170	57.09	0.1561	0.0861
	Ours	**25.42**	**0.7994**	**5.197**	**37.10**	**0.1208**	**0.0641**
Places2	EBIN	18.55	0.7505	12.87	10.46	0.1711	0.1126
	CE	21.05	0.7424	9.571	10.12	0.1557	0.1095
	CA	20.56	0.7644	8.940	**6.208**	0.1536	0.1031
	GConv	20.17	0.7686	9.341	6.651	0.1506	0.1058
	PENet	19.87	0.7516	10.00	12.79	0.1739	0.1219
	PICNet	21.18	0.7518	8.566	19.54	0.1856	0.1244
	Ours	**21.73**	**0.7715**	**7.867**	9.787	**0.1504**	**0.0985**

for all the datasets in terms of LPIPS [37] and DISTS [38], which are two state-of-the-art indicators for image quality assessment (IQA), meaning that the images inpainted by our method are of better visual quality. On the other hand, our method also performs the best on most of the rest metrics for the CelebA-HQ, PSV, DTD, and Places2 datasets. For Facade dataset, the results obtained by our method are just a little bit worse than those of GConv [15].

Finally, we present the results for multiple missing area with free-form masks. As shown in Fig. 6, our method can generate finer details as well as better structures compared to GConv [15] and PICNet [12].

4.3. Ablation study

In this subsection, we report ablation study results to analyze the selection of wavelet basis, the structure of completion network, and the structure of upsampling block, respectively, and also show the effectiveness of the proposed SD-mask. All experiments were conducted on the CelebA-HQ dataset[30].

[2] For image of size 256 × 256.

4.3.1. Ablation study on wavelet basis

In this experiment, we chose a series of wavelet bases on Daubechies *N* ("db*N*"). Following the same parameter setting and model selection protocol mentioned above, we trained four different models corresponding to four different wavelet bases (i.e., db1, db2, db3, and db4). The obtained quantization metrics and qualitative results are shown in Table 4 and Fig. 7, respectively. It can be seen that all four models are comparable, while the model equipped with "db2" slightly outperforms others. In addition, we also tried some non-orthogonal wavelets such as framelet [20] for decomposition. As shown in Table 4 and Fig. 7, the performance was comparable or slightly inferior to "db2" wavelet. The experimental results indicate that the selection of wavelet basis may not be an important factor. Considering its satisfactory performance and common usability, "db2" was selected as the default wavelet basis in our model.

4.3.2. Ablation study on the structure of completion network

To verify the benefits of using two branches in our completion network, we evaluated the performance by using each branch individually. To this end, for the G_{txt} branch, the network input

