Integrating Materials and Manufacturing Innovation
https://doi.org/10.1007/s40192-025-00402-9

TECHNICAL ARTICLE

Hybrid Algorithm for Filling in Missing Data in Electron Backscatter Diffraction Maps

Emmanuel Atindama[1] · Conor Miller-Lynch[1] · Huston Wilhite[2] · Cody Mattice[3] · Günay Doğan[4] · Prashant Athavale[1]

Received: 29 January 2025 / Accepted: 20 May 2025

Abstract

In this study, we present a novel hybrid algorithm for filling in missing orientation values in electron backscatter diffraction (EBSD) maps, which are instrumental in characterizing microstructures of polycrystalline materials. Traditional exemplar-based and machine-learning methods, which were originally designed for natural images captured with conventional cameras in macroscopic environments, typically perform poorly on EBSD images due to the distinct nature of EBSD data and grain geometries. To address this gap, we adapted a classical exemplar-based inpainting algorithm and a partial convolutional neural network-based method to the unique requirements of EBSD data. Each adapted method, however, presents its own set of strengths and limitations. To overcome these limitations, we propose a novel hybrid inpainting algorithm that integrates modified machine learning and exemplar-based methods. Our approach begins by using a tailored deep-learning model to produce an initial approximation of missing regions, followed by refinement using an adapted exemplar-based algorithm, ensuring the preservation of critical grain boundaries and structural integrity. We validate our method on 10,000 synthetic EBSD images generated with DREAM.3D software and demonstrate that the hybrid algorithm outperforms standalone techniques in both accuracy and visual coherence. This advancement results in high-quality restorations that are more reliable for material analysis. Furthermore, the hybrid approach is adaptable to broader inpainting problems, extending its applicability beyond EBSD maps to other orientation-based datasets.

Keywords EBSD · Orientation maps · Restoration · Inpainting · Exemplar-based · Deep learning

✉ Prashant Athavale
pathaval@clarkson.edu

Emmanuel Atindama
atindaea@clarkson.edu

Conor Miller-Lynch
millerlc@clarkson.edu

Huston Wilhite
wilhiteh@umich.edu

Cody Mattice
cmattice@middlebury.edu

Günay Doğan
gunay.dogan@nist.gov

[1] Department of Mathematics, Clarkson University, 8 Clarkson Avenue, Potsdam, NY 13699, USA

[2] Department of Mathematics, University of Michigan, 2074 East Hall, 530 Church Street, Ann Arbor, MI 48109, USA

[3] Department of Mathematics, Middlebury College, 14 Old Chapel Rd, Middlebury, VT 05753, USA

[4] National Institute of Standards and Technology, 100 Bureau Drive, Gaithersburg, MD 20899, USA

Introduction

Polycrystalline materials comprise crystal grains, which are regions with a homogeneous crystal orientation. There are a variety of experimental methods that can determine these grain orientations, such as electron-backscatter diffraction (EBSD), transmission electron microscopy (TEM) [1], automated crystal orientation mapping within TEM (ACOM-TEM) [2], and X-ray diffraction (XRD) [3]. Additionally, high-energy X-ray diffraction microscopy (HEDM or 3D-XRD) has recently been employed to acquire orientation measurements [4]. The orientation maps generated using these techniques are instrumental in calculating microstructural statistics [5] or in simulating the physics of microstructures [6].

In particular, the EBSD technique obtains such orientation data as Euler angles [7], which constitute the EBSD map of the polycrystalline material, analogous to how a collection of pixels forms a natural image [7]. Despite their extensive

Published online: 17 June 2025

applications and utility, EBSD maps are often imperfect and can have artifacts giving a noisy or incomplete appearance, and this can compromise the accuracy and reliability of the analysis based on the EBSD maps. Noisy and incomplete data could cause incorrect conclusions about the statistical properties of the material. Moreover, if such data is used to support the design of materials, the resulting materials manufactured could miss the intended specifications, and fail in real-life applications.

While our previous work [8] focused on denoising EBSD maps using weighted total variation flow, in the current work, we address the problem of filling the missing orientation data in an EBSD map. The problem of filling a missing region of pixels in an image is referred to as inpainting [9–11] in the image processing community. There are several methods for inpainting. Recent advances in image inpainting have leveraged generative image synthesis models, particularly those based on generative adversarial networks (GANs), diffusion-based models, and Fourier convolutional neural networks (FCNs) [12–15]. However, we emphasize that these methods focus on generating realistic-looking images and require a large amount of training data and computation time. In contrast, we aim to accomplish this with a modest amount of training data and computation time.

Moreover, EBSD maps differ fundamentally from traditional natural images in that they represent spatially structured crystallographic data to which perceptual or semantic cues common in natural image processing do not directly apply. For example, pixel intensities in EBSD maps correspond to crystallographic orientations rather than color or texture, making perceptual loss functions or pre-trained semantic models less suitable.

Two prominent classes of inpainting large regions are exemplar-based methods and machine-learning-based methods. We describe the exemplar-based inpainting algorithm by Criminisi et al. [16] in section "Exemplar-based Inpainting". In section "Neural Network-based Inpainting", we describe a recent inpainting algorithm based on a partial convolutional neural network (PNN) [17]. Both methods, originally developed for natural images captured with conventional cameras, are ill-suited for EBSD maps due to the distinct geometric and structural characteristics of these datasets. Unlike natural images, where pixel values represent intensity levels, EBSD maps encode orientation data as Euler angles. Consequently, traditional similarity measures, such as the sum of squared differences, are ineffective for EBSD maps, as numerical differences in Euler angles do not directly correspond to meaningful discrepancies in orientation. Consequently, adaptations are required to tailor these algorithms to the unique demands of EBSD inpainting. We describe these modifications in section "Modifications to Current Inpainting Approachesfor EBSD Maps". In section "Modified Exemplar-based Inpainting Algorithms (SSδ andSSδEu)", we propose changes to the inpainting algorithm by Criminisi et al. [16] to suit orientation data in EBSD maps. Similarly, in section "Modification to the Partial Convolutional Network for EBSD Maps", we propose changes to the loss function used in the partial convolutional neural network described in [17] to make it suitable for inpainting orientation maps. We find that even after these changes, the two algorithms have shortcomings. In particular, the neural network approach produces results that do not have the piecewise constant appearance of the grains. On the other hand, exemplar-based algorithms fail to generate reasonable grain boundaries in the EBSD data.

We propose a novel hybrid approach in section "The Proposed Hybrid Inpainting Algorithm" to address the shortcomings of these inpainting algorithms. In the proposed approach, we first obtain an approximation of the inpainting using machine learning and then use this approximation to find the best exemplar in the EBSD map. Our approach produces accurate completions of incomplete EBSD maps. Homogeneous values of orientations are preserved within the completed grains, resulting in an almost piecewise constant orientation map of the grains. At the same time, we have better continuation of grain boundaries due to the use of the first-approximation obtained with machine-learning. We show that the proposed hybrid approach provides the best empirical accuracy of inpainting.

Broader Impact of the Work

This manuscript primarily focuses on inpainting grain orientation data obtained through electron backscatter diffraction (EBSD) technology. However, the proposed algorithm is broadly applicable to various data structures containing grain orientations, such as Automated Crystal Orientation Mapping in Transmission Electron Microscopy (ACOM-TEM), X-ray diffraction (XRD), and high-energy X-ray diffraction microscopy (HEDM). Since deep-learning inpainting algorithms depend on effective training sets, they may miss the context of the image being inpainted. Our hybrid approach incorporates this context to enhance deep-learning inpainting results. Thus, even if the data does not pertain to grain orientations, our hybrid algorithm can be adapted to improve deep-learning inpainting by integrating it with other techniques that learn from the given image.

Organization of the Paper

In sections "Exemplar-based Inpainting" and "Neural Network-based Inpainting", we review the exemplar-based inpainting algorithm by Criminisi et al. [16] and the partial convolutional neural network (PNN) inpainting method, respectively. In section "Modifications to Current Inpainting Approaches for EBSD Maps" details

modifications to these algorithms to adapt them for EBSD data. In section "Hybrid Inpainting Approach" introduces a weighted hybrid inpainting approach, characterized by a weight parameter $w \in [0, 1]$, culminating in our proposed inpainting algorithm, which is further elaborated in section "The Proposed Hybrid Inpainting Algorithm". Section "Finding the Optimal Weight, ω, for the Hybrid Approach" discusses the experiments conducted to identify the optimal weight for the hybrid method, while section "Experiments" compares the performance of various inpainting algorithms through extensive experiments. In section "Illustrating Various Patterns of Missing Regions", we explore inpainting results for different patterns of missing regions, and section "Inpainting Real EBSD Maps" demonstrates the application of the proposed approach on real EBSD data. Finally, section "Conclusion" concludes the paper with a summary of findings and their implications.

Existing Inpainting Approaches

There are several inpainting approaches for natural images. One of the simplest inpainting methods is the nearest neighborhood method, which fills missing regions in an image by copying the pixel values from the nearest known pixels. However, this method assumes that the missing regions can be adequately reconstructed by propagating the values of surrounding pixels into the gap. It works well in scenarios where the missing region is small and surrounded by homogeneous textures or colors. However, it might not perform well when large blocks are missing from the image. Despite this drawback, we will use the nearest neighbor as a benchmark for comparing our proposed method.

Variational inpainting methods or inpainting with partial differential equations are the other class of inpainting methods [10]. While these methods work well for filling in missing data surrounded by homogeneous regions, these methods are unsuitable when the missing regions are large compared to the grain size.

The next section briefly describes two inpainting approaches most relevant to the EBSD maps with large missing regions.

Exemplar-based Inpainting

Exemplar-based inpainting methods reconstruct missing regions by copying patches (blocks of pixels) from known areas, matching texture and structure more effectively than the nearest-neighbor method. These methods prioritize areas based on texture continuity, ensuring seamless blending with surrounding content. Unlike nearest-neighbor inpainting, which uses individual pixels, exemplar-based methods preserve finer details and are better suited for complex patterns.

