Image inpainting using frequency-domain priors

Hiya Roy[a,*] **Subhajit Chaudhury**[b] **Toshihiko Yamasaki**[b] **and Tatsuaki Hashimoto**[a]

[a]The University of Tokyo, Department of Electrical Engineering and Information Systems, Tokyo, Japan

[b]The University of Tokyo, Department of Information and Communication Engineering, Tokyo, Japan

Abstract. We present an image inpainting technique using frequency-domain information. Prior works on image inpainting predict the missing pixels by training neural networks using only the spatial-domain information. However, these methods still struggle to reconstruct high-frequency details for real complex scenes, leading to a discrepancy in color, boundary artifacts, distorted patterns, and blurry textures. To alleviate these problems, we investigate if it is possible to obtain better performance by training the networks using frequency-domain information (discrete Fourier transform) along with the spatial-domain information. To this end, we propose a frequency-based deconvolution module that enables the network to learn the global context while selectively reconstructing the high-frequency components. We evaluate our proposed method on the publicly available datasets: celebFaces attribute (CelebA) dataset, Paris street-view, and describable textures dataset and show that our method outperforms current state-of-the-art image inpainting techniques both qualitatively and quantitatively. [DOI: 10.1117/1.JEI.30.2.023016]

Keywords: image inpainting; frequency-domain analysis; neural networks; machine learning; generative adversarial networks.

Paper 200631 received Sep. 11, 2020; accepted for publication Mar. 15, 2021; published online Apr. 6, 2021.

1 Introduction

In computer vision, the task of filling in missing pixels of an image is known as image inpainting. It can be extensively applied for creative editing tasks such as removing unwanted/distracting objects in an image, generating the missing region of an occluded image, or improving data availability for satellite imagery. The main challenge in this task is to synthesize the missing pixels in such a way that it looks visually realistic and coherent to human eyes.

Traditional image inpainting algorithms[1–11] can be broadly divided into two categories. Diffusion-based image inpainting algorithms[1–4] focus on propagating the local image appearance into the missing regions. Although these methods can fill in small holes, they produce smoothed results as the hole grows bigger. On the other hand, patch-based traditional inpainting algorithms[5–11] iteratively search for the best-fitting patch in the image to fill in the missing region. These methods can fill in bigger holes, but they are not effective either in inpainting missing regions that have complex structures or in generating unique patterns or novel objects that are not available in the image in the form of a patch.

Recent research on image inpainting[12–17] has leveraged the advancements in generative models such as generative adversarial networks (GANs)[18] and have shown that it is possible to learn and predict missing pixels in coherence with the existing neighboring pixels by training a convolutional encoder–decoder network. In this paradigm, generally speaking, the model is trained in a two-stage manner: (i) in the first stage, the missing regions are coarsely filled in with initial structures by minimizing the traditional reconstruction loss and (ii) in the second stage, the initially reconstructed regions are refined using an adversarial loss. Although these methods

*Address all correspondence to Hiya Roy, hiya@hal.t.u-tokyo.ac.jp

are good at generating visually plausible novel contents such as human faces, structures, and natural scenes in the missing region, they still struggle to reconstruct high-frequency details for real complex scenes, leading to a discrepancy in color, boundary artifacts, distorted patterns, and blurry textures. Additionally, the reconstruction quality of previous methods deteriorates as the size of the missing region increases. The above problems can be attributed to existing methods using only spatial-domain information during the learning process, similar to diffusion techniques, to obtain information from the mask boundary. Thus as the mask size increases, the interior reconstruction details are lost, and only a low-frequency component of the original patch is estimated by these methods.

To alleviate the above problem, we resort to frequency-based image inpainting. We show that image inpainting can be converted to the problem of deconvolution in the frequency domain, which can predict local structure in the missing regions using global context from the image. Qualitative analysis shows that our proposed frequency-domain image inpainting approach helps improve the texture details of missing regions. Previous methods make use of only spatial-domain information. Therefore, the reconstruction of the information close to the mask boundary is good compared with that of the interior region since the local context is available only in the boundary regions. On the other hand, a frequency-based approach would take information from the global context in the image because to discrete Fourier transforms (DFT) that considers all pixels for computing the frequency components. As a result, it captures more detailed structural and textural content of the missing regions in the learned representation. Due to these reasons, we propose a two-stage network consisting of (i) a deconvolution stage and (ii) a refinement stage. In the first stage, the DFT image from the original RGB image is computed. Each frequency component in the DFT image captures the global context, thus forming a better representation of the global structure. We employ a convolutional neural network (CNN) to learn the mapping between the masked DFT and the original DFT, which is a deconvolution operation obtained by minimizing the l_2 loss. Although DFT-based deconvolution can reconstruct the global structural outline, we observe that there exists a mismatch in the color space of the first-stage output. Therefore, in the second stage, we fine-tune the output of the first stage using adversarial methods to match the color distribution of the true image. Figure 1 shows an example of the reconstructed output using our method; Fig. 1(b) shows the DFT map of our first-stage reconstruction obtained from the deconvolution network. This additional frequency-domain information is later used by the refinement network to obtain the final output as shown in Fig. 1(c). Our main contributions in this paper are summarized as follows.

1. We introduce a new frequency-domain-based image inpainting framework that learns the high-frequency component of the masked region using the global context of the image. We find that the network learns to preserve image information in a better way when it is trained in the frequency domain. Therefore, adding the frequency-domain and spatial-domain information certainly improves the inpainting performance compared with the conventional spatial-domain image inpainting algorithms. To enable better inpainting, we train the network using both frequency-domain and spatial-domain information, which leads to a better consistency of inpainted results in terms of the local and global contexts.

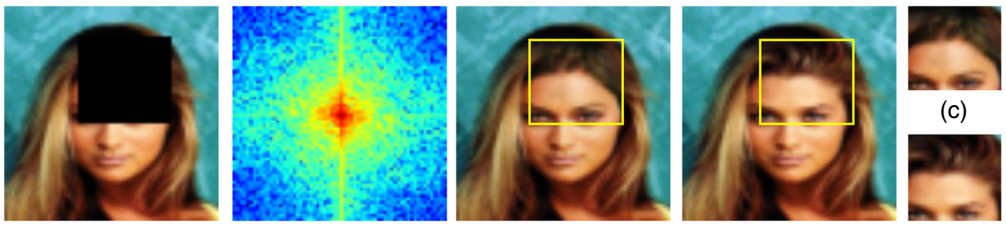

Fig. 1 (a) Input images with missing regions; (b) DFT of first-stage reconstruction by our deconvolution network; (c) image inpainting results (after the second stage) of our proposed approach; and (d) GT image. The last column shows the prediction of the missing region obtained from our method and original pixel values for the same region in the GT image.

2. We validate our method on benchmark datasets including celebFaces attributes (CelebA), Paris streetview (PSV), and describable textures dataset (DTD) and show that our method achieves better inpainting results in terms of visual quality and evaluation metrics, outperforming the state-of-the-art results. To the best of our knowledge, this is the first work that explores the benefits of using frequency-domain information for image inpainting.