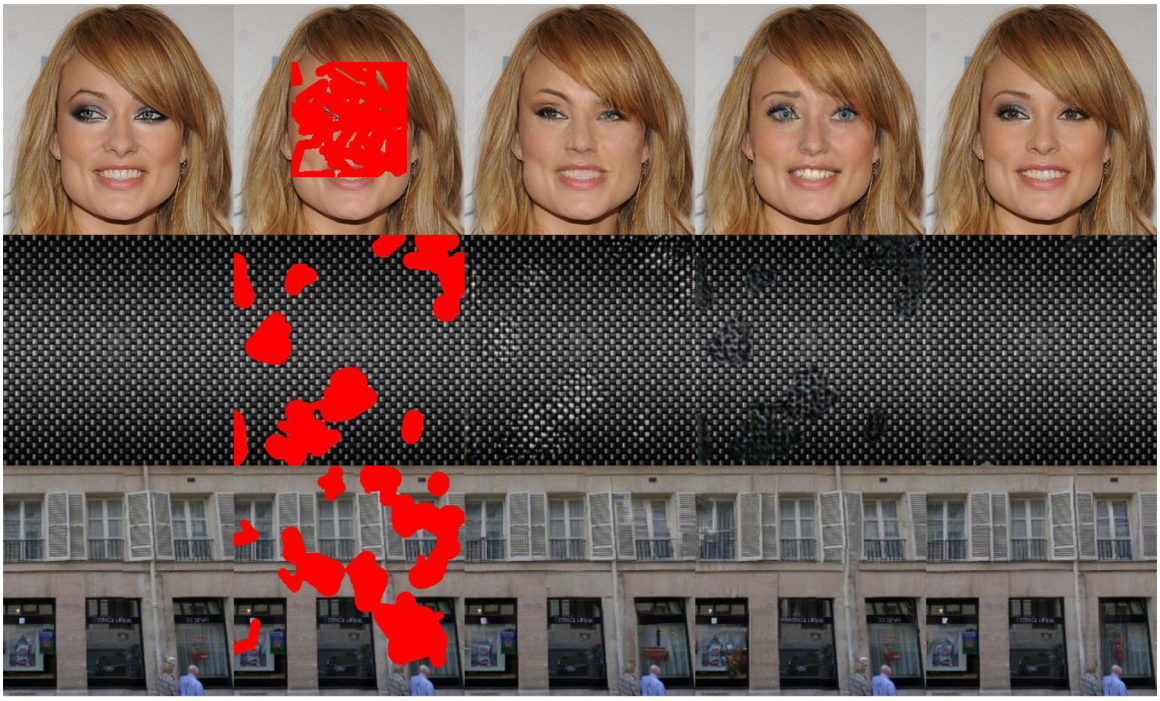

Fig. 6. Inpainted images with free-form masks. The original images are from CelebA-HQ [30] (Top), DTD [31] (Middle) and PSV [9] (Bottom), respectively.

Table 4
Quantitative results of ablation study on wavelet basis. The inpainted area is a central 128 × 128 square area. ↑/↓ means higher/lower is better.

Wavelet basis	PSNR↑	SSIM↑	MAE↓	FID↓	LPIPS↓	DISTS↓
db1(size:2 × 2)	26.00	**0.8399**	4.507	4.718	0.0747	0.0360
db2(size:4 × 4)	**26.17**	0.8390	**4.371**	5.168	**0.0735**	**0.0356**
db3(size:6 × 6)	26.10	0.8385	4.457	4.824	0.0745	0.0360
db4(size:8 × 8)	26.03	0.8386	4.517	4.811	0.0751	0.0360
framelet(size:3 × 3)	25.88	0.8358	4.595	**4.523**	0.0771	0.0365

Table 5
Quantitative results of ablation study on the structure of completion network. The inpainted area is a central 128 × 128 square area. ↑/↓ means higher/lower is better.

Completion network	PSNR↑	SSIM↑	MAE↓	FID↓	LPIPS↓	DISTS↓
G_{txt}	23.86	0.8084	5.825	6.637	0.1005	0.0472
G_{con}	25.94	0.8355	4.602	**4.198**	0.0738	0.0360
G_{txt} + G_{con}	**26.17**	**0.8390**	**4.371**	5.168	**0.0735**	**0.0356**

was changed to the combination of four wavelet subbands and the outputs were accordingly four inpainted wavelet subbands, while for G_{con} branch, the DWT was removed and a common U-net structure with two MLFM blocks was used in the network. The same parameter setting and model selection protocols were used as described in Section 4.1. The qualitative and quantitative comparisons are shown in Table 5 and Fig. 8, respectively. It can be seen that combining two branches in the completion network can achieve better inpainting quality than using only a single one.

4.3.3. Ablation study on the structure of upsampling block

To dig out a reasonable structure for the upsampling block, we have tried three types of upsampling block, i.e., *deconv* + *conv* (Fig. 9(a)), *bilinear* + *conv* (Fig. 9(b)), and the combination of them (the proposed one in Fig. 3). The visual results are shown in Fig. 10. It is observed that *deconv* + *conv* tends to produce better details but suffers from serious checkerboard artifacts and distorted color, while *bilinear* + *conv* can generate high-fidelity color but result in obscure details. Via combining the two types of upsampling, the proposed upsampling block takes their advantages and produces

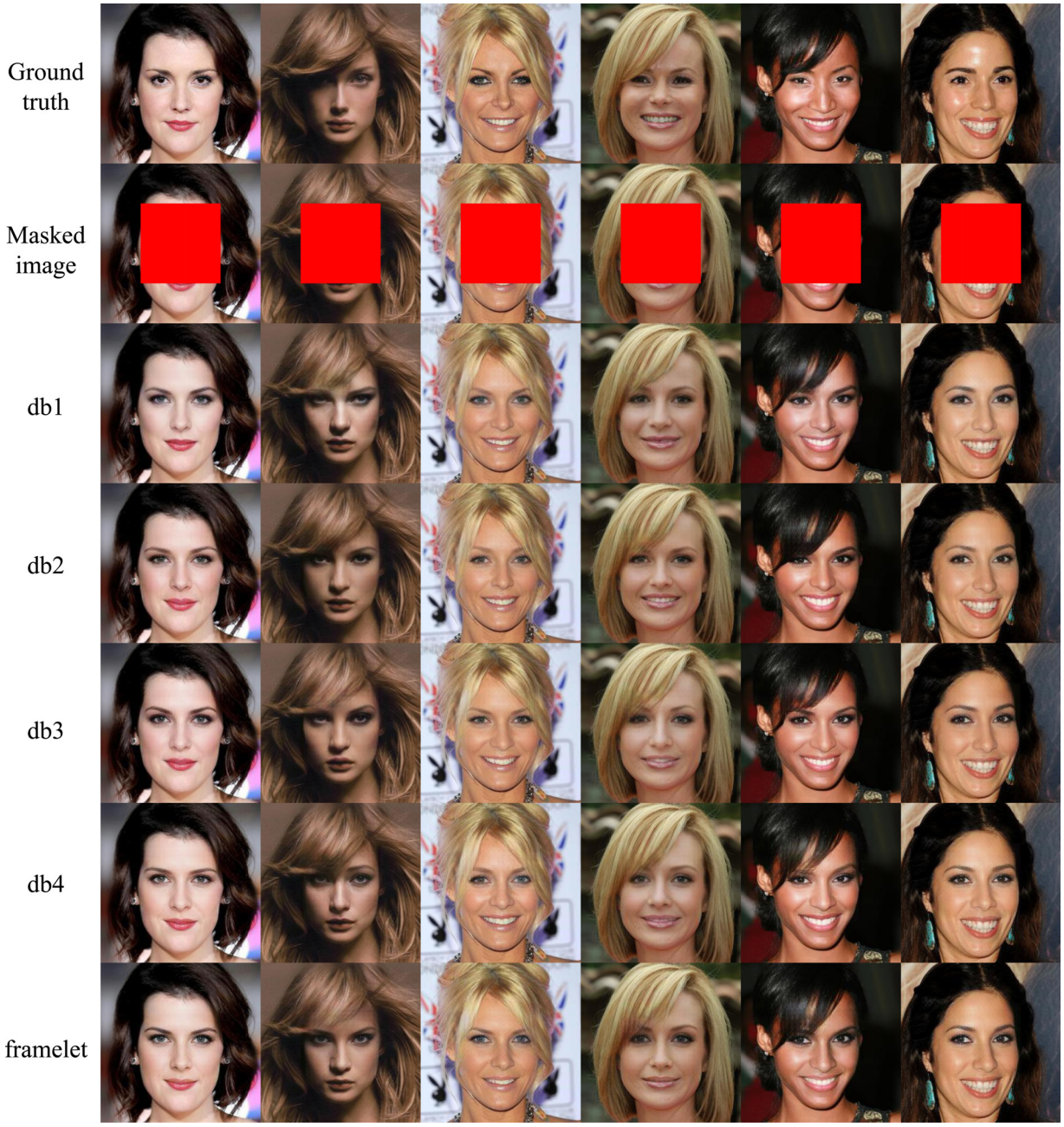

Fig. 7. Example results of Daubechies wavelet basis and framelet wavelet basis.

Table 6
Quantitative results of ablation study on the upsampling block. The inpainted area is a central 128 × 128 square area. ↑/↓ means higher/lower is better.

Upsampling module	PSNR↑	SSIM↑	MAE↓	FID↓	LPIPS↓	DISTS↓
deconv + conv	26.08	0.8371	4.486	**4.469**	0.0744	0.0358
bilinear + conv	25.92	0.8335	4.631	5.718	0.0807	0.0388
our upsampling block	**26.17**	**0.8390**	**4.371**	5.168	**0.0735**	**0.0356**

better details, which is also evidenced by the quantitative metrics in Table 6.

4.3.4. Ablation study on the SD-mask

To verify the benefit of using SD-mask, we conduct experiments by using loss functions with and without SD-masks. The results are shown in Table 7. It can be observed that the evaluation metrics of the loss functions with SD-masks are better than those of the loss functions without SD-masks.

4.4. Applications to object removal and inpainting noisy image

In fact, the trained model can be used for removing some undesired objects. We conducted an interactive image editing experiment for object removal. The results are shown in Fig. 11. As an

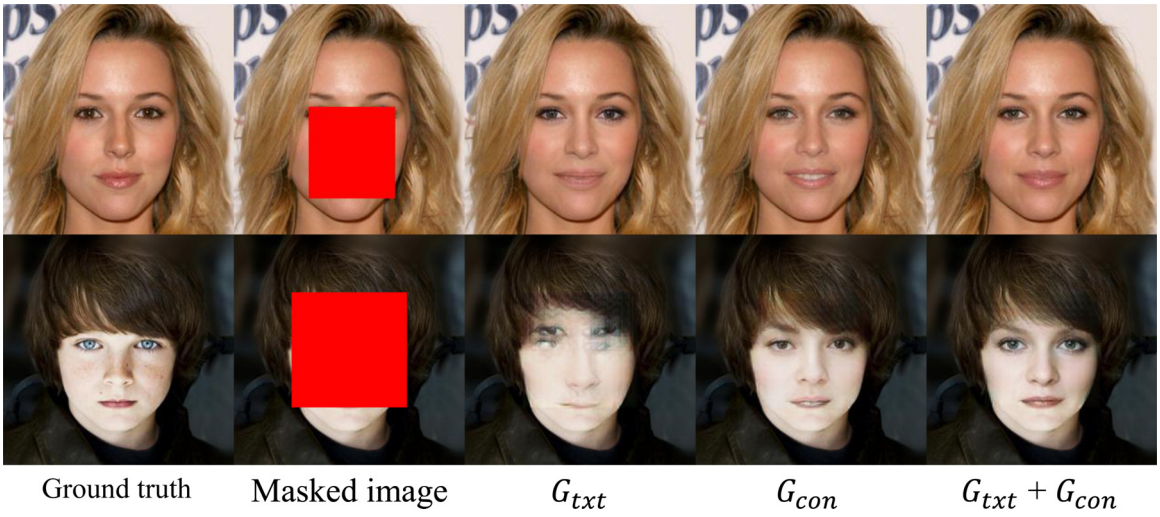

Fig. 8. Example results of different structures of completion network.