Criminisi et al. [16] is an exemplar-based inpainting algorithm that uses exemplar-based texture synthesis, prioritizing the filling process based on confidence in synthesized pixel values and the importance of preserving image structure. This approach allows for the simultaneous propagation of texture and structure information from existing pixels into missing regions using a single algorithm, avoiding the need for separate mechanisms to handle different image patterns. This algorithm splits the image $f : D \in \mathbb{R}^2 \rightarrow \mathbb{R}$ into two regions: the missing (target) region Ω and its complement, the known (source) region Ω^c. For a given point p on the fill front $\partial\Omega$, which is the boundary between Ω and Ω^c, we define a patch Ψ_p centered on p, and scan the known region of the original image to find a patch Ψ_q that is most similar to Ψ_p with respect to some dissimilarity measure. Refer to Fig. 1 for an illustration showing the regions Ω, Ψ_p and Ψ_q. We then replace the unknown data in Ψ_p with the corresponding known data in Ψ_q, shaded gray in Fig. 1. Criminisi et al. [16] argue that the filling order of the missing pixels significantly affects the quality of the inpainting results. To determine the filling order, Criminisi et al. [16] assign a priority $P(p)$ to the pixel p on $\partial\Omega$. This priority is computed as $P(p) = C(p) \times D(p)$, where $C(p)$ is the confidence and $D(p)$ is the data term. To

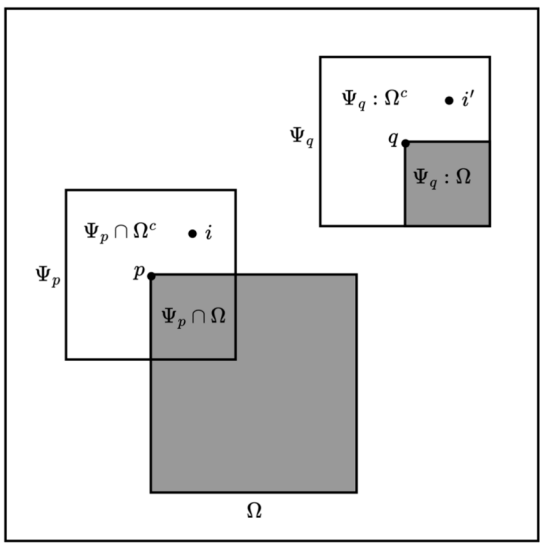


Fig. 1 In this illustration, we show the target region Ω, in which the pixel values are missing and need to be filled in. The region Ψ_p is the neighborhood of the point p on the fill front. The region Ψ_q is the region in Ω^c that we compare with the region Ψ_p. The locations i and i' are the corresponding locations in the regions Ψ_p and Ψ_q, respectively. The shaded region in Ψ_q is the region that corresponds to the missing region in Ψ_p, and we denote this region with the notation $\Psi_q : \Omega$. Similarly, the unshaded region in Ψ_q is the region that corresponds to the known region in Ψ_p, and we will denote this region with the notation $\Psi_q : \Omega^c$

initialize the algorithm, we set the confidence function $C(i) = 0$ for all locations i in the missing region, i.e., for all $i \in \Omega$, and $C(i) = 1$ for all $i \in \Omega^c$. The confidence terms are updated as $C(p) = \left(\sum_{i \in \Psi_p \cap \Omega^c} C(i)\right) \Big/ |\Psi_p|$ as missing pixel values are filled in. The confidence term $C(p)$ is a measure of the amount of reliable information surrounding the pixel p [16].

The data term is defined as $D(p) = |\nabla f_p^{\perp} \cdot \hat{\boldsymbol{n}}_p|$, which encourages continuation of edges with $\nabla f_p^{\perp}$ as the isophote direction at p (orthogonal to the image gradient ∇f) and $\hat{\boldsymbol{n}}_p$ as a unit normal to the fill front, $\partial\Omega$. The algorithm then selects a patch Ψ_q that minimizes some distance $d_{\Omega^c}(\Psi_p, \Psi_q)$ between the known pixels in Ψ_p, i.e. $\Psi_p \cap \Omega^c$ and the corresponding pixels in Ψ_q which we will denote by $\Psi_q : \Omega^c$. For reference, in Fig. 1, the region $\Psi_q : \Omega^c$ is depicted as the *unshaded* part of Ψ_q, and the $\Psi_q : \Omega$ is the *shaded* part of Ψ_q. Criminisi et al. [16] use the distance:

$$d_{\Omega^c}(\Psi_p, \Psi_q) = SSD(\Psi_p \cap \Omega^c, \Psi_q : \Omega^c), \tag{1}$$

which is the sum of squares of the difference between the pixel values of the regions Ψ_p and Ψ_q.

Once the best match Ψ_q to Ψ_p is found, based on the distance (1), the algorithm replaces the 'unknown' data in Ψ_p, i.e., in $\Psi_p \cap \Omega$ with the 'corresponding' pixels in Ψ_q i.e. $\Psi_q : \Omega$ (the shaded part of Ψ_q in Fig. 1).

After each iteration, the algorithm updates the target region Ω, source region Ω^c, and the confidence values. This process is then iterated on the highest priority patch until the entire target region is inpainted.

Neural Network-based Inpainting

An alternative inpainting approach leverages neural networks, offering a powerful tool for handling complex missing data patterns. In this work, we build upon and adapt the method proposed by Liu et al. [17], which introduced a partial convolutional autoencoder based on the U-Net architecture [18] for inpainting irregular holes in images. The following section provides a brief overview of this method and its relevance to our work.

The approach by Liu et al. [17] adapts the standard convolutional layer [19] to manage inputs with missing data by selectively performing convolutions on valid pixels only. The architecture uses a binary mask to indicate the missing region Ω and its complement Ω^c, the known region. This mask differentiates between known pixels, marked with a value of 1, and missing pixels, marked with a value of 0. Consequently, the architecture requires the image to be inpainted and its corresponding binary mask as inputs.

The U-Net architecture in [17] is formed with encoder layers followed by a cascade of decoder layers as in [18] with some modifications to support inpainting. The encoder consists of a sequence of downsampling operations, while the decoder comprises the same number of upsampling operations. This structured approach facilitates deep feature extraction and subsequent reconstruction of the inpainted image. The algorithm initializes the encoder by defining the dimensions and kernel size for partial convolutions. The partial convolutional layer then processes the input image and mask, outputting a convolved image and an updated mask. The batch normalization step [20, 21] normalizes the mean and standard deviation of each mini-batch, streamlining the training process by maintaining a consistent distribution of layer inputs throughout the network. Applying the Rectified Linear Unit (ReLU) [22] introduces necessary nonlinearity, enabling the network to capture complex data patterns.

The decoder layer further refines the reconstruction. It upsamples the encoder's features using nearest-neighbor interpolation and then concatenates them with the corresponding features from the encoder. This process reintegrates context and aids in accurately reconstructing the missing image regions. A subsequent partial convolution integrates information from both the decoder and encoder, ensuring comprehensive feature reconstruction. Following the batch normalization, a Leaky ReLU activation function processes the output [22–24], culminating in the reconstructed image features that merge contributions from both network halves.

In summary, this approach ensures effective inpainting by maintaining the integrity of known pixels while accurately reconstructing missing areas. The encoder captures and compresses image features, which the decoder then expands, employing contextual information from the encoder to achieve inpainting.

Modifications to Current Inpainting Approaches for EBSD Maps

Exemplar-based and machine-learning algorithms, while effective for natural image inpainting, face significant limitations when applied to EBSD maps due to the fundamental differences in data characteristics. Exemplar-based methods work by replicating patterns based on pixel intensity similarities, but this approach does not work well for EBSD data, where the values represent orientations as Euler angles. Traditional similarity measures, such as the sum of squared differences, are not suitable for capturing similarities between patches represented by Euler angles.

Machine-learning approaches, designed for smooth transitions in natural images, struggle with the piecewise constant structures and sharp grain boundaries of EBSD maps. They often fail to preserve these critical features, leading to inaccurate reconstructions. To address these

challenges, exemplar-based and machine-learning-based methods require an orientation-based approach. The following sections explore these necessary adaptations for effective EBSD inpainting.

Modified Exemplar-based Inpainting Algorithms (*SSδ* and *SSδEu*)

Criminisi et al. [16] proposed the inpainting algorithm for images with pixel values representing gray-scale intensity at that location. However, in an EBSD map, each point represents orientation angles.

In Fig. 2a, we see a typical EBSD map displayed using the IPF-z color scheme, and Fig. 2b shows the same map with de la Vallée Poussin noise of $b = 4$, and Fig. 2c shows the noisy map with a missing region. If we implement the classical Criminisi algorithm with the sum of squares of differences of Euler angles as the dissimilarity measure, we obtain the result shown in Fig. 3a.

The issue with employing the classical implementation of the Criminisi algorithm for inpainting EBSD maps lies in the fact that each data point represents an orientation. Hence, to account for this, we modified the Criminisi dissimilarity measure to be an appropriate distance measure between the orientation angles. To this effect, we note that if Ψ_p^i is the orientation at the location i in Ψ_p, we denote the corresponding rotation matrix by $R(\Psi_p^i)$. We denote the location corresponding to i in the region Ψ_q by i' as shown in the schematic Fig. 1. Note that as the location i in Ψ_p changes, the location i' changes accordingly. Similarly, $R(\Psi_q^{i'})$ denotes the rotation matrix for the corresponding orientation $\Psi_q^{i'}$. With this notation, the misorientation

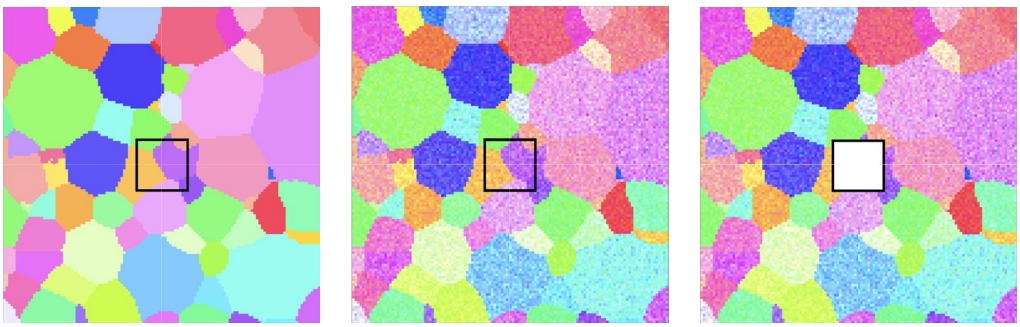


Fig. 2 Clean, noisy EBSD maps, and missing region used for demonstration. **a** IPF-z map of clean EBSD map generated using the Dream.3D software. **b** IPF-z map of original noisy EBSD map with de la Vallée Poussin noise of $b = 4$ **c** missing region of size 20×20 represented as a white block. The goal of inpainting algorithms is to take the map in (**c**) as an input and obtain an inpainting result as close as possible to the map in (**b**)

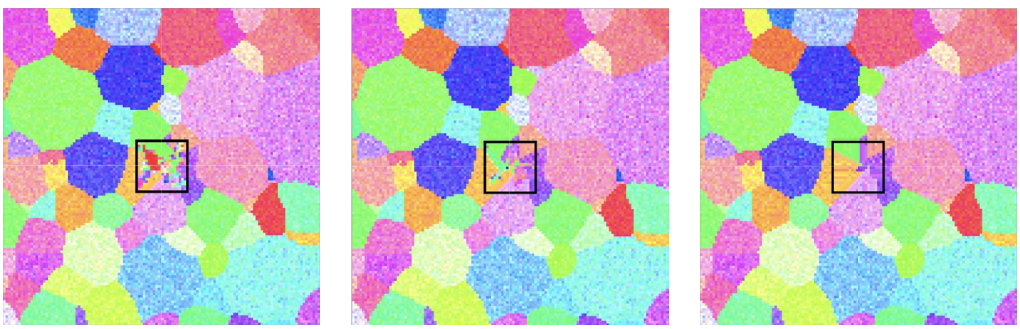


Fig. 3 Results of the comparison of Criminisi et al. [16], and its modifications. Inpainting results for the EBSD map shown in Fig. 2c. **a** shows results with the original Criminisi et al. [16] algorithm, **b** shows the results using the sum of squares of disorientations (*SSδ*) as the dissimilarity measure shown in Eq. (2), **c** depicts the results using the sum of squares of disorientations as the dissimilarity measure with the Euclidean distance penalty (*SSδEu*) as delineated in Eq. (3). Compare these results with the EBSD map in Fig. 2b

between the angles corresponding to the orientations at Ψ_p^i and $\Psi_q^{i'}$ can be written as $\delta(\Psi_p^i, \Psi_q^{i'}) = \arccos\left(\frac{\text{Trace}(R(\Psi_p^i)R(\Psi_q^{i'})^T) - 1}{2}\right)$, where $R(\Psi_q^{i'})^T$ denotes the transpose of the matrix $R(\Psi_q^{i'})$ [25]. With these considerations, we modify the dissimilarity measure to the following:

$$d_{\Omega^c}(\Psi_p, \Psi_q) = \sum_{i \in \Psi_p \cap \Omega^c} \delta^2(\Psi_p^i, \Psi_q^{i'}) \tag{2}$$

We will hereafter refer to the exemplar-based method that utilizes the similarity measure detailed in Eq. (2) as the *SSδ algorithm*.

Furthermore, the original Criminisi algorithm [16] is suitable for 'natural' images, which typically contain repeating patterns. For such images, searching for the matching patch in locations distant from the missing region is reasonable. However, EBSD maps consist of grains that may not have repetitions in distant locations of the map. Hence, we are likely to find a matching patch spatially close to the location of the missing region. We added the Euclidean distance to the dissimilarity measure to account for this. In conclusion, we propose to use the following dissimilarity measure between the region Ψ_p and Ψ_q for the EBSD inpainting when the data in Ω is missing:

$$d_{\Omega^c}(\Psi_p, \Psi_q) = \sum_{i \in \Psi_p \cap \Omega^c} \delta^2(\Psi_p^i, \Psi_q^{i'}) + \lambda \| \overline{p\,q} \|_{\ell^2}, \tag{3}$$

where the summation is over locations corresponding to all 'known' orientations in Ψ_p. The term $\| \overline{p\,q} \|_{\ell^2}$ is the Euclidean distance between the locations p and q. We will refer to the exemplar-based inpainting employing the dissimilarity measure in Eq. (3) as the *SSδEu* algorithm. We observed that the penalty parameter, λ applied to the Euclidean distance, demonstrated robustness across all experiments. Consequently, λ was fixed at a value of 1 for all experiments.

Comparison of the Three Exemplar-based Algorithms

In this section we present comparison of the *SSδ* and *SSδEu* algorithms with the original Criminisi algorithm using the paired t–test [26]. For this analysis, we selected 16 randomly chosen EBSD maps simulated using the Dream3D software [27, 28], subjected to various levels of de la Vallée Poussin noise [8, 29], with half-widths $b = 0, 2, 4, 6, 8$ degrees. From each image, we removed a 20×20 region.

Subsequently, we inpainted these regions using the three exemplar-based algorithms: (a) the original Criminisi algorithm [16], employing the dissimilarity measure shown in Eq. (1), (b) the modified Criminisi algorithm, using the dissimilarity measure shown in Eq. (2), referred to as the *SSδ* algorithm; and (c) the further modified Criminisi algorithm, incorporating the dissimilarity measure described in Eq. (3), denoted as the *SSδEu* algorithm.

For each specified value of the half-width parameter b, we identified the two algorithms that yielded the lowest mean disorientation errors between the inpainted maps and the original clean (noise-free, complete) maps. We then conducted paired t–tests between these two best-performing algorithms with sample size, $n = 16$ [26].

Figure 2a depicts a clean EBSD map. Then we added a de la Vallée Poussin noise of half-width parameter, $b = 4$. We depict the resulting map in Fig. 2b, c illustrates the same map with a missing region measuring 20×20 pixels. The inpainting results illustrated throughout this section use this map shown in Fig. 2b and c. The goal of the inpainting methods is to obtain an EBSD map as close as possible to the one in Fig. 2b. The result of the three algorithms (for inpainting of the EBSD map shown in Fig. 2c) is illustrated in Fig. 3.

Figure 3a shows the result of the original algorithm proposed by Criminisi et al. [16]. Figure 3b and c depicts the result of the *SSδ* and *SSδEu*, respectively. We encourage readers to compare these results with the map in Figure 2b. Additionally, the outcomes of the paired t–tests, which compare the three variants of the Criminisi algorithms on 16 randomly selected noisy EBSD maps, are presented in Table 1. We note that both *SSδ* and *SSδEu* algorithms perform better than the inpainting by Criminisi et al. [16]. Furthermore, the *SSδEu* is statistically significantly better than the second-best algorithm, *SSδ*, with p-value $< .001$ in every value of b.

Table 1 Reconstruction errors of exemplar-based algorithms with various dissimilarity measures

half-width	Criminisi et al. [16]	*SSδ* algorithm	*SSδEu* algorithm
(b)			
0	1.243	1.143	**0.756**
2	1.471	1.338	**0.430**
4	1.271	1.175	**0.739**
6	1.113	0.955	**0.570**
8	1.094	1.017	**0.582**

Per pixel reconstruction errors (in radians) using various inpainting algorithms for input noise with half-width parameters: 0, 2, 4, 6, and 8. We observed that the *SSδEu* was the best-performing (boldfaced) algorithm and *SSδ* was the second-best-performing algorithm (italicized) for all noise levels examined. We observe that our *SSδEu* is statistically significantly better than the second-best algorithm, *SSδ*, with p-value $< .001$ in every value of b

Modification to the Partial Convolutional Network for EBSD Maps

In this study, we adopt a U-Net architecture [18] with partial convolutional layers [17] for machine learning-based inpainting tasks, following a modified implementation from [30]. In the usual implementation of inpainting, the loss function we aim to minimize during the training is the mean of squared differences between the predicted and original images. In the context of EBSD maps, pixel intensities represent crystallographic orientations rather than color or texture, which makes traditional loss functions less applicable to this type of data. Thus, for EBSD maps, we need to account for the disorientation between the original and inpainted regions. However, incorporating disorientation directly into the loss function during neural network training is computationally intensive and would significantly increase training cost. To balance accuracy and efficiency, we instead incorporate the squared misorientation as a penalty term in the loss function to penalize inaccuracies within the inpainted regions:

$$\mathscr{L}(\Omega_{\text{clean}}, \Omega_{\text{predicted}}) = \sum_{i \in \Omega_{\text{clean}}} \delta^2(\Omega^i_{\text{clean}}, \Omega^{i'}_{\text{predicted}}), \tag{4}$$

where i' is the location in $\Omega_{\text{predicted}}$ corresponding to i. We acknowledge that the loss function based on misorientation does not account for crystallographic symmetries. However, we will use disorientations as a metric for comparing the performance of different inpainting algorithms in section "Finding the Optimal Weight, ω, for the Hybrid Approach". Our implementation of this inpainting is available at [31].

Data Generation: We generated a dataset of 10, 000 images of size 256×256 to train this neural network, consisting of noisy images and their corresponding clean versions. These images were synthesized using DREAM.3D [27] and subsequently processed into CTF files in Oxford Instruments' format [32]. To this effect, we generated EBSD maps with grain sizes following a lognormal probability density function, which is the default for Dream.3D. We chose grain sizes characterized by $\mu = 3$ and $\sigma = 0.3$. Consequently, the average grain diameter is approximately 20 microns (e^{μ}). To include a broad distribution of grain sizes, we allowed large grains with a maximum cutoff of $e^{\mu+6\sigma}$ and excluded small grains using a minimum cutoff of $e^{\mu-4\sigma}$. We initialized a volume of $256 \times 256 \times 30$ voxels with a spatial resolution of 1 micron per voxel and extracted a single slice from this microstructure for analysis. For all other morphological parameters, including the ellipsoidal axis ratios, we adopted the default Dream.3D presets and did not impose any crystallographic texture on the microstructures. In particular, the methodology has not been trained or evaluated on multi-phase materials, particularly morphologies other than equiaxed grains.

The 10, 000 clean images thus generated serve as the ground truth for training our predictive model. The noisy images had de la Vallée Poussin noise, simulated via MTEX [33], to generate noisy counterparts with half-width $b = 0, 2, 4$, and 8 degrees. The training data is publicly available [34].

To further augment the data:

- We removed a 20×20 pixel square from random locations in 80% of the noisy images, varying these locations throughout the training epochs.
- We introduced lines with diverse patterns, including zigzags, in the remaining 20% of the noisy images, with pixel widths of approximately 5 pixels at random slopes, and modified their positions across training epochs.

We chose not to train the model on other shapes, as any other forms of missing regions could be represented as linear combinations of lines and squares.

Training: The training involved predicting clean data within artificially created missing regions of noisy images. To train our neural network-based inpainting model, we deliberately selected the size of the missing region to be comparable to the average grain diameter (in pixels), representing a worst-case scenario. We randomized the location of the missing window across the map to ensure wide-ranging exposure during training. Since the missing region can sometimes cover entire grains, reconstructing them from a single example may be impossible. To address this, we trained the model for about 1,000 iterations (996 epochs). This ensures that the model will recover such grains in other training epochs.

Testing: We created the testing dataset of 1, 000 EBSD maps with similar patterns of missing regions and used it to evaluate the mean average disorientation for the trained model. We have made this dataset publicly available at the following link [35].

Comparing Partial-Convolutional Neural Network (PNN)-based Method and its Modification

We now outline the experiments conducted to evaluate the performance of the inpainting algorithms of the partial-convolutional neural network (PNN) as proposed by Liu et al. [17] and our modified version as described in section "Modification to the Partial Convolutional Network for EBSD Maps". For this comparison, we utilized the dataset that we previously employed to evaluate various exemplar-based methods in section "Modified Exemplar-based Inpainting Algorithms (SSδ and SSδEu)". Consistent with the methodology outlined in section "Modified

Exemplar-based Inpainting Algorithms (SSδ and SSδEu)", we conducted paired t-tests [26] on sixteen randomly selected simulated EBSD maps. The modification to the PNN algorithm incorporates the loss function specified in Eq. (4). Figure 4a shows the application of the inpainting algorithm from [17], and Fig. 4b depicts the result of our modification. Additionally, Table 2 presents the results of paired t-tests conducted to compare the performance of the two inpainting approaches across various noise levels.

The result of our modified PNN algorithm depicted in Fig. 4b is qualitatively better than the outcome in Fig. 4a obtained from the original implementation in by Liu et al. [17]. The paired t-test conducted between the two implementations indicates that our modified PNN significantly outperforms the PNN model described by Liu et al. [17] across all tested noise levels, with a p-value $< .001$. However, comparing with the map in Fig. 2b, we observe that the grain boundaries in Fig. 4b are not well-defined although well-defined grain boundaries are expected in typical polycrystalline orientation maps.