2 Related Work

2.1 *Traditional Inpainting Techniques*

Diffusion-based image completion methods[1–4] are based on partial differential equations (PDE) in which a diffusive process is modeled using PDE to propagate colors into the missing regions. These methods work well for inpainting small missing regions but fail to reconstruct the structural component or texture for larger missing regions.

Patch-based algorithms, on the other hand, are based on iteratively searching for similar patches in the existing image and pasting/stitching the most similar block onto the image. Efros and Freeman[5] first proposed a patch-based algorithm for texture synthesis based on this philosophy. These algorithms perform well on textured images by assuming that the texture of the missing region is similar to the rest of the image. However, they often fail in inpainting missing regions in natural images because the patterns are locally unique in such images. Moreover, these methods are computationally expensive because of the need for computing similarity scores for every target–source pair. For more accurate and faster image inpainting, several optimal patch search-based methods were proposed by Drori et al.[6] (fragment-based image completion algorithm) and Criminisi et al.[7] (patch-based greedy sampling algorithm). Another optimization method to synthesize visual data (images or video) based on bidirectional similarity measure was proposed by Simakov et al.[8] Afterward, these techniques were expedited by Barnes et al.[9] who proposed patchMatch (PM), a fast randomized patch search algorithm that could handle the high computational and memory cost. Later, such patch-based image completion techniques were improved by Darabi et al.[10] by incorporating gradient-domain image blending, He et al.[11] by computing the statistics of patch offsets, and Ogawa and Haseyama[19] by optimizing sparse representations with respect to structural similarity index (SSIM) perceptual metric. However, these methods rely only on existing image patches and use low-level image features. Therefore, they are not effective in filling complex structures by performing semantically aware patch selections.

2.2 *Deep Learning-Based Inpainting*

Recently, CNN models[20] have shown tremendous success in solving high-level tasks such as classification, object detection, and segmentation, as well as low-level tasks such as image inpainting problem. Xie et al.[21] proposed stacked sparse denoising autoencoders that combine sparse coding and deep networks pretrained with a denoising autoencoder to solve a blind image inpainting task. Blind image inpainting is harder because, in this case, the missing pixel locations are not available to the algorithm and it has to learn to find the missing pixel location and then restore them. Köhler et al.[22] showed a mask specific deep neural network-based blind inpainting technique for filling in small missing regions in an image. Chaudhury et al.[23] attempted to solve this problem by proposing a lightweight fully convolutional network and demonstrated that their method can achieve comparable performance as the sparse coding-based K-singular value decomposition[24] technique. However, these inpainting approaches were limited to very small sized masks.

More recently, adversarial learning-based inpainting algorithms have shown promising results in solving image inpainting problems because of their ability to learn and synthesize novel and visually plausible contents for different images such as objects,[12] scene completion,[13] and faces.[25] A seminal work by Pathak et al.[12] showed that their proposed context encoder (CE) network can predict missing pixels of an image based on the context of the surrounding areas of that region. They used both the standard l_2 loss and adversarial loss[18] to train their network. Later, Iizuka et al.[13] demonstrated that their encoder–decoder model could reconstruct pixels

in the missing region that are consistent both locally and globally by leveraging the benefits of dilated convolution layers, a variant of standard convolutional layers. Similar to Ref. 12, this approach also uses adversarial learning for image completion, but unlike it,[12] it can handle arbitrary images and mask sizes because of the proposed global and local context discriminator networks. Recently, Yu et al.[14] introduced the concept of attention for solving an image inpainting task by proposing a novel contextual attention (CA) layer and trained the unified feedforward generative network with the reconstruction loss and two Wasserstein GAN losses.[26,27] They showed that their method can inpaint images with multiple missing regions having different sizes and located arbitrarily in the image. Later, Liu et al.[28] proposed a partial convolution layer with an automatic mask-update rule that can handle free-form/irregular masks. Here the mask is updated in such a way that the missing pixels are predicted based on the real pixel values of the original image where the partial convolution can operate. Song et al.[15] showed that it is possible to perform image inpainting using segmentation information. To this end, they proposed a model that predicts the segmentation labels of the corrupted image at first and then fills in the segmentation mask so that it can be used as a guide to complete the image. Nazeri et al.[16] introduced an edge generator that at first predicts the edges of the missing regions and then uses the predicted edges as a guidance to the complete the image. Yu et al.[17] proposed a gated convolution-based approach to handle free-form image completion.

2.3 *Frequency-Domain Learning*

Recently, enabling the network to learn information in the frequency domain has gained popularity because the frequency-domain information contains rich representations that allow the network to perform the image understanding tasks in a better way than the conventional way of using only spatial-domain information. Gueguen et al.[29] proposed image classification using features from the frequency domain. Xu et al.[30] showed that it is possible to perform object detection and instance segmentation by learning information in the frequency domain with a slight modification to the existing CNN models that use RGB input. In this paper, we propose using frequency-domain information along with spatial-domain information to achieve better image inpainting performance.

3 Proposed Method

Given a corrupted input image, our aim is to predict the missing region in such a way that it looks similar to the clean images as viewed by human eyes. In this paper, we propose a frequency-based non-blind image inpainting framework that consists of two stages: (i) a frequency-domain deconvolution network and (ii) a refinement network. The overall framework of the proposed method is shown in Fig. 2. In the first stage, we compute the DFT of the masked image (both magnitude and phase information) and the original RGB image and train a CNN for deconvolution to learn the mapping between the two signals by minimizing the l_2 loss. Here we formalize the problem of inpainting in the spatial domain as deconvolution in the frequency domain. We employ the feed-forward denoising convolutional neural networks,[31] a manifestation of deconvolution that

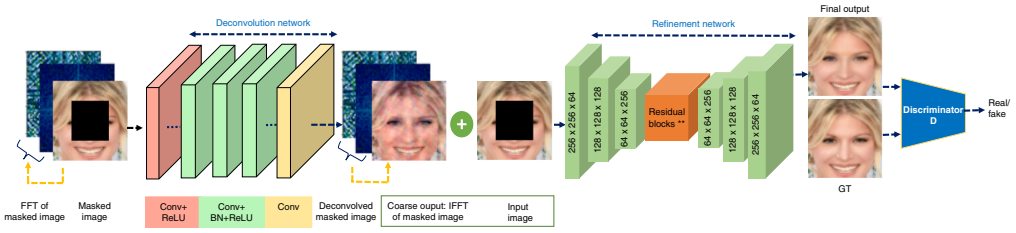

Fig. 2 Overview of our frequency-domain-based image inpainting framework. The deconvolution network is trained in the frequency domain with l_2 loss to learn the mapping between DFT of the masked image and the original image. The refinement network is trained in the spatial domain with the adversarial loss.

uses residual learning to predict the denoised image. The motivation behind this DFT-based deconvolution operation is to learn a better representation of the global structure that can serve as guidance to the second network. In the second stage, we use the spatial-domain information (of the masked image and the mask) and train a GAN-based model[18] by minimizing an adversarial loss along with the l_2 loss. The motivation to incorporate this stage is to fine-tune the output of the first stage by refining the structural details and matching the color distribution of the true image in a local scale. The various components of our model are explained in the following sections.

