博士論文

「Automatic Crater Detection using Segmentation Convolutional Neural Networks

(セグメンテーション畳み込みニューラルネットワークによるクレーター自動検出)」

Danielle Marguerite DeLatte

(ダニエル　マルゲリト　デラタ)

Advisor: Associate Professor Takehisa Yairi

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

PAPERS RELATED TO THESIS

JOURNAL PAPERS

- DeLatte, D.M. et al., *In review (accepted with major revision)*. Segmentation Using Convolutional Neural Networks for Automatic Crater Detection on Mars: Experiments and Analysis.
- DeLatte, D.M. et al., *In review*. Automated Crater Detection Algorithms from a Machine Learning Perspective in the Convolutional Neural Network Era.

CONFERENCE PAPERS

- DeLatte, D.M. et al., 2018. Experiments in Segmenting Mars Craters Using Convolutional Neural Networks. In i-SAIRAS. Madrid, pp. 1–8.

POSTERS

- DeLatte, D.M. et al, 2017. Space Applications of Machine Learning: Crater Counting. In: SELENE Symposium, Tokyo.
- DeLatte, D.M. et al., 2018. Exploration of Machine Learning Methods for Crater Counting on Mars. In: 49th Lunar and Planetary Science Conference, Houston. pp. 1–2.

ACKNOWLEDGEMENTS

No PhD thesis exists in a vacuum, and this one is no exception. I want to thank my advisor, Prof. Takehisa Yairi, and committee members, Prof. Koichi Hori, Prof. Shinichi Nakasuka, Prof. Akira Iwasaki, and Prof. Hideaki Miyamoto, for their support, comments, constructive criticism, and feedback throughout this process. I also thank my collaborators, Prof. Elizabeth Tasker, Dr. Sarah Crites, and Dr. Nicholas Guttenberg. The opportunity to work on an interdisciplinary project bridging machine learning and planetary geology with the three of you was an incredible experience. In particular Prof. Tasker's research guidance and laser focus on keeping to thesis-related activities, Dr. Crites' knowledge of the crater counting community and their tools, and Dr. Guttenberg's deep learning expertise made this a valuable collaboration.

Thank you also to lab mates, research collaborators, and friends at the University of Tokyo who made this process more fulfilling and enjoyable. Particular gratitude goes to my fellow Students for the Exploration and Development of Space (SEDS) board and members for embarking on the adventure of starting a new club with me. In particular (alphabetically by first name): An Zerbli, Caitlin Devor, Kana Ishimaru, Kentaro Abe, Dr. Lana Sinapayen, Dr. Liew Chun Fui, Lisa Hartwig, Mario Rodriguez, Matthew Richardson, Menghe Xu, Miki Takahiro, Dr. Naoya Takeishi, Petr Khrapchenkov, Ryo Sakagami, Ryodo Hemmi, Dr. Samir Khan, Dr. Sara Badr, Sara Ali, Tri-Thien Che, and Dr. Yan Jiang. Thank you to the wonderful ladies of the International Multidisciplinary Engineering office for all the support you give to the program's students: Ms. Yukimi Umeda, Ms. Rika Sakai, and Ms. Yoshie Minegishi. Many friends abroad supported remotely and a few even visited: Amy L., Anjaney K., Christina Q., Danielle A., Elizabeth O., George H., George S., Jeff O., Jennifer M., Jillian J., Kim J., Kriti G., Layla M., Lori S., Max B., Matt R., Michael L., Michelle E., Nora M., Reka K., Rosalind C., Samantha M., Sara C., Shyam S., and Yukiko N.

One of my greatest joys in Tokyo was developing and hosting the monthly interdisciplinary space speaker series, Space Cafe Tokyo. Sincere thanks and appreciation to all our speakers, audience members, and volunteers. You all truly made my time in Tokyo special. Thank you especially to the venue owners, Paul and Hisako Davies. Thank you also to Dr. Amanda Alvarez (Nerd Nite Tokyo), Suzana Ilic (Machine Learning Tokyo), Dr. Ayumi Koso (NIHU), Rei Kawashima (UNISEC) and Dr. Walid Yassin (UTokyo Inspire) for making the Tokyo technology, science communication, and event hosting community a welcoming and collaborative one.

During the course of the PhD, I had opportunities to work with and attend several special workshops around the world. Space Generation Advisory Council Fusion Forum, Caltech Space Challenge, Beihang University of Aeronautics and Astronautics International Graduate Summer School, and JAXA Y-ISEF students, mentors, volunteers, and staff greatly enriched my experience these last several years.

Professional and academic mentors from around the world have aided my journey over the past decade. Since this is probably my best chance to put this appreciation in print, thank you: Prof. Amos Winter, Dr. Ashitey Trebi-Ollennu, Becky Blackmon, Ben Reed, Brian Roberts, Prof. Dava Newman, Prof. David Darmofal, Prof. David Miller, Dr. Dino

Rossetti, Ed Rezac, Dr. Erika Brown Wagner, Frank Cepollina, Prof. Hideto Suzuki, Prof. Hugh Hill, Kathy Laurini, Prof. Kazuya Yoshida, Kelvin Garcia, Dr. Leon Alkalai, Dr. Nathan Britton, Dr. Vicki Balzano.

Last but never least, thank you to the best family a lady could ask for. My parents, David and Suzanne, have shown all four of their children unending support and patience whether with driving us to every activity imaginable or waking up for late night/early morning discussions. To my siblings — Caitlin, Joseph, and Michael — I'm so proud of you and all you have accomplished. You have so much potential, and it has been a pleasure watching you grow into it. Thank you all for supporting me as I pursue my goals. To my husband, Tommy: this was not easy, and you had a front row seat. Thank you for first draft edits, macro magic, and walking this and so many other journeys with me.

And to all those whom I have unfortunately left off (to be remembered immediately after submitting this, I'm sure) and to those who find themselves reading this: Thank you.

EXECUTIVE SUMMARY

Convolutional Neural Networks (CNN) offer promising opportunities to automatically glean scientifically relevant information directly from annotated images, without needing to hand craft features for detection. Crater counting started with hand counting hundreds, thousands, or even millions of craters in order to determine the age of geological units on planetary bodies of the solar system. Automated crater detection algorithms have attempted to speed up this process. Previous research has employed computer vision techniques with hand crafted features such as light and shadow patterns, circle finding, or edge detection. This research continues, but now some researchers use techniques like convolutional neural networks that enable the algorithm to develop its own features. As the field of machine learning undergoes exponential growth in terms of paper count and research methods, the crater counting application can benefit from the new research, especially when conducting joint interdisciplinary projects. Despite these advancements, the crater counting community has not yet adopted standard methods for automating the process despite decades of research. This research enumerates challenges for both planetary geologists and machine learning researchers, looks at the recent automatic crater detection advancements using machine learning techniques (primarily in methods using CNNs), and makes recommendations for the path toward greater automation.

Machine learning segmentation techniques show great promise for automating crater counting. Developing effective segmentation neural networks for this task involves multiple design choices in the network architecture and training set preparation. This research evaluates two target types, measures the impact of hyperparameters (kernel size, filters), and varies the amount of data used to train the models from using 3 to 15 of the 24 tiles. (Each tile is 30º by 30º and is within ±30º latitude.) The algorithm is trained using annotations of 2-32 km radius Martian craters and THEMIS Daytime Infrared (IR) Global Mosaic tiles. Pixel-based machine learning metrics like loss and accuracy are used during training and validation. In addition, crater count metrics such as recall (match ratio), precision, and F1 score are used to evaluate the performance and for model selection. The results innumerate how incorporating machine learning into the crater counting process is beneficial to planetary geologists: for example, by creating a list of craters in a region or suggesting potential degraded craters for further analysis. A segmentation network using convolutional neural networks is successfully implemented to find 65-76% of craters in common with a human annotated dataset, which is shown to obtain comparable ages of nine diverse regions.

CHAPTER 1: INTRODUCTION

The use of machine learning techniques in space science and engineering has grown in popularity and impact over the past several years. Scientists recognize the value in automating time-consuming tasks and identifying patterns in high dimensional data sets. Existing research marrying space and machine learning includes: prioritizing asteroid deflector technology (Nesvold et al., 2018), transient exoplanet discovery (Shallue & Vanderburg, 2018), hyperspectral image classification (Mou, Ghamisi, & Zhu, 2017), and classification of Cassini images (Stanboli, Bue, Wagstaff, & Altinok, 2017). When quality data is available and problems are well defined, there is potential for optimization and new discoveries. Shallue et al. found two new exoplanets using machine learning techniques that helped sift between transiting exoplanets and false positives (Shallue & Vanderburg, 2018). Stanboli et al. used a classification convolutional neural network to create a search tool to help organize the 800,000 images taken by the Cassini spacecraft by contents (rings, craters, etc.) (Stanboli et al., 2017). Another problem worthy of optimization is crater identification: the identification of craters caused by impacts or other erosion on the surface of a planetary body and the measurement of their size. Once crater locations and sizes are identified, the results can be used for age dating, hazard avoidance, or positioning.

Planetary geologists use crater counting to age-date regions of planetary bodies (Crater Analysis Techniques Working Group, 1979) (Ivanov, 2001) (Hartmann & Neukum, 2001) (Neukum, Ivanov, & Hartmann, 2001). The surface age is determined by counting the number of craters of various sizes in a region and comparing those counts to expected accumulation from a known production function based on an expected meteorite impact rate. Regions of interest could include a lava flow, a giant crater, or a dry river bed; interesting regions can be anywhere there is evidence of a common starting point from which craters have accumulated. More craters, and particularly more large craters, indicate older regions.

However, the count alone does not give an age directly. For Earth's Moon, radiometric dating of the lunar samples returned during the Apollo missions anchor the chronologies to absolute ages (Crater Analysis Techniques Working Group, 1979). For other planetary bodies, relative ages can be determined and assumptions are needed to make predictions of absolute ages. As samples are returned from other bodies, those relative ages will yield absolute ages. Traditionally, the crater counting task is done by citizen scientists, graduate students, and experts who label – by hand – the characteristics of hundreds of thousands of craters.

With the publication of large crater counting datasets and increased accessibility of machine learning methods and computational hardware, this historically tedious task can be automated. One challenge is that craters overlap and vary in size, depth, and visibility. One way to untangle this complex terrain is to use neural networks. Unlike other automated techniques which require a set of human-designed features (S. Liu, Ding, Gao, & Stepinski, 2012; Urbach & Stepinski, 2009), neural networks find their own, and complex pattern recognition criteria can be developed. A Convolutional Neural Network (CNN) (LeCun, Bengio, & Haffner, 1998) is a common machine learning technique for analyzing data and is especially useful for processing images (LeCun, Bengio, & Hinton, 2015) in tasks from handwritten digit recognition (Denker et al., 1989) to scene segmentation and human pose estimation (He, Gkioxari, Dollár, & Girshick, 2017). They are particularly successful in situations with large multidimensional datasets. For example, Maggiori et al. used CNNs to segment remote sensing images of Earth (Maggiori, Tarabalka, Charpiat, & Alliez, 2017), and Long et al. used CNNs to locate objects in remote sensing images (Long, Gong, Xiao, & Liu, 2017). In this study, the use of a segmentation convolutional neural network to identify craters from Mars satellite imagery is explored.

1.1 RELATED WORKS

Crater counting has developed over 40 years and anchors on the idea that understanding the distribution of craters by size and location gives insights into the relative age of regions. The founding techniques were described by the Crater Analysis Techniques Working Group in 1979 (Crater Analysis Techniques Working Group, 1979) and scientists have modified and extended the method since (Robbins et al., 2018). For example, further studies of the relationship of the crater ages on the moon linked the lunar absolute ages to relative ages on Mars (Ivanov, 2001) (Hartmann & Neukum, 2001) and other inner solar system bodies (Neukum et al., 2001). Insights from crater counting with these techniques led to the creation of a new global geological map of Mars and dating the different large craters or geological units (lowland, volcanic, impact, etc.) (Tanaka, 2014) (Tanaka et al., 2014) via crater counting techniques (Platz, Michael, Tanaka, Skinner, & Fortezzo, 2013). Regions that make up a geological unit are often unique shapes (many examples in Platz et al. (Platz et al., 2013)). Other authors have developed analysis tools for the crater counting community (Michael & Neukum, 2010) (Michael, 2008), investigated topics such as identification of crater clustering (Michael, Platz, Kneissl, & Schmedemann, 2012), studied the minimum effective area for crater counting with high resolution data (Warner et al., 2015), explored the reproducibility of crater counts by experts (Robbins et al., 2014), and determined the potential for citizen scientists to contribute to crater identification (Bugiolacchi et al., 2016).

Although researchers can glean insights by making direct observations while counting craters and general familiarity with the surface of the target celestial body is valuable, counting craters is a repetitive task that could be automated. While many methods for automated crater counting have been proposed (Cohen, 2016) (Yamamoto et al., 2017) (Salih et al., 2016), none of these methods has been definitive or adopted by the community to replace hand labelling data. Some techniques (Cohen, 2016) are exclusively devoted to classification, not localization (thus requiring significant pre-processing). Moreover, some techniques (Yamamoto et al., 2017) (Salih et al., 2016) require information beyond image data, such as digital elevation models or digital terrain models, which are not available for all planetary bodies, require non-standard instruments, and have resolution limits.

1.2 THESIS CONTRIBUTIONS

1. Interdisciplinary approach to automatic crater detection, incorporating aspects of machine learning (ML), planetary geology, and systems engineering
2. Analysis and categorization of the recent crater counting papers using convolutional neural networks
    - Enumerate challenges, advice for researchers entering this area, evaluate dataset "friendliness" for ML
3. First work to develop a segmentation architecture for THEMIS Daytime Infrared data for Mars
    - Discovered the crater segmentation could still happen despite missing pixel data in imagery
    - Evaluation of the use of multiple targets for this application
    - Exploration of kernel size and filter numbers on final training using both traditional machine learning metrics (loss, accuracy) and post-processed metrics (F1 score, recall, precision)
    - Few studies compare impact of kernel size and filters so closely: from loss & accuracy on a pixel level through F1 score of counts. This is the first such study for crater counting.
    - Sensitivity analysis to determine how much data is needed for training
4. Development of framework to consider crater counting as a machine learning-driven "human-in-the-loop" project

CHAPTER 2: LITERATURE REVIEW

2.1 SOURCE

This chapter is based on a paper called “Automated Crater Detection Algorithms from a Machine Learning Perspective in the Convolutional Neural Network Era.” It aims to bridge the gap between planetary geologists and machine learning researchers and provide ideas and tools for those entering the field of automated crater counting using machine learning from either direction.

2.2 INTRODUCTION

Planetary geologists identified craters as a key to understanding planetary development in the 1970s (Crater Analysis Techniques Working Group 1979). When a major geological event occurs (like a large meteorite impact), it wipes the surface geographic slate clean. Scientists may want to determine the age of that large crater or another area of geographic interest. By counting the number and size of the craters in the geographic unit, absolute age (on the Moon) or relative age (Mars, Mercury) can be determined.

Over time, researchers have developed various automated crater detection algorithms (CDA) which are intended to speed up the process of counting craters in new areas or to find smaller craters when higher resolution data is available. These automated methods largely track to the computer science methods of the time. Many computer vision techniques have been used and more recently, machine learning methods have become increasingly popular.

Although on Mars and the Moon craters are identified down to 1 or 2 km over the entire surface (Robbins & Hynek 2012a; Robbins 2018a) capturing the smaller craters, especially over a very large region, remains a daunting task. Additionally, as new data, especially higher resolution data, is provided to the scientific community, researchers reevaluate the surface looking for additional scientific potential (analysis of secondary craters or boulder degradation).

While survey papers including some machine learning methods have been written on the topic of crater detection on planetary bodies (Salamunićcar & Lončarić 2008b; Stepinski et al. 2012; Patil & Kini 2015), none have yet covered the new machine learning approaches like the Convolutional Neural Network (CNN) based architectures of the past few years. Several other papers (Sawabe et al. 2006; Chung et al. 2014) compared a variety of methods on their dataset and included machine learning techniques among the number, but these papers came before the Convolutional Neural Network papers and techniques that are the focus of this review.

Adopting one or several CDAs across the community is a big challenge (Stepinski et al. 2012) as researchers differ in their techniques for counting. Many techniques used have been specific to a terrain region. A few papers (DeLatte et al. 2019; Silburt et al. 2019) have applied their method to a much larger and diverse set of terrain: ±30º latitude, 0-360º longitude.

While this chapter focuses on the use of machine learning for crater counting, it fits into a larger body of work evaluating machine learning for space applications. For example, Kerner et al. (Kerner et al. 2018) assess Mars images from Mars Science Laboratory rover Curiosity with a neural network and evaluate each image's use for science. Nesvold et al. (Nesvold et al. 2018) use machine learning to prioritize technology development for deflecting asteroids. Nguyen et al. (Nguyen et al. 2018) evaluate six classifiers for detecting debris disks. Shallue and Vanderburg (Shallue & Vanderburg 2018) use a CNN to detect exoplanets in multi-planet systems. This is a small sample of the journal research from only the past year and speaks to the enormous potential of machine learning to help recognize patterns and aid space science.

The major contributions of this work include: (1) categorization of the CNN techniques used on the Moon, Mars, and other planetary datasets; (2) description of potential benchmark datasets: combinations of annotation and image datasets that have been used in this research; (3) enumeration of challenges in using machine learning for crater counting; (4) discussion of promising techniques being developed in the machine learning community.

2.3 RESEARCH MOTIVATION & METHODOLOGY

The goal of this survey is to understand where and how convolutional neural networks and other machine learning techniques have been applied to the crater detection and counting problem and consider future directions by looking to machine learning research in general. Accordingly, this work has two aims: (1) collect examples and evaluate a recent trend in using machine learning for crater counting, Convolutional Neural Networks (CNN), and (2) provide some context to both the machine learning and planetary geologist communities to help these communities understand the challenges of each discipline.

Applying machine learning, and specifically techniques using convolutional neural networks, to crater counting is a natural direction for this research, but for planetary geologists to collaborate most effectively with machine learning researchers, each group needs to understand the challenges of the other discipline.

This survey takes a close look at the use of convolutional neural networks in crater counting and puts that research in context by detailing other contemporary machine learning metrics. The focus of this study is papers that feature automatic crater detection using CNN and spanning 2015-2019 (and one early work from 2005). To accomplish the goals of the review, keywords such as “crater counting,” “crater detection algorithm” were combined with “convolutional neural network,” “support vector machines,” “machine learning,” and “deep learning” to find relevant papers. Further papers were found by reviewing citations within those papers.

In total, thirteen published journal papers, book chapters, theses, and conference papers (four of which were Lunar and Planetary Science Conference abstracts) are identified as related directly to the use of CNNs in crater detection. Table 2-1 lists the criteria for including or excluding papers from this count.

Table 2-1: Criteria for including and excluding articles from comparison

Inclusion (if all criteria are met)	Exclusion (if any below criteria met)
Published 2000 or later Included crater detection or crater counting application Included experimental results	Non-English Conference papers whose contents were included in one of the journal papers (multiple publications by author)

2.3.1 CHALLENGES

Collaboration between two disparate fields has both challenges and tremendous potential. By examining some of the key challenges for both planetary geology and machine learning, researchers can better understand where the synergies and friction points are in implementation. For example, a planetary data processing task that may seem monumental to a machine learning researcher may be easier for a geologist familiar with the tools and formats of satellite imagery. Implementing several existing machine learning architectures from Github repositories may be a herculean task or part of a typical day. By working together, these two groups can take advantage of existing strengths and focus on new challenges.

Key among the various planetary geology challenges are:

- Visualization challenges. Overlapping craters, degraded craters, and oddly shaped craters make it difficult to craft good “crater” features by hand. Light and shadows render differently in each type of data: visual, infrared, and elevation. It can be difficult for humans to accurately determine crater boundaries or coverage.

- Consistency and repeatability challenges. Experts disagree on image data interpretation. Prior research has shown that there is up to a 45% difference in how expert crater counters label the same regions (Robbins et al. 2014). With so much variability among experts' hand labeling of the same region, it will be difficult to create a single crater detection algorithm that is widely accepted. A machine learning algorithm would be consistent in the application of its feature detection, but for planetary geologists to agree on the best model would itself be a challenge. The treatment of secondary (craters formed after the primary impact, usually in an ejecta ring) and degraded craters would need to be chosen carefully.

Key machine learning challenges:

- Benchmark datasets. While annotations exist of various regions of planetary bodies, machine learning researchers entering the field do not initially have clear idea of the difficulty level of an area or interest. Currently, machine learning researchers choose datasets based on convenience, accessibility, and usability. Well formatted datasets on Github or another easily accessible, open source platform will be the first choice. Established datasets of agreed upon importance are needed, and those determinations can only be made by planetary geologists.

- Label criteria. Even with perfect ability to determine crater status, the choice of labels can significantly impact results. For example, if one crater is inside another, the inner crater can be hidden if the labels are crater/non-crater but visible using "crater rim" or "location/radius" parameters. Choices of this nature permeate the problem.

- Regularization and overfitting prevention. Large networks can "memorize" the training data, but these trained models most likely would not translate to a new region. Techniques like dropout (Srivastava et al. 2014) and data augmentation (Chollet 2017) can help here, but there is no standard fix.

- Quality training data. The phrase "garbage in, garbage out" holds true here. High quality annotations enable a well-trained network. Training annotations that confuse the network with false positives and false negatives decrease the network's ability to correctly identify new craters. However, there are studies showing that neural networks are robust to significant label noise (Rolnick et al. 2017) and in some cases may actually generalize better with a small about of label noise. An important caveat is quality does not offset a lack of data.

- Transfer challenges. An algorithm that works very well for one type of terrain may be useless on another if features detected are too specialized to that terrain. Algorithms may not transfer or may transfer incompletely between planetary bodies. Furthermore, the type of loss used in training may be ill-suited if the density of features differs significantly between terrains.

- Tuning hyperparameters. Hyperparameters consist of the precise values selected for the architecture like: base architecture (fully connected, U-Net, AlexNet, etc.), number of layers, number of filters, number of training epochs, learning rate, amount of training data, activation type for each layer, etc. Some have likened tuning hyperparameters to more art than science, and searching among various hyperparameter combinations efficiently is itself an area of ongoing research with methods like: grid search, Hyperopt (Bergstra et al. 2015), and a growing field of "automatic machine learning" (Zoph & Le 2016; H. Jin et al. 2018).

Some planetary bodies have properties that change crater appearance that are not directly related to age. Earth is the extreme example with craters being filled in by lakes, vegetation, or weathering over time decreasing the appearance of craters. (Even so, a terrestrial surface algorithm was developed to detect Earth craters (W. Li et al. 2017).) On Mars, weather can erode craters over time, causing degradation, but sometimes "fresh" looking craters are actually old. In these cases, the weathering may have exhumed the craters instead (Malin & Edgett 2000). Airless bodies like the Moon or Mercury are less susceptible to this type of weathering.

A potential source of confusion for researchers new to the area is the similarity between paper titles and the overloaded use of key terminology. Seven separate papers have a variation of "automated crater detection…using convolutional neural networks" as part of the title (Benedix et al. 2018; Cohen et al. 2016; DeLatte, Crites, Guttenberg, Tasker & Yairi 2018a; Emami & Bebis 2015; Emami, Ahmad, Bebis, Nefian & Fong 2018a; Norman et al. 2018; Palafox et al. 2017). The community needs to be more specific and unique with naming. Terms like "convolutional neural network," "automatic detection of craters," "novel detection algorithm," and their variations have become so ubiquitous that sifting through the differences is challenging. In early papers (Kim et al. 2005; Enke & Merline 2005), neural networks are a novelty, but now with so many different types of CNN architectures available, more descriptive names are necessary to differentiate the techniques. Convolutional neural networks are used in a wide variety of algorithms that perform and work differently. Using a segmentation CNN (like U-Net) versus using Faster R-CNN involve CNNs in different places in the crater counting pipeline.

A challenge to evaluating previous research in the field is the lack of similarity of datasets between papers. Even sharing statistics about the size of craters in meters, kilometers, etc. (as is common in planetary geology papers) could be somewhat misleading because of the difference in dataset resolution. (This issue is solved if the paper clearly states both the size in kilometers and the dataset resolution used.) A method that works on craters larger than 1 km using a particular dataset cannot necessarily be compared directly to another if they have different resolutions. Here, knowing the relative sizes of craters in pixels is useful for machine learning researchers. Additionally, challenges that plague some datasets are unique to that type. Digital Elevation Model (DEM) and Digital Terrain Model (DTM) data does not need to consider the angle of light, but the incidence angle is a vital consideration in visual and a moderate one in infrared. Thus, the reported detection percentages must be caveated with several clarifications: type of dataset (visual, infrared, DEM/DTM), size of the crater detection in pixels (converted from the range and resolution), type of instrument, resolution of the dataset, and angle of incidence.

The value of baseline datasets is found in both planetary geology and machine learning. In planetary geology, one region near the Apollo 16 landing site is used as a reference for lunar science and instrument calibration (Pieters et al. 2008) (pg. 251). In machine learning, datasets like MNIST (digit recognition, 0-9) and CIFAR-10 (images of common objects) are commonly used to compare classification algorithms. While these baselines provide value as points of comparison, one nuance is to ensure that they are not the only datasets used because optimizing for a single dataset limits transfer potential. (To address this, Kuzushiji-MNIST (Clanuwat et al. 2018), based on cursive Japanese characters, is proposed as an alternative to MNIST to expand the usage of those algorithms.) The closest baseline in the planetary science crater counting community could be the Mars Nanedi Valles region, which is used to compare classification accuracy. For machine learning, however, identifying a larger, more diverse baseline of image data and annotations would provide a better comparison of a network's potential across various types of terrain and planetary bodies.

2.4 CRATER COUNTING

2.4.1 HISTORY

Planetary geologists use crater counting on various solar system bodies to gain insight into the relative age of different geographic regions. An area contained within one geographic unit is chosen and the sizes of craters are grouped in logarithmic size bins. These techniques were initially developed for the Moon (Crater Analysis Techniques Working Group 1979) and later expanded to other planetary bodies like Mars and Mercury (Hartmann & Neukum 2001; Neukum et al. 2001; Ivanov 2006; Michael & Neukum 2010). Crater counts on the Moon can be anchored to absolute ages using returned Apollo samples. With an understanding of the population of projectiles, researchers can and do transfer the lunar chronology to other planets to get absolute age estimates (Hartmann & Neukum 2001; Neukum et al. 2001). The community awaits samples returned from other bodies to anchor the ages of Mars and other planetary bodies, but standards for transferring the lunar chronology are accepted (Michael & Neukum 2010).

