GEOPHYSICS, VOL. 89, NO. 2 (MARCH-APRIL 2024); P. D89–D98, 10 FIGS., 3 TABLES.
10.1190/GEO2022-0344.1

Filling borehole image gaps with a partial convolution neural network

Lei Jiang[1], Xu Si[1], and Xinming Wu[1]

ABSTRACT

Borehole images are measured by logging tools in a well, providing a microresistivity map of the rock properties surrounding the borehole. These images contain valuable information related to changes in mineralogy, porosity, and fluid content, making them essential for petrophysical analysis. However, due to the special design of borehole imaging tools, vertical strips of gaps occur in borehole images. We develop an effective approach to fill these gaps using a convolutional neural network with partial convolution layers. To overcome the challenge of missing training labels, we introduce a self-supervised learning strategy. Specifically, we replicate the gaps found in borehole images by randomly creating vertical blank strips that mask out certain known areas in the original images. We then use the original images as label data to train the network to recover the known areas masked out by the defined gaps. To ensure that the missing data do not impact the training process, we incorporate partial convolutions that exclude the null-data areas from convolutional computations during forward and backward propagation of updating the network parameters. Our network, trained in this way, can then be used to reasonably fill the gaps originally appearing in the borehole images and obtain full images without any noticeable artifacts. Through the analysis of multiple real examples, we determine the effectiveness of our method by comparing it with three alternative approaches. Our method outperforms the others significantly, as demonstrated by various quantitative evaluation metrics. The filled full-bore images obtained through our approach enable enhanced texture analysis and automated feature recognition.

INTRODUCTION

Borehole microresistivity images are high-resolution images of borehole walls, obtained by measuring resistivity changes of fluid, rock, and the contrast of the resistivity value among different structural features or layers. Meanwhile, formation resistivity is a function of the shale and fluid content in shale pores, lithology, and fractures (Slim, 2007). Therefore, borehole images are widely used for reservoir evaluation by providing detailed rock-property information on the subsurface (Lofts and Bourke, 1999). Borehole images used in this paper are obtained using the formation microscopic imager (FMI) tool (Figure 1). It is an advanced borehole logging tool that offers microresistivity measurements and is used while logging the borehole with water-based mud. By using 192 measuring button electrodes, the FMI generates a borehole image (Figure 2a). As the button electrodes emit current to penetrate the formation, the resistivity changes in the formation (directly facing the button) cause variations in the current density. These variations are sampled by the button arrays (Rider, 1986). The signals received by the FMI comprise contributions from the deeper and shallower sections. The former is offered by the whole current, which obtains the analogous depth of research as a laterolog tool. The latter is provided by the change in the button current itself. The microresistivity curve is recorded for each button, and subsequently, the FMI calculates the average microresistivity curve for the entire pad (Luthi, 2001).

A borehole image (Figure 2b) is a 2D image represented by a color scale reflecting the relative value of the resistivity of the borehole wall formation. The color palette commonly used for borehole images consists of black, brown, yellow, and white, with several color levels indicating the range of resistivity from low to high. The subtle variations in color signify changes in lithology and physical properties of the formation. However, restrictions in

Manuscript received by the Editor 19 May 2022; revised manuscript received 21 August 2023; published ahead of production 31 October 2023; published online 27 December 2023.
[1]University of Science and Technology of China, School of Earth and Space Sciences, Laboratory of Seismology and Physics of Earth's Interior, Hefei, China and University of Science and Technology of China, Mengcheng National Geophysical Observatory, Hefei, China. E-mail: jianglei0510@mail.ustc.edu.cn; xusi@mail.ustc.edu.cn; xinmwu@ustc.edu.cn (corresponding author).

the design of borehole image logging tools result in vertically distributed gaps within the generated images. The coverage of the FMI borehole image reaches 60%. The remaining 40% lacks information, making the interpretation of borehole images more challenging and leading to an inaccurate estimation of factors such as fracture cavities. Although the introduction of new logging tools or combinations of multiple tools may reduce the occurrence of gaps, completely eliminating them during data acquisition remains a challenge. Therefore, many postprocessing methods have been proposed to fill the gaps present in borehole images.

The FilterSim, proposed by Zhang et al. (2006), is a pattern-based multipoint statistics (MPS) algorithm used to fill gaps in borehole images. It accomplishes this by extracting patterns from other parts of the image or a database. Hurley and Zhang (2011) first apply the FilterSim algorithm to fill gaps in logging images with simple stratum and obtain reasonable full images. Similarly, Mariethoz and Renard (2010) propose a direct sampling MPS method to reconstruct blank strips in borehole images without the need for an external training image, provided that sufficient training data sets are available. However, this method exhibits limitations when dealing with large-scale structures or thin and highly undulating curvilinear features. To handle complex nonlinear geologic patterns, Mustapha and Dimitrakopoulos (2010) propose another advanced MPS method based on high-order spatial cumulants derived from training images. The term high-order spatial cumulants can be regarded as an advanced covariance function (Dimitrakopoulos et al., 2009). In contrast to the MPS method, Assous et al. (2013) propose a technique that uses a sparse representation of multiscale and multidirectional curvelet transformations to decompose the imaged parts into morphological components. Full borehole images are then obtained through the inverse transformation and reconstruction of these components. This approach is entirely data-driven and does not rely on any geologic model, although it may require significant computational resources. To solve this issue, Assous and Elkington (2018) enhance the inpainting method by introducing the shearlets transformation, which offers improved computational efficiency compared with the curvelet transformation. This method shows that sharp high-contrast edges yield more accurate reconstruction outcomes. However, all of these MPS-derived methods discussed here do not explicitly consider additional information about the structural information of the borehole images. Therefore, Zhang et al. (2017) make further improvements to the FilterSim algorithm by integrating a structural trend map to guide the simulation process.

Most of the aforementioned methods can be categorized as local image processing techniques, which may struggle to capture the global structure within borehole images. Over the past few decades, several image inpainting methods based on convolutional neural networks (CNNs) have emerged. Cai et al. (2017) propose a blind inpainting method called BICNN, using an encoder-decoder structure. Yan et al. (2018) introduce Shift-Net, which adopts the U-Net architecture (Ronneberger et al., 2015), achieving highly accurate filling of unknown regions by leveraging structures and detailed textures. For inpainting irregular masks, Liu et al. (2018) use partial convolutions and automatic mask update mechanisms. Zeng et al. (2019) develop PEN-NET, a pyramidal-context architecture dedicated to high-quality image inpainting. These CNN-based methods have demonstrated successful applications in inpainting natural images. In recent years, deep-learning methods have also been used for inpainting or interpolating seismic data sets (Kaur et al., 2019,

