

Toward Better Planetary Surface Exploration by Orbital Imagery Inpainting

Hiya Roy , Subhajit Chaudhury, Toshihiko Yamasaki , and Tatsuaki Hashimoto 

Abstract—Planetary surface images are collected by sophisticated imaging devices onboard the orbiting spacecraft. Although these images enable scientists to discover and visualize the unknown, they often suffer from the ‘no-data’ region because the data could not be acquired by the onboard instrument due to the limitation in operation time of the instrument and satellite orbiter control. This greatly reduces the usability of the captured data for scientific purposes. To alleviate this problem, in this article, we propose a machine learning-based ‘no-data’ region prediction algorithm. Specifically, we leverage a deep convolutional neural network (CNN) based image inpainting algorithm to predict such unphotographed pixels in a context-aware fashion using adversarial learning on planetary images. The benefit of using our proposed method is to augment features in the unphotographed regions leading to better downstream tasks such as interesting landmark classification. We use the Moon and Mars orbital images captured by the JAXA’s Kaguya mission and NASA’s Mars Reconnaissance Orbiter (MRO) for experimental purposes and demonstrate that our method can fill in the unphotographed regions on the Moon and Mars images with good visual and perceptual quality as measured by improved PSNR and SSIM scores. Additionally, our image inpainting algorithm helps in improved feature learning for CNN-based landmark classification as evidenced by an improved F1-score of 0.88 compared to 0.83 on the original Mars dataset. **COMP:** Please replace colons appearing after figure numbers and table numbers with period in all figure and table captions.

Index Terms—Classification, image inpainting, machine learning, mars, moon, neural networks, remote sensing, supervised learning.

I. INTRODUCTION

IN THE Quest of exploring and understanding planetary bodies, several missions to the Moon, Mars, and other planets in the solar system have been carried out over the years. Advancement in imaging devices has enabled humans to visualize the planetary terrains and inspired them to discover how planets have evolved. Such high-resolution orbital images are crucial in providing us unprecedented views of interesting planetary

surface features or characterizing potential candidates for future landing sites [1]. For example, onboard cameras of Kaguya mission’s Selene spacecraft [2], and Mars Reconnaissance Orbiter (MRO) [3] have provided scientists with Lunar and Mars orbital imagery. However, to obtain these high-resolution images, the swath width of the on-board cameras of the orbiting satellite is kept lower which in turn creates discontinuity or black lines on the Lunar or Mars surface image. Although Moon and Mars are the most extensively studied celestial bodies, there still exist small portions that are yet not covered by the onboard instruments. Moreover, there are other planets (e.g., Mercury, Pluto, etc.) or celestial bodies where no-data regions exist because a large percentage of the surface of these celestial bodies are not yet captured. Therefore, till the time the global mapping of the entire planetary surface is completed, the problem of no-data problem will exist. Examples of such unphotographed/missing regions on Moon, Mars, Mercury, and Earth remote sensing images are shown in Fig. 1.

Such unphotographed pixels limit the application and usability of data, in classifying or recognizing interesting morphological features in the planetary surface. Therefore, restoring them is of great significance for many practical applications such as improving classification accuracy, enhancing data availability, to make more accurate location adjustments while making the mosaic of the planetary surface where the region is not illuminated by Sunlight such as the Polar region, to improve the landing site candidate selection efficiency, etc. Although one might think that filling the unphotographed region with *artificial* pixel values might be harmful from the viewpoint of precise observation, nevertheless we show that such unphotographed pixel prediction can effectively improve the performance of terrain classification due to improved feature learning.

With the increasing amount of image data available from the ongoing planetary imaging investigations [2]–[5], there is an urgent need for automated vision-based algorithms that achieve good feature learning of interesting landmarks. However, because of the unphotographed regions, the interesting features sometimes appear incomplete. In this article, we aim to predict such a region to enable the network to learn the complete feature of the interesting landmark which in turn leads to better classification performance.

Previous research works have shown that it is possible to reconstruct missing data on remote sensing imagery on Earth [6]–[21]. While these previous methods perform well for Earth remote sensing data, they are not suitable for planetary images such as Moon or Mars. This is because planetary surface

Manuscript received June 20, 2020; revised September 1, 2020 and October 10, 2020; accepted November 8, 2020. Date of publication November 17, 2020; date of current version January 6, 2021. (Corresponding author: Hiya Roy.)

Hiya Roy is with the Department of Electrical Engineering and Information Systems, The University of Tokyo, 113-8656 Japan (e-mail: hiya@hal.t.u-tokyo.ac.jp).

Subhajit Chaudhury and Toshihiko Yamasaki are with the Department of Information and Communication Engineering, The University of Tokyo, 113-8656 Japan (e-mail: subhajit@hal.t.u-tokyo.ac.jp; yamasaki@hal.t.u-tokyo.ac.jp).

Tatsuaki Hashimoto is with the Department of Electrical Engineering and Information Systems, The University of Tokyo and Japan Aerospace Exploration Agency (JAXA), 252-5210, Japan (e-mail: hashimoto.tatsuaki@jaxa.jp).

Digital Object Identifier 10.1109/JSTARS.2020.3038778

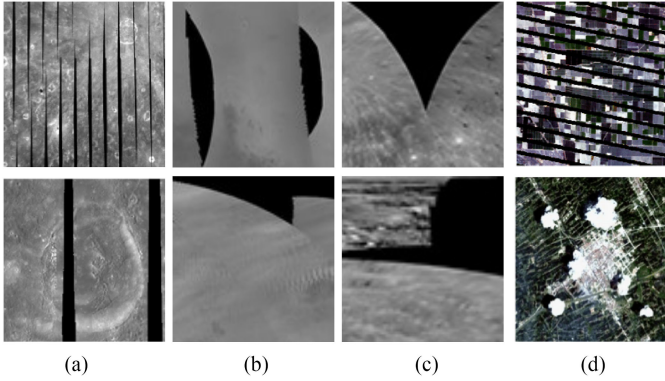


Fig. 1. Example of unphotographed/missing pixel regions on (a) Lunar orbital imagery acquired by Kaguya mission's SELENE spacecraft [2]. (b) Mars orbital imagery acquired by MRO [3]. (c) Mercury orbital imagery acquired by MESSENGER spacecraft [5]. (d) Earth remote sensing images [6].

images differ from Earth remote sensing data in terms of the histogram, contrast, presence of different geological features, etc. Moreover, the vast difference in temperature, presence of the atmosphere, water, vegetation, etc., on Earth makes geological features (valleys, channels, etc.) on earth remote sensing images look much different from that of other planetary bodies. Furthermore, in the case of the planetary image dataset, we found that there exists a gradation (different modes) of histogram distribution in the input images, which cannot be solved efficiently using existing missing data reconstruction techniques on remote sensing imagery. Therefore, we propose mode-specific expert neural networks that can handle such peculiarity of histogram distribution on any planetary surface images. In this article, we have shown the effectiveness of our proposed algorithm only on Lunar and Mars surface images. However, our algorithm is applicable to any planetary images that suffer from such no-data regions.

Recently, deep learning [23] has garnered tremendous success because of its ability to express nonlinear functions. Benefiting from this trend, CNNs have demonstrated outstanding performance in solving several high-level vision-based tasks such as image classification [25]–[28], object recognition [29], [30], etc., as well as low-level tasks such as image denoising [31], super-resolution [32], etc. Therefore, in this work, we employ a CNN for restoring planetary orbital imagery contaminated with unphotographed pixels. Here, we treat this problem as an image inpainting problem where the main challenge is to synthesize the unphotographed pixels in such a way that it looks visually realistic when compared to the original ones. Another challenge associated with planetary image restoration is that the input images have several modes of histogram distribution which inhibits the generative model to faithfully reproduce samples representing each histogram mode. We tackled this problem by clustering images with similar intensity distribution and then training regression models having expertise in restoring unphotographed pixels in the images with that particular intensity distribution. Our intuition is that mode-specific encoders will provide better inpainting results when compared with only one encoder trained on an average intensity distribution [33]. We

build upon the recently proposed context encoder (CE) [55] which is a generative adversarial networks (GAN) [56] based network where our network first learns to predict and fill in the unphotographed region. Then it uses the learned feature representation as guidance to classify the morphological features on the planetary surface. Our main contributions in this article can be summarized as follows.

- 1) We introduce an adversarial learning-based image inpainting framework for planetary images (Moon, and Mars) that learns a nonlinear end-to-end mapping from corrupted to clean images.
- 2) To enable better inpainting we extract various modes of histogram distribution in the input images by unsupervised clustering. We train mode-specific GAN models which are expert models for inpainting images belonging to that cluster of the histogram mode. The simulated and real experimental results show that our proposed approach can restore images with a significant improvement in terms of visual quality and evaluation metrics, thereby outperforming previous inpainting methods.
- 3) We show that our proposed inpainting method helps in augmenting features of interesting but masked/incomplete landmarks which in turn leads to better generalization. Our experimental results also validate this concept by boosting the classification accuracy of the morphological features on Mars images.

The rest of this article is organized as follows. Section II describes the related works on image inpainting techniques on standard datasets and remote-sensing datasets. Section III provides the details of the planetary dataset that we have used for our experimental purpose. Section IV explains our proposed method including the clustering of our training and testing images based on histogram distribution, the training and implementation details of our inpainting module, and our classification module. Section V provides the experimental results of the missing data reconstruction in both simulated and real-data experiments and its contribution to boosting classification performance. Finally, our conclusions are presented in Section VI.

II. RELATED WORKS

In computer vision, the task of filling in the missing pixels of an image is known as image inpainting. In this section, we briefly review the previous image inpainting works on standard real-life datasets [36]–[39] by broadly categorizing them into three sub-fields, (i) traditional inpainting techniques, (ii) CNN-based inpainting, and (iii) GAN-based inpainting. We also review previous works on remote sensing imagery inpainting on Earth and Moon.

A. Traditional Inpainting Techniques

Traditionally, a variety of image inpainting approaches have been proposed in the literature. One approach in this family is known as *diffusion-based image completion* [40]–[43] where a diffusive process is modeled using partial differential equations (PDE) to propagate colors into the missing regions. Chan *et al.* [44] proposed a novel adaptive total variation model by

combining the diffusion mechanism of the TV model based on PDE and an edge detection operation to improve inpainting performance by eliminating the staircase effect. These methods work well for inpainting small missing regions, but fail to reconstruct the structural component or texture for larger missing regions.