3.1 *Frequency-Domain Deconvolution Network*

3.1.1 *Problem formulation*

Let us consider $\mathbf{I}_{\text{in}}$ the corrupted/incomplete input image, $\mathbf{I}_{\text{gt}}$ the ground truth (GT) image, and $\mathbf{I}^1_{\text{pred}}$ the predicted output image after the first stage. At first, we calculate the DFT of $\mathbf{I}_{\text{in}}$ and $\mathbf{I}_{\text{gt}}$ as $\mathbf{I}^f_{\text{in}} = \text{DFT}(\mathbf{I}_{\text{in}})$ and $\mathbf{I}^f_{\text{gt}} = \text{DFT}(\mathbf{I}_{\text{gt}})$. Let us consider a mask function in spatial domain $\mathbf{M}$, with its frequency-domain counterpart $\mathbf{M}^f$.

A masked image is represented as $\mathbf{I}_{\text{in}}(x, y) = \mathbf{I}_{\text{gt}}(x, y) \odot \mathbf{M}(x, y)$, where $\odot$ denotes element-wise multiplication. Our contribution in this paper is to analyze this relation between the frequency-domain signals of $\mathbf{I}_{\text{in}}$, $\mathbf{I}_{\text{gt}}$, and $\mathbf{M}$. For example, if we consider a mask of size $(2W, 2H)$, the power spectral density for the DFT of mask signal is given as

$$|\mathbf{M}^f(p, q)|^2 \propto \frac{\sin(\pi p)}{\sin\left(\frac{\pi p}{N}\right)} \frac{\sin(\pi q)}{\sin\left(\frac{\pi q}{N}\right)}, \tag{1}$$

where $k = 0,1, \ldots (N-1)$ represents the discrete frequency, with N being the number of samples. The frequency-domain representation of the mask signal is shown in Fig. 3, which depicts a decaying pulse from the origin. By the convolution–multiplication property of DFT, we show that the multiplication of mask with the image in spatial domain is equivalent to the convolution of the mask with the image in the frequency domain (Fig. 3). Mathematically, this is represented as

$$\mathbf{I}^f_{\text{in}}(p, q) = \mathbf{I}^f_{\text{gt}}(p, q) \circledast \mathbf{M}^f(p, q), \tag{2}$$

where $\circledast$ denotes the convolution operation and the masked frequency signal is the output of the convolution of the mask and clean image DFT signal. Therefore, we perform a deconvolution

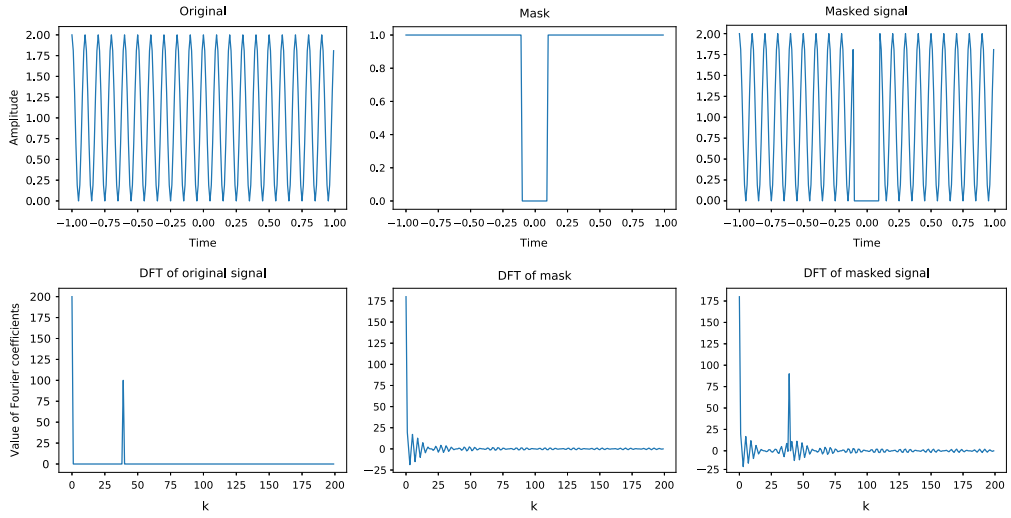

Fig. 3 Visualization of the masked signal in the frequency domain (using DFT). Here we use the convolution–multiplication property of DFT to transform signals from the spatial to frequency domains and vice versa.

operation to predict the missing region of the incomplete image. Let $\mathbf{F}(\mathbf{I}_{\text{in}};\theta)$ be the deconvolutional neural network that converts $\mathbf{I}_{\text{in}}$ to $\mathbf{I}^{1}_{\text{pred}}$, such that $\mathbf{I}^{1}_{\text{pred}} = \mathbf{F}(\mathbf{I}_{\text{in}};\theta)$. After calculating $\mathbf{I}^{f}_{\text{in}}$ and $\mathbf{I}^{f}_{\text{gt}}$, we train the network to learn the mapping between them to predict the first-stage output. We denote frequency-domain representation as $\mathbf{I}^{1f}_{\text{pred}}$, where $\mathbf{I}^{1f}_{\text{pred}} = \mathbf{F}(\mathbf{I}^{f}_{\text{in}};\theta)$. Next, we perform an inverse DFT of the first-stage output and get the predicted output image $\mathbf{I}^{1}_{\text{pred}} = \text{IDFT}(\mathbf{I}^{1f}_{\text{pred}})$.

It should be noted that, while calculating the DFT of the image, mask, and masked image, we compute both the magnitude and phase for all channels, normalize them to bring within the range of 0 to 1, and concatenate the normalized magnitude and phase information for all channels for training purposes.

3.1.2 *Network architecture*

To perform the deconvolution operation in the frequency domain, we adopt a CNN model having 17 layers similar to Zhang et al.[31]This deconvolution network contains three types of layers as shown in Fig. 2. The first layer is a conv layer with ReLU non-linearity, where 64 filters of $(3 \times 3 \times 3)$ size are used. The next layers (2nd to 16th) are a combination of the conv layer, a batch normalization layer,[32] and a ReLU layer, where 64 filters of $(3 \times 3 \times 64)$ size are used. The last layer is a conv layer, where three filters of $(3 \times 3 \times 64)$ size are used to reconstruct the output. Details of our first-stage deconvolution network are given in Table 1.