Table 2 Reconstruction errors of the inpainting with partial convolutional neural network (PNN) and its modification

half-width (b)	PNN Liu et al. [17]	modified PNN
0	1.238	0.608
2	1.001	0.448
4	1.121	0.559
6	1.115	0.466
8	1.073	0.515

Per pixel reconstruction errors (in radians) using various inpainting algorithms for input noise with half-width parameters: 0, 2, 4, 6, and 8. The modified PNN was significantly better than the one by Liu et al. [17] with $p-$value $< .001$ for all noise levels

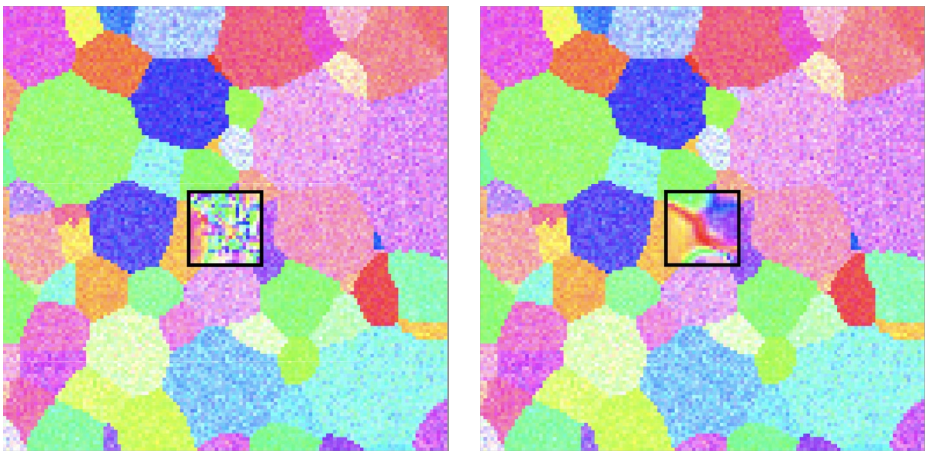
Hybrid Inpainting Approach

In Fig. 4, we observe that the results of inpainting using the machine learning approach generally lack the typical piecewise constant nature of orientation values in polycrystalline EBSD maps, although the mean disorientation errors measuring inpainting accuracy (with respect to the ground truth data) are comparatively low. Conversely, the exemplar-based methods yield the expected piecewise constant appearance but with larger errors, as illustrated in Table 1. This observation leads to the development of a hybrid approach that initially uses machine learning to achieve a preliminary approximation of the inpainting. This approximation then aids in identifying the most suitable exemplar within the EBSD map.

As previously explained in section "Exemplar-based Inpainting" with the help of Fig. 1, when we do not have any information on the missing region, Ω, we cannot compare the part of the potential matching patch Ψ_q corresponding to the data in the missing region. Hence, the classical exemplar-based algorithm uses only the 'known' region in Ψ_p. After the machine learning step, we have some data in the region Ω, which we did not have before. We incorporate this new information into the Criminisi et al. algorithm. To this effect, we modify the dissimilarity measure in Eq. (3) between the region Ψ_p and Ψ_q to the one below

$$d_h(\Psi_p, \Psi_q) = \omega\, d_{\Omega^c}(\Psi_p, \Psi_q) + (1-\omega)\, d_{\Omega}(\Psi_p, \Psi_q), \tag{5}$$

where w is a weight between 0 and 1. The term $d_{\Omega^c}(\Psi_p, \Psi_q)$ is the same as in Eq. (3), and $d_{\Omega}(\Psi_p, \Psi_q)$ is defined similarly but for region Ω which has some data now, thanks to the machine learning approach.



(a) (b)

Fig. 4 Inpainting results for the EBSD maps shown in Fig. 2c. **a** IPF-z map of the result using the partial-convolutional neural network due to Liu et al. [17], **b** IPF-z map of the result using our modified loss function in Eq. (4)

To summarize, we first run the machine learning-based algorithm to obtain an approximate inpainted map, then we use it as input to our modified Criminisi algorithm (SSδEu) with the new weighted dissimilarity measure (5).

Interpretation of the Weight, ω

The weight in Eq. (5) represents the importance given to the known part of the target region Ψ_p, compared to the part of Ψ_p that needs inpainting. The weight of $\omega = 1$ implies that the dissimilarity measure $d_h(\Psi_p, \Psi_q) = d_{\Omega^c}(\Psi_p, \Psi_q)$, meaning that we use only the part from the known region Ω^c to find the best patch. The case of $\omega = 1$ is close to the exemplar-based method. Indeed, the dissimilarity measure is the same as shown in Eq. (3). The weight $\omega = 0$ implies that the dissimilarity measure, $d_h(\Psi_p, \Psi_q) = d_{\Omega}(\Psi_p, \Psi_q)$, i.e., we use only the part of Ψ_p that has been filled with the machine learning algorithm to find the best patch in the exemplar-based inpainting.

It is important to note that the hybrid method at $\omega = 0$ is 'not' the same as the machine-learning-based method. Indeed, the algorithm at weight $\omega = 0$ uses the information filled in by the machine-learning algorithm in the region to be inpainted alone to identify the most suitable patch in the exemplar-based method.

Finding the Optimal Weight, ω, for the Hybrid Approach

To determine the optimal weight, ω, in Eq. 5, we inpainted 1, 000 EBSD maps with a 20×20 patch removed using various algorithms: the original Criminisi algorithm [16], Criminisi-*SSδ*, Criminisi-*SSδEu*, the partial convolutional neural network-based inpainting (PNN) [17], our modification to the PNN (In section "Modification to the Partial Convolutional Network for EBSD Maps"), and the hybrid algorithms (In section "Hybrid Inpainting Approach"). For the hybrid algorithms, we used various weight increments from $\omega = 0$ to $\omega = 1$ in steps of 0.05. We then documented the mean disorientations for these algorithms.

Figure 5 shows the average of the mean disorientation (AMD) errors over 1, 000 randomly selected simulated EBSD maps for various inpainting algorithms, including the hybrid algorithm with weights ranging from $\omega = 0$ to $\omega = 1$ in increments of 0.05.

We observe that the average mean disorientation (AMD) error for the modified PNN was 0.548 radians, whereas the model from Liu et al. produced an AMD error of 1.089 radians. The AMD errors for the [16], *SSδ*, and *SSδEu* were 1.210, 1.117, and 0.607 radians, respectively. Note that the modified PNN shows only a marginal advantage over the *SSδEu*. Moreover, we observe that the error for the proposed algorithm corresponding to $\omega = 0$ was 0.406, the least among all values of ω in the hybrid algorithms.

The Proposed Hybrid Inpainting Algorithm

Following a careful comparative analysis as shown in Fig. 5 we propose to use the hybrid inpainting algorithm with weight $\omega = 0$.

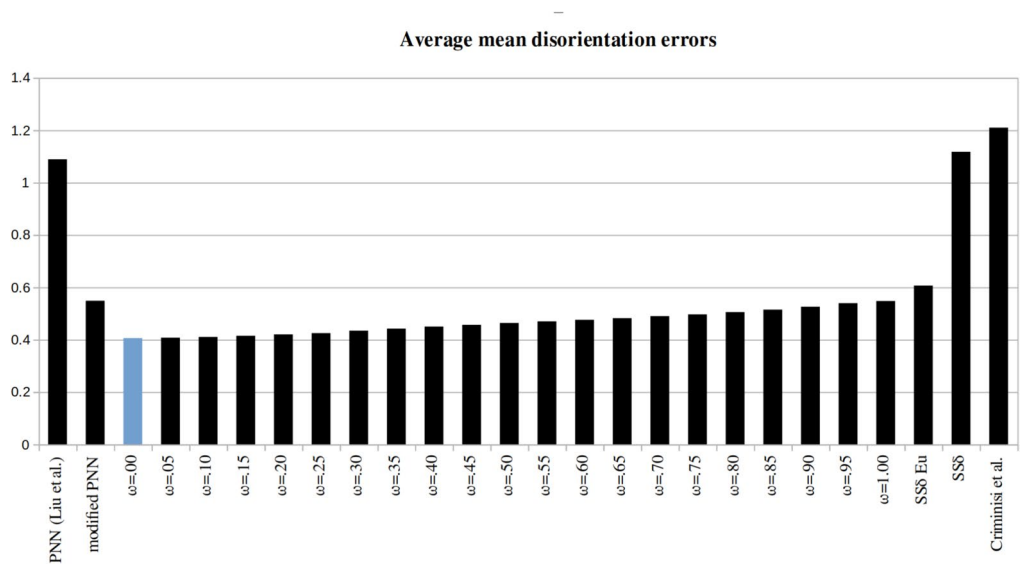


Fig. 5 Average mean disorientation (AMD) errors (radians) after inpainting with various algorithms of 1, 000 randomly selected simulated EBSD maps. The parameter w denotes the weight in Eq. (5), as utilized in the hybrid algorithm described in section "Hybrid Inpainting Approach"

The specific steps of our proposed algorithm are as follows:

1. Acquire a preliminary approximation of the inpainted map using the modified partial convolutional neural network, employing the loss function specified in Eq. (4).
2. Implement the exemplar-based algorithm, utilizing the dissimilarity measure detailed in Eq. (5) with $\omega = 0$. Thus, the proposed dissimilarity measure is defined as:

$$d_h(\Psi_p, \Psi_q) = d_\Omega(\Psi_p, \Psi_q). \tag{6}$$

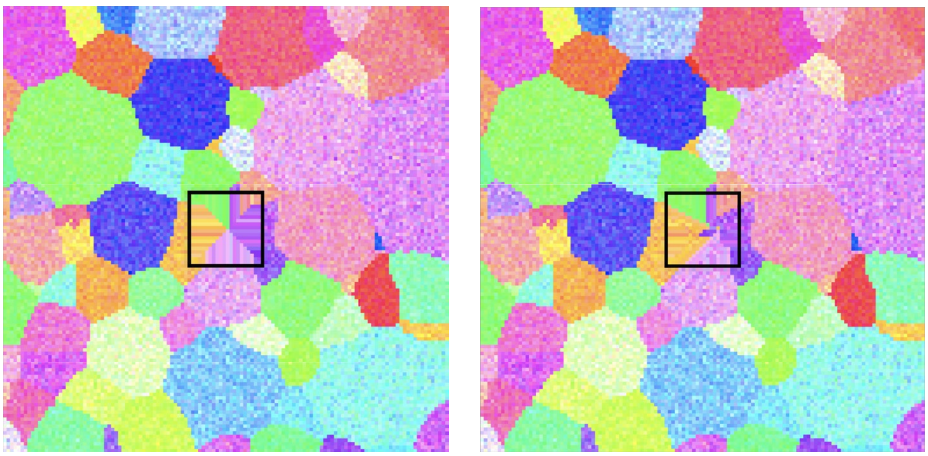
As explained in section "Interpretation of the Weight, ω", setting the weight $\omega = 0$ entails solely using the information filled in by the machine-learning algorithm to identify the most suitable patch in the exemplar-based method. The rationale behind $\omega = 0$ being the optimal weight stems from the fact that it directs the search toward a patch that most closely matches the best approximation of the missing region itself.

Note that the effectiveness of $\omega = 0$ in our case stems from the high quality of the EBSD maps generated by the machine learning step of the algorithm. However, in scenarios where the maps generated by the machine-learning step are less accurate, a nonzero $\omega > 0$ may offer improved performance by using complementary strengths of the exemplar-based and machine learning approaches.

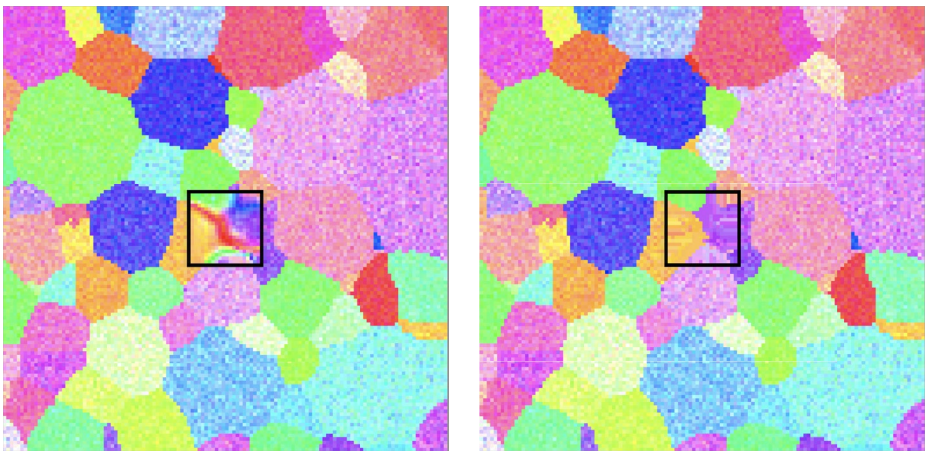
Experiments

Comparing the Proposed Inpainting Algorithm with the Alternatives

This section aims to find whether the proposed hybrid algorithm demonstrates a statistically significant improvement over the alternative methods. Figure 6a



(a) Nearest neighbor (b) modified Criminisi ($SS\delta Eu$)



(c) modified PNN (d) proposed hybrid inpainting

Fig. 6 IPF-z maps of the inpainting results for the EBSD map shown in Fig. 2. **a** using the Dream.3D neighborhood filling filter, **b** results with modified Criminisi algorithm, i.e., the *SSδEu* algorithm, **c** using the modified PNN using the loss function shown in Eq. (4), **d** using the proposed hybrid algorithm

depicts the result of the nearest neighbor inpainting. We use this result as the benchmark for comparison. Figure 6b depicts the outcome of inpainting using the exemplar-based algorithm, *SSδEu*, which utilizes the dissimilarity measure specified in Eq. (3). Figure 6c illustrates the result of inpainting using the modified PNN-based algorithm detailed in section "Modification to the Partial Convolutional Network for EBSD Maps". Finally, Fig. 6d shows the result of the algorithm proposed in section "Hybrid Inpainting Approach". From Fig. 6, we see that the result by the proposed method is qualitatively better than the result obtained using the *SSδEu* and modified PNN. Indeed, the result in Fig. 6c is visually close to the target EBSD map in Fig. 2b.

Following the procedure outlined in section "Comparison of the Three Exemplar-based Algorithms", we selected 16 simulated EBSD maps for each half-width parameter, $b = 0, 2, 4, 6, 8$ degrees, of the de la Vallée Poussin noise and removed a 20×20 region from each. We inpainted these maps using the nearest neighbor inpainting, modified Criminisi (*SSδEu*) algorithm (In section "Modified Exemplar-based Inpainting Algorithms (SSδ and SSδEu)"), the modified PNN (In section "Modification to the Partial Convolutional Network for EBSD Maps"), and the proposed hybrid approach (In section "The Proposed Hybrid Inpainting Algorithm"). We recorded the mean disorientation errors to identify the two best-performing algorithms. Subsequently, we employed the paired *t*-test to compare these two algorithms, utilizing the statistical methodology consistent with sections "Comparison of the Three Exemplar-based Algorithms".

Furthermore, Table 3 presents the outcomes of the statistical analyses. These results show that the proposed algorithm outperforms nearest neighbor inpainting, modified Criminisi, i.e., the *SSδEu* algorithm, and the modified PNN algorithm for all noise levels.

Illustrating Various Patterns of Missing Regions

We tested various algorithms on 20×20 missing regions from randomly selected simulated EBSD maps. However, missing regions can vary significantly in their characteristics. While it is impossible to test the algorithms on every possible type of missing region, it is beneficial to understand how these algorithms perform across a spectrum of missing region types. Recall from section "Modification to the Partial Convolutional Network for EBSD Maps" that the 20% of the training dataset of the modified PNN consisted of diverse types of missing regions, including multiple missing blocks of different shapes, uniformly random missing pixels, missing pixels on grain boundaries, and missing pixels along lines of varying thickness.

Figure 7 illustrates various patterns of missing regions. Specifically, Fig. 7a displays an EBSD map with three blocks missing; (b) features uniformly random missing pixels with 20% of the pixels missing; (c) shows 50% of the data missing from the grain boundaries; and (d) depicts the removal of one vertical and one horizontal line each one pixel thick, as well as a line of nonzero slope five pixels thick.

We present the outputs of various algorithms on these patterns in Fig. 8. The first column of Fig. 8 displays EBSD maps with various missing data patterns. The second column (b) illustrates the results achieved using our *SSδEu* algorithm, as described in section "Modified Exemplar-based Inpainting Algorithms (SSδ and SSδEu)". The third column (c) presents the outcomes using the modified PNN algorithm outlined in section "Modification to the Partial Convolutional Network for EBSD Maps". Finally, the images in column (d) demonstrate the results obtained with the proposed algorithm from section "The Proposed Hybrid Inpainting Algorithm". The reader is encouraged to compare the results in the last column with the target EBSD map in Fig. 2(b). We observe that the proposed method yields

Table 3 Reconstruction errors of exemplar-based algorithm (*SSδEu* from section "Modified Exemplar-based Inpainting Algorithms (SSδ and SSδEu)"), modified PNN (In section "Modification to the Partial Convolutional Network for EBSD Maps"), and the proposed inpainting algorithm from section "The Proposed Hybrid Inpainting Algorithm")

Half-width (***b***)	Nearest neighbor	Modified Criminisi ***SSδEu*** Section "Modified Exemplar-based Inpainting Algorithms (SSδ and SSδEu)"	Modified PNN Section "Modification to the Partial Convolutional Network for EBSD Maps"	Proposed hybrid inpainting Section "The Proposed Hybrid Inpainting Algorithm"	***p***-values
0	0.783	0.756	0.607	**0.528** **	.005
2	0.453	0.429	0.447	**0.230** ***	< .001
4	0.790	0.739	0.559	**0.386** ***	< .001
6	0.613	0.570	0.466	**0.345** ***	< .001
8	0.608	0.582	0.515	**0.408** ***	< .001

Per pixel reconstruction errors (in radians) using various inpainting algorithms for input noise with half-width parameters: 0, 2, 4, 6, and 8. The empirically best denoising error is boldfaced, while the error with the empirically second-best method is in italicized. The stars indicate the *p*-value for the statistically best performance compared with the second-best method. The *p*-values reported are for the paired *t*-test between the best methods and its closest competitor. * $p < 0.1$, ** $p < 0.01$, *** $p < 0.001$

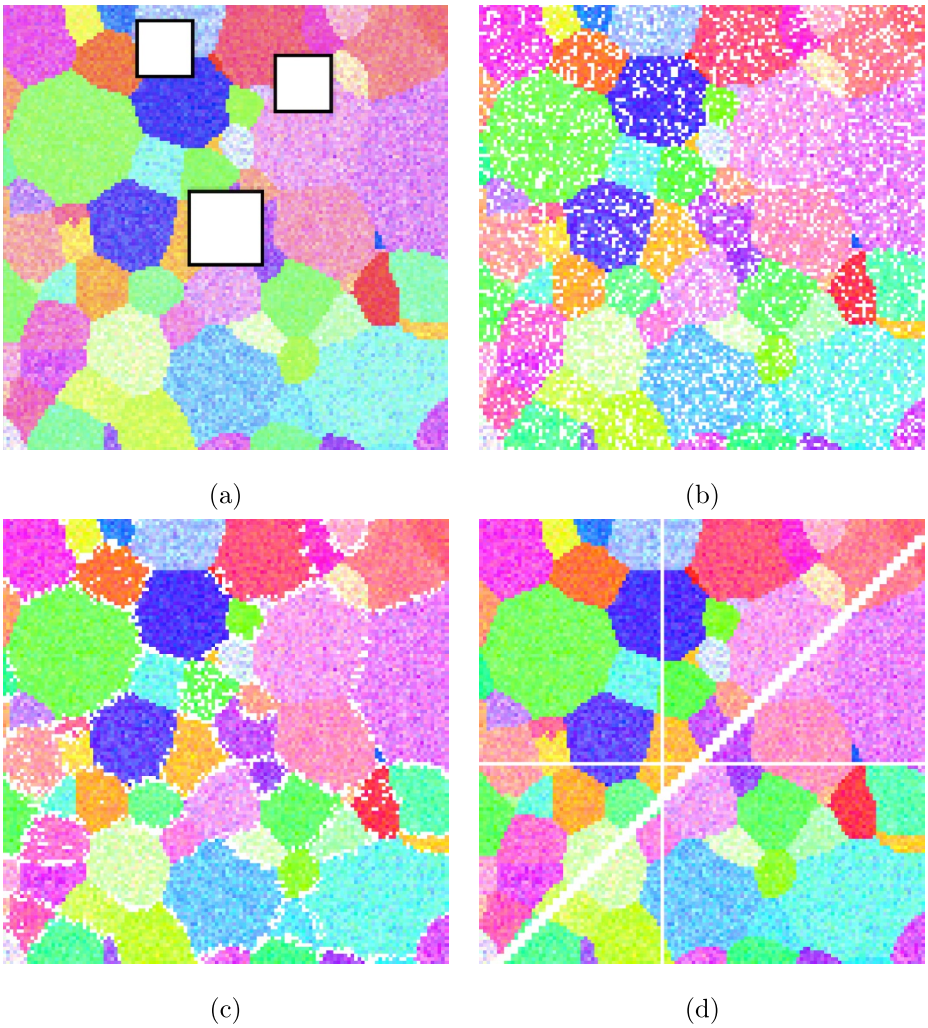


Fig. 7 Various patterns of missing regions. **a** EBSD map with three missing blocks, **b** uniformly distributed random missing pixels, **c** missing pixels on the grain boundaries, **d** missing lines of varying thickness

qualitatively better results than the other methods for all patterns of missing regions.

Inpainting Real EBSD Maps

Thus far, our inpainting experiments have been conducted on simulated EBSD maps. In this section, we extend our investigation to real EBSD maps. Figure 9a shows three real EBSD maps, courtesy of Adam Creuziger from NIST [36]. The missing points scattered around the EBSD maps shown in Fig. 9 were missing data at the time of data acquisition. The second column labeled (b) in Fig. 9 shows the inpainting results with the nearest neighbor inpainting for comparison. We used the nearest neighbor inpainting from Dream.3D software as a benchmark due to its wide use in the community. The last column (c) of Fig. 9c shows the results of the proposed algorithm. For real-world data, precise error quantification is not feasible; hence, the evaluation relies on visual inspection. The inpainting results appear satisfactory based on visual assessment.

In general, we find that the contrast with the results of our inpainting algorithm is more dramatic when an EBSD map contains large regions of missing data, as the majority of missing orientations are away from known orientations that can inform them. Such large missing regions are more naturally filled in using our algorithm. In the EBSD examples of Fig. 9, the missing regions are thin and scattered across, and more critically, most missing orientations have known orientations in their vicinities. Thus, the nearest neighbor inpainting from Dream.3D did a reasonable job filling in the missing orientations. However, these examples do not show the real utility of our algorithm.

To illustrate the advantage of our algorithm over the nearest neighbor filter, we artificially removed two squares from the first real image shown in Fig. 9, as depicted in Fig. 10a. The results of applying the nearest neighbor filter are presented in Fig. 10b, while Fig. 10c shows the outcome using our proposed algorithm. In this example, the advantage of our method is clear: the nearest neighbor inpainting produces a linear propagation of edges, which

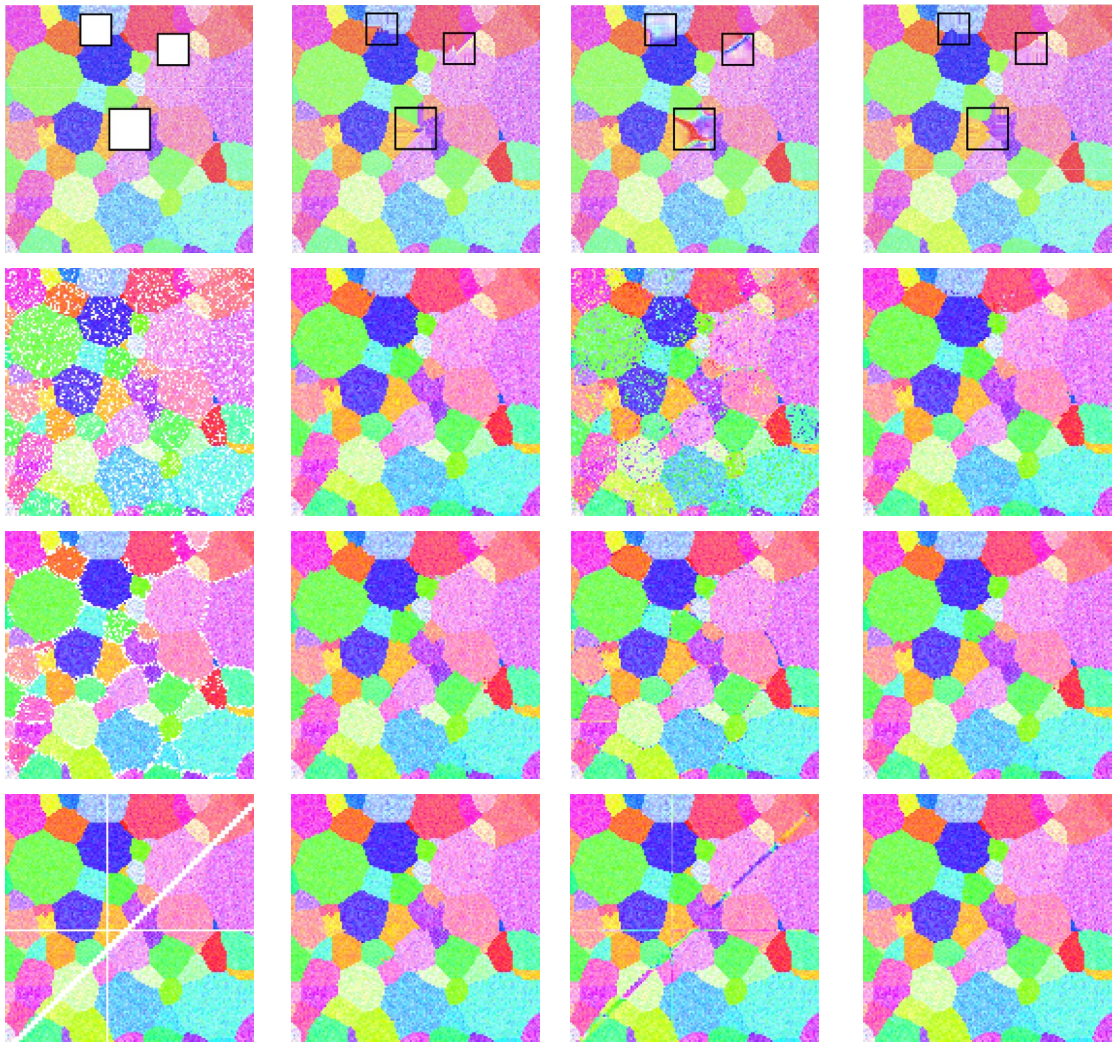


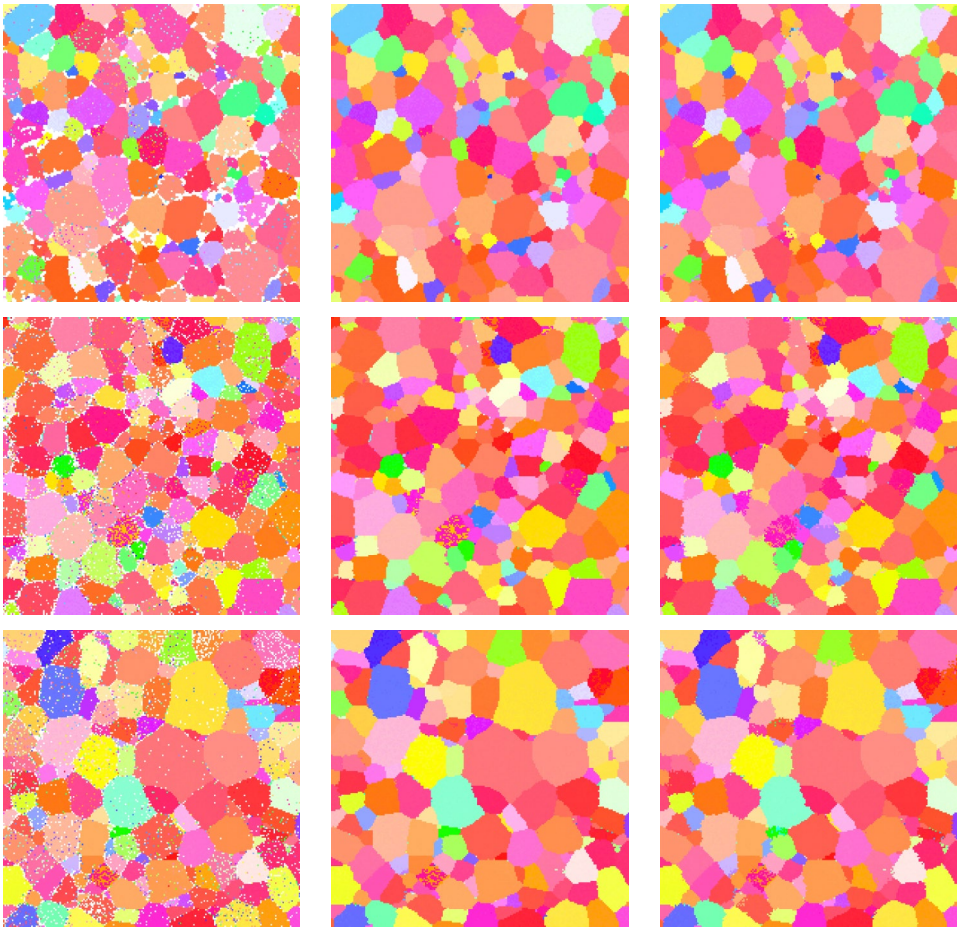
Fig. 8 IPF-z maps of the inpainting results for the various patterns of missing regions. The column **a** shows the missing regions. Columns **b**–**d** show the results obtained with various algorithms: (**b**) *SSδEu* algorithm, (**c**) modified PNN algorithm, and (**d**) proposed algorithm detailed in section "The Proposed Hybrid Inpainting Algorithm"

appears unnatural, while our algorithm yields more realistic results. This is consistent with expectations, as our approach represents a significant advancement over the exemplar-based Criminisi algorithm [16], which itself improves upon the nearest neighbor filter.

Limitations and Future Directions

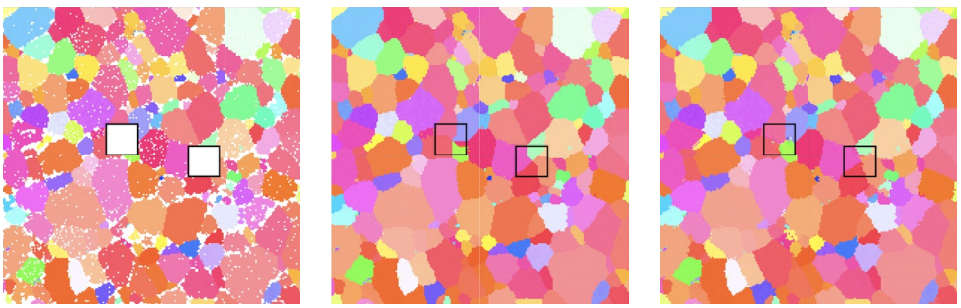
Currently, our algorithm can be applied to 3D datasets such as HEDM datasets on a slice-by-slice basis, but it does not yet support direct application to 3D volumes. This is due to high storage requirements and computational resources, which we currently do not possess. However, the core principles underlying our method, i.e., exemplar-based and neural network components, can be extended to 3D. This would require adapting the algorithm to incorporate the third spatial dimension and generating synthetic 3D EBSD datasets for training. We acknowledge that such an extension would significantly increase the computational demands of training and implementation, and we consider this a promising direction for future work.

Intragranular orientation gradients commonly occur in mechanically deformed specimens [37, 38]. However, our current algorithm was trained on EBSD data without such gradients, leading to oversmoothing in regions with real orientation variation. This, in turn, degrades the performance of the exemplar-based step, which depends on accurate local



(a) Real EBSD data (b) nearest neighbor (c) proposed inpainting

Fig. 9 IPF-z maps for the **a** real EBSD data from NIST [36], **b** inpainting result using the nearest neighbor inpainting from Dream.3D software, **c** inpainting using the proposed method



(a) Real EBSD data (a) nearest neighbor (c) proposed inpainting

Fig. 10 IPF-z maps for the **a** real EBSD data from NIST [36] with 20×20 sized missing blocks introduced, **b** inpainting result using the nearest neighbor inpainting from Dream.3D software, **c** inpainting using the proposed method

context for matching. To address this limitation, future work will focus on training with EBSD maps that include realistic intragranular gradients and modifying the exemplar-based algorithm to incorporate gradient-aware dissimilarity measures. Nonetheless, the core principles of the hybrid framework remain applicable.

In this work, our approach combined a modified exemplar-based inpainting method and machine-learning inpainting based on Liu et al. [17]. In the future, we plan to explore other deep-learning techniques, such as generative adversarial networks (GANs) or reinforcement learning, which can improve the machine learning stage of the hybrid approach.

Conclusion

In this manuscript, we presented a comprehensive study of missing values in orientation maps and developed a hybrid inpainting algorithm to fill in missing values in grain orientation maps, such as electron backscatter diffraction (EBSD) maps. The hybrid approach successfully integrates the strengths of machine learning approximations and exemplar-based techniques, and is thereby able to address the unique challenges of preserving and reconstructing the geometric structures in EBSD maps. With extensive experimentation, we were able to demonstrate that our proposed hybrid algorithm, when utilizing a weight parameter $\omega = 0$, outperforms both the modified exemplar-based and machine learning-based methods applied independently. Our algorithm not only retains the geometric integrity of the grain boundaries but also accurately fills in the missing orientation data. Thus, the algorithm produces high-quality reconstructions from corrupt EBSD maps, a crucial tool for reliable material analysis.

The proposed hybrid algorithm is applicable to various grain orientation data types like ACOM-TEM, XRD, and HEDM, by effectively using exemplar-based methods to mitigate the limitations of deep-learning-based inpainting.

Broader implication of our work: This paper addresses the inpainting challenges specific to electron backscatter diffraction (EBSD) data. However, the proposed hybrid algorithm is versatile and can be directly applied to other types of grain orientation data, including Automated Crystal Orientation Mapping in Transmission Electron Microscopy (ACOM-TEM), X-ray diffraction (XRD), and high-energy X-ray diffraction microscopy (HEDM). Moreover, the hybrid approach, which leverages exemplar-based methods to counterbalance the limitations of deep-learning techniques, holds potential value for a variety of data patterns beyond grain orientations.

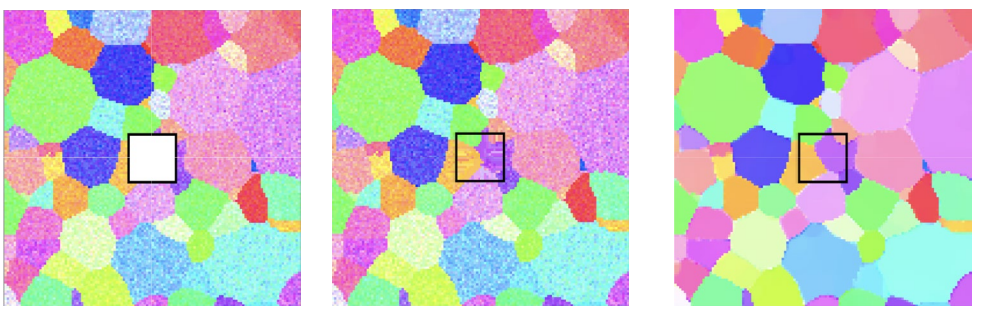
Appendix A: EBSD Restoration Pipeline: Denoising the Inpainted Results

The goal of this paper is to propose an inpainting algorithm for large missing regions. We tested the proposed algorithm to inpaint the given noisy EBSD map. However, in practice, we need to obtain a clean EBSD map. To this effect, we propose that we denoise the inpainted EBSD map using the denoising algorithm proposed in Atindama et al. [8]. This denoising method is based on weighted total variation flow [39–42]. Figure 11 shows the final restoration result. Figure 11a shows the noisy EBSD map with the missing region, and Fig. 11b shows the result of our proposed inpainting algorithm. Finally, Fig. 11c is the result of the denoising algorithm due to Atindama et al. [8]. The reader is encouraged to compare this result with the clean EBSD map in Fig. 2a. The denoised result has a misorientation error of 0.044 radian compared with the clean ground truth shown in Fig. 2a.

Moreover, we present the denoising of the inpainted EBSD maps with various categories of missing regions in Fig. 12. Table 4 details the results of the nearest neighbor, proposed method, and denoising of the inpainting results. We see that after denoising, the errors are significantly reduced.

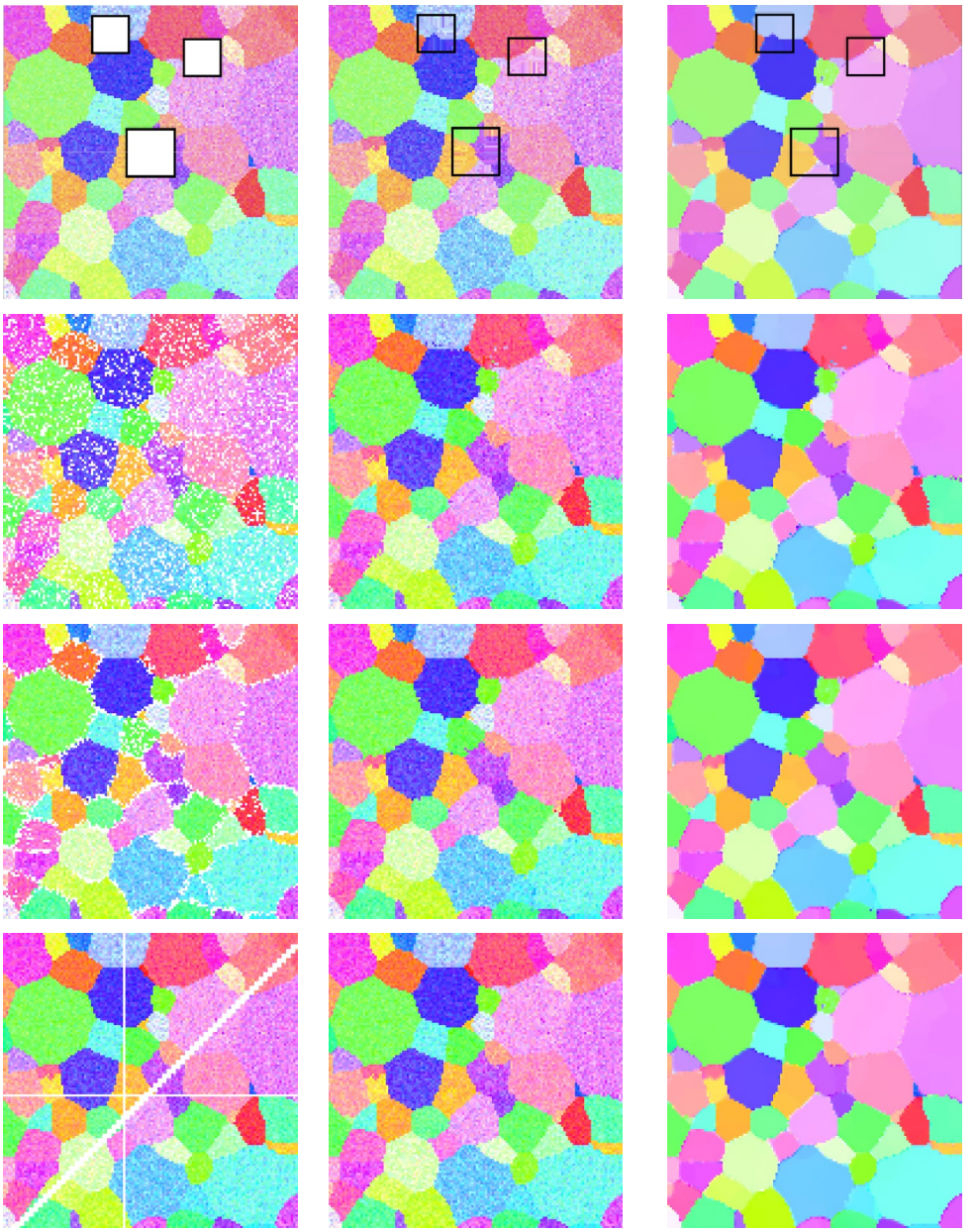
Appendix B: Disorientation Error Maps

Figure 13 shows the disorientation errors, with respect to the 'ground truth', corresponding to the inpainting results depicted in Fig. 6. The errors are primarily attributed to misaligned grain boundaries, which account for the relatively high mean disorientation error of 0.406 radians observed for the hybrid method in Fig. 5.



(a) Given EBSD map (b) proposed inpainting (c) inpainted & denoised [8]

Fig. 11 IPF-z maps of the **a** given noisy EBSD map with the missing region, **b** inpainting result using the proposed hybrid algorithm, **c** result of the denoising algorithm proposed in Atindama et al. [8]. Compare the restored map in (**c**) with the original clean EBSD map in Fig. 2a



(a) Given EBSD maps (b) proposed inpainting (c) inpainted & denoised [8]

Fig. 12 IPF-z maps of the **a** given noisy EBSD maps with various patterns of missing regions, **b** inpainting result using the proposed hybrid algorithm, **c** result of the denoising algorithm proposed in Atindama et al. [8]. Compare the restored map in (**c**) with the original clean EBSD map in Fig. 2a

Table 4 Reconstruction errors of the nearest neighbor fill-in method, the proposed inpainting algorithm, and the denoising of the results obtained using the proposed inpainting

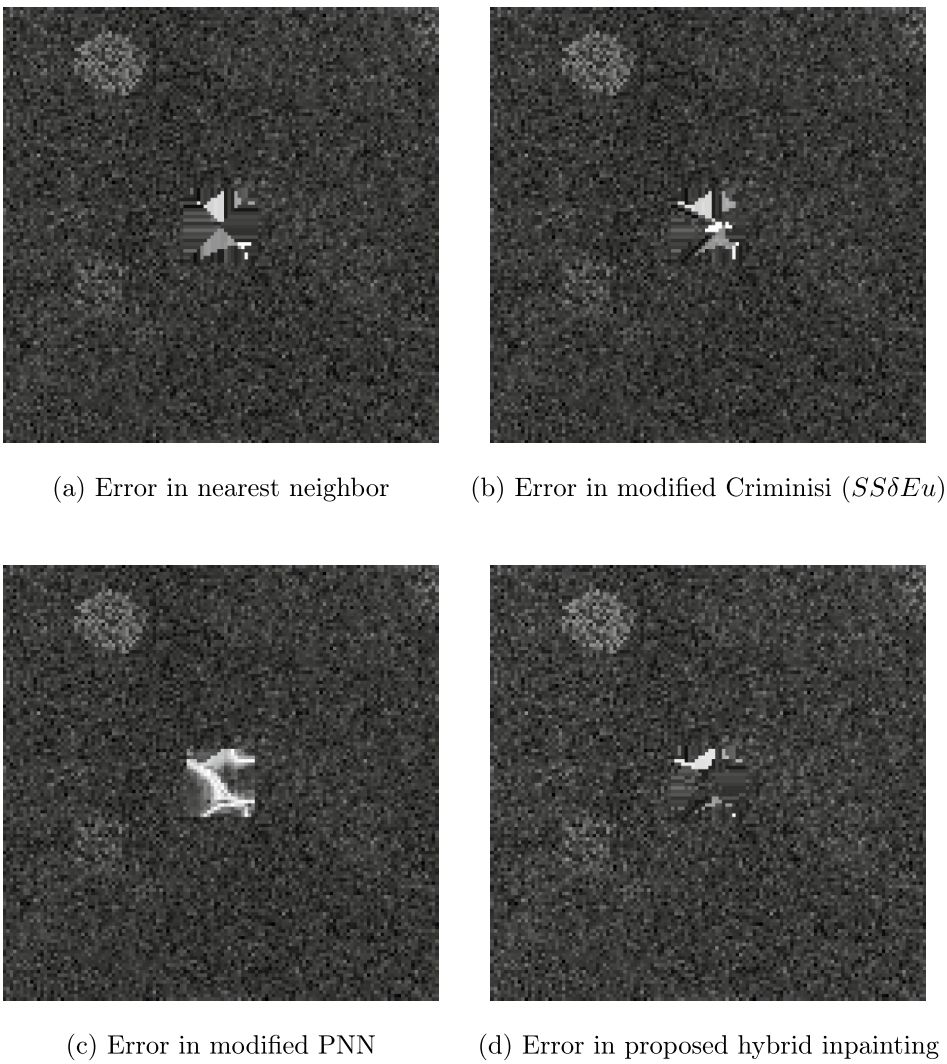
Missing region	Nearest neighbor	Proposed inpainting	Inpainting + denoising
oneblock	0.549	0.340	0.290
threeblocks	0.436	0.240	0.197
boundary	0.495	0.268	0.205
random	0.285	0.171	0.103
zigzag	0.285	0.113	0.056

Per pixel reconstruction errors (in radians) using various inpainting algorithms the EBSD maps in Figs. 11 and 12

Editor's Video Summary The online version of this article (https://doi.org/10.1007/s40192-025-00402-9) contains an Editor's Video Summary, which is available to authorized users.

Acknowledgements Real EBSD maps were courtesy of Dr. Adam Creuziger from NIST, Gaithersburg, MD [36].

Data Availability The denoising code and inpainting code are available in the following GitHub repository: https://github.com/Atindama/EBSD-Restoration-Inpainting-and-Denoising. The training and validation datasets for the neural network are available at https://doi.org/10.18434/mds2-3694. NIST Disclaimer: Certain equipment, instruments, software, or materials, commercial or non-commercial, are identified in this paper in order to specify the experimental procedure adequately. Such identification is not intended to imply recommendation or endorsement of any product or service by NIST, nor is it intended to imply that the materials or equipment identified are necessarily the best available for the purpose.



(a) Error in nearest neighbor

(b) Error in modified Criminisi ($SS\delta Eu$)

(c) Error in modified PNN

(d) Error in proposed hybrid inpainting

Fig. 13 Disorientation error maps of the inpainting results shown in Figure 6. **a** error map using the Dream.3D neighborhood filling filter, **b** error map with modified Criminisi algorithm, i.e., the *SSδEu* algorithm, **c** error map for the modified PNN using the loss function shown in Eq. (4), **d** error map for the proposed hybrid algorithm

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Zaefferer S (2011) A critical review of orientation microscopy in SEM and TEM. Crystal Res Technol 46(6):607–628. https://doi.org/10.1002/crat.201100125
2. Rauch EF, Portillo J, Nicolopoulos S, Bultreys D, Rouvimov S, Moeck P (2010) Automated nanocrystal orientation and phase mapping in the transmission electron microscope on the basis of precession electron diffraction. Zeitschrift für Kristallographie-Crystalline Materials 225(2-3): 103–109 https://doi.org/10.1524/zkri.2010.1205
3. Epp J (2016) 4 - X-ray diffraction (XRD) techniques for materials characterization. In: Hübschen G, Altpeter I, Tschuncky R, Herrmann HG (eds) Materials characterization using nondestructive evaluation (NDE) methods, pp 81–124. Woodhead Publishing, Sawston, UK. https://doi.org/10.1016/B978-0-08-100040-3.00004-3
4. Poulsen HF (2004) Three-dimensional X-ray diffraction microscopy: mapping polycrystals and their dynamics, 1st edn. Springer, Berlin. https://doi.org/10.1007/b97884
5. Creuziger A, Vaudin M (2011) Report on vamas round robin of iso 13067: microbeam analysis - electron backscatter diffraction - measurement of average grain size. In: Technical report NISTIR 7814, national institute of standards and technology. https://doi.org/10.6028/NIST.IR.7814
6. Langer SA, Fuller ER, Carter WC (2001) OOF: an image-based finite-element analysis of material microstructures. Comput Sci Eng 3(3):15–23. https://doi.org/10.1109/5992.919261
7. Nolze G (2015) Euler angles and crystal symmetry. Cryst Res Technol 50(2):188–201. https://doi.org/10.1002/crat.201400427
8. Atindama E, Lef P, Doğan G, Athavale P (2023) Restoration of noisy orientation maps from electron backscatter diffraction imaging. Integr Mater Manuf Innov 12(3):251–266. https://doi.org/10.1007/s40192-023-00304-8
9. Aubert G, Kornprobst P (2006) Mathematical problems in image processing: partial differential equations and the calculus of variations. Appl Math Sci, vol. 147. Springer, New York
10. Schönlieb CB (2015) Partial Differential Equation Methods for Image Inpainting, vol 29. Cambridge University Press, Cambridge
11. Athavale P, Dey S, Dharmatti S et al (2021) A novel entropy-based texture inpainting algorithm. SIViP 15:1075–1080. https://doi.org/10.1007/s11760-020-01833-x
12. Romero A, Castillo A, Abril-Nova J, Timofte R, Das R, Hira S, Pan Z, Zhang M, Li B, He D, Lin T, Li F, Wu C, Liu X, Wang X, Yu Y, Yang J, Li R, Zhao Y, Guo Z, Fan B, Li X, Zhang R, Lu Z, Huang J, Wu G, Jiang J, Cai J, Li C, Tao X, Tai YW, Zhou X, Huang H (2022) NTIRE 2022 image inpainting challenge: Report. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR) workshops, pp 1150–1182
13. Demir U, Unal G (2018) Patch-based image inpainting with generative adversarial networks . arxiv:1803.07422
14. Suvorov R, Logacheva E, Mashikhin A, Remizova A, Ashukha A, Silvestrov A, Kong N, Goka H, Park K, Lempitsky V (2022) Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF winter conference on applications of computer vision (WACV), pp 2149–2159
15. Huang Y, Huang J, Liu Y, Yan M, Lv J, Liu J, Xiong W, Zhang H, Cao L, Chen S (2025) Diffusion model-based image editing: a survey. IEEE Trans Pattern Anal Mach Intell 1–27. https://doi.org/10.1109/tpami.2025.3541625
16. Criminisi A, Perez P, Toyama K (2004) Region filling and object removal by exemplar-based image inpainting. IEEE Trans Image Process 9:1200–1212. https://doi.org/10.1109/TIP.2004.833105
17. Liu G, Reda F.A, Shih K.J, Wang T, Tao A, Catanzaro B (2018) Image inpainting for irregular holes using partial convolutions. CoRR arXiv:1804.07723
18. Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF (eds) Medical image computing and computer-assisted intervention – MICCAI 2015, pp 234–241. Springer, Cham . https://doi.org/10.1007/978-3-319-24574-4_28
19. LeCun Y, Boser B, Denker JS, Henderson D, Howard RE, Hubbard W, Jackel LD (1989) Backpropagation applied to handwritten zip code recognition. Neural Comput 1(4):541–551. https://doi.org/10.1162/neco.1989.1.4.541
20. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: Bach F, Blei D (eds) Proceedings of the 32nd international conference on machine learning. Proceedings of machine learning research, vol 37, pp 448–456. PMLR, Lille, France. https://proceedings.mlr.press/v37/ioffe15.html
21. Wu Y, He K (2018) Group Normalization. https://doi.org/10.48550/arXiv.1803.08494
22. Agarap AF (2018) Deep learning using rectified linear units (relu). arXiv preprint arXiv:1803.08375
23. Maas AL, Hannun AY, Ng AY (2013) Rectifier nonlinearities improve neural network acoustic models. In: Proceedings of the 30th international conference on machine learning (ICML), p 3, Vol. 30(1)
24. Xu B, Wang N, Chen T, Li M (2015) Empirical evaluation of rectified activations in convolutional networks. arXiv preprint arXiv:1505.00853
25. Watkins DS (2004) Fundamentals of Matrix Computations. Wiley, Hoboken
26. Neter J, Kutner MH, Nachtsheim CJ, Wasserman W (1996) Applied linear statistical models
27. Dream.3D: Dream.3D. http://dream3d.bluequartz.net/. Online, accessed September 9, 2020 (2020)
28. Groeber MA, Jackson MA (2014) DREAM.3D: a digital representation environment for the analysis of microstructure in 3D. 3:56–72. https://doi.org/10.1186/2193-9772-3-5
29. Schaeben H (1999) The de la Vallée Poussin standard orientation density function. Textures Microstruct 33(1–4):365–373. https://doi.org/10.1155/TSM.33.365
30. Tanimoto T (2023) PartialConv: partial convolution for inpainting. https://github.com/tanimutomo/partialconv. Accessed July 12, 2024
31. Atindama E, Miller-Lynch C, Wilhite H, Mattice C, Doğan G, Athavale P (2024) EBSD restoration: inpainting and denoising. https://github.com/Atindama/EBSD-Restoration-Inpainting-and-Denoising. Accessed July 12, 2024
32. EBSD – Electron Backscatter Diffraction. https://www.ebsd.com. Accessed Feb 9, 2024 (2024)
33. MTEX: MTEX Toolbox. https://mtex-toolbox.github.io/. Online; accessed September 9, 2020 (2020)
34. Atindama E, Athavale P, Miller-Lynch C, Doğan G (2025) Training and validation datasets for neural network to fill in missing data in EBSD maps. First released: 2025-03-24, Revised: 2025-03-24. Contact: Günay Doğan . https://doi.org/10.18434/mds2-3694

35. Atindama E, Miller-Lynch C, Doğan G, Athavale P (2024) Training and validation datasets for neural network to fill in missing data in EBSD maps. https://doi.org/10.18434/mds2-3694
36. Creuziger A, Vaudin M (2023) Data publication: data from NIST IR report on VAMAS round Robin of ISO 13067: Microbeam analysis—electron backscatter diffraction—measurement of average grain size . https://doi.org/10.18434/mds2-2951
37. Stinville JC, Hestroffer JM, Charpagne MA et al (2022) Multi-modal dataset of a polycrystalline metallic material: 3D microstructure and deformation fields. Dryad. [Dataset]. https://doi.org/10.5061/dryad.83bk3j9sj
38. Anjaria D, Stinville JC (2025) In-situ tensile test scanning electron microscopy dataset for solutionized Inconel 718. Dryad. [Dataset]. https://doi.org/10.5061/dryad.98sf7m0tt
39. Athavale P, Xu R, Radau P, Nachman A, Wright G (2013) Multiscale TV flow with applications to fast denoising and registration. In: Medical imaging 2013: image processing, vol 8669, p 86692. SPIE, Lake Buena Vista, Florida. https://doi.org/10.1117/12.2007190
40. Athavale P, Xu R, Radau P, Nachman A, Wright GA (2015) Multiscale properties of weighted total variation flow with applications to denoising and registration. Med Image Anal 23(1):28–42. https://doi.org/10.1016/j.media.2015.04.013
41. Athavale P, Jerrard RL, Novaga M, Orlandi G (2017) Weighted TV minimization and applications to vortex density models. J Convex Anal 24(4):1051–1084 https://doi.org/10.48550/arXiv.1509.03713
42. Xu R, Athavale P, Nachman A, Wright GA (2014) Multiscale registration of real-time and prior MRI data for image-guided cardiac interventions. IEEE Trans Biomed Eng 61(10):2621–2632. https://doi.org/10.1109/TBME.2014.2324998