Another approach is known as *patch-based image completion*, which can handle complicated image completion tasks such as large hole filling in natural images. Efros and Freeman [45] first proposed a patch-based algorithm for texture synthesis, which is based on iteratively searching for similar patches in the existing image and paste/stitch the most similar block onto the image. However, patch-based methods are computationally very expensive because of the need for computing similarity scores for every target-source pair. Therefore, for more accurate and faster image inpainting, an optimal patch search algorithm (fragment-based image completion) was proposed by Drori *et al.* [46]. Another optimization method to synthesize visual data (images or video) based on bi-directional similarity measure was proposed by Simakov *et al.* [47]. Later these techniques were expedited by Barnes *et al.* [48] who proposed PatchMatch, a fast randomized patch search algorithm that could handle the high computational and memory cost. For image completion, several *exemplar-based image completion* methods have also been proposed. Criminisi *et al.* [49] proposed a patch-based greedy sampling algorithm, which enables faster image inpainting. Meur *et al.* [50] introduced a hierarchical super-resolution algorithm for image inpainting. He *et al.* [51] approached the image completion problem by computing the statistics of patch offsets. However, the above methods rely only on existing image patches and use low-level image features. Therefore they are not effective in filling complex structures by performing semantically aware patch selections.

B. CNN-Based Inpainting

With the recent success of CNN models [24] in tackling harder problems such as classification, object detection, and segmentation, that need a high-level semantic understanding of an image, CNNs became a popular choice to solve image inpainting problems as well. Xie *et al.* [52] proposed Stacked Sparse Denoising Auto-encoders (SSDA), a combined approach of sparse coding and deep networks pre-trained with denoising auto-encoder to solve the blind image inpainting task, which is a more challenging inpainting task. This is because, in the case of blind image inpainting, the algorithm does not know the location of the missing pixels and it learns to find the location of the missing pixels and then restore them. Kohler *et al.* [54] showed a mask specific deep neural network-based blind inpainting technique for filling in small missing regions in an image. Chaudhury *et al.* [58] attempted to solve the blind image inpainting task using a lightweight fully convolutional network (FCN) demonstrating a comparable performance with the sparse coding based k singular value decomposition (K-SVD) [59] technique. However, initially, CNN-based image inpainting approaches were limited to very small sized masks.

C. GAN-Based Inpainting

More recently, GAN-based inpainting methods have been proposed which have achieved promising results in solving image inpainting problems. Pathak *et al.* [55] proposed Context Encoders, a channel-wise fully connected convolutional neural network-based approach, that could inpaint large holes or missing regions existing in an image by predicting missing pixels based on the context of the surrounding areas of that region. Their network was trained using both standard ℓ_2 loss and adversarial loss [56]. Later, Iizuka *et al.* [60] extended the work of [55] and demonstrated that by leveraging the benefits of dilated convolution layers, a variant of standard convolutional layers, their encoder-decoder based method could restore missing pixels that are consistent both locally and globally. Similar to [55], this approach also used an adversarial training approach for image completion, but unlike [55], this method could handle arbitrary image size and mask because of the proposed global and local context discriminator networks. Recently, Yu *et al.* [61] presented a unified feedforward generative network with a novel contextual attention layer, trained with reconstruction losses and two Wasserstein GAN [62], [63] and showed that the unified framework could inpaint images with multiple holes of variable sizes situated at arbitrary locations. Later, to handle free-form/irregular masks, Liu *et al.* [64] proposed a partial convolution layer with an automatic mask-update rule, where the mask is updated in such a way that the missing pixels are predicted based on the real pixel values of the original image where the partial convolution can operate. Song *et al.* [65] introduced a segmentation guidance and prediction network that first predicts the segmentation labels of the corrupted image, then fills in the segmentation mask to use it as a guidance to complete the image. Xiong *et al.* [66] showed that by predicting and completing the contour of the foreground image, it can be used as a guidance to inpaint the missing region of a corrupted image. In a similar spirit, Nazeri *et al.* [67] proposed an edge generator that hallucinates the edges of the missing regions which is used as a guidance to the image completion network. Yu *et al.* [68] proposed a gated convolution-based approach to handle free-form image completion.

D. Remote Sensing Imagery Inpainting

Image inpainting on Earth remote sensing images has been widely studied, where such missing pixels occur in the form of dead pixels or thick cloud cover because of the atmospheric environment or the working conditions of the satellite sensor [6], [7]. Remote sensing (RS) image inpainting using spatial information include interpolation-based methods [8]–[10], variation-based methods [11], [12], PDE-based methods [13], and exemplar-based methods [49]. Although spatial-based methods can reconstruct small missing areas, they fail to guarantee precise reconstruction for large missing regions. To overcome these limitations several other techniques such as spectral-based methods (utilizing information from different spectral bands) [14], [15], and temporal-based methods (using data taken at the same location in different periods) [16], [17] have been

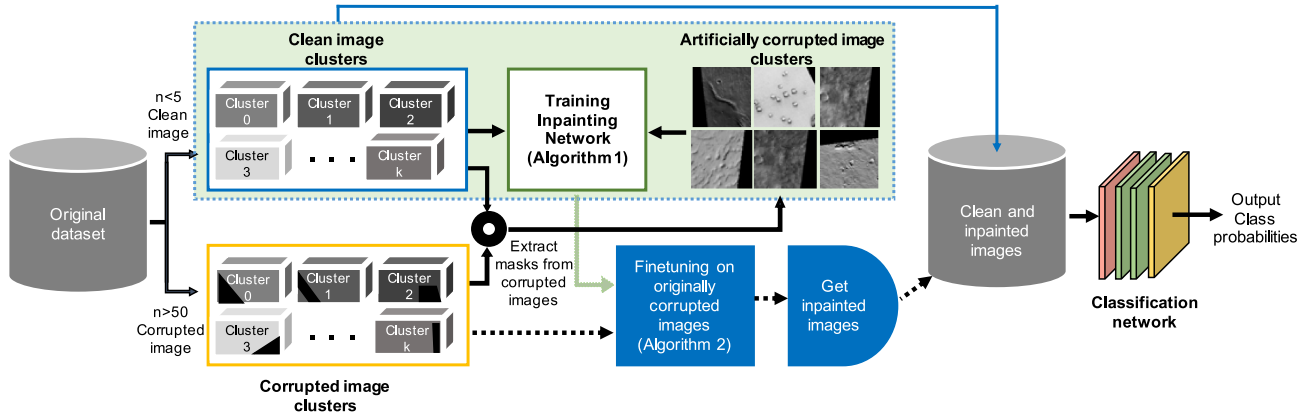


Fig. 2. Overview of our image inpainting algorithm. First, we extract the masks from the real corrupted images and superimpose them on clean images and create a pair of clean and *simulated/artificially corrupted* images. Next, we train the image inpainting module IV-B using these image pairs. Then, we fine-tune the trained inpainting model on real corrupted images. Finally, we store all the clean and inpainted version of the corrupted images and solve a classification problem.

proposed. Later more generalized algorithms (hybrid methods) were developed by integrating spatial, spectral, and temporal information [6]. To this end, Ji *et al.* [18] proposed a nonlocal low-rank tensor completion algorithm to reconstruct the missing information. Cheng *et al.* [20] introduced a double-weighted low-rank tensor (DWLRT) model and He *et al.* [19] proposed a TV-regularized tensor ring completion (TVTR) model to reconstruct missing data in RS images. Recently, Zhang *et al.* [21] proposed a progressive Spatio-temporal patch group learning approach for cloud and cloud shadow removal for RS data. On the other hand, to restore missing pixels on the Lunar surface image, Roy *et al.* [22] proposed a U-Net based approach that minimizes a standard ℓ_2 loss to restore the missing region on Lunar surface images collected by the multiband imager (MI) instrument on-board the Kaguya satellite.

III. EXPERIMENTAL DATA

In this work, we have used Lunar and Mars orbital images to show the effectiveness of our proposed algorithm.

A. Lunar Orbital Imagery by SELENE

We have used the averaged lower resolution mosaic data of the lunar surface captured by multiband imager onboard the JAXA lunar explorer satellite SELENE (Kaguya) [2]. This image covered the lunar surface with longitude (+180 to −180 degree) and latitude (+85 to −85 degree). This image consists of 46080×21760 pixels with an image size of 600 MB. We created this Moon dataset by converting the longitude and latitude of the crater locations in terms of pixel values, where each degree is considered to be 128 pixels and cropped the crater images with and without the black lines. This dataset contains only clean and real-corrupted crater images as shown in Fig. 3(c). Here, each image is of size 256×256 pixels. We created pairs of clean and artificially corrupted crater image pairs by randomly superimposing the black lines (extracted from the real corrupted images) on the clean crater images. A detailed number of such clean and artificially corrupted crater image pairs used for train and test purposes for the Moon dataset is summarized in the first

row of Table I. The second row of Table I describes the total number of real corrupted images in the dataset. Since we did not use the real corrupted Moon images for inference purposes, we did not divide them into clusters and kept the corresponding columns blank in Table I.

B. Mars Orbital Imagery by MRO

For Mars images, we have used the grayscale-version of the Mars orbital images collected by the HiRISE camera on-board the MRO having a spatial resolution of approximately 30 cm/pixel [4]. This dataset [69] is created and labeled by processing map-projected HiRISE images to find eight visually salient and interesting “landmarks” such as craters, dark and bright sand dunes, slope streaks, impact ejecta, swiss cheese, spider, etc., on the planetary surface as shown in Fig. 4. It consists of a total of 73 031 landmarks amongst which 10 433 landmarks are detected and extracted from 180 HiRISE browse images.¹ The remaining 62 598 landmarks are the augmented version (90 degrees, 180 degrees, 270 degrees clockwise rotation, horizontal flip, vertical flip, and random brightness adjusted) of 10 433 original landmarks. Each image is of size 227×227 pixels and for this experimental purpose, we resized them to size 256×256 . A detailed number of clean and artificially corrupted image pairs used for train and test purposes for the Mars dataset is summarized in the first row of Table I. The second row of Table I describes the number of real corrupted images for each cluster and the total number of real corrupted images in the dataset.

It is to be noted that the black region on the Lunar surface image as shown in Fig. 3(c) is a practical example of a no-data region which could not be captured by the onboard camera of the SELENE Kaguya satellite because of the limitation in operation time of the instrument and satellite orbiter control. However, the black regions on Mars HiRISE images as shown in Fig. 3(f) are map projections. Nevertheless, for demonstration purposes, we consider them as an example of the no-data region

¹The dataset is available at <https://zenodo.org/record/2538136#.XYjEuZMzagR>

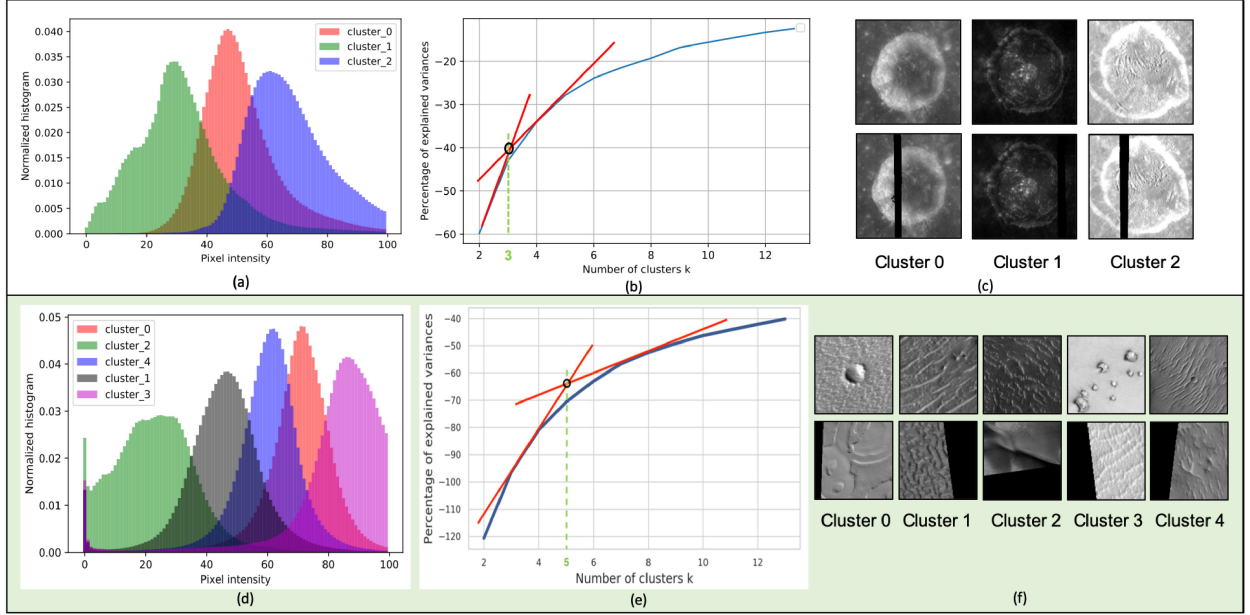


Fig. 3. Details of Moon and Mars dataset: (a, d) Histogram distribution of several modes of input images, (b, e) Knee point analysis for determining the optimal number of clusters, and (c, f) Examples of clean and corrupted images for each cluster. The first and second row demonstrates the details of Moon and Mars images, respectively.

TABLE I
DETAILED NUMBER OF TRAIN AND TEST IMAGES OF MOON AND MARS DATASET

		Moon dataset				Mars dataset					
		Cluster 0	Cluster 1	Cluster 2	Total	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Total
Clean	Train	3385	2052	2562	8000	10162	8620	2997	7786	12557	42122
	Test	847	513	641	2000	2541	2155	750	1947	3140	10533
Real Corrupted		-	-	-	5000	3133	3598	2060	2755	4108	15654

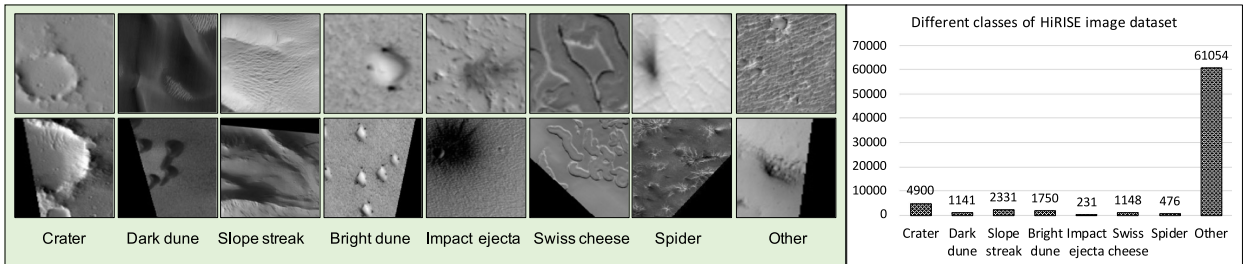


Fig. 4. Example of different classes of Mars dataset with the corresponding number of images for each class.

and propose an algorithm to show how to predict such a no-data region for better surface image analysis.

IV. PROPOSED METHOD

In this article, our goal is to restore a predict the ‘no-data’ region of a corrupted image so that it helps in better feature learning to improve the classification accuracy of interesting landmarks on planetary images. We solve this problem by the following four-stage approach.

1) In our Moon and Mars dataset, we found that the images have several modes in the histogram distributions. Therefore we divide the images into different clusters to prevent the mode-collapse problem in generative models and encourage the network to reliably generate samples from each cluster.

2) Both the Moon and Mars dataset consist of some real corrupted images (with unphotographed pixels) and a majority of clean images. From these real corrupted images, we extract the masks of unphotographed pixel regions and artificially superimpose them on the clean image samples, thus yielding artificially

corrupted and corresponding clean image pairs for each cluster. GAN-based inpainting is performed on such paired data.

3) However, we find that there exists some distribution shift between training and testing images (in terms of gray value, contrast, histogram, etc). Therefore we design a finetuning stage that uses partial ROIs of the real corrupted images to fine-tune the models for matching the testing distribution.

4) Finally, we perform a classification task on original images and compare that to the dataset where clean and inpainted versions of the corrupted images are combined to yield better F1-score due to improved landmark feature learning.

It is to be noted that we perform the first stage (separating images into clusters) and second stage (image inpainting task) for both Moon and Mars image datasets. However, we carry out the third stage (inference on real-corrupted images), and fourth stage (classification) only on the Mars dataset. This is because our Moon dataset consists of images only from one class (crater),² whereas the Mars dataset consists of eight different classes as shown in Fig. 4. Therefore it is more useful to demonstrate the results of the third and fourth stages using the Mars dataset. In the following sections, we first introduce how we preprocess the data and extract masks tailored for our task. Then we explain the detailed implementation of our image inpainting module and the fine-tuning process on real-corrupted images. Finally, we describe our image classification module. The overall framework of our proposed method is shown in Fig. 2.

A. Unsupervised Separation of Histogram Clusters

In our Moon and Mars dataset, we found that planetary images have several modes in the histogram distributions which can be separated into clusters as shown in Fig. 3(a) and (d). To encourage the generative model to faithfully reproduce samples from each such clusters, we separate the images with different histogram distribution into different clusters. Another intuition behind such clustering is that a regression model trained with images of particular intensity distribution such as $p_1(x)$ or $p_2(x)$ will give better performance compared to a single model with an average intensity distribution $p(x)$ [33]. Here, we first calculated the number of black pixels in every image and considered the images having the number of black pixels less than 5 as *clean image* and the number of black pixels more than 50 as *corrupted images*. Then we performed k -means clustering [34], to cluster these images with missing pixels into different groups based on their histogram distribution as shown in the following

$$h_k(x) = \frac{n(x[i, j] = k)}{\text{number of pixels}} \quad (1)$$

where k varies from 1 to 255 for the features and $h_0(x)$ is the number of black pixels.

To find the optimal number of clusters k , we carried out Knee point analysis [35]. As shown in Fig. 3(b) and (e), for Moon

images, the optimal number of image clusters comes out to be 3, whereas for Mars images the optimal number of image clusters comes out to be 5. Examples of clean and real corrupted images corresponding to each clusters are shown in Fig. 3(c) and (f). Subsequently, from the cluster centers in the grayscale histogram space, we assign a class label to each clean image, according to the cluster center having the closest Euclidean distance. Next, for each of the clusters, we extracted a mask of the missing pixels from the corrupted images and randomly superimposed them on clean images and created pairs of clean and artificially corrupted images for training and testing different regression models.

B. Image Inpainting Module

Given a real corrupted image, our goal is to fill in the missing region so that the network can predict/augment the incomplete/masked features of the interesting landmarks with seamless boundary transitions. Intuitively, the missing region can be filled in multiple plausible ways. However, here we aim to restore the missing pixels in such a way that it is the most coherent to its surrounding context. For solving the inpainting task of Moon and Mars images having different intensity distribution, we train adversarial learning based GAN models which has shown promising results in generative modeling of images [76] in recent years. The network architecture of our inpainting module is shown in Fig. 5. Our generator network takes an image with missing pixels and its corresponding binary mask indicating the missing regions as input pairs and outputs the inpainted image.

1) Network Architecture:

Generator: Our generator architecture is adapted from Johnson *et al.* [70] which has shown impressive results for neural style transfer and image-to-image translation [72]. This generator network contains three convolution layers (where Conv2 and Conv3 layers are stride-2 convolution layers responsible for down-sampling twice), eight residual blocks [27], and three convolution layers (where Conv4 and Conv5 layers are transpose convolution layers responsible for up-sampling twice back to the original image size). Here, we use instance normalization [74] and ReLU activation function across all layers of the generator network. A more detailed description of the generator network and output size of each layer is given in Table II.

Discriminator: Our discriminator network is a Markovian discriminator similar to 70×70 PatchGAN, adapted from [71], [72]. The main motivation behind using a PatchGAN discriminator is that it works on a particular patch-size of an image instead of a full image. Therefore, it has fewer parameters compared to a discriminator working on a full image. Moreover, it can be applied to any arbitrarily sized images in a fully convolutional fashion [71], [72]. The details of our discriminator network and output size of each layer is given in Table III. It should be noted that the sigmoid function applied after the last convolution layer produces a 1-D output score that predicts whether the 70×70 overlapping image patches are real or fake. For the discriminator network, we use spectral normalization [73] as our weight normalization method because it can stabilize the discriminator network training. Moreover, here all the ReLUs are leaky ReLUs [75] with slope of 0.2.

²The lunar surface has many geological features other than craters. However, in this article, we have used only one kind of geological feature for the Moon dataset for the sake of easy visual comparison and a lack of dataset availability of different kinds of geological features.

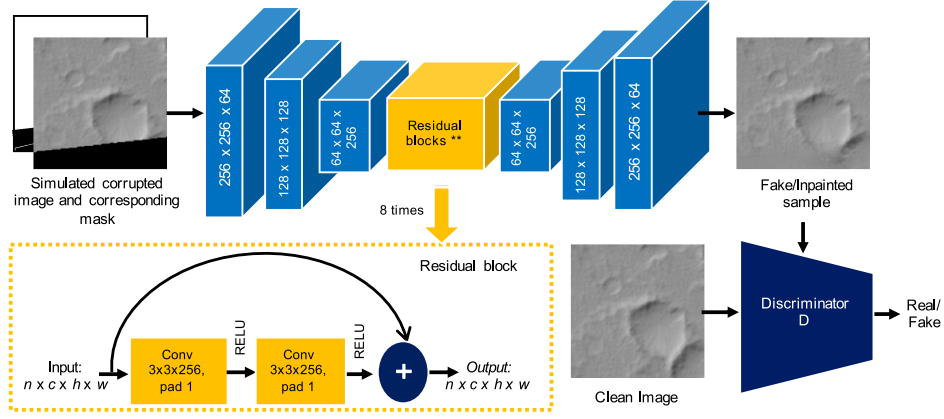


Fig. 5. Network architecture.