3.1.3 *Training*

To train our deconvolution network, we use the l_2 loss that minimizes the distance between the DFT of GT image $\mathbf{I}^{f}_{\text{gt}}$ and the DFT of inpainted image $\mathbf{I}^{1f}_{\text{pred}}$, which is given by

$$\mathcal{L}_{s1} = \|\mathbf{I}^{f}_{\text{gt}} - \mathbf{I}^{1f}_{\text{pred}}\|^{2}_{2}. \tag{3}$$

After training the first-stage deconvolution network, we compute the inverse DFT of $\mathbf{I}^{1f}_{\text{pred}}$ which is used as a guidance to train the refinement stage as shown in Fig. 2. The reason for choosing the frequency domain in the first network is that it contains rich information[30,33] for high-frequency preservation.

3.2 *Refinement Network*

The refinement network is a GAN-based model[18] that has shown promising results in generative modeling of images[34] in recent years. Our refinement network has a generator and a discriminator network, where the generator network takes the output of the first stage (frequency-domain deconvolution module), the original masked image, and the corresponding binary mask (spatial-domain information) as input pairs and outputs the generated image. The discriminator network takes this generator output and minimizes the Jensen–Shannon divergence between the input and output data distribution to match the color distribution and structural details of the output image to the true image.

Table 1 First-stage network architecture (deconvolution network).

Layer name	Layer no.	Stride, padding	Activation	Layer output size
Input	—	—	—	$1 \times 12 \times 64 \times 64$
Conv 3×3	1	1, 1	ReLU	
Conv 3×3	2 to 16 (15 layers)	1, 1	(Batch norm + ReLU)	
Conv 3×3	17	—	—	$1 \times 6 \times 64 \times 64$

3.2.1 *Network architecture*

Generator. We adapt the generator architecture from Johnson et al.[35] that has exhibited good performance for an image-to-image translation task.[36] Our generator network is an encoder–decoder architecture having three convolution layers for downsampling, eight residual blocks,[37] and three convolution layers for up-sampling. Here the conv-2 and conv-3 layers are stride-2 convolution layers that are responsible for down-sampling twice, and the conv-4 and conv-5 layers are transpose convolution layers that are responsible for up-sampling twice back to the original image size. We use instance normalization[38] and ReLU activation function across all layers of the generator network.

Discriminator. We adapt the discriminator network from several works;[36,39] it is a Markovian discriminator similar to 70×70 PatchGAN.[39] The advantage of using a PatchGAN discriminator is that it has fewer parameters compared with a standard discriminator because it works only on a particular image patch instead of an entire image. Furthermore, it can be applied to any arbitrarily sized images in a fully convolutional fashion.[36,39] We apply the sigmoid function after the last convolution layer, which produces a one-dimensional output score that predicts whether the 70×70 overlapping image patches are real or fake. To stabilize the discriminator network training, we use spectral normalization[40] as our weight normalization method. Moreover, we use leaky ReLUs[41] with a slope of 0.2. The details of our second-stage refinement network (generator and discriminator network) and the output size of each layer are given in Table 2.

Table 2 Second-stage network architecture.

Layer name	Stride	Activation	Layer output size
Generator network			
Input	—	—	$1 \times 9 \times 64 \times 64$
Encoder network			
Conv 7×7	1	ReLU	$1 \times 64 \times 64 \times 64$
Conv 4×4	2	ReLU	$1 \times 128 \times 32 \times 32$
Conv 4×4	2	ReLU	$1 \times 256 \times 16 \times 16$
Residual block ($\times 8$)			
Residual blocks		$1 \times 256 \times 16 \times 16$	
Decoder network			
Conv 4×4	2	ReLU	$1 \times 128 \times 32 \times 32$
Conv 4×4	2	ReLU	$1 \times 64 \times 64 \times 64$
Conv 7×7	1	Tanh	$1 \times 3 \times 64 \times 64$
Discriminator network			
Input	—	—	$1 \times 3 \times 64 \times 64$
Conv 4×4	2	LeakyReLU	$1 \times 64 \times 32 \times 32$
Conv 4×4	2	LeakyReLU	$1 \times 128 \times 16 \times 16$
Conv 4×4	2	LeakyReLU	$1 \times 256 \times 8 \times 8$
Conv 4×4	1	LeakyReLU	$1 \times 512 \times 7 \times 7$
Conv 4×4	1	Sigmoid	$1 \times 1 \times 6 \times 6$

Algorithm 1 Training the refinement network.

1: **while** Generator G has not converged **do**

2: Sample batch images $\mathbf{I}_{\text{in}}$ from training data;

3: Generate random masks $\mathbf{M}$;

4: Construct combined input ($\mathbf{I}_{\text{in}}$, $\mathbf{M}$, and $\mathbf{I}^1_{\text{pred}}$);

5: Get masked region prediction $\mathbf{I}^2_{\text{pred}} = G(\mathbf{I}_{\text{in}}, \mathbf{M}, \mathbf{I}^1_{\text{pred}})$;

6: Generate inpainted image by modifying the masked region $\mathbf{I}_{\text{pred}} \leftarrow \mathbf{I}_{\text{in}} + \mathbf{I}^2_{\text{pred}} \odot (\mathbf{1} - \mathbf{M})$;

7: Update G with l_1 loss and adversarial critic loss;

8: Update discriminator critic D with $\mathbf{I}_{\text{in}}$, $\mathbf{I}_{\text{pred}}$;

9: **end while**

3.2.2 *Training*

After obtaining the first-stage output, we feed it along with the spatial-domain information (of the masked image and the mask) to the refinement network. While training, the generator of the inpainting network G takes a combination of input image $\mathbf{I}_{\text{in}}$, image mask $\mathbf{M}$, and the first-stage output image $\mathbf{I}^1_{\text{pred}}$ and generates $\mathbf{I}^2_{\text{pred}} = G(\mathbf{I}_{\text{in}}, \mathbf{M}, \mathbf{I}^1_{\text{pred}})$ as output. Then by adding $\mathbf{I}^2_{\text{pred}}$ to the input image, we get the completed image as $\mathbf{I}_{\text{pred}} = \mathbf{I}_{\text{in}} + [\mathbf{I}^2_{\text{pred}} \odot (\mathbf{1} - \mathbf{M})]$. The training procedure of the refinement stage is described in Algorithm 1. We train our refinement module using two loss functions: a reconstruction loss and an adversarial loss.[18] Here for the reconstruction loss, we use the l_1 loss[12] that minimizes the distance between the clean/GT image $\mathbf{I}_{\text{gt}}$ and the completed/inpainted image $\mathbf{I}_{\text{pred}}$, which is given by

$$\mathcal{L}_{\ell_1}(x) = \|\mathbf{I}_{\text{gt}} - \mathbf{I}_{\text{pred}}\|_1, \tag{4}$$

where $\mathbf{I}_{\text{pred}} \leftarrow \mathbf{I}_{\text{in}} + G(\mathbf{I}_{\text{in}}, \mathbf{M}, \mathbf{I}^1_{\text{pred}}) \odot (\mathbf{1} - \mathbf{M})$. For the adversarial loss, we follow the min–max optimization strategy, where the generator G is trained to produce inpainted samples from the artificially corrupted images such that the inpainted samples appear as "real" as possible and the adversarially trained discriminator critic D tries to distinguish between the GT clean samples and the generator predictions/inpainted samples. The objective function is expressed as follows:

$$G^*, D^* = \arg \min_G \max_D \mathcal{L}_{\text{adv}}(G, D) = \mathbb{E}_{\mathbf{x} \sim \mathbb{P}_r}[\log\, D(x)] + \mathbb{E}_{\tilde{\mathbf{x}} \sim \mathbb{P}_g}[\log(1 - D(\tilde{\mathbf{x}})],$$

where $\mathbb{P}_r$ is the real/GT data distribution and $\mathbb{P}_g$ is the model/generated data distribution defined by $\tilde{\mathbf{x}} = G(\mathbf{I}_{\text{in}}, \mathbf{M}, \mathbf{I}^1_{\text{pred}})$. Thus our overall loss function for the refinement stage becomes