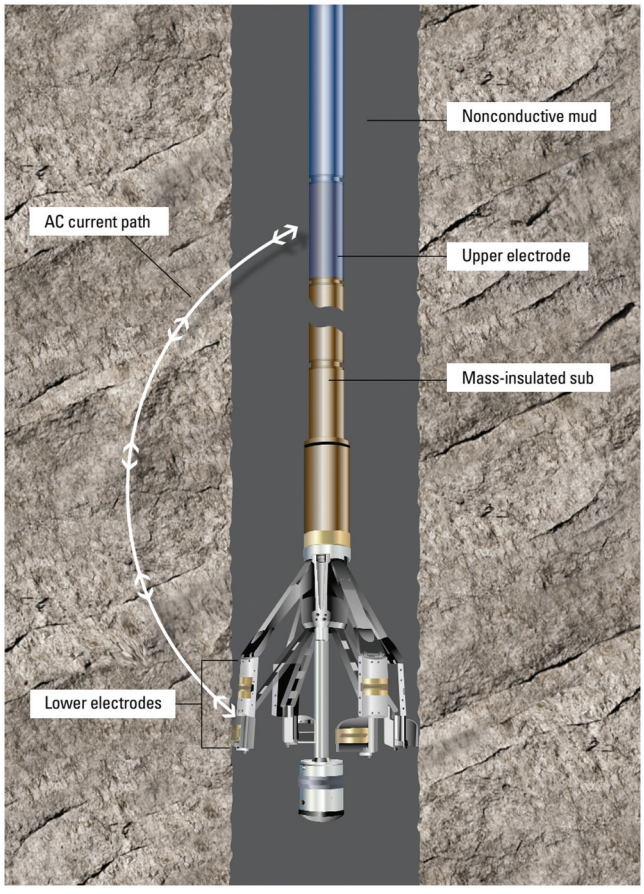

Figure 1. The FMI tool (Schlumberger, 2013). The FMI tool measures the variations in formation resistivity by recording the current changes and then creates a microresistivity image of the borehole wall. Because of the special setup of the instrument, the gaps between pads result in blank strips in the logging image generated by FMI.

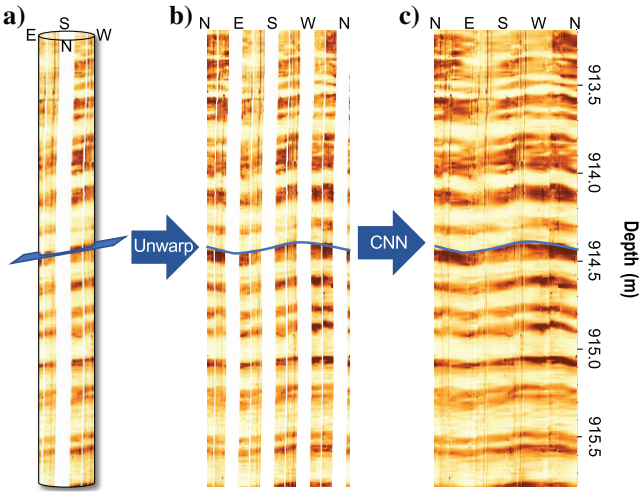

Figure 2. (a) A cylindrical FMI borehole image, obtained by the FMI tool (Figure 2), often contains several vertically distributed gaps. We propose a deep-learning method with partial convolution to fill gaps in the (b) unwrapped borehole image and obtain (c) a full image without filling artifacts.

DOI:10.1190/geo2022-0344.1

2021; Mandelli et al., 2019). However, directly applying these methods to borehole images is not straightforward. One significant challenge is that these methods typically need numerous full images as training labels, and we do not have any full borehole images in practice.

Recently, Wang et al. (2019) use a sample-free deep-learning method known as the deep image prior (DIP) network (Ulyanov et al., 2018) to fill borehole images. The key advantage of this method is that it does not rely on full images for network training. However, this method may introduce artifacts in the filled images. More recently, Du et al. (2020) propose an enhanced version of DIP that incorporates attention modules yielding satisfactory results in complex formation structures. Nevertheless, these methods are time consuming, typically requiring more than 2000 iterations to reconstruct a single borehole image. Moreover, determining the optimal number of iterations for achieving a high-quality inpainting result can be challenging. In addition to the DIP-based methods, Saad et al. (2020) propose a texture inpainting method based on generative adversarial networks. This method exhibits improved capability in restoring texture features, but it necessitates full borehole images as labels.

We propose a deep-learning workflow (Figure 3) to further improve the efficiency and accuracy of filling gaps in borehole images. In this workflow, we adopt a self-supervised learning strategy to overcome the challenge of missing training labels. By randomly masking out some vertical known areas in the original images, we generate training samples using the original borehole images as label data. This concept of masking and reconstruction is in alignment with traditional geophysical techniques such as prediction error filters used by Claerbout (1992), offering a bridge between modern deep-learning and classical inversion methods. Subsequently, a CNN is trained to restore the masked regions. Once trained, the original borehole images can be processed through the trained CNN to obtain complete images without any gaps, as shown at the bottom of Figure 3. To ensure high-quality recovered images, we propose an advanced CNN architecture and train the network using a combination of multiple loss functions including the mean absolute error (MAE) (ℓ_1) loss, perceptual loss (Johnson et al., 2016), and multiscale structural similarity index method (MS-SSIM) loss (Wang et al., 2003). Furthermore, we incorporate partial convolutional layers (Liu et al., 2018) into our CNN architecture instead of normal convolutions. This enables the CNN to focus solely on valid pixels, excluding the original gaps or masked pixels from convolutional operations in the forward and backward propagation. Through this training approach, our network is able to reasonably restore images without introducing any filling artifacts. The recovered full borehole images, along with various quantitative comparison results, show that our trained network performs well in filling gaps in test images.

TRAINING DATA SETS

Training a CNN model for borehole image completion typically requires a large number of training samples accompanied by labels of complete images. However, in practical scenarios, obtaining a sufficient quantity of full borehole images can be challenging. To address this issue, we propose a self-supervised learning strategy that alleviates the reliance on labeled full borehole images.

Data preprocessing

The data sets used in this study originate from two wells. The first well, namely C0009A, is situated in the Nankai Trough (McNeill et al., 2010). The lithology of this well primarily comprises silty mud and mudstone with rare sand and volcanic ash interbeds and abundant wood/lignite fragments. The drilling depth of this well spans from 707.7 m below the seafloor (mbsf) to 1603.7 mbsf, covering the late Miocene to the Holocene period. Table 1 provides detailed information about this particular well. The second well, denoted as C0020A, is located off the coast of the Shimokita Peninsula (Inagaki et al., 2016). The lithology encountered in this well consists of diatom-rich silty clay, shale, sandstone, siltstone, and coal-rich horizons. It is drilled from 647 to 2466 mbsf encompassing the late Oligocene to the late Pliocene timeframe. Table 2 presents the specific characteristics of this well. Both of these wells exhibit a vertical orientation.

Both data sets used in this study are obtained from the International Ocean Discovery Program (IODP) as open-source FMI logging images (Columbia University, Lamont-Doherty Earth Observatory, 2022). The FMI tool offers coverage of approximately 60% of the borehole wall within an 8 in diameter borehole. To transform the electrical current measurements acquired by the logging tool into an image representing the variations in conductivity, the IODP achieves two processing phases: data restoration and image display. To facilitate the CNN training process, we convert the original colorful borehole images to grayscale. This conversion not only eliminates color-related interference, preventing erroneous color filling, but also reduces the data size for processing, thereby expediting network training. In Figure 2, it is evident that there are four narrow gaps, each two pixels wide, and four wider

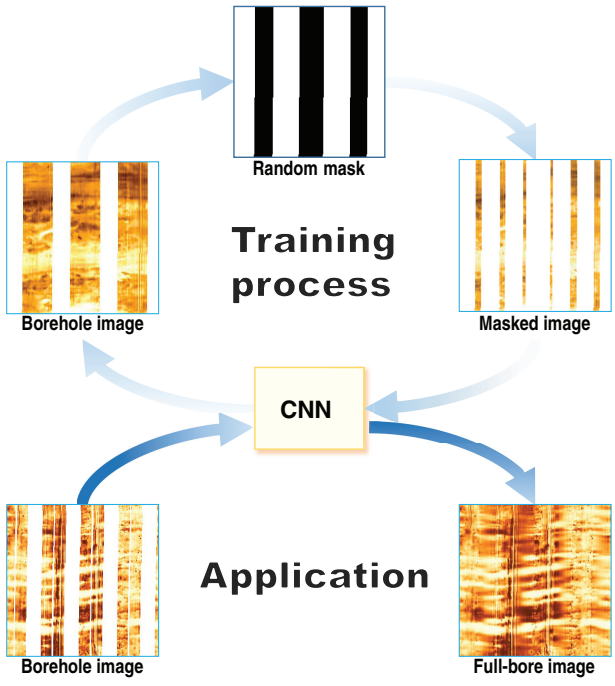

Figure 3. The proposed workflow of filling gaps in borehole images. We propose a self-supervised learning strategy to train our CNN and then apply the trained CNN to fill the gaps in real borehole images.

gaps. Prior to further analysis, we use a nearest neighbor interpolation method to prefill these narrow gaps.