TABLE II
GENERATOR NETWORK

Layer name	Stride	Activation	Layer output size
Input	-	-	$1 \times 2 \times 256 \times 256$
Encoder network			
Conv 7×7	1	ReLU	$1 \times 64 \times 256 \times 256$
Conv 4×4	2	ReLU	$1 \times 128 \times 128 \times 128$
Conv 4×4	2	ReLU	$1 \times 256 \times 64 \times 64$
Residual block ($\times 8$)			
Residual blocks	$1 \times 256 \times 64 \times 64$		
Decoder network			
Conv 4×4	2	ReLU	$1 \times 128 \times 128 \times 128$
Conv 4×4	2	ReLU	$1 \times 64 \times 256 \times 256$
Conv 7×7	1	tanh	$1 \times 1 \times 256 \times 256$

TABLE III
DISCRIMINATOR NETWORK

Layer name	Stride	Activation	Layer output size
Conv 4×4	2	LeakyReLU	$1 \times 64 \times 128 \times 128$
Conv 4×4	2	LeakyReLU	$1 \times 128 \times 64 \times 64$
Conv 4×4	2	LeakyReLU	$1 \times 256 \times 32 \times 32$
Conv 4×4	1	LeakyReLU	$1 \times 512 \times 31 \times 31$
Conv 4×4	1	Sigmoid	$1 \times 1 \times 30 \times 30$

2) *Training*: We train our proposed inpainting network in two scenarios: (i) using images from different clusters separately, and (ii) using all images together (not dividing them into clusters). The detailed number of clean and artificially corrupted image pairs for each cluster is given in the first row of Table I. While training, for each real corrupted image x_c , we extract a binary *image mask* \mathbf{m} (which takes the value 0 on the regions to be filled-in and 1 elsewhere). Now for each clean image \mathbf{x} , we randomly superimpose the extracted masks \mathbf{m} and obtain artificially corrupted *input image* $\mathbf{z} = \mathbf{x} \odot \mathbf{m}$, where \odot denotes element-wise product operation. The generator of the inpainting

Algorithm 1: Training of Our Inpainting Framework.

- 1: **while** Generator G has not converged **do**
- 2: Sample batch images \mathbf{x} from clean training data;
- 3: Extract masks \mathbf{m} from corrupted training data;
- 4: Artificially construct corrupted inputs $\mathbf{z} \leftarrow \mathbf{x} \odot \mathbf{m}$;
- 5: Generate inpainted images by modifying masked region, $\tilde{\mathbf{x}} \leftarrow \mathbf{z} + G(\mathbf{z}, \mathbf{m}) \odot (\mathbf{1} - \mathbf{m})$;
- 6: Update G with ℓ_1 loss and adversarial critic loss;
- 7: Update discriminator critic D with \mathbf{x} , $\tilde{\mathbf{x}}$;
- 8: **end while**

network G takes this concatenated *input image* \mathbf{z} and *image mask* \mathbf{m} as input, and produces a *predicted image* $\mathbf{x}' = G(\mathbf{z}, \mathbf{m})$ as output. Then by adding the masked region of \mathbf{x}' to *input image*, we get *completed image* as $\tilde{\mathbf{x}} = [\mathbf{x} \odot \mathbf{m}] + [\mathbf{x}' \odot (\mathbf{1} - \mathbf{m})]$. For clustered training, masks are extracted from corrupted images from the same cluster, whereas no such restriction is imposed for “all” case. The training procedure is described in Algorithm 1.

3) *Loss Functions*: To train our inpainting module to restore the input corrupted image realistically, we use two loss functions: a reconstruction loss and an adversarial loss [56]. Although reconstruction loss helps in capturing the structural details, using only ℓ_1 or ℓ_2 loss often leads to blurry or overly smooth reconstructions [71]. Therefore using adversarial loss along with reconstruction loss is important, because adversarial loss tries to make the prediction look realistic, by fooling the discriminator.

Reconstruction Loss: Previous inpainting approaches [55] have shown that GAN objective function along with a traditional ℓ_2 loss helps in better reconstruction and stabilized GAN training. Here for the reconstruction loss, we use ℓ_1 loss that minimizes the distance between the clean/ground-truth image \mathbf{x} and the completed/inpainted image $\tilde{\mathbf{x}}$

$$\mathcal{L}_{\ell_1}(\mathbf{x}) = \|\mathbf{x} - \tilde{\mathbf{x}}\|_1. \quad (2)$$

Here, $\tilde{\mathbf{x}} = [\mathbf{x} \odot \mathbf{m}] + [\mathbf{x}' \odot (\mathbf{1} - \mathbf{m})]$ and $\mathbf{x}' = G(\mathbf{z}, \mathbf{m})$.

Adversarial Loss: For the adversarial loss, we follow the min-max optimization strategy, where the generator G is trained

to produce inpainted samples from the artificially corrupted images such that the inpainted samples appear as “real” as possible and the adversarially trained discriminator critic D tries to distinguish between the ground truth clean samples and the generator predictions/inpainted samples. The objective function can be expressed as follows

$$G^*, D^* = \arg \min_G \max_D \mathcal{L}_{\text{adv}}(G, D) = \mathbb{E}_{x, \tilde{x}} [\log D(x, \tilde{x})] + \mathbb{E}_{\tilde{x}} [\log(1 - D(\tilde{x}, x'))]. \quad (3)$$

Thus, our overall loss function becomes

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\ell_1} + \lambda_2 \mathcal{L}_{\text{adv}} \quad (4)$$

where we choose $\lambda_1 = 1$ and $\lambda_2 = 0.1$. The weighted sum of these two loss functions compliments each other in the following way. 1) The GAN loss helps to improve the realism of the inpainted images, by fooling the discriminator. 2) The ℓ_1 reconstruction loss serves as a regularization term for training GANs, helps in stabilizing GAN training, and encourages the generator to generate images from the modes that are close to the ground truth in an ℓ_1 sense.

4) *Implementation Details:* Our proposed model is implemented in PyTorch.³ We train our network by optimizing the encoder–decoder and discriminator using the Adam optimizer [77] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. In our experiments, we use a batch size of 14 and the training iterations of 100. While training, we resize the image to 256×256 and linearly scale the pixel values from range $[0, 256]$ to $[-1, 1]$. The Generator G is trained with a learning rate of 10^{-4} until convergence, whereas the Discriminator D is trained with a learning rate of 10^{-5} , one-tenth of that of the generator’s. Both the generator and discriminator networks were trained together on a TITAN Xp (12 GB) GPU.

5) *Fine-Tuning The Network:* We found that the clean images and real corrupted images were visually different based on their grayscale value, contrast, and histogram distribution. This results in poor transfer from training on artificially masked images to true corrupted images. Therefore, it is required to fine-tune the network on images having histogram distribution that matches our intended test images. For this purpose, we fine-tune our pre-trained inpainting model (explained in Section IV-B) on limited regions of the real corrupted images, to get the inpainted images, as described in Algorithm 2.

To fine-tune our inpainting model, during training, we heuristically identify that each corrupted image has four mask directions: *North, South, East, and West*. The first step is to detect the mask direction in the real corrupted image using standard image processing tools (like connected components and center of mass detection). From observation, we find that if an image has a mask of direction d , the image region in the opposite half is usually clean and can be used for creating an artificial training set as before. Thus, we crop the opposite image region and synthetically corrupted that part after resizing it to full resolution

Algorithm 2: Fine-Tuning Our Inpainting Framework For Real Corrupted Images.

```

1: while Generator  $G$  has not converged do
2:   Sample batch images  $x$  from originally corrupted images;
3:   Detect mask  $m$  and its direction  $d$  in  $x$ ;
4:   Crop other half of  $x$  and consider it as clean image;
5:   Resize cropped clean image  $\hat{x}$  to full resolution;
6:   Artificially construct corrupted inputs  $z \leftarrow \hat{x} \odot m$ ;
7:   Get predictions  $\tilde{x} \leftarrow z + G(z, m) \odot (1 - m)$ ;
8:   Update discriminator critic  $D$  with  $\tilde{x}, \hat{x}$ ;
9:   Update  $G$  with  $\ell_1$  loss and adversarial critic loss;
10: end while

```

of 256×256 . A detailed description to artificially create pairs of clean and corrupted images for a sample image (with mask direction in the *East*) during the training stage is shown in Fig. 6(a). After tailoring the real corrupted data for our task, we fine-tune the pre-trained image inpainting model keeping the same optimization conditions as mentioned in Section IV-B4.

During inference, we take a real corrupted image from the test set and heuristically identify the mask direction d similar to training. Next, we crop the clean side (half) of the image, resize it to full resolution of 256×256 and consider it a clean image. Similarly, we crop the corrupted side (half) of the image, resize it to full resolution of 256×256 and consider it as the corrupted image, that needs to be inpainted. While inference, this step is different from training. This is because in training we need to artificially corrupt the clean side by extracting a mask from the corrupted side to get a clean-artificially corrupted image pair. Whereas, during inference, we need to inpaint the originally corrupted side of the image. After we get the inpainted half, we resize the inpainted side and the clean side back to their previous resolution which is 256×128 . After that, we add both the sides to get the inpainted image of full resolution (256×256) corresponding to the real corrupted image. A detailed description of a sample image (with mask direction in the *East*) during inference stage is shown in Fig. 6(b). Here, after cropping the image into half, we follow the standard practice of resizing the image into the full resolution (256×256) before feeding it into the network and then resize it back to its original size. Therefore, we believe resizing the image will not cause distortion and will not affect prediction quality. It is to be noted that although the black regions on the Mars images are generated because of map projection, we consider these images as an example of no-data region images or real corrupted images, to demonstrate how to predict such ‘no-data’ region in case of any planetary images, if they are corrupted by unphotographed pixels.