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\ell_1} + \lambda_2 \mathcal{L}_{\text{adv}}, \tag{5}$$

where $\lambda_1 = 1, \lambda_2 = 0.1$. The weighted sum of these two loss functions compliments each other in the following ways: (i) the GAN loss helps to improve the realism of the inpainted images by fooling the discriminator. (ii) The ℓ_1 reconstruction loss serves as a regularization term for training GANs,[14] helps in stabilizing GAN training, and encourages the generator to generate images from the modes that are close to the GT in an l_1 sense.

4 Implementation Details

Our proposed model is implemented in PyTorch and our code is available in GitHub (https://github.com/hiyaroy12/DFT_inpainting). In our experiments, we resize the image to 64×64 and linearly scale the pixel values from the range [0, 256] to $[-1,1]$. For the first stage, we initialize

the weights using He initialization[42] and use the SGD optimizer with a weight decay of 0.0001, momentum of 0.9, and mini-batch size of 128. To train the first-stage network, we decay the learning rate exponentially from 10^{-1} to 10^{-4} for 50 epochs. For the second stage, both our generator G and discriminator D are trained together using the following settings: (i) G and D learning rates of 10^{-4} and 10^{-5}, respectively, and (ii) optimized using the Adam optimizer[43] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. In our experiments, we use a batch size of 14 and the training iterations of 100. Both stages are implemented on a TITAN Xp (12 GB) GPU.

5 Experiments

In this section, we evaluate the inpainting performance of our proposed method on three standard datasets: CelebA,[44] PSV,[45] and DTD.[46] For our experiments, we use both regular and irregular masks. Regular masks refer to square masks having a fixed size consisting of 25% of total image pixels and being randomly located in the image. For irregular masks, during training, we use the masks from the work of Liu et al.,[28] where the irregular mask dataset contains the augmented versions of each mask (0 deg, 90 deg, 180 deg, and 270 deg rotated, horizontally reflected) and are divided based on the percentage of mask size on the image in increments of 10% such as 0% to 10%, 10% to 20%, etc.

5.1 *Qualitative Evaluation*

Figures 4 and 5 compare the inpainting results of our method with previous image inpainting methods: PM,[9] CE,[12] CA,[14] and generative inpainting (GI),[17] for regular masks on CelebA and PSV datasets. The last six columns of these figures demonstrate the magnitude spectrum of the DFT map obtained from different methods,[9,12,14,17] our method (first-stage reconstruction), and

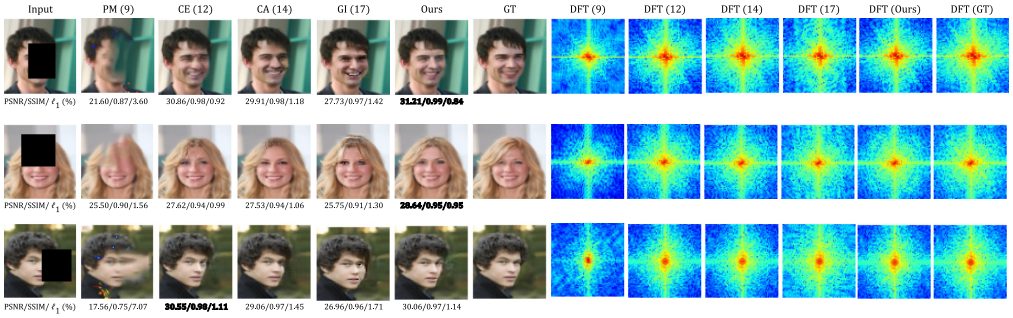

Fig. 4 Visual comparison of semantic feature completion results for different methods on the CelebA dataset along with the DFT maps corresponding to different methods, our first-stage output, and the GT image.

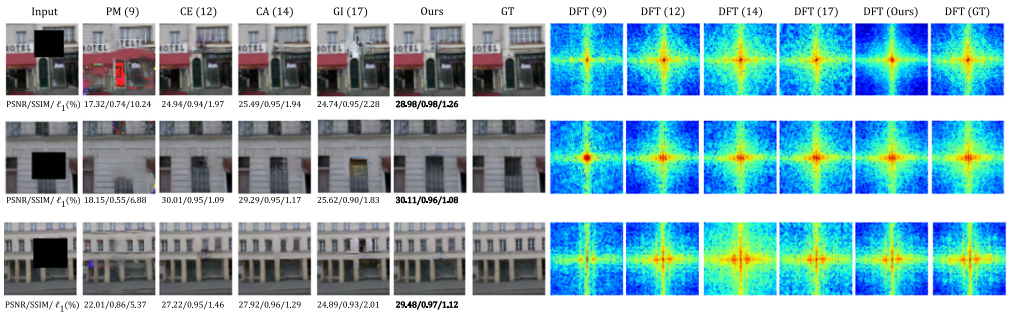

Fig. 5 Visual comparison of semantic feature completion results for different methods on the PSV dataset along with the DFT maps corresponding to different methods, our first-stage output, and the GT image.