Self-supervised learning

Deep-learning methods rely on numerous data for training, which can be challenging when working with limited borehole images. Unlike natural image data sets that are abundant in data, the availability of borehole images is relatively constrained. In addition, the absence of complete borehole images for training labels further complicates the matter. To address this issue, we propose the self-supervised learning workflow shown in Figure 3, in which we randomly create vertical blank strips, called random masks, masking out parts of known areas on the borehole images. These masked images and the corresponding random masks serve as inputs to our CNN model. The original borehole images are used as partially labeled targets. To achieve partial supervision, we use an adaptive loss function that evaluates the output images exclusively at the known pixels within the partially labeled targets (original borehole images). To ensure effective gap filling, we incorporate partial convolutional layers into the network architecture. This inclusion ensures that the convolution operations exclude original gaps. Through this approach, the CNN learns to fill gaps in the masked images using information derived solely from the borehole images themselves, thus addressing the challenge of lacking full borehole images as labels.

Training data generation

We now illustrate our training strategy with one sample in Figure 4. First, given an original borehole image $\mathbf{I}_{gt}$ (Figure 4a), we detect the missing-data area $\mathbf{M}_1$, as shown in Figure 4b. Next, we horizontally shift $\mathbf{M}_1$ and apply dilation or erosion operations to the shifted image, resulting in $\mathbf{M}_2$ (Figure 4d) with a width comparable to $\mathbf{M}_1$. Using $\mathbf{M}_2$, we perform a pixel-wise multiplication with $\mathbf{I}_{gt}$, generating a masked image denoted as $\mathbf{I}_{M_2}$ (Figure 4e). Here, $\mathbf{I}_{M_2}$ retains the structural characteristics of the original borehole $\mathbf{I}_{gt}$. Consequently, we obtain a comprehensive training sample comprising the following components: a masked image $\mathbf{I}_{M_2}$, a mask $\mathbf{M}_2$, and the corresponding label data $\mathbf{I}_{gt}$. Finally, to increase the diversity of the training samples, we introduce random adjustments to the width of $\mathbf{M}_2$ through the operations of lateral shift, dilation, and erosion. Dilation and erosion are fundamental morphological operations (Soille, 1999), commonly applied to binary images. Dilation expands the highlighted or white regions in an image by adding pixels to the perceived boundaries of objects. Conversely, erosion removes pixels along object boundaries and contracts bright white areas.

The complete process for generating training data sets is shown in Figure 5. To increase the generalization ability of the CNN and enable the CNN to learn more diverse patterns from borehole images, we randomly load a borehole image log (Figure 5a) and then randomly crop a batch of training samples from this log. In each batch

Table 1. Lithologic units with sediment age information, Hole C0009A. LSF, logging-while-drilling depth below seafloor.

Unit	Depth interval MSF (m)	Age	Lithology
I	0–467 (LSF)	Holocene-Pleistocene	Silty mud with cyclical sand-rich layers
II	707.7–812.7	Pleistocene	Silty mud with silt and sand interbeds
III	812.7–1287.7	Pleistocene-late Pliocene	Silty mudstone with rare silty-sand interbeds
IV	1287.7–1603.7	Late Miocene	Silty mudstone with minor silt interbeds and rare interbeds of fine vitric tuff

MSF, mud depth below the seafloor.

Table 2. Lithologic units with sediment age information, Hole C0020A.

Unit	Depth interval MSF(m)	Age	Lithology	
I	a	647–926.5	Late-middle Pliocene	Diatom-bearing siltstone and claystone
	b	926.5–1116.5	Middle Pliocene/early Pliocene-late Miocene	Semiconsolidated diatom-bearing clayey siltstone with common fine sandstone
	c	1116.5–1236.5	Early Pliocene-Miocene	Unconsolidated to semiconsolidated sandstone and silty sandstone with rare clayey siltstone
	d	1236.5–1256.5	Early Pliocene-Miocene	Semiconsolidated clayey siltstone with medium loose sand
II	a	1256.5–1506.5	Miocene	Sandstone and siltstone associated with marine fossiliferous
	b	1506.5–1826.5	Miocene	Organic-rich shale and sandstone associated with plant remains
III		1826.5–2046.5	Early-middle Miocene	Organic-rich sandstone and shale associated with coalbeds
IV	a	2046.5–2426.5	Early Miocene-late Oligocene	Shale and sandstone associated with carbonate and glauconitic material
	b	2426.5–2466	Early Miocene-late Oligocene	Sandstone and shale associated with coalbeds

MSF, mud depth below the seafloor.

of training, we randomly crop 16 256 × 256 images (Figure 5b) as training samples, so that the CNN can learn different patterns in each batch. In this case, we also minimize the risk of overfitting caused by a small number of training samples. We simultaneously load the corresponding original masks and generate corresponding random masks (Figure 5c) while randomly cropping training samples. The random masks are used to mask the cropped images and obtain masked images (Figure 5d) for our training samples which consist of masked images, random masks, and original images. Furthermore, to augment the diversity of the training data, we apply data augmentations such as lateral (Figure 5e) or vertical flips (Figure 5f).

ARCHITECTURE AND TRAINING DETAILS OF THE NETWORK

Our gap-filling network architecture draws inspiration from Liu et al. (2018), which uses stacked partial convolutions and renewing masks for image inpainting purposes. To enhance the visual quality of the results, we combine three loss functions, the ℓ_1 loss, perceptual loss, and MS-SSIM loss.

Partial convolution layer

The partial convolution layer contains partial convolution operations and mask renewing operations. We use $\mathbf{W}$ and $\mathbf{b}$ to denote the convolution filter weights and bias, respectively. Here, $\mathbf{X}$ represents the borehole image values $\mathbf{I}_{M_2}$ of the current convolution sliding window and $\mathbf{M}$ represents the corresponding mask $\mathbf{M}_2$, again, in the sliding window. After convolution, the output values are defined as

$$\mathbf{x}' = \begin{cases} \frac{\text{sum}(\mathbf{Id})}{\text{sum}(\mathbf{M})}\mathbf{W}^T(\mathbf{X}\odot\mathbf{M}) + \mathbf{b}, & \text{if sum}(\mathbf{M}) > 0 \\ 0, & \text{otherwise} \end{cases}, \quad (1)$$

where $\odot$ represents the element-wise multiplication and $\mathbf{Id}$ is the unit matrix and has the same shape as $\mathbf{M}$. This equation demonstrates that the output values $\mathbf{x}'$ are exclusively computed using unmasked inputs ($\mathbf{M} = 1$). The scaling factor $\text{sum}(\mathbf{Id})/\text{sum}(\mathbf{M})$ is used to adjust the changing number of valid (unmasked) pixels.

Upon completion of the partial convolution process, the mask $\mathbf{M}$ undergoes the following update: if the input within the sliding window for convolution encompasses more than one valid pixel, the corresponding location is designated as valid. The formula is as follows:

$$m' = \begin{cases} 1, & \text{if sum}(\mathbf{M}) > 0 \\ 0, & \text{otherwise} \end{cases}. \quad (2)$$

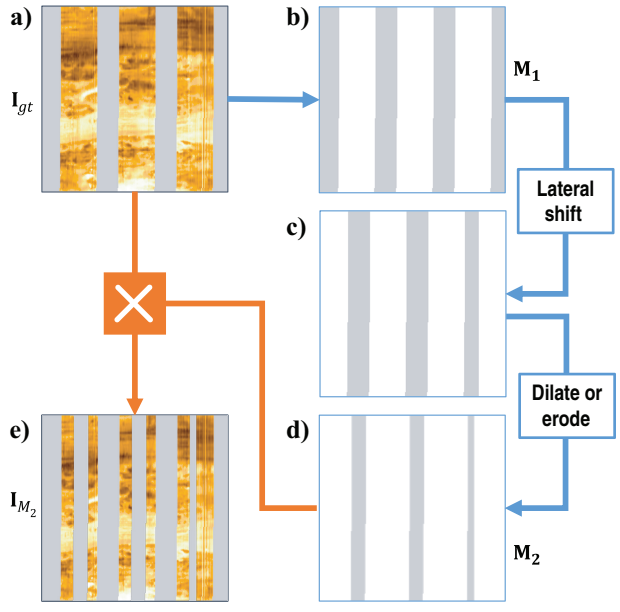

Figure 4. The workflow of generating a training sample. From the (a) original borehole image $\mathbf{I}_{gt}$, we obtain (b) the original mask $\mathbf{M}_1$. We laterally shift (b) $\mathbf{M}_1$ to obtain (c), which is a circular shift on the x-axis. Then, we dilate or erode (c) the shifted image to obtain (d) a random mask $\mathbf{M}_2$. Finally, we multiply (a) $\mathbf{I}_{gt}$ and (d) $\mathbf{M}_2$ to obtain (e) a masked image $\mathbf{I}_{M_2}$.

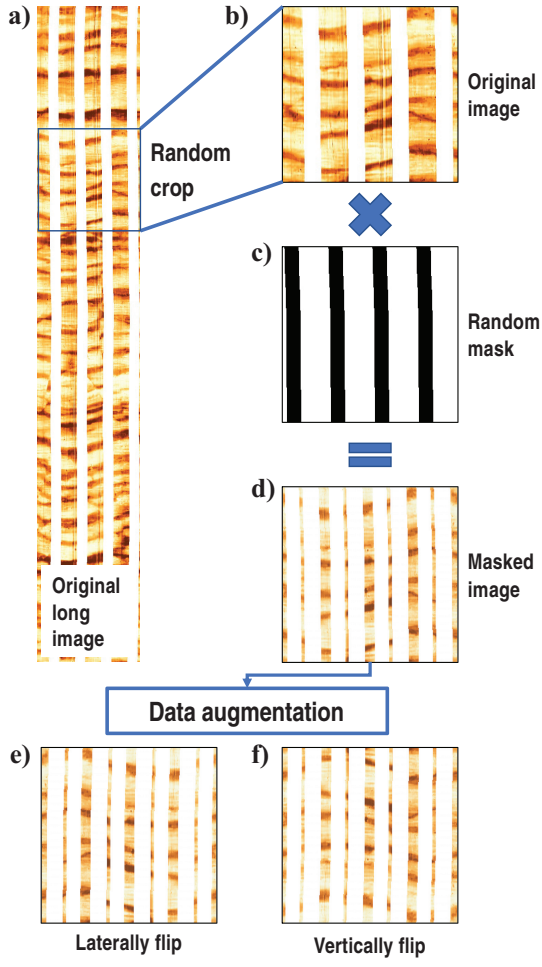

Figure 5. The proposed workflow of generating training samples while training the network. (a) A long borehole image randomly loaded from a large data set. We randomly crop (b) a 256 × 256 image from (a) and generate (c) the corresponding random mask in real time. We multiply (c) the random mask and (b) the cropped original image to generate (d) a masked image. We further apply (e) lateral or (f) vertical flip to the generated training sample.

Under sufficiently continuous operations of partial convolution layers, the mask will ultimately be updated to a matrix filled entirely with ones, as long as the input includes at least one unmasked pixel.

CNN architecture

Figure 6 shows the proposed CNN architecture, where the input and output are single-channel images with dimensions of $1 \times 256 \times 256$. The input of the CNN consists of a masked image $\mathbf{I}_{M_2}$ and the corresponding random mask $\mathbf{M}_2$, whereas the output is a mask-filled logging image $\mathbf{I}_{\text{out}}$. We have devised an encoder-decoder network based on a modified version of U-Net (Ronneberger et al., 2015). Specifically, we use partial convolutional layers instead of the conventional 3×3 convolutional layers. The skip connection concatenates two feature maps and two masks at the same level, respectively. The encoding stage is composed of five basic blocks. Each block sequentially consists of two 3×3 partial convolution layers with stride 1, a 2×2 max-pooling layer for downsampling, a batch normalization (BN) layer (except for the first encoding block), and a leaky rectified linear unit (LeakyReLU) activation with an alpha of 0.2. The number of feature channels is 32, 64, 128, 256, and 512, respectively. Similarly, the decoding stage is composed of five basic blocks corresponding to the encoding stage. Each block contains 2×2 nearest-neighbor upsampling, concatenation with the corresponding feature maps and masks from the encoding stage, respectively, a 3×3 partial convolution layer, a BN layer (except for the last decoding block), and a LeakyReLU activation with an alpha of 0.2. The number of feature channels is 256, 128, 64, 32, and 3, respectively. In addition, a 1×1 convolution is used in the final stage to reshape the feature channels into a single channel representing a grayscale borehole image.

Loss functions

Given the restored output image $\mathbf{I}_{\text{out}}$ obtained from the network, the label image (origin borehole image) $\mathbf{I}_{gt}$, and the corresponding original mask $\mathbf{M}_1$, we first define our pixel-wise ℓ_1 loss as follows:

$$\mathcal{L}_{\ell_1} = \|(\mathbf{I}_{\text{out}} - \mathbf{I}_{gt}) \cdot \mathbf{M}_1\|_1. \tag{3}$$

Here, we multiply $\mathbf{M}_1$ with each term in each loss function to eliminate the effects of the missing data without a label.

To enhance the visual quality, we then incorporate the perceptual loss that is defined as follows:

$$\mathcal{L}_{\text{perceptual}} = \sum_{n=0}^{N-1} \|\Psi_n(\mathbf{I}_{\text{out}} \cdot \mathbf{M}_1) - \Psi_n(\mathbf{I}_{gt} \cdot \mathbf{M}_1)\|_1 + \sum_{n=0}^{N-1} \|\Psi_n(\mathbf{I}_{\text{comp}} \cdot \mathbf{M}_1) - \Psi_n(\mathbf{I}_{gt} \cdot \mathbf{M}_1)\|_1, \tag{4}$$

where $\mathbf{I}_{\text{comp}} = \mathbf{I}_{\text{out}} \cdot (1 - \mathbf{M}_2) + \mathbf{I}_{gt} \cdot \mathbf{M}_2$, and $\mathbf{M}_2$ represents a randomly defined mask, and $\mathbf{I}_{\text{comp}}$ denotes the raw-filled image $\mathbf{I}_{\text{out}}$ with its nonmasked pixels replaced by $\mathbf{I}_{gt}$. Before calculating the ℓ_1 distances in equation 4, we project output images into a high-level feature space using the pretrained VGG-16 CNN model (Simonyan and Zisserman, 2014). In equation 4, Ψ_n is the activation of the nth chosen layer of the VGG-16 model, which corresponds to the feature maps from layers pool1, pool2, and pool3, respectively.