C. Image Classification Module

After performing the image inpainting to augment the incomplete features on the real corrupted images, we use these inpainted images along with the clean images for the classification task. We perform these additional experiments to check if image inpainting on real corrupted images helps in better

³Our code is available at <https://github.com/hiyaroy12/mars-image-inpainting>

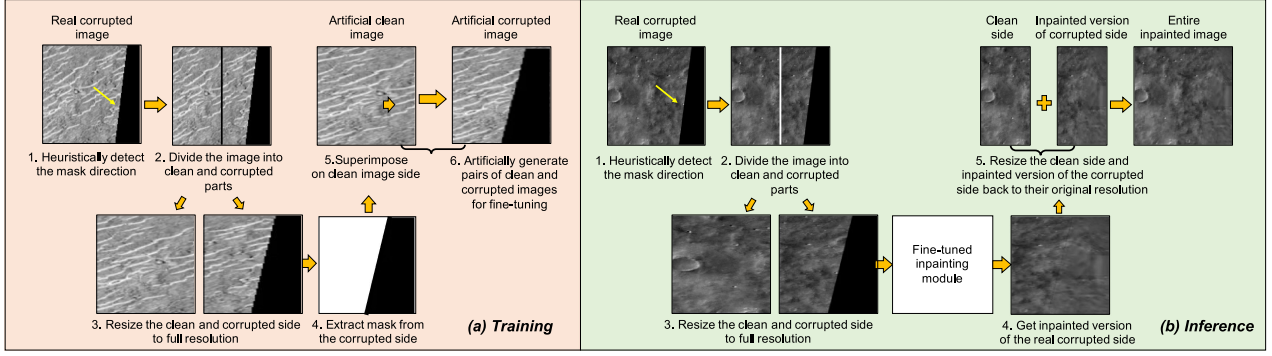


Fig. 6. Processing of real corrupted images to fine-tune our inpainting network (a) during training and (b) during inference. We perform the fine-tuning only on the real corrupted Mars images for demonstration purpose.

feature learning, which in turn leads to improved classification performance. Since our Mars dataset is highly imbalanced, we take a natural approach to resample the given dataset by “oversampling” the minority classes [80] and “undersampling” the majority classes [81]. Such resampling of data helps in achieving a balanced distribution during training. For our experiments, we train two variants of ResNet [27] model ResNet-50 and ResNet-101 for 50 epochs with mini-batch size 80, and a weight decay of 1×10^{-4} . We train both the models via ADAM [77] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We set the initial learning rate to 0.002, resize the input images to 224×224 , and ensure that all images over the training dataset are normalized. We use ResNet classifier [27] for classification because it is the state-of-the-art deep CNN model that can deal with the vanishing gradient problem because of the proposed “identity shortcut connections” implemented as “residual blocks”. Moreover, it was the best performing approach for this dataset.

V. EXPERIMENTAL RESULTS

In this section, we discuss the quantitative and qualitative results obtained from our inpainting module. We perform several experiments to seek answers to the following two questions: 1) Can image inpainting be used for filling in the unphotographed pixels in planetary images?, 2) Does explicit clustering of our training and testing images based on their histogram distribution help in improving inpainting performance? 3) How does fine-tuning on clean portions of the real corrupted images help in improving inpainting quality? and 4) Can our inpainting method contribute to better feature learning for interesting landmark classification thereby improving classification performance?

Quantitative Results: We report the quantitative performance of our method in terms of the following metrics 1) peak-signal-to-noise ratio (PSNR); 2) structural similarity index (SSIM) [79] and 3) mean absolute error (MAE). PSNR is measured in terms of MSE and is still the most common quality measure for reconstructed images. PSNR of a reconstructed image is given by

$$\text{PSNR} = 20 \log_{10} \left(\frac{\text{MAX}_I}{\sqrt{\text{MSE}}} \right) \quad (5)$$

where MAX_I the maximum value of the pixel in the original image. A higher PSNR normally indicates higher quality

reconstruction. SSIM index [79] provides a quantitative assessment of the perceptual quality of the reconstructed image. We calculate these metrics on the test set of the *artificially corrupted* images and compare them to their corresponding clean ground truth images. We report the quantitative evaluation results for both Moon and Mars images using GI [68] and our method in Table IV, where the metric values for images from each of the clusters, the corresponding mean metric values for each clusters, and the metric values when the inpainting module was trained using all images together (i.e., not dividing the images into clusters) are provided. Our method outperforms the previous method [68] in terms of all the metric values for both the datasets. Moreover, the improvement in metric values (particularly PSNR (log-scale) values for inpainted images) over the baseline (e.g., 2.1 dB improvement for ‘mean of clusters’ and ‘all images’ in case of Moon images) demonstrates the validity of our proposed idea of leveraging the benefits of clustering our training and testing images based on their histogram distribution.

Qualitative Results: Figs. 7 and 8 show the qualitative performance of our proposed inpainting model when tested on *artificially corrupted images*. Here we can see that for both Moon and Mars dataset, previous inpainting methods, Patch-Match (PM) [48], and Generative Inpainting (GI) [68] generate significant artifact, however our method can predict the missing region that looks similar to the ground truth data. On the contrary, Fig. 9 demonstrates the qualitative inpainting results obtained from our inpainting model, when tested on *real corrupted test set* of Mars images. For example, as seen in Fig. 9 the crater or dark dune that are originally masked (first column) can be successfully augmented in shape (second column) by using our inpainting algorithm. This proves the generalization ability of our inpainting model to complete different landmarks on the planetary images. Fig. 10 compares the qualitative inpainting results achieved by our model on *real corrupted test set* of Mars images when compared with the previous inpainting methods PM [48], and GI [68]. It is clear that our method produces more photo-realistic results with seamless boundary transitions. We also show the visual comparison of the inpainted image quality when the inpainting module was trained using all images together (i.e., not dividing the images into clusters) versus images divided into clusters in Fig. 11. We can see the previous methods generate artifacts in the boundary causing the inpainted

TABLE IV
QUANTITATIVE EVALUATION RESULTS OF *SIMULATED/ARTIFICIALLY CORRUPTED* MOON AND MARS DATASET FOR DIFFERENT CLUSTERS AND ALL IMAGES TOGETHER (WHEN NOT DIVIDED INTO CLUSTERS) USING GENERATIVE INPAINTING (GI) [68], AND OUR METHOD

		Moon dataset					Mars dataset						
Method		Cluster 0	Cluster 1	Cluster 2	Mean of Clusters	All images	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Mean of Clusters	All images
PSNR ⁺	GI [68]	39.59	41.29	41.13	40.67	40.13	30.46	31.08	31.71	32.84	31.93	31.60	31.04
	Ours	42.32	43.69	40.98	42.33	40.23	33.02	33.46	34.01	33.72	33.32	33.51	33.42
SSIM ⁺	GI [68]	0.968	0.981	0.977	0.975	0.971	0.904	0.909	0.907	0.922	0.913	0.911	0.906
	Ours	0.989	0.989	0.984	0.987	0.982	0.928	0.931	0.933	0.926	0.928	0.930	0.928
ℓ_1 (%) ⁻	GI [68]	0.4	1.0	0.3	0.5	0.6	1.0	1.6	2.7	0.6	1.0	1.4	1.4
	Ours	0.1	0.3	0.1	0.2	0.3	0.7	1.1	1.7	0.5	0.8	0.9	0.9

The best results for each row is shown in bold. ⁻ lower is better. ⁺ higher is better

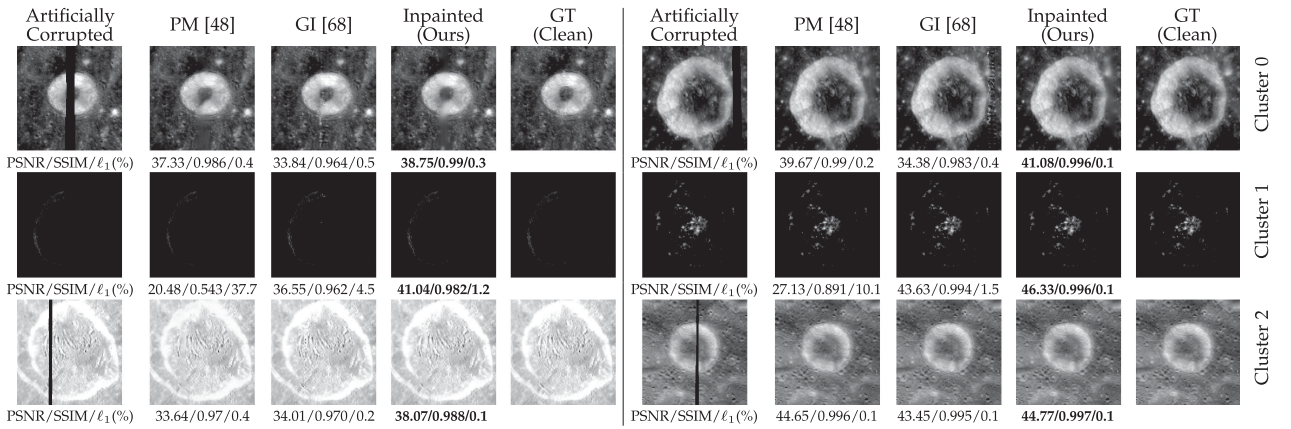


Fig. 7. Visual examples of semantic feature completion of *simulated/artificially corrupted* images on Moon dataset using different methods: PatchMatch [48], generative inpainting (GI) [68], and our method. Since we artificially corrupt the clean images, therefore we can consider the clean images as ground truth data in this case.

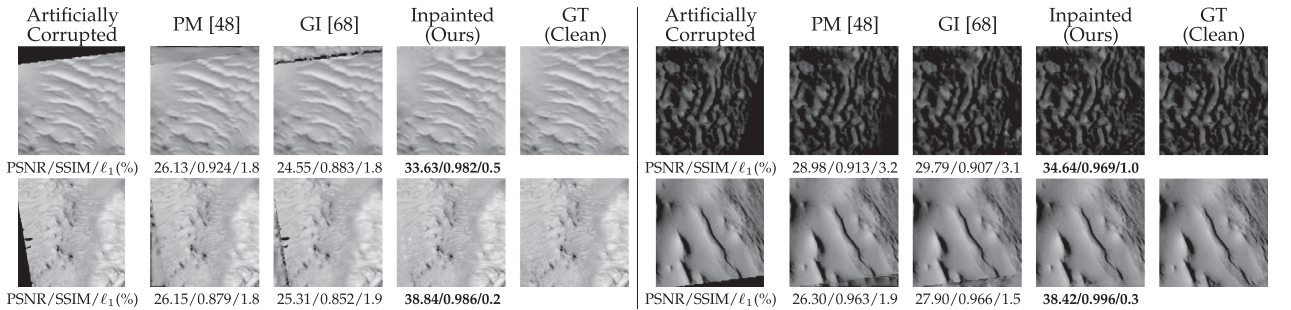


Fig. 8. Visual examples of semantic feature completion of *simulated/artificially corrupted* images on Mars dataset using different methods: PatchMatch [48], generative inpainting (GI) [68], and our method. Since we artificially corrupt the clean images, therefore we can consider the clean images as ground truth data in this case.

images to look unrealistic. On the contrary, our fine-tuned inpainting model can reconstruct an image with significantly fewer artifacts and a seamless boundary that looks more realistic to human eyes. Here, it should be noted that, in Figs. 9, 10, and 11 we are showing inpainting results on *real corrupted Mars images*, which do not have their corresponding ground truth images available. Hence the reconstruction quality of the real corrupted images can only be shown qualitatively, not quantitatively.