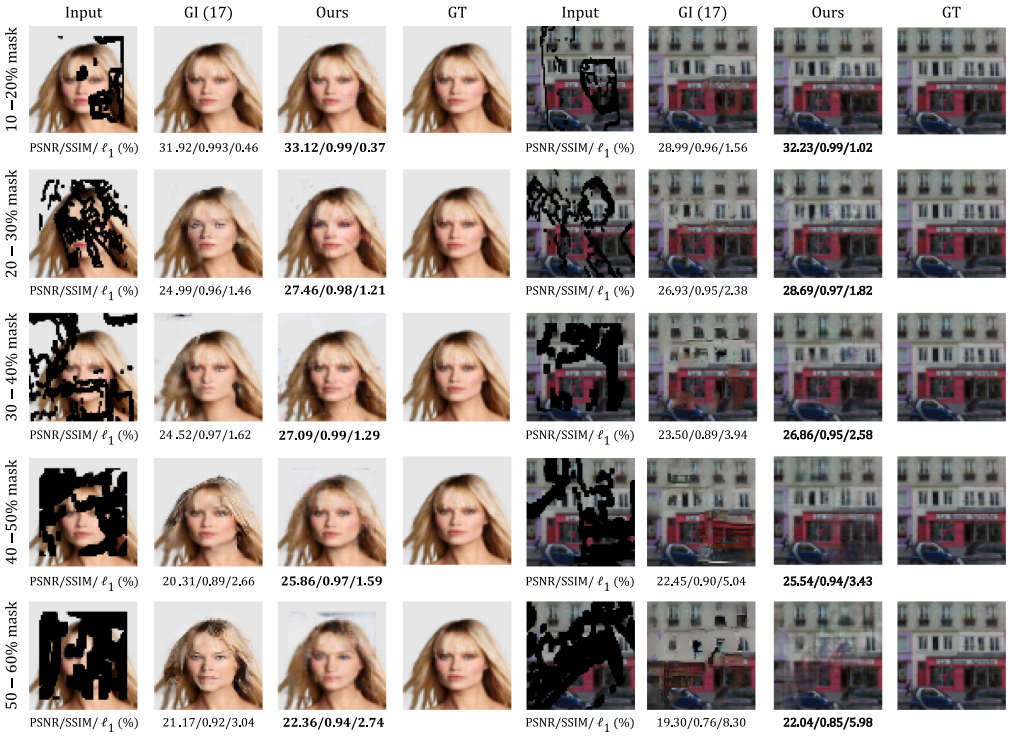

Fig. 6 Visual comparison of semantic feature completion results for irregular masks on the CelebA and PSV datasets.

the GT image. We can see that previous methods (PM) copy incorrect patches in the missing regions, whereas others (CE, CA, and GI) sometimes fail to achieve plausible results and generate distinct artifacts. However, our method can restore the missing regions with sharp structural details, minimal blurriness, and hardly any "checkerboard" artifacts. Moreover, the inpainting results using our method look the most similar to the GT images. We conjecture that in the presence of frequency-domain information, the network efficiently learns the high-frequency details, which enables it to preserve the structural details in the restored image. This can be confirmed from the DFT maps in which we see that our deconvolution network learns to predict the missing region in such a way that the DFT map of our first-stage reconstruction looks similar to that of the GT image. Later, the refinement network uses this frequency-domain information to produce better inpainting results.

We also show the performance of our proposed method on the CelebA and PSV datasets for irregular masks. Figure 6 shows the inpainting results using GI[17] and our proposed method for different percentages (10% to 50%) of mask size. Our method can generate photorealistic images having similar texture and structures as the original clean images even when a large region (50% to 60%) of the image is missing.

5.2 *Quantitative Evaluation*

We report the quantitative performance of our method in terms of the following metrics: (i) peak-signal-to-noise ratio (PSNR), (ii) SSIM,[47] and (iii) mean absolute error. Table 3 demonstrates the comparison in metric values on the CelebA, PSV, and DTD datasets for the state-of-the-art inpainting methods and our method. Our method outperforms the other methods in terms of these metrics on both regular and irregular masks. This proves the effectiveness of using frequency-domain information. Note that we obtain the metrics for the CE[12] using the l_1 and adversarial loss in our network settings.

We also report the quantitative performance of previous methods and our method in terms of two inpainting specific metrics, namely gradient magnitude similarity deviation (GMSD)[48] and visual saliency-induced index (VSI)[49] for perceptual image quality assessment of the inpainted

Table 3 Quantitative results on CelebA,[44] PSV,[45] and DTD[46] for different inpainting models: patchMatch, PM;[9] context encoder, CE;[12] contextual attention, CA;[14] generative inpainting, GI;[17] and ours. The best results for each row are shown in bold.

		CelebA dataset					PSV dataset					DTD dataset				
	Mask (%)	PM[9]	CE[12]	CA[14]	GI[17]	Ours	PM[9]	CE[12]	CA[14]	GI[17]	Ours	PM[9]	CE[12]	CA[14]	GI[17]	Ours
PSNR[a]	10 to 20	15.78	32.49	29.81	30.65	**32.69**	22.03	31.59	30.68	30.42	**32.34**	22.43	29.28	28.43	29.29	**29.89**
	20 to 30	15.09	29.62	27.06	27.22	**29.78**	20.42	28.69	27.40	27.09	**29.25**	21.11	27.02	25.73	26.34	**27.38**
	30 to 40	14.42	27.31	24.77	24.83	**27.49**	19.36	27.02	25.42	24.95	**27.33**	20.12	25.33	23.76	24.41	**25.65**
	40 to 50	13.63	25.10	23.03	22.86	**25.27**	18.52	25.09	23.99	23.23	**25.13**	19.26	23.89	22.35	22.75	**23.95**
	Regular	14.96	**28.17**	27.86	26.06	28.13	19.23	27.32	28.29	25.12	**28.42**	14.75	27.33	27.26	25.73	**27.49**
SSIM[a]	10 to 20	0.632	0.991	0.986	0.987	**0.992**	0.766	0.978	0.972	0.969	**0.981**	0.704	0.933	0.922	0.935	**0.942**
	20 to 30	0.579	0.983	0.971	0.971	**0.984**	0.692	0.958	0.945	0.936	**0.963**	0.634	0.890	0.861	0.872	**0.901**
	30to 40	0.513	0.972	0.953	0.952	**0.973**	0.613	0.938	0.912	0.896	**0.942**	0.563	0.841	0.793	0.804	**0.854**
	40 to 50	0.421	0.954	0.930	0.927	**0.956**	0.515	0.904	0.873	0.850	**0.910**	0.475	0.773	0.717	0.714	**0.785**
	Regular	0.571	0.970	0.968	0.953	**0.971**	0.659	0.923	0.934	0.880	**0.936**	0.149	0.876	0.869	0.833	**0.879**
l_1 (%)[b]	10 to 20	13.14	0.84	1.37	1.21	**0.82**	6.15	1.09	1.40	1.44	**0.97**	7.87	1.87	1.92	1.81	**1.67**
	20 to 30	14.58	1.41	2.24	2.07	**1.39**	7.78	1.93	2.45	2.52	**1.78**	8.85	2.85	3.02	2.93	**2.62**
	30 to 40	16.07	2.13	3.28	3.09	**2.09**	9.39	2.70	3.43	3.66	**2.57**	9.76	3.82	4.20	4.11	**3.58**
	40 to 50	17.89	3.13	4.40	4.22	**3.08**	10.8	3.75	4.40	4.79	**3.58**	10.70	4.94	5.40	5.43	**4.74**
	Regular	13.67	1.55	1.76	2.12	**1.55**	9.04	1.97	1.93	2.76	**1.77**	17.60	2.12	2.40	2.74	**2.05**

[a]Higher is better.
[b]Lower is better.

Table 4 Quantitative results in terms of image perceptual quality metrics on PSV dataset[45] for different deep-learning-based inpainting models: context encoder, CE;[12] contextual attention, CA;[14] generative inpainting, GI;[17] and ours. The best results for each row are shown in bold.

	Mask (%)	CE[12]	CA[14]	GI[17]	Ours
GMSD[a]	10 to 20	0.0621	0.0637	0.0736	**0.0574**
	20 to 30	0.0877	0.0946	0.1078	**0.0819**
	30 to 40	0.1094	0.1184	0.1345	**0.1069**
	40 to 50	0.1344	0.1432	0.1585	**0.1317**
	Regular	0.1032	0.1043	0.1196	**0.0987**
VSI[b]	10 to 20	0.985	0.981	0.981	**0.987**
	20 to 30	0.973	0.966	0.964	**0.976**
	30 to 40	0.962	0.952	0.947	**0.964**
	40 to 50	0.947	0.937	0.928	**0.949**
	Regular	0.967	0.967	0.956	**0.969**

[a]Lower is better.
[b]Higher is better.

image. Table 4 provides the quantitative values on the PSV dataset for both regular and irregular masks. We can see our method consistently achieves low GMSD scores (indicating low distortion range and high perceptual quality of the inpainted image) compared with the other methods. On the other hand, our method also achieves high VSI scores compared with the other algorithms, which ensures that the inpainted images obtained using our method have higher perceptual image quality. These metrics furthermore prove the effectiveness of our proposed frequency-based inpainting algorithm.

5.3 *Ablation Study*

We perform an ablation study to investigate the role of our frequency deconvolution network and to analyze the effect of different loss components used to train our model. Figure 7 shows the inpainting results using only l_1 loss, l_1 with adversarial loss, and our proposed method of incorporating frequency-domain information (DFT component). We can see blurry reconstructions in

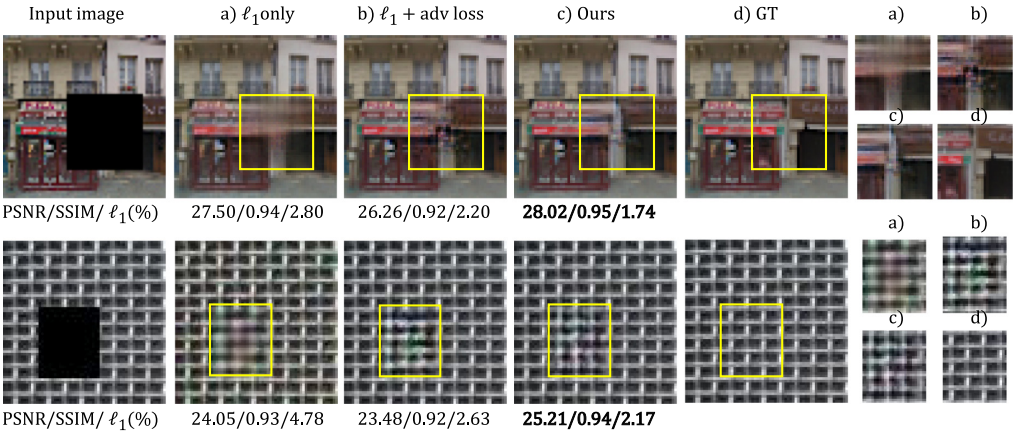

Fig. 7 Visual results on the PSV dataset (first row) and DTD (second row) showing the effect of different components in our model on the input incomplete images (first column): (a) results using standard l_1 loss; (b) results using l_1 + adversarial loss; (c) results of our model trained using l_1 + adversarial loss (with DFT component); and (d) the GT image.

Table 5 Analysis of computation time of our model on CelebA,[44] PSV,[45] and DTD.[46]

Mask (%)	CelebA[44] (s)	PSV[45] (s)	DTD[46] (s)
10 to 20	0.048	0.045	0.050
20 to 30	0.103	0.045	0.058
30 to 40	0.052	0.047	0.057
40 to 50	0.053	0.058	0.052
Regular	0.059	0.117	0.036

Fig. 7(a) when we use only l_1 loss in the spatial domain. However, inpainting performance improves to a certain extent if we add the adversarial loss component. Nevertheless, in Fig. 7(b), we can still find structural and blurry artifacts on the reconstructions. Figure 7(c) demonstrates the inpainting results of our proposed method of training the model using both frequency and spatial components. We can see in that using our method the model can perform significantly better by restoring fine structural details. Therefore, we can conclude that training the model along with frequency-domain information certainly helps the network to learn high-frequency components and restore the missing region with better reconstruction quality.

5.4 *Computational Complexity*

We measure the computational complexity of our method in terms of inference time for three different datasets: CelebA, PSV, and DTD. Table 5 shows the average computation time of image inpainting using our method for regular masks and various percentages of irregular masks. Note again that we evaluate our method on a NVIDIA GeForce TITAN Xp (12 GB) GPU. Our method can generate inpainted images in millisecond order using a GPU. Based on the computation time, our method can inpaint 10 to 20 frames in each second, thus making its application to real-time frame-wise video reconstruction a possibility.

6 Conclusions

We presented a frequency-based image inpainting algorithm that enables the network to use both frequency and spatial information to predict the missing region of an image. Our model first learned the global context using frequency-domain information and selectively reconstructed the high-frequency components. Then it used the spatial-domain information as a guidance to match the color distribution of the true image and fine-tuned the details and structures obtained in the first stage, leading to better inpainting results. Experimental results showed that our method could achieve results better than state-of-the-art performances on challenging datasets by generating sharper details and perceptually realistic inpainting results. Based on our empirical results, we believe that methods using both frequency and spatial information should gain dominance because of their superior performance. Our proposed approach can be extended to videos and to a temporally changing flow of images as well using continuity constraints on each frame of the videos to ensure that there is no discontinuity in the frame-wise motion of the pixels. This can be implemented by reducing the sum of L1 distance between each frame after inpainting the corresponding frame. In the future, we want to extend this work to using other kinds of frequency-domain transformations, e.g., discrete cosine transform and to solving other kinds of image restoration tasks, e.g., image denoising.

Acknowledgments

This paper was partially financially supported by Japan Society for the Promotion of Science KAKENHI Grant (No. JP19K22863).

References

1. M. Bertalmio et al., "Image inpainting," in *27th Annu. Conf. Comput. Graphics and Interactive Tech.*, pp. 417–424 (2000).
2. C. Ballester et al., "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Process.* **10**(8), 1200–1211 (2001).
3. M. Bertalmio et al., "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.* **12**(8), 882–889 (2003).
4. A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics," in *IEEE Int. Conf. Comput. Vision*, Vol. 1, pp. 305–312 (2003).
5. A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *28th Annu. Conf. Comput. Graphics and Interactive Tech.*, pp. 341–346 (2001).
6. I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," in *ACM SIGGRAPH*, pp. 303–312 (2003).
7. A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.* **13**(9), 1200–1212 (2004).
8. D. Simakov et al., "Summarizing visual data using bidirectional similarity," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1–8 (2008).
9. C. Barnes et al., "PatchMatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graphics* **28**(3), 1–11 (2009).
10. S. Darabi et al., "Image melding: combining inconsistent images using patch-based synthesis," *ACM Trans. Graphics* **31**(4), 1–10 (2012).
11. K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Trans. Pattern Anal. Mach. Intell.* **36**(12), 2423–2435 (2014).
12. D. Pathak et al., "Context encoders: feature learning by inpainting," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2536–2544 (2016).
13. S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graphics* **36**(4), 1–14 (2017).
14. J. Yu et al., "Generative image inpainting with contextual attention," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 5505–5514 (2018).
15. Y. Song et al., "SPG-Net: segmentation prediction and guidance network for image inpainting," in *Br. Mach. Vision Conf.* (2018).
16. K. Nazeri et al., "Edgeconnect: structure guided image inpainting using edge prediction," in *IEEE Int. Conf. Comput. Vision Workshops* (2019).
17. J. Yu et al., "Free-form image inpainting with gated convolution," in *IEEE Int. Conf. Comput. Vision*, pp. 4471–4480 (2019).
18. I. Goodfellow et al., "Generative adversarial nets," in *Adv. Neural Inf. Process. Syst.*, pp. 2672–2680 (2014).
19. T. Ogawa and M. Haseyama, "Image inpainting based on sparse representations with a perceptual metric," *EURASIP J. Adv. Signal Process.* **2013**(1), 179 (2013).
20. Y. LeCun et al., "Handwritten digit recognition with a back-propagation network," in *Adv. Neural Inf. Process. Syst.*, pp. 396–404 (1990).
21. J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Adv. Neural Inf. Process. Syst.*, pp. 341–349 (2012).
22. R. Köhler et al., "Mask-specific inpainting with deep neural networks," *Lect. Notes Comput. Sci.* **8753**, 523–534 (2014).
23. S. Chaudhury and H. Roy, "Can fully convolutional networks perform well for general image restoration problems?" in *IAPR Int. Conf. Mach. Vision Appl.*, pp. 254–257 (2017).
24. J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.* **17**(1), 53–69 (2008).
25. R. A. Yeh et al., "Semantic image inpainting with deep generative models," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 5485–5493 (2017).
26. M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," in *Proc. 34th Int. Conf. Machine Learn.*, Vol. 70, pp. 214–223 (2017).
27. I. Gulrajani et al., "Improved training of Wasserstein GANs," in *Adv. Neural Inf. Process. Syst.*, pp. 5767–5777 (2017).

28. G. Liu et al., "Image inpainting for irregular holes using partial convolutions," *Lect. Notes Comput. Sci.* **11215**, 85–100 (2018).
29. L. Gueguen et al., "Faster neural networks straight from JPEG," in *Adv. Neural Inf. Process. Syst.*, pp. 3933–3944 (2018).
30. K. Xu et al., "Learning in the frequency domain," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1740–1749 (2020).
31. K. Zhang et al., "Beyond a Gaussian denoiser: residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.* **26**(7), 3142–3155 (2017).
32. S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Int. Conf. Mach. Learn.*, pp. 448–456 (2015).
33. K. Xu, Z. Zhang, and F. Ren, "LAPRAN: a scalable Laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction," *Lect. Notes Comput. Sci.* **11214**, 485–500 (2018).
34. A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Int. Conf. Learn. Represent.* (2016).
35. J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," *Lect. Notes Comput. Sci.* **9906**, 694–711 (2016).
36. J.-Y. Zhu et al., "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *IEEE Int. Conf. Comput. Vision*, pp. 2223–2232 (2017).
37. K. He et al., "Deep residual learning for image recognition," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 770–778 (2016).
38. D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: the missing ingredient for fast stylization," arXiv:1607.08022 (2016).
39. P. Isola et al., "Image-to-image translation with conditional adversarial networks," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1125–1134 (2017).
40. T. Miyato et al., "Spectral normalization for generative adversarial networks," in *Int. Conf. Learn. Represent.* (2018).
41. A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Int. Conf. Mach. Learn.*, Vol. 30, p. 3 (2013).
42. K. He et al., "Delving deep into rectifiers: surpassing human-level performance on ImageNet classification," in *IEEE Int. Conf. Comput. Vision*, pp. 1026–1034 (2015).
43. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *Int. Conf. Learn. Represent.* (2015).
44. Z. Liu et al., "Deep learning face attributes in the wild," in *IEEE Int. Conf. Comput. Vision*, pp. 3730–3738 (2015).
45. C. Doersch et al., "What makes Paris look like Paris?" *ACM Trans. Graphics* **31**(4), 1–9 (2012).
46. M. Cimpoi et al., "Describing textures in the wild," in *IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 3606–3613 (2014).
47. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).
48. W. Xue et al., "Gradient magnitude similarity deviation: a highly efficient perceptual image quality index," *IEEE Trans. Image Process.* **23**(2), 684–695 (2014).
49. L. Zhang, Y. Shen, and H. Li, "VSI: a visual saliency-induced index for perceptual image quality assessment," *IEEE Trans. Image Process.* **23**(10), 4270–4281 (2014).

Hiya Roy received her BE degree in electrical engineering from Jadavpur University, India, in 2012 and her MS and PhD degrees in electrical engineering and information systems from the University of Tokyo in 2017 and 2021, respectively. Her research interests are computer vision, machine learning, deep learning, and planetary sciences. She was an MEXT scholar from September 2015 to August 2020. She is a member of IEEE.

Subhajit Chaudhury received his BE degree in electrical engineering from Jadavpur University, Kolkata, India, in 2012 and his MTech degree in electrical engineering from the Indian Institute of Technology Bombay in 2014. He is a research scientist at IBM Research AI, Tokyo, Japan. Concurrently, he received his PhD degree in information and communication

engineering from the University of Tokyo in 2021. He worked as a researcher at NEC Research Laboratories, Japan, from October 2014 to March 2017. His current research interests include reinforcement learning and computer vision. He is a member of IEEE and ACM.

Toshihiko Yamasaki received his BS degree in electronic engineering, his MS degree in information and communication engineering, and his PhD from the University of Tokyo in 1999, 2001, and 2004, respectively. He is currently an associate professor in the Department of Information and Communication Engineering at the University of Tokyo. He was a JSPS fellow for Research Abroad and a visiting scientist at Cornell University from Feburary 2011 to Feburary 2013. His current research interests include attractiveness computing based on multimedia big data analysis and machine learning. He is a member of IEEE, ACM, IEICE, ITE, and IPSJ.

Tatsuaki Hashimoto received his PhD in electrical engineering from the University of Tokyo in 1990. Since April 1990, he has been working at the Institute of Space and Astronautical Science (ISAS). He is a professor of the ISAS/Japan Aerospace Exploration Agency and at the Graduate School of Engineering at the University of Tokyo. Currently, he is working on the research and development of spacecraft guidance, navigation, and control systems. He was a visiting scientist at the Jet Propulsion Laboratory of NASA in 2000. His research interests include image processing for spacecraft navigation and lunar surface exploration. He is a member of AIAA, IAA, IEEJ, SICE, etc.