To achieve high-resolution filled images, we incorporate the MS-SSIM, which measures structural similarity at the multiscale of images, and is an improved algorithm based on the structural similarity index method (SSIM) (Wang et al., 2004). The SSIM is defined as follows:

$$\text{SSIM}(\mathbf{x}, \mathbf{y}) = [l(\mathbf{x}, \mathbf{y})]^{\alpha} \cdot [c(\mathbf{x}, \mathbf{y})]^{\beta} \cdot [s(\mathbf{x}, \mathbf{y})]^{\gamma}, \tag{5}$$

where

$$\begin{aligned} l(\mathbf{x}, \mathbf{y}) &= \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}, \\ c(\mathbf{x}, \mathbf{y}) &= \frac{2\sigma_{xy} + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}, \\ s(\mathbf{x}, \mathbf{y}) &= \frac{\sigma_{xy} + c_3}{\sigma_x\sigma_y + c_3}, \end{aligned} \tag{6}$$

where $\mathbf{x}$ and $\mathbf{y}$ represent two different images and α, β, and γ are the positive constants. The first term $l(\mathbf{x}, \mathbf{y})$ quantifies the similarity of the mean luminance between the two images. Local means μ_x and μ_y are computed by averaging the pixel values within a local window around each pixel. The second term $c(\mathbf{x}, \mathbf{y})$ assesses the similarity of contrast in images $\mathbf{x}$ and $\mathbf{y}$, measured by the standard deviation of image σ_i. Like luminance, contrast is considered within a local context, emphasizing the importance of local variations. The final term $s(\mathbf{x}, \mathbf{y})$ compares the structure by evaluating the correlation coefficient between the two images, with σ_{xy} denoting the covariance between $\mathbf{x}$ and $\mathbf{y}$. This term considers the spatial relationship between corresponding pixels in the local windows of the two images. Positive constants c_1, c_2, and c_3 are introduced to avoid

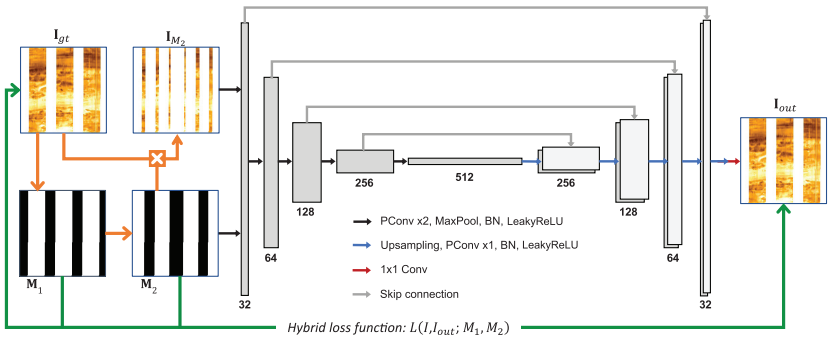

Figure 6. The proposed network architecture. We generate a vertical mask $\mathbf{M}_2$ from the original mask $\mathbf{M}_1$. We then multiply $\mathbf{I}_{gt}$ and $\mathbf{M}_2$ to obtain the masked image $\mathbf{I}_{M_2}$. Feeding $\mathbf{I}_{M_2}$ and $\mathbf{M}_2$ to our network, we obtain the mask-filled image $\mathbf{I}_{\text{out}}$. Here, PConv is the partial convolution layer. Note that we use partial convolution layers instead of the conventional 3×3 convolution layers.

division by zero and to stabilize the metric. These values, set as $c_1 = 0.01$ and $c_2 = c_3 \times 2 = 0.03$, are chosen empirically to balance mathematical stability and perceptual relevance. The MS-SSIM is defined as

$$\text{MS-SSIM}(\mathbf{x},\mathbf{y}) = [l_M(\mathbf{x},\mathbf{y})]^{\alpha_M} \cdot \prod_{j=1}^{M} [c_j(\mathbf{x},\mathbf{y})]^{\beta_j} [s_j(\mathbf{x},\mathbf{y})]^{\gamma_j}, \tag{7}$$

where α_M, β_j, and γ_j are the constant exponents used to adjust the relative importance of the three terms in equation 6. In general, we set $\alpha = \beta = \gamma = [0.0448; 0.2856; 0.3001; 0.23630.1333]$. Note that the range of MS-SSIM is not limited to [0,1] because a negative covariance σ_{xy} could lead to a negative $(\mathbf{x},\mathbf{y})$. To transform it into a loss function, we normalize it to [0,1] by calculating MS-SSIM = (MS-SSIM + 1)/2. The closer the MS-SSIM value is to one, the more similar the two images are.

We combine all three aforementioned loss functions to construct the total loss L_{total} as follows:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\ell_1} + \lambda_2 \mathcal{L}_{\text{perceptual}} + \lambda_3 \mathcal{L}_{\text{MS-SSIM}}, \tag{8}$$

where

$$\mathcal{L}_{\text{MS-SSIM}} = 1 - \text{MS-SSIM}(\mathbf{I}_{\text{comp}} \cdot \mathbf{M}_1, \mathbf{I}_{gt} \cdot \mathbf{M}_1). \tag{9}$$

We empirically set $\lambda_1 = 2$, $\lambda_2 = 3$, and $\lambda_3 = 5$ for balancing these individual loss functions.

Training details

To mitigate the impact of outliers, we normalize every pixel value in a borehole image by subtracting the mean and dividing by the standard deviation, resulting in a normalized borehole image with zero mean and unit standard deviation. The formula is as follows:

$$\mathbf{x}^* = \frac{\mathbf{x} - \mu}{\sigma}, \tag{10}$$

where $\mathbf{x}^*$ is the normalized borehole image, $\mathbf{x}$ is the original borehole image, and μ and σ are the mean and standard deviation of $\mathbf{x}$, respectively.

To construct the training and validation data sets, we load all the borehole images into memory and allocate the first 10% of each image to the validation data set and the remaining 90% to the training data set. Then, to further diversify the data sets, we randomly crop 1600 images for training and 160 for validating at each epoch (Figure 5). In this way, we randomly generate training samples while training the network, which ensures the samples fed into the network at each training epoch are all different.

The proposed network is implemented in TensorFlow, and the training process uses 256×256 images with random vertical masks. We use an Adam optimizer with a default learning rate of 0.001 to update the model parameters. To prevent overfitting, we halt the training when the validation loss ceases to decrease. With a batch size of 16, the network is trained for 200 epochs. Training the CNN on a single NVIDIA Tesla V100 (32 G) graphics processing unit takes approximately 2 h. Figure 7 shows the history of the loss function, wherein the blue and orange curves in Figure 7a represent the total training and validation losses, respectively; Figure 7b shows the history of the three individual losses in the test sets. Remarkably, our trained model can process a 256×256 borehole image in only 1.3 ms.

RESULTS

In this section, we apply our trained CNN to fill gaps in field borehole images that are not included in the training data set. Before feeding images to the CNN, we apply mean-variance normalization (equation 10) to each test image as we did for the training images. Note that the data size does not have to be 256, but it needs to be divisible by 2^t, where $t = 5$ is the number of downsampling implemented in our network architecture.

We feed borehole images with the shape of 2048×256 to our trained CNN model and obtain restored full borehole images. To demonstrate the feasibility of our CNN-based method, we select two field examples with distinct sedimentary patterns. The first field

Table 3. Evaluation metrics on the test data set.

	Our method	CIFLog	DGP	DIP
SSIM↑	**0.915**	0.835	0.850	0.856
PSNR↑	**0.853**	0.731	0.675	0.608
EVS↑	**0.914**	0.757	0.750	0.738
MAE↓	**0.023**	0.033	0.034	0.038
MSE↓	**0.003**	0.007	0.007	0.009
MDAE↓	**0.018**	0.023	0.026	0.029

Bold highlights the best result.

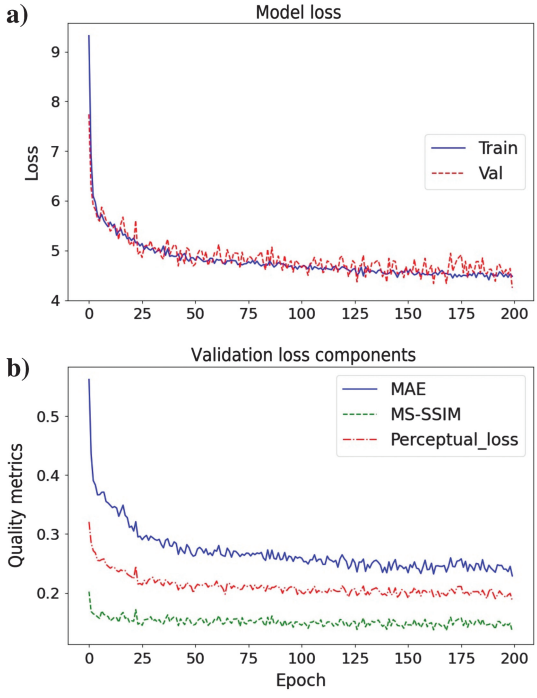

Figure 7. (a) Curves of the total loss function at each epoch during training and testing. (b) Curves of every loss function at each epoch during testing. Here, MAE means the ℓ_1 loss.

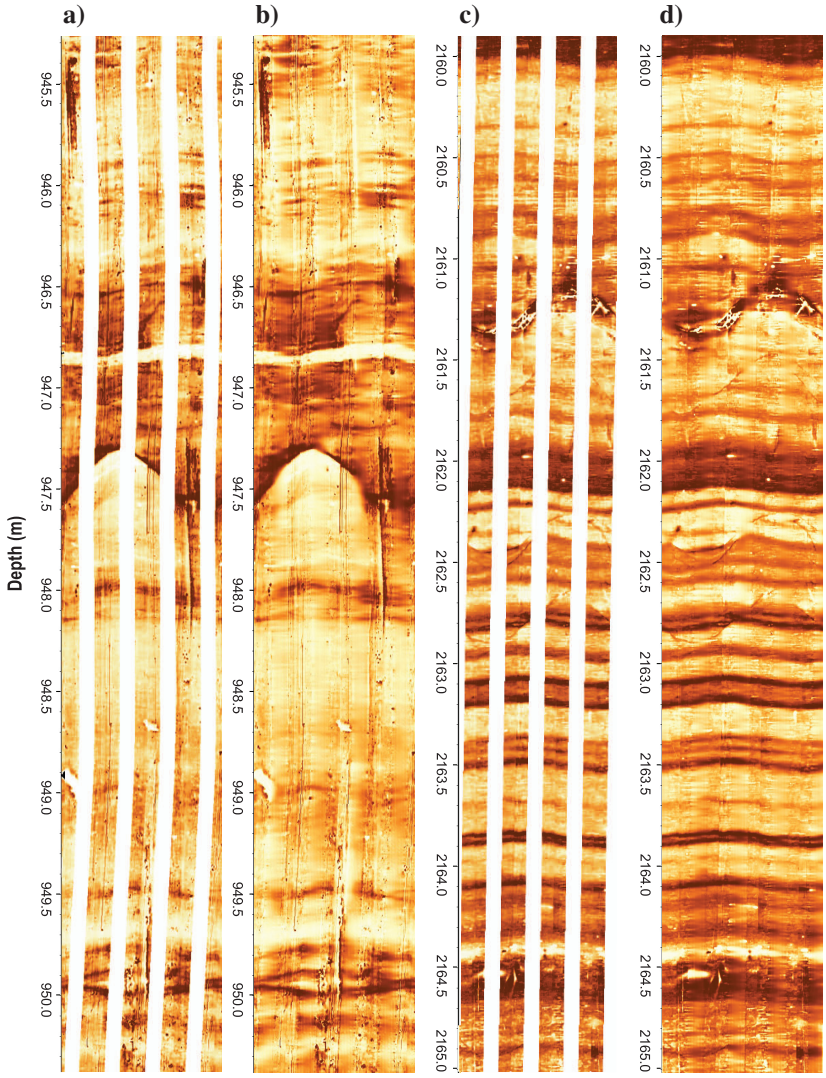

Figure 8. Results on test images. (a and c) Original borehole images from the Nankai Trough and offshore Shimokita Peninsula, respectively. (b and d) Corresponding full borehole images filled by our well-trained model.

example, shown in Figure 8a, is from the Nankai Trough. It exhibits partially visible high-angle fractures, whereas the resulting image after restoration in Figure 8b shows fracture-clear textures. This example illustrates the ability of our CNN method to capture large-scale structures and high-angle fracture shapes. The second field example, shown in Figure 8c, is from the offshore Shimokita Peninsula and comprises parallel-laminated sandstone beds. The reconstructed borehole image shown in Figure 8d represents the well-reconstructed sedimentary features of parallel-laminated sandstone. It effectively captures linearly connected edges and curvilinear structures. Overall, as shown in Figure 8, the resulting full borehole images exhibit visually high quality and uphold the essential attributes and characteristics of the original imagery. The reconstructed gaps are laterally consistent with the features in the surrounding known areas. Such results greatly facilitate geologic interpretation and formation evaluation.

To comprehensively assess our method, we conduct a quantitative comparison with three different methods: two deep-learning-based methods, called DIP (Ulyanov et al., 2018) and deep generative prior (DGP) (Pan et al., 2021), and one traditional interpolation-based method of CIFLog developed by the Research Institute of Petroleum Exploration & Development. Because the output ranges of these methods vary, we normalize each predicted and labeled image to the range $[0, 1]$ with a size of 256×256 for fair comparison. In one 256×256 image, we set the number of iterations to 2000 for DIP. We use the publicly available pretrained model for DGP while maintaining the initial inpainting configuration.