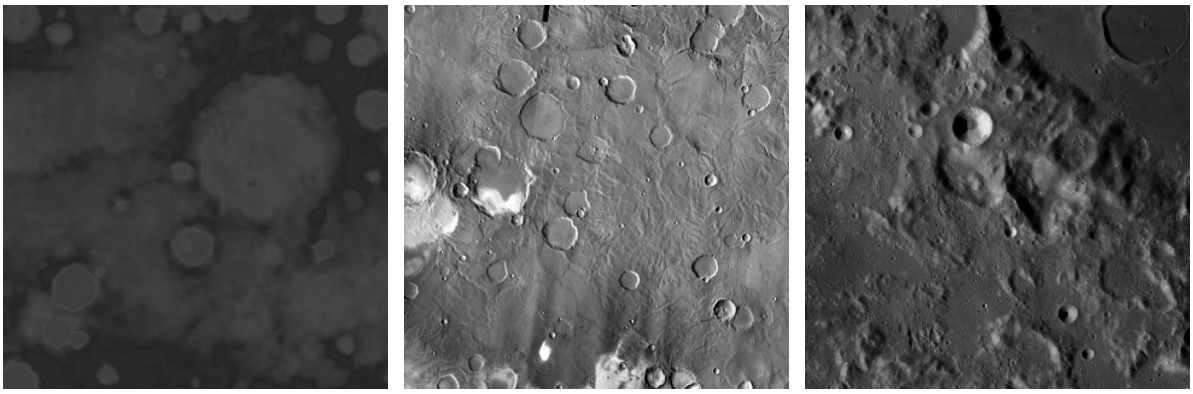
2.4.2 DEVELOPMENT: FROM HAND COUNTING TO MACHINE LEARNING

Crater counting started with hand counting (Crater Analysis Techniques Working Group 1979) and various techniques have been employed to optimize researcher time using existing technology. Expert hand labels are valued for their ability to discern the context of the terrain to distinguish between impact craters and other crater-like features. To speed up this time consuming process, many computer vision techniques have been investigated, including recent research using object-based image analysis (Vamshi et al. 2016), hand crafted 3D features (Salih et al. 2017), rotational pixel swapping (Yamamoto et al. 2017), and terrain analysis (Zhou et al. 2018). The methods are contemporaries of machine learning methods that detect craters and other geographic features like decision trees, AdaBoost (Martins et al. 2009; S. Jin & Zhang 2014; Wang et al. 2017), Support Vector Machines, and Convolutional Neural Networks (see Table 2-3). Hand crafting features lets researchers be very specific but the creation process is time consuming and prone to bias. Using machine learning methods like CNNs where the algorithm finds its own features lets the decision burden shift to human annotators who use their full visual experience to identify the craters. Instead of a human crafting a feature for detection, the CNN learns by the expert annotator’s example. Hundreds or thousands of examples of craters and non-craters tune a model, which creates the features via the training process. CNNs can therefore pick out more complex features than what could be hand crafted. There is even potential for CNNs to need fewer pixels to identify a crater than a human, but this requires further research. (For this experiment, the human would annotate a higher resolution image, and the CNN could be trained on slightly or significantly lower resolution data.) However, even today, human counts remain the standard for detecting craters. While numerous techniques for aiding crater counters have been researched, none have yet been accepted by the broader community as a replacement for human eyes and analysis (Stepinski et al. 2012). In terms of evaluating terrain context, secondary craters, and potential clusters, no machine learning algorithm has fully addressed these issues.

DATABASES

Existing planetary databases vary in their ease of use for non-planetary specialists. This is an area ripe for collaboration: databases in non-standard image formats are challenging for machine learning researchers to use, but planetary geologists are familiar with the data. Digital Elevation Models (DEM) represent one promising data type. Recent machine learning work by Silburt et al. (Silburt et al. 2019) uses DEM data from the Moon and Mercury. Other DEM datasets are available for planets and dwarf planets: Mars (Fergason et al. 2017), Venus (Magellan Team 1997), Mercury (Becker et al. 2016; Denevi et al. 2018), Vesta (Preusker et al. 2014), and Ceres (Preusker et al. 2016). DEM datasets are particularly valuable in this research due to their lack of encroaching shadows.

However, in DEM data, the complex terrain context is lost. Visual data and infrared data keep this context but are "noisier" in this regard as well. Visual and infrared images have shadows. Thus, craters at different latitudes appear vastly different due to the differing sun angle. There are myriad sources of visual and infrared images of planetary bodies in the solar system. A comprehensive resource for all types of data is the PDS Geosciences Node Orbital Data Explorer (NASA n.d.), which contains links to instrument data from Mars, Moon, Mercury, and Venus missions. Craters look different with each type of data; examples of DEM, infrared, and visible light imagery are in Figure 2-1. (These regions are chosen only to illustrate the differences between the data types; the locations have no particular significance.)



(a) DEM, Mercury **(b) Infrared, Mars** **(c) Visible, Moon**

Figure 2-1: Examples of DEM, infrared, and visible light images of craters. Craters pictured in (a) come from Mercury/Messenger (Denevi et al. 2018), (b) is infrared data from Mars/THEMIS (NASA Mars Odyssey/THEMIS Team 2010), (c) is visible data from the Moon/Wide Angle Camera Global.

Annotation datasets exist for various regions of the solar system. While full planetary coverage by a single counter or method is rare, they do exist. Table 2-2 lists several of these existing crater location databases. Researchers have used these and other catalogs to obtain ages on Mars (Platz et al. 2013; Pasckert et al. 2015) and the Moon (Hiesinger et al. 2000; Hiesinger et al. 2010). Regional databases for Mars include: GT-57633 (Salamunićcar & Lončarić 2008a), MA130301GT (Salamunićcar et al. 2011), and Nanedi Valles region (#h0905_0000) (Bandeira et al. 2010; Cohen et al. 2016). A regional database for the Moon is: LU78287GT (Salamunićcar et al. 2014).

Table 2-2: Full planetary body crater location databases

Region covered	Size of craters detected	Original image data used for counting & type	Number of craters identified	Comments	Citation
Mars	≥ 8 km (≥ 5 km)	Viking 1:2M photomosaics	25,826 craters (42,283)	Paper notes use of ≥ 8 km for crater statistical analysis purposes and to exclude most secondary craters	(Barlow 1988)
Moon	≥20 km	Lunar Orbiter Laser Altimeter (LOLA), 64 pixels per degree DTM data	5,158 craters	LOLA is an instrument on the Lunar Reconnaissance Orbiter (LRO) mission	(Head et al. 2010)
Mars	≥1 km	THEMIS (near infrared)	300,000+ craters	Largest database of Mars craters, hereafter: RH2012	(Robbins & Hynek 2012a; Robbins & Hynek 2012b)
Moon	200 km	Gravity Recovery and Interior Laboratory (GRAIL)	74 basins	Basins detected using changes in gravitational data	(Neumann et al. 2015)
Moon	5-20 km	LROC Wide Angle Camera (WAC), 100 meters per pixel, monochrome (643 nm) mosaic and DTM	22,746 craters	Extended the work of (Head et al. 2010)	(Povilaitis et al. 2018)
Moon	≥1-2 km	LRO WAC 100 m/px, LOLA, LOLA-Selene DTM 60 m/px, Kaguya 20 m/px Terrain Camera	2 million+ craters	In review (as of January 2019), will be the largest database of Moon craters	(Robbins 2018a; Robbins 2018b)

2.5 APPLICATION OF MACHINE LEARNING

2.5.1 CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolutional Neural Network research has been active for decades. Recent advances and the availability of more powerful computational hardware has made CNNs a viable option for more types of image research in the past decade. Research has proven CNNs to be effective in two distinct image processing steps relevant to the crater counting pipeline, classification and segmentation (see below). The earliest found reference to a "neural network" applied to crater counting comes in Enke & Merline in 2005 (Enke & Merline 2005), who deemed the technique not useful at the time. Much changes in ten years, especially in the machine learning community. The successful use of CNNs for classification started in 2015 (Emami & Bebis 2015) and 2016 (Cohen et al. 2016). Segmentation is being applied more recently (Silburt et al. 2019; DeLatte, Crites, Guttenberg, Tasker & Yairi 2018a). Applying this research has the potential to reduce the time spent counting craters by hand, provide a more consistent application of a crater counting technique (labeling varies between experts and even between areas done by the same expert), and eventually provide a list of smaller craters without human counting as the techniques and resolution of imagery improves.

A CNN learns features important to analyzing the image by dragging a window (kernel) across the image. The kernel size refers to the number of pixels in a square window of interest. For example, a kernel size of 3 means that a sliding window of 3x3 pixels evaluates those nine pixels according to each of the filters. By using these square windows, a CNN captures two-dimensional location information. The weights (numerical values) of a filter each determine one feature that the network can detect. CNNs learn features (filters) through the training process. In order to learn the best weights, a lot of training data is needed and additional data is necessary to validate the results.

The most popular data types in existing research are: visual/panchromatic, infrared, and elevation/terrain data. The type of data has a significant impact on the study. Visual data is available at very high resolutions for the planetary bodies of interest and is a popular choice for hand labeling craters. Some of the highest resolution data is available for this type and there are several complete sets of images for different planetary bodies. The major downside to using visible light data for automatic crater counting is the variety of lighting conditions, which can make craters near the equator look very different than craters near the poles. There are also significant variations in lighting between datasets because of the incidence angle. Infrared data, like the Mars THEMIS Daytime IR image set (NASA Mars Odyssey/THEMIS Team 2006), partially addresses the lighting condition issue. Shadows are still visible, but the gradient is less than with visual data. Another data type, used successfully in Silburt et al. 2019 (Silburt et al. 2019), is digital elevation model or digital terrain model data. This data type has no issues with shadows as only the elevation data is reported for each pixel. With the regularity and symmetry of crater circular shapes, this type works very well for detecting craters; Silburt et al. 2019 also found many new crater candidates. (Yamamoto et al. 2017 (Yamamoto et al. 2017) also take advantage of this symmetry in their non-CNN method.) The biggest drawbacks to this type of data are it not being available for all planetary bodies and resolution limitations. To collect DEM data, a laser is bounced over the entire surface. The specialized instrument required is less common on planetary missions compared to imagers.

For any of these types of data used, researchers need to take care to include a representative sample of terrain types and levels of crater degradation in training if they are trying to apply their method over diverse terrain. Randomization of the separation of data into training, validation, and test sets is part of the solution. Across the Martian equatorial mid-latitudes (±30º N), for example, encompasses volcanic, highlands, and even a small amount of basin (Tanaka et al. 2014). It does not include polar, so methods trained in the equatorial mid-latitudes may not transfer as well to the polar region without additional training. Craters look visually different in those regions for both panchromatic and infrared data. Additionally, some of the typical methods of generating additional data (like rotating, flipping, etc.) images may not work as well when the features are learning to detect shadow patterns to distinguish between craters and mountains or boulders in visible and infrared data. Other techniques like sliding and resizing do work well to generate more data and rotations do aid in DEM data (Silburt et al. 2019).

Table 2-3: List of crater counting papers that use CNN in their pipeline

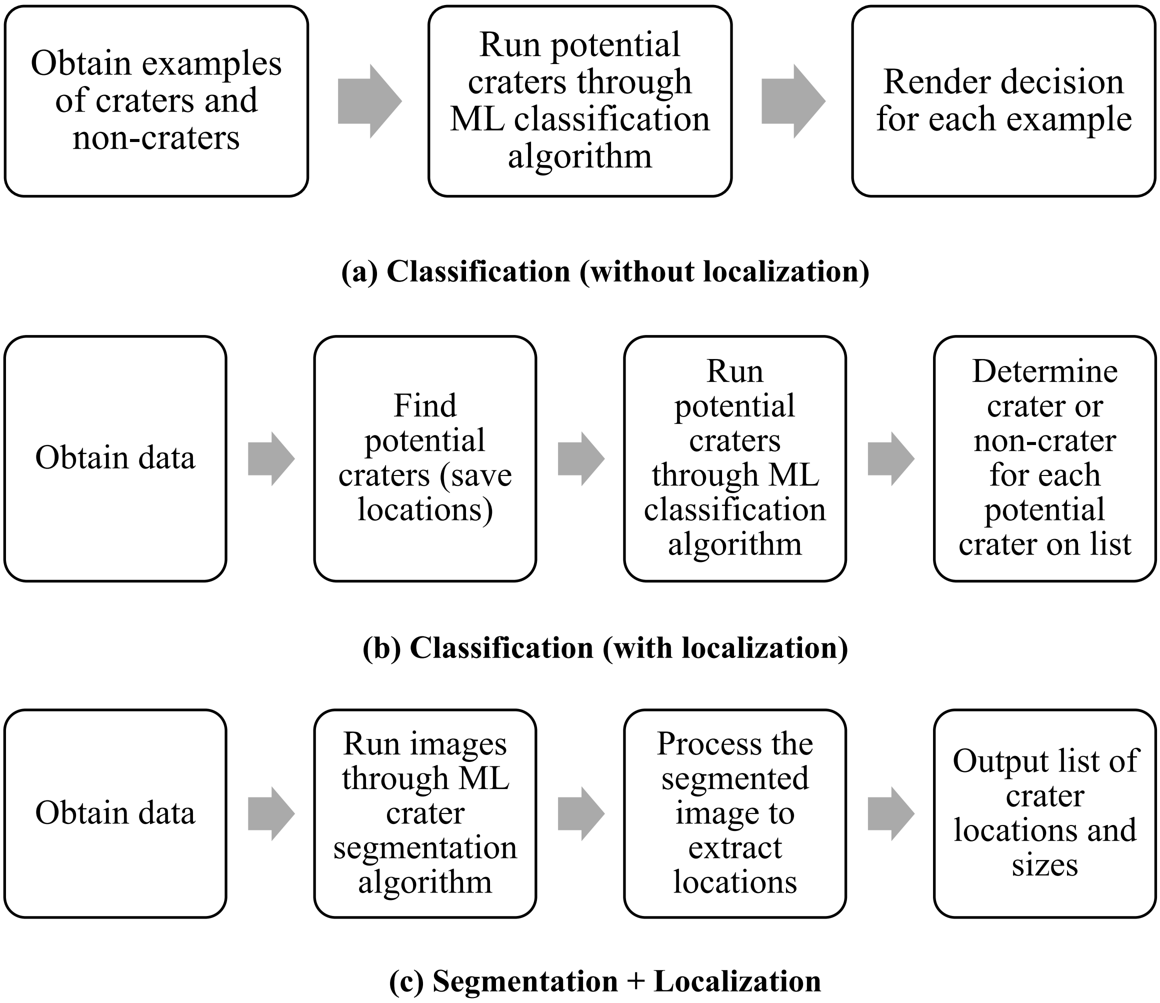
Paper Title	Source Type	Region Studied	Datasets Used	Resolution	Technique	Citation
Learning to Detect Small Impact Craters	Conference	Not specified	Not specified	Not specified	Neural network	(Enke & Merline 2005)
Lunar crater Identification via deep learning	Journal	Moon ±30º latitude, 0-360º longitude	DEM (Lunar Reconnaissance Orbiter, Kaguya)	512 px/deg, 59 m/px	Segmentation (U-Net)	(Silburt et al. 2019)
Segmentation Convolutional Neural Networks for Automatic Crater Detection on Mars	Conference, Journal (in review)	Mars ±30º latitude, 0-360º longitude	Infrared (THEMIS Daytime IR); Annotations RH2012	231.55 m/px	Segmentation (Custom U-Net)	(DeLatte, Crites, Guttenberg, Tasker & Yairi 2018a; DeLatte et al. 2019)
Automated Detection of Craters in Martian Satellite Imagery Using Convolutional Neural Networks	Conference	Mars ±30º latitude, 0-360º longitude	Not specified	Not specified		(Norman et al. 2018)
Automated Detection of Martian Craters Using a Convolutional Neural Network	LPSC abstract	Mars ±30º latitude, 0-360º longitude	Infrared (THEMIS), Panchromatic (CTX); Annotations RH2012 (>1 km), Wener (0.1-1 km)	Not specified	Pretrained CNN, GoogLeNet-Overfeat	(Benedix et al. 2018)
Lunar Crater Detection via Region-Based Convolutional Neural Networks	LPSC abstract	Moon	Hand labeled (200 tiles of 600x400 px); annotations, 270 craters larger than 20x20px	Not specified	Localization + Classification (Faster R-CNN)	(Emami, Ahmad, Bebis, Nefian & Fong 2018a)
On Crater Classification Using Deep Convolutional Neural Networks	LPSC abstract	Moon	Lunar Reconnaissance Orbiter; annotations hand labeled	Not specified	Classification (VGGNet, GoogLeNet, ResNet)	(Emami, Ahmad, Bebis, Nefian & Fong 2018b)
Recognizing terrain features on terrestrial surface using a deep learning model	Conference	Earth	Color images; self annotated various Earth craters	Not specified	Classification (Faster R-CNN, ZF-net)	(W. Li et al. 2017)
Automated detection of geological landforms on Mars using	Journal	Mars	Mars Reconnaissance Orbiter, HiRISE, CTX	HiRISE = 0.3 m/px; CTX = 6 m/px	Multi-class CNN, SVM + HOG	(Palafox et al. 2017)

Convolutional Neural Networks						
Deep Networks: Applications, Interpretability, and Optimization	Thesis	Mars	Visual; Mars Express HRSC	12.5 m/px	Haar-initialized, CNN classifier (scaled down versions of LeNet, AlexNet, GoogleNet)	(Lo 2016)
Crater Detection via Convolutional Neural Networks	LPSC abstract	Mars	Visual; Mars Express HRSC	12.5 m/px	Classification; fully connected CNN	(Cohen et al. 2016)
Automated Crater Detection Using Machine Learning	Thesis	Mars	Visual; Mars Express HRSC	12.5 m/px	Classification; fully connected CNN; RandomOut	(Cohen 2016)
Automatic Crater Detection Using Convex Grouping and Convolutional Neural Networks	Book	Moon	Visual; Lunar Reconnaissance Orbiter; annotations of hand labeled custom set	Not specified	Classification (Regions proposed with multi scale edge detection; candidates classified with CNN)	(Emami & Bebis 2015)

2.5.2 USE OF CNN FOR CRATER COUNTING

Machine learning, and specifically CNNs, can be used at various points within the crater counting pipeline. The pipeline consists of the steps between taking an image that contains craters and outputting a list of the crater locations.

While the descriptor "Convolutional Neural Network" helps identify machine learning papers, the use of a CNN model in a crater counting pipeline can vary greatly. Two major CNN research directions have emerged for the crater detection application: (1) classification methods (Figure 2-2a, Figure 2-2b) and (2) segmentation (plus localization, finding the relative pixel locations within the image of the craters) methods (Figure 2-2c). The main difference from the CNN model's perspective between the categories is the scope of classification. In the first, the entire pre-processed (usually) square image is classified as a crater or non-crater image. In the second, a large image containing multiple craters (zero to hundreds) is passed in and each pixel gets classified as belonging to a crater rim or not. (This means each pixel is designated as either a crater or non-crater pixel.) The rim crater pixels form rough circles or ellipses which can then be detected and used for localization.


Obtain examples of craters and non-craters
Run potential craters through ML classification algorithm
Render decision for each example
Obtain data
Find potential craters (save locations)
Run potential craters through ML classification algorithm
Determine crater or non-crater for each potential crater on list
Obtain data
Run images through ML crater segmentation algorithm
Process the segmented image to extract locations
Output list of crater locations and sizes

(a) Classification (without localization)

(b) Classification (with localization)

(c) Segmentation + Localization

Figure 2-2: Comparison in the use of trained CNNs (or machine learning) for classification vs segmentation: (a) Shows the classification pipeline, (b) classification with localization, (c) segmentation and localization.

Some CNN-based research classifies pre-processed images as crater or non-crater while other research segments the craters using a pixel-wise CNN classifier to create image maps of circles indicating where craters are and separately uses computer vision based circle finding methods to make a list of craters. The machine learning components of these two approaches serve a difference purpose, thus are not directly comparable. The main difference between these two methods is the function of machine learning. Comparing results between classification and segmentation methods is challenging because it is important to compare metrics created by a similar process. In classification methods (after training), crater candidates must be pre-processed and rescaled, then each candidate is presented to the network for evaluation. This can be done by employing various computer vision techniques such as generating PHOG (Liu et al. 2012) or Gist (Yin et al. 2015) features, then passing the candidates through a CNN classifier. The location of the candidate examples must be known. The recall scores (number of matches divided by number of annotations) (Chinchor 1992) tend to be extremely high, even 99% (Emami, Ahmad, Bebis, Nefian & Fong 2018b), but there is a significant effort in preparing crater candidates. (This is very promising research, but the test set in that particular paper was limited). In segmentation plus localization methods, the entire image is given to the algorithm, which splits it into chunks irrespective of the locations of craters, identifies crater candidates, which then can be post-processed to find the location. Traditional computer vision methods like template matching can be used to turn the newly identified circular shapes into a list of location and radius.

The feature identification step is treated differently in these two categories. Several of the classification methods used either existing image databases of crater and non-crater examples or created their own for the purpose of training the classifier. (One example that is less labor intensive in advance is to use a Region Proposal Network, as used in Faster R-CNN (Ren et al. 2016).) Then the crater examples are passed through the CNN-based classifier, which could have one of various architectures. Example architectures used in image classification include GoogLeNet, Alexnet, VGGNet, and ResNet.

This second segmentation category is less labor intensive in data preparation for the crater counter because during the training process, the features are automatically generated by the CNN. Further images can be presented as they are without the pre-processing. Once the model is trained, this is quick. Post-processing, however, can be time consuming because the computer vision template matching algorithms are comparatively slow. For example, for the work in Chapter 3, to evaluate a 30º by 30º THEMIS tile takes 2 minutes with a GPU, but the post-processing with the match template algorithm to obtain the full list of craters sized from 7 to 140 px in the tile (7680 x 7680 px) takes 20 minutes with a CPU.

While it is tempting to compare the percentages of recall between methods directly, results must be taken in context of terrain, crater size (in relation to pixels), network type, pre-processing complexity, and post-processing complexity. Comparing CNN classification methods versus segmentation methods recall percentages directly removes the vital nuances of the implementation complexities, scaling up challenges, and transfer learning potential. Another important distinction is understanding which terrain was used in training and test. A clear understanding of these differences allows future researchers to choose the best option for the scale of data they plan to evaluate.

Several researchers, including Emami et al. (2015) (Emami & Bebis 2015), Cohen et al. (2016) (Cohen et al. 2016), and Palafox et al. (2017) (Palafox et al. 2017) evaluated the performance of CNN classifiers on planetary datasets. In Emami et al. (2015) (Emami & Bebis 2015) and Cohen et al. (2016) (Cohen et al. 2016), resized and centered examples of craters and non-craters were used to train the CNN classifier. Palafox et al. (2017) use a multi-size classifier structure that could distinguish between examples of volcanic rootless cones and transverse aeolian ridges. (The same architecture was used to train separately for each geologic feature of interest.) While that research was not explicitly looking for craters, the same method could be used to find other geologic points of interest such as craters. For some applications, training a multi-class classifier can improve the recognition of objects of each class. This is theorized to be due to the additional training data available and ubiquity of some types of features (such as edges or shapes) (Emami, Ahmad, Bebis, Nefian & Fong 2018b). Future research validating that for craters may inspire crater counters to more directly collaborate with boulder counters or groups interested in other geological features.

A promising lead for classification is to use a network first trained on other data. Benedix et al. (Benedix et al. 2018) and Norman et al. (Norman et al. 2018) use a pre-trained GoogLeNet-OverFeat to detect craters. Emami et al. (Emami, Ahmad, Bebis, Nefian & Fong 2018b) pre-trained using ImageNet, then compare VGGNet, GoogLeNet, and ResNet, obtaining over 99% recall on their hand-annotated dataset. While very promising, this research is difficult to independently evaluate due to the lack of a baseline.

Emami et al. (Emami, Ahmad, Bebis, Nefian & Fong 2018a) continued research in the use of CNNs and implement Faster R-CNN for full pipeline crater detection. Although the scope of the craters used was limited, this research is promising and should be expanded to other annotation datasets and crater sizes. Unlike the methods that need hand crafted features, Faster R-CNN uses Region Proposal Networks to identify potential objects, which are then each resized and classified. Faster R-CNN outputs the location correction with the classifications, which removes the need for a post-processing step like template matching.

Segmentation methods, those that extract the edges of the target object, are used by both this work (Chapter 3) and Silburt et al. (Silburt et al. 2019). Both research groups used segmentation networks inspired by U-Net (Ronneberger et al. 2015). Chapter 3 describes evaluating different segmentation targets (edge-only and filled in crater) on a custom "Crater U-Net" to determine which had the best recall score for craters using infrared Mars images and the Robbins & Hynek annotations (RH2012) (Robbins & Hynek 2012a). Silburt et al. (Silburt et al. 2019) use the original U-Net architecture to segment crater edges for lunar digital elevation model data and the Povilaitis et al. (5-20 km) (Povilaitis et al. 2018) and Head et al. (>20 km) (Head et al. 2010) annotation datasets. Both use a computer vision technique, template matching, to extract the crater locations and create the final list of craters. Silburt et al. also use transfer learning to apply their model to Mercury craters. The use of segmentation for crater counting has several advantages: the images do not need to be pre-processed into smaller "potential crater" images before being sent to the network and automatic localization is possible through the use of computer vision circle-finding methods. To perform localization, no bookkeeping needs to be done, and this method most closely resembles that of a human crater counter.

2.5.3 OTHER MACHINE LEARNING METHODS

Several other machine learning methods have been applied to crater counting, including Support Vector Machines (SVM) and decision trees. With most of these methods, the crater candidates must be localized prior to the machine learning algorithm being run. Accordingly, the locations are known a priori, but the identity of “crater” or “non-crater” is not. Several of these examples involve hand crafted features, especially in earlier research. Machado et al. (Machado et al. 2015) use Haar textural features and a SVM classifier to detect craters in Kaguya Terrain Camera (evening illumination, 7.4 m/px) data of the lunar maria. Li et al. (B. Li et al. 2015) use binary decision trees to evaluate LOLA DEM data. These and other early machine learning research fall in the classification category with hand crafted feature development (PHOG, Gist, Haar).

2.6 CONCLUSIONS

Analysis of sensor and image data presents a valuable collaboration opportunity for machine learning researchers and planetary scientists. Planetary scientists can contribute their vast knowledge of dataset and formatting nuances. They can enable better collaboration by deciding which benchmark datasets are the best for algorithm comparison. Machine learning researchers can contribute expertise in image processing. Methods like segmentation and localization have been extensively developed in that community and are highly applicable to planetary analysis. Those who want their work to be used by the planetary science community can export their crater lists in formats friendly to the most popular crater analysis software, like JMARS (for verifying the list of craters) and Craterstats (for obtaining ages).

Since CNNs are used in several distinct ways, including segmentation and classification, throughout the crater counting pipeline, it is more important than ever to carefully define the methods. In order to improve the collaboration potential and enable machine learning researchers to build on existing research, key information needed in papers includes: technique(s) being used during each stage of crater identification and processing, annotation dataset(s) used to create training data, source of data, data augmentation methods, regions used for training, regions used for testing, hardware (i.e., specifications for Graphical Processing Units, Tensor Processing Units), training time, hyperparameters, and source code for repeatability.

Although there is not yet an accepted crater detection algorithm to replace human crater counters, the techniques presented represent tremendous progress to eventually reduce the time needed for new data to be processed. The recent trend of starting with pre-trained networks may ultimately represent the best solution, but until the community agrees on a set of benchmark annotations to be used for such training, exporting results to the formats of popular age dating programs and working with a human in the loop is the best short-term solution. Collaboration between planetary geologists and machine learning experts will enable great research and improvements in the way craters are detected.

CHAPTER 3: EXPERIMENTS WITH CRATER U-NET

3.1 SOURCE

This chapter is based on a paper called “Segmentation Using Convolutional Neural Networks for Automatic Crater Detection on Mars: Experiments and Analysis.” It explores the use of a custom segmentation CNN (based on the U-Net architecture) for the crater counting application.

3.2 INTRODUCTION

Here, the authors explore a machine learning technique called segmentation (J. Long, Shelhamer, & Darrell, 2015) (Garcia-Garcia, Orts-Escolano, Oprea, Villena-Martinez, & Garcia-Rodriguez, 2017), where an image map is created of all the detected objects (in this case, craters). The image map marks each pixel with a value related to whether the pixel in the original image belongs to the object category of interest. Segmentation relies only on images and a training set of existing annotations for crater size and location. Once trained, the machine learning model is used to detect craters in new images. The source of annotations is the Robbins & Hynek’s Mars crater dataset (Robbins & Hynek, 2012a) (Robbins & Hynek, 2012b) (hereafter RH2012). These annotations consist of the latitude, longitude, and diameters of all craters larger than 1 km on Mars. The source of image data is the Mars daytime infrared images from the NASA THEMIS instrument (NASA Mars Odyssey/THEMIS Team, 2006).

Challenges to automation include: the orders of magnitude differences in object sizes, faded or old craters being nearly invisible, need to work with multiple datasets, different resolutions of images, projection difficulties, differing techniques for counting, and secondary craters influencing the count.

3.2.1 CONTRIBUTIONS

This work advances a collection of efforts by crater counters and machine learning researchers to automate the detection of craters. This paper is the first to use a segmentation Convolutional Neural Network to find Martian craters in THEMIS thermal infrared data (NASA Mars Odyssey/THEMIS Team, 2006). Crater U-Net introduced here is a Convolutional Neural Network (CNN) for image segmentation that is inspired by the U-Net (Ronneberger, Fischer, & Brox, 2015a). Mars crater annotations (Robbins & Hynek, 2012a) are used for training. Contemporary researchers have independently used a U-Net to find lunar craters in digital elevation data (Silburt et al., 2019). Additional contributions of the current work include an exploration of the impact of kernel size and filter numbers on accuracy, an evaluation of the effect of using two different types of targets (solid circles vs. edge targets), and an assessment of the effect of varying the amount of training data used to create the model, which sheds light on the tradeoff between quantity of hand annotations and model quality. Together these contributions provide insights that support the design of automated crater counting frameworks for new datasets.

3.3 METHODS

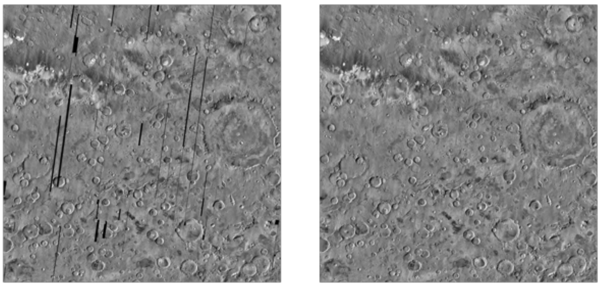
These experiments study the use of segmentation Convolutional Neural Network (CNN) architectures applied to the crater detection and localization problem, specifically using a variation on a CNN called a U-Net (Ronneberger, Fischer, & Brox, 2015b). CNNs capture local features in a layer by sliding a square kernel across the image, restricting the input to a sequence of small regions of size, e.g. 3x3 or 7x7 pixels, and multiplying it by a filter. The number of times this process occurs depends on the number of filters in the layer. A series of tests systematically explores hyperparameter, or architecture design choices (different kernel sizes and number of filters), target options (solid and edge), and the impact of using different amounts of training data to train the models (from 3 to 15 of 24 tiles).

3.3.1 DATA

For supervised learning, a dataset of annotated data-target pairs is needed to train a convolutional neural network in the art of detecting craters. The largest, most complete (Robbins & Hynek, 2012a) dataset of Martian craters available is that of Robbins & Hynek (RH2012) (Robbins & Hynek, 2012a) (Robbins & Hynek, 2012b). This comprehensive dataset identifies craters down to 1 km diameter. Each row of the annotation spreadsheet includes latitude, longitude, and diameter for a circle and ellipse, each representative of the crater. For training the neural network, circular representations of craters are used to create the targets and all annotations are included, regardless of crater degradation level. As a result, non-experts inspecting data images may not be able to identify the old craters by eye. It is important to note that since a single annotation set is used, this model is being trained to find craters in the style of RH2012. Previous work (Robbins et al., 2014) (Bugiolacchi et al., 2016) indicates even experts disagree. Up to ±45% difference in annotated crater sizes and distribution between crater counting experts is found.

THEMIS Daytime Infrared (IR) Global Mosaic (12.7um) images (Figure 3-1) of Mars are chosen because of the dataset's high resolution and their use by (Robbins & Hynek, 2012a) in creating the annotation database. The THEMIS mosaic tiles are each 30º per side (7680 x 7680 pixels) with a resolution of 256 pixels per degree (PPD) and have excellent visual match with the RH2012 annotations without adjusting the projection.

One challenge in using the THEMIS images is the absence of data in some areas, seen as the black streaks in Figure 3-1. To mitigate the missing pixel issue, all black pixels in the image are replaced with the average value of the tile before training. This preprocessing step keeps the missing data from becoming a hard edge that might interfere with training. This approach was successful: no obvious network errors were detected in the vicinity of the missing pixels, and the network is able to identify craters which partially overlap the missing region. Craters entirely within a missing pixel region are not detected.



(a) (b)

Figure 3-1: original THEMIS tile, thm_dir_N-30_030 (NASA Mars Odyssey/THEMIS Team, 2006) (a), tile with filled in missing pixels (b)

Choosing the latitude range ±30º simplifies processing and avoids the need to re-project high latitude THEMIS data. Within ±30º latitudes, the craters in the cylindrical projection remain circular. In extreme latitudes by the poles, the cylindrical projection stretches the appearance of the craters into ellipses. This subset of the THEMIS dataset contains 24 “tiles” of 30º by 30º, shown in Figure 3-2. Each is 7680 x 7680 pixels (px). These initial tests use a subset of the available (Figure 3-3) annotations: craters sized 2-32 km in radius (Figure 3-4). This range is comfortably within the limits of the annotations (all craters larger than 1 km diameter) and is clearly visible within the 512 x 512 px sub-images chosen as the network input size. This sub-image divides evenly in the full tile pixel size (15 per side). For reference, a 2 km radius crater is 8.6 pixels at this dataset’s resolution (256 pixels per degree or 231.55 meters per pixel (NASA Mars Odyssey/THEMIS Team, 2006)).

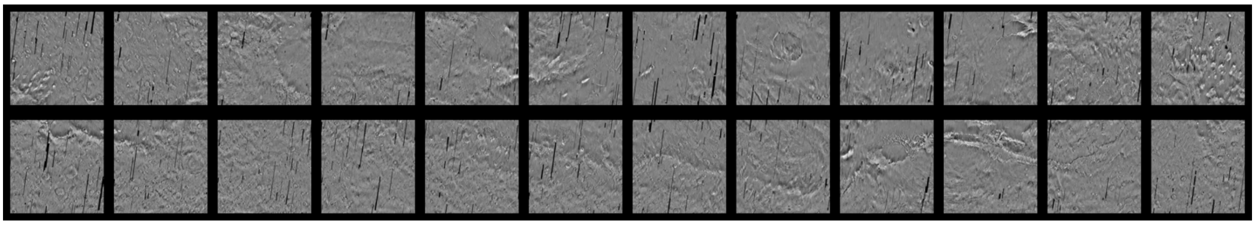


Figure 3-2: Mars in the Mid-Latitudes, original THEMIS data (NASA Mars Odyssey/THEMIS Team, 2006). The top row consists of 0º to 30º latitude, and the bottom row -30º to 0º latitude.

The left edge is 0º longitude. Each tile is 30º by 30º (7680 by 7680 pixels) with a resolution of 256 pixels per degree (231.55 m/px).

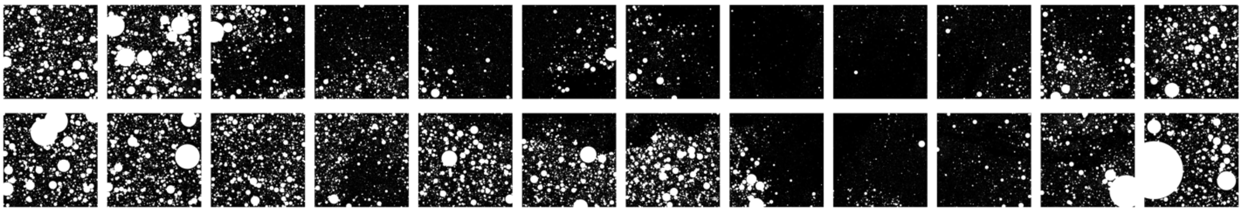


Figure 3-3: Graphical display of annotations of all craters annotated by RH2012 (Robbins & Hynek, 2012a).

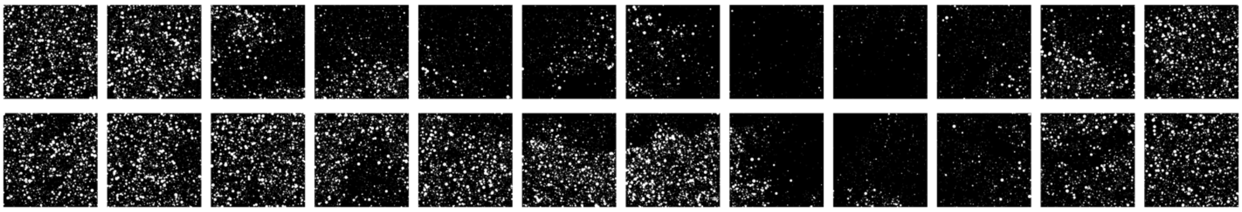


Figure 3-4: Graphical display of annotations of craters 2-32 km radius from (Robbins & Hynek, 2012a). These are one set of targets for the data.

For the training process, the 24 tiles are split randomly into one of three groups: training, validation, and test. For each neural network architecture, the training tiles and corresponding annotations (target) are used to learn a model. Validation data is used to compare architectures on data unseen in training. In Section 3.4.1, six tiles are used for training and three for validation. In Section 3.4.2, 3 to 15 tiles are used for training, depending on the test, and six are used for validation. The tiles used for training and validation are selected randomly with no overlap. The data tiles and generated target images are split into 512 x 512 pixel sub-images before being fed through the network (3.3.2).

3.3.2 SEGMENTATION NETWORK DESIGN

While both classification (Krizhevsky, Sutskever, & Hinton, 2012) and segmentation (J. Long et al., 2015) have uses in image processing, for this application, instance segmentation is most appropriate for automating crater counting. Segmentation differs from standard image classification in that its result effectively separates the objects of interest from the background. On the smallest scale (one pixel) and for a one-object detection, "segmentation" amounts to a binary classification of each pixel, giving a measure of whether the pixel is part of an example of the objects being identified. Semantic segmentation does not distinguish between examples of the object while instance segmentation identifies each example object separately (Garcia-Garcia et al., 2017).

Humans who manually count craters consider many details such as edge geometry, patterns of light and shadow, local geologic structures, and angle of impact when classifying a true crater. For example, secondary craters (ejecta from the primary impact) are typically omitted from the count, even though they largely look like primary craters (Robbins & Hynek, 2011). Segmentation methods are similar, from a data processing perspective, to what humans would do: take a whole image, identify the craters, and determine the number and locations of those craters inside the region of interest.

The family of Crater U-Nets explored here is inspired by results of Ronneberger et al. (Ronneberger, Fischer, & Brox, 2015a), who originally developed U-Net for the segmentation of biological cells in images. The default network (Figure 3-5) has the following components: Rectified Linear Unit (ReLU) activation function (Krizhevsky et al., 2012), convolutional kernel size of 3x3, and filter values of the layers (starting at the upper left of Figure 3-5): [16, 24, 32, 48, 64, 96, 128, 128, 128, 96, 64, 48, 32, 24, 16]. Convolution (Chollet, 2017), dropout (Srivastava, Hinton, & Salakhutdinov, 2014), average pooling (Chollet, 2017), and upsampling (Chollet, 2017) layers are used. The network is coded in Python using Keras (Chollet, 2017), a framework for developing neural network models, and a TensorFlow (Abadi et al., 2016) backend.

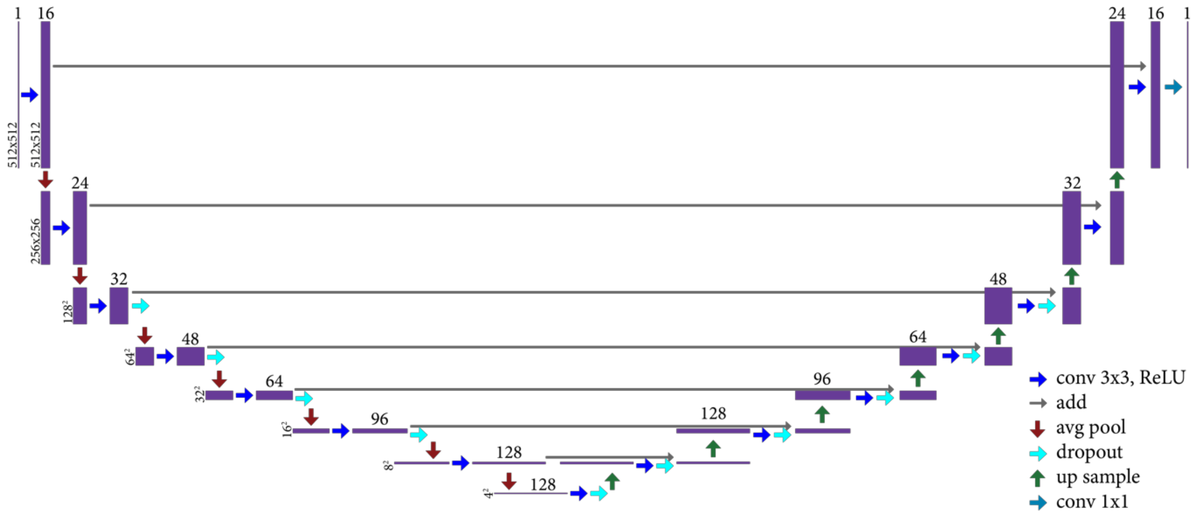


Figure 3-5: Default Crater U-Net structure, visualized in the style of (Ronneberger, Fischer, & Brox, 2015a). For this network, the kernel size is 3x3, and filter values for each layer are on top of the purple boxes.

Data (grayscale 512 x 512 px image) enters the Crater U-Net on the upper left. As it "goes down" the left half of the network, downsampling occurs as features at larger scales are computed. After seven rounds of downsampling, upsampling though seven layers ("going up") produces an output array of the same size as the original input. Names of the Keras layers are used in Table 3-1; see (Chollet, 2017) for layer definitions):

Table 3-1: Layers in Crater U-Net

Layer type	Comments
Going down	
Conv2d	Filter value, kernel size, “same” padding, ReLU activation
Dropout (where indicated)	Increasing from 0.2 to 0.5
Averagepooling2d	
Bottom of the “u”	
Conv2d	
Dropout	

Going up	
Upsampling2d	Size 2
Conv2d	
Dropout	
Add	Adding the result of the previous dropout with the correspondingly sized conv2d layer from going "down"
Final layer	
Conv2d	Kernel size 1, "same" padding, sigmoid activation

In Section 3.4.1, variations on this default include (1) changing the kernel size (3x3, 7x7, 11x11) and (2) changing the number of filters for each layer (½, 1, and 2 times the number of filters in the default array).

The reported loss uses the binary cross-entropy function, and the accuracy uses the default Keras definition (Chollet, 2017). The hardware used for the tests is a Deep Learning Box with four 8GB NVIDIA GeForce GTX 1080 Graphical Processing Units (GPUs). Up to four tests can run simultaneously, each on one GPU. The number of filters and kernel size for the largest test is at the limit of this GPU's capability.

3.3.3 CRATER DETECTION PIPELINE

After training, in order to get from raw data to a list of detected craters, each of the 24 large THEMIS tiles goes through the following:

- Split 7680 x 7680 px large THEMIS tile into 512 x 512 px sub-tiles
- Each 512 x 512 px sub-tile is fed to the segmentation network
- Results from the prediction are stitched back together into the full-size 7680 px tile. At this point, the prediction is still an image that can be compared to an image of the targets.
- The match template algorithm is used to detect the locations and radii, using the

method described by Silburt et al. (Silburt et al., 2019). The matching algorithm looks only for matches within the range of interest (2-32 km). Craters found by the network that are larger or smaller are excluded.

- The crater counts are compared to the human annotations, again using the Silburt et al. method (Silburt et al., 2019).
- Crater counts are converted from their pixel x, y, radius to latitude, longitude, diameter.

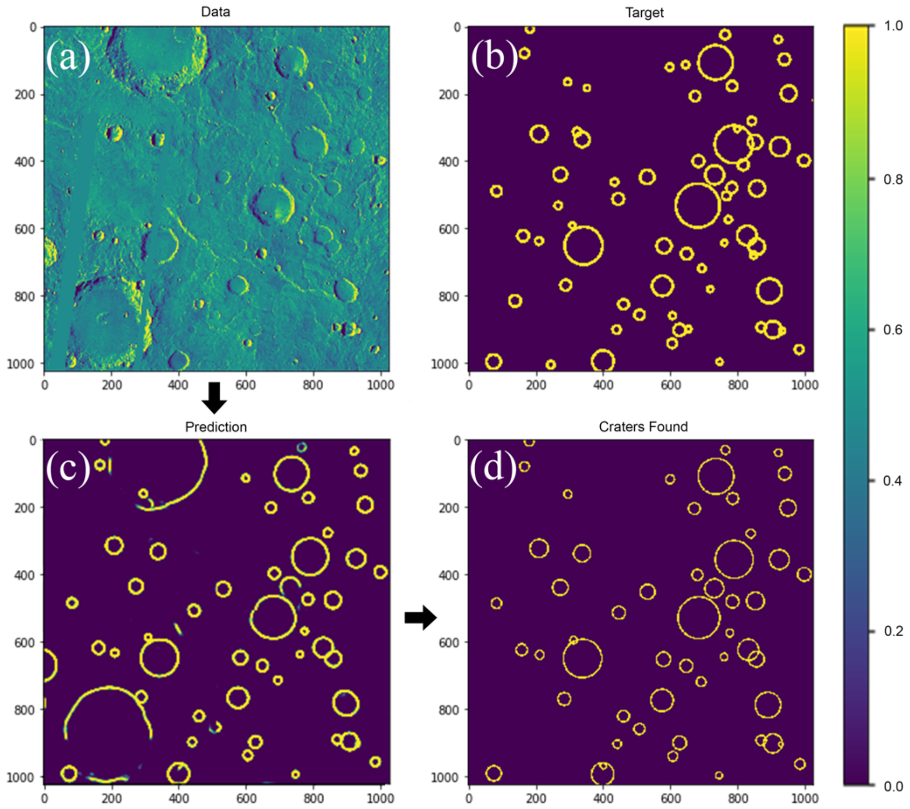


Figure 3-6: Images representing the crater detection pipeline. First, the data (with black pixels filled in) is passed to the network as a grayscale image, shown here with a color palette for visualization (a). The network creates a "Prediction" image which gives each pixel a value 0-1. (The "target" (b) image is included for the reader's visual comparison to the "craters found" image, but this example was not used during network training.) Finally, a threshold is selected (0.4 here) and the prediction image (c) is run through the matching algorithm. (The matching algorithm looks only for craters that match the 2-32 km range and excludes others.) This list of x, y, radius is plotted to create the "Craters Found" (d) image.

The example in Figure 3-6 shows a 1024 x 1024 pixel subset of the full tiles (after they have been stitched back together from 512 x 512 px). The data sub-tiles are each passed to the network, which evaluates the sub-tile, resulting in the “Prediction” image. This is processed through the match algorithm (Silburt et al., 2019), generating a list of crater location and sizes (x, y, radius), which are used to generate the final “Craters Found” image. In the final image, the width of the edges has no meaning since this was generated from a list of coordinates and sizes.

3.3.4 EVALUATING CRATER DETECTIONS

Many pixel-based measurements are available for evaluating the results of a trained neural network. However, for the application of creating lists of craters in a region, these loss and accuracy measures are not the most suitable for evaluating which model will give the best scientific result. Scientists are less concerned by a pixel out of place and more concerned with whether the crater was detected. A metric derived from the actual crater counts is preferred. As a result, the F1 score (Chinchor, 1992), the harmonic mean of precision and recall (Sasaki, 2007), is used on a per identified crater basis.

In order to calculate a F1 score for crater counts, definitions of true positives (TP), false negatives (FN), and false positives (FP) are needed. True positives are matches between the human annotated set and the list produced by the pipeline. False negatives are the craters from the annotations list that are not found. False positives are craters that are identified but that do not match the human annotation list. In other words, FP in Eq. 1 is the difference between the number of found craters and the number of matches. It is worth noting that false positives may indeed be craters missed in the original annotated set. These could be newly identified craters, but here we note this as a failure and discuss it further in Section 3.5.4.

$$Precision = \frac{TP}{TP + FP} = \frac{matches}{matches + (found - matches)} = \frac{matches}{found} \qquad \text{Eq. 1 Precision}$$

$$Recall = \frac{TP}{TP + FN} = \frac{matches}{matches + (annotations - matches)} = \frac{matches}{annotations} \qquad \text{Eq. 2 Recall}$$

$$F1 = 2 * \frac{precision * recall}{precision + recall} = 2 * \frac{\frac{matches}{found} * \frac{matches}{annotations}}{\frac{matches}{found} + \frac{matches}{annotations}} = \frac{2 * matches}{found + annotations}$$ Eq. 3 F1

In order to evaluate the matches, found, and annotations, the match template method used in (Silburt et al., 2019) is adapted to the specifications of this dataset. This method is a brute force template matching algorithm that sets a target threshold (tt) on the prediction image, finds the examples that match the size range of the circle shapes, then compares the results to the annotations (in this case RH2012 (Robbins & Hynek, 2012a)).

3.4 RESULTS

This section describes experiments aimed at implementing the network then tuning it to improve performance. The next sections detail results from each of the test campaigns. Each campaign includes the exact same set of architecture tests, detailed in Section 3.4.1. Sections 3.4.1.1 and 3.4.1.2 contain tests evaluating the use of different targets on the crater data tiles (with missing pixels filled). The same training data is used for each of the tests. The key difference is in the "target" or goal. Sections 3.4.1.1 includes tests using a filled circle while Section 3.4.1.2 discusses two sizes of edge targets. Section 3.4.1.3 aggregates the results and Section 3.4.1.4 evaluates the models using crater counts. Section 3.4.2 presents results from changing the selection of tiles used in training. Section 3.4.3 uses one of the models to do a proof of concept example of age dating.

3.4.1 NETWORK TESTING WITH DIFFERENT TARGETS

Data-target pairs are used to train the machine learning algorithm. This section details the results of testing with two types of targets. The results are evaluated by comparing the loss and accuracy. Two types of targets are explored: solid circle targets (experiments using this target have names starting with C) and edge targets of fixed width (E, F). Examples of each type are found in Figure 3-7. The streaks visible in Figure 3-7a are always filled with the tile’s average gray value prior to injection in the pipeline.

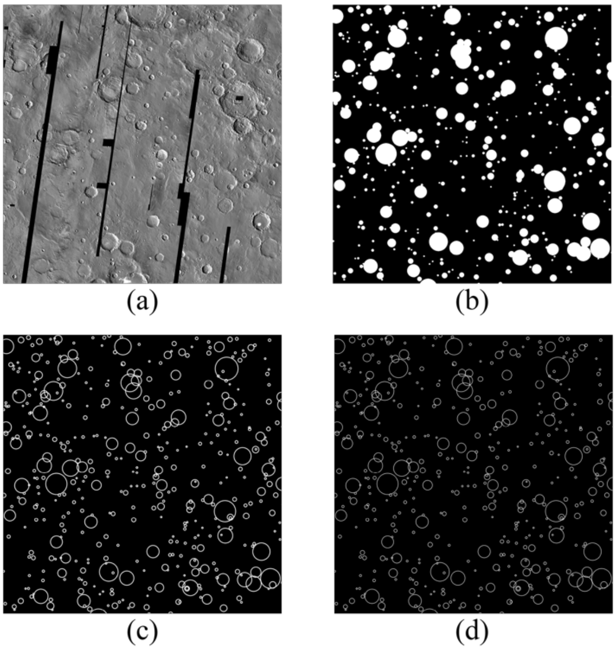


Figure 3-7: Examples of the data and targets are pictured: unmodified data from (NASA Mars Odyssey/THEMIS Team, 2006) (a), solid circle target (C) (b), edge target (E) (c), and thinner (half width) edge target (F) (d).

Table 3-2: List of Tests Used in Each Target Campaign

Test	Number of Filters by Layer	Kernel	Network Tests IDs	Seconds per epoch
*-1	default (Figure 3-5)	3x3	C-1, E-1, F-1	33-35
*-2	default	3x3	C-2, E-2, F-2	34-37
*-3	default	3x3	C-3, E-3, F-3	35-37
*-4	default	7x7	C-4, E-4, F-4 (Note, tests in Section 3.4.2 also use this: 1A to 1E, 2A to 2E, 3A to 3E, 4A to 4E)	67-75
*-5	½ default	3x3	C-5, E-5, F-5	29-30
*-6	2x default	3x3	C-6, E-6, F-6	52-59
*-7	default	11x11	C-7, E-7, F-7	129-137
*-8	2x default	7x7	C-8, E-8, F-8	126-130

For all tests in Section 3.4.1, the network architecture numbers are the same between datasets. For example, tests C-3, E-3 and F-3 have the same network architecture (same convolutional kernel size, filters, activation, layer types, etc.). The letters C, E, and F refer to the different data-target pairs each was trained on: filled circle targets, thicker edge targets, and thinner edge targets, respectively. For all tests in this section, six tiles are used for training and three are used for validation.

For relative training times when using the hardware (Section 3.3.2), the seconds per epoch are given. Although exact values can differ by a few seconds on different runs using different GPUs (or even throughout training), the relative ratios give insight into training time between models. Test *-8 takes almost four times as long as test *-1 to train.

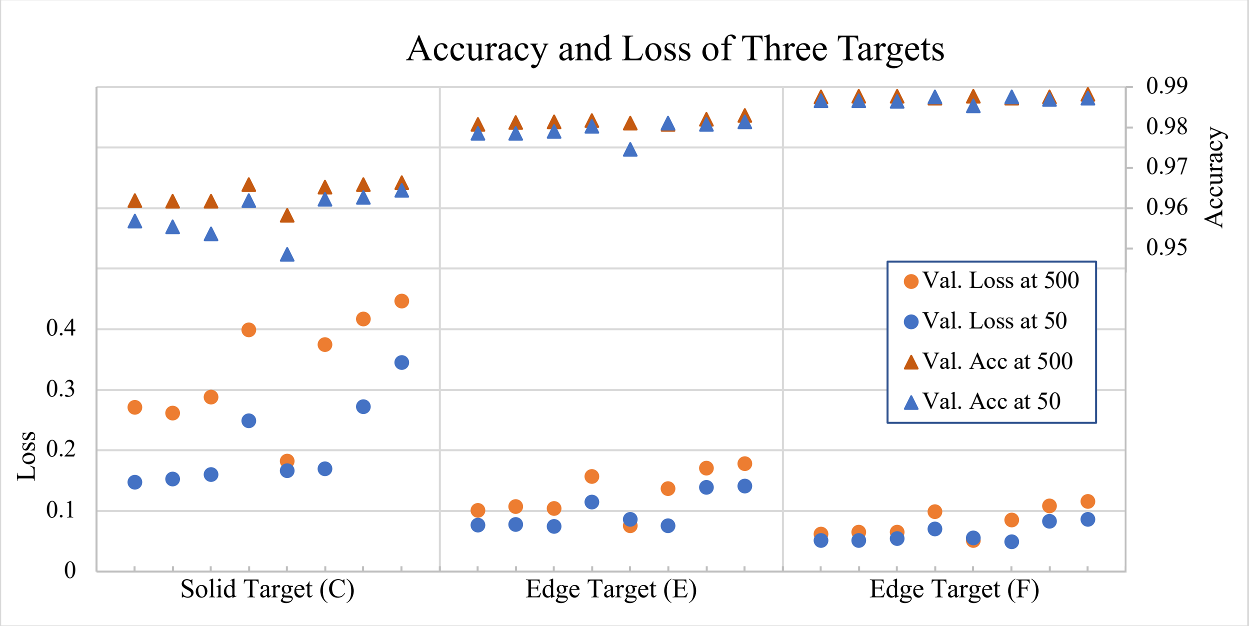


Figure 3-8: Summary of network training on the three different targets. Loss and accuracy values are shown at 50 and 500 epochs during training. Within each of section for target type, network architecture 1 - 8 are shown consecutively so that the left-most point is the result from test C-1, a 3x3 kernel network with default filters, and the right-most point is test F-8, a 7x7 kernel network with twice the number of filters.

Figure 3-8 summarizes loss and accuracy among the three targets. While loss (lower is better) and accuracy (out of 1, higher is better) cannot be compared directly across target types because of the different number of pixels engaged for training in each target type, within a single target type we can use this comparison to select the most promising architectures to explore in detail.

3.4.1.1 CIRCLE TARGETS (FILLED)

This section details the results from the experiments using crater data with filled circle targets. An example of a target tile, corresponding to one of the THEMIS 30º tiles is in Figure 3-7b. The test numbers match the architecture from Table 3-2. Kernel and filter numbers are not shown.

To create these targets, a solid circle of the appropriate radius is added to the image using the “white” color (set pixel value to 1), leaving all remaining pixels black. This method falls under semantic segmentation because it does not distinguish between different instances of craters.

The accuracy of each of the eight architectures plotted for each target in Figure 3-8 improves by a small amount between 50 and 500 epochs of training. (One pass through all of the training data is one epoch.) The loss (difference between the expected and actual output) went up dramatically in all but C-5, which likely indicates the model is overfitting where features specific to the training examples and not those relevant to generalized crater identification are being found.

Figure 3-8 reveals small differences between the training performance of the individual architectures. While little difference is seen between the loss and accuracy measurements in training, visual inspection of the results reveals that kernel size is a significant factor in successful detection of faded craters. Several examples of this are shown in Figure 3-9. A large crater in the bottom right of the tile (Example #1, red box) is only identified by networks with larger kernel values (C-4, C-7). This effect is also seen for smaller craters in Example #2 and #3. The original validation 512x 512 pixel image along with the solution “target” image are included to help the reader evaluate the relevant predicted images from tests C-3 (3x3), C-4 (7x7), and C-7 (11x11) in Figure 3-9.

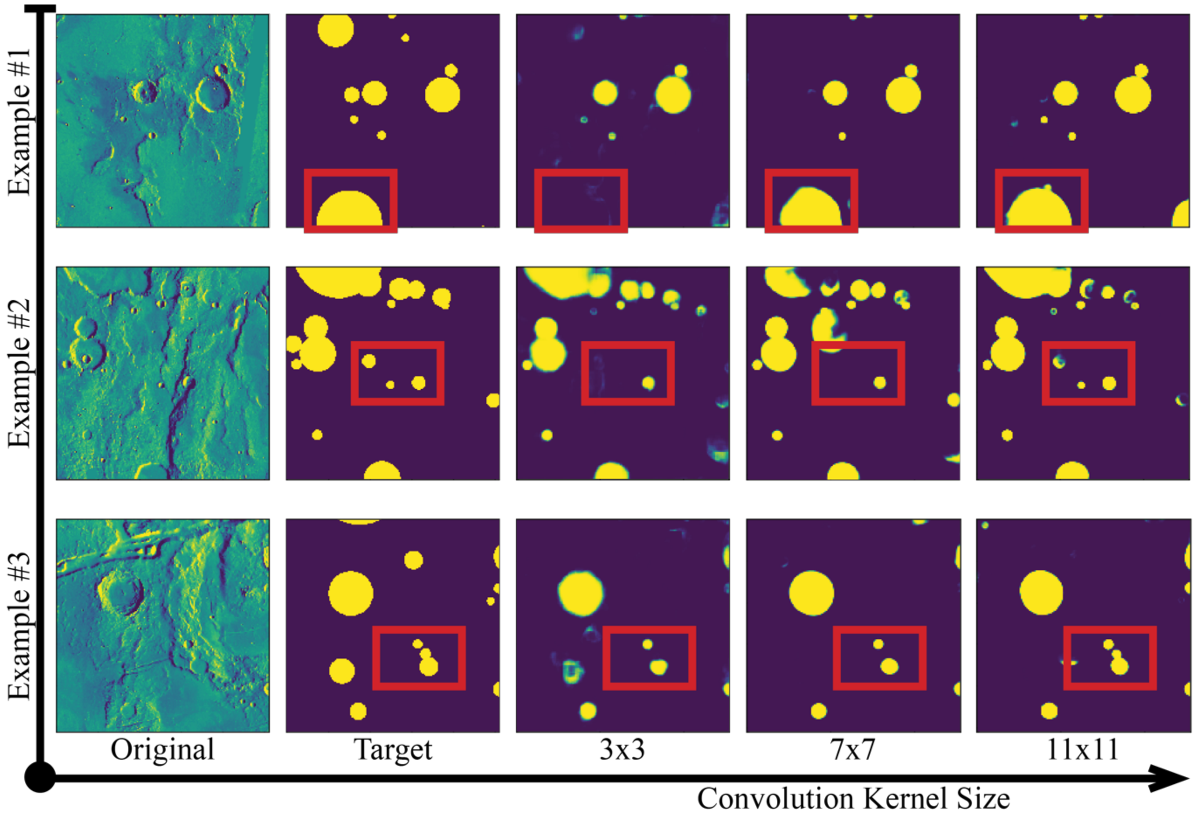


Figure 3-9: Effect of kernel size. Three examples of a 512 x 512 px data image (Original) passed to networks and not used in training, target image for reader reference (Target), result of 3x3 convolution kernel size (C-3), result 7x7 convolution kernel size (C-4), 11x11 convolution size (C-7). Red boxes show craters that are visible only in larger kernel sizes.

3.4.1.2 EDGE TARGETS

This section details the results from the experiments using crater data with two sizes of edge targets. Examples of a target tile, corresponding to one of the THEMIS 30º tiles, is in Figure 3-7. Two edge targets are generated in thicker (E, Figure 3-7c) and thinner (F, half the width of E, Figure 3-7d), options.

The purpose of the edge target testing is to see how a target with the same width for all sizes would perform compared to the two previous types. Edge targets are useful because, unlike filled circle targets, they allow craters within craters to be detected.

In Figure 3-10, one prediction output is shown across four architectures. In the training process, pairs of images like Figure 3-10a & b are passed to the network for training or validation. While some craters are clear to human eye, some, like the largest crater on the right edge, are difficult to spot. Many such examples of faded craters exist in the dataset, and detecting them is difficult. In Figure 3-10, none of the tests could detect that particular crater, although the distinct craters are detected well.

Each image includes the test designation in the bottom left corner. In this example, the larger kernel size (7x7) predicts more circular craters. While the majority of the craters detected show up in all panels, a few appear only in a subset. A specific example of this is the center-most crater in Figure 3-10b (blue box). That crater is detected in only E-6 (Figure 3-10c). To contrast, the small double ring in Figure 3-10b (green box) on the right side is detected in E-3, E-6, and E-4 but is not detected in E-8. This may indicate a combination of trained models may be optimal to detect the faintest, most degraded craters.

For more eccentric elliptical craters, the predicted image is more circular with a larger kernel size (e.g., E-4, E-8). Other architectures with a smaller kernel (e.g., E-3, E-6) result in an elliptical prediction more reflective of the original data. An example is the small elliptical crater at the bottom of the original image in Figure 10a (orange box) and the corresponding regions in Figure 3-10c-f.

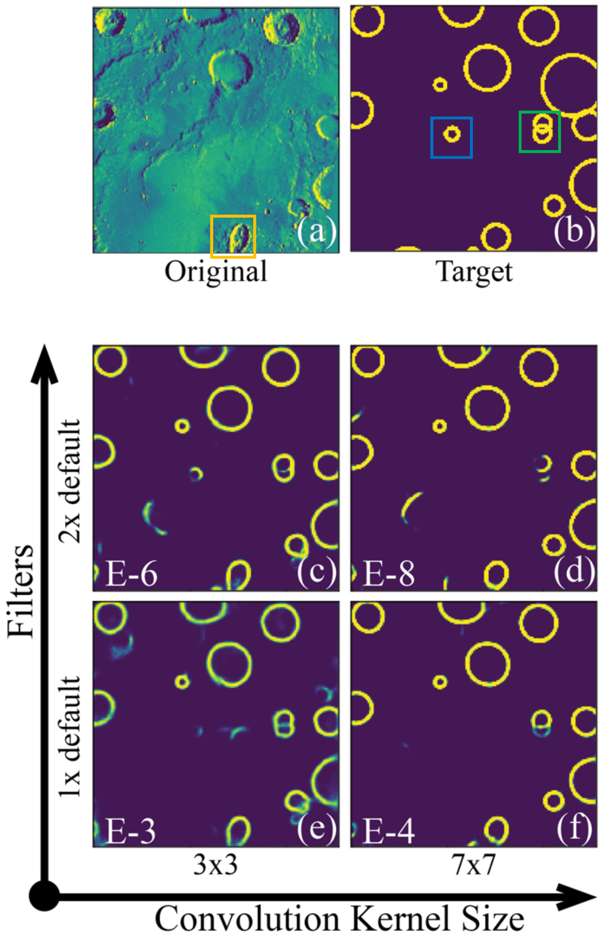


Figure 3-10: Comparison of effects of filters and kernel size with larger edge target. Example of a 512x512 pixel validation sub-tile passed to networks (a), target image for reader reference (b), result of E-3 test: 3x3 convolution kernel size (e), result of E-4 test: 7x7 convolution kernel size (f), result of E-6 test: 3x3 convolution size and double the number of default filters compared to E-3 (c), result of E-8 test: 7x7 convolutional kernel size and double the number of filters compared to E-4 (d). The blue box and green box in (b) are example craters that show up in different subsets of the models.

3.4.1.3 COMPARING TARGETS & ARCHITECTURES

The loss and accuracy measures as a result of training are shown in Table 3-3. These are measured against the same tiles of validation images (not used to train the model). The top four accuracy measurements are bolded in each column for ease of readability. This is the same data shown graphically in Figure 3-8. Loss and accuracy varied across both target types and architectures. As with the plot (Figure 3-8), in Table 3-3, only values within one target type are comparable. Across architectures, more complexity (*-4, *-6, *-7, *-8) generally but not always led to better results.

Table 3-3: Summary of Validation Loss and Accuracy Across All Target Types

Test	Filled Circle (C)		Edge-thick (E)		Edge-thin (F)	
	Loss at 500 ep	Acc at 500 ep	Loss at 500 ep	Acc at 500 ep	Loss at 500 ep	Acc at 500 ep
*-1	0.271	0.9618	0.1008	0.9806	0.0615	0.9875
*-2	0.261	0.9616	0.1064	0.9811	0.0651	**0.9876**
*-3	0.288	0.9617	0.1038	**0.9813**	0.0651	**0.9876**
*-4	0.398	**0.9658**	0.1560	**0.9816**	0.0982	0.9872
*-5	0.182	0.9582	0.0756	0.9810	0.0510	**0.9877**
*-6	0.374	**0.9652**	0.1361	0.9807	0.0849	0.9872
*-7	0.416	**0.9658**	0.1702	**0.9820**	0.1083	0.9875
*-8	0.446	**0.9662**	0.1777	**0.9829**	0.1156	**0.9881**

For the specific application of crater counting, the output of the model (prediction image) needs to be converted to a list of craters consisting of x, y, and radius. However, to generate crater counts, distinct craters need to be detected. For overlapping craters and craters within craters, this is an issue. While the solid circle targets are less precise for this specific application, those initial tests did prove that craters can be detected using Crater U-Net. Additionally, as seen in Figure 3-11, the solid target architecture confidently detected a few craters that the edge target missed, particularly faded craters (Figure 3-11, Example #3). However, the solid target entirely missed craters within craters and distinguished overlapping craters poorly.

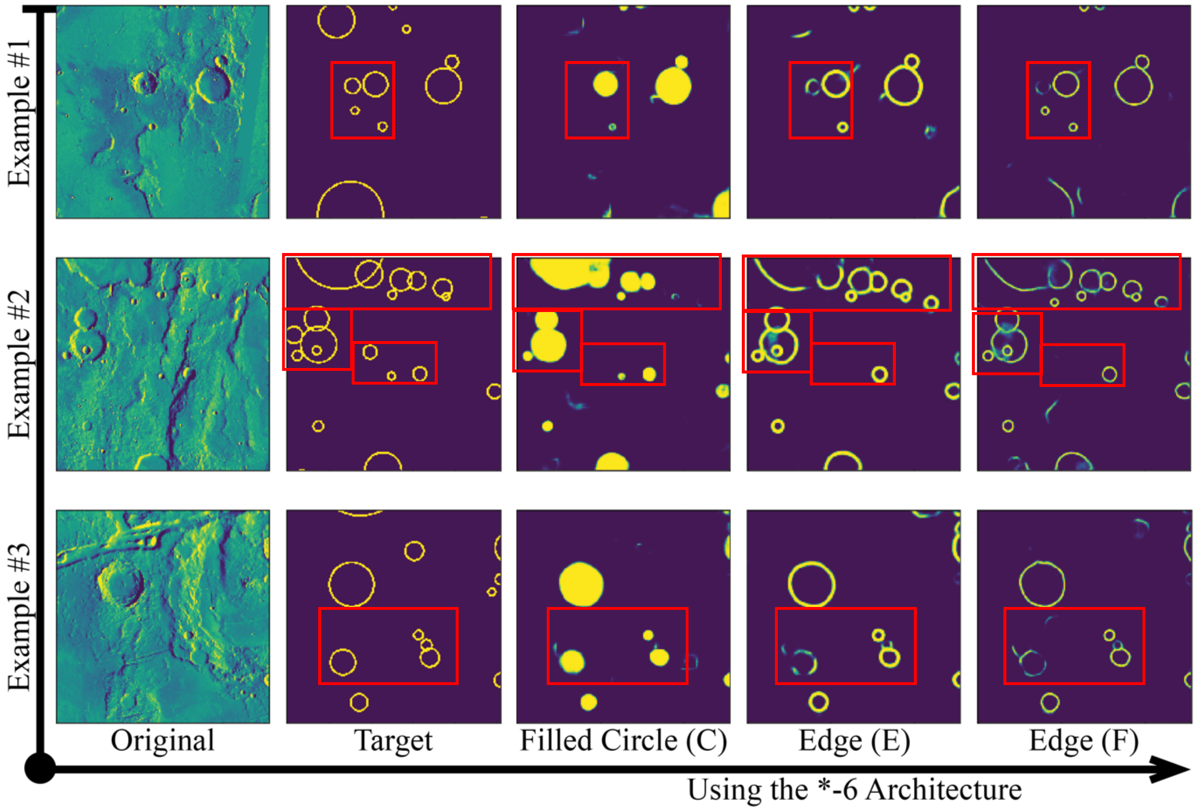


Figure 3-11: Comparison of Solid and Edge Targets. Craters of interest are highlighted in red boxes.

While solid targets are less useful for direct counts, it could be a useful for a human annotator to train models for both types of targets. For the evaluation of crater counts, E and F are selected for further analysis.

3.4.1.4 EVALUATING CRATER COUNTS FOR EDGE TARGET CRATERS

The next step is turning each of the prediction tiles into a list of crater locations and radii for use in crater counting applications. To evaluate the crater counts, the Silburt et al. (Silburt et al., 2019) match algorithm and the scores outlined in Section 3.3.4 are used.

An important parameter to select is the threshold used when making a match. As part of the prediction output of the trained Crater U-Net model, each pixel in the image has a value between 0 and 1. The pixel value indicates the pixel's fit with the crater category; it is a normalized measure of how "crater edge-like" the pixel is. A very low threshold (i.e., 0.1) would mean that even shapes weakly identified as craters would be included in the final count. A high threshold would exclude many craters. To quantify these differences, an architecture is selected and post-processing performed at each of nine different threshold values from 0.1 to 0.9 in steps of 0.1 for each of the two edge targets. Graphing the precision and recall results of E-4 in Figure 3-12 illustrates an example of the trade between the two scores.

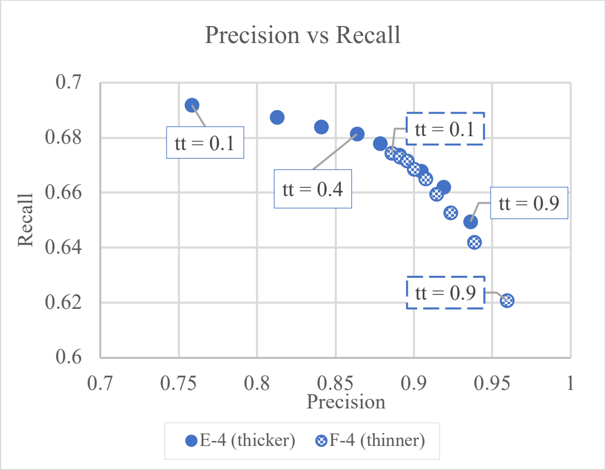


Figure 3-12: Precision vs. Recall for E-4 and F-4 using nine target thresholds (tt) from 0.1 to 0.9.

As a standard for further testing, 0.4 is selected as the target threshold to compromise between precision and recall, while slightly favoring recall. Recall is favored for the crater counting application as it represents the match ratio, the percentage of craters found by the network that are in the human annotations list.

The next important decision is whether the E-type tests (thicker edge) or the F-type tests (thinner edge) are better for the crater counting application. Looking at the recall scores of the test tiles of E-4 vs F-4 in Figure 3-12, the E-type tests collectively have the higher recall score, but the range of values between F-7 (lowest) and F-8 (highest) is less than 0.01. However, the F tests have a lower recall score, and between recall, precision, and F1 metrics, the higher recall is more important to crater counters who want to age date a region. Obtaining an age depends on the identification of true craters, so having a high recall, or match ratio, is important. This quantitatively supports the visual findings displayed in Figure 3-11.

For all of the tests in Figure 3-13, the same six tiles are used in training, and three for validation. Fifteen tiles are used to calculate the recall (Eq. 2), precision (Eq. 1), and F1 scores (Eq. 3) and the chosen target threshold 0.4 is used.

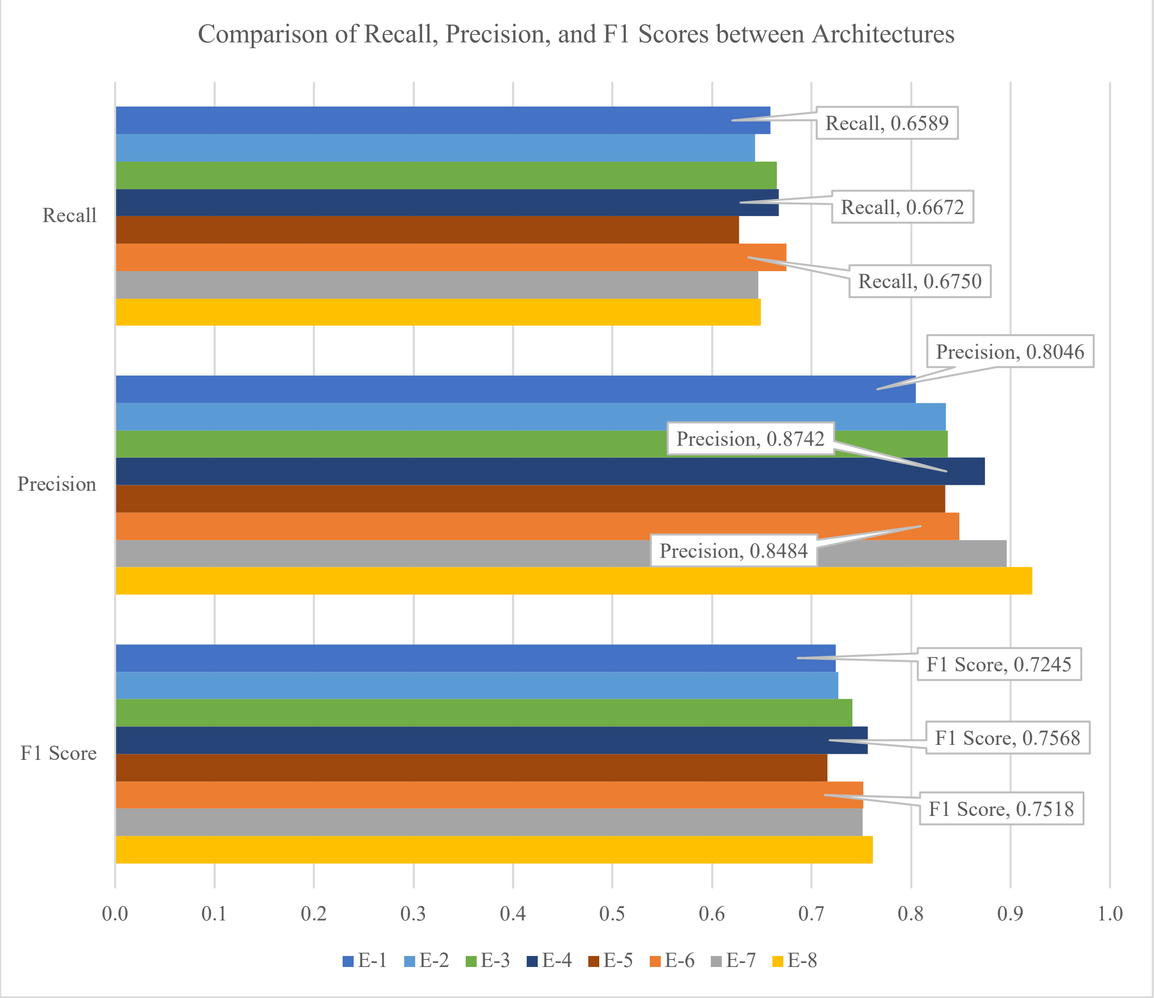


Figure 3-13: Comparison of test data across the eight architectures. All of these models were trained with six tiles for 500 epochs. The three score types are calculated as the overall score across fifteen test tiles with a target threshold of 0.4. (For quick reference: E-* is the thicker, edge target. *-1 through *-3 is 3x3 kernel, default number of filters; *-4 is 7x7 kernel, default number of filters; *-5 is 3x3 kernel, half the default number of filters; *-6 is 3x3 kernel, double filters; *-7 is 11x11 kernel, default filters; *-8 is 7x7 kernel, double filters.)

The two models with the highest recall score are E-4 and E-6 (Figure 3-13). The difference between the two scores is very close, less than 0.01. E-4 is higher in both precision

(+0.026) and F1 (+0.005). For the upcoming series of tests in Section 3.4.2, the volume of tests necessitated choosing a single model type (see Table 3-2 for average training times per epoch), so only E-4 is extensively explored. These differences are overall very small and could vary if other training data is used.

3.4.2 SENSITIVITY TO TRAINING DATA

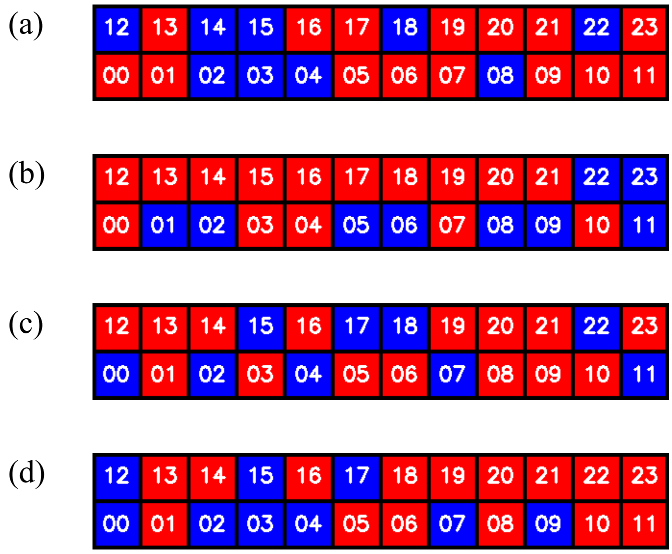
Two natural questions are: how do these results change if different training tiles are selected? how does training with different amounts of data affect the results? Mars' surface is diverse and terrain types and surface age vary. Geologic surface differences may affect crater identification, so these experiments were repeated four times, each run with a distinct mix of training and validation tiles. In addition to choosing different tiles, the authors study the effect of choosing different numbers of tiles for the training set. Tiles selection was made by random assignment. Like in Section 3.4.1.4, all numbers in this section refer to a “per crater” instead of “per pixel” value.

Five combinations of training, validation, and test data ratios were used. The twenty tests are named in Table 3-4. For example, in test 4E, these fifteen tiles are used in training: 23, 10, 20, 14, 11, 1, 16, 5, 22, 13, 21, 19, 18, 8, and 6. (See these geographically placed in Figure 3-14.) This leaves nine tiles in each sequence that can be used to compare the results between tests in the same case. Each “Case” corresponds to the sequence of tiles for that random seed value.

Table 3-4: Test Identifiers

Case (Random Seed)		Number of Tiles Used				
	Training	*3*	*6*	*9*	*12*	*15*
	Validation	*6*	*6*	*6*	*6*	*6*
	Test	*15*	*12*	*9*	*6*	*3*
1		1A	1B	1C	1D	1E
2		2A	2B	2C	2D	2E
3		3A	3B	3C	3D	3E
4		4A	4B	4C	4D	4E

Figure 3-14 shows the tiles used, corresponding to the tiles in Figure 3-2, Figure 3-3, and Figure 3-4 in Section 3.3.1. For example, test 1E uses the random assortment of tiles in Figure 3-14a.



(a)

12	13	14	15	16	17	18	19	20	21	22	23
00	01	02	03	04	05	06	07	08	09	10	11

(b)

12	13	14	15	16	17	18	19	20	21	22	23
00	01	02	03	04	05	06	07	08	09	10	11

(c)

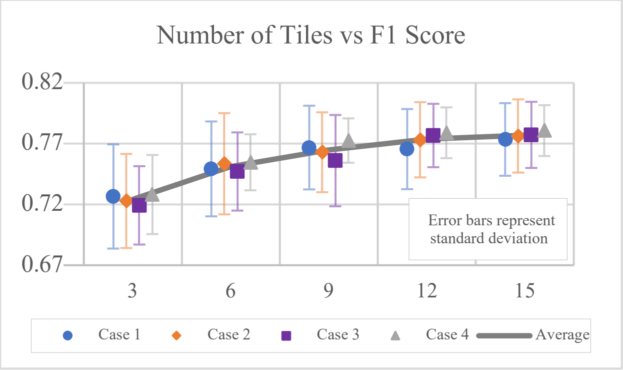
12	13	14	15	16	17	18	19	20	21	22	23
00	01	02	03	04	05	06	07	08	09	10	11

(d)

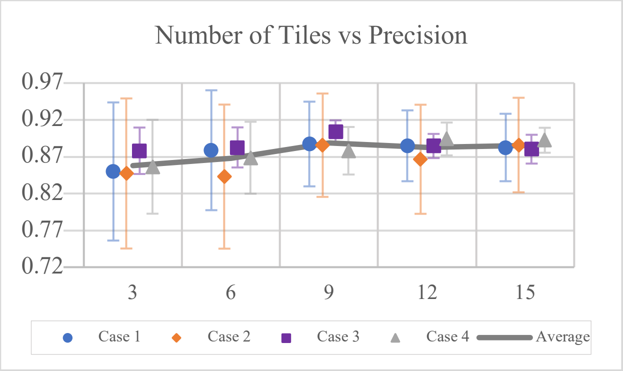
12	13	14	15	16	17	18	19	20	21	22	23
00	01	02	03	04	05	06	07	08	09	10	11

Figure 3-14: Tiles used for comparison test sets. Training tiles are shown in red and validation and test in blue. The red tiles are the 15 training tiles of 1E (a), 2E (b), 3E (c), and 4E (d). The locations of these tiles match those of Figure 3-2-Figure 3-4.

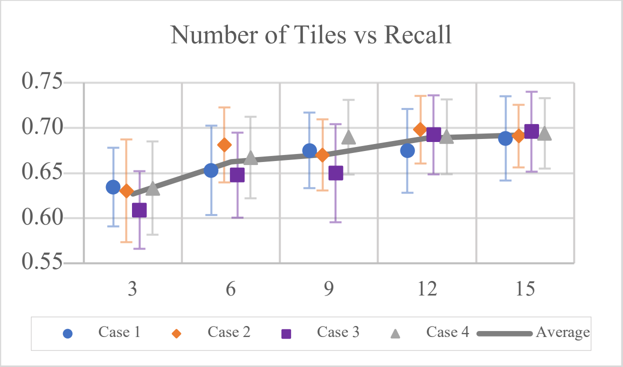
The F1 scores (Eq. 3) in Figure 3-15 are comparing whether the annotated 2-32 km craters were found or not found are on the crater level, not the pixel level.



(a)



(b)

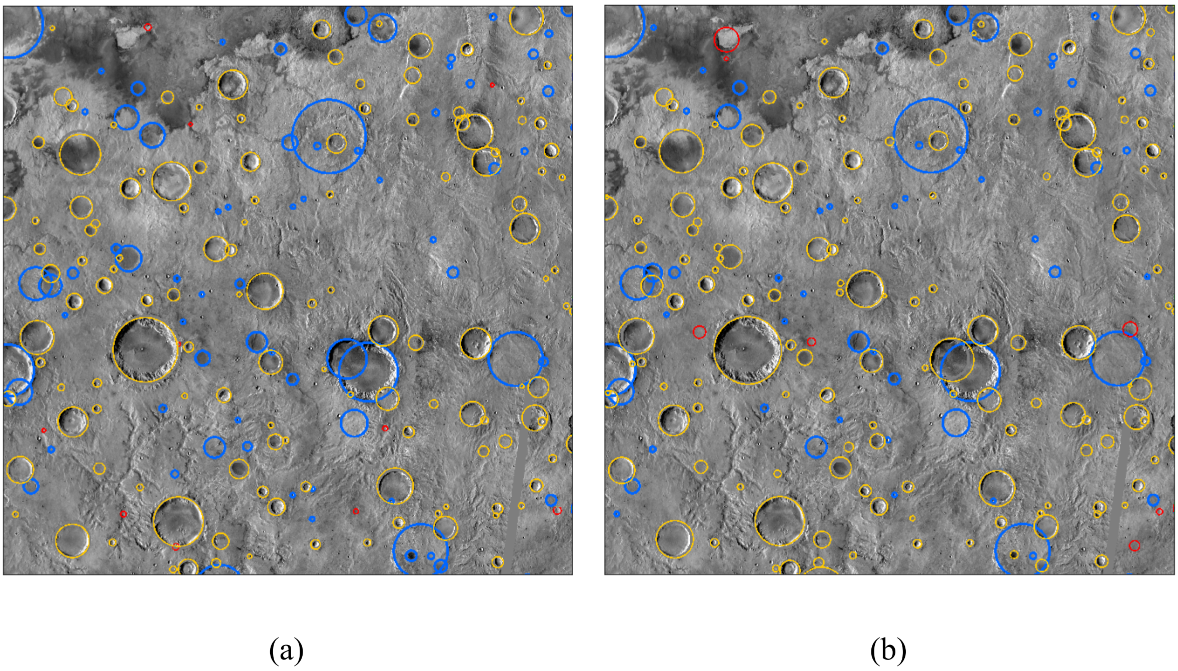


(c)

Figure 3-15: Graphs for the data amount tests (using E-4 architecture, see 3.4.1.4), comparing F1, precision, and recall scores for each of the four cases. Error bars for each point show the standard deviation between the scores in the nine tiles used for testing the model at each point. Each point represents the overall F1, precision, or recall value at that number of training tiles, calculated using the same nine validation and test tiles within the case. The overall average between the point values for the four cases is included to show the general trend. Important note: since the tiles used in training between cases are different, some cases are particularly affected by tough terrain in a particular tile. The values between cases are not directly comparable because they are the average of different tiles. Tiles within one case *are* directly comparable because each point of one particular case uses the same nine tiles.

Figure 3-15 shows that depending on the exact tiles used in training, the F1 score trends upward as more training tiles are used, but there is variability (see standard deviation error bars). Although the F1 score (Eq. 3) and recall (Eq. 2) mostly increased with additional training tiles then started to flatten, the precision (Eq. 1) scores varied more while trending toward increasing. The recall score, representing the match ratio, on average gains 5.9% between 3 and 15 tiles, but those gains are not linear across the four different examples. For an example of visual representation of the difference between the matches in three and fifteen tiles, see Figure 3-16.

Based on the results above, it is beneficial to use at least 9-12 tiles to train this architecture. For the 4E model, the recall score ranged from 65.1% to 76.4% between the tiles with an overall score of 69.4%. The images in Figure 3-16 are taken from the most conservative example with recall 65.1%. To make the strongest case for the robustness of this approach, the same worst-case choice is made for age dating.



(a) (b)

Figure 3-16: This is Tile 00 (Figure 3-14d) using 3 (left image, test 4A) and 15 (right, test 4E) training tiles. Blue is false negatives, (annotated craters); yellow is true positives (matches); and red is false positives (detected by network but not in annotations).

3.4.3 AGE DATING

One of the reasons scientists count craters is to age date a geologic unit. While this work is not a study in age dating, it is worth comparing the age obtained from one of the neural network's crater counts versus the annotation's estimate and ages for the same geological unit obtained by independent crater counters (Platz et al., 2013) (Figure 3-17). A region from the same tile shown in Figure 3-16 is chosen from within the Middle Noachian highland unit (mNh) (Tanaka et al., 2014).

The Craterstats tool (Michael & Neukum, 2010) (Michael, 2008) is used with the Ivanov 2001 Mars production function (Ivanov, 2001) and the Hartmann & Neukum 2001 chronology function (Hartmann & Neukum, 2001), to analyze crater statistics obtained from the 4E model and to compare with RH2012 annotations for the same region. The RH2012 annotations obtain an age of $3.90^{+.023}_{-.028}$ billion years (Ga) and the 4E model obtains an age of $3.87^{+.027}_{-.034}$ Ga. Platz et al. (Platz et al., 2013) dated the mNh unit using crater counts on High Resolution Stereo Camera image data and obtained ages ranging from 3.85 to 3.98 Ga at a nearby location. Not only do the two calculated ages have overlapping error bars, both of these age calculations are within the error bars of P2103 and larger than the minimum, making the results of Figure 3-17 low but very promising. These results demonstrate that this model's crater counts in that area are consistent with both RH2012 and P2013.

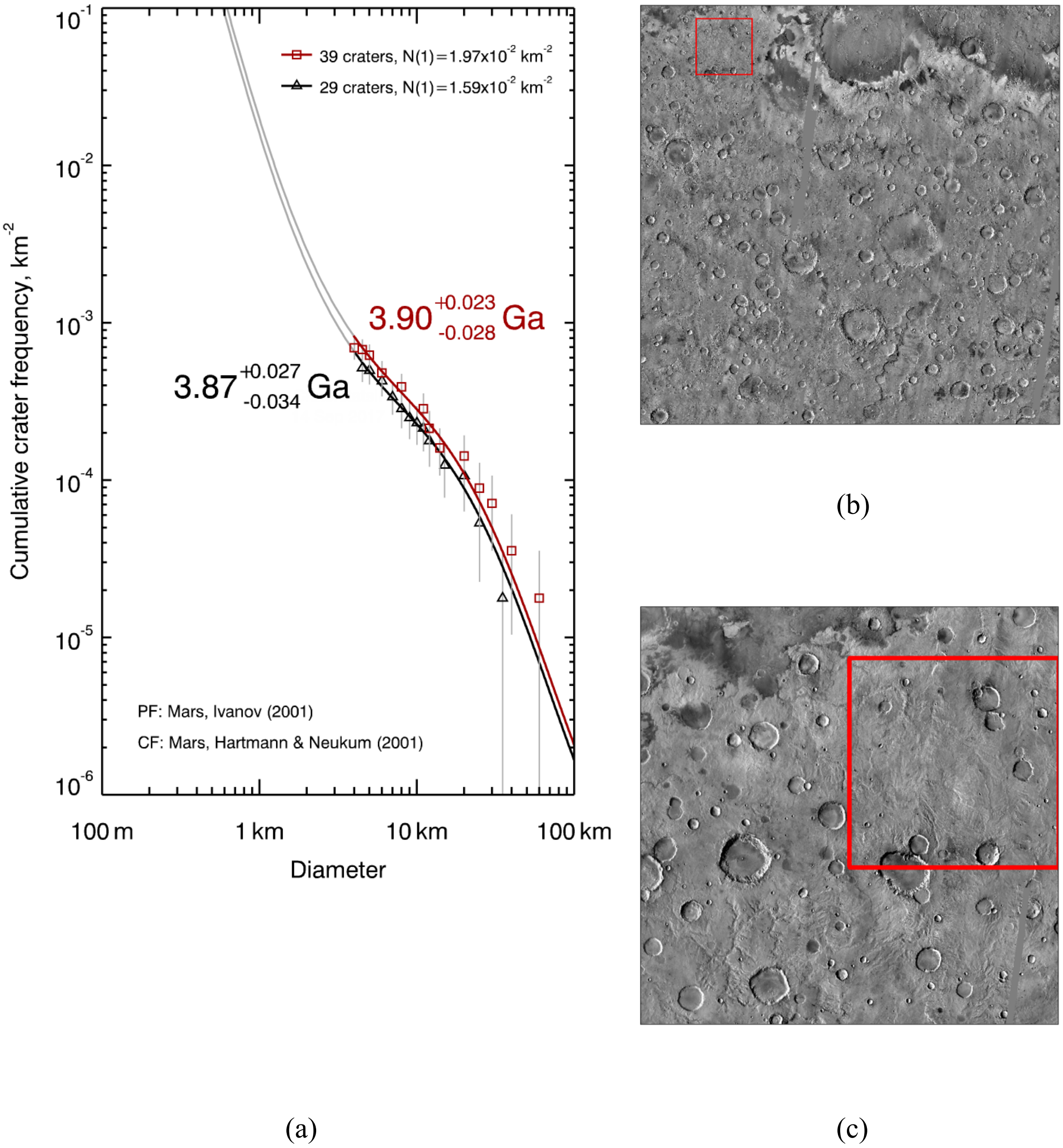


Figure 3-17: Cumulative crater count frequency comparison (a) area contained within the red boxes in Tile 00 (b) and the same box zoomed in (c). The black line is the counts according to the test 4E, trained with the 15 red tiles from Figure 3-14d. This region is taken from one of the tiles not used in training, Tile 00 (shown in blue in Figure 3-14d). The whole tile is pictured in (b). The area is chosen to stay within one geologic unit, mNh, and avoid neighboring units. Sub-figure (c) is the same portion of Tile 00 as in Figure 3-16b.

3.5 DISCUSSION

3.5.1 CRATER U-NET

Crater U-Net is a close cousin of the original U-Net (Ronneberger, Fischer, & Brox, 2015a). The Crater U-Net architecture is deeper which may generate features that draw out information from multiple scales useful for fitting the wide range of crater sizes. The Crater U-Net final output image is the same size as the input image, which helps with localization of the craters in post-processing. Dropout (Srivastava et al., 2014) is used instead of cropping the layers to preserve the edge information. From these experiments, the best architecture of the variations in Section 3.4.1.4 was the one with the 7x7 kernel size (architecture *-4 from Table 3-2), though the 3x3 kernel with double the number of filters (*-6) was not much different. Further research could explore the similar architectures through the lens of tests in 3.4.2 and 3.4.3.

3.5.2 TARGET TYPE & ARCHITECTURE COMPARISON

The target comparison tests (3.4.1) yielded a variety of insights. Edge fixed-width targets are superior in terms of the loss and accuracy, in ease of post processing, and most importantly in crater detection. Edge targets also enable detection of craters within craters.

From the perspective of applications choosing the "best" target and architecture combination is a challenge. Although edge targets had better properties overall, visual inspection of images suggests that solid targets improved detection of some faded craters. Metrics available during the image segmentation training process are imperfect for judging crater count quality and probably not the metrics that a human annotator would design. Crater-count-based metrics like the recall, precision, and F1 score are available after post processing. Scientists have the flexibility to choose which is more important to their application to inform their choice of network.

3.5.3 IMPACTS OF TILE CHOICE AND AMOUNT OF TRAINING DATA

The tests show noticeable gains in the recall and F1 scores for each additional three tiles used in training up to 9 or 12 tiles, after which the differences begin to level off. The geology of Mars may make the amount of data from some tiles more useful than others in training. Tiles with a variety of geological features may be particularly useful.

3.5.4 FUTURE PROSPECTS

Future work could connect the geologic map of Mars (Tanaka et al., 2014) with tiles used during training to predict how good the model would be for other tiles not in the ±30º range. Another consideration is how to record crater predictions that are not in the original annotations, i.e., false positives in this investigation. Some of these may be unlabeled craters, craters above or below the crater size threshold, faded craters, or objects of interest worth investigating further. An expert crater counter could be enlisted to provide an appraisal of the false positives. A few potential craters showed up in multiple models, and these are of particular interest. One such example is in the bottom left (red circle) of both Figure 3-16a and Figure 3-16b.

Ultimately, these lessons could be synthesized into a crater counting system that may include an ensemble of networks. There is a large subset of craters that appeared in every prediction, but some annotated craters showed up in only one model's prediction and an ensemble would allow these to be captured. Training the same architecture with multiple target types may improve effectiveness. With the larger kernel size, solid filled circle targets seem to help detect some particularly faded craters. Even though the solid target is not useful for creating a list of craters, it may be helpful for human annotators to run as a visual suggestion of where craters may be.

3.5.5 POTENTIAL APPLICATIONS

Scientists may be motivated to prioritize precision or recall depending on their application. By a judicious choice of architecture and threshold, the work that is shifted from human annotators to computers can be tailored. For example, a high precision model will have few false positives but many missed craters. In a crater counting application, a human may prefer to add to the computer-provided results by manual annotation. A high recall model will find more craters but may have more false positives. This may be preferred by many crater counters.

Crater maps for hazard avoidance applications may favor emphasizing recall. For a mission considering using these techniques to support entry, descent, and landing of spacecraft on a planetary surface, a low target threshold will detect more potential crater hazards while potentially producing more false positives.

Overall, using these methods in the crater counting pipeline may save time for a crater counter. The architectures are a starting point for training models for new datasets.

3.6 CONCLUSION

The authors detail a successful implementation of Crater U-Net, a convolutional neural network to identify craters on the surface of Mars. In the chosen model (E-4 architecture and 4E tiles for training), untrained regions of craters have a match accuracy of 65-76%, and returns age dating results consistent with human annotations of the same geological units. In addition to exploration of the architecture design, a pre- and post-processing framework is provided, resulting in a cumulative crater frequency plot that is within the error range of a corresponding plot based on human annotations.

This work fits into a larger body of research demonstrating that convolutional neural networks offer an advantageous approach to the challenging and labor-intensive task of analyzing space image data. General approaches to image processing enabled by deep neural networks provide a promising start toward work reduction and eventually automation. The full realization will require continued collaboration with space scientists to design tailored algorithms that fully incorporate deep expertise derived from human annotators.

CHAPTER 4: EXPANDING CRATER DETECTION

4.1 SUMMARY

This section details work related to expanding and using the Crater U-Net described in Chapter 3. Three extension projects are explored: comparison of original U-Net and Crater U-Net (Chapter 3), using results from Chapter 3 to age date nine diverse regions of Mars terrain, and discussion of two machine learning model improvement techniques (ensemble learning and bagging) that could improve the results.

4.2 COMPARISON OF U-NET AND CRATER U-NET

The original U-Net by Ronneberger et al. (Ronneberger, Fischer, & Brox, 2015) segmented the edges of cells in a grayscale image. The roughly round shape and success of the method indicated a potentially promising method for segmenting craters, which also have distinct edges and are often identified from grayscale images. The work by Silburt et al. (Silburt et al., 2019) showing the U-Net was successful at identifying craters from a digital elevation model on the moon was promising, but there was not a guarantee that the method would work to find features for the more complex infrared data used to train Crater U-Net.

Original U-Net differs from the Crater U-Net used in Chapter 3 in several key ways: it is deeper, it does single convolutions at each layer down, and it uses average pooling instead of max pooling. To compare the two on the same crater detection problem, U-Net is re-implemented in Keras.

Original U-Net is adjusted to accept the same 512 by 512 pixel images that Crater U-Net does. The original U-Net was designed for a specific competition where the input image was 572 x 572 px and the output segmentation map is 388 x 388 px (Ronneberger et al., 2015). Those sizes hold no value outside that specific experiment, so a 512 x 512 input and same size output are adopted for this comparison. Additionally, it is a common U-Net variant to use dropout instead of copy and crop (e.g., (Silburt et al., 2019), (Kohl et al., 2018)), so that is explored as well for the following comparison.

The tables below show results of the match ratio (recall score) from testing various target thresholds for different models that are closer to original U-Net. Hardware limits prevented the test of an exact match, but these provide reasonable comparisons. The comparison tests fell into two categories: (1) using another researcher's implementation of the original U-Net against the best Crater U-Net, results in Table 4-1; (2) modifying the Crater U-Net model to resemble the structure of original U-Net as closely as possible but with dropout instead of copy and crop, results in Table 4-2.

Table 4-1: Comparison between U-Net (with copy and crop), Crater U-Net with 3x3, and 7x7 kernel size

Tile	Tile Type	Match ratio (tt=0.3)	Match ratio (tt=0.5)	Match ratio (tt=0.3)	Match ratio (tt=0.5)	Match ratio (tt=0.3)	Match ratio (tt=0.5)
Model		*U-Net (ZZ)*	*U-Net (ZZ)*	*Crater U-Net*	*Crater U-Net*	*Crater U-Net*	*Crater U-Net*
				3x3 kern	*3x3 kern*	*7x7 kern*	*7x7 kern*
Tile 04	Test	0.69282	0.68770	0.71136	0.69992	**0.71293**	0.70584
Tile 15	Test	0.63007	0.62076	0.64338	0.63473	**0.66267**	0.65070
Tile 12	Test	0.64397	0.64131	0.65728	0.64397	**0.66603**	0.65804
Tile 03	Test	0.69241	0.68670	0.70996	0.70118	**0.71610**	0.70513
Tile 09	Test	0.75819	0.75367	0.75819	0.75028	**0.76836**	0.76158
Tile 07	Test	0.65955	0.64650	0.62515	0.61922	**0.66311**	0.65599

U-Net (ZZ) uses a slightly modified version of the model by Zizhao Zhang in a Github repository[1]. This model implements copy and crop, as did the original U-Net.

[1] https://github.com/zizhaozhang/unet-tensorflow-keras/blob/master/model.py

However, since the output image is the same size as the input image for this application, copy and crop is not an effective regularization method. To compare, two models from Chapter 3 that use 3x3 and 7x7 kernel sizes are shown in Table 4-1. Crater U-Net has a higher match ratio in each of the six test tiles, with the 7x7 kernel producing the best result.

In this second comparison, an original-adjacent U-Net (using Dropout at Crater U-net levels and using similar reduced number of filters to be run-able with the test hardware) is compared to tests with Crater U-Net with 3x3 and 7x7 (pixel) kernels. The most striking difference is the amount of training time per epoch: over twice as much training time for the original Dropout U-Net. Another interesting note is the similarity of the validation accuracy between the models and the difference in the match ratio. This is further evidence (seen in Chapter 3) of the machine learning metrics not necessarily reflecting the metrics that matter to crater counters, like recall (match ratio) and precision. It is also interesting that Crater U-Net is overfitting less (judging by the difference between the match ratio for the test vs training tile), but this is not an entirely fair comparison because the regularization strategy is optimized to Crater U-Net and not original U-Net.

Table 4-2: Comparison between dropout U-Net, Crater U-Net with 3x3 kernel size, and Crater U-Net with 7x7 kernel size

Stats at 200 epochs	U-Net (dropout)	Crater U-Net (3x3 kern)	Crater U-Net (7x7 kern)
Time per epoch	**151 - 167 sec**	**66 - 71 sec**	**68 - 73 sec**
Val loss	0.1426	0.0896	0.0905
Val acc	0.9776	0.9773	0.9774
Match ratio, Tile 04 (test)	**0.710962**	**0.704259**	**0.699527**
Match ratio, Tile 05 (training)	0.865377	0.753396	0.762454

Despite the longer training time per epoch which made Crater U-Net better for the rapid architecture testing in Chapter 3, the results with the original-adjacent U-Net do indicate that it is would be a good network for segmenting craters in general and would be a good candidate for ensemble learning with Crater U-Net. (This is discussed more in Section 4.4.)

4.3 EVALUATING AGES PRODUCED BY CRATER U-NET IN MULTIPLE LOCATIONS

In order to evaluate the best Crater U-Net model (7x7 kernel size), nine locations are chosen[2] that span across the three major eras of terrain: Noachian, Hesperian, and Amazonian. These locations can be found in Figure 4-1. The models used for this analysis are those of Chapter 3 with the same randomly selected lists of craters, each trained with 12 of the 24 tiles for 200 epochs. The match is performed with a target threshold of 0.2. Since the locations vary across all the tiles and some models use those tiles in their training, only results from a model that had the relevant tile as validation or test data are reported. (Training tile matches potentially would be artificially high.)

Since the tiles were used as training or test data for different models, it is vital to keep track of which tiles were used in training and use a different one for the age evaluation in Table 4-2. The “model” column indicates which of the models (per Chapter 3) are used. For the “D” models, there are twelve training tiles, which limits the available test tiles by half. The relevant tile’s match ratio (per the model) is included for comparison, but note that the match ratio is for the entire tile, not the rectangular region being used to obtain an age.

[2] The author thanks Ryodo Hemmi and Prof. Hideaki Miyamoto for their assistance choosing appropriate locations. The results of this work are used to create the DIAM files that are loaded into CraterStats, and Ryodo Hemmi used them to obtain the network ages listed in Table 4-3 this section.

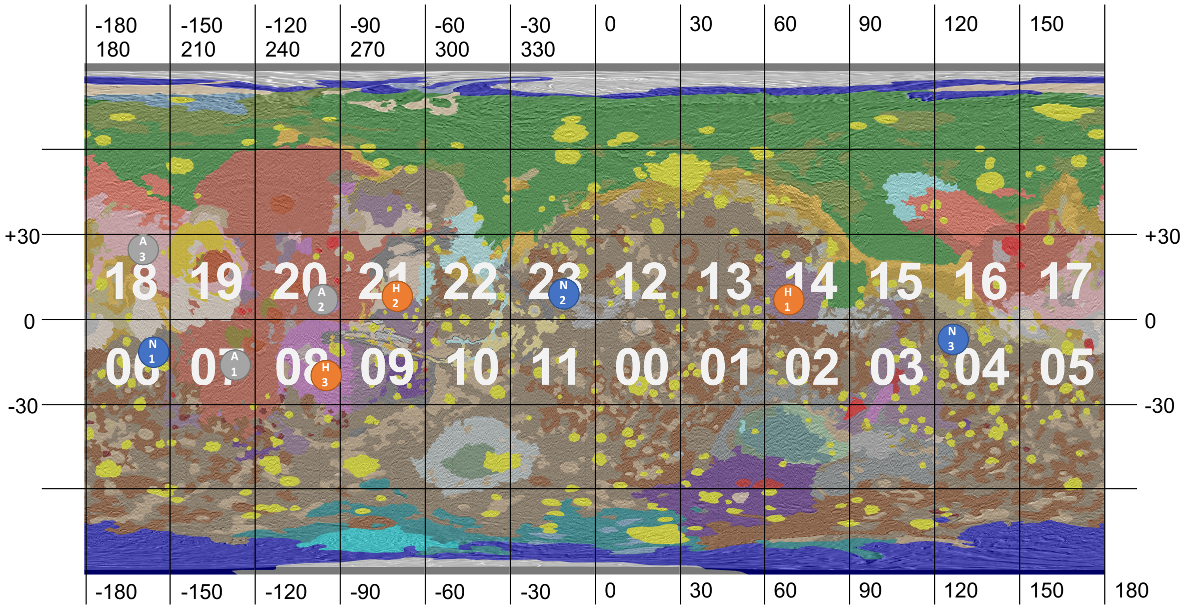


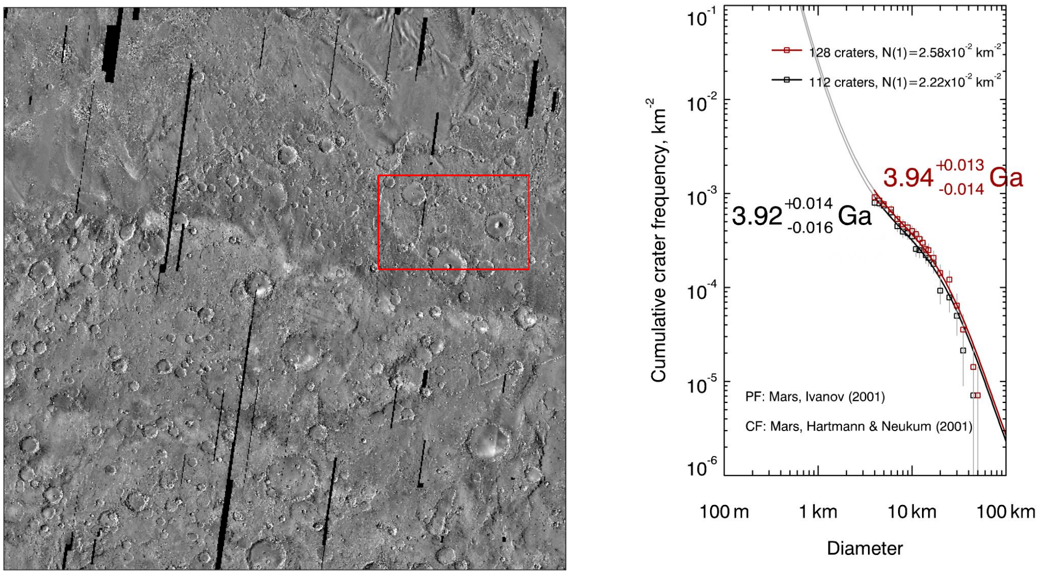
Figure 4-1: Locations of age analysis overlaid on the Tanaka map of geologic units (Tanaka et al., 2014), visualized here as a cylindrical projection. White mosaic tile numbers are included for comparison. These match with the tile numbers in Figure 3-14.

Table 4-3: Locations and ages obtained using Crater U-Net

Name	Coordinates Upper left	Tile	Model	Tile's Match Ratio	Network Age	RH2012 Age
Noachian1	-160E, -9N	06	4D	0.66347	$3.92^{+0.14}_{-0.16}$	$3.94^{+0.13}_{-0.14}$
Noachian2	-15E, +11N	23	2D	0.68571	$3.86^{+0.019}_{-0.022}$	$3.88^{+0.018}_{-0.020}$
Noachian3	+121E, -4N	04	3D	0.73028	$3.92^{+0.014}_{-0.016}$	$3.95^{+0.013}_{-0.015}$
Hesperian1	+67E, +12N	14	1D	0.60081	$3.72^{+0.030}_{-0.037}$	$3.68^{+0.033}_{-0.043}$
Hesperian2	-73.5E, +12N	21	2D	0.78297	$3.76^{+0.026}_{-0.031}$	$3.73^{+0.027}_{-0.033}$
Hesperian3	-97E, -15N	08	2D	0.72139	$3.44^{+0.086}_{-0.19}$	$3.40^{+0.096}_{-0.24}$
Amazonian1	-135E, -11N	07	3D	0.68565	$2.20^{+0.37}_{-0.37}$	$1.54^{+0.30}_{-0.30}$
Amazonian2	-104E, +19N	20	3D	0.79699	$2.51^{+0.37}_{-0.39}$	$1.71^{+0.31}_{-0.31}$
Amazonian3	-170E, +30N	18	3D	0.62681	$2.32^{+0.55}_{-0.57}$	$2.26^{+0.50}_{-0.51}$

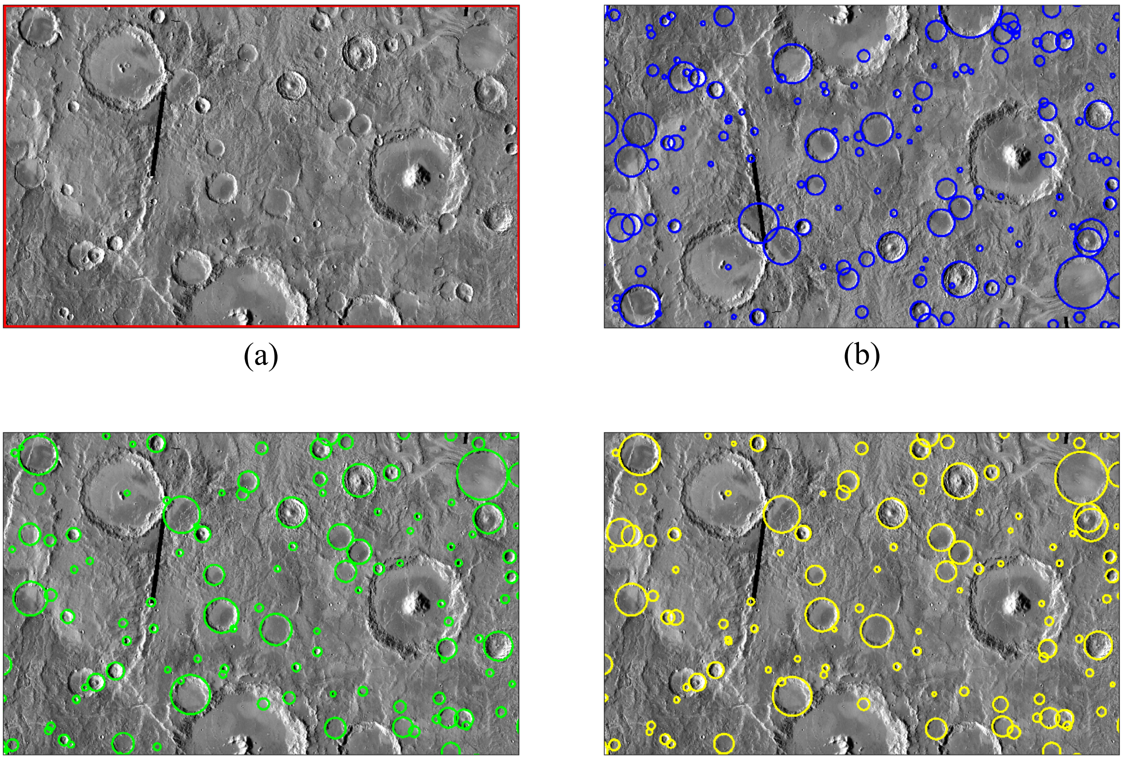
To provide more context for the ages shared in Table 4-3 above, Figure 4-1 shows the locations of the tiles, Figure 4-2 shows the precise location of Noachian1 in Tile 06 with the cumulative crater frequency graph (indicating the obtained ages), and Figure 4-3 shows a zoomed in view of the region with the annotation, network prediction, and matches highlighted. Figure 4-4 and Figure 4-5 show the same for Hesperian2. Figure 4-6 and Figure 4-7 show the results for Amazonian3.

The Amazonian terrain shows noticeably different ages despite being relatively similar in the tile's match ratio. First, the tile match ratios are for the whole 30ºx30º tile and therefore are not necessarily representative of the terrain in a smaller ~5º box that the craters are taken from at the location to obtain ages. Second, Amazonian terrain is the newest part of Mars and has the fewest craters. For age dating, smaller craters than the 2-32 km radius (4-64 km diameter) are typically used to age date this region. Craters that are 1 km in diameter are considered better here, so these age tests could be repeated with higher resolution data to get more representative results. Interestingly, the model seems to have identified **more** craters than were annotated by RH2012.



(a) (b)

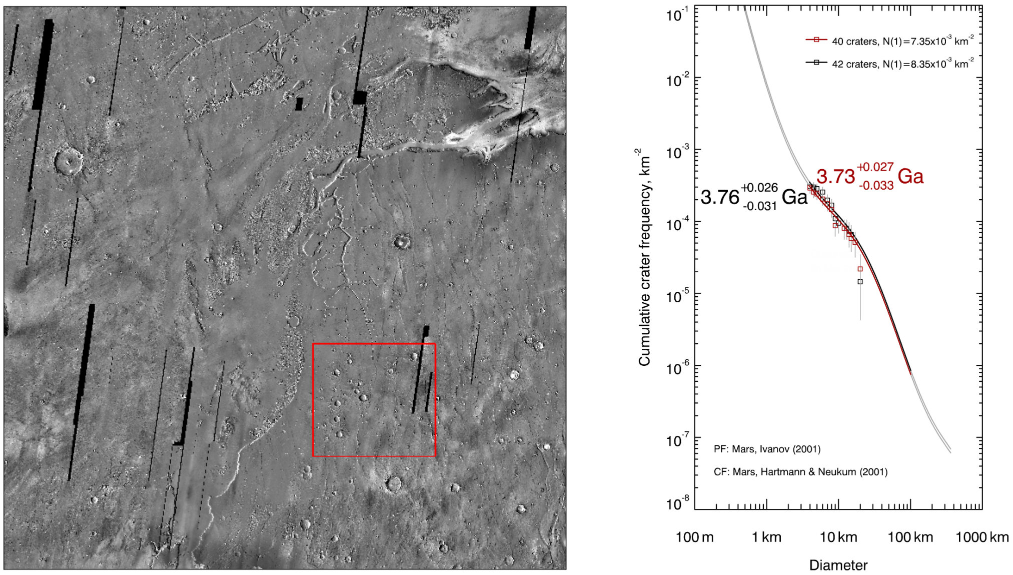
Figure 4-2: (a) THEMIS data of Tile 06 with Noachian1 boxed in red. (b) The age obtained for this region. The black line represents the output of the network prediction; the red line represents the craters annotated by RH2012.



(a) (b)

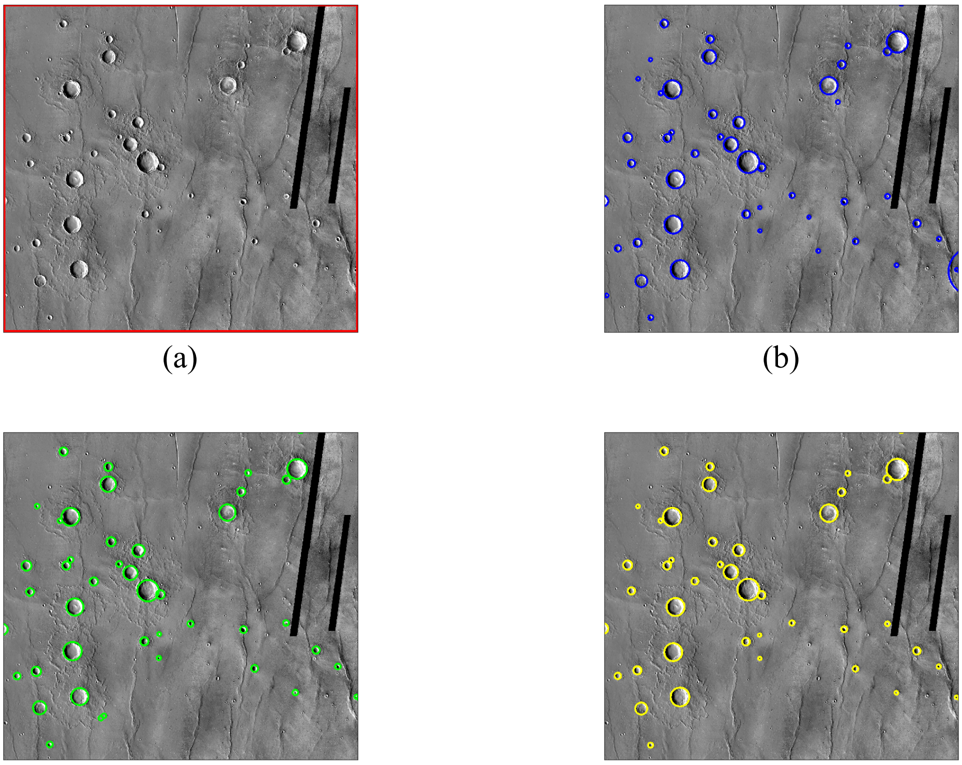
(c) (d)

Figure 4-3: Noachian1 location pictured as (a) original data, (b) RH2012 annotations (within 2-32 km radius range), (c) network predictions (within range), and (d) matches highlighted.



(a) (b)

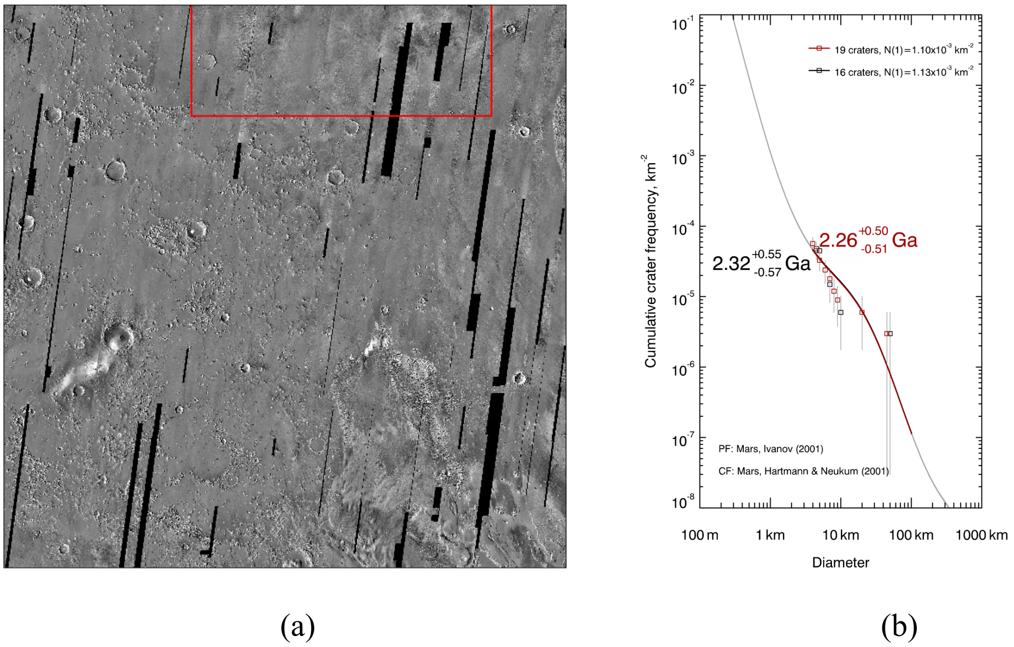
Figure 4-4: (a) THEMIS data of Tile 21 with Hesperian2 boxed in red. (b) The age obtained for this region. The black line represents the output of the network prediction; the red line represents the craters annotated by RH2012.



(a) (b)

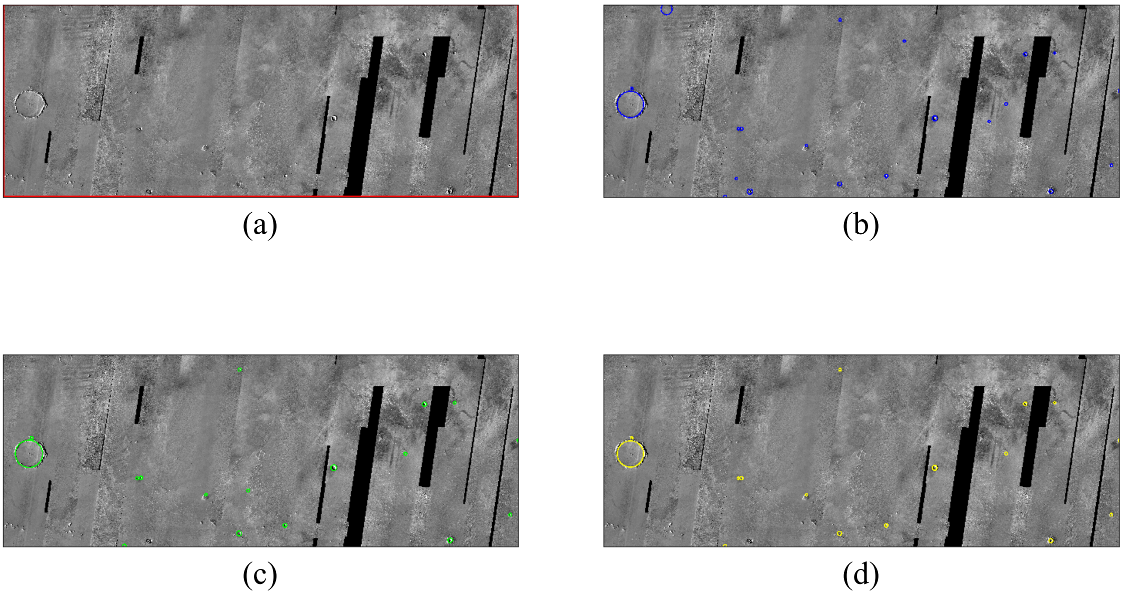
(c) (d)

Figure 4-5: Hesperian2 location pictured as (a) original data, (b) RH2012 annotations (within 2-32 km radius range), (c) network predictions (within range), and (d) matches highlighted.



(a) (b)

Figure 4-6: (a) THEMIS data of Tile 18 with Amazonian3 boxed in red. (b) The age obtained for this region. The black line represents the output of the network prediction; the red line represents the craters annotated by RH2012.



(a) (b)

(c) (d)

Figure 4-7: Amazonian3 location pictured as (a) original data, (b) RH2012 annotations (within 2-32 km radius range), (c) network predictions (within range), and (d) matches highlighted.

4.4 ENSEMBLE LEARNING & MODEL BAGGING

Ensemble learning is a method used to combine multiple trained models to improve the overall classification score. This general methodology is effective on a wide variety of machine learning tasks and has become very popular in machine learning competitions, such as those run by Kaggle. The core idea is that if models that focus on different detections are combined, the combination is stronger than either of the models alone. Models can either be averaged directly, as in:

```
pred1 = model1.predict(X)
pred2 = model2.predict(X)
pred3 = model3.predict(X)
pred4 = model4.predict(X)
pred5 = model5.predict(X)
num_models = 5
pred_final = (pred1 + pred2 + pred3 + pred4 + pred5) /
      num_models
```

Or models can be weighted to indicate stronger detection potential from some models:

```
pred_final = 0.5 * pred1 + 0.3 * pred2 + 0.2 * pred3
```

Regardless of the exact distribution, the goal is to use diversity to let the biases cancel each other out. Chollet summarizes, "you should ensemble models that are *as good as possible* while being as *different as possible*" (Chollet, 2017).

To contrast, bagging refers to using the same model architecture multiple times but changing the data used in training. For example, implementing a bagging method could consist of (1) split the data into five pieces; (2) train on four and reserve one for validation; (3) repeat so each piece is the validation, creating five distinct models; (4) make an ensemble by averaging the five models.

4.4.1 MODEL COMBINATIONS FOR CRATER COUNTING

For the work presented in Chapter 3, all of the craters are known through the annotations. An interesting problem to consider is how to detect smaller craters than were detectable by the first model. To test ensemble learning for crater counting, a few tests are envisioned:

1. Ensemble method: averaging a Crater U-Net with a dropout U-Net (similar to original U-Net but using dropout instead of cropping for regularization)
2. Bagging method: training with overlapping sections of the data (leaving the remaining for validation) — repeat six[3] times, average.

Exploration into the nuances and optimization of this is future work.

4.4.2 OPPORTUNITY WITH THEMIS-593

The NASA Mars Odyssey mission has a visible and infrared camera called Thermal Emission Imaging System (THEMIS) that has produced global datasets of the red planet. These images continue to be valuable in many facets of research including landing site selection, crater counting, and understanding the formation of Mars. In the infrared, both daytime and nighttime images are published. The THEMIS Daytime IR camera is published at two resolutions:

1. THEMIS-256, 256 pixels per degree (PPD) and 231.55 meters per pixel (MPP)
2. THEMIS-593, 593 PPD and 99.7 MPP

The THEMIS-256 data has some advantageous properties in terms of the size of each tile. Each tile is 30º by 30º, 7680x7680 px. When the crater counting pipeline splits the large

[3] Six is chosen because it is a factor of 24, the number of data tiles (7680 x 7680 px) used from THEMIS-256.

image into smaller 512 x 512 boxes, this makes for a perfect 15 by 15 box grid. A tile of THEMIS-593, however, is 35565 x 17783 px. For the tiles near the equator (±30º latitude), each is 30º latitude (Y-direction) by 60º longitude (X- direction). An important decision needs to be made regarding how to treat the differently-sized image within the pipeline: what is the best way to split the new image. Since this discussion may be useful for adapting this code or method to future datasets, it is included here.

The first option is to use the image exactly as it is. The code will start from its 0,0 point (top right corner of the image) and slice 512 x 512 chunks. However, the 0º by 0º point of the tile is actually the bottom left corner. This would cause a mismatch when matching the latitude and longitude to the pixel values later. (This can be accommodated for by adding the offset pixels back, but must be tracked.) The second option is to resize or crop the image to something that is a good multiple of 512. The closest multiple of 512 to the original size (35565 x 17783 px) is 34816 x 17408 px, which contains 68 boxes (of 512 x 512 px) across and 34 boxes down. This is relatively close to the original size, but a little data is lost, and it relies on choosing a good image resizing algorithm. If cropping, that data is lost entirely. The third option is to add a border of pixels to the top and right sides of the image so that the pixels of the image is a multiple 512. Here, adding 275 pixels to the top edge and 137 pixels to the right edge accomplishes this, giving 70 boxes (of 512 x 512 px) across and 35 boxes down. Since we are evaluating an image, not training, adding the buffer seems like the best option in this situation. We can maintain the bottom left origin point and calculate the final relative longitude and latitudes using that point as the start.

One potential use for this is to use a model, bagged models, or an ensemble of models trained on THEMIS256, and generate craters lists for data from THEMIS-593.

4.5 CRATER DETECTION CODE PACKAGE

These extensions represent only a fraction of the tests and studies that could be done. To facilitate further exploration by other researchers, the code is published. The crater detection code package contains useful functions, classes, and Python Jupyter notebooks demonstrating their use.

The code is published at https://github.com/ddelatte/.

4.6 WHAT'S NEXT?

The extensions enumerate a sample of other experiments that could be conducted using this research. The results of the original U-Net comparison show the complexity in determining a "best" network. Depending on how "original" is defined (and limited by hardware realities), the networks performed similarly in some metrics (i.e., accuracy) and vastly different in others (i.e., time). A promising future direction is optimizing the use of both networks in an ensemble to capture more craters.

CHAPTER 5: CRATER COUNTING FRAMEWORK

5.1 MOTIVATION

A baseline framework anchors the discussion and comparison of crater detection algorithms and methods. Depending on the needs of a researcher, choosing to hand count or use a particular technique will be impacted by the availability of the tool, its ease of use, and the similarity between the tool and the new application. With an increase in collaborations between machine learning researchers and planetary geologists, having a central frame of reference aids in each's understanding of the problem. Between the two groups, there can be misunderstanding regarding problems that might be considered simple or very difficult in each field. For example, a machine learning researcher may be confident in a CNN based technique's ability to create features related to craters through training, while a planetary geologist used to hand crafting crater features may initially be skeptical that this step is automated.

In order to anchor the discussion, the key interfaces are identified, a few CDA pipelines are enumerated, and the existing hand counting pipeline is examined. Then, the improved pipeline is presented and the merits of a single framework are discussed.

5.2 OBJECT PROCESS METHODOLOGY

Object Process Methodology (OPM) is a systems architecture diagram used to document the interaction of "objects" and "processes" in a system (Dori, 2011; 2016). The method is used here to represent the crater counting processes and framework. "Objects" are parts of the system which are input, created, consumed, or changed by a process. In this chapter's diagrams, objects are designated with a rectangle and arrows indicating whether they are inputs or outputs. A "process," designated with a circle, is something that transforms one or more objects. An "instrument" is an object that can be used by a process but is not changed during the processes; this is indicated as a box with a closed circle (instead of an arrow) to a process. An “agent” is a person that provides input to a process and is designated with an open circle (instead of an arrow). When the same object is being used by a sequence of processes, an “invocation” is shown, represented by a broken arrow.

Demystifying the “Black Box”:

- **Obtain data** (process): A lot is contained in this bubble: the satellite downlink and transfer by which scientists receive usable data from other planets. This is assumed to be at the level of an image file and outputs DATA.
- **Data** (object): This DATA object is the image file obtained from the planetary science mission. It could be a digital elevation model, infrared, or visible image file.
- **Formating data** (process): Formatting the DATA object can be a simple or complex process depending on the dataset. For THEMIS data for example, it is possible to directly download a PNG file. For other datasets, the NASA Planetary Image format must be converted to a tiff or png for processing by a machine learning algorithm.
- **Pre-process** (process): This pre-processing is what gets the DATA (which has been passed through the pipeline) into the format necessary for the machine learning algorithm. With the Crater U-Net experiment, this consisted only of filling in black pixels with the average gray of the tile, but for other pipelines could be extensive. For some computer vision methods, this has included exaggerating the image to enable edge detectors. For classification machine learning methods, this is the step that identifies the regions of interest that will be sorted into crater and non-crater by a machine learning classifier.
- **Find craters** (process): This step is done manually for hand counting craters or automatically with a trained machine learning model. For hand counting, this can take

days, weeks, or months depending on the amount of data. For a trained machine learning model, this takes minutes or hours.

- **Make list** (process): This step may be done by a program like JMARS (used to aid crater counters), may be an output of a machine learning algorithm (like Faster R-CNN), or may involve using a computer vision technique on a segmented image (like with Crater U-Net).
- **Crater List** (object): This is the list of craters found in the images provided. It should be in either x, y, radius (diameter) with known latitude and longitude bounds or directly in latitude, longitude, radius (diameter).
- **Format list** (process): This may happen automatically if using a software tool like JMARS to count the craters (users can select export options) or may require separate code to turn a CRATER LIST into the file type necessary for the software used to obtain an age.
- **Load list** (process): Here, the generated list is loaded into a tool that aids in obtaining an age.

5.3 PIPELINE FOR HAND COUNTING: BASELINE FRAMEWORK

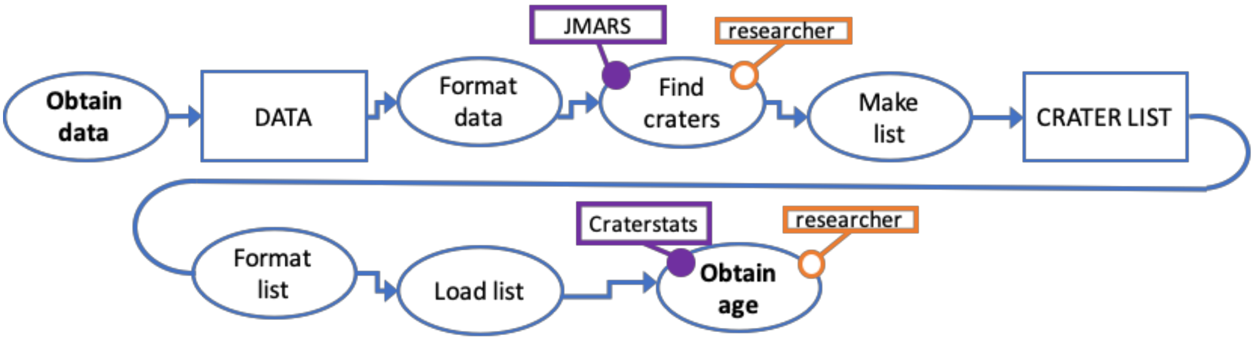


Figure 5-1: Object Process Methodology diagram for the hand counting craters

5.4 IMPROVING THE PIPELINE WITH MACHINE LEARNING

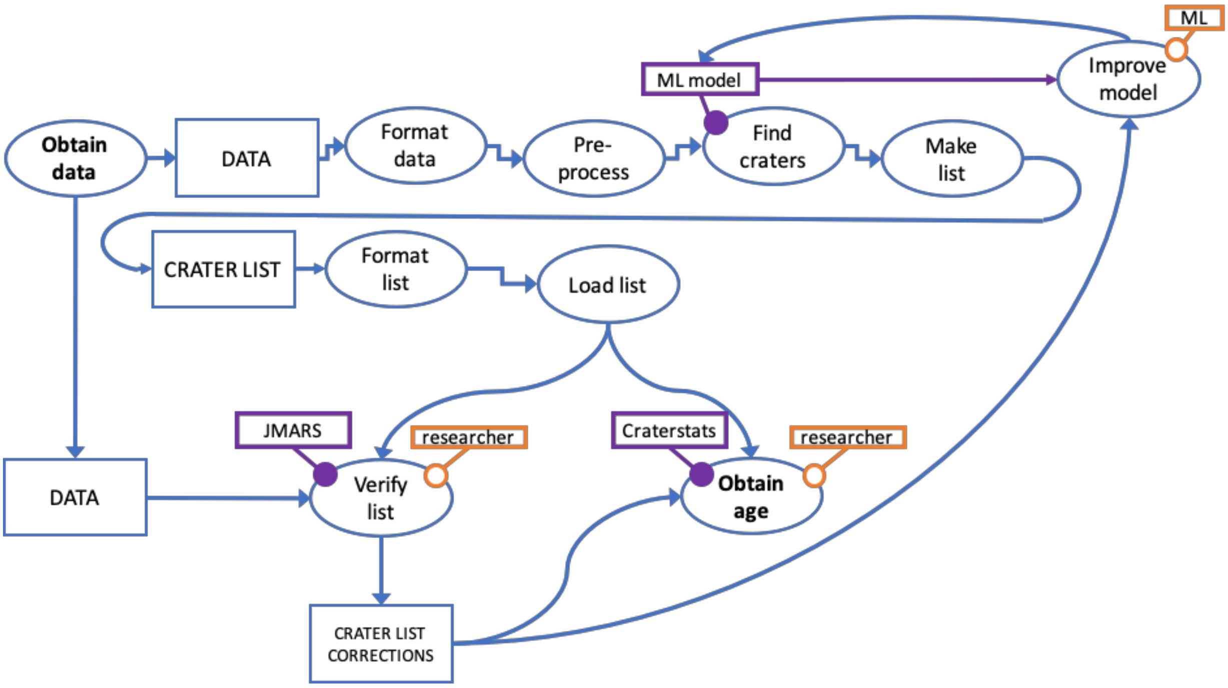


Figure 5-2: Object Process Methodology diagram for the crater counting process with segmentation machine learning included

5.5 ONE FRAMEWORK?

Although it is tempting to try to fit crater counting in a single framework, the reality is more complex. Iterations and ensemble learning with many methods prevent a single simple diagram from being the gold standard. Since some algorithms specialize in a particular type of terrain, size, or data type, the needs of a particular researcher may necessitate using several to complete her work. Despite this nuance, understanding the way the data flows through this framework is important especially for new researchers entering the field so they can make process improvements and avoid inefficiencies.

Even if one framework is not possible, there are a family of machine learning techniques that do fit in the framework described here. For these algorithms that fit, a set of regulations can be made that could organize these similar methods in such a way that they can be combined into a single software that could be used by crater counters to make their crater list for their region of interest.

CHAPTER 6: DISCUSSION

6.1 SUMMARY

This chapter details the significance of various threads throughout this work and notes some of the outstanding challenges and potential collaborations of crater detection research. First, U-Net and Crater U-Net are compared and the method's significance to crater counting and other research is explored. Second, extensions of this work to other applications and fields are considered. Finally, the significance of this thesis's work applied to crater counting is examined, best practices are shared, and high impact next steps are listed.

6.2 EXTERNAL APPLICATIONS

There are numerous extensions of this research, including: using results to analyze chain craters and secondary craters, spacecraft localization and navigation, and expanding to other types of targets like boulders.

Current research allows for detecting secondary craters via proximity algorithms, and having clean output of crater locations would be an asset for such research. Secondary craters are typically not included in the official tally of craters but their presence is somewhat accounted for within the age dating formulas.

Given the success of the basic U-Net architecture on the crater segmentation problem, future work in the domain could include implementing multiple classes that indicate the level of crater degradation, detection of other geological features (boulders, dunes, aeolian ridges, or dry river beds have been of interest to other researchers (Palafox, Hamilton, Scheidt, & Alvarez, 2017)), or trying U-Net on visible images, or comparing the performance on each type of data (digital elevation, infrared, and visible light). Outside the crater counting domain, problems of interest could include detecting objects with night vision and detecting camouflage animals or objects.

Another future application is spacecraft localization and navigation. An algorithm can be used to detect known distinct craters and their pattern could be matched to an expected pattern to aid in landing. This has been considered (Lee et al., 2010) and while presently machine learning algorithms are not robust enough to be the primary instrument, adding a similar capability on upcoming missions would be valuable. There is clear interest in such technology with missions like JAXA's Smart Lander for Investigating Moon (SLIM) mission, which is working on pin-point landing technology (JAXA, n.d.). Should Mars landings become frequent in the future, this could be a good way to localize on the surface, especially if guiding markers were created to aid in the spacecraft navigation.

6.3 COMPARING CRATER U-NET AND ORIGINAL U-NET

Section 4.2 discusses the actual numerical comparison between the original U-Net and Crater U-Net. One of the challenges in machine learning is maximizing training output with other parameters. The two biggest differences between the two is (1) Crater U-Net trains much faster for comparable performance to original U-Net and (2) can be Crater U-Net can be trained on the hardware available. These two differences were crucial in the implementation of this research. Since the main point of Chapter 3 was to try as many architectures as possible (and sometimes the same repeatedly), being able to iterate quickly without crashing the code is of primary importance.

As discussed in Section 4.4, ensemble learning and bagging have much potential for improving the results and finding additional craters. Optimizing this process would itself be another project and one worth studying further.

6.4 NEED FOR COMMUNITY AGREEMENT

The community's agreement on two topics is vital to efficiently and rapidly explore the possibilities of machine learning for crater detection and age dating:

1. A baseline reference dataset must be established. This is not a trivial feat. As found in previous research, up to 45% difference (Robbins et al., 2014) can be found between crater counters labeling the same region.
2. Tools are needed to rapidly test and compare models.

The reference dataset provides much needed continuity and comparison between papers. Currently, researchers are testing on a wide variety of ranges, pixel sizes, and actual sizes of craters, and this is crucial information to comparing methods. Results of algorithms on elevation (DEM) data with large craters (5 km+) are not directly comparable to results of based on visual data that includes smaller crater sizes (1-10 km). To begin to compare datasets, instead of talking about the meter or kilometer sizes, the relative pixels should also be discussed. Accordingly, the reference dataset should have the following characteristics:

- Easily accessible, well formatted data in an image format that can be read by any computer (recommended: PNG)
- Cover a wide range of terrain on the planetary body. Cylindrical projections stretch out the appearance of roughly circular craters, so keeping the reference to ±30º limits issues
- Open source reference code that describes how to run the data and analyze it, preferably in multiple languages and machine learning frameworks, but at a minimum in Python using Keras/Tensorflow in a Jupyter notebook

Ideally, the reference dataset would also have the following characteristics:

- Image and annotation data from multiple planetary bodies, starting with Mars and the Moon

This is not a perfect solution, but it would vastly improve the current situation. Currently, machine learning researchers interested in the field (who lack a planetary geologist collaborator) must sift through multitudes of previous research and dataset types. Those outside the field are unfamiliar with standards like NASA's Planetary Data System (PDS) format.

6.4.1 COMMUNITY NEEDS TOOLS & BASELINE

Comparing the differences between algorithms remains a challenge due to the lack of an established baseline. Salamuniccar et al. (Salamunićcar & Lončarić, 2008) proposed an open framework, but key elements have yet to be broadly adopted. However, several dataset combinations are better studied in crater counting and machine learning research. These regions are not necessarily better studied because of scientific value, but rather ease of use, especially if one researcher has published an easily accessible dataset. In a few cases (DeLatte, Crites, Guttenberg, Tasker, & Yairi, 2019; Silburt et al., 2019), the ±30º region was selected because the circular form of craters is preserved in the cylindrical projection. For example, (Benedix et al., 2018; DeLatte et al., 2019; DeLatte, Crites, Guttenberg, Tasker, & Yairi, 2018a; Norman et al., 2018) use THEMIS images (NASA Mars Odyssey/THEMIS Team, 2006) and RH2012 annotations (Robbins & Hynek, 2012) are used with THEMIS in (DeLatte et al., 2019; DeLatte, Crites, Guttenberg, Tasker, & Yairi, 2018a). The Mars Nanedi Valles region, imaged by the Mars Express High Resolution Stereo Camera (HRSC) nadir panchromatic (#h0905_0000, 12.5 meters per pixel) (Bandeira, Ding, & Stepinski, 2010) is studied by numerous research groups (Bandeira et al., 2010; Cohen, Lo, Lu, & Ding, 2016; DeLatte, Crites, Guttenberg, Tasker, & Yairi, 2018b; Ding et al., 2011; Urbach & Stepinski, 2009). On the Moon, annotations by Head et al. (>20 km) (Head et al., 2010) and Povilaitis et al. (5-20 km) (Povilaitis et al., 2018) are used for machine learning research (e.g., by (Silburt et al., 2019)).

A larger-scale candidate for an annotation baseline on Mars is the use of Robbins and Hynek Mars annotations (Robbins & Hynek, 2012) and on the Moon, a combination of

Povilaitis et al. (2018) (5-20 km) and Head et al. (2010) (>20 km), evaluated on ±30º latitude and all longitudes. Both datasets are well formatted, annotated by expert crater counters, and easy to use. Even if researchers are interested in other regions, also running their method on one or both of these datasets will allow real comparisons to be made between methods. An established set of crater baselines will provide the community with valuable comparisons for evaluating machine learning research.

6.5 HUMAN IN THE LOOP

In order to work toward generally accepted increased automation, a phased autonomy approach could help. Currently, crater detection algorithms are primarily used only by their original developers and no standard exists for analyzing new image data or regions.

Table 6-1: Increasing autonomy for crater counting with human in the loop

Phases of Autonomy	Characteristics	Requirements
Phase 1	Hand counted	Humans trained, agree on standard crater counting method
Phase 2	Human in the loop	Using established hand annotations, research many machine learning algorithms in use in other domains, output results to formats that can be loaded into programs like JMARS to evaluate, modify the predictions, and obtain ages
Phase 3	Full automation	Crater detection algorithms have gained sufficient accuracy and confidence among the planetary science community

The current state of research is in Phase 2: Human in the loop. There have been a few notable periods in the history of crater counting with massive shifts forward in thinking (i.e., adopting the cumulative crater curves to evaluate Mars (Michael & Neukum, 2010)), and the community will likely need such a consensus to adopt either a single or a collection of automated algorithms. In the future, like how Craterstats is used to determine isochrons via various production functions (Michael, 2008), planetary geologists may be able to load one or an ensemble of several models for detecting craters in new data.

In order to move to the next stage of crater detection algorithms, development of code and software that uses one or more CDAs to generate a crater list that is then verified by humans may be the solution. The feedback from users could provide valuable additional training data. Over time, the algorithms could become tailored to the user if the corrections are used as additional training data.

6.6 FUTURE USE OF MACHINE LEARNING

Several of the trends in machine learning research such as the use of region proposal networks and classifiers (i.e., Faster R-CNN (Ren, He, Girshick, & Sun, 2016), Mask R-CNN (He, Gkioxari, Dollár, & Girshick, 2017)) and Generative Adversarial Networks (GAN) (Goodfellow et al., 2014) may have uses in the crater counting and localization problem. GANs in particular may be used to produce additional training data to avoid overfitting and the adversarial techniques approach may help improve transfer between different conditions. Although no journal papers have been published on these topics yet (that we could find), early results using techniques like Faster R-CNN have been published at the Lunar and Planetary Science Conference (Emami, Ahmad, Bebis, Nefian, & Fong, 2018) and other conference poster sessions, showing this is a promising direction within the community. With additional annotation datasets, more objects like boulders or lava tubes could possibly be detected using these techniques. Some research, such as (Palafox et al., 2017), already looks for multiple geological landforms on Mars, and Wang et al. (Wang, Di, Xin, & Wan, 2017) looks for dark slope streaks. As more annotated databases become accessible and are formatted in ways that enable loading them into a machine learning algorithm, it is likely more research will go in this direction.

An interesting trend in machine learning is transfer learning, which can mean less specificity in training. A technique that has proved useful even in the crater counting domain is using pre-trained networks as the base (trained on other images) and doing additional training with craters. This type of transfer learning will only improve as new networks are trained and represents an opportunity for scientists to use less data. Training from scratch requires more hyperparameter tuning. By using pre-trained or partially pre-trained networks, training time may be reduced and more types of network architectures can be used. For example, Norman et al. use GoogLeNet-OverFeat pre-trained with ImageNet to detect craters on Mars (Benedix et al., 2018; Norman et al., 2018).

6.7 LESSONS

Numerous lessons emerged throughout this research. From the experiments with Crater U-Net, the following recommendations emerge:

- Different applications will want to use different hyperparameters for the model. Crater counters value repeatability and similarity to the trained set, so the recall score is highly valued but needs to be balanced with precision. Those using this to find crater hazards would want a low target threshold to get more false positives.

- When consider the application, architecture matters. For example, the 3x3 kernel preserves more of the crater edge size while larger kernels forces roundedness. Having a variety of kernel sizes helps find different faded craters. Higher kernel values help with finding large faded craters.

- Improvements from adding more data may plateau for a given kernel size and number of filters. This will be dependent on the dataset used.

- Even using the exact same training data, model results can vary. If constrained to using a smaller dataset, running multiple times may yield better results.

- Fixed edge targets were consistently preferred over variable edge and solid targets.

From the review of CNN’s use for crater counting:

- A priority for the community is identifying which annotations are the correct ones for various planetary bodies. (Even a small number of “correct” annotations would be beneficial. The important point is that the best ones are identified and accessible to machine learning researchers.)

- Using massively huge databases of labeled craters can provide foundation of a classifier/model which then is used as the pre-trained model for other new databases.

CHAPTER 7: CONCLUSION

This is the first work to evaluate a segmentation U-Net on Mars thermal infrared data. It details several projects related to automated crater detection using segmentation Convolutional Neural Networks and supports the conclusion that machine learning methods can augment manual work by human experts.

In Chapter 1, the interdisciplinary crater counting and automatic crater detection is introduced. Contributions of this thesis are reported, and the reader is introduced to some of the problem's complexities.

In Chapter 2, a comprehensive review of the crater counting papers utilizing CNNs is completed, and two categories (classification and segmentation) are used to sort the papers by their machine learning function. This chapter also lists many of the challenges experienced while doing crater counting and machine learning research to enable each group of researchers to understand better where the difficulties lie. The two groups are encouraged to collaborate because of the highly specialized nature of the data formats, similarity between the crater counting problem and other applications of interest within the machine learning community, and the tremendous potential of working together.

In Chapter 3, Crater U-Net, in the same architecture family as U-Net, is introduced and experiments are run to learn the impact of various hyperparameters on the specific application of crater detection. In the process, several models are created and the use of metrics (loss, accuracy, recall, precision, F1) are evaluated for their usefulness throughout the process. Sensitivity analysis is performed by training with 3, 6, 9, 12, and 15 tiles (out of 24, each is 30ºx30º, 7680x7680 pixels, THEMIS Daytime IR) and training with four different randomly selected sets of tiles.

In Chapter 4, the Crater U-Net is both compared to original U-Net and used to age date nine locations around Mars that span the Noachian, Hesperian, and Amazonian terrain.

In Chapter 5, a systems engineering diagram, Object Process Methodology, is used to visualize the pieces and flow of crater counting by hand versus with a human in the loop.

In Chapter 6, recommendations for those interested in performing their own machine learning crater counting research are made. In particular, there is a need for a diverse baseline with which to compare all machine learning algorithms is sorely needed to enable comparisons between techniques.

Together, this research makes a strong case for using segmentation for crater detection, particularly with a human in the loop. Segmentation can actively reduce the workload for crater counters.

REFERENCES

CHAPTER 1

Bugiolacchi, R., Bamford, S., Tar, P., Thacker, N., Crawford, I. A., Joy, K. H., et al. (2016). The Moon Zoo citizen science project: Preliminary results for the Apollo 17 landing site. *Icarus*, *271*(C), 30–48. http://doi.org/10.1016/j.icarus.2016.01.021

Cohen, J. P. (2016, May 10). *Automated Crater Detection Using Machine Learning*.

Crater Analysis Techniques Working Group. (1979). Standard Techniques for Presentation and Analysis of Crater Size-Frequency Data. *Icarus*, *37*, 467–474.

Denker, J. S., Gardner, W. R., Graf, H. P., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Neural Network Recognizer for Hand-Written Zip Code Digits. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 1* (pp. 323–331). Morgan-Kaufmann.

Hartmann, W. K., & Neukum, G. (2001). Cratering Chronology and the Evolution of Mars. *Space Science Reviews*, *96*(1-4), 165–194. http://doi.org/10.1023/A:1011945222010

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN (pp. 2980–2988). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy: IEEE. http://doi.org/10.1109/ICCV.2017.322

Ivanov, B. A. (2001). Mars/Moon Cratering Rate Ratio Estimates. *Chronology and Evolution of Mars*, *96*, 87–104.

LeCun, Y., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. http://doi.org/10.1109/5.726791

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. http://doi.org/10.1038/nature14539

Liu, S., Ding, W., Gao, F., & Stepinski, T. F. (2012). Adaptive Selective Learning for automatic identification of sub-kilometer craters. *Neurocomputing*, *92*, 78–87. http://doi.org/10.1016/j.neucom.2011.11.023

Long, Y., Gong, Y., Xiao, Z., & Liu, Q. (2017). Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(5), 2486–2498. http://doi.org/10.1109/TGRS.2016.2645610

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(2), 645–657. http://doi.org/10.1109/TGRS.2016.2612821

Michael, G. (2008). Surface dating using *Craterstats* program (pp. 1–8). Presented at the Mars Express Data Workshop, St. Louis.

Michael, G. G., & Neukum, G. (2010). Planetary surface dating from crater size–frequency distribution measurements: Partial resurfacing events and statistical age uncertainty, *294*(3), 223–229. http://doi.org/https://doi.org/10.1016/j.epsl.2009.12.041

Michael, G. G., Platz, T., Kneissl, T., & Schmedemann, N. (2012). Planetary surface dating from crater size–frequency distribution measurements: Spatial randomness and clustering. *Icarus*, *218*(1), 169–177. http://doi.org/10.1016/j.icarus.2011.11.033

Mou, L., Ghamisi, P., & Zhu, X. (2017). Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(7).

Nesvold, E. R., Greenberg, A., Erasmus, N., van Heerden, E., Galache, J. L., Dahlstrom, E., & Marchis, F. (2018). The Deflector Selector: A machine learning framework for prioritizing hazardous object deflection technology development. *Acta Astronautica*, *146*, 33–45. http://doi.org/10.1016/j.actaastro.2018.01.049

Neukum, G., Ivanov, B. A., & Hartmann, W. K. (2001). Cratering Records in the Inner Solar System in Relation to the Lunar Reference System. *Chronology and Evolution of Mars*, *96*, 55–86. http://doi.org/https://doi.org/10.1023/A:1011989004263

Platz, T., Michael, G., Tanaka, K. L., Skinner, J. A., Jr., & Fortezzo, C. M. (2013). Crater-based dating of geological units on Mars: Methods and application for the new global geological map. *Icarus*, *225*(1), 806–827. http://doi.org/10.1016/j.icarus.2013.04.021

Robbins, S. J., Antonenko, I., Kirchoff, M. R., Chapman, C. R., Fassett, C. I., Herrick, R. R., et al. (2014). The variability of crater identification among expert and community crater analysts. *Icarus*, *234*, 109–131. http://doi.org/10.1016/j.icarus.2014.02.022

Robbins, S. J., Riggs, J. D., Weaver, B. P., Bierhaus, E. B., Chapman, C. R., Kirchoff, M. R., et al. (2018). Revised recommended methods for analyzing crater size-frequency distributions. *Meteoritics & Planetary Science*, *53*(4), 891–931. http://doi.org/10.1111/maps.12990

Salih, A. L., Mühlbauer, M., Grumpe, A., Pasckert, J. H., Wöhler, C., & Hiesinger, H. (2016). Automatic Age Map Construction for the Floor of Lunar Crater Tsiolkovsky (pp. 1–2). Presented at the 47th Lunar Planetary Science Conference.

Shallue, C. J., & Vanderburg, A. (2018). Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90. *The Astronomical Journal*, *155*(2), 94. http://doi.org/10.3847/1538-3881/aa9e09

Stanboli, A., Bue, B., Wagstaff, K., & Altinok, A. (2017). Automated Content Detection for Cassini Images (p. Abstract #7048). Presented at the 3rd Planetary Data Workshop, Houston: Lunar and Planetary Institute.

Tanaka, K. L. (2014). The New Geologic Map of Mars: Guiding Research and Education. *Eos, Transactions American Geophysical Union*, *95*(38), 341–342. http://doi.org/10.1002/2014EO380001

Tanaka, K. L., Skinner, J. A., Dohm, J. M., Irwin, R. P., III, Kolb, E. J., Fortezzo, C. M., et al. (2014). *Geologic map of Mars* (No. 3292, scale 1:20,000,000, pamphlet 43 p.) (pp. 1–1).

Urbach, E. R., & Stepinski, T. F. (2009). Automatic detection of sub-km craters in high resolution planetary images. *Planetary and Space Science*, *57*(7), 880–887. http://doi.org/10.1016/j.pss.2009.03.009

Warner, N. H., Gupta, S., Calef, F., Grindrod, P., Boll, N., & Goddard, K. (2015). Minimum effective area for high resolution crater counting of martian terrains. *Icarus*, *245*, 198–240. http://doi.org/10.1016/j.icarus.2014.09.024

Yamamoto, S., Matsunaga, T., Nakamura, R., Sekine, Y., Hirata, N., & Yamaguchi, Y. (2017). An Automated Method for Crater Counting Using Rotational Pixel Swapping Method. *IEEE Transactions on Geoscience and Remote Sensing*, 1–14. http://doi.org/10.1109/TGRS.2017.2691758

CHAPTER 2

Bandeira, L., Ding, W. & Stepinski, T.F., 2010. Automatic Detection of Sub-km Craters Using Shape and Texture Information. In 41st Lunar and Planetary Science Conference.

Barlow, N.G., 1988. Crater size-frequency distributions and a revised Martian relative chronology. *Icarus*, 75(2), pp.285–305.

Becker, K.J. et al., 2016. First Global Digital Elevation Model of Mercury. In 47th Lunar Planetary Science Conference.

Benedix, G.K. et al., 2018. Automated Detection of Martian Craters Using a Convolutional Neural Network. In 49th Lunar and Planetary Science Conference. Houston: Lunar and Planetary Institute. Available at: http://www.hou.usra.edu/meetings/lpsc2018/pdf/2202.pdf.

Bergstra, J. et al., 2015. Hyperopt: a Python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, 8(1), p.014008.

Chinchor, N., 1992. MUC-4 evaluation metrics. In Fourth Message Understanding Conference (MUC-4). McLean, VA: Association for Computational Linguistics, pp. 22–29. Available at: http://www.aclweb.org/anthology/M92-1002.

Chollet, F., 2017. *Deep Learning with Python* T. Arritola, ed., Manning Publication Co.

Chung, C.P., Chung, C.O. & Yam, C.H., 2014. Lunar Crater Identification from Machine Learning Perspective. *Acta Futura*, 9, pp.41–47.

Clanuwat, T. et al., 2018. Deep Learning for Classical Japanese Literature. *arXiv*, cs.CV.

Cohen, J.P., 2016. *Automated Crater Detection Using Machine Learning*.

Cohen, J.P. et al., 2016. Crater Detection via Convolutional Neural Networks. In 47th Lunar Planetary Science Conference. p. Abstract #1143. Available at: https://www.hou.usra.edu/meetings/lpsc2016/pdf/1143.pdf.

Crater Analysis Techniques Working Group, 1979. Standard Techniques for Presentation and Analysis of Crater Size-Frequency Data. *Icarus*, 37, pp.467–474.

DeLatte, D.M. et al., 2019. Segmentation Convolutional Neural Networks for Automatic Crater Detection on Mars. *unpublished*, pp.1–14.

DeLatte, D.M., Crites, S.T., Guttenberg, N., Tasker, E.J. & Yairi, T., 2018a. Experiments in Segmenting Mars Craters Using Convolutional Neural Networks. In i-SAIRAS. Madrid, pp. 1–8.

DeLatte, D.M., Crites, S.T., Guttenberg, N., Tasker, E.J. & Yairi, T., 2018b. Exploration of Machine Learning Methods for Crater Counting on Mars. In 46th Lunar and Planetary Science Conference. pp. 1–2.

Denevi, B.W. et al., 2018. Calibration, Projection, and Final Image Products of MESSENGER's Mercury Dual Imaging System. *Space Science Reviews*, 214(1), pp.1–52.

Ding, W. et al., 2011. Subkilometer crater discovery with boosting and transfer learning. *ACM Transactions on Intelligent Systems and Technology*, 2(4), pp.1–22.

Dori, D., 2011. *Object-Process Methodology*, Berlin, Heidelberg: Springer Science & Business Media.

Emami, E. & Bebis, G., 2015. Automatic Crater Detection Using Convex Grouping and Convolutional Neural Networks. In G. Bebis et al., eds. *Advances in Visual Computing*. Lecture Notes in Computer Science. Cham: Springer International Publishing, pp. 213–224. Available at: https://link.springer.com/chapter/10.1007/978-3-319-27863-6_20.

Emami, E., Ahmad, T., Bebis, G., Nefian, A. & Fong, T., 2018a. Lunar Crater Detection via Region-Based Convolutional Neural Networks. In 49th Lunar and Planetary Science Conference. Houston: Lunar and Planetary Institute. Available at: http://www.hou.usra.edu/meetings/lpsc2018/pdf/2381.pdf.

Emami, E., Ahmad, T., Bebis, G., Nefian, A. & Fong, T., 2018b. On Crater Classification Using Deep Convolutional Neural Networks. In 49th Lunar and Planetary Science Conference. Houston: Lunar and Planetary Institute, p. Abstract #2566. Available at: http://www.hou.usra.edu/meetings/lpsc2018/pdf/2566.pdf.

Enke, B. & Merline, W.J., 2005. Learning to Detect Small Impact Craters. In 2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1. Breckenridge, CO, USA, pp. 178–184.

Fergason, R., Hare, T. & Laura, J., 2017. *HRSC and MOLA blended digital elevation model at 200m*, Available at: https://astrogeology.usgs.gov/search/map/Mars/Topography/HRSC_MOLA_Blend/Mars_HRSC_MOLA_BlendDEM_Global_200mp.

Goodfellow, I. et al., 2014. Generative Adversarial Nets. In Z. Ghahramani et al., eds. *Advances in Neural Information Processing Systems 27*. Curran Associates, Inc., pp. 2672–2680.

Hartmann, W.K. & Neukum, G., 2001. Cratering Chronology and the Evolution of Mars. *Space Science Reviews*, 96(1-4), pp.165–194.

He, K. et al., 2017. Mask R-CNN. In 2017 IEEE International Conference on Computer Vision (ICCV). Venice, Italy: IEEE, pp. 2980–2988.

Head, J.W. et al., 2010. Global Distribution of Large Lunar Craters: Implications for Resurfacing and Impactor Populations. *Science*, 329(5998), pp.1504–1507.

Hiesinger, H. et al., 2010. Ages and stratigraphy of lunar mare basalts in Mare Frigoris and other nearside maria based on crater size-frequency distribution measurements. *Journal of Geophysical Research*, 115(E3), p.141.

Hiesinger, H. et al., 2000. Ages of mare basalts on the lunar nearside. *Journal of Geophysical Research: Planets*, 105(E12), pp.29239–29275.

Ivanov, B.A., 2006. Earth/Moon impact rate comparison: Searching constraints for lunar secondary/primary cratering proportion. *Icarus*, 183(2), pp.504–507.

Jin, H., Song, Q. & Hu, X., 2018. Auto-Keras: Efficient Neural Architecture Search with Network Morphism. *arXiv*.

Jin, S. & Zhang, T., 2014. Automatic detection of impact craters on Mars using a modified adaboosting method. *Planetary and Space Science*, 99, pp.112–117.

Kerner, H.R., Bell, J.F., III & Ben Amor, H., 2018. Context-dependent image quality assessment of JPEG compressed Mars Science Laboratory Mastcam images using convolutional neural networks. *Computers and Geosciences*, 118, pp.109–121.

Kim, J.R. et al., 2005. Automated Crater Detection, A New Tool for Mars Cartography and Chronology. *Photogrammetric Engineering & Remote Sensing*, 71(10), pp.1205–1217.

Li, B. et al., 2015. Automatic Detection and Boundary Extraction of Lunar Craters Based on LOLA DEM Data. *Earth, Moon, and Planets*, 115(1-4), pp.59–69.

Li, W. et al., 2017. Recognizing terrain features on terrestrial surface using a deep learning model. In Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery - GeoAI '17.

Liu, A., Chen, M. & Pan, W., 2012. Crater detection algorithm with part PHOG features for safe landing. In 2012 International Conference on Systems and Informatics (ICSAI). Yantai, China: IEEE, pp. 103–106.

Lo, H.Z., 2016. *Deep Networks: Applications, Interpretability, and Optimization*.

Machado, M., Bandeira, L. & Pina, P., 2015. Automatic Crater Detection in Large Scale on Lunar Maria. In 46th Lunar and Planetary Science Conference. p. Abstract #1832.

Magellan Team, 1997. *Magellan Global Topography 4641m v02 (1997)*, PDS GeoScience Node. Available at: https://astrogeology.usgs.gov/search/map/Venus/Magellan/RadarProperties/Venus_Magellan_Topography_Global_4641m_v02.

Malin, M.C. & Edgett, K.S., 2000. Sedimentary Rocks of Early Mars. 290(5498), pp.1927–1937.

Martins, R. et al., 2009. Crater Detection by a Boosting Approach. *IEEE Geoscience and Remote Sensing Letters*, 6(1), pp.127–131.

Michael, G., 2008. Surface dating using *Craterstats* program. In Mars Express Data Workshop. St. Louis, pp. 1–8.

Michael, G.G. & Neukum, G., 2010. Planetary surface dating from crater size–frequency distribution measurements: Partial resurfacing events and statistical age uncertainty. 294(3), pp.223–229.

NASA, PDS Geosciences Node Orbital Data Explorer (ODE) S. Slavney, ed. *PDS Geosciences Node*. Available at: http://ode.rsl.wustl.edu/ [Accessed December 28, 2018].

NASA Mars Odyssey/THEMIS Team, 2010. THEMIS Day IR 100m Global Mosaic. Available at: http://www.mars.asu.edu/data/thm_dir_100m/.

NASA Mars Odyssey/THEMIS Team, 2006. THEMIS Day IR Global Mosaic. (92160 x 46080 px). Available at: http://www.mars.asu.edu/data/thm_dir/.

Nesvold, E.R. et al., 2018. The Deflector Selector: A machine learning framework for prioritizing hazardous object deflection technology development. *Acta Astronautica*, 146, pp.33–45. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0094576517313747.

Neukum, G., Ivanov, B.A. & Hartmann, W.K., 2001. Cratering Records in the Inner Solar System in Relation to the Lunar Reference System. *Chronology and Evolution of Mars*, 96, pp.55–86.

Neumann, G.A. et al., 2015. Lunar impact basins revealed by Gravity Recovery and Interior Laboratory measurements. *Science Advances*, 1(9), p.e1500852.

Nguyen, T. et al., 2018. Computer-aided discovery of debris disk candidates: A case study using the Wide-Field Infrared Survey Explorer (WISE) catalog. *Astronomy and Computing*, 23, pp.72–82.

Norman, C.J. et al., 2018. Automated Detection of Craters in Martian Satellite Imagery Using Convolutional Neural Networks. In Planetary Science Informatics and Data Analytics Conference. Houston: Lunar and Planetary Institute, p. Abstract #6004. Available at: http://www.hou.usra.edu/meetings/informatics2018/pdf/6004.pdf.

Palafox, L.F. et al., 2017. Automated detection of geological landforms on Mars using Convolutional Neural Networks. 101, pp.48–56.

Pasckert, J.H., Hiesinger, H. & van der Bogert, C.H., 2015. Small-scale lunar farside volcanism. *Icarus*, 257, pp.336–354.

Patil, J. & Kini, S.N., 2015. A Survey Paper on Crater Detection. *International Journal of Science and Research (IJSR)*, 4(12), pp.81–85.

Pieters, C.M. et al., 2008. Lunar international science coordination/calibration targets (L-ISCT). *Advances in Space Research*, 42(2), pp.248–258.

Platz, T. et al., 2013. Crater-based dating of geological units on Mars: Methods and application for the new global geological map. *Icarus*, 225(1), pp.806–827.

Povilaitis, R.Z. et al., 2018. Crater density differences: Exploring regional resurfacing, secondary crater populations, and crater saturation equilibrium on the moon. *Planetary and Space Science*, 162, pp.41–51.

Preusker, F. et al., 2016. Dawn at Ceres --- Shape Model and Rotational State. *Lunar and Planetary Science Conference*, 47, p.1954.

Preusker, F. et al., 2014. Global Shape of (4) Vesta from Dawn FC Stereo Images. In Vesta in the Light of Dawn: First Exploration of a Protoplanet in the Asteroid Belt. p. 2027.

Ren, S. et al., 2016. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. pp.1–14.

Robbins, S.J., 2018a. A Global Lunar Crater Database, Complete for Craters ≥1 KM, III: Reassessing the Lunar Crater Production Function, and Lessons Learned Applied to the Global Mars Crater Database. In 49th Lunar and Planetary Science Conference. p. Abstract #2443. Available at: https://www.hou.usra.edu/meetings/lpsc2018/pdf/2443.pdf.

Robbins, S.J., 2018b. A Global Lunar Crater Database, Complete for Craters ≥1-2 KM, IV. *NASA Exploration Science Forum*.

Robbins, S.J. & Hynek, B.M., 2012a. A new global database of Mars impact craters ≥1 km: 1. Database creation, properties, and parameters. *Journal of Geophysical Research: Planets*, 117(E5).

Robbins, S.J. & Hynek, B.M., 2012b. A new global database of Mars impact craters ≥1 km: 2. Global crater properties and regional variations of the simple-to-complex transition diameter. *Journal of Geophysical Research: Planets*, 117(E6), pp.1–21.

Robbins, S.J. et al., 2014. The variability of crater identification among expert and community crater analysts. *Icarus*, 234, pp.109–131. Available at: http://dx.doi.org/10.1016/j.icarus.2014.02.022.

Rolnick, D. et al., 2017. Deep Learning is Robust to Massive Label Noise. *arXiv*.

Ronneberger, O., Fischer, P. & Brox, T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab et al., eds. Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015. Cham: Springer International Publishing, pp. 234–241.

Salamunićcar, G. & Lončarić, S., 2008a. GT-57633 catalogue of Martian impact craters developed for evaluation of crater detection algorithms. 56(15), pp.1992–2008.

Salamunićcar, G. & Lončarić, S., 2008b. Open framework for objective evaluation of crater detection algorithms with first test-field subsystem based on MOLA data. *Advances in Space Research*, 42(1), pp.6–19.

Salamunićcar, G. et al., 2014. Hybrid method for crater detection based on topography reconstruction from optical images and the new LU78287GT catalogue of Lunar impact craters. *Advances in Space Research*, 53(12), pp.1783–1797.

Salamunićcar, G. et al., 2011. MA130301GT catalogue of Martian impact craters and advanced evaluation of crater detection algorithms using diverse topography and image datasets. 59(1), pp.111–131.

Salih, A.L. et al., 2017. Automatic crater detection and age estimation for mare regions on the lunar surface. In 2017 25th European Signal Processing Conference (EUSIPCO). IEEE, pp. 518–522.

Sawabe, Y., Matsunaga, T. & Rokugawa, S., 2006. Automated detection and classification of lunar craters using multiple approaches. *Advances in Space Research*, 37(1), pp.21–27.

Shallue, C.J. & Vanderburg, A., 2018. Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90. *The Astronomical Journal*, 155(2), p.94.

Silburt, A. et al., 2019. Lunar crater identification via deep learning. *Icarus*, 317, pp.27–38.

Srivastava, N., Hinton, G. & Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting Y. Bengio, ed. *Journal of Machine Learning Research*, 15, pp.1929–1958.

Stepinski, T.F., Ding, W. & Vilalta, R., 2012. Detecting Impact Craters in Planetary Images Using Machine Learning. In R. Magdalena-Benedito et al., eds. *Intelligent Data Analysis for Real-Life Applications: Theory and Practice*. pp. 146–159.

Tanaka, K.L. et al., 2014. *Geologic map of Mars*,

Urbach, E.R. & Stepinski, T.F., 2009. Automatic detection of sub-km craters in high resolution planetary images. *Planetary and Space Science*, 57(7), pp.880–887. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0032063309000956.

Vamshi, G.T., Martha, T.R. & Vinod Kumar, K., 2016. An object-based classification method for automatic detection of lunar impact craters from topographic data. *Advances in Space Research*, 57(9), pp.1978–1988.

Wang, Y. et al., 2017. Automatic detection of Martian dark slope streaks by machine learning using HiRISE images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 129, pp.12–20.

Yamamoto, S. et al., 2017. An Automated Method for Crater Counting Using Rotational Pixel Swapping Method. *IEEE Transactions on Geoscience and Remote Sensing*, pp.1–14.

Yin, J., Li, H. & Jia, X., 2015. Crater Detection Based on Gist Features. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8(1), pp.23–29.

Zhou, Y. et al., 2018. Automatic detection of lunar craters based on DEM data with the terrain analysis method. *Planetary and Space Science*, 160, pp.1–11.

Zoph, B. & Le, Q.V., 2016. Neural Architecture Search with Reinforcement Learning. *arXiv*, cs.LG.

CHAPTER 3

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., et al. (2016). Tensorflow: a system for large-scale machine learning. (pp. 265–283). Presented at the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA.

Automatic detection of sub-km craters in high resolution planetary images. (2009). Automatic detection of sub-km craters in high resolution planetary images, *57*(7), 880–887. http://doi.org/10.1016/j.pss.2009.03.009

Bugiolacchi, R., Bamford, S., Tar, P., Thacker, N., Crawford, I. A., Joy, K. H., et al. (2016). The Moon Zoo citizen science project: Preliminary results for the Apollo 17 landing site. *Icarus*, *271*(C), 30–48. http://doi.org/10.1016/j.icarus.2016.01.021

Chinchor, N. (1992). MUC-4 evaluation metrics (pp. 22–29). Presented at the Fourth Message Understanding Conference (MUC-4), McLean, VA: Association for Computational Linguistics. http://doi.org/10.3115/1072064.1072067

Chollet, F. (2017). Deep Learning with Python. (T. Arritola, Ed.) (pp. 1–386). Manning Publication Co.

Cohen, J. P. (2016, May 10). *Automated Crater Detection Using Machine Learning*.

Crater Analysis Techniques Working Group. (1979). Standard Techniques for Presentation and Analysis of Crater Size-Frequency Data. *Icarus*, *37*, 467–474.

Denker, J. S., Gardner, W. R., Graf, H. P., Henderson, D., Howard, R. E., Hubbard, W., et al. (1989). Neural Network Recognizer for Hand-Written Zip Code Digits. In D. S. Touretzky (Ed.), *Advances in Neural Information Processing Systems 1* (pp. 323–331). Morgan-Kaufmann.

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S. O., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017, April 25). A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv*.

Hartmann, W. K., & Neukum, G. (2001). Cratering Chronology and the Evolution of Mars. *Space Science Reviews*, *96*(1-4), 165–194. http://doi.org/10.1023/A:1011945222010

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN (pp. 2980–2988). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy: IEEE. http://doi.org/10.1109/ICCV.2017.322

Ivanov, B. A. (2001). Mars/Moon Cratering Rate Ratio Estimates. *Chronology and Evolution of Mars*, *96*, 87–104.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 25* (pp. 1097–1105). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

LeCun, Y., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. http://doi.org/10.1109/5.726791

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. http://doi.org/10.1038/nature14539

Liu, S., Ding, W., Gao, F., & Stepinski, T. F. (2012). Adaptive Selective Learning for automatic identification of sub-kilometer craters. *Neurocomputing*, *92*, 78–87. http://doi.org/10.1016/j.neucom.2011.11.023

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation (pp. 3431–3440). Presented at the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE. http://doi.org/10.1109/CVPR.2015.7298965

Long, Y., Gong, Y., Xiao, Z., & Liu, Q. (2017). Accurate Object Localization in Remote Sensing Images Based on Convolutional Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(5), 2486–2498. http://doi.org/10.1109/TGRS.2016.2645610

Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(2), 645–657. http://doi.org/10.1109/TGRS.2016.2612821

Michael, G. (2008). Surface dating using *Craterstats* program (pp. 1–8). Presented at the Mars Express Data Workshop, St. Louis.

Michael, G. G., & Neukum, G. (2010). Planetary surface dating from crater size–frequency distribution measurements: Partial resurfacing events and statistical age uncertainty, *294*(3), 223–229. http://doi.org/https://doi.org/10.1016/j.epsl.2009.12.041

Michael, G. G., Platz, T., Kneissl, T., & Schmedemann, N. (2012). Planetary surface dating from crater size–frequency distribution measurements: Spatial randomness and clustering. *Icarus*, *218*(1), 169–177. http://doi.org/10.1016/j.icarus.2011.11.033

Mou, L., Ghamisi, P., & Zhu, X. (2017). Deep Recurrent Neural Networks for Hyperspectral Image Classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(7).

NASA Mars Odyssey/THEMIS Team. (2006, November 16). THEMIS Day IR Global Mosaic. Mars Space Flight Facility. Retrieved from http://www.mars.asu.edu/data/thm_dir/

Nesvold, E. R., Greenberg, A., Erasmus, N., van Heerden, E., Galache, J. L., Dahlstrom, E., & Marchis, F. (2018). The Deflector Selector: A machine learning framework for prioritizing hazardous object deflection technology development. *Acta Astronautica*, *146*, 33–45. http://doi.org/10.1016/j.actaastro.2018.01.049

Neukum, G., Ivanov, B. A., & Hartmann, W. K. (2001). Cratering Records in the Inner Solar System in Relation to the Lunar Reference System. *Chronology and Evolution of Mars*, *96*, 55–86. http://doi.org/https://doi.org/10.1023/A:1011989004263

Platz, T., Michael, G., Tanaka, K. L., Skinner, J. A., Jr., & Fortezzo, C. M. (2013). Crater-based dating of geological units on Mars: Methods and application for the new global geological map. *Icarus*, *225*(1), 806–827. http://doi.org/10.1016/j.icarus.2013.04.021

Robbins, S. J., & Hynek, B. M. (2011). Secondary crater fields from 24 large primary craters on Mars: Insights into nearby secondary crater production, *116*(E10), 467. http://doi.org/10.1029/2011JE003820

Robbins, S. J., & Hynek, B. M. (2012a). A new global database of Mars impact craters ≥1 km: 1. Database creation, properties, and parameters. *Journal of Geophysical Research: Planets*, *117*(E5). http://doi.org/10.1029/2011JE003966

Robbins, S. J., & Hynek, B. M. (2012b). A new global database of Mars impact craters ≥1 km: 2. Global crater properties and regional variations of the simple-to-complex transition diameter. *Journal of Geophysical Research: Planets*, *117*(E6), 1–21. http://doi.org/10.1029/2011JE003967

Robbins, S. J., Antonenko, I., Kirchoff, M. R., Chapman, C. R., Fassett, C. I., Herrick, R. R., et al. (2014). The variability of crater identification among expert and community crater analysts. *Icarus*, *234*, 109–131. http://doi.org/10.1016/j.icarus.2014.02.022

Robbins, S. J., Riggs, J. D., Weaver, B. P., Bierhaus, E. B., Chapman, C. R., Kirchoff, M. R., et al. (2018). Revised recommended methods for analyzing crater size-frequency distributions. *Meteoritics & Planetary Science*, *53*(4), 891–931. http://doi.org/10.1111/maps.12990

Ronneberger, O., Fischer, P., & Brox, T. (2015a). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), (Vol. 9351, pp. 234–241). Presented at the Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015, Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-24574-4

Ronneberger, O., Fischer, P., & Brox, T. (2015b, May 18). U-Net: Convolutional Networks for Biomedical Image Segmentation. *arXiv*. Retrieved from https://arxiv.org/abs/1505.04597

Salih, A. L., Mühlbauer, M., Grumpe, A., Pasckert, J. H., Wöhler, C., & Hiesinger, H. (2016). Automatic Age Map Construction for the Floor of Lunar Crater Tsiolkovsky (pp. 1–2). Presented at the 47th Lunar Planetary Science Conference.

Sasaki, Y. (2007). *The truth of the F-measure*.

Shallue, C. J., & Vanderburg, A. (2018). Identifying Exoplanets with Deep Learning: A Five-planet Resonant Chain around Kepler-80 and an Eighth Planet around Kepler-90. *The Astronomical Journal*, *155*(2), 94. http://doi.org/10.3847/1538-3881/aa9e09

Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., et al. (2019). Lunar crater identification via deep learning. *Icarus*, *317*, 27–38. http://doi.org/10.1016/j.icarus.2018.06.022

Srivastava, N., Hinton, G., & Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, *15*, 1929–1958.

Stanboli, A., Bue, B., Wagstaff, K., & Altinok, A. (2017). Automated Content Detection for Cassini Images (p. Abstract #7048). Presented at the 3rd Planetary Data Workshop, Houston: Lunar and Planetary Institute.

Tanaka, K. L. (2014). The New Geologic Map of Mars: Guiding Research and Education. *Eos, Transactions American Geophysical Union*, *95*(38), 341–342. http://doi.org/10.1002/2014EO380001

Tanaka, K. L., Skinner, J. A., Dohm, J. M., Irwin, R. P., III, Kolb, E. J., Fortezzo, C. M., et al. (2014). *Geologic map of Mars* (No. 3292, scale 1:20,000,000, pamphlet 43 p.) (pp. 1–1).

Warner, N. H., Gupta, S., Calef, F., Grindrod, P., Boll, N., & Goddard, K. (2015). Minimum effective area for high resolution crater counting of martian terrains. *Icarus*, *245*, 198–240. http://doi.org/10.1016/j.icarus.2014.09.024

Yamamoto, S., Matsunaga, T., Nakamura, R., Sekine, Y., Hirata, N., & Yamaguchi, Y. (2017). An Automated Method for Crater Counting Using Rotational Pixel Swapping Method. *IEEE Transactions on Geoscience and Remote Sensing*, 1–14. http://doi.org/10.1109/TGRS.2017.2691758

CHAPTER 4

Chollet, F. (2017). Deep Learning with Python. (T. Arritola, Ed.) (pp. 1–386). Manning Publication Co.

Kohl, S. A. A., Romera-Paredes, B., Meyer, C., De Fauw, J., Ledsam, J. R., Maier-Hein, K. H., et al. (2018). A Probabilistic U-Net for Segmentation of Ambiguous Images. Presented at the 32nd Conference on Neural Information Processing Systems, Montréal, Canada.

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. In N. Navab, J. Hornegger, W. M. Wells, & A. F. Frangi (Eds.), (Vol. 9351, pp. 234–241). Presented at the Medical Image Computing and Computer-Assisted Intervention -- MICCAI 2015, Cham: Springer International Publishing. http://doi.org/10.1007/978-3-319-24574-4

Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., et al. (2019). Lunar crater identification via deep learning. *Icarus*, *317*, 27–38. http://doi.org/10.1016/j.icarus.2018.06.022

Tanaka, K. L., Skinner, J. A., Dohm, J. M., Irwin, R. P., III, Kolb, E. J., Fortezzo, C. M., et al. (2014). *Geologic map of Mars* (No. 3292, scale 1:20,000,000, pamphlet 43 p.) (pp. 1–1).

CHAPTER 5

Dori, D. (2011). Object-Process Methodology. Berlin, Heidelberg: Springer Science & Business Media. http://doi.org/10.1007/978-3-642-56209-9

Dori, D. (2016). Model-Based Systems Engineering with OPM and SysML. New York, NY: Springer. http://doi.org/10.1007/978-1-4939-3295-5

CHAPTER 6

Bandeira, L., Ding, W., & Stepinski, T. F. (2010). Automatic Detection of Sub-km Craters Using Shape and Texture Information (Vol. 41). Presented at the 41st Lunar and Planetary Science Conference, Lunar and Planetary ….

Benedix, G. K., Norman, C. J., Bland, P. A., Towner, M. C., Paxman, J., & Tan, T. (2018). Automated Detection of Martian Craters Using a Convolutional Neural Network. Presented at the 49th Lunar and Planetary Science Conference, Houston: Lunar and Planetary Institute. Retrieved from http://www.hou.usra.edu/meetings/lpsc2018/pdf/2202.pdf

Cohen, J. P., Lo, H. Z., Lu, T., & Ding, W. (2016). Crater Detection via Convolutional Neural Networks (p. Abstract #1143). Presented at the 47th Lunar Planetary Science Conference. Retrieved from https://www.hou.usra.edu/meetings/lpsc2016/pdf/1143.pdf

DeLatte, D. M., Crites, S. T., Guttenberg, N., Tasker, E. J., & Yairi, T. (2018a). Experiments in Segmenting Mars Craters Using Convolutional Neural Networks (pp. 1–8). Presented at the i-SAIRAS, Madrid.

DeLatte, D. M., Crites, S. T., Guttenberg, N., Tasker, E. J., & Yairi, T. (2018b). Exploration of Machine Learning Methods for Crater Counting on Mars (pp. 1–2). Presented at the 46th Lunar and Planetary Science Conference.

DeLatte, D. M., Crites, S. T., Guttenberg, N., Tasker, E. J., & Yairi, T. (2019). Segmentation Convolutional Neural Networks for Automatic Crater Detection on Mars. *Unpublished*.

Ding, W., Stepinski, T. F., Mu, Y., Bandeira, L., Ricardo, R., Wu, Y., et al. (2011). Subkilometer crater discovery with boosting and transfer learning. *ACM Transactions on Intelligent Systems and Technology*, *2*(4), 1–22. http://doi.org/10.1145/1989734.1989743

Emami, E., Ahmad, T., Bebis, G., Nefian, A., & Fong, T. (2018). Lunar Crater Detection via Region-Based Convolutional Neural Networks. Presented at the 49th Lunar and Planetary Science Conference, Houston: Lunar and Planetary Institute. Retrieved from http://www.hou.usra.edu/meetings/lpsc2018/pdf/2381.pdf

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative Adversarial Nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, & K. Q. Weinberger (Eds.), *Advances in Neural Information Processing Systems 27* (pp. 2672–2680). Curran Associates, Inc.

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask R-CNN (pp. 2980–2988). Presented at the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy: IEEE. http://doi.org/10.1109/ICCV.2017.322

Head, J. W., Fassett, C. I., Kadish, S. J., Smith, D. E., Zuber, M. T., Neumann, G. A., & Mazarico, E. (2010). Global Distribution of Large Lunar Craters: Implications for Resurfacing and Impactor Populations. *Science*, *329*(5998), 1504–1507. http://doi.org/10.1126/science.1195050

JAXA. (n.d.). JAXA | Smart Lander for Investigating Moon (SLIM). Retrieved January 2, 2019, from http://global.jaxa.jp/projects/sat/slim/

Lee, A., Ely, T., Sostaric, R., Strahan, A., Riedel, J., Ingham, M., et al. (2010). Preliminary Design of the Guidance, Navigation, and Control System of The Altair Lunar Lander (pp. 1–62). Presented at the AIAA Guidance, Navigation, and Control Conference, Toronto: American Institute of Aeronautics and Astronautics. http://doi.org/10.2514/6.2010-7717

Michael, G. (2008). Surface dating using *Craterstats* program (pp. 1–8). Presented at the Mars Express Data Workshop, St. Louis.

Michael, G. G., & Neukum, G. (2010). Planetary surface dating from crater size–frequency distribution measurements: Partial resurfacing events and statistical age uncertainty, *294*(3), 223–229. http://doi.org/https://doi.org/10.1016/j.epsl.2009.12.041

NASA Mars Odyssey/THEMIS Team. (2006, November 16). THEMIS Day IR Global Mosaic. Mars Space Flight Facility. Retrieved from http://www.mars.asu.edu/data/thm_dir/

Norman, C. J., Paxman, J., Benedix, G. K., Tan, T., Bland, P. A., & Towner, M. (2018). Automated Detection of Craters in Martian Satellite Imagery Using Convolutional Neural Networks (p. Abstract #6004). Presented at the Planetary Science Informatics and Data Analytics Conference, Houston: Lunar and Planetary Institute. Retrieved from http://www.hou.usra.edu/meetings/informatics2018/pdf/6004.pdf

Palafox, L. F., Hamilton, C. W., Scheidt, S. P., & Alvarez, A. M. (2017). Automated detection of geological landforms on Mars using Convolutional Neural Networks, *101*, 48–56. http://doi.org/10.1016/j.cageo.2016.12.015

Povilaitis, R. Z., Robinson, M. S., van der Bogert, C. H., Hiesinger, H., Meyer, H. M., & Ostrach, L. R. (2018). Crater density differences: Exploring regional resurfacing, secondary crater populations, and crater saturation equilibrium on the moon. *Planetary and Space Science*, *162*, 41–51. http://doi.org/10.1016/j.pss.2017.05.006

Ren, S., He, K., Girshick, R., & Sun, J. (2016). Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 1–14.

Robbins, S. J., & Hynek, B. M. (2012). A new global database of Mars impact craters ≥1 km: 1. Database creation, properties, and parameters. *Journal of Geophysical Research: Planets*, *117*(E5). http://doi.org/10.1029/2011JE003966

Robbins, S. J., Antonenko, I., Kirchoff, M. R., Chapman, C. R., Fassett, C. I., Herrick, R. R., et al. (2014). The variability of crater identification among expert and community crater analysts. *Icarus*, *234*, 109–131. http://doi.org/10.1016/j.icarus.2014.02.022

Salamunićcar, G., & Lončarić, S. (2008). Open framework for objective evaluation of crater detection algorithms with first test-field subsystem based on MOLA data. *Advances in Space Research*, *42*(1), 6–19. http://doi.org/10.1016/j.asr.2007.04.028

Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., et al. (2019). Lunar crater identification via deep learning. *Icarus*, *317*, 27–38. http://doi.org/10.1016/j.icarus.2018.06.022

Urbach, E. R., & Stepinski, T. F. (2009). Automatic detection of sub-km craters in high resolution planetary images. *Planetary and Space Science*, *57*(7), 880–887. http://doi.org/10.1016/j.pss.2009.03.009

Wang, