博士論文　Doctoral Dissertation

東京大学
THE UNIVERSITY OF TOKYO

Deep Learning for Planetary Exploration: Improving image analysis capabilities under limited data resources
(深層学習の惑星探査への応用：
データリソース制限下での画像解析能力の向上)

Hiya Roy　ヒヤ ロイ

Student ID: 37-177268

Supervisor: Tatsuaki Hashimoto

Department of Electrical Engineering and Information Systems

Graduate School of Engineering

The University of Tokyo, Japan

This dissertation is submitted for the degree of

Doctor of Philosophy

December 2020

Deep Learning for Planetary Exploration:
Improving image analysis capabilities under limited data resources

Abstract

Deep Learning for Planetary Exploration: Improving image analysis capabilities under limited data resources

by

Hiya Roy

Doctor of Philosophy in Electrical Engineering and Information Systems

The University of Tokyo, Japan

Today, more than 25 space probes are actively exploring different planets and celestial bodies across the universe. Several others have either finished their mission or are planned to begin their journey. These spacecraft and planetary rovers carry several instruments and cameras onboard, which can capture a huge amount of planetary data. However, the interplanetary communication capacity through Deep Space Network is limited by the law of physics. Therefore, it is not always possible to send the entire planetary data, back to Earth for further analysis. This can create problems such as a missed scientific opportunity during planetary exploration. On the other hand, even if the acquired data is sent back to Earth, it is sometimes corrupted or have missing pixels because of data unavailability caused due to the technical limitation of the onboard instrument operation timing and satellite orbiter control.

There is a need for systems that can overcome these problems (i) by analyzing data onboard and returning much smaller sized meta-data to Earth, to improve the productivity of the mission, (ii) by predicting the "no-data" region of corrupted images to efficiently analyze the returned data to Earth, even if it is partially corrupted. This dissertation proposes machine learning algorithms to improve the planetary image analysis capabilities for future explorations under limited data constraints. These include (i) image-text retrieval algorithm which can enhance the onboard autonomy of future space missions by detecting objects seen in the image and sending only much smaller-sized metadata to Earth or can improve the efficiency of image search from a huge database by retrieving images of interest-based on textual queries, (ii) inpainting algorithms that can predict the missing regions on Mars or Lunar orbital images acquired by MRO or SELENE/Kaguya mission to enhance the data available for downstream tasks such as classification of interesting morphological features. Overall, this dissertation presents how machine learning can improve the image analysis capabilities in future space missions by overcoming the data constraints posed by various factors.

Acknowledgments

This dissertation is the final outcome of the last three and a half years of learning and hard work. I am indebted to my advisors, mentors, labmates, friends, and family who made it possible through their direct and indirect contributions. First, I would like to express my sincere gratitude to my supervisor Professor Tatsuaki Hashimoto for his excellent guidance and academic support. His help and encouragement enabled me to complete my Ph.D. course. I am grateful to my co-supervisor Professor Toshihiko Yamasaki for making me a part of his research team. He has taught me the importance of being well-read and guided me on the right track while pursuing ideas. I learned from him how to stay positive, be confident, and not to give up in the hardest times. I could always count on him in any constrained situation to help me find a way out. I am thankful to him for his invaluable guidance and financial support that helped me to complete this wonderful journey of my Ph.D. course.

I am grateful to the Government of Japan for supporting my Masters and Ph.D. studies by providing the Japanese Government MEXT scholarship. I am also thankful to the GSDM faculty members of the University of Tokyo for their constructive suggestions to obtain an interdisciplinary perspective of my research. I would like to thank Masahiro Ono for allowing me to work with his team-MAARS at the Jet Propulsion Laboratory (JPL), NASA. I am grateful to Yumi Iwashita for her advice and constant support during my stay at JPL. I would also like to thank Makiko Ohtake from JAXA for all the helpful discussions related to planetary science.

I am grateful to my peer Subhajit for collaborating with me and sharing his valuable research insights with me. Thank you for motivating me to conduct good research, sharing useful tricks to debug codes, and entertaining my numerous last-minute requests! Cheers to our futile attempts to visit Gotenshita! A special thanks to Sourav for all the thoughtful discussions about AI and Indian food!

Finally, I would like to thank my parents Mausumi Roy and Amit Kumar Roy for believing in my abilities. Thank you for your unconditional love, endless support, hard work, and innumerable sacrifices, that helped me reach where I am today.

Dedicated to my

PARENTS

with love and gratitude

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Humans have always looked at the night sky and marveled at the vastness of space. For over 60 years, they have ventured into space by sending more than 250 robotic spacecraft—and 24 humans, to discover information about the solar system and the distant stars [2]. Today there are more than 25 space probes that are actively exploring different planets and planetary bodies across the solar system [98]. Many more planetary exploration missions to even distant planets and planetary bodies are scheduled to be launched in the near future. Some of them are ESA's JUICE Explorer to Jupiter (2022) [3], JAXA's Martian Moons eXploration (MMX) mission to the two moons of Mars (Phobos and Deimos) (2024) [4], NASA's Dragonfly Mission to Saturn's icy moon (Titan) (2026) [5], etc. All these spacecraft (orbiting or roving) will be carrying a suite of sophisticated cameras, spectrometers, and other instruments, that can capture a plethora of data. However, due to the limited bandwidth and interplanetary communication through the deep space network [1], it will be increasingly difficult to return all the captured data to Earth. Similarly, a future rover can easily collect gigabytes to terabytes of data (e.g., high-resolution images, hyperspectral images, ground-penetrating radar observations) over a single operation cycle. However, it may not be able to downlink all the raw data due to the limitation in communication bandwidth. For example, the downlink capacity from the Curiosity rover to

[1] The Deep Space Network [77] enables deep space communication using three giant radio antennas in located in Goldstone (California), Madrid, and Canberra.

Earth is typically $\sim$ 500Mbit ($\cong$ 60MB) while data-intensive instruments, such as hyperspectral imagers and ground-penetrating radars, can easily produce hundreds of megabytes to gigabytes of data. Such situations may result in "missing science opportunities" meaning that science opportunities might be passed up by necessity or missed entirely simply because the data cannot be fully downlinked to Earth [138]. Therefore, in future, it will be impossible to return all the acquired data to Earth due to the limited communication bandwidth through Deep Space Network.

On the other hand, the data that has been returned by the mission might have some "no-data" region (regions that could not be captured by the onboard camera/sensors) for several reasons such as limitation in operation time of the instrument and satellite orbiter control, poor illumination because of lack of Sunlight in the Polar region, etc. Machine learning methods can provide solutions to enhance the planetary image analysis capabilities for future explorations under such limited data constraints. This dissertation proposes an image-text retrieval algorithm that can enhance the onboard autonomy of future space missions by detecting objects seen in the image or automatically retrieve images based on texts. This dissertation also presents inpainting algorithms that can predict the missing regions on Mars or Lunar orbital images acquired by Mars Reconnaissance Orbiter (MRO) or SELENE/Kaguya mission to enhance the data available for downstream tasks such as classification of interesting morphological features. More specifically, two image inpainting algorithms based on adversarial learning on planetary images are proposed, where the first approach uses only spatial domain information, whereas the second approach takes advantage of the frequency domain information along with the spatial domain information to selectively reconstruct the high-frequency components of the missing/no-data region.

The research works presented in this dissertation are interdisciplinary in nature requiring the background knowledge about machine learning and planetary exploration missions. The rest of this chapter provides the necessary background knowledge regarding machine learning (deep learning) techniques employed in this dissertation along with the necessary understanding of planetary science. The following sections explain the challenges associated with designing and implementing ML algorithms for planetary datasets, the motivation of this research, and finally, an outline of the research topics covered in different chapters of this dissertation.

1.1 Background: Machine Learning

Professor Tom Mitchell from Carnegie Mellon University defined machine learning as "the study of computer algorithms that allow computer programs to automatically improve through experience" or in his other words: "a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E" [123].

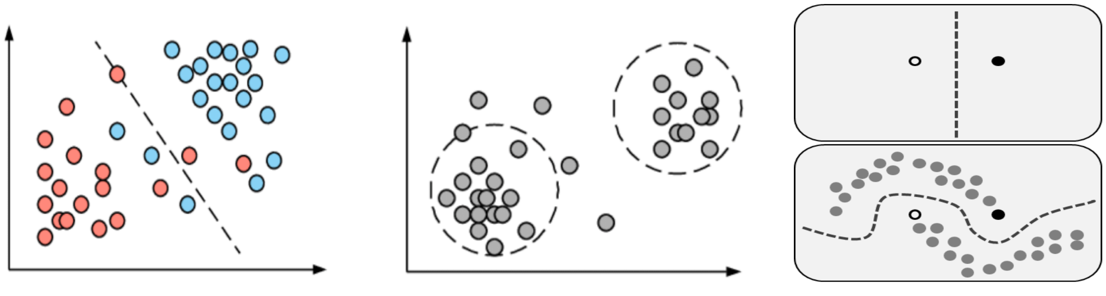
1.1.1 Types of Machine Learning Algorithms

Machine learning (ML) algorithms can be supervised, unsupervised, or semi-supervised depending on the availability of labeled datasets, as shown in Figure 1-1.

Supervised learning: In supervised learning, the algorithm learns a mapping between the input-output pairs of a labeled dataset. Examples of supervised learning algorithms are naïve bayes [125], decision tree [143], support vector machines [40], variants of neural networks such as convolutional neural networks [101, 97, 161, 165, 67, 73] etc.

Unsupervised learning: In unsupervised learning, the algorithm do not require labeled data, rather they learn to find structure in the data by clustering the data points. Examples of unsupervised learning algorithms are principal component analysis (PCA) [174, 84], *k*-means clustering [62], variants of neural networks such as autoencoders [119], restricted Boltzmann machines [71], and generative adversarial networks [59].

Semi-Supervised Learning: Semi-supervised learning is a class of supervised learning algorithm that makes use of a small amount of labeled data with a large amount of unlabeled data for training. An example of a semi-supervised learning algorithm is Ladder networks [146].



(a) Supervised learning (b) Unsupervised learning (c) Semi-supervised learning

Figure 1-1: Various machine learning algorithms. Image source: Google.

Weakly-Supervised Learning is also a class of supervised learning algorithm, where the supervision to label training data is provided using noisy, limited, or imprecise sources. For example, sentiment analysis based product reviews sold at different commercial websites such as Amazon, Rakuten etc. to provide feedback to the sellers can be done using weakly-supervised learning.

Machine Learning Problems: Amongst ML algorithms, supervised ML algorithms are the most popular and successful ones [19]. There are two major types of supervised ML problems are classification and regression.

Classification: In a supervised classification problem, the goal is to predict a class label from a pre-defined list of classes. In planetary science context, one example of a supervised classification problem is to classify different morphological features such as craters, dark dunes, bright dunes, slope streak, impact ejecta, etc. in the input orbital/planetary surface image [154, 172]. In this dissertation, a multi-class classification problem is solved (chapter 3), where the ML model is trained to distinguish images of various classes.

Regression: In a regression problem, the goal is to predict outputs that are real numbers and not class labels. In planetary science context, one example of a regression task is learning to predict the missing pixels in corrupted images as proposed in (Chapter 3, 4). In the appendix, aesthetic scores are predicted to decide the aesthetics quality of the image, by treating it as a regression problem.

It is to be noted that this dissertation employs only supervised ML algorithms.

1.1.2 Deep Learning

In recent years, deep learning (DL) has become the driving force for most of the ongoing work in the field of ML because of their outstanding performance in various fields ranging from vision-language-acoustics [19]. Some of the most popular DL algorithms are convolutional neural network (CNN) [101, 97, 161, 165, 67, 73], generative adversarial network (GAN) [59], auto-encoders [119], recurrent neural network (RNN)-long short-term memory network (LSTM) [72], deep belief network (DBN) [70] and many more. The solutions presented in this dissertation are also based on deep learning algorithms (particularly CNN and GAN) in the domain of computer vision. A brief technical overview of CNN and GAN is provided in the following subsection.

Convolutional neural network (CNN) is the biologically-inspired variant of feed-forward multilayer perceptron, which mainly consists of three kinds of layers: convolution layer, pooling layer, and fully-connected layer (similar to regular neural networks). These layers are stacked together to form a full CNN architecture. Unlike traditional machine learning techniques that require handcrafted feature design, CNNs have the ability to automatically learn hierarchical features from the input data. Because of this, deep neural networks can discover complex structures in high-dimensional data (such as images) ignoring irrelevant information and focusing on subtle but important information [100]. To emulate the behavior of the animal visual cortex, several neurally-inspired models [75, 103] have been proposed in the literature, where the output from each neuron is controlled by an "activation" function and is given by

$$y = h(\mathbf{W}^T\mathbf{x} + \mathbf{b}), \tag{1.1}$$

where $h(.)$ is a non-linear activation function, $\mathbf{x}$ is the input tensor having outputs from the previous layer neurons (or network input if it is the first hidden layer), $\mathbf{W}$ is a tensor having weight vectors, where w_{ij}^k represents the k^{th} weight connecting layer i to layer j, and $\mathbf{b}$ is a tensor of biases.

An activation layer or a non-linear layer is applied immediately after each convolution layer to introduce the non-linearity of the system which significantly improves the performance of a CNN for a particular task. Rectified linear unit (ReLU) [128] is one of the most notable non-saturated activation functions and is used in the proposed CNN architectures. The ReLU activation function is defined as

$$R(z) = max(0; z). \tag{1.2}$$

ReLU layer helps the network to train faster compared to sigmoid or tanh activation functions. It also helps to alleviate the vanishing gradient problem.

During training, the weight **W** and bias **b** parameters are continuously updated by backpropagating the gradient through the network such that the loss function is minimized using an optimization algorithm.

Loss function: Choosing an appropriate loss function for a specific task is of great importance. *Softmax* loss is the most commonly used loss function for predicting a single class from K mutually exclusive classes. The softmax function is the gradient-log-normalizer of the categorical probability distribution and is widely used in various probabilistic multiclass classification methods. The input to the softmax function is the output of the neural network which is normalized to a probability distribution over predicted output classes. In other words, the softmax function takes an input vector z of K real numbers, normalizes it into a probability distribution consisting of K probabilities proportional to the exponentials of the input numbers. The softmax function is defined as

$$softmax(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}. \tag{1.3}$$

where $i = 1, ..., K$. Summation of the probability values of all classes is equal to 1.

Optimization: The goal of optimization is to find a local optimum on the manifold of the parameter W that minimizes the loss function. The standard training of the CNN model is done using the backpropagation algorithm which uses gradient descent to update its parameters. Stochastic Gradient Descent (SGD) is the most popular algorithm to optimize neural networks which perform a parameter update for each training example $(x^i; y^i)$ as

$$\Theta = \Theta - \eta \nabla_\Theta J(\Theta; x^i; y^i). \tag{1.4}$$

In SGD, the convergence speed is controlled by the learning rate η and each parameter update in SGD is computed with respect to a mini-batch which could help to reduce the variance in the parameter update and can lead to more stable convergence. Other popular optimizers include Nesterov accelerated gradient [131], Adagrad [56], Adadelta [187], RMSprop [6], Adaptive Moment Estimation (Adam) [93].

Generative adversarial network (GAN) [59] is a type of neural network, where the goal is to learn the data distribution and be able to generate something that looks like the original data distribution. In a typical GAN architecture, a min-max optimization is solved for two networks, generator G and discriminator D, that learn to improve their performance by competing with each other i.e. G tries to trick D into classifying the generated fake data as real data by improving the generated output. Here the Generator G takes a noise vector z from $p(z)$ [where z is a sample from the probability distribution $p(z)$] and tries to generate an image x that resembles an image from the original data distribution. Generated image x is then fed into the discriminator $D(x)$ to classify as real or fake. The discriminator solves a binary classification problem by minimizing the binary cross-entropy loss during training. The objective function can be expressed as follows

$$\arg\min_G \max_D \mathcal{L}_{adv}(G, D) = \mathbb{E}_{\mathbf{x}\sim p_{data(x)}}[\log D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p(z)}[\log(1 - D(G(\mathbf{z}))], \tag{1.5}$$

where the discriminator tries to maximize the objective function and the generator tries to minimize the objective function. Therefore, by alternating between gradient ascent and descent, the network can be trained.

Recently GANs have become one of the most successful and promising framework in modeling complex data distributions and performed extremely well for various tasks e.g. image-to-image translation (CycleGAN [196], Pix2Pix [79]), generating very high-resolution images (1024 × 1024) (Progressive GAN [87]), recovering photo-realistic textures from heavily downsampled images (SRGAN [104]), image inpainting (Context Encoder [140]), synthesizing photorealistic images (GauGAN [139]) etc [2].

This dissertation presents new solutions using deep learning methods to solve challenges in planetary science and standard dataset consisting of grayscale and RGB color images.

1.1.3 Related Works of DL Algorithms in Planetary Science Applications

Recently there have several works on developing DL algorithms for solving different tasks using the planetary dataset of Mars, Moon, and Europa clipper. Rothrock et al. [154] developed a DL-based terrain classification algorithm called Soil Property and Object Classification (SPOC), based on DeepLab FCNN [35] implementation for rover missions. This algorithm can classify different terrain types (e.g., sand, bedrock) on both Mars orbital images (acquired by High Resolution Imaging Science Experiment (HiRISE) [120] camera on the Mars Reconnaissance Orbiter (MRO)) and surface images (images acquired by the navigation camera of Mars Science Laboratory (MSL) Curiosity rover [7]). Wagstaff et al. [172] proposed another classification algorithm based on AlexNet convolutional neural network [97] for Mars orbital [120] and surface [7] images. They also deploy these classifiers to the publicly accessible web interface called PDS Imaging Atlas [8], to enable the first content-based image search for NASA's Mars images. Delatte et al. [48] proposed a U-Net [152] based segmentation CNN model that can automatically detect craters in THEMIS thermal infrared Mars images [9]. Qiu et al. [142] developed SCOTI, a deep learning-based algorithm that takes an input image and generates captions explaining the geological content of the terrain image. This work is built upon the work of Xu et al. [181] having an encoder-decoder network where the encoder uses a VGG-19 [161] convolutional neural network for extract image features and the decoder uses a long short-term memory

[2] Various other cool applications of GANs (with creative namings) can be found at: urlhttps://github.com/hindupuravinash/the-gan-zoo

(LSTM) [72] network that produces a caption by generating the words sequentially based on the context. Ono et al. [137] conceptualized the idea of a content-based image search algorithm by introducing deep neural networks for PDS [8] and onboard datasets to make planetary images easily searchable from a large database. Kerner et al. [92] showed that it is possible to detect changes on the surface of planetary bodies by using a deep learning approach. They proposed a binary patch-level change detection algorithm based on transfer learning and nonlinear dimensionality reduction using convolutional autoencoders on various Mars images and Lunar Reconnaissance Orbiter Camera's (LROC) Narrow-Angle Camera (NAC) images [10]. All the datasets used in this work is made available to the public very recently [11]. Kerner et al. [91] also proposed a novelty detection algorithm for multispectral images obtained from MSL rover Mastcam camera with application to planetary exploration. They showed that autoencoders [119] trained with structural similarity (SSIM) [173] loss can detect morphological novelties that are not detected by PCA [84], GANs [59], and mean squared error (MSE) autoencoders. Wagstaff et al. [171] designed algorithms for developing onboard intelligence for the upcoming Europa Clipper mission, to support three scientific objectives: detection of thermal anomalies, compositional anomalies, and plumes; on the Europa dataset [12]. These research works show the gradual inclination of the research community to adopt deep learning algorithms for planetary science applications.

1.2 Challenges for Machine Learning in Space Applications

The main roadblock to a planetary exploration rollout is that the best computers are on Earth, but the best data is on those planets or planetary bodies. Since it is not possible to send all the acquired data to Earth because of limited communication bandwidth, therefore, one promising alternative is to perform onboard analysis/computation. However, the processor devices that is being used on-board Mars

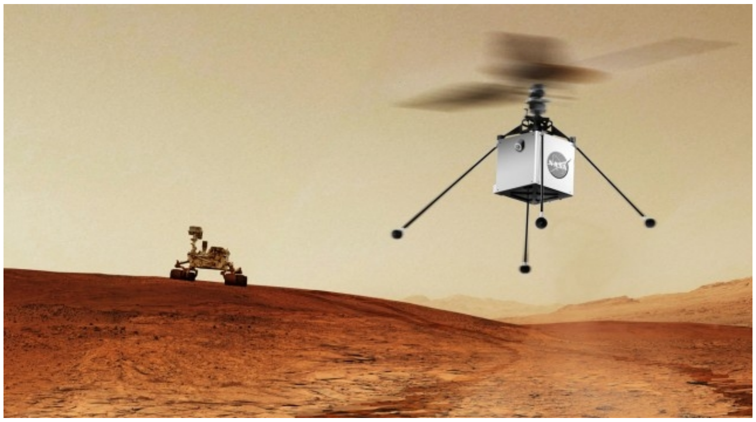


Figure 1-2: An artist's concept of the Mars Helicopter Scout (MHS), which will use modern system-on-a-chip (SoC) for onboard autonomy and will piggyback a ride onboard the Mars 2020 Rover to fly in the Martian for the first time. [138]. Image credit: NASA/JPL-Caltech.

Reconnaissance Orbiter [13], Mars rover Curiosity [7] as well as the most recent Mars 2020 rover Perseverance [14], is a RAD 750 developed by BAE Systems which is extremely reliable, resilient, and can function in solar flare-ravaged deep space. This chip is derived from the PowerPC 750 processor that dates back to the 1990s and is not suitable to perform huge computation required for deep learning models.

Recently, NASA and Air Force are developing a new generation of radiation-hardened (RAD-hard) multicore processor named High-Performance Spaceflight Computing (HPSC) which is qualified for space. HPSC would enable a vast suite of new mission concepts [54]. In the meantime, the Mars Helicopter Scout (Figure 1-2, the first vehicle to fly on Mars, uses Qualcomm's Snapdragon system-on-a-chip (SoC) for visual navigation [21]. The computation power of such modern commercial off-the-shelf (COTS) SoCs for mobile devices far surpasses the existing spacecraft computers such as the RAD750. For example, the Snapdragon 855 SoC has the ability to run deep neural networks in real-time with the support of its graphics processing unit (GPU), its digital signal processor (DSP), and its AI processor (AIP). Therefore, although there exist challenges, with the invention of HPSC and Snapdragon, it is possible to think about performing onboard computation on space computers that will be much faster ~2.2 GHz [14].

Compared to the standard datasets such as MNIST [103], Fashion-MNIST [176], CIFAR-10 [96], ImageNet [50], MS-COCO [108] etc. that are used to benchmark various machine learning algorithms, datasets available for solving problems related to planetary science are smaller in size and harder to label. Often the planetary dataset is not publicly available or even if it is publicly available, sometimes domain knowledge is required to understand the nomenclature or the typicalities related to that dataset. Also, planetary datasets are usually not labeled. Therefore, if they require labeling for a particular kind of task, it has to be done by planetary scientists or by people who have domain knowledge. Moreover, these datasets cannot be crowd-sourced to the public for labeling because of affiliation constraints or lack of expertise of the common public.

1.3 Motivation of This Research

The motivation of this research is to propose ML solutions to overcome the problem of limited data resources which occurs because of two reasons (i) limited bandwidth and inter-planetary communication through deep space network and (ii) technical limitation of the onboard instrument operation timing and satellite orbiter control that causes partial corruption in returned data to Earth from various planetary exploration missions.

1. Motivation to solve the first problem: In the future, planetary exploration missions are going to be in distant planetary bodies such as Jupiter's moon-Europa, or Saturn's icy moon-Titan, which will have even more limited bandwidth/communication opportunities than those at Moon or Mars. Developing onboard intelligence can play a crucial role in enabling these missions to infer interesting scientific insights by themselves (i.e. without returning the entire acquired data to Earth) [171]. Onboard intelligence using ML algorithms can be developed in several ways: (i) by detecting objects in the observed image and sending only smaller-sized meta-data, instead of the entire high-resolution image, (ii) by prioritizing the most novel or salient observations in addition to the targeted observations and sending only the prioritized data to Earth for sci-

entists to investigate etc. This way, not only the problem of limited bandwidth can be handled but also the chances of missed scientific opportunities can be reduced.

2. Motivation to solve the second problem: Image inpainting can be a helpful first step for planetary scientists for further analysis such as to automatically classifying or recognizing interesting morphological features in the planetary surface (improve classification performance), 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; or to improve the landing site candidate selection efficiency, etc.

3. Other motivations: Automated ML algorithms can help planetary scientists to analyze complex datasets in a faster and efficient way. This can be done by automating time-consuming tasks for humans such as finding images of interest, based on textual queries (image retrieval).

1.4 Thesis Outline

Chapter 1 describes the background knowledge related to machine learning and deep learning about certain methods employed in this dissertation; challenges associated with applying ML in planetary science, and finally the motivation of the research works presented in this dissertation.

Chapter 2 presents a supervised machine learning approach to enhance onboard analysis capabilities by detecting objects seen in the image and sending this smaller sized meta-data back to Earth. This will not only help to overcome the problem of limited bandwidth but also will help to retrieve images from a large database based on query text. This study required curation of new planetary datasets along with labeling of each image (annotating objects and corresponding captions for the entire image) for developing a new machine learning method for planetary image retrieval based on textual query. The novel contribution is the specially designed “MarsDetect” object detection dataset for Mars surface images, which enabled the proposed text-image retrieval algorithm to achieve superior performance compared to previous baseline methods.

Chapter 3 proposes an adversarial training based image inpainting technique to restore unphotographed/no-data region on planetary images (that are already returned data from planetary exploration missions to Earth) to facilitate improved scientific discoveries by enhancing the data availability for various downstream tasks. The novel contribution made in this work is the proposed idea of clustering the planetary images into several modes of histogram distributions, which helps to prevent the mode-collapse problem in generative models and encourage the network to reliably generate samples from each cluster. The proposed image inpainting algorithm predicts the "no-data" region of a corrupted image and helps the network to learn better features to improve the classification accuracy of interesting landmarks on planetary images. By using diverse planetary datasets of Moon and Mars, it is shown that this approach is applicable to various planetary images.

Chapter 4 extends this ideology of image inpainting by incorporating a frequency domain component that enables the network to use both frequency and spatial information to predict the missing region of an image. Furthermore, it is shown that the proposed frequency-based image inpainting algorithm also works well for standard datasets that look fundamentally very different from planetary images. This shows the generalization ability of the proposed algorithm. To the best of our knowledge, this is a novel work because it is the first attempt to solve the image inpainting problem by using frequency-domain information that is applicable to both planetary images and standard images.

The final chapter summarizes the work presented in Chapters 2–4, presents some ideas for future research directions, and provides some philosophy that if practiced, could bolster long-term interdisciplinary research of ML and planetary science.

Overall, this thesis leverages the recent progress of computer-based processing in the field of machine learning (particularly deep learning technology) and shows that computers can make decisions/conclusions on the subjects where human decisions used to be necessary. However, the main roadblock to applying ML algorithms for space applications is that the best quality/maximum quantity of data is with the spacecraft orbiting or roving the planets and planetary bodies; whereas the best computers are on the Earth. Therefore, either the best computers have to be sent to different planets or the entire data has to be brought back to Earth for further analysis. Whereas the first problem seems to be solvable by developing the latest radiation-hardened computers for spacecrafts (this is still under research or experimentation phase)), the second problem is limited by the laws of Physics. Therefore, this research attempts to solve these problems by proposing various deep-learning algorithms. Putting together, the philosophy in this thesis is that although there exist several limitations, it is possible to overcome them by leveraging the benefits of ML technology for better planetary exploration in the future.

Chapter 2

Retrieving Interesting Planetary Images based on Captions

This chapter deals with solving the problem of limited inter-planetary communication opportunities through deep space network. As the number of acquired images continues to grow exponentially with each ongoing mission, it becomes increasingly difficult to return all the data to Earth because of limited bandwidth between Earth and other planets and planetary bodies. This chapter proposes an image-text matching algorithm to enhance the onboard analysis capabilities of the orbiting or roving spacecraft by detecting objects in the observed image and sending only much smaller-sized metadata to Earth that describes the image. The proposed image-text matching algorithm is also helpful to retrieve images of interest (images with desired geologic and/or non-geologic features), from a large database based on query text. For experimental purposes, the Mars surface images captured by the navigation camera of NASA's Mars science laboratory (MSL) Curiosity rover is used. Experimental results demonstrate that the proposed method can accurately detect objects on the Mars surface images. To develop a machine learning algorithm for planetary image retrieval based on text/captions, a labeled Mars image dataset (MarsDetect) is curated by drawing bounding boxes around objects/regions of interest corresponding to the captions of the images.

2.1 Introduction

Over the last few decades, planetary missions have acquired a huge amount of image datasets. For example, over 25 million images have been acquired by NASA's Mars rovers Spirit (MER-A), Opportunity (MER-B), and Curiosity (MSL). [142]. As the number of acquired images continues to grow with each ongoing mission, it becomes increasingly difficult to return all the data to Earth for further analysis because of the limited bandwidth constraints through a deep space network. Let us consider the current situation in Mars: the downlink capacity from the curiosity rover to Earth is typically $\sim$ 500Mbit ($\cong$ 60MB) while sophisticated data-intensive instruments such as hyperspectral imagers, can easily produce hundreds of megabytes to gigabytes of data. Therefore, it is already difficult to return this huge amount of data to Earth [138]. With the rapid development of more sophisticated instruments, a future rover will easily collect gigabytes to terabytes of data (e.g., high-resolution images, hyperspectral images) over a single operation cycle. Moreover, in near future, many more planetary missions are planned even to distant planetary bodies such as Jupiter's moon-Europa, or Saturn's icy moon-Titan, where the communication opportunities will have even more limited than those at Moon or Mars. Therefore, it can be safely argued that there is an urgent need for developing onboard intelligence to solve this problem of limited communication bandwidth through deep space network (DSN). In this context, the goal is to develop onboard intelligent systems that can reduce the missed scientific opportunity.

Recently DL algorithms have shown tremendous success in several vision-based tasks such as image classification [67, 73, 161, 165], object recognition [147, 149], etc. Therefore, in this work, DL algorithms are employed for solving this image-text matching task, which stands for searching an image from a database by generating an image description and matching it with the textual query by semantically aligning their latent representation. The recently developed object detection algorithm [148] has been leveraged to detect objects of interest in the planetary images and return the encoded representation of image regions at the object level to Earth instead of actual images and to effectively tackle the problem of limited bandwidth. There are a few works in the literature that employs DL algorithms for planetary image analysis tasks. Rothrock et al. [154] developed a deep learning-based terrain classification algorithm called soil property and object classification (SPOC), based on DeepLab FCNN [35] implementation for rover missions. This algorithm can classify different terrain types (e.g., sand, bedrock) on both Mars orbital [120] and surface [7] images. Wagstaff et al. [172] proposed another classification algorithm based on AlexNet convolutional neural network [97] for Mars orbital [120] and surface [7] images that can classify objects. Qiu et al. [142] proposed SCOTI, a deep learning algorithm for generating captions (based on the work of Xu et al. [181]) that takes an input image and generates captions explaining the captured image. In this work, the motivation is similar to that of Qiu et al. [142], where they generate captions by extracting general image features such as edge, shapes, or geometry of objects. This chapter attempts to solve this problem by detecting multiple objects at once with their actual object labels and then associating them with the caption, thus finding the image-text similarity score. A lot of research works have been done to solve image-text matching which plays an important role in bridging two domains - vision and language. A seminal work in this direction was done by Karpathy et al. [86] who proposed detecting objects in an image based on R-CNN [58], encoding these image regions corresponding to objects and then computing the aggregated similarity scores for all possible region-word pairs to find the image-text similarity. Drawing inspiration from [86] this chapter proposes an image-text matching algorithm for planetary images which is done in two stages: (i) detecting objects in the image, (ii) matching words in the textual query with the corresponding representation of the objects detected in the image. The main contributions in this chapter can be summarized as follows:

1. An object detection based text-image matching algorithm is introduced to detect objects of interest in planetary images. The object detection algorithm can generate smaller-sized metadata representations (such as the object bounding box locations, labels along with confidence values) onboard which can be returned to Earth instead of the entire image. This can solve the problem of

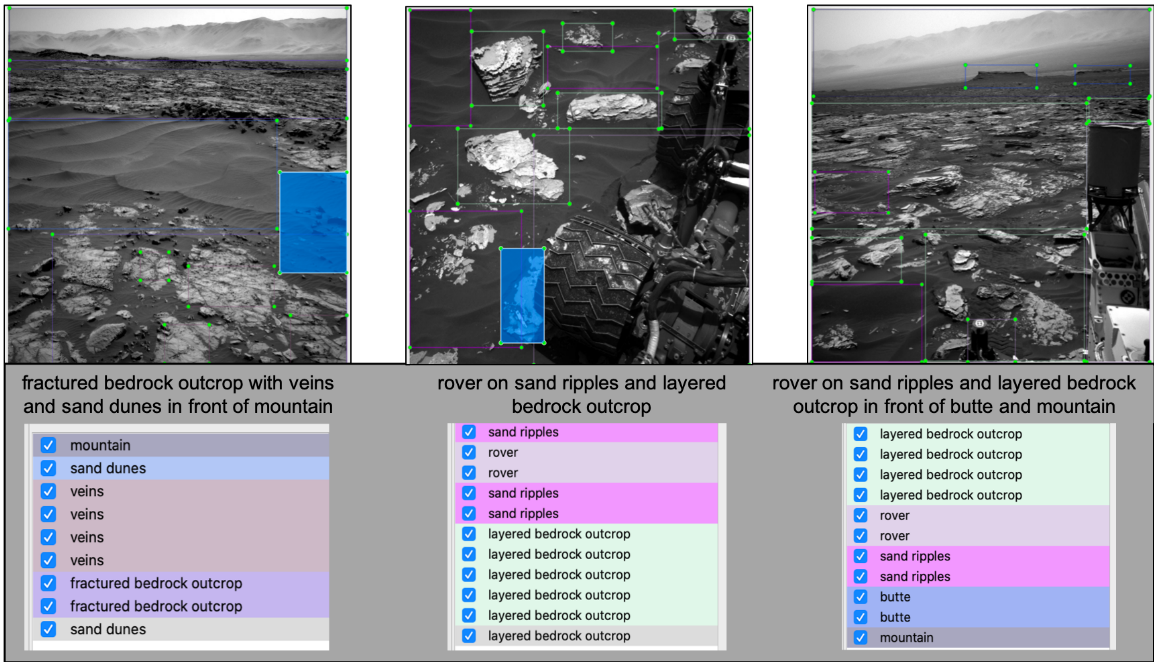
limited bandwidth through the deep space network and enhance the onboard image analysis capabilities of the orbiting/roving spacecraft.

2. The developed algorithm is also useful for other ground applications such as image retrieval. Experimental results show that the proposed image-text matching algorithm can efficiently retrieve images of interest (images with desired geologic and/or non-geologic features), from a large database based on query text. The novel contribution is the specially designed "MarsDetect" object detection dataset for Mars images, which enabled the proposed text-image retrieval algorithm to achieve superior performance compared to previous baseline methods.

2.2 Related Works

2.2.1 Object Detection

Object detection is a widely studied topic and a lot of revolutionary works have been done on this topic. Apart from several hand-engineered features such as SIFT [112], HOG [43] used for object detection, several noteworthy deep CNN-based object detectors have been proposed in the literature, some of which are described in this subsection. Girshick et al. [58] proposed R-CNN that adopted a region proposal-based strategy [169] and use a deep CNN to classify the scale-normalized object proposals. Faster R-CNN [149] utilizes a Region Proposal Network (RPN) that shares the convolutional features of the entire image with the detection network in an efficient manner compared to R-CNN and Fast R-CNN. The necessity of such region proposal was eliminated in SSD [110] and YOLO [147] object detectors. While the SSD network exploits the pyramidal feature hierarchy of CNN and uses the varying size of convolutional layers, YOLO utilizes two fully connected layers. YOLO treats object detection as a regression problem and uses a single neural network to predict bounding box locations along with their confidence and corresponding class probability scores, directly from the image. YOLO is known for its ability to work extremely fast in real-time.



fractured bedrock outcrop with veins and sand dunes in front of mountain

- ☑ mountain
- ☑ sand dunes
- ☑ veins
- ☑ veins
- ☑ veins
- ☑ veins
- ☑ fractured bedrock outcrop
- ☑ fractured bedrock outcrop
- ☑ sand dunes

rover on sand ripples and layered bedrock outcrop

- ☑ sand ripples
- ☑ rover
- ☑ rover
- ☑ sand ripples
- ☑ sand ripples
- ☑ layered bedrock outcrop
- ☑ layered bedrock outcrop
- ☑ layered bedrock outcrop
- ☑ layered bedrock outcrop
- ☑ layered bedrock outcrop
- ☑ layered bedrock outcrop

rover on sand ripples and layered bedrock outcrop in front of butte and mountain

- ☑ layered bedrock outcrop
- ☑ layered bedrock outcrop
- ☑ layered bedrock outcrop
- ☑ layered bedrock outcrop
- ☑ rover
- ☑ rover
- ☑ sand ripples
- ☑ sand ripples
- ☑ butte
- ☑ butte
- ☑ mountain

Figure 2-1: Labeling interface along with captions and labels.

2.2.2 Image to Text Matching

There have been a lot of studies related to image-text matching that maps images and sentences to a common embedding space. Some previous methods [86, 133] take advantage of the bottom-up attention [31, 88] by drawing inspiration from the human visual system. Karpathy et al. [86] attempts to solve this problem by combining CNN features (over image regions), and bidirectional RNN (over sentences) and tries to align image and text modalities. Niu et al. [133] presented a Hierarchical Multimodal LSTM (HM-LSTM) model that maps image regions and noun phrases into a shared embedding space. Other image-text matching algorithms [74, 129] use conventional attention-based models [181]. Huang et al. [74] introduced a multimodal context-modulated attention scheme to selectively attend to a pair of instances appearing in both image and sentence. Nam et al. [129] proposed another attention based network that allows visual and textual attention mechanisms to estimate the similarity between images and sentences. Song et al. [163] used a multi-head self-attention and residual learning-based approach that combines global and local context to find the visual-semantic embedding for text-image retrieval.

2.3 Experimental Data: Bounding Box Annotation

In this work, Mars images acquired by the Mars Science Laboratory (MSL) Curiosity rover are used. Since a supervised learning algorithm requires a labeled dataset and the proposed algorithm is also based on supervised learning, it is necessary to label the dataset. The Mars dataset used in this chapter is available at NASA-PDS [15]. Amongst all the available Mars images, only 640 images are annotated (with captions) by JPL planetary scientists. Here, for experimental purposes, only 640 number of annotated images are used. The noun phrases from these captions are extracted and 20 different geological features present in the Mars images are identified as followed: regolith, bedrock, sedimentary bedrock, bedrock outcrop, fractured bedrock outcrop, layered bedrock outcrop, boulder outcrop, butte, float rocks, slope, mound, mountain, sand dunes, sand ripples, clasts, rover, rover tracks, alteration halos, veins, and layered strata. These identified objects in the images are then annotated by drawing bounding boxes and thus a labeled image dataset named “MarsDetect” is curated. This MarsDetect dataset contains the bounding box locations with object labels and the corresponding captions. A graphical image annotation tool “labelimg” [16] based on Python and Qt is used for drawing bounding boxes in different images based on the objects present in the image. The annotations are saved in YOLO format to use for the object recognition task. Figure 2-1 shows some examples of annotated images (with bounding boxes and captions) on the graphical interface. It is to be noted that the captions for each image are given by JPL planetary scientists, however, the object label list for each image is created by us, by matching them with the corresponding image captions.

2.4 Workflow

The detailed workflow of the data collection and model retraining pipeline is shown in Figure 2-2. At first, the terrain images are collected and annotated. The object-wise bounding box annotation is done by us and the caption annotation is given by

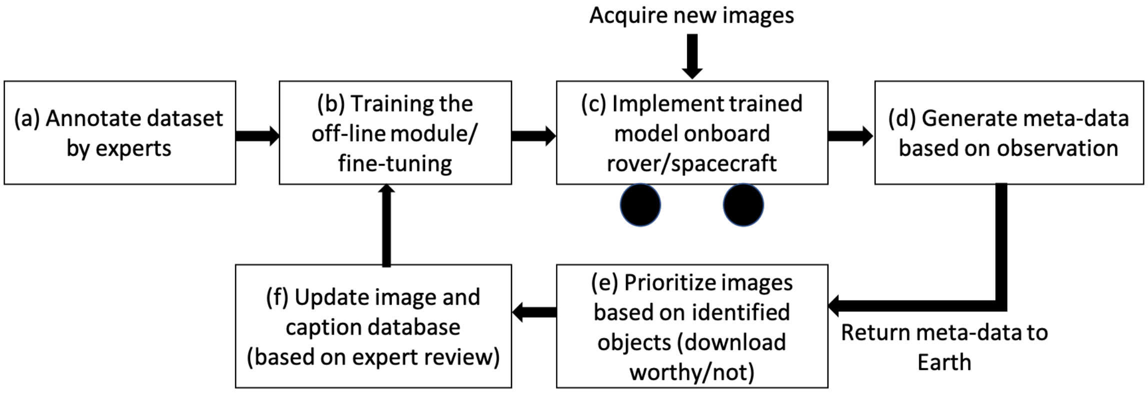

Acquire new images
(a) Annotate dataset by experts
(b) Training the off-line module/ fine-tuning
(c) Implement trained model onboard rover/spacecraft
(d) Generate meta-data based on observation
Return meta-data to Earth
(e) Prioritize images based on identified objects (download worthy/not)
(f) Update image and caption database (based on expert review)

Figure 2-2: The workflow of the data collection and model retraining pipeline.

the JPL planetary scientists. After annotating the dataset in step (a), the proposed model is trained in such a way that it learns the latent semantic alignment between salient objects (e.g. float rocks, sand dunes, mountain, etc.) and the corresponding words in the caption. After the model is trained, it can be implemented onboard the rover or orbiting spacecraft to perform onboard image analysis and can (d) generate meta-data such as the bounding box co-ordinates of different objects with confidence scores, for newly acquired images. The generated meta-data for all images can be (e) returned to Earth for scientists to analyze the priority of downloading a particular image, based on the observation content. Finally (f) the image and caption database can be updated to retrain/fine-tune the model to make it applicable to the acquired images from the latest missions.

2.5 Proposed Method

In this work, the goal is to infer the similarity between an image and its caption by mapping words and object label image features into a common embedding space. This is performed in two stages, (i) first salient objects (or objects of interest) are detected in the planetary images, and then an encoded representation of image regions at object level is generated which is much small than the actual image size, and (ii) bring this object-level representation to an embedding space that is similar to the word feature space and compute the aggregated similarity scores for all possible combinations of region-word pairs to infer the image-text similarity.

Let us consider a database of images and corresponding captions given as $(x_i,\ y_i)$. The goal is to find the closest matching image corresponding to an input text query which is represented as y_{query}. For this purpose, a score function $s_i = f_{sim}(y_i, y_{query})$ is developed, which finds the similarity score between the query text and all the captions in the database. The task is to find the corresponding image x_k, such that

$$K = \arg\min_j f_{sim}(y_i, y_{query}). \tag{2.1}$$

Let us assume a constraint such that the true caption for the images are not available in the database and must be inferred from the images in the database. Thus this problem is formalized as a training task with the training image having the true caption information, $\{x_i^{train}, y_i^{train}\}_{i=1\to N}$ and the test set is the database of images (where the text-to-image retrieval task has to be performed) only has the images, given as $\left\{x_j^{test}\right\}_{j=1\to M}$. Therefore, this task consists of the two following subtasks: (i) Salient object detection for images in the search database, (ii) finding text-image similarity.

2.5.1 Salient object detection

In this first step, the objective is to detect the salient objects in the search database images $\left\{x_j^{test}\right\}_{j=1\to M}$. Therefore, the proposed network has to be trained using the object bounding box annotations in the training images given as $\{x_i^{train}, y_i^{train}\}_{i=1\to N}$. To obtain the object categories, the annotated captions given by scientists for each image are used. From each caption, the unique set of noun-phrases are extracted. After that, some rare adjectives are manually removed. Thus 20 different geological features/classes are obtained. Next, the “MarsDetect” dataset is curated by extensively annotating the images based on the aforementioned 20 geological features. For each image, multiple object annotations $\{a_0, a_1, ...a_{N_i}\}$ (where N_i is the number of annotated objects in the image) are created, using which the object detector is trained. The proposed object detection module for the Mars dataset is described in

Table 2.1: Object detection network architecture for Mars images.

	Type	**Filters**	**Size**	**Output**
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3/2	128 × 128
	Convolutional	32	1 × 1	
1×	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3/2	64 × 64
	Convolutional	64	1 × 1	
2×	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3/2	32 × 32
	Convolutional	128	1 × 1	
8×	Convolutional	256	3 × 3	
	Residual			32 × 3
	Convolutional	512	3 × 3/2	16 × 16
	Convolutional	256	1 × 1	
8×	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3/2	8 × 8
	Convolutional	512	1 × 1	
4×	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

the following subsection.

Object Detection Module: Here, the YOLO-v3 [148] object detection module is employed, which is an improved version of the YOLO-v1 [147] and is one of the most popular object detection algorithms. In this case, the YOLO-v3 network pre-trained on MSCOCO [108] dataset is used. YOLO-v3 [148] predicts bounding boxes at 3 different scales where feature extraction is done from each of these scales similar to feature pyramid networks [107] concept. This provides an output of 3-d tensor encoding the bounding box locations, objectness score, and the corresponding class probability scores. The objectness score for each bounding box is calculated using

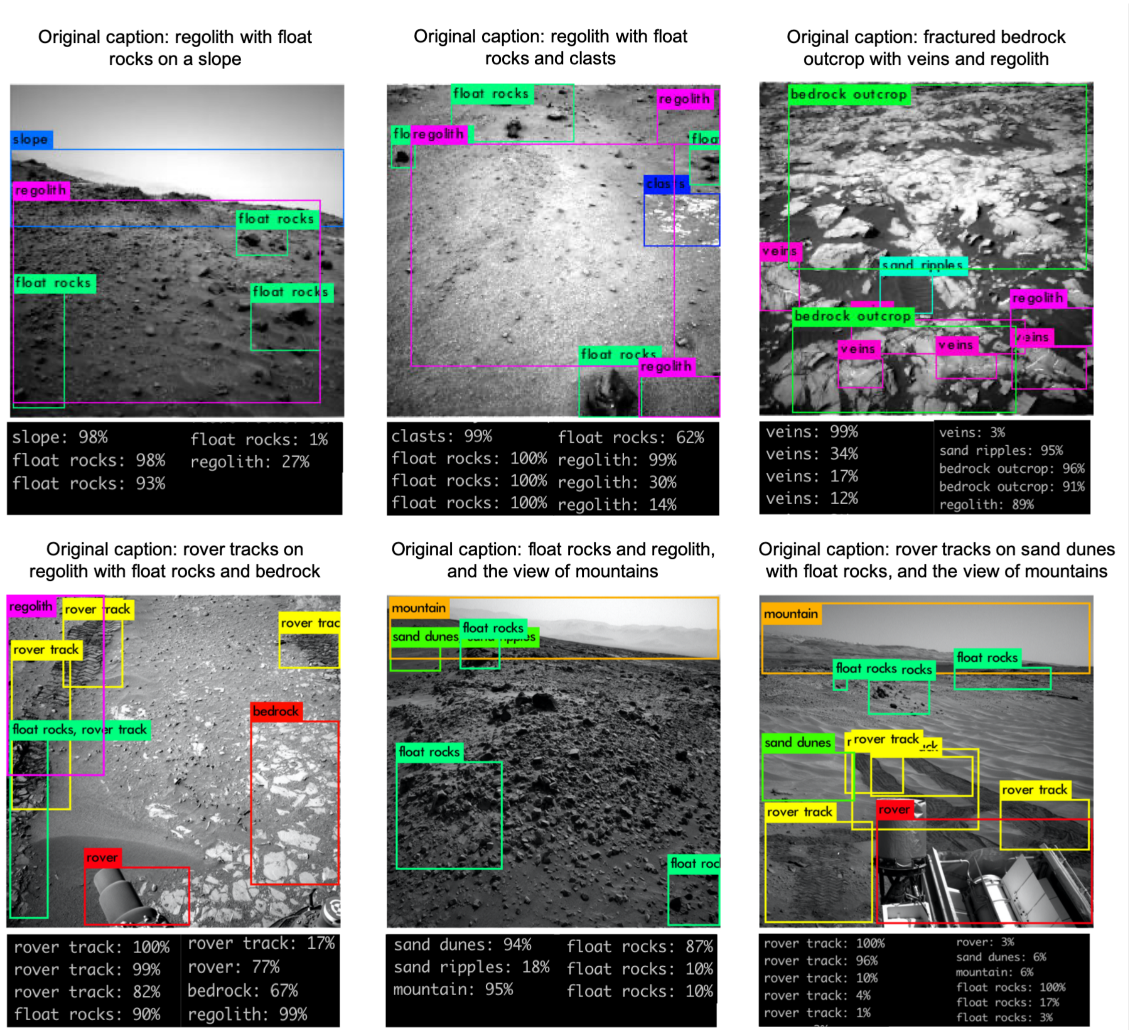


Figure 2-3: Inference results from the proposed object detection module. The object detection system can successfully detect various salient geologic features (such as float rocks, mountains, clasts, etc.) and non-geologic features (rover, rover tracks, etc.)

logistic regression. To extract image features, the Darknet-53 network is used as the backbone. This Darknet-53 architecture contains 53 convolutional layers (3×3 and 1×1 successive convolutional layers) along with residual shortcut connections in between layers as described in Table 2.1.

2.5.2 Text-image similarity

After the object detector $obj(x)$ is trained, the annotations obtained for each image are as followed: $f_{obj}(x_j) = \{\widehat{a_0}, \widehat{a_1}, ...\widehat{a_{M_j}}\}$ where M_j is the number of predicted

Table 2.2: Comparison of the text-image retrieval results in terms of Recall@K (R@K) on Mars dataset.

Method	Image retrieval		
	R@1	R@5	R@10
PVSE [163]	0.78%	5.47%	11.72%
Ours	15%	40%	53%

annotations. For the words in the captions, word2vec (w2v) model [122] is used which is primarily trained on a large corpus of text to produce word embeddings in vector space. Thus, both captions (words) and object label image features are brought into a common embedding space. Finally the similarity score is computed by using the following formula

$$s_j = \frac{1}{M_j} \sum_{k=1}^{M_j} d(avg\text{-}w2v(y_{query}), w2v(\widehat{a_k})). \tag{2.2}$$

where w2v takes a textual query y_{query} and predicted object annotations $\widehat{a_k}$ as input and calculate their vector representation and s_j computes the aggregated similarity score for all possible combinations of image region-word pairs.

2.6 Experimental Results

To evaluate the performance of the object detection module on the Mars dataset, several experiments are performed. The qualitative results obtained from the test dataset are shown in Figure 2-3. It can be seen that the object detection system can successfully detect salient features such as float rocks, mountain, rover tracks, clasts, etc. in the test images. The experimental results are reported using the metric Recall@k (R@k) at k = 1; 5; 10 in Table 2.2, where R@k measures the number of correct/relevant items amongst the top-k results. The proposed method is compared with the previous text-image retrieval method PVSE [163]. It can be seen that this method shows better performance compared to the previous attention-based PVSE

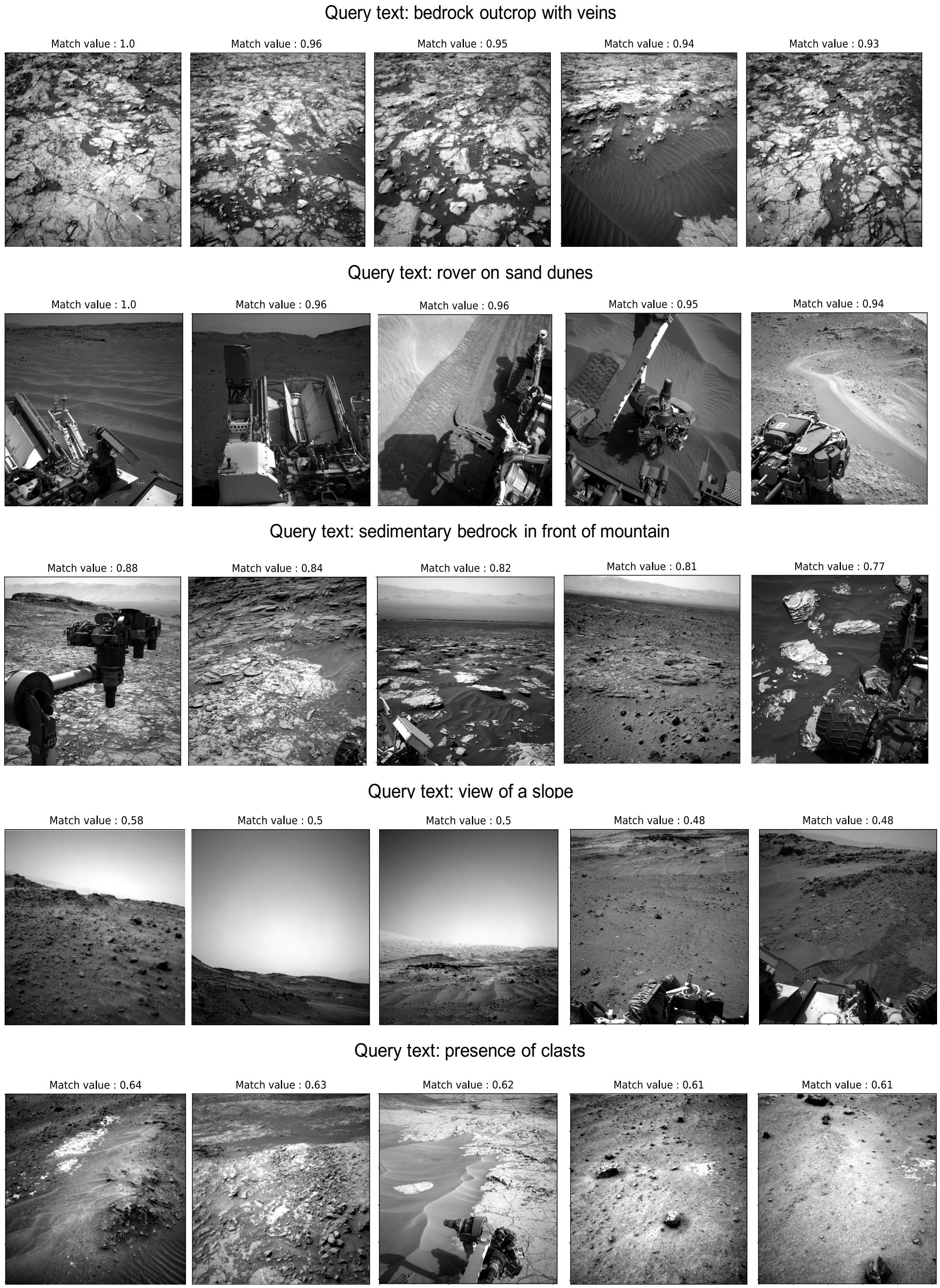


Figure 2-4: Text-to-image retrieval results on Mars dataset. For each query text the top five retrieved images, along with their similarity scores are shown.

method. I believe that the object labeling of the MarsDetect dataset has enabled the proposed method to retrieve images more efficiently compared to PVSE. The qualitative results of text-to-image retrieval are provided in Figure 2-4, where the top five retrieved images from the database along with the similarity scores for each query text have been shown. It can be seen that the proposed method can retrieve images with good confidence values.

2.7 Conclusions

This chapter presented a text-image matching algorithm based on a sub-task of object-detection. The major contribution made in this chapter is the labeled object detection dataset which is called "MarsDetect" that improves performance of text-image matching via an intermediate stage of object detection. A YOLO-v3 based object detection algorithm was employed to detect objects of interest in the planetary images. Therefore, in this way, a much smaller-sized encoded representation of image regions at the object level can be generated and sent back to Earth instead of the actual images which can effectively solve the limited bandwidth problem through deep space. Experimental results showed that the proposed algorithm can successfully detect objects in the planetary image. Moreover, the proposed text-image retrieval system could successfully retrieve images of interest with superior performance (as evidenced by higher R@K value) compared to the state-of-the-art approaches. This work demonstrated that it is possible to retrieve images of interest from a huge database of planetary image servers using a textual query which will automate the tedious task of searching each image one by one and save a lot of time for scientists to make faster decisions. This research was initially aimed to solve the limited bandwidth problem by developing onboard image analysis capabilities for better space exploration in the future. Nonetheless, the developed algorithm is also useful for other applications on the ground such as image retrieval.

Chapter 3

Toward Better Planetary Surface Exploration by Orbital Imagery Inpainting

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, this chapter proposes a machine learning-based “no-data” region prediction algorithm. Specifically, a deep convolutional neural network (CNN) based image inpainting algorithm is employed to predict such unphotographed pixels in a context-aware fashion using adversarial learning on planetary images. The benefit of using the proposed method is to augment features in the unphotographed regions leading to better downstream tasks such as interesting landmark classification. The Moon and Mars orbital images captured by the JAXA’s Kaguya mission and NASA’s Mars Reconnaissance Orbiter (MRO) are used for experimental purposes and the results demonstrate that the proposed 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, the proposed 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.

3.1 Introduction

In the quest of exploring and understanding planetary bodies, several missions to 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 [166], and Mars Reconnaissance Orbiter (MRO) [1] have provided scientists with Lunar and Mars orbital imagery. However, to obtain these high-resolution images, the swath width of the onboard 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 Figure 3-1.

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 a 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 land-

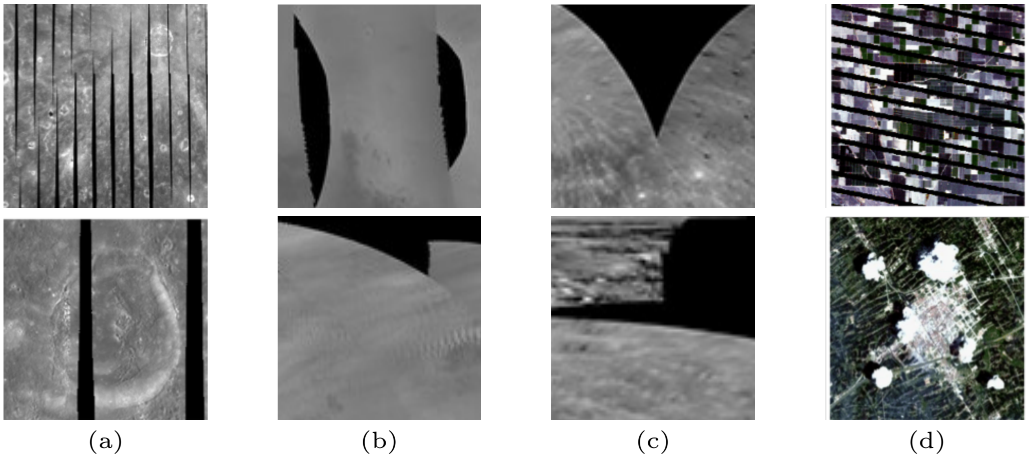


Figure 3-1: Example of unphotographed/missing pixel regions on (a) Lunar orbital imagery acquired by Kaguya mission's SELENE spacecraft [166], (b) Mars orbital imagery acquired by MRO [1], (c) Mercury orbital imagery acquired by MESSENGER spacecraft [26], and (d) Earth remote sensing images [194].

ing 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 here it is shown 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 [1, 26, 120, 166], 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 chapter, the aim is 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 [34, 36, 37, 68, 69, 81, 121, 145, 158, 182, 184, 189, 190, 192, 193, 194]. 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 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, 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, mode-specific expert neural networks are proposed that can handle such peculiarity of histogram distribution on any planetary surface images. Although in this chapter, the effectiveness of the proposed algorithm is shown only on Lunar and Mars surface images, this algorithm can be applied to any planetary images that suffer from such "no-data" regions.

Recently, deep learning [100] has garnered tremendous success because of its ability to express non-linear functions. Benefiting from this trend, CNNs have demonstrated outstanding performance in solving several high-level vision-based tasks such as image classification [67, 73, 161, 165], object recognition [147, 149], etc. as well as low-level tasks such as image denoising [191], super-resolution [53], etc. Therefore, in this work, a CNN is employed for restoring planetary orbital imagery contaminated with unphotographed pixels. Here, this problem is treated 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. This problem is tackled 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. The intuition is that mode-specific encoders will provide better inpainting results when compared with only one encoder trained on an average intensity distribution [153]. The proposed method builds upon the recently work on image inpainting called Context Encoder (CE) [140] which is a Generative Adversarial Networks (GAN) [59] based network where the 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. The main contributions in this chapter can be summarized as follows:

1. An adversarial learning-based image inpainting framework is introduced for planetary images (Moon, and Mars) that learns a non-linear end-to-end mapping from corrupted to clean images.

2. To enable better inpainting various modes of histogram distribution in the input images are extracted by unsupervised clustering. Here, mode-specific GAN models (which are expert models) are trained for inpainting images belonging to that cluster of the histogram mode. The novel contribution made in this work is the proposed idea of clustering the planetary images into several modes of histogram distributions, which helps to prevent the mode-collapse problem in GAN models and encourage the network to reliably generate samples from each cluster. This technique has not been applied by previous inpainting algorithms.

3. The simulated and real experimental results show that the proposed approach can restore images with a significant improvement in terms of visual quality and evaluation metrics, thereby outperforming previous inpainting methods.

4. Furthermore, it is shown that the proposed inpainting method helps in augmenting features of interesting but masked/incomplete landmarks which in turn leads to better generalization. The experimental results also validate this concept by boosting the classification accuracy of the morphological features on Mars images.

The rest of this chapter is organized as follows. Section 3.2 describes the related works on image inpainting techniques on standard datasets and remote-sensing datasets. Section 3.3 provides the details of the planetary datasets that are used for the experimental purpose. Section 3.4 explains the proposed method including the clustering of the training and testing images based on histogram distribution, the training and implementation details of the inpainting module, and classification module. Section 3.5

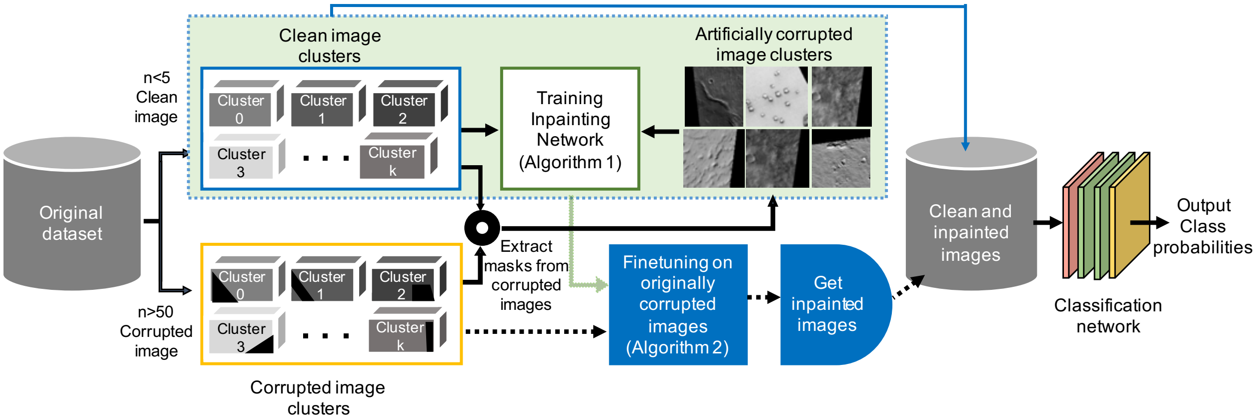


Figure 3-2: Overview of the proposed image inpainting algorithm. First, the masks from the real corrupted images are extracted and are superimposed on clean images and a pair of clean and *simulated/artificially corrupted* images is created. Next, the image inpainting module 4.3.2 is trained using these image pairs. Then, the trained inpainting model is fine-tuned on the real corrupted images. Finally, all the clean and inpainted version of the corrupted images are stored to solve a classification problem.

provides the experimental results of the missing data reconstruction in both simulated and real-data experiments and its contribution to boosting classification performance. Finally, the conclusions are presented in Section 3.6.

3.2 Related Works

In computer vision, the task of filling in the missing pixels of an image is known as image inpainting. This section briefly reviews the previous image inpainting works on standard real-life datasets [38, 50, 52, 111] by broadly categorizing them into three sub-fields, (i) traditional inpainting techniques, (ii) CNN-based inpainting, and (iii) GAN-based inpainting. This section also reviews previous works on remote sensing imagery inpainting on Earth and Moon.

3.2.1 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* [24, 27, 28, 105] where a diffusive process is modeled using Partial Differential

Equations (PDE) to propagate colors into the missing regions. Chan et al. [32] proposed a novel adaptive total variation (ATV) 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 [57] 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. [55]. Another optimization method to synthesize visual data (images or video) based on bi-directional similarity measure was proposed by Simakov et al. [159]. Later these techniques were expedited by Barnes et al. [25] 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. [41] proposed a patch-based greedy sampling algorithm, which enables faster image inpainting. Meur et al. [99] introduced a hierarchical super-resolution algorithm for image inpainting. He et al. [64] 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.

3.2.2 CNN-based Inpainting

With the recent success of CNN models [102] 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. [177] 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. [95] showed a mask specific deep neural network-based blind inpainting technique for filling in small missing regions in an image. Chaudhury et al. [33] 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) [117] technique. However, initially, CNN-based image inpainting approaches were limited to very small sized masks.

3.2.3 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. [140] 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 [59]. Later, Iizuka et al. [76] extended the work of [140] 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 [140], this approach also used an adversarial training approach for image completion, but unlike [140], this method could handle arbitrary image size and mask because of the proposed global and local context discriminator networks. Recently, Yu et al. [185] presented a unified feedforward generative network with a novel contextual attention layer, trained with reconstruction losses and two Wasserstein GAN [20, 61] 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. [109] 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. [164] 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. [178] 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. [130] 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. [186] proposed a gated convolution-based approach to handle free-form image completion.

3.2.4 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 [158, 194]. Remote Sensing (RS) image inpainting using spatial information include interpolation-based methods [190, 192, 184], variation-based methods [36, 69], PDE-based methods [121], and exemplar-based methods [41]. 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) [145, 182], and temporal-based methods (using data taken at the same location in different periods) [189, 34] have been proposed. Later more generalized algorithms (hybrid methods) were developed by integrating spatial, spectral, and temporal information [194]. To this end, Ji et al. [81] proposed a non-local low-rank tensor completion algorithm to reconstruct the missing information. Cheng et al. [37] introduced a double-weighted low-rank tensor (DWLRT) model and He et al. [68] proposed a TV-

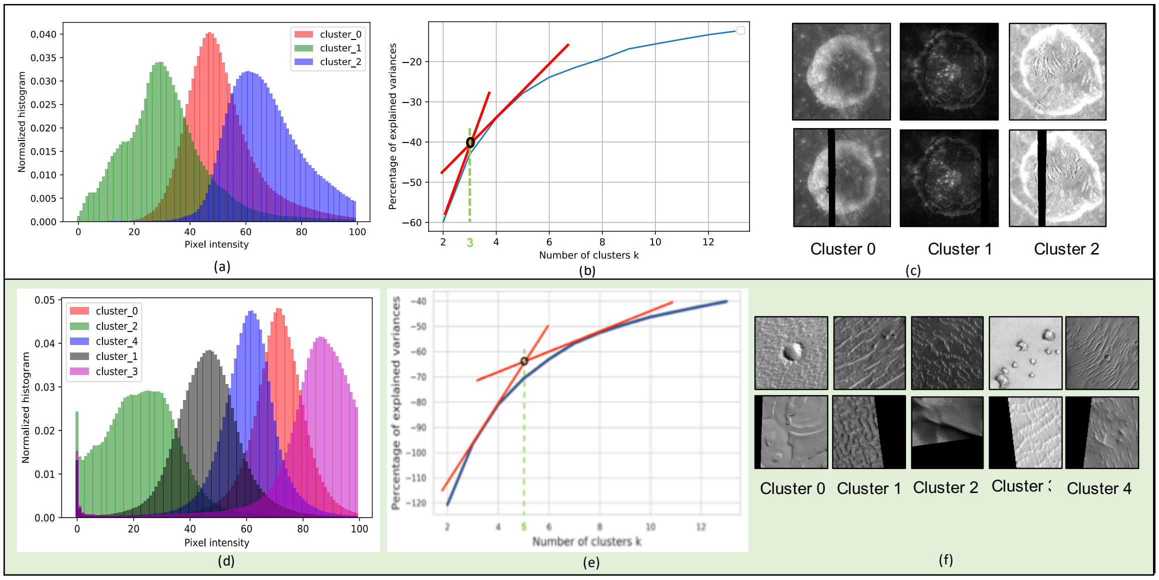


Figure 3-3: Details of Moon and Mars dataset: (a, d) Histogram distribution of several modes of input images, (b, e) Knee point analysis for determining the optimal number of clusters, (c, f) Examples of clean and corrupted images for each cluster. The first and second row demonstrates the details of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

regularized tensor ring completion (TVTR) model to reconstruct missing data in RS images. Recently, Zhang et al. [193] 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. [156] 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.

3.3 Experimental Data

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

3.3.1 Lunar Orbital Imagery by SELENE

Here the averaged lower resolution mosaic data of the lunar surface is used which was captured by Multiband Imager onboard the JAXA lunar explorer satellite SE-

Table 3.1: 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

LENE (Kaguya) [166]. This image covered the lunar surface with longitude (+180 to -180 degree) and latitude (+85 to -85 degree). This entire Lunar image consists of 46080×21760 pixels with an image size of 600MB. The Moon dataset is created 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 Figure 3-3(c). Here, each image is of size 256×256 pixels. Several pairs of clean and artificially corrupted crater image are generated 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 3.1. The second row of Table 3.1 describes the total number of real corrupted images in the dataset. Since the real corrupted Moon images are not used for inference purposes, they are not divided into clusters and are kept blank in the corresponding columns of Table 3.1.

3.3.2 Mars Orbital Imagery by MRO

For Mars images, the grayscale-version of the Mars orbital images are used, which were collected by the HiRISE camera onboard the MRO having a spatial resolution of approximately 30 cm/pixel [120]. This dataset [172] 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 Figure 3-4. 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, they are resized 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 3.1. The second row of Table 3.1 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 Figure 3-3(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 Figure 3-3(f) are map projections. Nevertheless, for demonstration purposes, they are considered as an example of the "no-data" region and the proposed algorithm show how to predict such a "no-data" region for better surface image analysis.

3.4 Proposed Method

In this chapter, the 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. This problem is solved by the following four-stage approach:

1. The Moon and Mars dataset have images from several modes in the histogram distributions. Therefore the images are divided into different clusters to prevent the mode-collapse problem in generative models and encourage the network to reliably generate samples from each clusters.

[1] The dataset is available at `https://zenodo.org/record/2538136#.XYjEuZMzagR`.

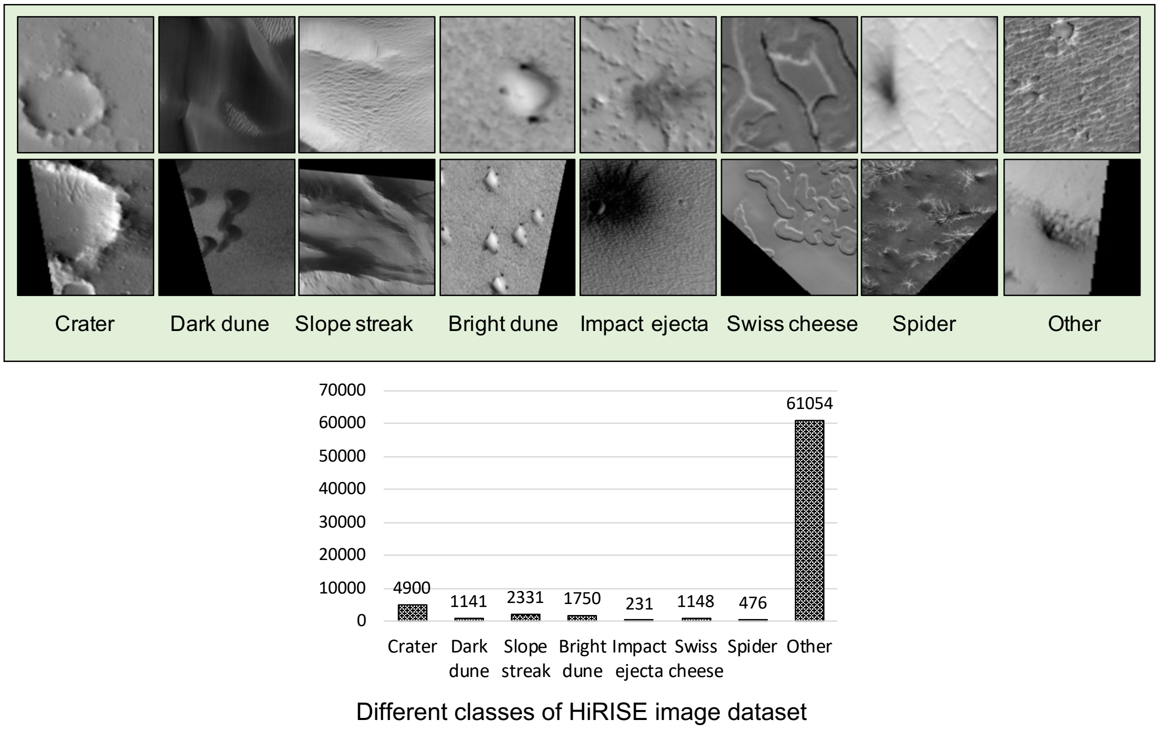


Figure 3-4: Example of different classes of Mars dataset with the corresponding number of images for each class.

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, the masks/unphotographed pixel regions are extracted and are artificially superimpose on the clean image samples, thus yielding artificially corrupted and corresponding clean image pairs for each clusters. GAN-based inpainting is performed on such paired data.

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

4. Finally, a classification task is performed on original images and is compared 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 the first stage (separating images into clusters) and second

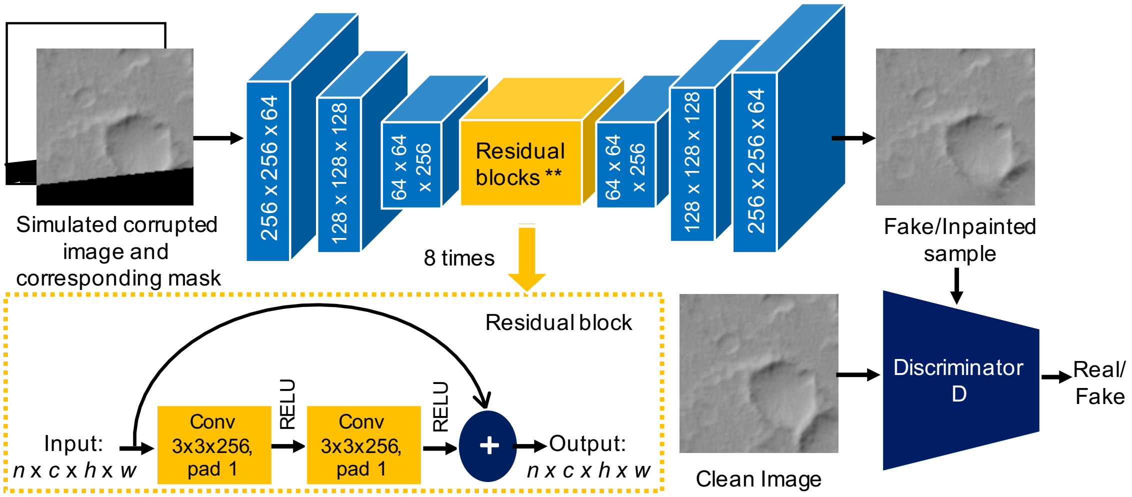


Figure 3-5: Network architecture.

stage (image inpainting task) are performed for both Moon and Mars image datasets. However, the third stage (inference on real-corrupted images), and fourth stage (classification) are performed only on the Mars dataset. This is because the Moon dataset consists of images only from one class (crater) [2], whereas the Mars dataset consists of eight different classes as shown in Figure 3-4. Therefore it is more useful to demonstrate the results of the third and fourth stages using the Mars dataset. In the following subsections, at first the preprocessing step for the data and details of extracting masks tailored for this task are introduced. Then the detailed implementation of the image inpainting module and the fine-tuning process on real-corrupted images are explained. Finally, the image classification module is described. The overall framework of the proposed method is shown in Figure 3-2.

3.4.1 Unsupervised Separation of Histogram Clusters

During experiments, it was found that the Moon and Mars dataset have planetary images from several modes in the histogram distributions and they can be separated into clusters as shown in Figure 3-3(a, d). To encourage the generative model to faithfully reproduce samples from each such clusters, the images with different his-

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

togram distribution are separated 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 $p(x)$ [153]. Here, first the number of black pixels in every image is calculated. Images that have number of black pixels less than 5 are considered as *clean image* and images that have number of black pixels more than 50 are considered as *corrupted images.* Then a k-means clustering [62] is performed to cluster these images with missing pixels into different groups based on their histogram distribution as shown in the following equation

$$h_k(x) = \frac{n(x[i,j]=k)}{\text{number of pixels}}, \tag{3.1}$$

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, a Knee point analysis [195] is carried out. As shown in Figure 3-3(b, 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 Figure 3-3(c, f). Subsequently, from the cluster centers in the grayscale histogram space, a class label is assigned to each clean image, according to the cluster center having the closest Euclidean distance. Next, for each of the clusters, a mask (missing pixels from the corrupted images) is extracted and is randomly superimposed on clean images to create pairs of clean and artificially corrupted images for training and testing different regression models.

3.4.2 Image Inpainting Module

Given a real corrupted image, the 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 the aim is 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, adversarial learning based GAN models are trained, which has shown promising results in generative modeling of images [144] in recent years. The network architecture of the proposed inpainting module is shown in Figure 3-5. The 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.

3.4.2.1 Network Architecture

Generator: The generator architecture is adapted from Johnson et al. [83] which has shown impressive results for neural style transfer and image-to-image translation [196]. 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 [67], 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 instance normalization [170] and ReLU activation function are used 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 3.2.

Table 3.2: 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 3.3: 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	$1 \times 1 \times 30 \times 30$

Discriminator: The discriminator network is a Markovian discriminator similar to 70×70 PatchGAN, adapted from [79, 196]. 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 [79, 196]. The details of the discriminator network and output size of each layer is given in Table 3.3. It should be noted that the sigmoid function applied after the last convolution layer produces a 1-dimensional output score that predicts whether the 70×70 overlapping image patches are real or fake. For the discriminator network, spectral normalization [124] is used as the weight normalization method because it can stabilize the discriminator network training. Moreover, here all the ReLUs are leaky ReLUs [116] with slope of 0.2.

3.4.2.2 Training

The proposed inpainting network is trained 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 3.1. While training, for each real corrupted image x_c, a binary *image mask* $\mathbf{m}$ (which takes the value 0 on the regions to be filled-in and 1 elsewhere) is extracted. Now for each clean image $\mathbf{x}$, the extracted masks $\mathbf{m}$ are randomly superimposed to obtain artificially corrupted *input image* $\mathbf{z} = \mathbf{x} \odot \mathbf{m}$, where $\odot$ denotes element-wise product operation. The generator of the inpainting network G takes this concatenated *input image* $\mathbf{z}$ and *image mask* $\mathbf{m}$ as input, and produces an *predicted image* $\mathbf{x}' = G(\mathbf{z}, \mathbf{m})$ as output. Then by adding the masked region of $\mathbf{x}'$ to *input image*, *completed image* is obtained 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 3.

3.4.2.3 Loss Functions

To train the inpainting module to restore the input corrupted image realistically, two loss functions are used: a reconstruction loss and an adversarial loss [59]. 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 [79]. 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 [140] 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, ℓ_1 loss is used that minimizes the distance between the clean/ground-truth image $\mathbf{x}$ and the completed/inpainted image $\tilde{\mathbf{x}}$.

$$\mathcal{L}_{\ell_1}(x) = [\|\mathbf{x} - \tilde{\mathbf{x}}\|_1]. \tag{3.2}$$

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, the min-max optimization strategy is followed, 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

Algorithm 1 Training of the proposed inpainting framework.

```
while Generator G has not converged do
    Sample batch images $\mathbf{x}$ from clean training data;
    Extract masks $\mathbf{m}$ from corrupted training data;
    Artificially construct corrupted inputs $\mathbf{z} \leftarrow \mathbf{x} \odot \mathbf{m}$;
    Generate inpainted images by modifying masked region, $\tilde{\mathbf{x}} \leftarrow \mathbf{z}+G(\mathbf{z}, \mathbf{m}) \odot (\mathbf{1} - \mathbf{m})$;
    Update G with $\ell_1$ loss and adversarial critic loss;
    Update discriminator critic D with $\mathbf{x}$, $\tilde{\mathbf{x}}$;
end while
```

objective function can be expressed as follows

$$G^*, D^* = \arg\min_G \max_D \mathcal{L}_{adv}(G, D) = \mathbb{E}_{x,\tilde{\mathbf{x}}}[\log D(x, \tilde{\mathbf{x}})] + \mathbb{E}_{\tilde{\mathbf{x}}}[\log(1 - D(\tilde{\mathbf{x}}, \mathbf{x}')], \tag{3.3}$$

Thus, the overall loss function becomes

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{\ell_1} + \lambda_2 \mathcal{L}_{adv}, \tag{3.4}$$

where $\lambda_1 = 1$ 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.

3.4.2.4 Implementation Details

The proposed model is implemented in PyTorch.[3] The network is trained by optimizing the encoder-decoder and discriminator using the Adam optimizer [93] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. In the experiments, a batch size of 14 and training iterations of 100 are used. While training, the image is resized to 256×256 and linearly scale

[3] The code is available at `https://github.com/hiyaroy12/mars-image-inpainting`.

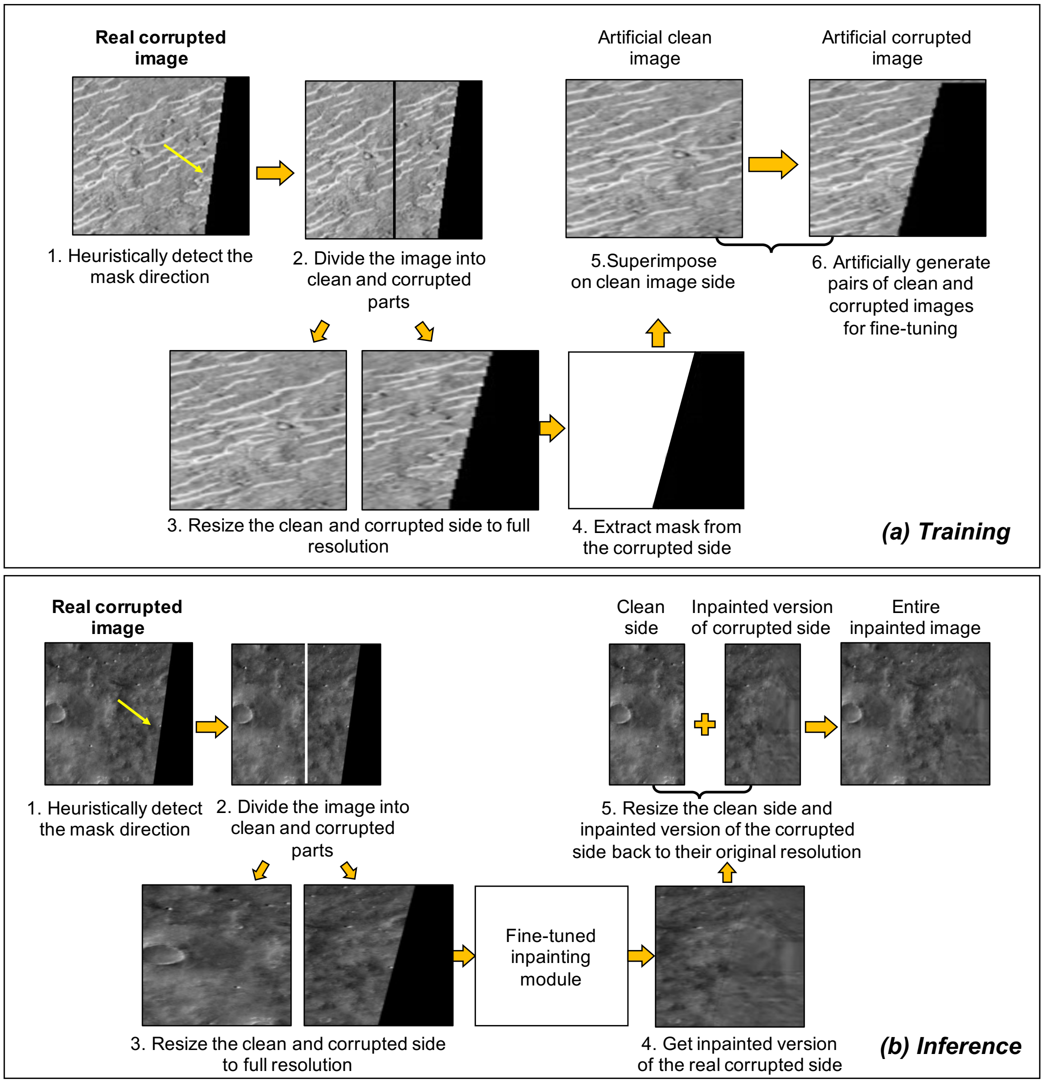

Real corrupted image
Artificial clean image
Artificial corrupted image
1. Heuristically detect the mask direction
2. Divide the image into clean and corrupted parts
5.Superimpose on clean image side
6. Artificially generate pairs of clean and corrupted images for fine-tuning
3. Resize the clean and corrupted side to full resolution
4. Extract mask from the corrupted side
(a) Training
Real corrupted image
Clean side
Inpainted version of corrupted side
Entire inpainted image
1. Heuristically detect the mask direction
2. Divide the image into clean and corrupted parts
5. Resize the clean side and inpainted version of the corrupted side back to their original resolution
Fine-tuned inpainting module
3. Resize the clean and corrupted side to full resolution
4. Get inpainted version of the real corrupted side
(b) Inference

Figure 3-6: Processing of real corrupted images to fine-tune the inpainting network (a) during training and (b) during inference. The fine-tuning is performed only on the real corrupted Mars images for demonstration purpose.

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 are trained together on a TITAN Xp (12 GB) GPU.

3.4.2.5 Fine-tuning The Network

It was found that the clean images and real corrupted images were visually different based on their grayscale value, contrast, and histogram distribution. This results in

Algorithm 2 Fine-tuning the inpainting framework for real corrupted images.

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

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 the intended test images. For this purpose, the pre-trained inpainting model (explained in Section 3.4.2) is fine-tuned on limited regions of the real corrupted images, to get the inpainted images, as described in Algorithm 2.

To fine-tune the inpainting model, during training, it was heuristically identified that each corrupted image has four mask directions: *North, South, East, and West.* Here, 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, it is found 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, the opposite image region is cropped and synthetically corrupted that part after resizing it to full resolution 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 Figure 3-6(a). After tailoring the real corrupted data for this task, the pre-trained image inpainting model is fine-tuned keeping the same optimization conditions as mentioned in Section 3.4.2.4.

During inference, a real corrupted image from the test set is taken and the mask direction d is heuristically identified similar to training. Next, the clean side (half) of the image is cropped and is resized to full resolution of 256 × 256 and then it is considered as a clean image. Similarly, the corrupted side (half) of the image (which needs to be inpainted) is cropped and is resized to full resolution of 256 × 256 and then it is considered as the corrupted image. While inference, this step is different from training. This is because in training the clean side is needed to be artificially corrupted by extracting a mask from the corrupted side to get a clean-artificially corrupted image pair. Whereas, during inference, the originally corrupted side of the image needs to be inpainted. After the inpainted half is obtained, both the inpainted side and the clean side are resized back to their previous resolution which is 256 × 128. After that, both the sides are added 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 Figure 3-6(b). Here, after cropping the image into half, the standard practice of resizing the image into the full resolution (256 × 256) is followed, before feeding it into the network and then it is resized back to its original size. Therefore, I 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, these images are considered as 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.

3.4.3 Image Classification Module

After performing the image inpainting to augment the incomplete features on the real corrupted images, these inpainted images along with the clean images are used for the classification task. These additional experiments are performed to check if image inpainting on real corrupted images helps in better feature learning, which in turn leads to improved classification performance. Since the Mars dataset is highly imbalanced, a natural approach is taken which resamples the given dataset by "oversampling" the minority classes [80] and "undersampling" the majority classes [63]. Such resampling of data helps in achieving a balanced distribution during training.

Table 3.4: Quantitative evaluation results of *simulated/artificially corrupted* Moon dataset for different clusters and all images together (when not divided into clusters) using Generative Inpainting (GI) [186], and the proposed method. The best results for each row is shown in bold. $^-$Lower is better. $^+$Higher is better.

		Moon dataset				
	Method	Cluster 0	Cluster 1	Cluster 2	Mean of Clusters	All
PSNR$^+$	GI [186]	39.59	41.29	41.13	40.67	40.13
	Ours	42.32	43.69	40.98	**42.33**	40.23
SSIM$^+$	GI [186]	0.968	0.981	0.977	0.975	0.971
	Ours	0.989	0.989	0.984	**0.987**	0.982
ℓ_1 (%)$^-$	GI [186]	0.4	1.0	0.3	0.5	0.6
	Ours	0.1	0.3	0.1	**0.2**	0.3

For the experiments, two variants of ResNet [67] model ResNet-50 and ResNet-101 are trained for 50 epochs with mini-batch size 80, and a weight decay of 1×10^{-4} . Both the models are trained using ADAM [93] optimizer with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. The initial learning rate is set to 0.002, resize the input images to 224×224, and ensure that all images over the training dataset are normalized. The ResNet classifier [67] is used 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.

3.5 Experimental Results

This section discusses the quantitative and qualitative results obtained from the inpainting module. Several experiments are performed 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 the 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 im-

Table 3.5: Quantitative evaluation results of *simulated/artificially corrupted* Mars dataset for different clusters and all images together (when not divided into clusters) using Generative Inpainting (GI) [186], and the proposed method. The best results for each row is shown in bold. $^{-}$Lower is better. $^{+}$Higher is better.

	Mars dataset							
	Method	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Mean of Clusters	All
PSNR^{+}	GI [186]	30.46	31.08	31.71	32.84	31.93	31.60	31.04
	Ours	33.02	33.46	34.01	33.72	33.32	**33.51**	33.42
SSIM^{+}	GI [186]	0.904	0.909	0.907	0.922	0.913	0.911	0.906
	Ours	0.928	0.931	0.933	0.926	0.928	**0.930**	0.928
ℓ_1 (%)$^{-}$	GI [186]	1.0	1.6	2.7	0.6	1.0	1.4	1.4
	Ours	0.7	1.1	1.7	0.5	0.8	**0.9**	0.9

proving inpainting quality? and 4) Can the proposed inpainting method contribute to better feature learning for interesting landmark classification thereby improving classification performance?

Quantitative Results: The quantitative performance of the proposed method is reported in terms of the following metrics 1) peak-signal-to-noise ratio (PSNR); 2) structural similarity index (SSIM) [173] 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

$$PSNR = 20 \log_{10} \left(\frac{MAX_I}{\sqrt{MSE}} \right), \tag{3.5}$$

where MAX_I the maximum value of the pixel in the original image. A higher PSNR normally indicates higher quality reconstruction. SSIM index [173] provides a quantitative assessment of the perceptual quality of the reconstructed image. These metrics are calculated on the test set of the *artificially corrupted* images and compare them to their corresponding clean ground truth images. The quantitative evaluation results for both Moon and Mars images using Generative Inpainting [186] and the proposed method are reported in Table 3.4 and 3.5 where the metric values for images from

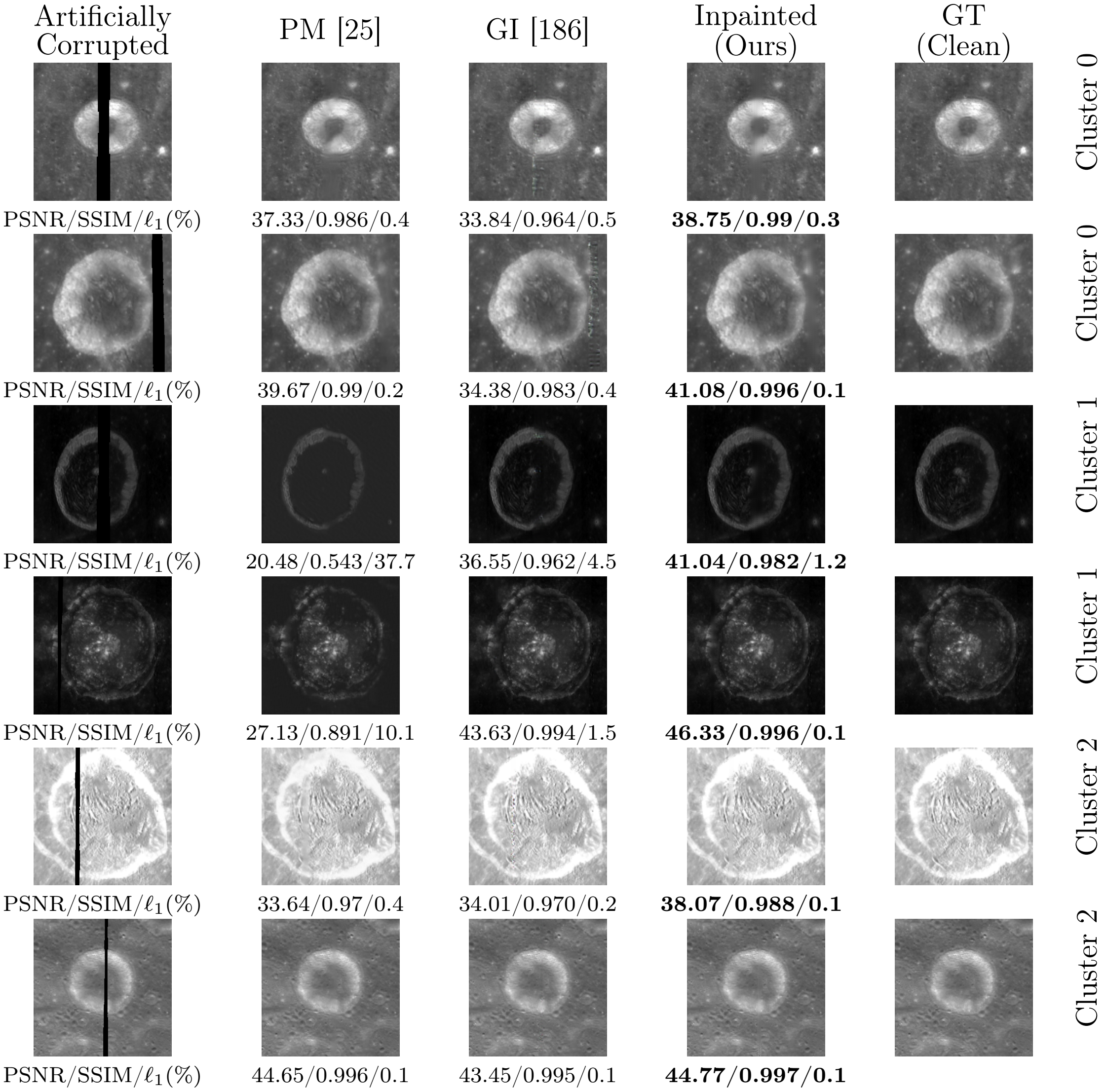


Figure 3-7: Visual examples of semantic feature completion of *simulated/artificially corrupted images* on Moon dataset using different methods: PatchMatch [25], Generative Inpainting (GI) [186], and the proposed method. Since these are artificially corrupt the clean images, therefore the clean images are considered as ground truth data in this case.

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. The proposed method outperforms the previous method [186] 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'

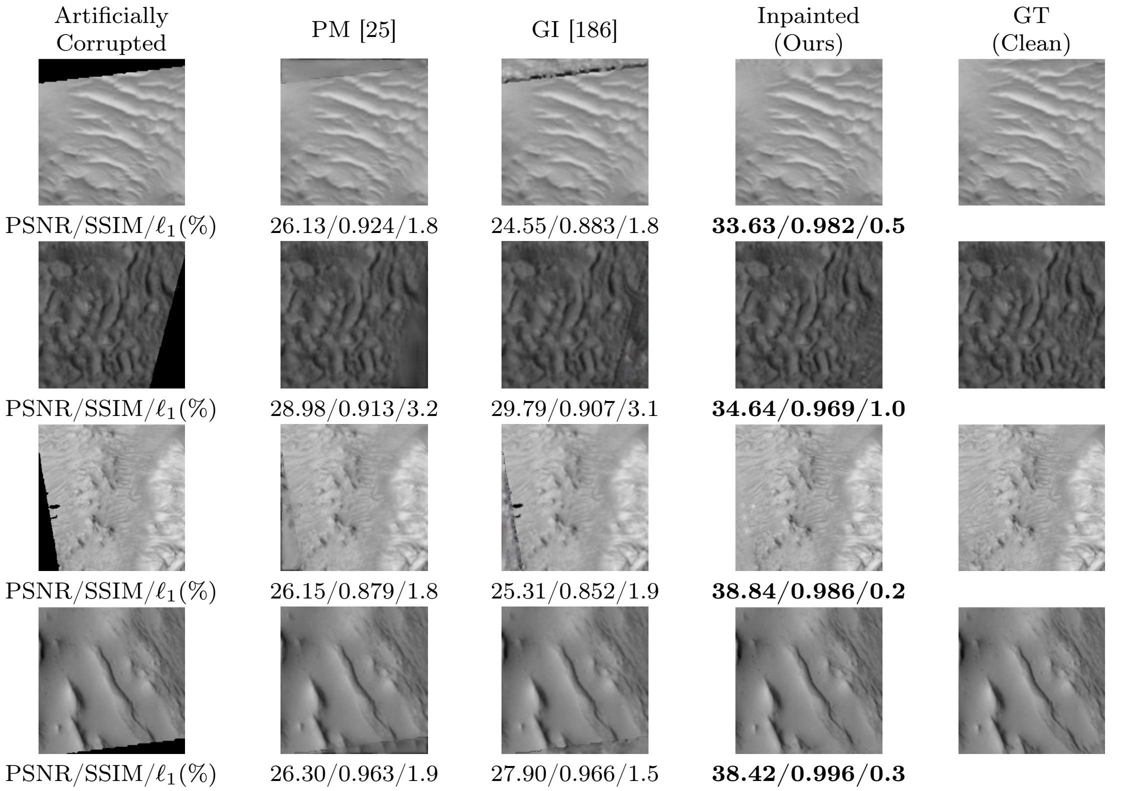


Figure 3-8: Visual examples of semantic feature completion of *simulated/artificially corrupted images* on Mars dataset using different methods: PatchMatch [25], Generative Inpainting (GI) [186], and the proposed method. Since these are artificially corrupt the clean images, therefore the clean images are considered as ground truth data in this case.

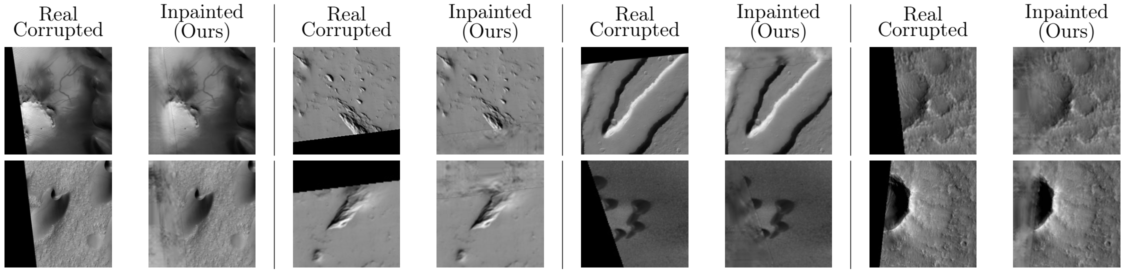


Figure 3-9: Visual examples of semantic feature completion of the *real corrupted images* using the proposed method. Since these are real corrupted data, they do not have corresponding ground truth images available.

and 'all images' in case of Moon images as shown in Table 3.4) demonstrates the validity of the proposed idea of leveraging the benefits of clustering the training and testing images based on their histogram distribution.

Qualitative Results: Figure 3-7 and 3-8 show the qualitative performance of the

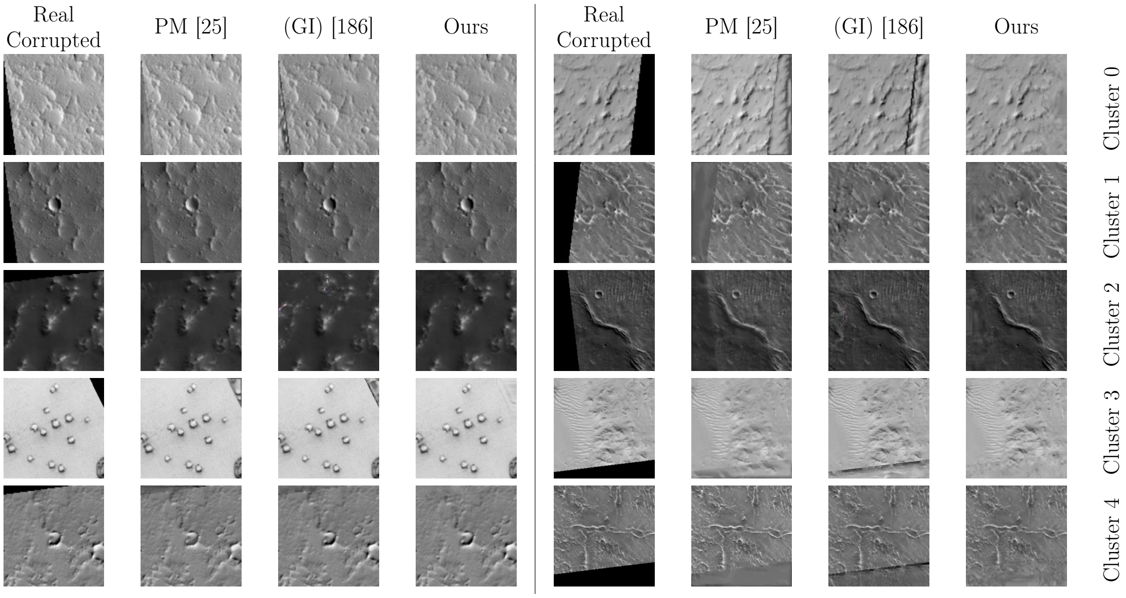


Figure 3-10: Visual examples of semantic feature completion of *real corrupted images* on Mars dataset using different methods: PatchMatch [25], Generative Inpainting (GI) [186], and the proposed method. Since these are real corrupted data, they do not have corresponding ground truth images available.

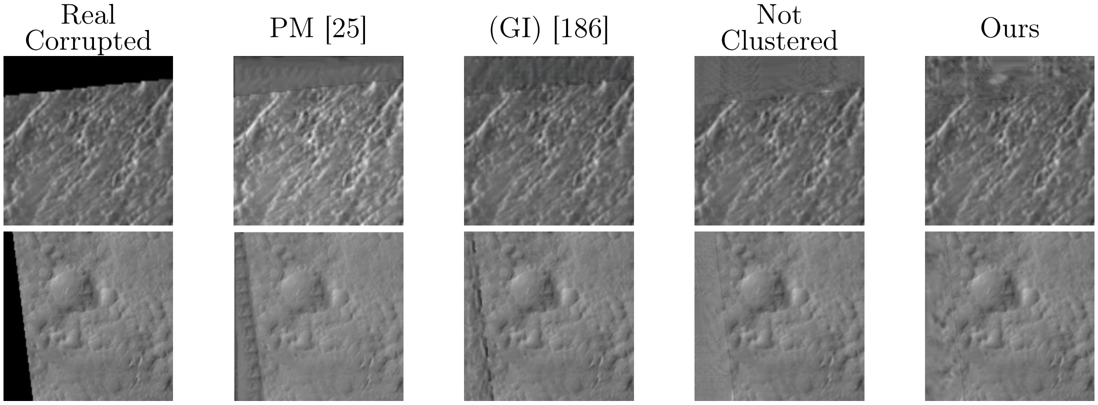


Figure 3-11: Visual examples of semantic feature completion of *real corrupted images* on Mars dataset using different methods: PatchMatch [25], Generative Inpainting (GI) [186], inpainting results when all images are trained together (when not divided into clusters), and inpainting results (when divided into clusters). Since these are real corrupted data, they do not have corresponding ground truth images available.

proposed inpainting model when tested on *artificially corrupted images*. Here it can be seen that for both Moon and Mars dataset, previous inpainting methods, PatchMatch (PM) [25], and Generative Inpainting (GI) [186] generate significant artifact, however the proposed method can predict the missing region that looks similar to the ground truth data. On the contrary, Figure 3-9 demonstrates the qualitative inpainting results obtainted from the proposed inpainting model, when tested on *real corrupted test set* of Mars images. For example, as seen in Figure 3-9 the crater or dark dune that are originally masked (first column) can be successfully augmented in shape (second column) by using the proposed inpainting algorithm. This proves the generalization ability of this inpainting model to complete different landmarks on the planetary images. Figure 3-10 compares the qualitative inpainting results achieved by the proposed model on *real corrupted test set* of Mars images when compared with the previous inpainting methods PM [25], and GI [186]. It is clear that the proposed method produces more photo-realistic results with seamless boundary transitions. Figure 3-11 shows 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) vs. images divided into clusters. It can be seen the previous methods generate artifacts in the boundary causing the inpainted images to look unrealistic. On the contrary, the proposed 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 Figure 3-9, 3-10, and 3-11 inpainting results on *real corrupted Mars images* are shown, 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.

3.5.1 Ablation Study For Inpainting Results

Here, the effect of different loss components used to train the proposed model is analyzed. It is also investigated if fine-tuning the inpainting model contributes to the better reconstruction of the real corrupted image. Table 3.6 and 3.7 report the quantitative results achieved by the proposed 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 that the adversarial loss component has a great contribution in improving the inpainting quality in terms of the metric values.

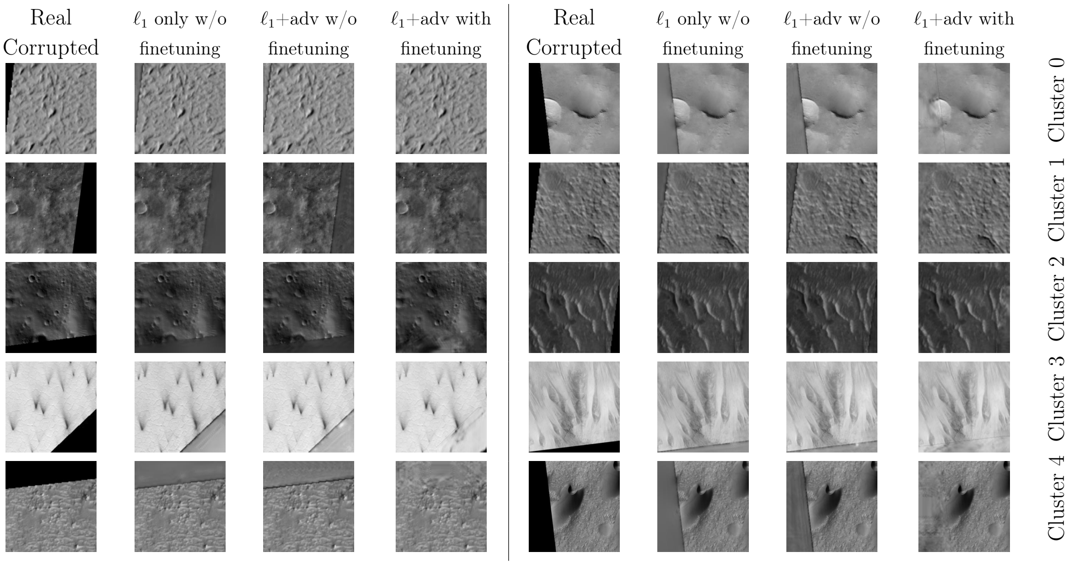

Real Corrupted
ℓ_1 only w/o finetuning
ℓ_1+adv w/o finetuning
ℓ_1+adv with finetuning
Real Corrupted
ℓ_1 only w/o finetuning
ℓ_1+adv w/o finetuning
ℓ_1+adv with finetuning
Cluster 0
Cluster 1
Cluster 2
Cluster 3
Cluster 4

Figure 3-12: 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 fine-tuning, (ℓ_1+adversarial loss) without and with fine-tuning respectively.

Figure 3-12 shows the qualitative inpainting results for *real corrupted* Mars images using only ℓ_1 loss (without any fine-tuning), and ℓ_1 with adversarial loss (with and without fine-tuning). Clearly, if the proposed model is trained using only ℓ_1 loss, the reconstruction contains a significant amount of artifacts, as seen in the second and sixth column of Figure 3-12. Whereas, if the adversarial loss component is added, it improves the inpainting performance to a certain extent, as seen in the third and seventh column. Therefore, it can be concluded that for planetary image inpainting, adversarial loss is an essential ingredient. Next, the performance of the proposed model is compared with and without fine-tuning. From the fourth and eight column of Figure 3-12, it can be seen that fine-tuning greatly improves the performance of the proposed image inpainting model. Moreover, fine-tuning 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 3.6: Quantitative comparison of different components of the proposed method on Moon dataset.

	Moon dataset					
	Method	Cluster 0	Cluster 1	Cluster 2	Mean of Clusters	All
PSNR$^+$	ℓ_1 loss	37.07	35.54	33.96	35.52	37.34
	ℓ_1+adv loss	42.32	43.69	40.98	**42.33**	40.23
SSIM$^+$	ℓ_1 loss	0.979	0.96	0.97	0.969	0.979
	ℓ_1+adv loss	0.99	0.99	0.98	**0.99**	0.98
ℓ_1 (%)$^-$	ℓ_1 loss	1.0	2.7	12.9	5.5	1.1
	ℓ_1+adv loss	0.1	0.3	0.1	**0.2**	0.3

3.5.2 Classification Results

Here, the image classification results on the Mars dataset, after applying the proposed image inpainting technique is provided. Figure 3-13 provides 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. It should be noted that the 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 Figure 3-4. Therefore, the classification performance of the proposed model is shown in terms of precision, recall, and F1-score (harmonic mean of precision and recall) metrics, which are more appropriate to handle class imbalance. As can be seen in Figure 3-13, 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 the proposed model learns better features for classification tasks when trained on the inpainted images in comparison with original images (having partially masked interesting landmarks).

The area under the curve (AUC) for receiver operating characteristics (ROC) for original and inpainted images for ResNet-50 and ResNet-101 models are shown in

Table 3.7: Quantitative comparison of different components of the proposed method on Mars dataset.

	Mars dataset							
	Method	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4	Mean of Clusters	All
PSNR$^+$	ℓ_1 loss	32.62	33.16	31.72	32.87	32.87	32.64	32.48
	ℓ_1+adv loss	33.02	33.46	34.01	33.72	33.32	**33.51**	33.42
SSIM$^+$	ℓ_1 loss	0.923	0.926	0.909	0.924	0.925	0.921	0.930
	ℓ_1+adv loss	0.928	0.931	0.933	0.926	0.928	**0.930**	0.928
ℓ_1 (%)$^-$	ℓ_1 loss	1.0	1.5	3.7	1.0	1.1	1.7	1.1
	ℓ_1+adv loss	0.7	1.1	1.7	0.5	0.8	**0.9**	0.9

Figure 3-14. Since the AUROC metric is also appropriate to handle class imbalance, the AUC of all the classes and for each case is reported. It can be seen that for minority classes such as "Impact Ejecta", ResNet-50 performs better for inpainted images than original images as indicated by the higher area under ROC (AUROC) curve value.

Additionally, the classification accuracy of the proposed method is provided in Table 3.8 and compare the results with previous classification results [172] 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 dataset. Because in such case, high accuracy can be achieved by a non-expert 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 (Figure 3-13, and 3-14) for both ResNet-50 and ResNet-101 models, the effectiveness of the proposed inpainting algorithm for better feature learning for classification is proved.

Table 3.8: Classification results using the proposed inpainted images.

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

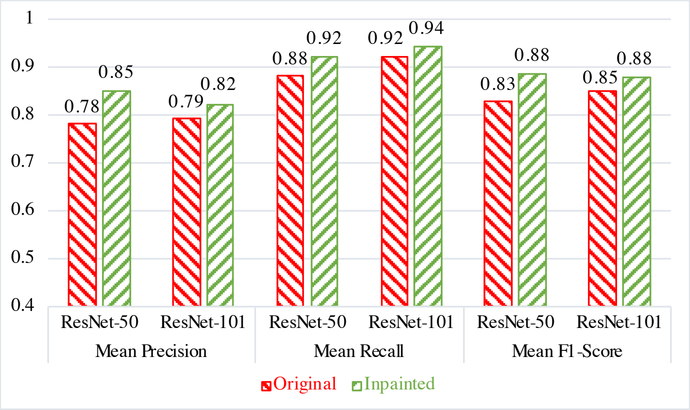


Figure 3-13: Precision/Recall/F1 score for ResNet-50 and ResNet-101 in HiRISE inpainted dataset (ADAM).

3.6 Conclusions

This chapter presents an adversarial training based image inpainting technique for planetary images to facilitate improved scientific discoveries. The new contribution in this work is the idea of performing unsupervised clustering to divide the images into different modes of histogram distribution and then predict the unphotographed pixels by training a GAN-based model on input images belonging to different clusters. This proposed idea of clustering helps to prevent the mode-collapse problem in GAN models and encourage the network to reliably generate samples from each cluster. It is found that the proposed inpainting algorithm helps the network to learn better features by augmenting the incomplete landmarks leading to better generalization. This analysis reveals that by performing such image inpainting as a first step, the classification performance can be boosted with an improved F1 score. I believe that this work will benefit the planetary science community to analyze and explore the planetary images in a more efficient way. The proposed method can also be a helpful

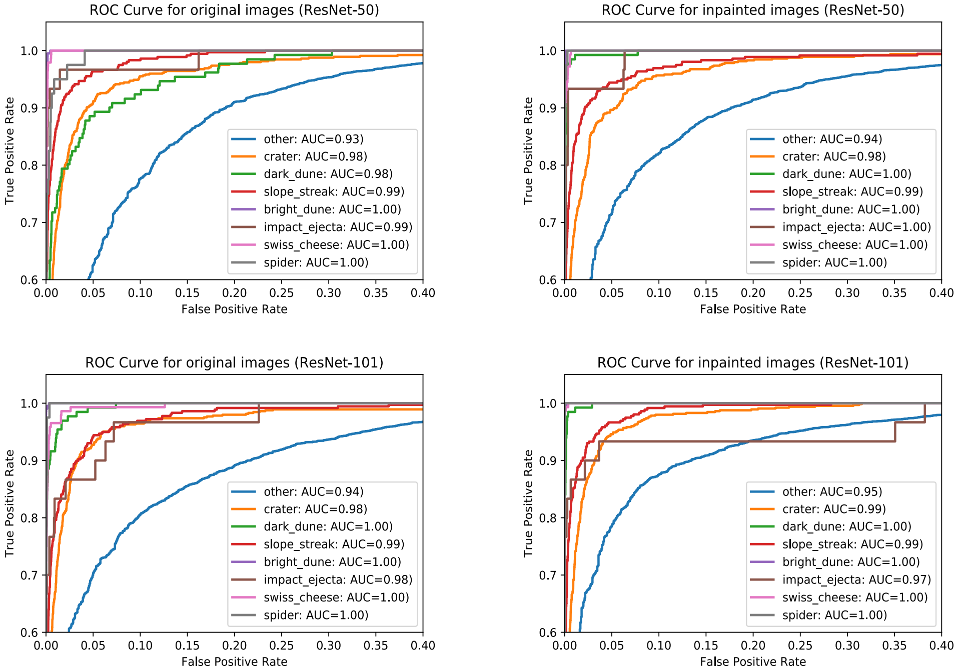


Figure 3-14: ROC curves corresponding to ResNet-50 and ResNet-101 for original and inpainted HiRISE dataset. For most classes, AUROC for inpainted images is better than original images.

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.

Chapter 4

Image Inpainting using Frequency Domain Priors

This chapter presents a novel image inpainting technique using frequency domain information. Prior works on image inpainting predict the missing pixels by training neural networks using only the spatial domain information. However, these methods still struggle to reconstruct high-frequency details for real complex scenes, leading to a discrepancy in color, boundary artifacts, distorted patterns, and blurry textures. To alleviate these problems, in this chapter, it is investigated if it is possible to obtain better performance by training the networks using frequency domain information (Discrete Fourier Transform) along with the spatial domain information. To this end, a frequency-based deconvolution module is proposed that enables the network to learn the global context while selectively reconstructing the high-frequency components. The proposed method is evaluated on the Mars dataset and it is shown that the proposed method using both frequency and spatial domain information outperforms current state-of-the-art image inpainting techniques both qualitatively and quantitatively.

Additional experiments are performed on the standard datasets namely CelebA, Paris Streetview, and DTD texture dataset as well to check the validity of the proposed algorithm. Here also, the proposed method could outperform state-of-the-art image inpainting techniques proving it's generalization ability.

4.1 Introduction

In computer vision, the task of filling in missing pixels of an image is known as image inpainting. It can be applied for improving data availability for satellite imagery. The main challenge in this task is to synthesize the missing pixels in such a way that they look visually realistic and coherent to human eyes. Traditional image inpainting algorithms [27, 24, 28, 105, 57, 55, 41, 159, 25, 44, 64] that use diffusion-based itechniques [27, 24, 28, 105] focus on propagating the local image appearance into the missing regions. Although these methods can fill in small holes but produce smoothed results as the hole grows bigger. On the other hand, patch-based traditional inpainting algorithms [57, 55, 41, 159, 25, 44, 64] iteratively search for the best-fitting patch in the image to fill in the missing region. These methods can fill in bigger holes, but they are not effective either in inpainting missing regions that have complex structures or in generating unique patterns or novel objects that are not available in the image in the form of a patch.

Recent research works on image inpainting [140, 76, 185, 164, 130, 186] leverage the advancements in generative models such as GANs [59] and show that it is possible to learn and predict missing pixels in coherence with the existing neighboring pixels by training a convolutional encoder-decoder network. In this paradigm, generally speaking, the model is trained in a two-stage manner - i) in the first stage, the missing regions are coarsely filled in with initial structures by minimizing traditional reconstruction loss; ii) in the second stage, the initially reconstructed regions are refined using an adversarial loss. Although these methods are good in generating visually plausible novel contents such as human faces, structures, natural scenes in the missing region, they still struggle to reconstruct high-frequency details for real complex scenes, leading to a discrepancy in color, boundary artifacts, distorted patterns, and blurry textures. Additionally, the reconstruction quality of previous methods deteriorates as the size of the missing region increases. The above problems can be attributed to the following reason. Existing methods use only spatial domain information during the learning process similar to diffusion like techniques to obtain information from the mask boundary. Thus as the mask size increases, the interior reconstruction details are lost and only a low-frequency component of the original patch is estimated by these methods.

To alleviate the above problem, a frequency-based image inpainting techique is proposed. It is shown that image inpainting can be converted to the problem of deconvolution in the frequency domain which can predict local structure in the missing regions using global context from the image. Qualitative analysis shows that the proposed frequency domain image inpainting approach helps in improving the texture details of missing regions. Previous methods make use of only spatial domain information. Therefore, the reconstruction of the information close to the mask boundary is good compared to the interior region since the local context is available only in the boundary regions. In contrast, a frequency-based approach would take information from the global context in the image because of Discrete Fourier Transforms (DFT) that considers all pixels for computing the frequency components. As a result, it captures more detailed structural and textural content of the missing regions in the learned representation. Due to these reasons, a two-stage network is proposed which consists of i) deconvolution stage and ii) refinement stage. In the first stage, the DFT image from the original grayscale/RGB image is computed. Each frequency component in the DFT image captures the global context thus forming a better representation of the global structure. A CNN is employed to learn the mapping between masked DFT and original DFT, which is a deconvolution operation obtained by minimizing the ℓ_2 loss. While DFT based deconvolution can reconstruct the global structural outline, it is observed that there exists a mismatch in the color space of the first stage output. Therefore, in the second stage, the output of the first stage is fine-tuned using adversarial methods to match the pixel distribution of the true image. Figure 4-1 shows an example of the reconstructed output using the proposed method where Figure 4-1b) shows the DFT map of the first stage reconstruction obtained from the deconvolution network). This additional frequency domain information is later used by the refinement network to obtain the final output as shown in Figure 4-1c). The main contributions in this chapter can be summarized as follows:

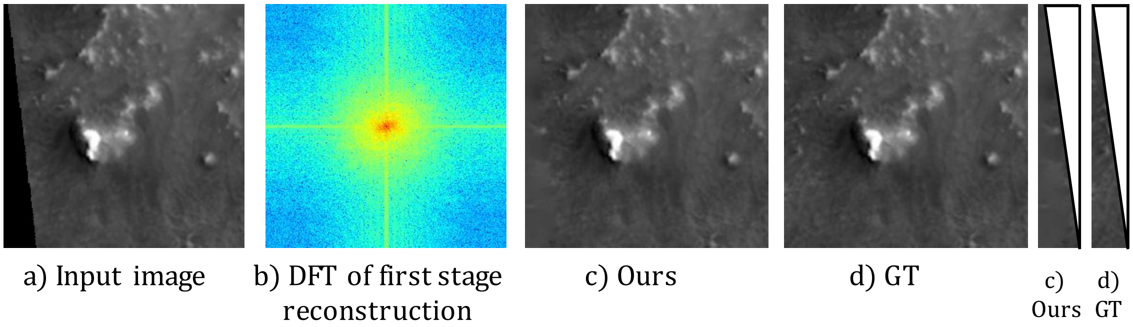


Figure 4-1: a) Artificially corrupted input Mars images with unphotographed/missing pixels, b) DFT of first stage reconstruction by the proposed deconvolution network, c) Image inpainting results (after the second stage) of the proposed approach, and d) Ground Truth (GT) image. The last two columns show the prediction of the missing region obtained from the proposed method and original pixel values for the same region in the GT image.

1. A novel frequency domain-based image inpainting framework is introduced that learns the high-frequency component of the masked region by using the global context of the image. It is found that the network learns to preserve image information in a better way when it is trained in the frequency domain. Therefore, adding the frequency domain and spatial domain information certainly improves the inpainting performance compared to the conventional spatial domain image inpainting algorithms. To enable better inpainting, the network is trained using both frequency-domain and spatial domain information which leads to a better consistency of inpainted results in terms of the local and global context.

2. The proposed method is validated on Mars dataset and it is shown that the proposed method achieves better inpainting results in terms of visual quality and evaluation metrics outperforming the state-of-the-art results. Additional experiments are performed on other benchmark datasets including CelebA faces, Paris Streetview, and DTD texture datasets. Experimental results demonstrate that the proposed algorithm can outperform state-of-the-art inpainting results both qualitatively and quantitatively for standard datasets as well.

3. To the best of our knowledge, this is the first work that explores the benefits

of using frequency domain information for image inpainting on both planetary images and standard datasets which proves the generalization ability of the proposed method.

4.2 Related Work

4.2.1 Image Inpainting on Remote Sensing and Planetary Dataset

While image completion on Earth remote sensing images has been studied widely, image inpainting on planetary images (Moon or Mars images) is still a new area to explore. This subsection reviews the existing works on image inpainting both on Earth remote sensing (RS) images and planetary images.

Missing pixels on Earth RS images occur in the form of dead pixels or thick cloud cover because of the atmospheric environment or the working conditions of the satellite sensor [158, 194]. There have been a lot of work in this direction using spatial information and based on interpolation techniques [190, 192, 184], variation-based algorithms [36, 69], PDE techniques [121], and exemplar methods [41]. Since spatial-based methods are not effective in precise reconstruction for large missing regions, several other techniques such as spectral-based methods (utilizing information from different spectral bands) [145, 182], and temporal-based methods (using data taken at the same location in different periods) [189, 34] have been proposed. Later more generalized algorithms known as hybrid methods [194, 81, 37, 68, 193] were developed by integrating spatial, spectral, and temporal information.

On the other hand, missing pixels on planetary images occur because of several reasons: (i) onboard instrument could not acquire the image data at that region due to the limitation in operation time of the instrument and satellite orbiter control, (ii) onboard cameras failed to photograph the region on the planetary surface (e.g. Polar region) because the region was not illuminated by the Sunlight. To restore such missing pixels, Roy et al. [156] first 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. Furthermore, Roy et al. [155] showed that it is possible to predict such unphotographed pixels on planetary images (Moon and Mars) in a context-aware fashion using adversarial learning algorithms. They use only spatial domain information to solve this task.

4.2.2 Image Inpainting on Standard Dataset

Image inpainting on standard datasets is a well-studied topic. Early image inpainting techniques explored *diffusion-based image completion methods* [27, 24, 28, 105], where a diffusive process is modeled using Partial Differential Equations (PDE) to propagate colors into the missing regions and *patch-based techniques* [57, 55, 41, 159, 25, 44, 64, 135] where similar patches are iteratively searched in an existing image to stitch it onto the most similar block of the image. However, these methods are not effective in the case of filling in complex structures or larger missing regions.

Recently CNN models [102] have shown tremendous success in solving high-level tasks such as classification, object detection, and segmentation as well as low-level tasks such as image inpainting problem. Several methods based on Stacked Sparse Denoising Auto-encoders (SSDA) [177], fully convolutional network (FCN) [33] were proposed to solve image inpainting tasks. However, these inpainting approaches were limited to small-sized masks as well. More recently, adversarial learning-based inpainting algorithms have shown promising results in solving image inpainting problems because of their ability to learn and synthesize novel and visually plausible contents for different images such as objects [140], scene completion [76], faces [183] etc. A seminal work by Pathak et al. [140] showed that their proposed Context Encoder network can predict missing pixels of an image based on the context of the surrounding areas of that region. They used both standard ℓ_2 loss and adversarial loss [59] to train their network. Later, Iizuka et al. [76] demonstrated that their encoder-decoder model could reconstruct pixels in the missing region that are consistent both locally and globally, by leveraging the benefits of dilated convolution layers, a variant of standard convolutional layers.

Recently, Yu et al. [185] introduced the concept of attention for solving an image inpainting task by proposing a novel contextual attention layer and trained the unified feedforward generative network with reconstruction loss and two Wasserstein GAN losses [20, 61]. They showed that their method can inpaint images with multiple missing regions having different sizes and located arbitrarily in the image.

Nazeri et al. [130] introduced an edge generator that at first predicts the edges of the missing regions and then use the predicted edges as guidance to complete the image. Yu et al. [186] proposed a gated convolution-based approach to handle free-form image completion.

4.2.3 Frequency Domain Learning

Recently enabling the network to learn information in the frequency domain has gained popularity because the frequency domain information contains rich representations that allow the network to perform the image understanding tasks in a better way in comparison to the conventional way of using only spatial domain information. Gueguen et al. [60] proposed image classification using features from the frequency domain. Xu et al. [179] showed that it is possible to perform object detection and instance segmentation by learning information in the frequency domain with a slight modification to the existing CNN models that use RGB input. This chapter proposes to using frequency-domain information along with spatial domain information to achieve better image inpainting performance.

4.3 Proposed Method

Given a corrupted input image, the aim is to predict the missing region similar to its surrounding context. This chapter proposes a frequency-based non-blind image inpainting framework that consists of two stages: i) frequency domain deconvolution network and ii) refinement network. The overall framework of the proposed method is shown in Figure 4-2. In the first stage, the DFT of the masked image (both magnitude and phase information) and the original input image is computed and then a CNN for deconvolution is trained so that it learns the mapping between the two signals by minimizing the ℓ_2 loss. Here the problem of inpainting in the spatial domain is formalized as deconvolution in the frequency domain. Here the feed-forward denoising convolutional neural networks (DnCNNs) [191] is employed, which is a manifestation of deconvolution and uses residual learning to predict the denoised image. The motivation behind this DFT-based deconvolution operation is to learn a better representation of the global structure that can serve as guidance to the second network. In the second stage, the spatial domain information (of the masked image and the mask) is used to train a generative adversarial network (GAN) based model [59] by minimizing an adversarial loss along with ℓ_2 loss. The motivation to incorporate this stage is to fine-tune the output of the first stage by refining the structural details and matching the pixel distribution of the true image in a local scale. The various components of the proposed model are explained in the following subsections.

4.3.1 Frequency-domain Deconvolution Network

4.3.1.1 Problem Formulation

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

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

$$|\mathbf{M}^f(p,q)|^2 \propto \frac{sin(\pi p)}{sin(\frac{\pi p}{N})} \frac{sin(\pi q)}{sin(\frac{\pi q}{N})}, \tag{4.1}$$

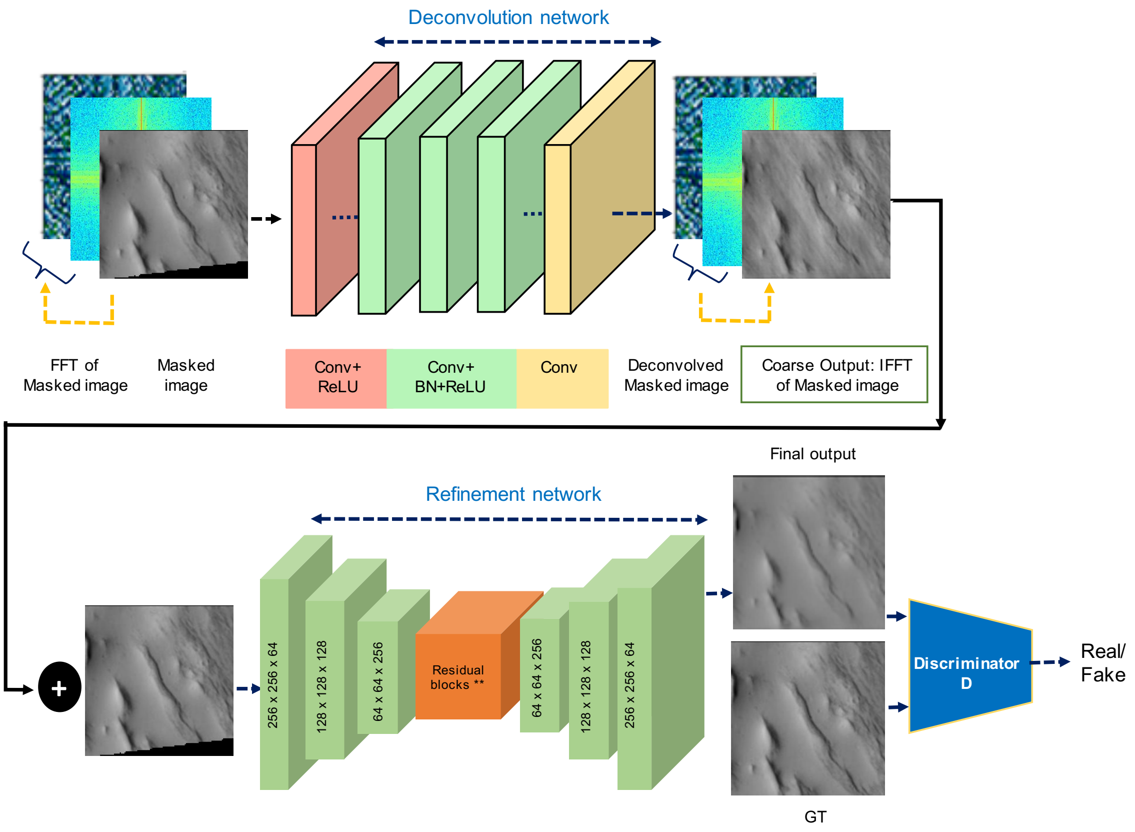


Figure 4-2: Overview of the proposed frequency domain-based image inpainting framework. The deconvolution network is trained in the frequency domain with ℓ_2 loss to learn the mapping between DFT of masked image and the original image. The refinement network is trained in the spatial domain with adversarial loss.

where $k = 0, 1, ...(N-1)$ represents the discrete frequency, with N being the number of samples. The frequency domain representation of the mask signal is shown in Figure 4-3, which depicts a decaying pulse from the origin. By the convolution-multiplication property of DFT, it can be shown that the multiplication of mask with the image in spatial domain is equivalent to convolution of mask with image in frequency domain (Figure 4-3). Mathematically, this is represented as

$$\mathbf{I^f_{in}}(p,q) = \mathbf{I^f_{gt}}(p,q) \circledast \mathbf{M}^f(p,q) \tag{4.2}$$

where $\circledast$ denotes the convolution operation and the masked frequency signal is the output of the convolution of the mask and clean image DFT signal. Therefore, a deconvolution operation is performed to predict the missing region of the incomplete

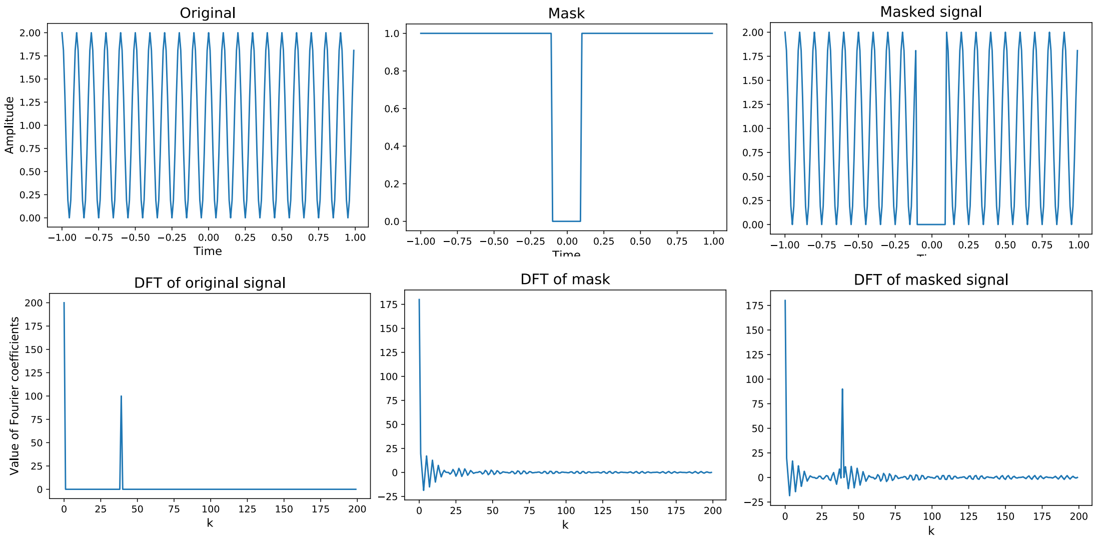


Figure 4-3: Visualization of masked signal in frequency domain (using DFT). Here, the convolution-multiplication property of DFT is used to transform signals from spatial to frequency domain and vice-versa.

Table 4.1: First stage network architecture (Deconvolution network).

Layer name	**Layer No.**	**Stride, Padding**	**Activation**	**Layer output size (Mars)**	**Layer output size (Standard dataset)**
Input	-	-	-	$1 \times 4 \times 256 \times 256$	$1 \times 12 \times 64 \times 64$
Conv 3×3	1	1, 1	ReLU		
Conv 3×3	2-16 (15 layers)	1, 1	(Batch Norm + ReLU)		
Conv 3×3	17	-	-	$1 \times 2 \times 256 \times 256$	$1 \times 6 \times 64 \times 64$

image. Let $\mathbf{F}(\mathbf{I_{in}}; \mathbf{\Theta})$ be the Deconvolutional neural network that converts $\mathbf{I_{in}}$ to $\mathbf{I^1_{pred}}$, such that $\mathbf{I^1_{pred}} = \mathbf{F}(\mathbf{I_{in}}; \mathbf{\Theta})$. After calculating $\mathbf{I^f_{in}}$ and $\mathbf{I^f_{gt}}$, the network is trained to learn the mapping between them, to predict the first stage output. Let us denote frequency domain representation as $\mathbf{I^{1f}_{pred}}$ where $\mathbf{I^{1f}_{pred}} = \mathbf{F}(\mathbf{I^f_{in}}; \mathbf{\Theta})$. Next, an inverse DFT of the first stage output is performed and the predicted output image is obtained as $\mathbf{I^1_{pred}} = \text{IDFT}(\mathbf{I^{1f}_{pred}})$.

4.3.1.2 Network Architecture

To perform the deconvolution operation in the frequency domain, a CNN model having 17 layers similar to [191] is adopted. This deconvolution network contains three types of layers as shown in Figure 4-2. The first layer is a Conv layer with ReLU non-linearity where 64 filters of (3x3x3) size are used. Next layers (2^{nd}-16^{th}) are a combination of Conv layer, a batch normalization layer [78] and a ReLU layer, where 64 filters of ($3\times3\times64$) size are used. The last layer is a Conv layer, where 3 filters of (3x3x64) size are used to reconstruct the output. Details of the first stage deconvolution network used for both Mars and standard datasets is given in Table 4.1.

4.3.1.3 Training

To train the proposed deconvolution network, ℓ_2 loss is used which minimizes the distance between the DFT of ground-truth image $\mathbf{I^{f}_{gt}}$ and the DFT of inpainted image $\mathbf{I^{1f}_{pred}}$, which is given by

$$\mathcal{L}_{s1} = \left\| \mathbf{I^{f}_{gt}} - \mathbf{I^{1f}_{pred}} \right\|_2^2 \tag{4.3}$$

After training the first stage deconvolution network, the inverse DFT of $\mathbf{I^{1f}_{pred}}$ is computed and it is used as a guidance to train the refinement stage as shown in Figure 4-2. The reason to choose the frequency domain in the first network is that it contains rich information [179, 180] for high-frequency preservation.

4.3.2 Refinement Network

The refinement network is a GAN based model [59] that has shown promising results in generative modeling of images [144] in recent years. The refinement network has a generator and a discriminator network, where the generator network takes the output of the first stage (frequency domain deconvolution module), the original masked image, and the corresponding binary mask (spatial domain information) as input pairs, and outputs the generated image. The discriminator network takes this generator output and minimizes the Jensen–Shannon divergence between the input and output data distribution to match the color distribution and structural details of the output image to the true image.

4.3.2.1 Network Architecture

Generator: The generator architecture is adopted from Johnson et al. [83] that has exhibited good performance for image-to-image translation task [196]. The generator network is an encoder-decoder architecture having three convolution layers for down-sampling, eight residual blocks [67], and three convolution layers for up-sampling. Here, Conv-2 and Conv-3 layers are stride-2 convolution layers that are responsible for down-sampling twice, and Conv-4 and Conv-5 layers are transpose convolution layers that are responsible for up-sampling twice back to the original image size. Instance normalization [170] and ReLU activation function are used across all layers of the generator network.

Discriminator: The discriminator network is adpoted from [79, 196] which is a Markovian discriminator similar to 70×70 PatchGAN. The advantage of using a PatchGAN discriminator is that it has fewer parameters compared to a standard discriminator because it works only on a particular image patch instead of an entire image. Furthermore, it can be applied to any arbitrarily-sized images in a fully convolutional fashion [79, 196]. The sigmoid function is applied after the last convolution layer which produces a 1-dimensional output score and predicts whether the 70×70 overlapping image patches are real or fake. To stabilize the discriminator network training, spectral normalization [124] is used as the weight normalization method. Moreover, leaky ReLUs [116] with slope of 0.2 is used. The details of the proposed second stage refinement network (generator and discriminator network) and output size of each layer for both the Mars dataset and standard datasets are given in Table 4.2.

Table 4.2: Second stage network architecture

Generator network				
Layer name	**Stride**	**Activation**	**Layer output size (Mars)**	**Layer output size (Standard dataset)**
Input	-	-	$1 \times 3 \times 256 \times 256$	$1 \times 9 \times 64 \times 64$
		Encoder network		
Conv 7×7	1	ReLU	$1 \times 64 \times 256 \times 256$	$1 \times 64 \times 64 \times 64$
Conv 4×4	2	ReLU	$1 \times 128 \times 128 \times 128$	$1 \times 128 \times 32 \times 32$
Conv 4×4	2	ReLU	$1 \times 256 \times 64 \times 64$	$1 \times 256 \times 16 \times 16$
		Residual block ($\times 8$)		
Residual blocks			$1 \times 256 \times 64 \times 64$	$1 \times 256 \times 16 \times 16$
		Decoder network		
Conv 4×4	2	ReLU	$1 \times 128 \times 128 \times 128$	$1 \times 128 \times 32 \times 32$
Conv 4×4	2	ReLU	$1 \times 64 \times 256 \times 256$	$1 \times 64 \times 64 \times 64$
Conv 7×7	1	tanh	$1 \times 1 \times 256 \times 256$	$1 \times 3 \times 64 \times 64$

Discriminator network				
Layer name	**Stride**	**Activation**	**Layer output size (Mars)**	**Layer output size (Standard)**
Input	-	-	$1 \times 1 \times 256 \times 256$	$1 \times 3 \times 64 \times 64$
Conv 4×4	2	LeakyReLU	$1 \times 64 \times 128 \times 128$	$1 \times 64 \times 32 \times 32$
Conv 4×4	2	LeakyReLU	$1 \times 128 \times 64 \times 64$	$1 \times 128 \times 16 \times 16$
Conv 4×4	2	LeakyReLU	$1 \times 256 \times 32 \times 32$	$1 \times 256 \times 8 \times 8$
Conv 4×4	1	LeakyReLU	$1 \times 512 \times 31 \times 31$	$1 \times 512 \times 7 \times 7$
Conv 4×4	1	Sigmoid	$1 \times 1 \times 30 \times 30$	$1 \times 1 \times 6 \times 6$

Algorithm 3 Training the refinement network.

```
while Generator G has not converged do
    Sample batch images $\mathbf{I_{in}}$ from training data;
    Generate random masks $\mathbf{M}$;
    Construct combined input ($\mathbf{I_{in}}$, $\mathbf{M}$, and $\mathbf{I^1_{pred}}$);
    Get masked region prediction $\mathbf{I^2_{pred}} = G(\mathbf{I_{in}}, \mathbf{M}, \mathbf{I^1_{pred}})$;
    Generate inpainted image by modifying the masked region $\mathbf{I_{pred}} \leftarrow \mathbf{I_{in}} + \mathbf{I^2_{pred}} \odot (\mathbf{1} - \mathbf{M})$;
    Update G with $\ell_1$ loss and adversarial critic loss;
    Update discriminator critic D with $\mathbf{I_{in}}$, $\mathbf{I_{pred}}$;
end while
```

4.3.2.2 Training

After obtaining the first stage output, it is fed to the refinement network along with the spatial domain information (of the masked image and the mask). While training, the generator of the inpainting network G takes a combination of *input image* $\mathbf{I_{in}}$, *image mask* $\mathbf{M}$, and the *first stage output image* $\mathbf{I^1_{pred}}$ and generates $\mathbf{I^2_{pred}} = G(\mathbf{I_{in}}, \mathbf{M}, \mathbf{I^1_{pred}})$ as output. Then by adding $\mathbf{I^2_{pred}}$ to the *input image*, *completed image* is obtained as $\mathbf{I_{pred}} = \mathbf{I_{in}} + [\mathbf{I^2_{pred}} \odot (\mathbf{1} - \mathbf{M})]$. The training procedure of the refinement stage is described in Algorithm 3. The proposed refinement module is trained by using two loss functions: a reconstruction loss and an adversarial loss [59]. Here for the reconstruction loss, ℓ_1 loss [140] is used, that minimizes the distance between the clean/ground-truth image $\mathbf{I_{gt}}$ and the completed/inpainted image $\mathbf{I_{pred}}$, which is given by

$$\mathcal{L}_{\ell_1}(x) = \|\mathbf{I_{gt}} - \mathbf{I_{pred}}\|_1, \tag{4.4}$$

where $\mathbf{I_{pred}} \leftarrow \mathbf{I_{in}} + G(\mathbf{I_{in}}, \mathbf{M}, \mathbf{I^1_{pred}}) \odot (\mathbf{1} - \mathbf{M})$. For the adversarial loss, the min-max optimization strategy is followed, 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

Table 4.3: Quantitative results on Mars dataset for different inpainting models: Generative Inpainting (GI) [186], the proposed method in spatial domain [155], and the proposed method (using frequency domain). The best results for each row is shown in bold. $^{-}$Lower is better. $^{+}$Higher is better.

	GI [186]	Ours (spatial domain)	Ours (frequency domain)
PSNR^{+}	31.04	33.42	**33.80**
SSIM^{+}	0.906	0.928	**0.932**
$\ell_1(\%)^{-}$	1.4	0.9	**0.9**

predictions/inpainted samples. The objective function can be expressed as follows

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

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

$$\mathcal{L}_{total} = \lambda_1 \mathcal{L}_{\ell_1} + \lambda_2 \mathcal{L}_{adv}, \tag{4.5}$$

where $\lambda_1 = 1, \lambda_2 = 0.1$. The weighted sum of these two loss functions compliments each other in the following way: i) The GAN loss helps to improve the realism of the inpainted images, by fooling the discriminator. ii) The ℓ_1 reconstruction loss serves as a regularization term for training GANs [185], 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.

4.3.2.3 Implementation Details

The proposed model is implemented in PyTorch. [1] The Mars dataset have images of size 227×227 pixels and for experimental purpose, they are resized to size 256×256.

[1] Our code is available at `https://github.com/hiyaroy12/DFT_inpainting`.

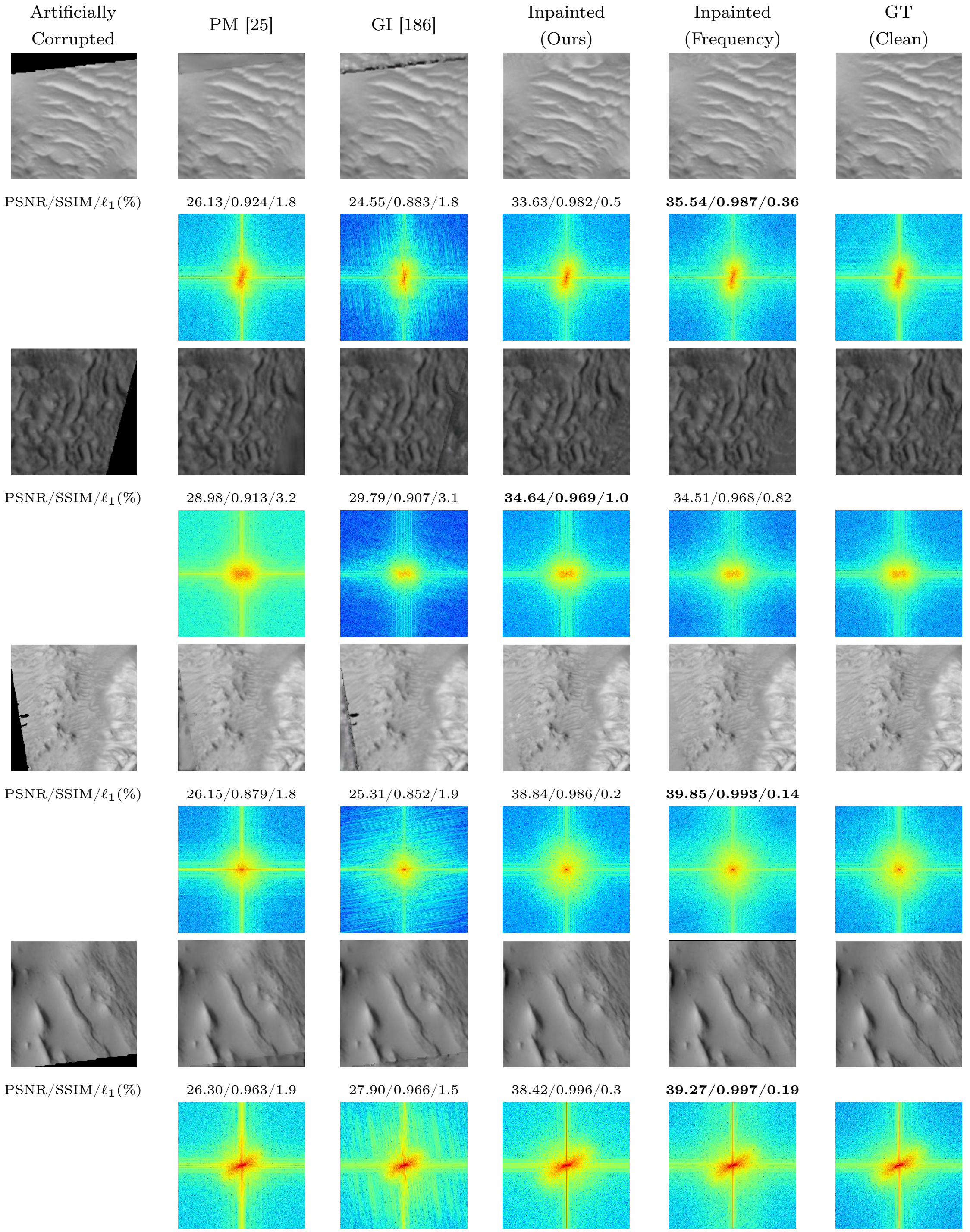


Figure 4-4: Visual examples of semantic feature completion of *simulated/artificially corrupted images* on Mars dataset using different methods: PatchMatch [25], Generative Inpainting (GI) [186], the proposed method using only spatial domain information and the proposed method using both frequency and spatial domain information. The DFT maps corresponding to different methods are shown here.

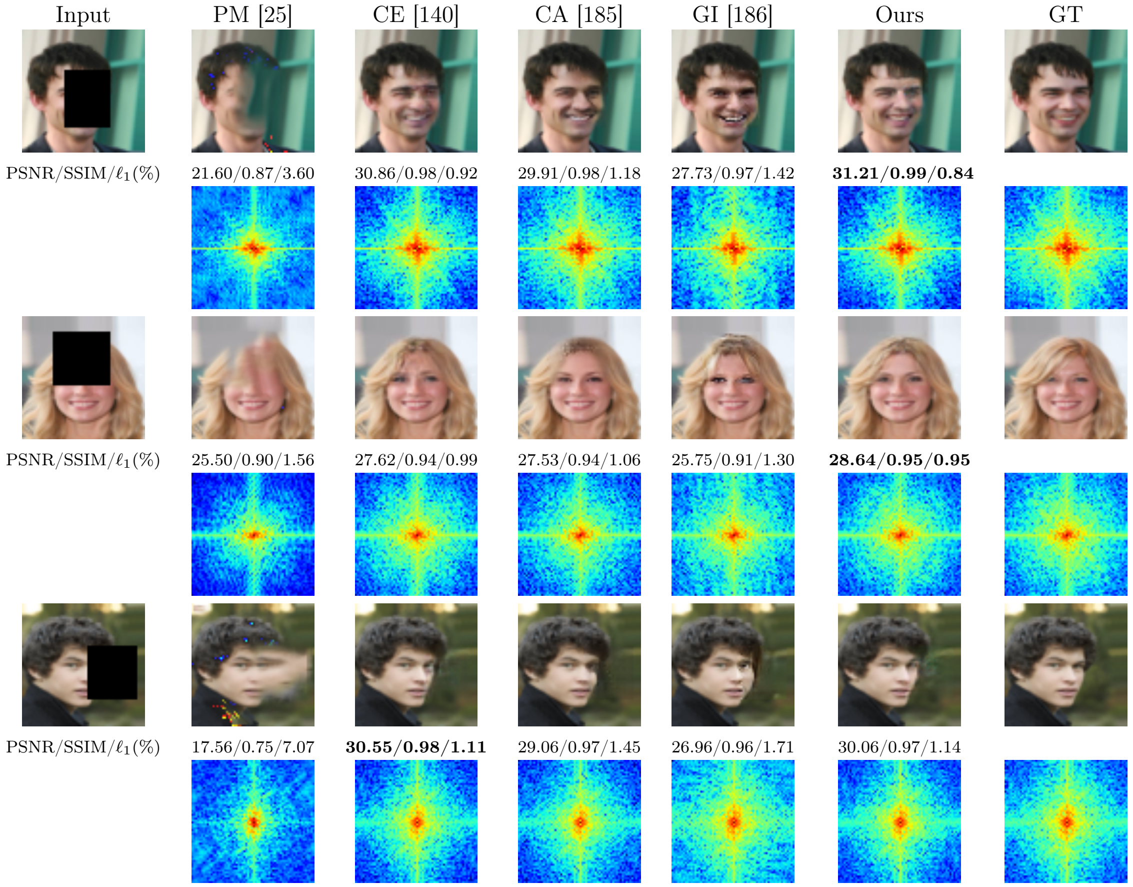


Figure 4-5: Visual comparison of semantic feature completion results for different methods on CelebA dataset along with the DFT maps corresponding to different methods, the first stage output, and GT image.

During additional experiments on the standard dataset, the images are resized to 64×64 and linearly scale the pixel values from range $[0, 256]$ to $[-1, 1]$. For the first stage, the weights are initialized by using He initialization [66] and use SGD optimizer with weight decay of 0.0001, the momentum of 0.9, and mini-batch size of 128. To train the first stage network the learning rate is decayed exponentially from 10^{-1} to 10^{-4} for 50 epochs. For the second stage, both the Generator G and Discriminator D are trained together using the following settings: i) G and D learning rate of 10^{-4}, and 10^{-5} respectively, ii) optimized using Adam optimizer [93] with $\beta_1 = 0.5$ and $\beta_2 = 0.999$. In the experiments, a batch size of 14 and the training iterations of 100 are used. Both stages are implemented on a TITAN Xp (12 GB) GPU.

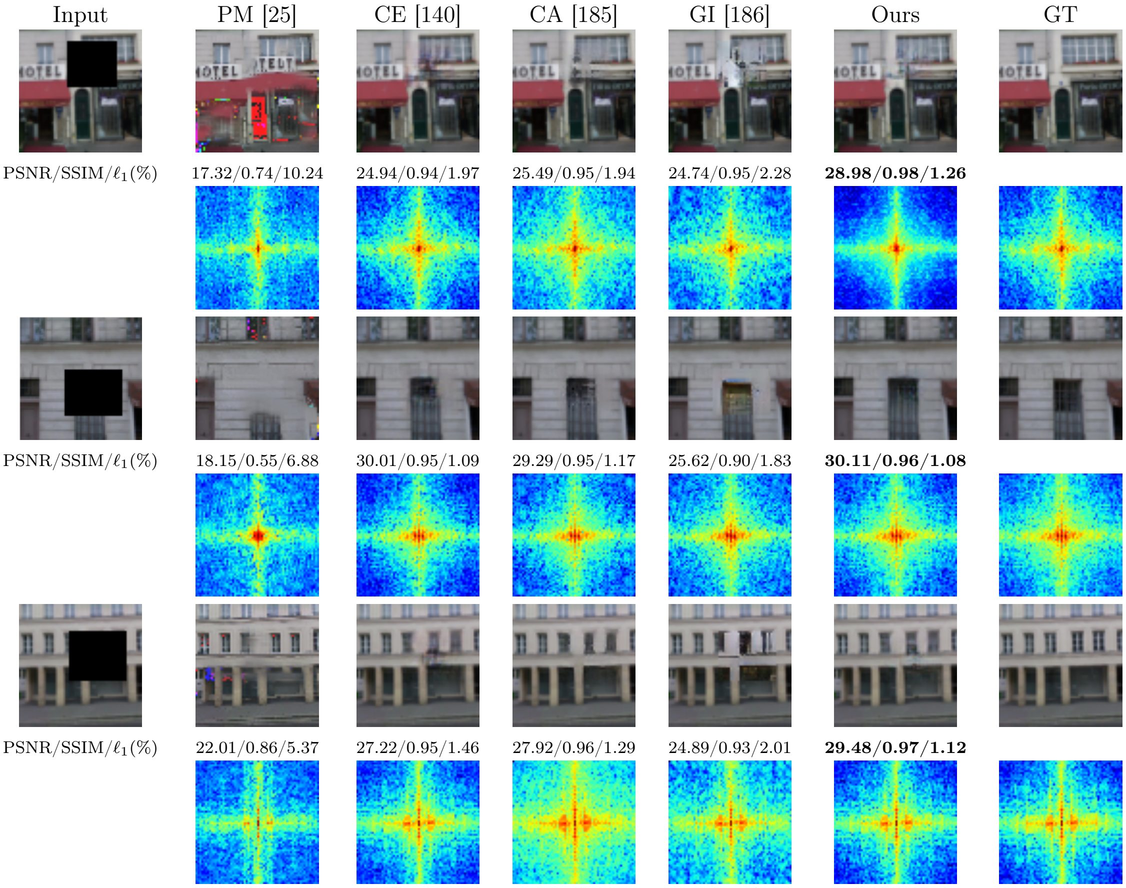


Figure 4-6: Visual comparison of semantic feature completion results for different methods on Paris Streetview dataset along with the DFT maps corresponding to different methods, the first stage output, and GT image.

4.4 Experiments

This section evaluates the inpainting performance of the proposed method on the grayscale-version of the Mars orbital images collected by the HiRISE camera onboard the MRO having a spatial resolution of approximately 30 cm/pixel [120]. The Mars dataset consists of a total of 73,031 landmarks amongst which 10,433 landmarks are detected and extracted from 180 HiRISE browse images [2]. Amongst these 52655 images are clean and 15,654 images are corrupted (have missing regions at the extremities of the images). These regions, extracted as masks, are artificially superimposed on clean images to create pairs of clean and corrupted images. For training purposes, 42124 pairs of clean and artificially corrupted images are used and 10531 images are

[2] This dataset is available at `https://zenodo.org/record/2538136#.XYjEuZMzagR`.

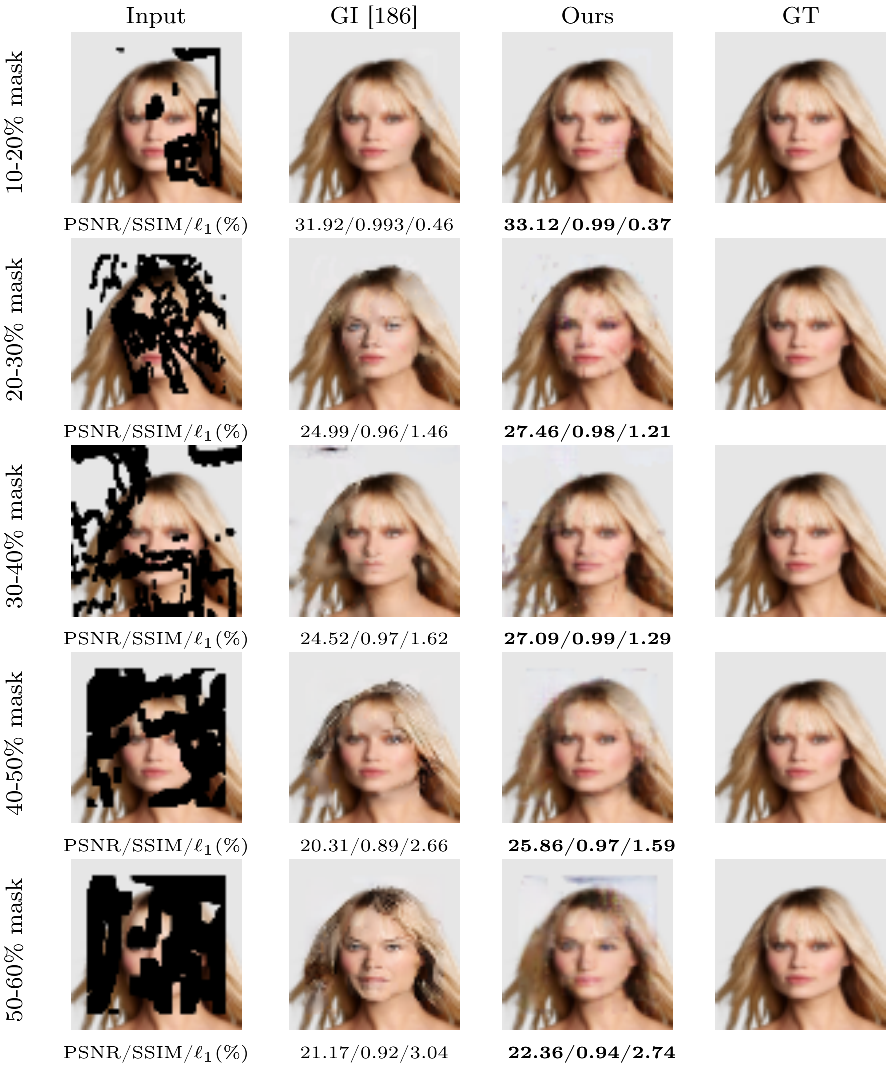


Figure 4-7: Visual comparison of semantic feature completion results for irregular masks on CelebA dataset.

used for testing purposes.

Additional experiments performed on three standard datasets: CelebFaces Attributes Dataset (CelebA) [111], Paris StreetView (PSV) [52], and Describable Texture Dataset (DTD) [38] as well. The CelebA face dataset contains 162770 training images and 19867 test images. The ParisStreetView dataset has 14900 images in the training set and 100 images in the test set. DTD texture dataset has 5076 number of training images and 564 number of test images. For experiments on these datasets, both regular and irregular masks are used. Regular masks refer to square masks having a fixed size consisting of 25% of total image pixels and are randomly located in the image. For

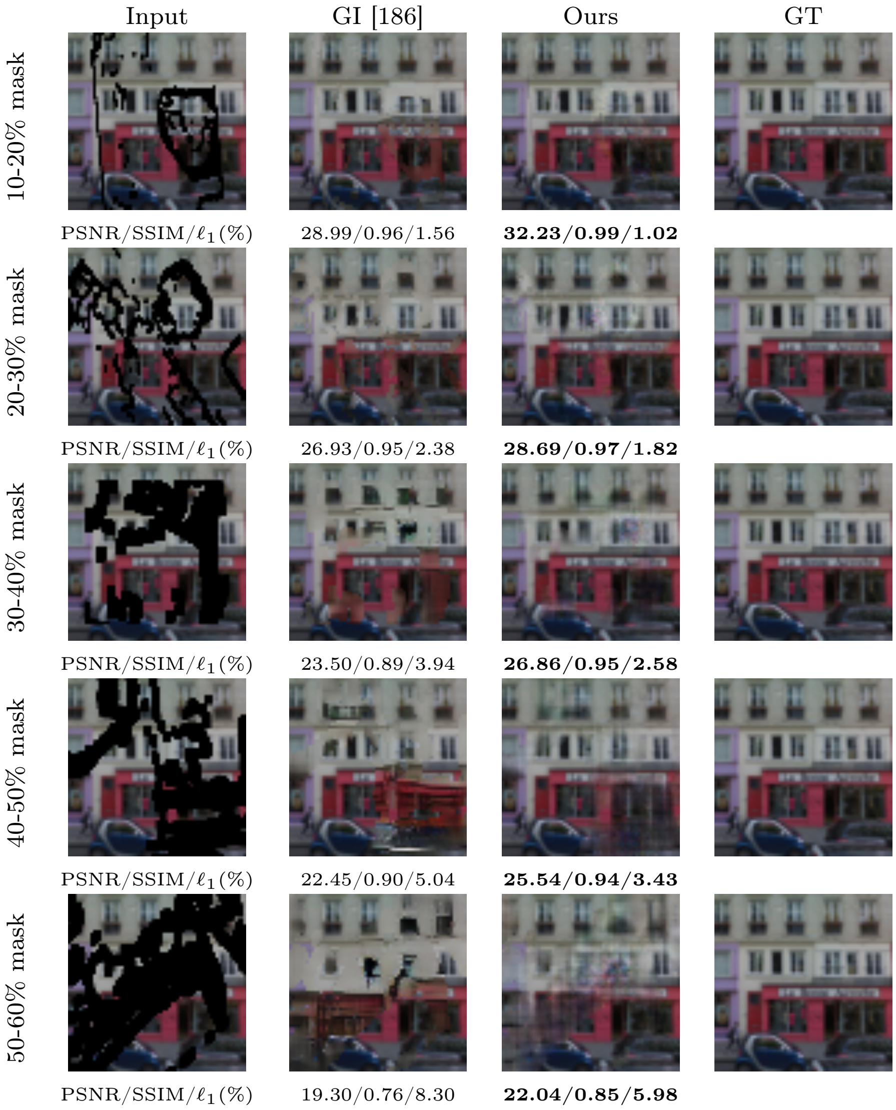


Figure 4-8: Visual comparison of semantic feature completion results for irregular masks on Paris StreetView dataset.

irregular masks, during training, the masks from the work of Liu et al. [109] are used, where the irregular mask dataset contains the augmented versions of each mask (0, 90, 180, 270 degrees rotated, horizontally reflected) and are divided based on the percentage of mask size on the image in increments of 10% e.g. 0-10%, 10-20%, etc.

4.4.1 Qualitative Evaluation

Figure 4-4 shows the inpainting results on Mars images using different inpainting algorithms: PatchMatch (PM) [25], Generative Inpainting (GI) [186], proposed method

Table 4.4: Quantitative results on CelebA [111] for different inpainting models: PatchMatch (PM) [25], Context Encoder (CE) [140], Contextual Attention (CA) [185], Generative Inpainting (GI) [186], and the proposed method. The best results for each row is shown in bold. $^{-}$Lower is better. $^{+}$Higher is better.

		CelebA dataset				
	Mask	PM [25]	CE [140]	CA [185]	GI [186]	Ours
PSNR^{+}	10-20%	15.78	32.49	29.81	30.65	**32.69**
	20-30%	15.09	29.62	27.06	27.22	**29.78**
	30-40%	14.42	27.31	24.77	24.83	**27.49**
	40-50%	13.63	25.10	23.03	22.86	**25.27**
	Regular	14.96	**28.17**	27.86	26.06	28.13
SSIM^{+}	10-20%	0.632	0.991	0.986	0.987	**0.992**
	20-30%	0.579	0.983	0.971	0.971	**0.984**
	30-40%	0.513	0.972	0.953	0.952	**0.973**
	40-50%	0.421	0.954	0.930	0.927	**0.956**
	Regular	0.571	0.970	0.968	0.953	**0.971**
$\ell_1\ (\%)^{-}$	10-20%	13.14	0.84	1.37	1.21	**0.82**
	20-30%	14.58	1.41	2.24	2.07	**1.39**
	30-40%	16.07	2.13	3.28	3.09	**2.09**
	40-50%	17.89	3.13	4.40	4.22	**3.08**
	Regular	13.67	1.55	1.76	2.12	**1.55**

in spatial domain [155], and proposed method (using frequency and spatial domain information). The magnitude spectrum of the DFT map for each case is provided in the row below for each result. It can be seen that the inpainting performance improves if the frequency information is used along with spatial information (second last column of Figure 4-4).

Figures 4-5 and 4-6 compare the inpainting results of the proposed method with previous image inpainting methods: PatchMatch (PM) [25], Context Encoder (CE) [140], Contextual Attention (CA) [185], and Generative Inpainting (GI) [186], for regular masks on CelebA and Paris StreetView datasets. The magnitude spectrum of the DFT map obtained from different methods [25, 140, 185, 186], the proposed method (first stage reconstruction), and the ground truth image are shown in the row below

Table 4.5: Quantitative results on Paris Streetview [52] for different inpainting models: PatchMatch (PM) [25], Context Encoder (CE) [140], Contextual Attention (CA) [185], Generative Inpainting (GI) [186], and the proposed method. The best results for each row is shown in bold. $^-$Lower is better. $^+$Higher is better.

		Paris Streetview dataset				
	Mask	PM [25]	CE [140]	CA [185]	GI [186]	Ours
PSNR$^+$	10-20%	22.03	31.59	30.68	30.42	**32.34**
	20-30%	20.42	28.69	27.40	27.09	**29.25**
	30-40%	19.36	27.02	25.42	24.95	**27.33**
	40-50%	18.52	25.09	23.99	23.23	**25.13**
	Regular	19.23	27.32	28.29	25.12	**28.42**
SSIM$^+$	10-20%	0.766	0.978	0.972	0.969	**0.981**
	20-30%	0.692	0.958	0.945	0.936	**0.963**
	30-40%	0.613	0.938	0.912	0.896	**0.942**
	40-50%	0.515	0.904	0.873	0.850	**0.910**
	Regular	0.659	0.923	0.934	0.880	**0.936**
ℓ_1 (%)$^-$	10-20%	6.15	1.09	1.40	1.44	**0.97**
	20-30%	7.78	1.93	2.45	2.52	**1.78**
	30-40%	9.39	2.70	3.43	3.66	**2.57**
	40-50%	10.8	3.75	4.40	4.79	**3.58**
	Regular	9.04	1.97	1.93	2.76	**1.77**

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

Table 4.6: Quantitative results on DTD texture dataset [38] for different inpainting models: PatchMatch (PM) [25], Context Encoder (CE) [140], Contextual Attention (CA) [185], Generative Inpainting (GI) [186], and the proposed method. The best results for each row is shown in bold. $^-$Lower is better. $^+$Higher is better.

		DTD texture dataset				
	Mask	PM [25]	CE [140]	CA [185]	GI [186]	Ours
PSNR$^+$	10-20%	22.43	29.28	28.43	29.29	**29.89**
	20-30%	21.11	27.02	25.73	26.34	**27.38**
	30-40%	20.12	25.33	23.76	24.41	**25.65**
	40-50%	19.26	23.89	22.35	22.75	**23.95**
	Regular	14.75	27.33	27.26	25.73	**27.49**
SSIM$^+$	10-20%	0.704	0.933	0.922	0.935	**0.942**
	20-30%	0.634	0.890	0.861	0.872	**0.901**
	30-40%	0.563	0.841	0.793	0.804	**0.854**
	40-50%	0.475	0.773	0.717	0.714	**0.785**
	Regular	0.149	0.876	0.869	0.833	**0.879**
ℓ_1 (%)$^-$	10-20%	7.87	1.87	1.92	1.81	**1.67**
	20-30%	8.85	2.85	3.02	2.93	**2.62**
	30-40%	9.76	3.82	4.20	4.11	**3.58**
	40-50%	10.70	4.94	5.40	5.43	**4.74**
	Regular	17.60	2.12	2.40	2.74	**2.05**

duce better inpainting results.

The qualitative performance of the proposed method in comparison with Generative Inpainting (GI) [186] algorithm on CelebA and Paris StreetView dataset for irregular masks is shown in Figure 4-7, and Figure 4-8 for different percentage (10-50%) of mask size. The proposed method can generate photo-realistic images having similar texture and structures as the original clean images even when a large region (50-60%) of the image is missing.

4.4.2 Quantitative Evaluation

The quantitative performance of the proposed method is reported in terms of the following metrics i) peak-signal-to-noise ratio (PSNR); ii) structural similarity index (SSIM) [173] and iii) mean absolute error (MAE). Table 4.3 provides the quantitative results on Mars dataset and compares the inpainting performance of previously

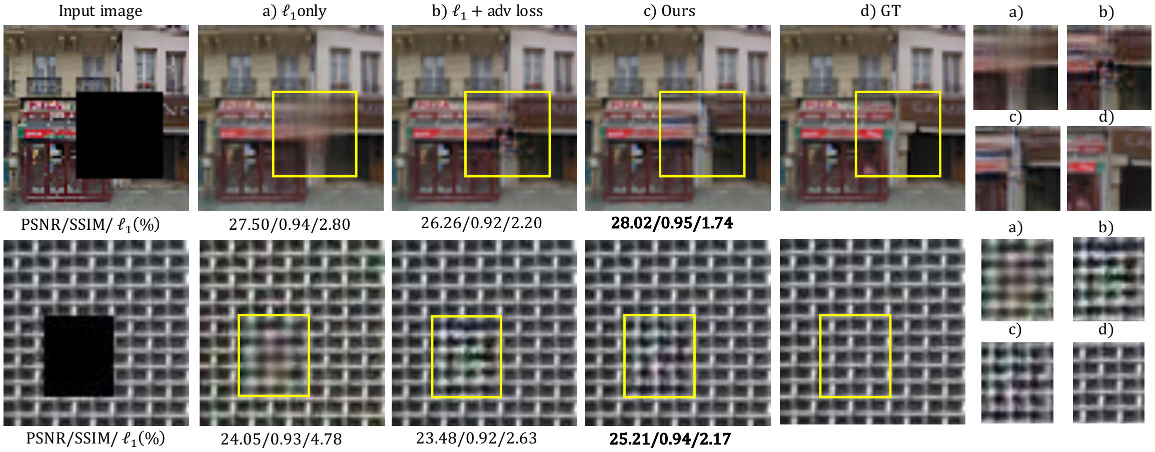


Figure 4-9: Visual results on Paris StreetView dataset (first row) and DTD (second row) showing the effect of different components in the proposed model on the input incomplete images (first column), a) results using standard ℓ_1 loss, b) results using ℓ_1 + adversarial loss, c) results of the proposed model trained using ℓ_1 + adversarial loss (with DFT component), and d) GT image.

proposed inpainting algorithms, GI [186], the proposed method using only spatial domain information [155] with the proposed method using both frequency and spatial domain information, in terms of the aforementioned metrics. The proposed method outperforms all the cases proving the usefulness of using the frequency domain information.

Tables 4.4, 4.5, and 4.6 demonstrate the comparison in metric values on the CelebA, Paris StreetView, and DTD dataset for the state-of-the-art inpainting methods and the proposed method. The frequency based approach outperforms previous methods in terms of these metrics on both regular and irregular masks. This proves the effectiveness of using frequency domain information. Note that, the metrics for Context Encoder [140] are obtained by using the ℓ_1 and adversarial loss in the proposed network settings.

It is to be noted that, the proposed inpainting algorithm using frequency domain and spatial domain information (explained in this chapter) (approach-2), is an extension of the inpainting algorithm using only spatial domain information proposed in the previous chapter (approach-1). Although the approach-2 performs better (both qualitatively and quantitatively) than approach-1; it consumes comparatively higher computational resources (in terms of training time) compared to the approach-1. This is because approach-1 is trained using only one stage, whereas approach-2 is trained in two stages. Therefore, computational resources vs. performance trade-off should be considered regarding the usage of the proposed inpainting algorithms.

4.4.3 Ablation Study

An ablation study is performed on standard datasets to investigate the role of the frequency deconvolution network and to analyze the effect of different loss components used to train the proposed model. Figure 4-9 shows the inpainting results using only ℓ_1 loss, ℓ_1 with adversarial loss and the proposed method of incorporating frequency domain information (DFT component). It can be seen in Figure 4-9a) that using only ℓ_1 loss in the spatial domain often cause blurry reconstructions. However, inpainting performance improves to a certain extent if the adversarial loss component is added. Nevertheless, in Figure 4-9b) the structural and blurry artifacts still exist on the reconstructions. Figure 4-9c) demonstrates the inpainting results of the proposed method of training the model using both frequency and spatial components. It can be seen that using this method the model can perform significantly better by restoring fine structural details. Therefore, it can be concluded that training the model along with frequency-domain information certainly helps the network to learn high-frequency components and restore the missing region with better reconstruction quality.

4.5 Conclusions

A frequency-based image inpainting algorithm is presented in this chapter that enables the network to use both frequency and spatial information to predict the missing region of an image. To the best of our knowledge, this is the first attempt to solve the image inpainting problem by using frequency-domain information which was not explored in previous inpainting works. The proposed model first learned the global context using frequency domain information and selectively reconstructed the high-frequency components. Then it used the spatial domain information as a guidance to match the pixel distribution of the true image and fine-tuned the details and structures obtained in the first stage, leading to better inpainting results. Experimental results showed that the proposed method could achieve results better than state-of-the-art performances on different kinds of challenging datasets (planetary and standard dataset) by generating sharper details and perceptually realistic inpainting results. This proved the generalization ability of the proposed algorithm. Based on the empirical results, I believe that methods using both frequency and spatial information should gain dominance because of their superior performance. In the future, this work can be extended to using other kinds of frequency domain transformations e.g. DCT, and solve other kinds of image restoration tasks e.g. image denoising.

Chapter 5

Conclusions

5.1 Discussion

Instruments and sensors onboard spacecraft and rover capture a plethora of data that can enable scientists to discover the yet unknown. Nevertheless, because of the limited bandwidth and inter-planetary communication through deep space, it is impossible to return all of the images to Earth for further analysis. Moreover, sometimes the received data on Earth is partially corrupted with unphotographed/missing pixels. Machine learning methods can provide solutions to these problems in the following ways: by analyzing data onboard (i.e. by detecting objects seen in the image); by automating time-consuming tasks for human such as finding images of interest, based on textual queries (image retrieval); or by predicting missing regions on the orbital images (image inpainting) to enhance the usability of the acquired data. The studies explained in the previous chapters demonstrate how ML can improve image analysis capabilities in future space missions by overcoming the data constraints from both in-situ and orbital planetary exploration missions.

For in-situ missions such as the MSL Curiosity rover, this dissertation demonstrated solutions to enhance onboard analysis capabilities by detecting objects seen in the image and sending this smaller sized meta-data back to Earth. This will not only help to overcome the problem of limited bandwidth but also will help to retrieve images from a large database based on query text. This study required curation of a new planetary dataset, along with labeling of each image (annotating objects and corresponding captions for the entire image) for developing a new machine learning method for planetary image retrieval using captions. This research was initially aimed to solve the limited bandwidth problem by developing onboard image analysis capabilities for better space exploration in the future. Nonetheless, the developed algorithm is also useful for other applications on the ground such as image retrieval.

For orbital missions such as the MRO or SELENE/Kaguya, this dissertation presented solutions related to image inpainting to predict the missing regions of the partially corrupted images (received on Earth) and enhance the data usability for further analysis such as automatically classifying or recognizing interesting morphological features in the planetary surface or improving the landing site candidate selection efficiency, etc. Two approaches related to solving the image inpainting task are proposed, where the first approach uses only spatial domain information, whereas the second approach takes advantage of the frequency domain information along with the spatial domain information to selectively reconstruct the high-frequency components of the missing region. The first image inpainting approach (as explained in Chapter 3) takes less time to train compared to the second inpainting approach (explained in Chapter 4) because the former is trained in one stage, whereas the latter is trained in two stages. However, the second approach performs better than the first one both qualitatively and quantitatively. Therefore, computational resources vs performance trade-off should be considered regarding the usage of the proposed inpainting algorithms. Experimental results showed that image inpainting can be a helpful first step for planetary scientists for further analysis such as to improve classification performance or 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. Overall, the solutions presented in this thesis demonstrate that, although there exist several data-constrained situations while exploring the planets and planetary bodies, it is possible to overcome them both on the planet-side (onboard applications) and Earth-side (ground applications) by leveraging the benefits of ML technology for better planetary exploration in the future.

5.2 Directions for Future Work

While ML can be a promising tool to solve several challenges faced by planetary scientists, there are a few challenges related to the availability of planetary datasets in public. Although this intersection of ML and planetary science is still in the nascent stage, there is a great potential in ML algorithms that can advance planetary science in near future. Both the planetary science and ML community can benefit from each other, where the ML community can gain new insights by developing new algorithms on planetary datasets that might not be revealed using only standard datasets; and the planetary science community can benefit themselves with automated algorithms that can improve future space mission. The next subsections present some ideas for future work as well as some efforts worth-considering to foster the long-term collaboration of these two fascinating fields of ML and planetary science.

5.2.1 Compression

To deal with the bandwidth-constrained situation to transmit images, one prominent future direction would be to compress the images. Currently, images captured by the rover are compressed onboard (in JPEG format) before they are downlinked to Earth. While compression is important in this regard, JPEG compression results in noticeable artifacts such as inconsistency in brightness or color [90]. Recently the outstanding performance of deep neural networks has attracted the computer vision community to use deep neural networks for solving lossy image compression tasks as well. Such deep learning-based image compression frameworks [167, 168, 22, 23, 17, 18, 150, 106]. Deep learning-based image compression is considered to be more generic, can be implemented quickly, and are efficient in terms of performance compared to standard JPEG codec. Therefore, in the future, I would like to develop deep learning compression algorithms for the planetary dataset.

5.2.2 Interpretability in Machine Learning

A great deal of conservative mindset works within the planetary science community when it comes to adopting the machine learning/deep learning solutions that are designed to benefit future space exploration. This is a valid concern because proper interpretations as to why the results obtained using deep learning algorithms is good or why certain decisions or predictions were made, will help the scientists to trust and adopt the system more logically. In recent years, interpretability in machine learning is booming and is being actively pursued by the research community. Within the planetary science domain also, it is important to bring interpretability in the machine learning/deep learning solutions so that scientists are more willing to adopt it for future space missions.

5.2.3 Making Planetary Datasets Publicly Available for Research Purposes

The Machine learning community usually evaluate the performance of any model on the benchmark datasets such as MNIST [103], Fashion-MNIST [176], CIFAR-10 [96], ImageNet [50], MS-COCO [108], and other standard datasets [38, 52, 111], depending on the task. Therefore, it is much easier to compare the performance of the proposed algorithm directly with the previously proposed methods on standard datasets. However, in the case of developing ML solutions for space application, often the planetary dataset is not publicly available or even if it is publicly available, sometimes domain knowledge is required to understand the nomenclature or the typicalities related to that dataset. For example, there are some publicly available datasets for Mars and Moon craters [141, 151], which are designed for Geographic Information System (GIS) analysis. However, to use them for ML tasks, domain expertise, as well as a lot of preprocessing, is required. Recently some studies have published the planetary datasets used by the authors along with the labeling [172]. However, this is not a common practice. Therefore, it is important to make planetary datasets publicly available to encourage the ML community, to develop better and new solutions for scientific analysis and mission operations, promote research works in this direction, and enable reproducibility of the published work. The NASA Planetary Data System a.k.a. PDS Imaging Node - NASA [8] is a promising image archive curating images from past and present planetary missions to look up for planetary datasets. Only time will reveal the true potential and benefits of these research directions for future planetary explorations.

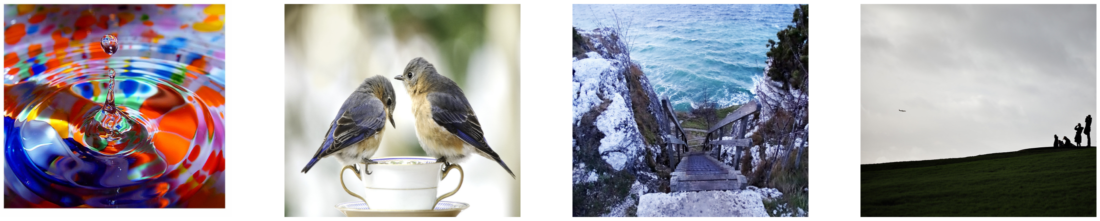
Appendix A

Image Aesthetics Analysis

Analyzing the aesthetic quality of images is a highly challenging task because of its subjectiveness. With the exponential rise of digital images in social media, it is of great demand to assess the aesthetics of images for several multimedia applications such as increasing social popularity, etc. Previous approaches to address this problem have used hand-designed features or automated features extracted by deep convolutional neural network architectures. In this chapter, the aesthetics of images is predicted by using the inferential information depending on the visual content found in an image. To the best of our knowledge, this is the first attempt to address such a problem by using the tags predicted. Experimental results show that the proposed method outperforms the traditional machine learning methods and demonstrate competitive performance compared to the state-of-the-art methods of image aesthetics prediction.

A.1 Introduction

In recent years, image aesthetics analysis has drawn a significant attention of the computer vision community because of its potential applications in the visual experience domain, such as image enhancement, image cropping, image retrieval, and photo management [46, 89, 115, 51, 157, 134]. Evaluating the aesthetics of images using a computational algorithm is a very difficult task for computers because differ-



Avg. score: 8.571 Avg. score: 6.882 Avg. score: 5.980 Avg. score: 4.463

Figure A-1: Photos with ratings given by viewers collected from www.dpchallenge.com

ent people perceive beauty in different ways and rate aesthetics of images differently as seen in Figure A-1. Therefore, to deal with the subjectivity of the human's aesthetic evaluation, several machine learning approaches have been proposed over the years [46, 89, 51, 118, 115, 45, 85, 114, 47, 132, 29, 160, 175, 82, 30, 157, 134]. Most of the early research works have focused on designing intuition-based features ranging from low-level features such as color, hue, saturation, etc. [46] to high-level describable image attributes such as compositional, content-based or illumination based attributes [51] for image aesthetics prediction. However, it is very difficult to choose appropriate features to map the human perception of images to their aesthetic score. To tackle this problem researchers have also adopted generic features such as SIFT and Fisher Vector [118] for predicting the aesthetics of images. However, over the past few years, deep convolutional neural network (CNN) based models have shown outstanding performance on various challenging visual recognition tasks [97, 39, 188, 161, 165, 67] and they have the capability of learning features automatically from image examples in a hierarchical way. Therefore, in the most recent studies researchers have exploited the automated feature learning power of deep convolutional neural network (CNN) for the image aesthetics prediction task, to avoid the requirement of domain-related knowledge to choose appropriate features.

In this work, a new approach is introduced to predict image aesthetics using tags generated by the computer vision API of the Microsoft Azure Cognitive Services. The generated tags are mostly object-names, but there are some context-related tags such as *indoor*, *outdoor*, etc. The AVA dataset [126] contains approximately 2,50,000 images, where each image is associated with an aesthetic score on a scale ranging from 1 - 10. Three types of aesthetic prediction models are used to understand the mapping between human-understandable semantic features to the aesthetic score of every image. It is shown that the proposed idea of using tags for aesthetics prediction can produce comparative results close to the state-of-the-art methods and that is the main finding of this chapter.

The rest of the chapter is organized as follows. Section A.2 discusses the related work on this topic. Section A.3 provides the details of the used dataset and the proposed prediction models. Section A.4 presents the experimental results of image aesthetics prediction and the conclusion of this chapter is provided in Section A.5.

A.2 Related Works

This section talks about the related works where the problem of image aesthetics analysis has been formulated as a classification or a regression problem. The hand-designed features and generic features proposed over the years to solve this problem are reviewed. The recent studies showing that convolutional neural networks can be successfully applied for image aesthetics prediction achieving state-of-the-art results are also discussed.

The approaches that formulate aesthetic quality assessment as a classification problem distinguish aesthetically pleasing and displeasing images. The hand-designed features that were proposed in the literature for appropriate representation of image aesthetic characteristics are as follows: Datta et al. [46] designed 56 visual features for each image based on intuition. Apart from considering visual cues like colorfulness, brightness, saturation, hue, etc. they also considered features related to wavelet-based texture, size and aspect ratio, shape convexity, low depth of field, etc, and trained a statistical model to automatically classify images of having good or bad aesthetic quality. Ke et al. [89] used high-level semantic features to describe the spatial distribution of the high-frequency edges, color distribution, hue count, blur, etc. for the classification task. Dhar et al. [51] proposed a different type of human-perceived high-level image attributes related to image configuration, the content of the image, and the natural lighting conditions of the image, to predict image aesthetics and interestingness of the image. Nishiyama et al. [132] proposed an approach based on color harmony and bags of color patterns to deal with the complex color distribution of an image. They combined the color harmony feature along with blur, edges, and saliency features of the photos to improve their aesthetic classification performance. An alternative approach of using generic image descriptors such as GIST, Bag-of-Visual-Words descriptor, Fisher vector was proposed by Marchesotti et al. [118] which being able to implicitly encode the aesthetic characteristics of an image from SIFT information [113], could outperform traditional hand-designed features. Despite the success of the prior works using handcrafted features and generic features, in recent years several deep convolutional neural networks have been proposed for image aesthetics prediction. Lu et al. [114] proposed a double-column CNN to improve aesthetic categorization using style attributes and semantic attributes. Two heterogeneous inputs, i.e., global and local views of an image, were fed to both the columns of the double-column deep CNN, to capture both global and local characteristics of images.

On the other hand, the approaches that formulate aesthetic quality assessment as a regression problem focuses on finding the aesthetic scores using several data-driven machine learning techniques. Datta et al. [46] predicted the numerical aesthetics ratings by using Linear Regression (LR) on polynomial terms of the features. Bhattacharya et al. [29] used an interactive application, based on user-guided object segmentation and inpainting for extracting aesthetic features subsequently used for training a Support Vector Regression (SVR) model. Wu et al. [30] designed a new regression algorithm called support vector distribution regression (SVDR) and two separate learning strategies (RSL and LR) to tackle the difficulties in learning a visual quality distribution prediction model. Leveraging the success of CNN architectures, Kao et al. [19] proposed a regression model based on CNN and showed impressive results. Later, Jin et al. [82] came up with a CNN-based histogram prediction model that not only predicts the aesthetic score but also can obtain an aesthetically pleasing crop of an input image using the same regression model. Most recently, Murray et

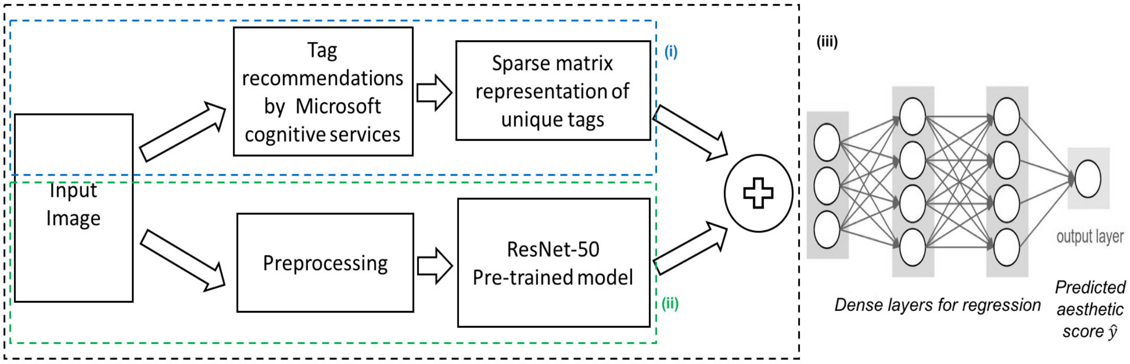

Tag recommendations by Microsoft cognitive services
Sparse matrix representation of unique tags
(i)
Input Image
Preprocessing
ResNet-50 Pre-trained model
(ii)
(iii)
output layer
Dense layers for regression
Predicted aesthetic score $\hat{y}$

Figure A-2: Proposed architecture

al. [127] showed that using only one deep CNN model (trained only for the distribution prediction task), three different kinds of tasks namely aesthetic quality classification, aesthetic score regression, and aesthetic score distribution prediction, can be solved.

However, none of the previous works focused on what is inside the photos. This work proposes a novel approach to predicting image aesthetic scores using machine-generated tags based on the content of the image. This research is the first trial that uses object detection in aesthetics analysis tasks. Surprisingly, only machine-generated tags can achieve comparable prediction performance to the state-of-the-art results.

A.3 Proposed Method

In this section, the details of the proposed idea for mapping raw RGB images to aesthetics tags are explained. First, the AVA dataset [126] and how the labels are extracted from the data are described. The details of the different models used for the determination of the aesthetic scores are also described.

Let us denote each image by $\mathbf{X}_i$ and the corresponding score by s_i. The score is usually in the range of 0 to 10. Let us assume that the tags obtained from all the images in the dataset belong to a set $\mathcal{T}$. A generator of tag from images $g_{tag} : \mathbb{R}^{M\times N}$ $\mapsto \mathcal{T}$. The predicted score for each image $\mathbf{X}_i$ is given by y_i. The indicator function $\mathbf{1}(x)$ produces 1 if x is true else 0.

A.3.1 Dataset

The large-scale image database for Aesthetic Visual Analysis known as AVA dataset [126] containing more than 2,55,000 images and covering a wide variety of subjects on 1,447 challenges is used in this task. These images along with a rich variety of meta-data are collected from an online community of photography amateurs such as `www.dpchallenge.com`. A total of 255,494 images are used for the experimental purposes. For each image $\mathbf{X}_i$, a distribution of user votes ranging from score 1 to 10 is provided. The weighted average for these score distributions with votes as the weights is computed to obtain a single real values score depicting the aesthetic rating of the image.

A.3.2 Prediction Model

In this chapter, the main focus is to illustrate the dependence of human-understandable semantic features to the aesthetics of each image. More specifically, the aim is to find the relationship between machine-generated tags and the aesthetic score of the image. The Microsoft Azure Cognitive Services framework is used to generate tags of each image. These tags are used for building regressor models for the aesthetic score. Here, mainly illustrate three kinds of models are illustrated: (i) naive bayes model using only tag information, (ii) CNN based score regression, and (iii) combination of CNN and sparse tag information vector for score prediction. The details of the architectures are shown in Figure A-2.

A.3.2.1 Naive Bayes using Only Tags

In this approach, only the machine-generated tags are considered to find their effect on the aesthetic scores. Here, the naive Bayes regressor is employed, where for each tag, the average aesthetic scores of all the images having that tag is computed. The mathematical formulation of the same is given as follows

$$p(y_i|t_1, t_2, ..., t_{n_i}) = p(y) \prod_{j=1}^{n_i} p(t_j|y_i), \tag{A.1}$$

Let us assume an uninformative uniform prior to the aesthetic score distribution, and two-sided exponential distribution,

$$p(t_j|y_i) = \frac{\lambda}{2}\exp\left(-\lambda|y_i - \mu_j|\right),$$

where $\mu_j = \frac{\sum_{i=1}^{N} s_i \mathbf{1}(t_j \in g_{tag}(\mathbf{X}_i)}{\sum_{i=1}^{N} \mathbf{1}(t_j \in g_{tag}(\mathbf{X}_i)}$ is the mean score for each tag across all images and λ is the inverse mean of the distribution. With the above assumption of two-sided exponential distribution, the mode of the aesthetic score distribution is obtained as follows

$$y^* = \frac{1}{n_i}\sum_{i=1}^{n_i} \mu_j. \tag{A.2}$$

This gives a closed-form solution to the aesthetic score prediction model from tag information and there is no iterative learning procedure involved in this method. While being very simple and straight-forward modeling, in the experimental section, it is shown that, it does a very good job of modeling the relationship between high-level human-understandable tag information and aesthetic score, comparable to certain early methods in aesthetics prediction.

A.3.2.2 Convolutional Neural Network

It has been shown that residual networks [65] can produce state-of-the-art results in image classification tasks and the intermediate feature representations from the learned parameters of Residual Network (ResNet) can extract meaningful semantic knowledge about the content of the images. Thus these features have been used for various kinds of end-to-end image-based tasks such as segmentation and depth estimation from a single image and other high-level computer vision tasks. These semantically meaningful features are leveraged to extract aesthetic scores from images. Although image aesthetics analysis is a highly subjective matter, I believe that humans are aesthetically inclined to certain features in images whereas some other features decrease the attractiveness of an image. Given this, the aesthetic score prediction problem reduces to correlating features that improve or decrease the aesthetic score of an image. To achieve the above mapping between semantically meaningful features and aesthetic score a multi-layer perceptron with a single output depicting the real-valued score is deployed. Similar to recent works in deep models a large number of image data available from the AVA dataset [126] is leveraged to learn generalizable features for the above mapping. This is obtained by simply minimizing the mean square error between the feed-forward prediction of the network and the ground-truth score while back-propagating the error gradients through the network and weight update using ADAM optimization method [94]. This method is similar to prior works but here ResNet [65] features are used, which are more expressive and can lead to a better understanding of raw RGB to real-valued aesthetic scores. Although only high-level fully connected features are used, I believe there is room for improvement concatenating low-level, mid-level, and high-level ResNet features similar to prior work; where low-level features of the image such as dots, lines, edges, gradients, pixel intensities, or colors; mid-level features such as color histograms, texture or shape descriptors; and high-level features such as features related to shapes and objects of the image are considered. For experimental purposes, each image in the AVA dataset [126] is resized to $224 \times 224 \times 3$ so that it matches the Imagenet [49] dimensions and this stage is referred to as the pre-processing stage. Then, the ResNet-50 architecture pre-trained on Imagenet dataset [49] is used to extract the high-level features obtained from fully connected layers.

A.3.3 Visualizing Dependence of Tags on Aesthetic Score

A.3.3.1 CNN using Tag Spare Vectors

In this approach, both the features (i) extracted by CNN and (ii) a sparse feature vector obtained from the tag generator using Microsoft Azure Cognitive Services are concatenated. Although prior work has shown that a combination of low-level, mid-level, and high-level features results in the good performance of aesthetic score prediction, neither of these features were human-understandable. In this method, the latent feature vectors from deep models are combined with object tag features to in-

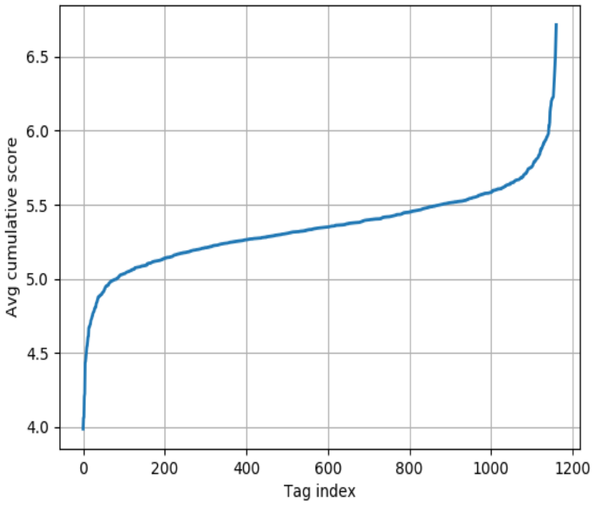


Figure A-3: Relationship of tag index with average cumulative aesthetic scores.

spect how human-understandable features such as object tags influence the aesthetics of an image. This is important because it gives a hint of interpretability as to why a certain image is aesthetically more appealing to the subjective human brain.

To achieve this the ResNet-features are fused in two ways: (i) early fusion, where the sparse feature vectors obtained from the one-hot representation of tags in a particular image is directly fused with real-valued ResNet features of size 2,048, (ii) late fusion, where first an intermediate real-valued representation from the one-hot representation is found and then it is fused with the ResNet features.

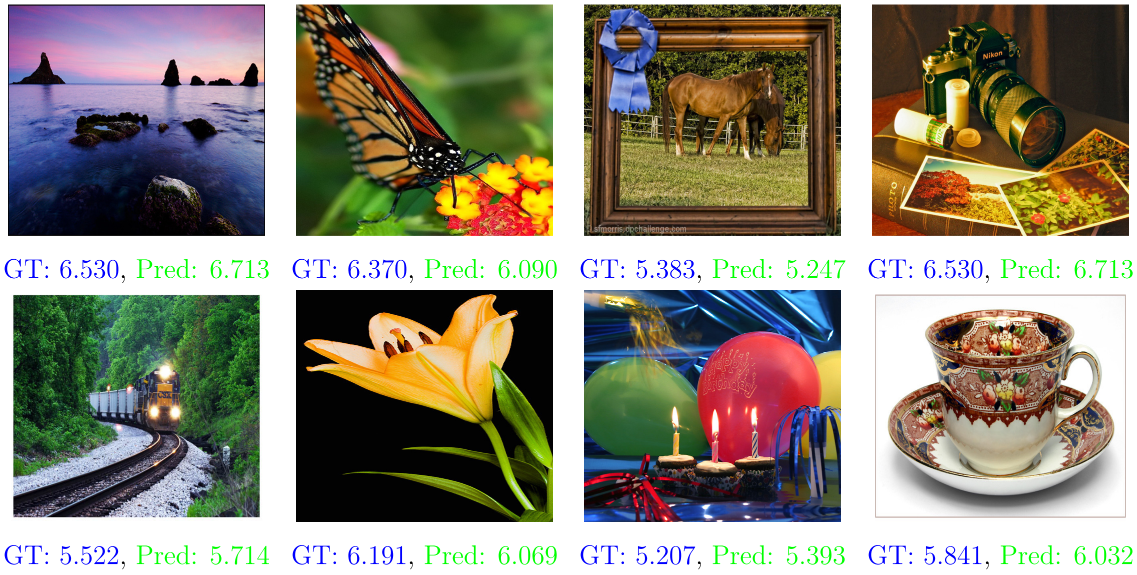
Figure A-3 shows the relationship between the tag index with the average cumulative score, where it can be seen that in the middle region, the gradient is low and the randomness is greater. Therefore these tags affect the average cumulative score very little. However, on both sides, the gradient is quite high and the randomness is much lower, which means that the tags in those two regions affect the average aesthetic score in a great fashion.

Table A.1: Mean Square Error for different methods

	MSE
GIST linear-SVR	0.5222
GIST rbf-SVR	0.5307
BoVW SIFT linear-SVR	0.5401
BoVW SIFT rbf-SVR	0.5513
Kao et al. 2015 [85]	0.4510
Jin et al. 2016 [82]	0.337
Murray et al. 2017 [127]	0.279
(i) Naive Bayes using only tags	0.512
(ii) CNN based score regression	0.3569
(iii) Combination of CNN and sparse tag information vector	**0.3562**

A.4 Experimental Results

In this section, the experimental results are reported as obtained from the proposed three architectures as explained above. Table A.1 shows the mean square errors for each of the methods. The top four rows in Table 1 report the results obtained by combining the generic image descriptors, GIST [136], SIFT [113] and Bag-of-Visual-Words (BoVW) [42], along with the linear and non-linear Support Vector Regression (SVR) [162]. Note that these results are reported directly from the paper. Details of these methods can be found in [118, 85]. The last three rows show the result achieved by the proposed architectures. Overall it shows that the proposed method of aesthetic score prediction using CNN and using the combination architecture performs almost comparable with the previous methods reported in the literature. Figure A-4 shows the qualitative results and the correlation between the ground truth scores (GT) and the aesthetic scores predicted (pred) by the proposed model (iii). It can be seen that the model can predict aesthetic scores close to the ground-truth scores. Some failure cases (i.e. the images for which the proposed method failed to predict the aesthetic scores correctly) are shown in Figure A-5. The ground truth scores (GT) and predicted scores (pred) are mentioned for each of the images.


GT: 6.530, Pred: 6.713 GT: 6.370, Pred: 6.090 GT: 5.383, Pred: 5.247 GT: 6.530, Pred: 6.713

GT: 5.522, Pred: 5.714 GT: 6.191, Pred: 6.069 GT: 5.207, Pred: 5.393 GT: 5.841, Pred: 6.032

Figure A-4: The ground truth scores (GT) and predicted scores (Pred) for some images in the test set. The GT scores are shown in blue. The predicted scores are shown in green.

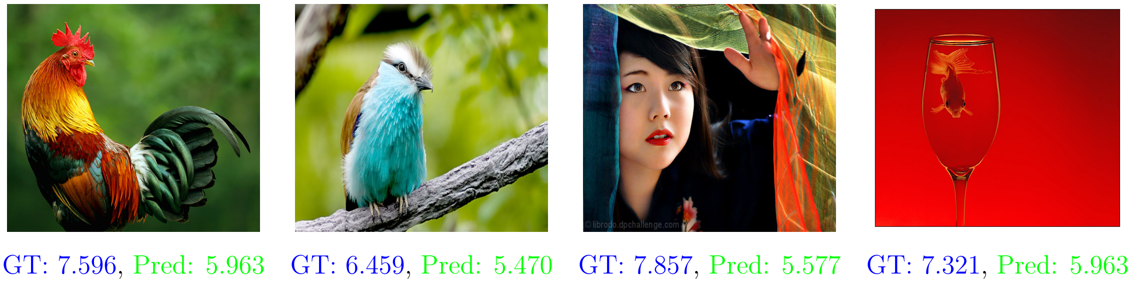

GT: 7.596, Pred: 5.963 GT: 6.459, Pred: 5.470 GT: 7.857, Pred: 5.577 GT: 7.321, Pred: 5.963

Figure A-5: Failure cases of the proposed aesthetic prediction model.

A.5 Conclusions

This chapter presented a novel idea of using tags for image aesthetics prediction. This problem is formulated as a regression problem with three kinds of models: (i) naive bayes model using only tag information, (ii) CNN based aesthetic score regression, and (iii) a combinational architecture of CNN and sparse tag information vector for aesthetic score prediction. Experimental results on the AVA dataset, which is the benchmark dataset with the rich aesthetic rating, showed that the proposed idea performed comparably with the state-of-the-art results. Although it could not outperform the previously reported results on the same dataset, it is worth mentioning that, this is the very first attempt of using inferential information to predict the aesthetic score. Moreover, it is shown that even a simple tag-based naive regressor could produce quite good results for aesthetic score prediction. I believe that using a combination of low-level deep features combined with inferential information like tags can be used to produce state-of-the-art systems that can be used to increase the number of views/likes on Social Networking Sites.

Publications

Publications Related to This Thesis

International Journal

[1] Roy H., Chaudhury S., Yamasaki T., Hashimoto T. “Toward Better Planetary Surface Exploration by Orbital Imagery Inpainting,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, `https://ieeexplore.ieee.org/document/9261928`.

[2] Roy H., Chaudhury S., Yamasaki T., Hashimoto T. “Image inpainting using frequency domain priors,” *SPIE Journal of Electronic Imaging*, (under review, submitted on 11-Sept-2020).

International Conference

[3] Ono M., Rothrock B., Otsu K., Higa S., Iwashita Y., Didier A., Islam T., Laporte C., Sun V., Stack K., Sawoniewicz J., Daftry S., Timmaraju V., Sahnoune S., Mattmann C., Lamarre O., Ghosh S., Qiu D., Nomura S., Roy H., “MAARS: Machine Learning-Based Analytics for Rover Systems,” *IEEE Aerospace conference 2020*, `https://ieeexplore.ieee.org/document/9172271`.

[4] Roy H., Chaudhury S., Yamasaki T., DeLatte D.M., Ohtake M., Hashimoto T., “Lunar surface image restoration using U-Net based deep neural networks,” *50th Lunar and Planetary Science Conference 2019*, `https://www.hou.usra.edu/meetings/lpsc2019/pdf/2656.pdf`.

[5] Roy H., Yamasaki T., Hashimoto T., “Do hashtags help? – Image aesthetics prediction using only hashtags,” *Women in Computer Vision Workshop (WICV) in conjunction with CVPR 2018*, Salt Lake City, USA (Workshop paper).

[6] Roy H., Yamasaki T., Hashimoto T., “Predicting Image Aesthetics using Objects in the Scene,” *International Joint Workshop on Multimedia Artworks Analysis and Attractiveness Computing in Multimedia (MMArt and ACM) in conjunction with ICMR, June 2018*, Yokohama, Japan, `https://dl.acm.org/doi/10.1145/3209693.3209698`.

Domestic Conference

[7] Roy H., Yamasaki T., Hashimoto T., “Predicting image aesthetics using objects in the scene,” *Meeting on Image Recognition and Understanding MIRU 2018.*

[8] Roy H., Yamasaki T., Hashimoto T., “Retrieving Interesting Planetary Images based on Captions,” PRMU 2021.

Book Chapter

[9] Roy H., Chaudhury S., Yamasaki T., Hashimoto T., “Chapter 10: Enhancing Spatial Resolution of Remotely Sensed Imagery Using Deep Learning and/or Data Restoration,” *Machine Learning for Planetary Science, 1st Edition*, to be published by *Elsevier Science and Technology Books* on 1st March 2021, `https://www.elsevier.com/books/machine-learning-for-planetary-science/helbert/978-0-12-818721-0`

Publications non-related to the thesis

International Journal

[10] Chaudhury S., Roy H., Mishra S., Yamasaki T., "Adversarial Training Time Attack against Discriminative and Generative Convolutional Models," *IEEE Access* (under review).

[11] Verspieren Q., Coral G., Pyne B., Roy H., "An Early History of the Philippine Space Development program," *Acta Astronautica*, Volume 151, October 2018, Pages 919- 927, `https://doi.org/10.1016/j.actaastro.2018.06.043`.

References

[1] https://pds-imaging.jpl.nasa.gov/volumes/mro/release54.html.

[2] https://solarsystem.nasa.gov/missions/.

[3] https://sci.esa.int/web/juice.

[4] http://mmx.isas.jaxa.jp/en/index.html.

[5] https://www.nasa.gov/dragonfly/dragonfly-overview/index.html.

[6] http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf.

[7] https://mars.nasa.gov/msl/spacecraft/rover/brains/.

[8] https://pds.nasa.gov/.

[9] https://www.mars.asu.edu/data/thm_dir/.

[10] http://wms.lroc.asu.edu/lroc/thumbnails.

[11] https://zenodo.org/record/2373798#.X8E4YZMza3I.

[12] https://zenodo.org/record/2552814#.X8EvcZMza3I.

[13] https://en.wikipedia.org/wiki/Mars_Reconnaissance_Orbiter.

[14] https://mars.nasa.gov/mars2020/spacecraft/rover/brains/.

[15] https://pds-imaging.jpl.nasa.gov/volumes/msl.html.

[16] https://github.com/tzutalin/labelImg.

[17] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 221–231, 2019.

[18] Mohammad Akbari, Jie Liang, and Jingning Han. Dsslic: Deep semantic segmentation-based layered image compression. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2042–2046. IEEE, 2019.

[19] Ethem Alpaydin. *Introduction to machine learning.* MIT press, 2020.

[20] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[21] Bob Balaram, Timothy Canham, Courtney Duncan, Håvard F Grip, Wayne Johnson, Justin Maki, Amelia Quon, Ryan Stern, and David Zhu. Mars helicopter technology demonstrator. In *2018 AIAA Atmospheric Flight Mechanics Conference*, page 0023, 2018.

[22] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016.

[23] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. *arXiv preprint arXiv:1802.01436*, 2018.

[24] Coloma Ballester, Marcelo Bertalmio, Vicent Caselles, Guillermo Sapiro, and Joan Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE transactions on image processing*, 10(8):1200–1211, 2001.

[25] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, 28(3):24, 2009.

[26] KJ Becker, MS Robinson, TL Becker, LA Weller, S Turner, L Nguyen, C Selby, BW Denevi, SL Murchie, RL McNutt, et al. Near global mosaic of mercury. *AGUFM*, 2009:P21A–1189, 2009.

[27] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2000.

[28] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, 12(8):882–889, 2003.

[29] Subhabrata Bhattacharya, Rahul Sukthankar, and Mubarak Shah. A framework for photo-quality assessment and enhancement based on visual aesthetics. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pages 271–280, New York, NY, USA, 2010. ACM.

[30] Simone Bianco, Luigi Celona, Paolo Napoletano, and Raimondo Schettini. Predicting image aesthetics with deep learning. In Jacques Blanc-Talon, Cosimo Distante, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, pages 117–125, Cham, 2016. Springer International Publishing.

[31] Timothy J Buschman and Earl K Miller. Top-down versus bottom-up control of attention in the prefrontal and posterior parietal cortices. *science*, 315(5820):1860–1862, 2007.

[32] T Chan. Local inpainting models and tv inpainting. *SIAM J. Appl. Math.*, 62(3):1019–1043, 2001.

[33] Subhajit Chaudhury and Hiya Roy. Can fully convolutional networks perform well for general image restoration problems? In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 254–257. IEEE, 2017.

[34] Bin Chen, Bo Huang, Lifan Chen, and Bing Xu. Spatially and temporally weighted regression: A novel method to produce continuous cloud-free landsat imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 55(1):27–37, 2016.

[35] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv preprint arXiv:1412.7062*, 2014.

[36] Qing Cheng, Huanfeng Shen, Liangpei Zhang, and Pingxiang Li. Inpainting for remotely sensed images with a multichannel nonlocal total variation model. *IEEE Transactions on Geoscience and Remote Sensing*, 52(1):175–187, 2013.

[37] Qing Cheng, Qiangqiang Yuan, Michael Kwok-Po Ng, Huanfeng Shen, and Liangpei Zhang. Missing data reconstruction for remote sensing images with weighted low-rank tensor model. *IEEE Access*, 7:142339–142352, 2019.

[38] Mircea Cimpoi, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3606–3613, 2014.

[39] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. *CoRR*, abs/1202.2745, 2012.

[40] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[41] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, 13(9):1200–1212, 2004.

[42] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[43] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. IEEE, 2005.

[44] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B Goldman, and Pradeep Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on graphics (TOG)*, 31(4):1–10, 2012.

[45] R. Datta, Jia Li, and J. Z. Wang. Algorithmic inferencing of aesthetics and emotion in natural images: An exposition. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 105–108, 2008.

[46] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z. Wang. Studying aesthetics in photographic images using a computational approach. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part III*, ECCV'06, pages 288–301, Berlin, Heidelberg, 2006. Springer-Verlag.

[47] Ritendra Datta, Jia Li, and James Z. Wang. Learning the consensus on visual quality for next-generation image management. In *Proceedings of the 15th ACM International Conference on Multimedia*, MM '07, pages 533–536, New York, NY, USA, 2007. ACM.

[48] Danielle M DeLatte, Sarah T Crites, Nicholas Guttenberg, Elizabeth J Tasker, and Takehisa Yairi. Segmentation convolutional neural networks for automatic crater detection on mars. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8):2944–2957, 2019.

[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[50] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[51] Sagnik Dhar, Vicente Ordonez, and Tamara L. Berg. *High level describable attributes for predicting aesthetics and interestingness*, pages 1657–1664. 2011.

[52] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? 2012.

[53] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.

[54] Richard Doyle, Raphael Some, Wesley Powell, Gabriel Mounce, Montgomery Goforth, Stephen Horan, and Michael Lowry. High performance spaceflight computing (hpsc) next-generation space processor (ngsp): A joint investment of nasa and afrl. In *Proceedings of the Workshop on Spacecraft Flight Software*, 2013.

[55] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. In *ACM SIGGRAPH 2003 Papers*, pages 303–312. 2003.

[56] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7), 2011.

[57] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 341–346, 2001.

[58] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[59] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[60] Lionel Gueguen, Alex Sergeev, Ben Kadlec, Rosanne Liu, and Jason Yosinski. Faster neural networks straight from jpeg. In *Advances in Neural Information Processing Systems*, pages 3933–3944, 2018.

[61] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777, 2017.

[62] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the royal statistical society. series c (applied statistics)*, 28(1):100–108, 1979.

[63] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[64] Kaiming He and Jian Sun. Image completion approaches using the statistics of similar patches. *IEEE transactions on pattern analysis and machine intelligence*, 36(12):2423–2435, 2014.

[65] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[66] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[67] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[68] Wei He, Naoto Yokoya, Longhao Yuan, and Qibin Zhao. Remote sensing image reconstruction using tensor ring completion and total variation. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11):8998–9009, 2019.

[69] Wei He, Hongyan Zhang, Liangpei Zhang, and Huanfeng Shen. Total-variation-regularized low-rank matrix factorization for hyperspectral image restoration. *IEEE transactions on geoscience and remote sensing*, 54(1):178–188, 2015.

[70] Geoffrey E Hinton. Deep belief networks. *Scholarpedia*, 4(5):5947, 2009.

[71] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

[72] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[73] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[74] Yan Huang, Wei Wang, and Liang Wang. Instance-aware image and sentence matching with selective multimodal lstm. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2310–2318, 2017.

[75] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[76] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, 36(4):1–14, 2017.

[77] William A Imbriale. *Large antennas of the deep space network*, volume 1. John Wiley & Sons, 2005.

[78] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[79] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[80] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *Proc. of the Int'l Conf. on Artificial Intelligence*, volume 56. Citeseer, 2000.

[81] Teng-Yu Ji, Naoto Yokoya, Xiao Xiang Zhu, and Ting-Zhu Huang. Nonlocal tensor completion for multitemporal remotely sensed images' inpainting. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3047–3061, 2018.

[82] B. Jin, M. V. O. Segovia, and S. Süsstrunk. Image aesthetic predictors based on weighted cnns. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2291–2295, Sept 2016.

[83] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[84] Ian Jolliffe. *Principal Component Analysis*, pages 1094–1096. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[85] Yueying Kao, Chong Wang, and Kaiqi Huang. Visual aesthetic quality assessment with a regression model. In *2015 IEEE International Conference on Image Processing, ICIP 2015, Quebec City, QC, Canada, September 27-30, 2015*, pages 1583–1587, 2015.

[86] Andrej Karpathy and Li Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137, 2015.

[87] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[88] Fumi Katsuki and Christos Constantinidis. Bottom-up and top-down attention: different processes and overlapping neural systems. *The Neuroscientist*, 20(5):509–521, 2014.

[89] Yan Ke, Xiaoou Tang, and Feng Jing. The design of high-level features for photo quality assessment. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR '06, pages 419–426, Washington, DC, USA, 2006. IEEE Computer Society.

[90] Hannah R Kerner, James F Bell III, and Heni Ben Amor. Context-dependent image quality assessment of jpeg compressed mars science laboratory mastcam images using convolutional neural networks. *Computers & Geosciences*, 118:109–121, 2018.

[91] Hannah R Kerner, Danika F Wellington, Kiri L Wagstaff, James F Bell, Chiman Kwan, and Heni Ben Amor. Novelty detection for multispectral images with application to planetary exploration. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 9484–9491, 2019.

[92] Hannah Rae Kerner, Kiri L Wagstaff, Brian D Bue, Patrick C Gray, James F Bell III, and Heni Ben Amor. Toward generalized change detection on planetary surfaces with convolutional autoencoders and transfer learning. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(10):3900–3918, 2019.

[93] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[94] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[95] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*, pages 523–534. Springer, 2014.

[96] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[97] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[98] E. Lakdawalla. *Where We Are*, 2019.

[99] Olivier Le Meur, Mounira Ebdelli, and Christine Guillemot. Hierarchical super-resolution-based inpainting. *IEEE transactions on image processing*, 22(10):3779–3790, 2013.

[100] Yann LeCun. Yoshua bengio, and geoffrey hinton. *Deep learning. nature*, 521(7553):436–444, 2015.

[101] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2:396–404, 1989.

[102] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[103] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[104] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[105] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *null*, page 305. IEEE, 2003.

[106] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3214–3223, 2018.

[107] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[108] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[109] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[110] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[111] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015.

[112] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[113] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110, November 2004.

[114] Xin Lu, Zhe L. Lin, Hailin Jin, Jianchao Yang, and James Z. Wang. Rating image aesthetics using deep learning. *IEEE Trans. Multimedia*, 17(11):2021–2034, 2015.

[115] Wei Luo, Xiaogang Wang, and Xiaoou Tang. Content-based photo quality assessment. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, *ICCV*, pages 2206–2213. IEEE Computer Society, 2011.

[116] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. icml*, volume 30, page 3, 2013.

[117] Julien Mairal, Michael Elad, and Guillermo Sapiro. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2007.

[118] Luca Marchesotti, Florent Perronnin, Diane Larlus, and Gabriela Csurka. Assessing the aesthetic quality of photographs using generic image descriptors. In Dimitris N. Metaxas, Long Quan, Alberto Sanfeliu, and Luc J. Van Gool, editors, *ICCV*, pages 1784–1791. IEEE Computer Society, 2011.

[119] Jonathan Masci, Ueli Meier, Dan Cireşan, and Jürgen Schmidhuber. Stacked convolutional auto-encoders for hierarchical feature extraction. In *International conference on artificial neural networks*, pages 52–59. Springer, 2011.

[120] Alfred S McEwen, Eric M Eliason, James W Bergstrom, Nathan T Bridges, Candice J Hansen, W Alan Delamere, John A Grant, Virginia C Gulick, Kenneth E Herkenhoff, Laszlo Keszthelyi, et al. Mars reconnaissance orbiter's high resolution imaging science experiment (hirise). *Journal of Geophysical Research: Planets*, 112(E5), 2007.

[121] Roi Mendez-Rial, María Calvino-Cancela, and Julio Martin-Herrero. Anisotropic inpainting of the hypercube. *IEEE Geoscience and Remote Sensing Letters*, 9(2):214–218, 2011.

[122] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

[123] TM Mitchell. Machine learning, mcgraw-hill higher education. *New York*, 1997.

[124] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[125] Kevin P Murphy et al. Naive bayes classifiers. *University of British Columbia*, 18:60, 2006.

[126] N. Murray, L. Marchesotti, and F. Perronnin. Ava: A large-scale database for aesthetic visual analysis. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2408–2415, June 2012.

[127] Naila Murray and Albert Gordo. A deep architecture for unified aesthetic prediction. *CoRR*, abs/1708.04890, 2017.

[128] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.

[129] Hyeonseob Nam, Jung-Woo Ha, and Jeonghee Kim. Dual attention networks for multimodal reasoning and matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 299–307, 2017.

[130] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

[131] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady an ussr*, volume 269, pages 543–547, 1983.

[132] M. Nishiyama, T. Okabe, I. Sato, and Y. Sato. Aesthetic quality classification of photographs based on color harmony. In *CVPR 2011*, pages 33–40, June 2011.

[133] Zhenxing Niu, Mo Zhou, Le Wang, Xinbo Gao, and Gang Hua. Hierarchical multimodal lstm for dense visual-semantic embedding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1881–1889, 2017.

[134] P. Obrador, L. Schmidt-Hackenberg, and N. Oliver. The role of image composition in image aesthetics. In *2010 IEEE International Conference on Image Processing*, pages 3185–3188, Sept 2010.

[135] Takahiro Ogawa and Miki Haseyama. Image inpainting based on sparse representations with a perceptual metric. *EURASIP Journal on Advances in Signal Processing*, 2013(1):179, 2013.

[136] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *Int. J. Comput. Vision*, 42(3):145–175, May 2001.

[137] M Ono, B Rothrock, C Mattmann, T Islam, A Didier, VZ Sun, D Qiu, P Ramirez, K Grimes, and G Hedrick. Make planetary images searchable: Content-based search for pds and on-board datasets. *LPI*, (2132):2552, 2019.

[138] Masahiro Ono, Brandon Rothrock, Kyohei Otsu, Shoya Higa, Yumi Iwashita, Annie Didier, Tanvir Islam, Christopher Laporte, Vivian Sun, Kathryn Stack, et al. Maars: Machine learning-based analytics for automated rover systems. In *2020 IEEE Aerospace Conference*, pages 1–17. IEEE, 2020.

[139] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Gaugan: semantic image synthesis with spatially adaptive normalization. In *ACM SIGGRAPH 2019 Real-Time Live!*, pages 1–1. 2019.

[140] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[141] RZ Povilaitis, MS Robinson, CH Van der Bogert, Harald Hiesinger, HM Meyer, and LR Ostrach. Crater density differences: Exploring regional resurfacing, secondary crater populations, and crater saturation equilibrium on the moon. *Planetary and Space Science*, 162:41–51, 2018.

[142] Dicong Qiu, Brandon Rothrock, Tanvir Islam, Annie K Didier, Vivian Z Sun, Chris A Mattmann, and Masahiro Ono. Scoti: Science captioning of terrain images for data prioritization and local image search. *Planetary and Space Science*, page 104943, 2020.

[143] J. Ross Quinlan. Simplifying decision trees. *International journal of man-machine studies*, 27(3):221–234, 1987.

[144] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[145] Preesan Rakwatin, Wataru Takeuchi, and Yoshifumi Yasuoka. Restoration of aqua modis band 6 using histogram matching and local least squares fitting. *IEEE Transactions on Geoscience and Remote Sensing*, 47(2):613–627, 2008.

[146] Antti Rasmus, Mathias Berglund, Mikko Honkala, Harri Valpola, and Tapani Raiko. Semi-supervised learning with ladder networks. In *Advances in neural information processing systems*, pages 3546–3554, 2015.

[147] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[148] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.

[149] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[150] Oren Rippel and Lubomir Bourdev. Real-time adaptive image compression. *arXiv preprint arXiv:1705.05823*, 2017.

[151] Stuart J Robbins. A new global database of lunar impact craters> 1–2 km: 1. crater locations and sizes, comparisons with published databases, and global analysis. *Journal of Geophysical Research: Planets*, 124(4):871–892, 2019.

[152] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[153] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 860–867. IEEE, 2005.

[154] Brandon Rothrock, Ryan Kennedy, Chris Cunningham, Jeremie Papon, Matthew Heverly, and Masahiro Ono. Spoc: Deep learning-based terrain classification for mars rover missions. In *AIAA SPACE 2016*, page 5539. 2016.

[155] H. Roy, S. Chaudhury, T. Yamasaki, and T. Hashimoto. Toward better planetary surface exploration by orbital imagery inpainting. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, pages 1–1, 2020.

[156] Hiya Roy, Subhajit Chaudhury, Toshihiko Yamasaki, Danielle DeLatte, Makiko Ohtake, and Tatsuaki Hashimoto. Lunar surface image restoration using u-net based deep neural networks. *arXiv preprint arXiv:1904.06683*, 2019.

[157] Jose San Pedro, Tom Yeh, and Nuria Oliver. Leveraging user comments for aesthetic aware image search reranking. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, pages 439–448, New York, NY, USA, 2012. ACM.

[158] Huanfeng Shen, Xinghua Li, Qing Cheng, Chao Zeng, Gang Yang, Huifang Li, and Liangpei Zhang. Missing information reconstruction of remote sensing data: A technical review. *IEEE Geoscience and Remote Sensing Magazine*, 3(3):61–85, 2015.

[159] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[160] F. Simond, N. Arvanitopoulos, and S. Süsstrunk. Image aesthetics depends on context. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3788–3792, Sept 2015.

[161] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[162] Alex J. Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, August 2004.

[163] Yale Song and Mohammad Soleymani. Polysemous visual-semantic embedding for cross-modal retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1979–1988, 2019.

[164] Yuhang Song, Chao Yang, Yeji Shen, Peng Wang, Qin Huang, and C-C Jay Kuo. Spg-net: Segmentation prediction and guidance network for image inpainting. *arXiv preprint arXiv:1805.03356*, 2018.

[165] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[166] Yoshisada Takizawa. Kaguya (selene) mission overview. In *Proceedings of the 26th International Symposium on Space Technology and Science (ISTS), June 2008*, 2008.

[167] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders. *arXiv preprint arXiv:1703.00395*, 2017.

[168] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5306–5314, 2017.

[169] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[170] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

[171] Kiri L Wagstaff, Gary Doran, Ashley Davies, Saadat Anwar, Srija Chakraborty, Marissa Cameron, Ingrid Daubar, and Cynthia Phillips. Enabling onboard detection of events of scientific interest for the europa clipper spacecraft. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2191–2201, 2019.

[172] Kiri L Wagstaff, You Lu, Alice Stanboli, Kevin Grimes, Thamme Gowda, and Jordan Padams. Deep mars: Cnn classification of mars imagery for the pds imaging atlas. In *AAAI*, pages 7867–7872, 2018.

[173] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[174] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[175] Ou Wu, Weiming Hu, and Jun Gao. Learning to predict the perceived visual quality of photos. In *2011 International Conference on Computer Vision*, pages 225–232, Nov 2011.

[176] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[177] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *Advances in neural information processing systems*, pages 341–349, 2012.

[178] Wei Xiong, Jiahui Yu, Zhe Lin, Jimei Yang, Xin Lu, Connelly Barnes, and Jiebo Luo. Foreground-aware image inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5840–5848, 2019.

[179] Kai Xu, Minghai Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. Learning in the frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1740–1749, 2020.

[180] Kai Xu, Zhikang Zhang, and Fengbo Ren. Lapran: A scalable laplacian pyramid reconstructive adversarial network for flexible compressive sensing reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 485–500, 2018.

[181] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057, 2015.

[182] Meng Xu, Xiuping Jia, Mark Pickering, and Sen Jia. Thin cloud removal from optical remote sensing images using the noise-adjusted principal components transform. *ISPRS Journal of Photogrammetry and Remote Sensing*, 149:215–225, 2019.

[183] Raymond A Yeh, Chen Chen, Teck Yian Lim, Alexander G Schwing, Mark Hasegawa-Johnson, and Minh N Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5485–5493, 2017.

[184] Chao Yu, Liangfu Chen, Lin Su, Meng Fan, and Shenshen Li. Kriging interpolation method and its application in retrieval of modis aerosol optical depth. In *2011 19th International Conference on Geoinformatics*, pages 1–6. IEEE, 2011.

[185] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5505–5514, 2018.

[186] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[187] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[188] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[189] Chao Zeng, Huanfeng Shen, and Liangpei Zhang. Recovering missing pixels for landsat etm+ slc-off imagery using multi-temporal regression analysis and a regularization method. *Remote Sensing of Environment*, 131:182–194, 2013.

[190] Chuanrong Zhang, Weidong Li, and David Travis. Gaps-fill of slc-off landsat etm+ satellite image using a geostatistical approach. *International Journal of Remote Sensing*, 28(22):5103–5122, 2007.

[191] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[192] Lei Zhang and Xiaolin Wu. An edge-guided image interpolation algorithm via directional filtering and data fusion. *IEEE transactions on Image Processing*, 15(8):2226–2238, 2006.

[193] Qiang Zhang, Qiangqiang Yuan, Jie Li, Zhiwei Li, Huanfeng Shen, and Liangpei Zhang. Thick cloud and cloud shadow removal in multitemporal imagery using progressively spatio-temporal patch group deep learning. *ISPRS Journal of Photogrammetry and Remote Sensing*, 162:148–160, 2020.

[194] Qiang Zhang, Qiangqiang Yuan, Chao Zeng, Xinghua Li, and Yancong Wei. Missing data reconstruction in remote sensing image with a unified spatial–temporal–spectral deep convolutional neural network. *IEEE Transactions on Geoscience and Remote Sensing*, 56(8):4274–4288, 2018.

[195] Qinpei Zhao, Ville Hautamaki, and Pasi Fränti. Knee point detection in bic for detecting the number of clusters. In *International conference on advanced concepts for intelligent vision systems*, pages 664–673. Springer, 2008.

[196] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.