To gain a visual understanding of the filling effects, we display the results of the four methods in Figure 9. Feeding the inputs (Figure 9a) into each method, we obtain different full-bore

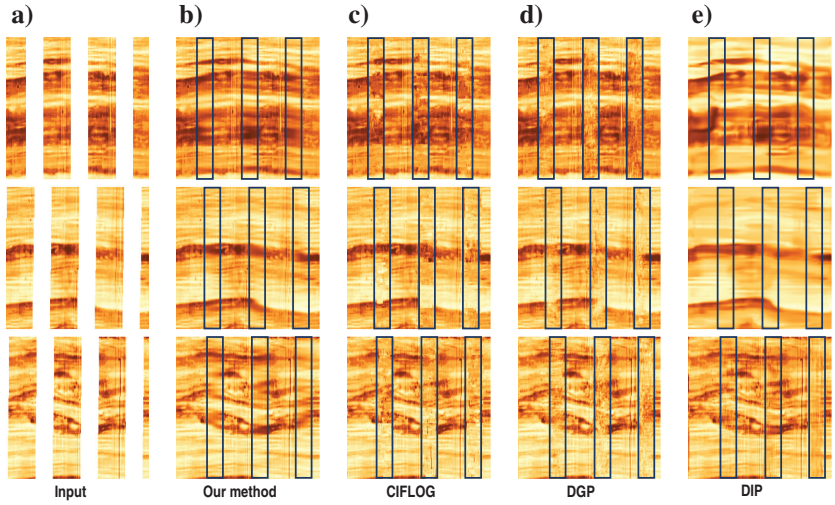

Figure 9. Comparison of different methods. (a) The original borehole image and (b–e) full borehole images restored by our method, the CIFLOG software, the DGP method (Pan et al., 2021), and the DIP method (Ulyanov et al., 2018), respectively.

DOI:10.1190/geo2022-0344.1

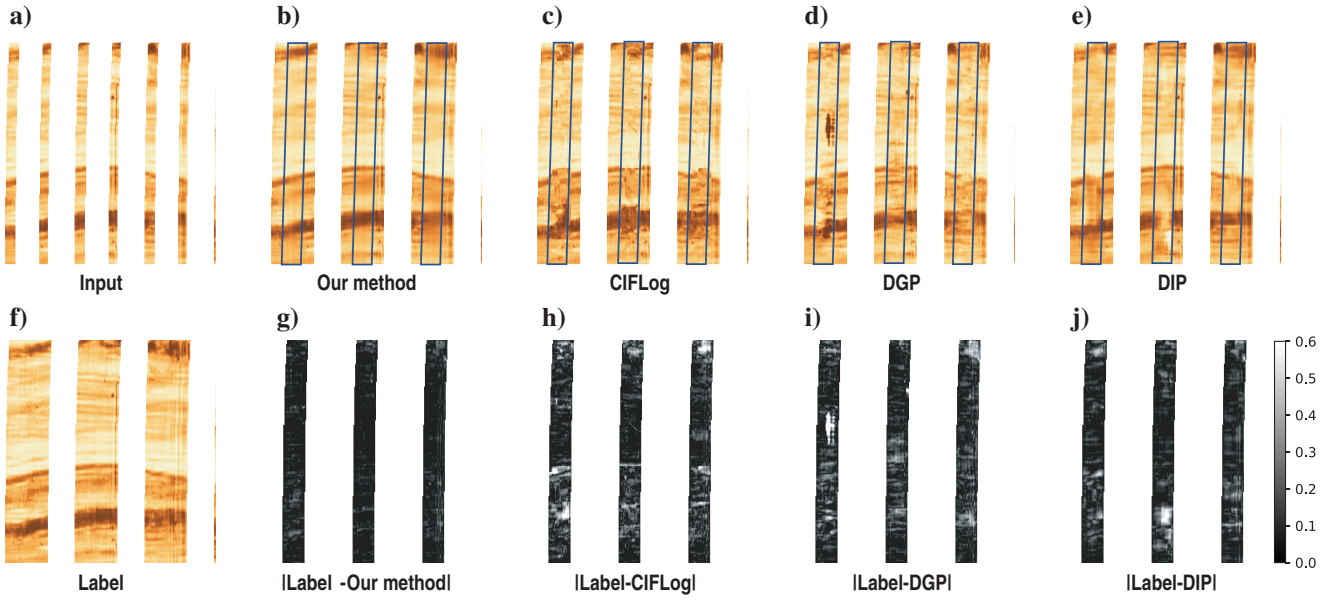

Figure 10. Comparison of different methods. The first row shows (a) the input masked image and (b–e) the mask-filled images using four different methods. The second row shows (f) the label image and (g–j) the pixel-by-pixel absolute error images between the label and images filled by different methods. The colorbar represents the absolute value of the error between the method's results and the labels, with lighter colors indicating a larger error.

images, as shown in Figure 9b–9d. We observe that CIFLog (Figure 9c) and DGP (Figure 9d) occasionally introduce unreasonable patterns to fill the gaps, whereas DIP (Figure 9e) fails to restore clear texture features in certain areas. In contrast, our method (Figure 9b) recovers reasonable texture features and maintains lateral consistency with the surrounding known features. Furthermore, our method demonstrates remarkable efficiency by filling 100 images with a size of 256×256 in only 1.3 s, whereas the other methods require several minutes to process a single image.

We now quantitatively compare these methods. We use the strategy of generating training sets to generate verification sets (Figure 10a and 10f), randomly masking out some vertical parts of the known data and applying each method to fill the masked regions. Figure 10 shows the comparison results. The first row shows the different filled images, and the second row shows the absolute error between different filled images and label images (Figure 10f), respectively. The brighter the color is, the greater the error. Note that we only calculate the error in the masked area. As shown in Figure 10, the reconstruction by CIFLog (Figure 10c and 10h) and DGP (Figure 10d and 10i) suffers from distorted structures and inconsistencies near the middle regions, whereas the filling by DIP (Figure 10e and 10j) contains large artifacts and blurry textures. In contrast, the image restored by our method (Figure 10b) is more reliable and shows fewer artifacts and the least error (Figure 10g), resembling the ground truth (Figure 10f) closely.

To provide a more objective evaluation, we use several metrics, including SSIM, peak signal-to-noise ratio (PS/N), explained variance score (EVS), MAE, mean-squared error (MSE), and median absolute error (MDAE). Table 3 shows a comparison of these metrics. Our method achieves the best performance based on every metric, which proves the superiority of our method compared with the methods of CIFLog, DGP, and DIP. Note that we evaluate these metrics only in the masked regions filled by these methods, and the test images were not included in the training data set.

CONCLUSION

In this paper, we propose a self-supervised learning workflow to generate training data by using random masks, effectively addressing the issue of missing labels of full borehole images. We apply an end-to-end deep network to effectively fill gaps in borehole images without introducing any unreasonable features, facilitating image interpretation, subsequent fault identification, and fracture analysis. Our network, based on a U-Net variant supplemented with partial convolutional layers, enables real-time processing, completing 100 images of size 256×256 pixels in only 1.3 s. Through multiple real examples and quantitative comparisons, we demonstrate the significant superiority of our method over conventional and previous deep-learning approaches.

Some limitations still remain in our CNN-based method. First, although our random masking and partial convolution approach allows us to directly use incomplete borehole images for training the network, and our strategy of randomly cropping training samples during training can reduce the repetition of samples between different training epochs, the limited number of real borehole images available to us severely limits the overall richness and diversity of the training set. This actually restricts the generalization ability of the trained network on a wider variety of borehole images. Generating synthetic borehole images might be a feasible solution to address the lack of training samples. Second, although our method produces restored images of better quality compared with other methods, the restored results in some areas may be slightly blurry and lack details and resolution. This limitation may be attributed to the network's capacity, indicating that there is still room for improvement in network model optimization in future studies. Finally, the completion of missing borehole images should not rely solely on surrounding image features; geologic background knowledge of the corresponding region is also crucial. Future deep-learning-based completion methods could consider designing prompt engines to

introduce expert geologic prior knowledge constraints to improve the reliability of completion results.

ACKNOWLEDGMENTS

We thank the USTC supercomputing center for providing computational resources for this work. This research was supported by the National Science Foundation of China under grant no. 42050104.

DATA AND MATERIALS AVAILABILITY

Data associated with this research are available and can be accessed via the following URL: https://mlp.ldeo.columbia.edu/logdb/scientific_ocean_drilling.

REFERENCES

Assous, S., and P. Elkington, 2018, Shearlets and sparse representation for microresistivity borehole image inpainting: Geophysics, **83**, no. 1, D17–D25, doi: 10.1190/geo2017-0279.1.

Assous, S., P. Elkington, and J. Whetton, 2013, Microresistivity borehole image inpainting: Geophysics, **79**, no. 2, D31–D39, doi: 10.1190/geo2013-0188.1.

Cai, N., Z. Su, Z. Lin, H. Wang, Z. Yang, and B. W.-K. Ling, 2017, Blind inpainting using the fully convolutional neural network: The Visual Computer, **33**, 249–261, doi: 10.1007/s00371-015-1190-z.

Claerbout, J. F., 1992, Earth soundings analysis: Processing versus inversion: Blackwell Scientific Publications 6.

Columbia University, Lamont-Doherty Earth Observatory, https://mlp.ldeo.columbia.edu/logdb/scientific_ocean_drilling/, accessed 6 April 2022.

Dimitrakopoulos, R. G., H. Mustapha, and E. Gloaguen, 2009, High-order statistics of spatial random fields: Exploring spatial cumulants for modeling complex non-Gaussian and non-linear phenomena: Mathematical Geosciences, **42**, 65–99, doi: 10.1007/s11004-009-9258-9.

Du, C., Q. Xing, J. Zhang, J. Wang, B. Liu, and Y. Wang, 2020, Inpainting electrical logging images based on deep CNN with attention mechanisms: IEEE Symposium Series on Computational Intelligence, 607–610.

Hurley, N. F., and T. Zhang, 2011, Method to generate full-bore images using borehole images and multipoint statistics: SPE Reservoir Evaluation and Engineering, **14**, 204–214, doi: 10.2118/120671-PA.

Inagaki, F., K.-U. Hinrichs, Y. Kubo, and the IODP Expedition 337 Scientists, 2016, IODP Expedition 337: Deep coalbed biosphere off Shimokita: Microbial processes and hydrocarbon system associated with deeply buried coalbed in the ocean: Scientific Drilling, **21**, 17–28, doi: 10.5194/sd-21-17-2016.

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

Kaur, H., N. Pham, and S. Fomel, 2019, Seismic data interpolation using CycleGAN: 89th Annual International Meeting, SEG, Expanded Abstracts, 2202–2206, doi: 10.1190/segam2019-3207424.1.

Kaur, H., N. Pham, and S. Fomel, 2021, Seismic data interpolation using deep learning with generative adversarial networks: Geophysical Prospecting, **69**, 307–326, doi: 10.1111/1365-2478.13055.

Liu, G., F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, 2018, Image inpainting for irregular holes using partial convolutions: Proceedings of the European Conference on Computer Vision, 85–100.

Lofts, J., and L. Bourke, 1999, The recognition of artefacts from acoustic and resistivity borehole imaging devices: Geological Society, London, Special Publications 159, 59–76.

Luthi, S., 2001, Geological well logs: Their use in reservoir modeling: Springer Science & Business Media.

Mandelli, S., V. Lipari, P. Bestagini, and S. Tubaro, 2019, Interpolation and denoising of seismic data using convolutional neural networks: arXiv preprint, doi: 10.48550/arXiv.1901.07927.

Mariethoz, G., and P. Renard, 2010, Reconstruction of incomplete data sets or images using direct sampling: Mathematical Geosciences, **42**, 245–268, doi: 10.1007/s11004-010-9270-0.

McNeill, L., D. Saffer, T. Byrne, E. Araki, S. Toczko, N. Eguchi, K. Takahashi, and , IODP Expedition 319 Scientists, 2010, IODP Expedition 319, NanTroSEIZE stage 2: First IODP riser drilling operations and observatory installation towards understanding subduction zone seismogenesis: Scientific Drilling, **10**, 4–13, doi: 10.5194/sd-10-4-2010.

Mustapha, H., and R. Dimitrakopoulos, 2010, High-order stochastic simulation of complex spatially distributed natural phenomena: Mathematical Geosciences, **42**, 457–485, doi: 10.1007/s11004-010-9291-8.

Pan, X., X. Zhan, B. Dai, D. Lin, C. C. Loy, and P. Luo, 2021, Exploiting deep generative prior for versatile image restoration and manipulation: IEEE Transactions on Pattern Analysis and Machine Intelligence, **44**, 7474–7489, doi: 10.1109/TPAMI.2021.3115428.

Rider, M. H., 1986, The geological interpretation of well logs: Rider-French Consulting Ltd.

Ronneberger, O., P. Fischer, and T. Brox, 2015, U-Net: Convolutional networks for biomedical image segmentation: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 234–241.

Saad, A. B., Y. Tamaazousti, J. Kherroubi, and A. He, 2020, Where is the fake? Patch-wise supervised GANs for texture inpainting: IEEE International Conference on Image Processing, 568–572.

Schlumberger, 2013, FMI-HD high-definition formation microimager, https://www.slb.com/-/media/files/fe/brochure/fmi-hd-br.ashx, accessed 6 April 2022.

Simonyan, K., and A. Zisserman, 2014, Very deep convolutional networks for large-scale image recognition: arXiv preprint, doi: 10.48550/arXiv.1409.1556.

Slim, M. I., 2007, Borehole-image log interpretation and 3D facies modeling in the Mesaverde group, Greater Natural Buttes field, Uinta Basin, Utah: Ph.D. thesis, Colorado School of Mines.

Soille, P., 1999, Erosion and dilation: Springer Berlin Heidelberg, 49–88.

Ulyanov, D., A. Vedaldi, and V. Lempitsky, 2018, Deep image prior: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 9446–9454.

Wang, Z., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, 2004, Image quality assessment: From error visibility to structural similarity: IEEE Transactions on Image Processing, **13**, 600–612, doi: 10.1109/TIP.2003.819861.

Wang, Z., N. Gao, R. Zeng, X. Du, X. Du, and S. Chen, 2019, A gaps filling method for electrical logging images based on a deep learning model: Well Logging Technology, **43**, 578–582.

Wang, Z., E. Simoncelli, and A. Bovik, 2003, Multiscale structural similarity for image quality assessment: 37th Asilomar Conference on Signals, Systems & Computers, 1398–1402.

Yan, Z., X. Li, M. Li, W. Zuo, and S. Shan, 2018, Shift-Net: Image inpainting via deep feature rearrangement: Proceedings of the European Conference on Computer Vision, 1–17.

Zeng, Y., J. Fu, H. Chao, and B. Guo, 2019, Learning pyramid-context encoder network for high-quality image inpainting: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 1486–1494.

Zhang, T., A. Gelman, and R. Laronga, 2017, Structure- and texture-based fullbore image reconstruction: Mathematical Geosciences, **49**, 195–215, doi: 10.1007/s11004-016-9649-7.

Zhang, T., P. Switzer, and A. Journel, 2006, Filter-based classification of training image patterns for spatial simulation: Mathematical Geology, **38**, 63–80, doi: 10.1007/s11004-005-9004-x.

Biographies and photographs of the authors are not available.