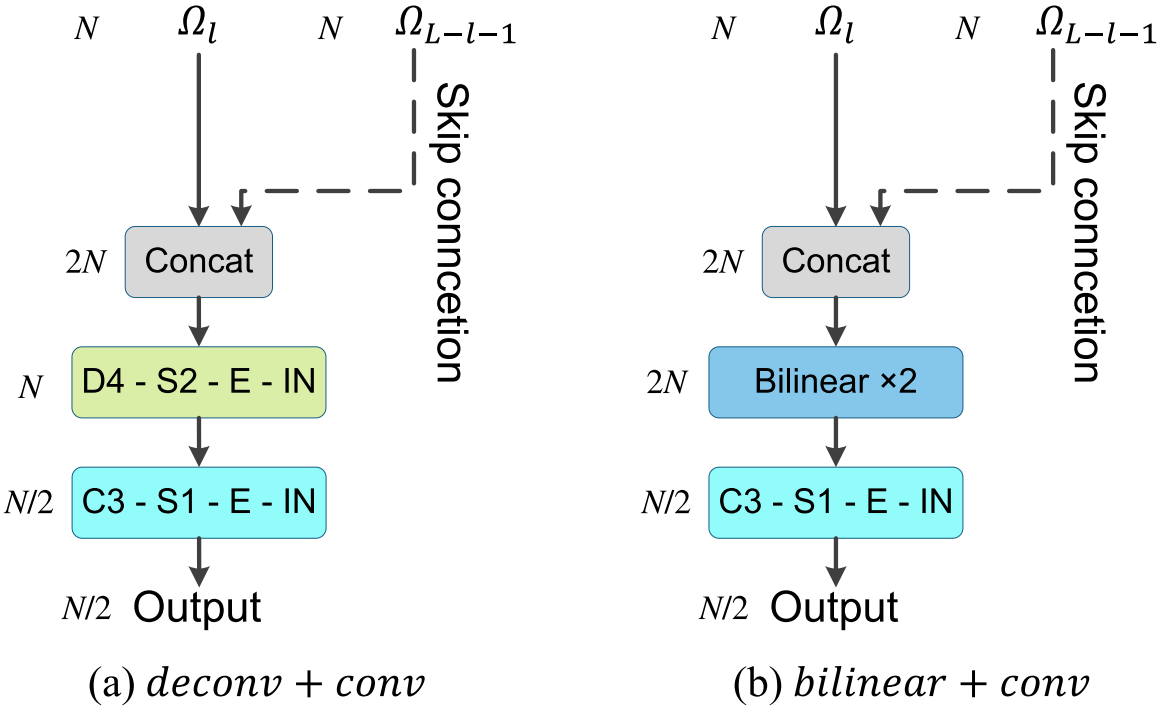

Fig. 9. Two different types of upsampling block. The meanings of notations are the same as those in Fig. 3.

Table 7
Quantitative results of ablation study on the SD-mask. The inpainted area is a central 128 × 128 square area. ↑/↓ means higher/lower is better.

	PSNR↑	SSIM↑	MAE↓	FID↓	LPIPS↓	DISTS↓
Without SD-mask	26.03	0.8374	4.496	6.179	0.0753	0.0364
With SD-mask	**26.17**	**0.8390**	**4.371**	**5.168**	**0.0735**	**0.0356**

example, in order to remove the girl's glasses, we created the missing mask for the regions of the glasses, and then fed the image into our trained model. The output was a girl without glasses. When some texts, visible watermarks, or scratches exist, we can remove them in a similar way, as shown in the figure.

In some cases, the image to be inpainted may be corrupted by noise. Since image inpainting is usually used for completing connected regions instead of scattered pixels, an inpainting model may fail to perform denoising. As shown in Fig. 12(a), when we directly used the trained model to an image contaminated by some salt&pepper noise, the outcome is unsatisfactory, even when there is a denoising operation afterwards. In contrast, we can perform denoising using some conventional or deep learning based methods at first to remove the noise in non-missing region, and then perform inpainting for the missing region. As shown in Fig. 12(b), we can obtain a satisfactory result.

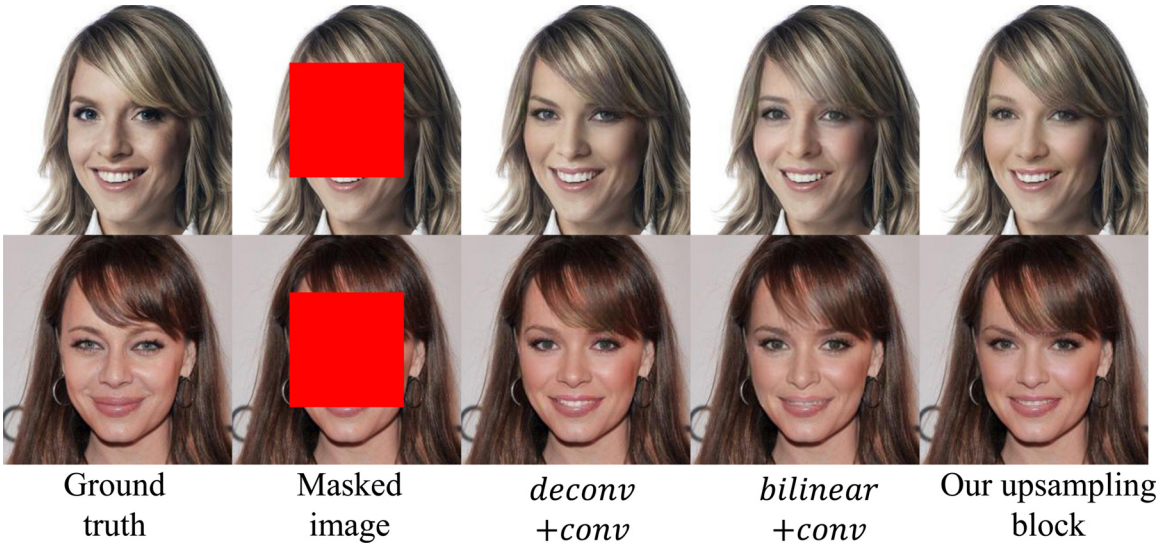

Fig. 10. Example results of different types of upsampling block.

Fig. 11. Example results of object removal and image editing.

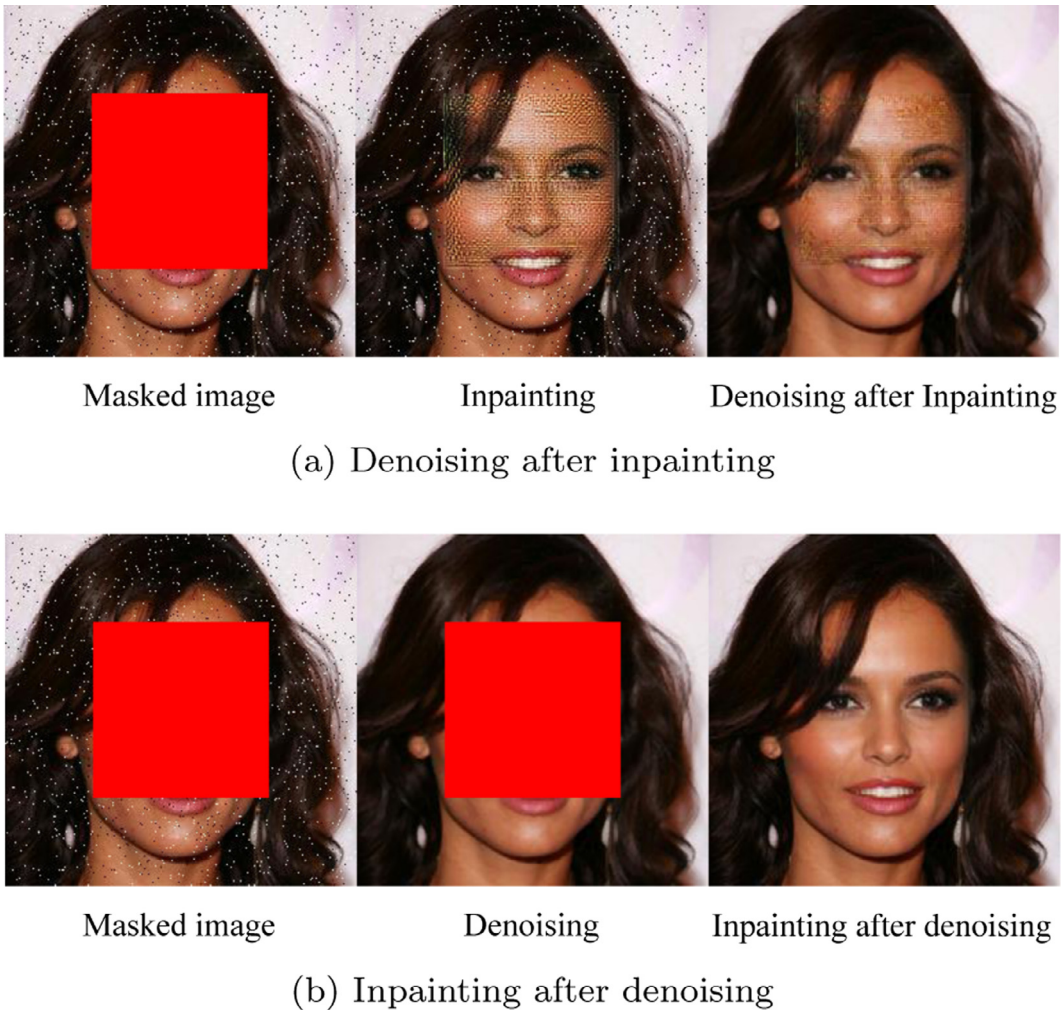

Fig. 12. Example results of (a) denoising after inpainting, and (b) inpainting after denoising.

5. Conclusion

In this paper, we proposed a novel detail-enhanced image inpainting method based on DWT. Specifically, given a damaged image, a content branch is used to fill semantic content in spatial domain and a texture branch is adopted to generate high-frequency details in wavelet domain. The output of both branches are combined to reconstruct an inpainted image via IDWT. To improve the capability in semantic understanding, a multi-level fusion module is designed to enlarge the receptive field of the content branch. In addition, we propose an effective way to generate free-form spatially discounted mask. As a result, our method can achieve more global reasonable structure and acceptable details compared to the state-of-the-art methods. We note that most existing methods can only handle images of a constant resolution. In the future work, we plan to investigate how to address the issue of simultaneously inpainting and denoising, and how to handle images of different resolutions with a unified model.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Bin Li: Conceptualization, Methodology, Formal analysis, Writing – original draft. **Bowei Zheng:** Software, Investigation, Data curation, Writing – original draft. **Haodong Li:** Conceptualization, Methodology, Validation, Writing – review & editing. **Yanran Li:** Resources, Writing – review & editing.

Acknowledgments

This work was supported in part by NSFC under Grants 61802262 and 61872244, Guangdong Basic and Applied Basic Research Foundation under Grant 2019B151502001, and Shenzhen R&D Program under Grants JCYJ20200109105008228 and JCYJ20180305124325555.

References

[1] M. Bertalmio, G. Sapiro, V. Caselles, C. Ballester, Image inpainting, in: Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques, 2000, pp. 417–424.
[2] L. He, Y. Wang, J. Liu, C. Wang, S. Gao, Single image restoration through $\ell 2$-relaxed truncated $\ell 0$ analysis-based sparse optimization in tight frames, Neurocomputing 443 (2021) 272–291.
[3] L. He, Y. Wang, S. Gao, A support-denoiser-driven framework for single image restoration, J. Comput. Appl. Math. 393 (2021) 113495.
[4] T.F. Chan, J. Shen, Nontexture inpainting by curvature-driven diffusions, J. Vis. Commun. Image Represent. 12 (4) (2001) 436–449.

[5] A. Levin, A. Zomet, Y. Weiss, Learning how to inpaint from global image statistics, in: Proceedings of the IEEE International Conference on Computer Vision, volume 1, 2003, pp. 305–312.
[6] A. Criminisi, P. Pérez, K. Toyama, Region filling and object removal by exemplar-based image inpainting, IEEE Trans. Image Process. 13 (9) (2004) 1200–1212.
[7] T. Ružić, A. Pižurica, Context-aware patch-based image inpainting using Markov random field modeling, IEEE Trans. Image Process. 24 (1) (2014) 444–456.
[8] C. Barnes, E. Shechtman, A. Finkelstein, D.B. Goldman, Patchmatch: a randomized correspondence algorithm for structural image editing, ACM Trans. Graph. 28 (3) (2009) 24.
[9] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: feature learning by inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2536–2544.
[10] S. Iizuka, E. Simo-Serra, H. Ishikawa, Globally and locally consistent image completion, ACM Trans. Graph. 36 (4) (2017) 1–14.
[11] G. Liu, F.A. Reda, K.J. Shih, T.-C. Wang, A. Tao, B. Catanzaro, Image inpainting for irregular holes using partial convolutions, in: Proceedings of the European Conference on Computer Vision, 2018, pp. 85–100.
[12] C. Zheng, T.-J. Cham, J. Cai, Pluralistic image completion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1438–1447.
[13] Y. Zeng, J. Fu, H. Chao, B. Guo, Learning pyramid-context encoder network for high-quality image inpainting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 1486–1494.
[14] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Generative image inpainting with contextual attention, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5505–5514.
[15] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Free-form image inpainting with gated convolution, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 4471–4480.
[16] G.E. Hinton, R.S. Zemel, Autoencoders, minimum description length and Helmholtz free energy, in: Proceedings of the Conference Neural Information Processing Systems, 1994, pp. 3–10.
[17] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, Commun. ACM 63 (11) (2020) 139–144.
[18] F. Li, T. Zeng, A universal variational framework for sparsity-based image inpainting, IEEE Trans. Image Process. 23 (10) (2014) 4242–4254.
[19] L. He, Y. Wang, Iterative support detection-based split Bregman method for wavelet frame-based image inpainting, IEEE Trans. Image Process. 23 (12) (2014) 5470–5485.
[20] Y.-R. Li, L. Shen, B.W. Suter, Adaptive inpainting algorithm based on DCT induced wavelet regularization, IEEE Trans. Image Process. 22 (2) (2012) 752–763.
[21] C. Wang, J. Wang, Q. Zhu, B. Yin, Generative image inpainting based on wavelet transform attention model, in: Proceedings of the IEEE International Symposium on Circuits and Systems, 2020, pp. 1–5.
[22] J. Wang, C. Wang, Q. Huang, Y. Shi, J.-F. Cai, Q. Zhu, B. Yin, Image inpainting based on multi-frequency probabilistic inference model, in: Proceedings of the ACM International Conference on Multimedia, 2020, pp. 1–9.
[23] D.-A. Clevert, T. Unterthiner, S. Hochreiter, Fast and accurate deep network learning by exponential linear units (ELUs), arXiv preprint arXiv:1511.07289
[24] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, 2015, pp. 234–241.
[25] L.A. Gatys, A.S. Ecker, M. Bethge, A neural algorithm of artistic style, arXiv preprint arXiv:1508.06576
[26] C. Doersch, S. Singh, A. Gupta, J. Sivic, A.A. Efros, What makes paris look like paris? Commun. ACM 58 (12) (2015) 103–110.
[27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A.C. Courville, Improved training of Wasserstein GANs, in: Proceedings of the Conference Neural Information Processing Systems, 2017, pp. 5767–5777.
[28] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556
[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255.
[30] T. Karras, T. Aila, S. Laine, J. Lehtinen, Progressive growing of GANs for improved quality, stability, and variation, arXiv preprint arXiv:1710.10196
[31] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, A. Vedaldi, Describing textures in the wild, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 3606–3613.
[32] R. Tyleček, R. Šára, Spatial pattern templates for recognition of objects with regular structure, in: Proceedings of the German Conference on Pattern Recognition, 2013, pp. 364–374.
[33] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, A. Torralba, Places: a 10 million image database for scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 40 (6) (2017) 1452–1464.
[34] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, arXiv preprint arXiv:1406.2661
[35] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.
[36] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, S. Hochreiter, Gans trained by a two time-scale update rule converge to a local Nash equilibrium, in: Proceedings of the Conference Neural Information Processing Systems, 2017, pp. 6626–6637.
[37] R. Zhang, P. Isola, A.A. Efros, E. Shechtman, O. Wang, The unreasonable effectiveness of deep features as a perceptual metric, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 586–595.
[38] K. Ding, K. Ma, S. Wang, E.P. Simoncelli, Comparison of image quality models for optimization of image processing systems, Int. J. Comput. Vis. 129 (4) (2021) 1258–1281.
[39] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980