A. Ablation Study For Inpainting Results

Here, we analyze the effect of different loss components used to train our model and investigate if finetuning the inpainting model contributes to the better reconstruction of the real corrupted image. Table V reports the quantitative results achieved by our inpainting model using different loss components, i.e., by using only ℓ_1 loss, and by using ℓ_1 with adversarial loss, on the *artificially corrupted* Moon and Mars dataset. It can be seen

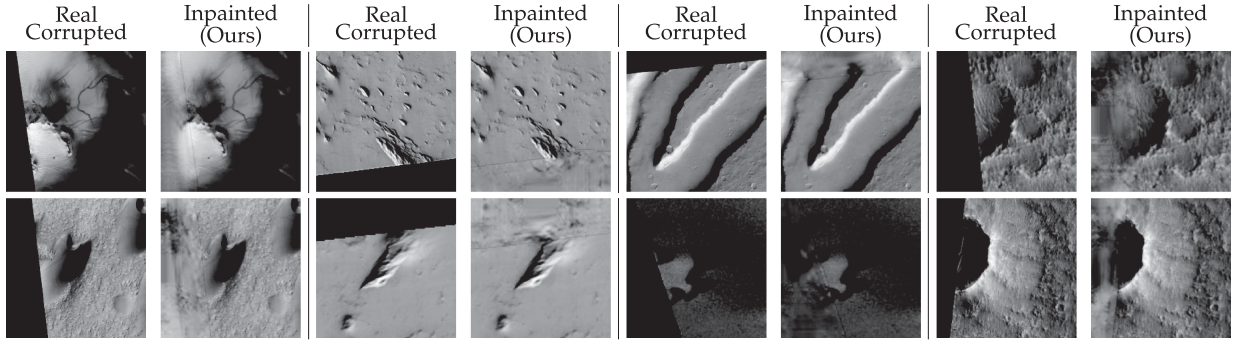


Fig. 9. Visual examples of semantic feature completion of the *real corrupted images* using our method. Since these are real corrupted data, we do not have ground truth images available in this case.

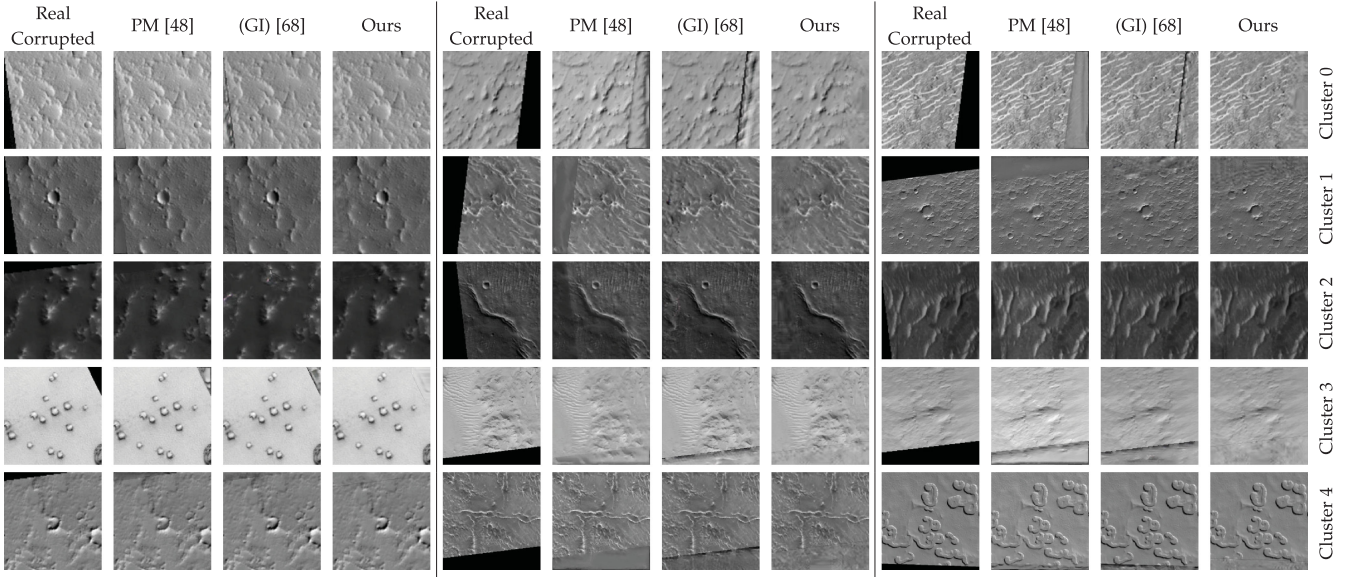


Fig. 10. Visual examples of semantic feature completion of *real corrupted images* on Mars dataset using different methods: PatchMatch [48], generative inpainting (GI) [68], and our method. Since these are real corrupted data, we do not have ground truth images available in this case.

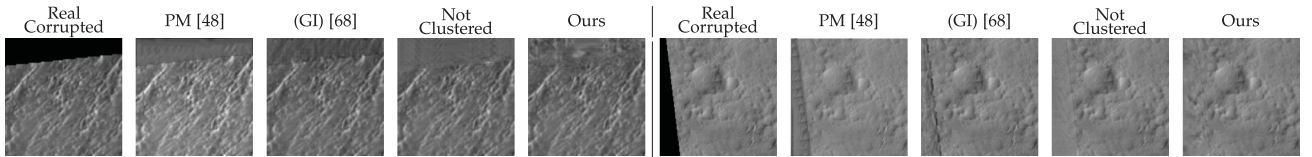


Fig. 11. Visual examples of semantic feature completion of *real corrupted images* on Mars dataset using different methods: PatchMatch [48], generative inpainting (GI) [68], inpainting results when all images are trained together (when not divided into clusters) and our method (when divided into clusters). Since these are real corrupted data, we do not have ground truth images available in this case.

that the adversarial loss component has a great contribution in improving the inpainting quality in terms of the metric values.

Fig. 12 shows the qualitative inpainting results for *real corrupted* Mars images using only ℓ_1 loss (without any finetuning), and ℓ_1 with adversarial loss (with and without finetuning). Clearly, if we train our model using only ℓ_1 loss, it generates a significant amount of artifacts, as seen in the second and sixth column of Fig. 12. Whereas, if we add the adversarial loss component, it improves the inpainting performance to a certain extent, as seen in the third and seventh column. Therefore,

we can draw a conclusion that for planetary image inpainting, adversarial loss is an essential ingredient. Next, we compare the performance of our model with and without finetuning. From the fourth and eighth column of Fig. 12, we can see that finetuning greatly improves the performance of our image inpainting model. Moreover, finetuning helps in completing the edges of the morphological structures or restoring the texture of the image. Therefore it is a crucial guidance to an image inpainting model for restoring artifacts that exist near the boundaries rather than in the center region of an input image.

TABLE V
QUANTITATIVE COMPARISON OF DIFFERENT COMPONENTS OF OUR METHOD ON MOON AND MARS DATASET

		Moon dataset					Mars dataset						
Method		Cluster 0	Cluster 1	Cluster 2	Mean of Clusters	All images	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Mean of Clusters	All images
PSNR ⁺	ℓ_1 loss	37.07	35.54	33.96	35.52	37.34	32.62	33.16	31.72	32.87	32.87	32.64	32.48
	ℓ_1 +adv loss	42.32	43.69	40.98	42.33	40.23	33.02	33.46	34.01	33.72	33.32	33.51	33.42
SSIM ⁺	ℓ_1 loss	0.979	0.96	0.97	0.969	0.979	0.923	0.926	0.909	0.924	0.925	0.921	0.930
	ℓ_1 +adv loss	0.99	0.99	0.98	0.99	0.98	0.928	0.931	0.933	0.926	0.928	0.930	0.928
ℓ_1 (%) ⁻	ℓ_1 loss	1.0	2.7	12.9	5.5	1.1	1.0	1.5	3.7	1.0	1.1	1.7	1.1
	ℓ_1 +adv loss	0.1	0.3	0.1	0.2	0.3	0.7	1.1	1.7	0.5	0.8	0.9	0.9

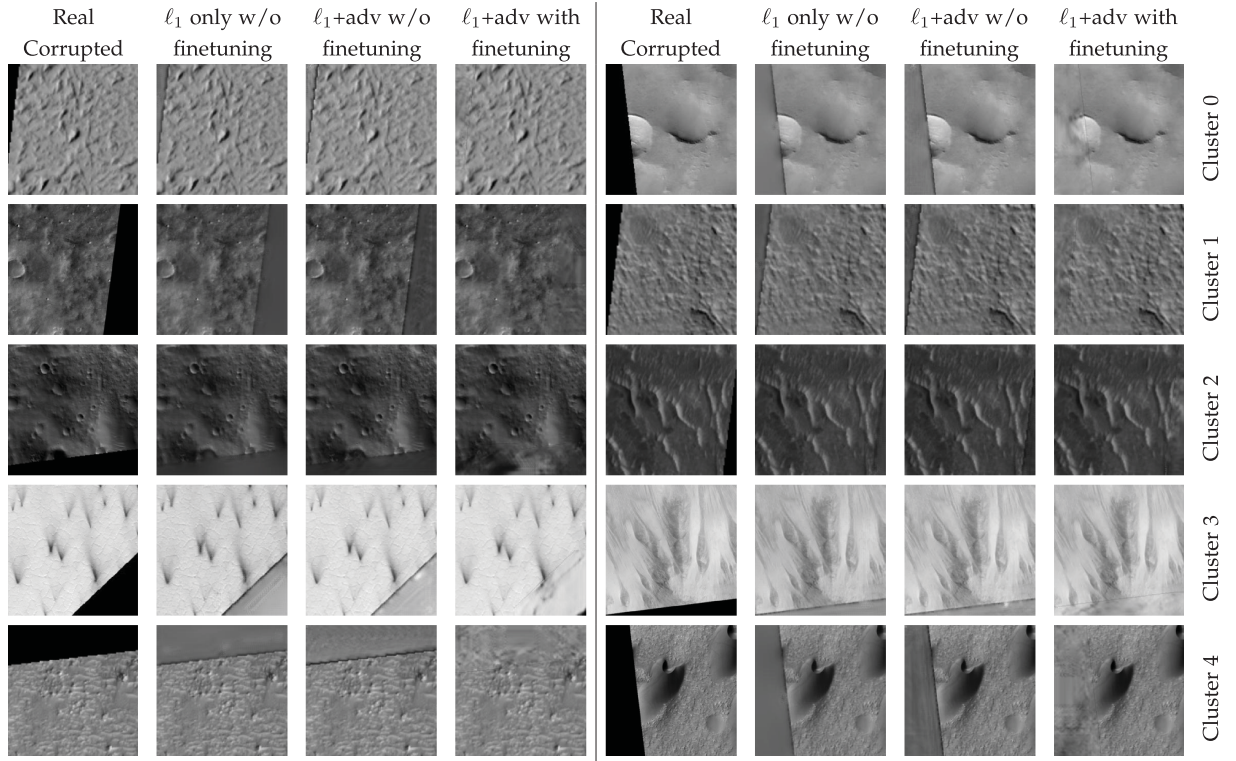


Fig. 12. Effect of different loss components and fine-tuning for each cluster. Here the first column shows the real corrupted images, second, third, and fourth column depict ℓ_1 loss without finetuning, (ℓ_1 +adversarial loss) without and with finetuning, respectively.

B. Classification Results

Here, we show the image classification results on the Mars dataset after applying our proposed image inpainting technique. We show the comparison of mean precision, recall, and F1-score on the original and inpainted images based on the same ResNet-50 and ResNet-101 classifier and the same data distribution in Fig. 13. It should be noted that our Mars dataset is highly imbalanced (with majority class Other having 61054 no of images, while minority class Impact Ejecta having 231 number of images) as shown in Fig. 4. Therefore, we show the classification performance of our model in terms of precision, recall, and F1-score (harmonic mean of precision and recall) metrics which are more appropriate to handle class imbalance.

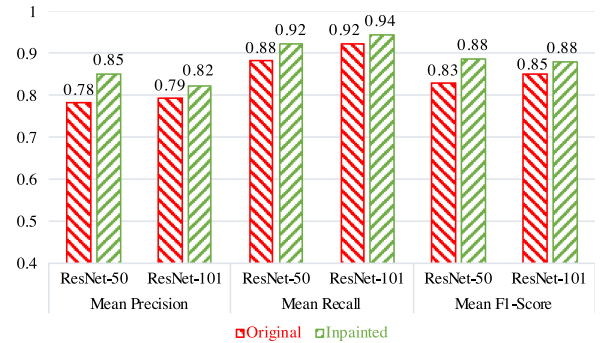


Fig. 13. Precision/Recall/F1 score for ResNet-50 and ResNet-101 in HiRISE inpainted dataset (ADAM).

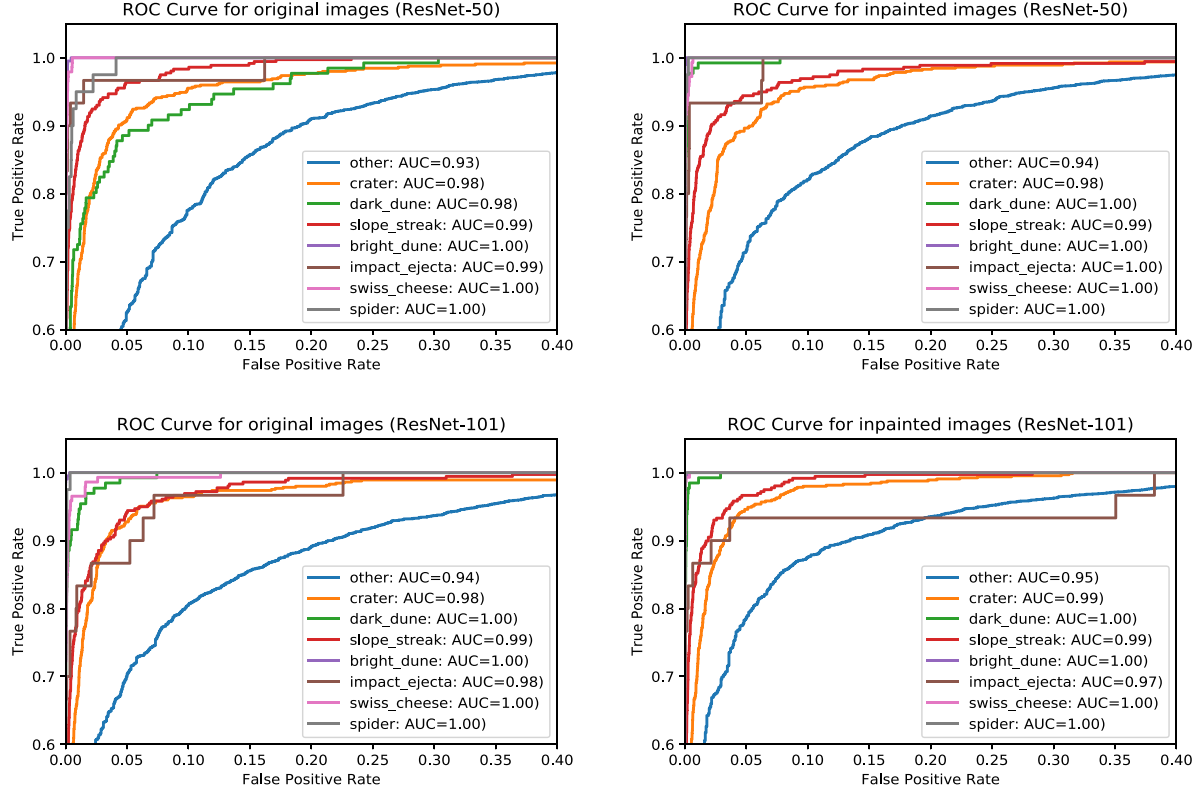


Fig. 14. ROC curves corresponding to ResNet-50 and ResNet-101 for original and inpainted HiRISE dataset. For most classes, AUROC for inpainted images is better than original images.

As can be seen in Fig. 13, for both ResNet-50 and ResNet-101 models, the inpainted image classification provides high mean F1-score of 0.88 outperforming the mean F1-score of 0.83 and 0.85 for original image classification. Also, there is a consistent improvement for inpainted images in the mean precision, and recall score by a large margin when compared to the original dataset. This reflects the fact that our model learns better features for classification tasks when trained on the inpainted images in comparison with original images (having partially masked interesting landmarks).

We also show the area under the curve (AUC) for receiver operating characteristics (ROC) for original and inpainted images for ResNet-50 and ResNet-101 models in Fig. 14. Since the AUROC metric is also appropriate to handle class imbalance, we report the AUC of all the classes for each case. We can see that for minority classes like Impact Ejecta, ResNet-50 performs better for inpainted images than original images as indicated by the higher area under ROC (AUROC) curve value.

Additionally, we provide the classification accuracy of our method in Table VI and compare our results with previous classification results [69] on the same Mars dataset. Since the network architecture is different in both cases, it cannot be considered as a one-to-one comparison. However, the classification accuracy is shown here as a reference for the reader for classification tasks on the same dataset.

It should be noted that classification accuracy is not an appropriate metric to measure the performance of the model, in case of such an imbalanced classification performed on a highly skewed

TABLE VI
CLASSIFICATION RESULTS USING OUR PROPOSED INPAINTED IMAGES

Original images [69]	Inpainted images	
	ResNet-50	ResNet-101
90.6%	94.00%	94.16%

dataset. Because in such case, high accuracy can be achieved by a nonexpert model by predicting only the majority class. Therefore, with the improved mean precision, recall, F1-score, and AUROC metric values achieved on the inpainted images (Figs. 13 and 14) for both ResNet-50 and ResNet-101 models, we prove the effectiveness of our proposed inpainting algorithm for better feature learning for classification.

VI. CONCLUSION

We present an adversarial training based image inpainting technique for planetary images to facilitate improved scientific discoveries. Our method predicts the unphotographed pixels by training a GAN-based model on input images belonging to various modes of histogram distribution. We find that our proposed inpainting algorithm helps the network to learn better features by augmenting the incomplete landmarks leading to better generalization. Our analysis reveals that by performing such image inpainting as a first step, we can boost the

classification performance with an improved F1 score. We believe that this work will benefit the planetary science community to analyze and explore the planetary images in a more efficient way. Our method can also be a helpful first step for planetary scientists to make more accurate location adjustments while making the mosaic of the planetary surface where the region is not illuminated by Sunlight such as the Polar region. In the future, we plan to extend this work by incorporating extra information such as digital elevation map or multispectral image data.

ACKNOWLEDGMENT

The authors would like to thank Dr. Makiko Ohtake, Planetary Scientist of the Institute of Space and Astronautical Science/Japan Aerospace Exploration Agency, and Dr. Yumi Iwashita of the Jet Propulsion Laboratory, National Aeronautics and Space Administration for sharing their expert knowledge on planetary images.

REFERENCES

- [1] <https://mars.nasa.gov/mro/mission/instruments/hirise/>
- [2] Y. Takizawa, "KAGUYA (SELENE) mission overview," *Proc. 26th Int. Symp. Space Technol. Sci.*, Jun. 2008.
- [3] Planetary Data Systems (PDS) Cartography and Imaging Sciences (IMG) (2018). Mars Reconnaissance Orbiter (MRO).
- [4] A. S. McEwen *et al.*, "Mars reconnaissance orbiter's high resolution imaging science experiment (HiRISE)," *J. Geophys. Res.: Planets*, vol. 112, no. E5, 2007.
- [5] K. J. Becker *et al.*, Near global mosaic of mercury: Eos, vol. 90, no. 52, Dec. 2009, Fall Mtg. Suppl., abs. pp. P21A–1189.
- [6] Q. Zhang *et al.*, "Missing data reconstruction in remote sensing image with a unified spatial temporal spectral deep convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 8, pp. 4274–4288, Aug. 2018.
- [7] H. Shen *et al.*, "Missing information reconstruction of remote sensing data: A technical review," *IEEE Geosci. Remote Sens. Mag.*, vol. 3, no. 3, pp. 61–85, Sep. 2015.
- [8] C. Zhang, L. Weidong, and T. David, "Gapsfill of SLCoff Landsat ETM+ satellite image using a geostatistical approach," *Int. J. Remote Sens.*, vol. 28, no. 22, pp. 5103–5122, 2007.
- [9] L. Zhang and W. Xiaolin, "An edge-guided image interpolation algorithm via directional filtering and data fusion," *IEEE Trans. Image Process.*, vol. 15, no. 8, pp. 2226–2238, Aug. 2006.
- [10] C. Yu *et al.*, "Kriging interpolation method and its application in retrieval of MODIS aerosol optical depth," in *Proc. 19th Int. Conf. Geoinformatics.*, 2011.
- [11] Q. Cheng *et al.*, "Inpainting for remotely sensed images with a multichannel nonlocal total variation model," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 1, pp. 175–187, Jan. 2013.
- [12] W. He *et al.*, "Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 178–188, Jan. 2015.
- [13] R. Mendez-Rial, C.-C. María, and M.-H. Julio, "Anisotropic inpainting of the hypercube," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 2, pp. 214–218, Mar. 2011.
- [14] P. Rakwatin, T. Wataru, and Y. Yoshifumi, "Restoration of Aqua MODIS band 6 using histogram matching and local least squares fitting," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 2, pp. 613–627, Feb. 2008.
- [15] M. Xu *et al.*, "Thin cloud removal from optical remote sensing images using the noise-adjusted principal components transform," *ISPRS J. Photogrammetry Remote Sens.*, vol. 149, pp. 215–225, 2019.
- [16] C. Zeng, H. Shen, and L. Zhang, "Recovering missing pixels for Landsat ETM+ SLC-off imagery using multi-temporal regression analysis and a regularization method," *Remote Sens. Environ.*, vol. 131, pp. 182–194, 2013.
- [17] B. Chen *et al.*, "Spatially and temporally weighted regression: A novel method to produce continuous cloud-free Landsat imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 1, pp. 27–37, Jan. 2016.
- [18] T.-Y. Ji *et al.*, "Nonlocal tensor completion for multitemporal remotely sensed images inpainting," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 6, pp. 3047–3061, Jun. 2018.
- [19] W. He *et al.*, "Remote sensing image reconstruction using tensor ring completion and total variation," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 8998–9009, Nov. 2019.
- [20] Q. Cheng *et al.*, "Missing data reconstruction for remote sensing images with weighted low-rank tensor model," *IEEE Access*, vol. 7, pp. 142339–142352, Sep. 2019.
- [21] Q. Zhang *et al.*, "Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning," *ISPRS J. Photogrammetry Remote Sens.*, vol. 162, pp. 148–160, 2020.
- [22] H. Roy *et al.*, "Lunar surface image restoration using U-net based deep neural networks," 2019, *arXiv:1904.06683*.
- [23] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [24] Y. LeCun *et al.*, "Handwritten digit recognition with a back-propagation network," *Adv. Neural Inf. Process. Syst.*, 1990.
- [25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
- [26] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2015, pp. 1–9.
- [27] K. He *et al.*, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2016, pp. 770–778.
- [28] G. Huang *et al.*, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2017, pp. 2261–2269.
- [29] J. Redmon *et al.*, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2016, pp. 779–788.
- [30] S. Ren *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural Inf. Process. Syst.*, 2015.
- [31] K. Zhang *et al.*, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.
- [32] C. Dong *et al.*, "Image super-resolution using deep convolutional networks," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, Jul. 2015.
- [33] S. Roth and M. J. Black, "Fields of experts: A framework for learning image priors," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Patt. Recognit.*, vol. 2, 2005.
- [34] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *J. Roy. Statist. Soc., Series C (Applied Statistics)* vol. 28, no. 1, pp. 100–108, 1979.
- [35] Q. Zhao, V. Hautamaki, and P. Fränti, "Knee point detection in BIC for detecting the number of clusters," in *Proc. Int. Conf. Adv. Concepts Intell. Vis. Syst.*, Berlin, Heidelberg: Springer, 2008, pp. 431–438.
- [36] J. Deng *et al.*, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2009, pp. 248–255.
- [37] C. Doersch, S. Singh, A. Gupta, J. Sivic, and A. Efros, "What makes paris look like paris?," *ACM Trans. Graph.*, 2012.
- [38] Z. Liu *et al.*, "Deep learning face attributes in the wild," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 3730–3738.
- [39] M. Cimpoi *et al.*, "Describing textures in the wild," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2014, pp. 3606–3613.
- [40] M. Bertalmio *et al.*, "Image inpainting," in *Proc. 27th Annu. Conf. Comput. Graph. Interactive Techniq.*, 2000, pp. 3606–3613.
- [41] C. Ballester *et al.*, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Process.*, vol. 10, no. 8, pp. 1200–1211, Aug. 2001.
- [42] M. Bertalmio *et al.*, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.
- [43] A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics," *Null.*, 2003.
- [44] T. Chan, "Local inpainting models and TV inpainting," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2001.
- [45] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. 28th Annual Conf. Comput. Graph. Interact. Tech.*, 2001, pp. 763–767.
- [46] I. Drori, D. Cohen-Or, and H. Yeshurun, "Fragment-based image completion," *ACM SIGGRAPH 2003 Papers.*, 2003, pp. 303–312.
- [47] D. Simakov *et al.*, "Summarizing visual data using bidirectional similarity," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2008, pp. 1–8.

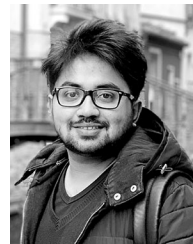
- [48] C. Barnes *et al.*, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph. (ToG)*, vol. 28, no. 3, ACM, 2009.
- [49] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.
- [50] L. Meur, M. E. Olivier, and C. Guillemot, "Hierarchical super-resolution-based inpainting," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3779–3790, Oct. 2013.
- [51] K. He and J. Sun, "Image completion approaches using the statistics of similar patches," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2423–2435, Dec. 2014.
- [52] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," *Adv. Neural Inf. Process. Syst.* 2012.
- [53] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 127–144, Jan. 2013.
- [54] R. Köhler *et al.*, "Mask-specific inpainting with deep neural networks," in *German Conference on Pattern Recognition*. Cham, Switzerland: Springer, 2014.
- [55] D. Pathak *et al.*, "Context encoders: Feature learning by inpainting," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.* 2016.
- [56] I. Goodfellow *et al.*, "Generative adversarial nets," *Adv. Neural Inf. Process. Syst.*, 2014.
- [57] R. Yeh *et al.*, "Semantic image inpainting with perceptual and contextual losses," vol. 2, no. 3, 2016, *arXiv:1607.07539*.
- [58] S. Chaudhury and H. Roy, "Can fully convolutional networks perform well for general image restoration problems?," in *Proc. Fifteenth IAPR Int. Conf. Mach. Vis. Appl.*, 2017.
- [59] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2007.
- [60] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Trans. Graph. (ToG)*, vol. 36, no. 4, pp. 1–14, 2017.
- [61] J. Yu *et al.*, "Generative image inpainting with contextual attention," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.* 2018.
- [62] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," 2017, *arXiv:1701.07875*.
- [63] I. Gulrajani *et al.*, "Improved training of wasserstein gans," *Adv. Neural Inf. Process. Syst.* 2017.
- [64] G. Liu *et al.*, "Image inpainting for irregular holes using partial convolutions," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 778–784.
- [65] Y. Song *et al.*, "SPFG-Net: Segmentation prediction and guidance network for image inpainting," 2018, *arXiv:1805.03356*.
- [66] W. Xiong *et al.*, "Foreground-aware image inpainting," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.* 2019, pp. 5833–5841.
- [67] K. Nazeri *et al.*, "Edgeconnect: Generative image inpainting with adversarial edge learning," 2019, *arXiv:1901.00212*.
- [68] J. Yu *et al.*, "Free-form image inpainting with gated convolution," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019.
- [69] K. L. Wagstaf *et al.*, "Deep Mars: CNN classification of mars imagery for the PDS imaging atlas," *Thirty-Second AAAI Conf. Artif. Intell.*, 2018.
- [70] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2016.
- [71] P. Isola *et al.*, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Patt. Recognit.*, 2017.
- [72] J.-Y. Zhu *et al.*, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.* 2017, pp. 2242–2251.
- [73] T. Miyato *et al.*, "Spectral normalization for generative adversarial networks," 2018, *arXiv:1802.05957*.
- [74] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," 2016, *arXiv:1607.08022*.
- [75] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," *Proc. Icml*, vol. 30, no. 1, 2013.
- [76] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.
- [77] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
- [78] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Adv. Neural Inf. Process. Syst.*, 2012.
- [79] Z. Wang *et al.*, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

- [80] N. Japkowicz, "The class imbalance problem: Significance and strategies," in *Proc. Intl. Conf. Artif. Intell.*, 2000.
- [81] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Trans. Knowl. Data Eng.*, vol. 21, no. 9, pp. 1263–1284, Sep. 2009.



Hiya Roy received the B.E. degree in electrical engineering from Jadavpur University, Kolkata, India, in 2012. She received the M.S. degree in electrical engineering from The University of Tokyo, Japan, in 2017. She is currently a Ph.D. candidate in the electrical engineering and information systems department at the University of Tokyo.

From 2012 to 2015 she worked as a Lead Engineer in Tata Power Company Limited, India. She worked as an intern at the NASA Jet Propulsion Laboratory, California Institute of Technology, and NEC Central Research Laboratories, Japan. She is a MEXT scholar from 2015 to 2020. Her research interests lie at the intersection of deep learning and planetary sciences.



Subhajit Chaudhury received the B.E. degree in electrical engineering from Jadavpur University, Kolkata, India, in 2012. He received the M.Tech. degree in electrical engineering from the Indian Institute of Technology Bombay, Mumbai, Maharashtra, in 2014.

He worked as a Researcher at NEC Research Laboratories, Japan from Oct 2014 to Mar 2017. He is currently working as a Research Scientist at IBM Research AI, Tokyo. Concurrently, he is a Ph.D. Candidate at the Department of Information and Commu-

nication Engineering, The University of Tokyo. His current research interests include reinforcement learning and computer vision. He is also a member of IEEE and ACM.



Toshihiko Yamasaki received the B.S. degree in electronic engineering, the M.S. degree in information and communication engineering, and the Ph.D. degree from The University of Tokyo in 1999, 2001, and 2004, respectively.

From 2004 to 2006, he was an Assistant Professor at Department of Frontier Informatics, Graduate School of Frontier Sciences, The University of Tokyo. He is currently an Associate Professor at Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo. His publication includes three book chapters, more than 60 journal papers, more than 160 international conference papers, more than 500 domestic conference papers. His current research interests include attractiveness computing based on multimedia big data analysis, pattern recognition, machine learning, and so on.

Dr. Yamasaki has received around 50 awards. He is a member of IEEE, ACM, IEICE, ITE, and IPSJ. He was a JSPS Fellow for Research Abroad and a Visiting Scientist at Cornell University from 2011 to 2013.



Tatsuaki Hashimoto received the Ph.D degree in electrical engineering from The University of Tokyo, Japan, in 1990. From 1990, he has been working for the Institute of Space and Astronautical Science (ISAS).

Currently, he is a Professor with ISAS, Japan Aerospace Exploration Agency and also Graduate school of engineering, the University of Tokyo. He has been working for research and development of spacecraft guidance, navigation, and control system.

He was with the Jet Propulsion Laboratory, NASA as a Visiting Scientist, in 2000. His research interest includes image processing for spacecraft navigation and lunar surface exploration. He is a member of AIAA, IAA, IEEE, SICE, etc.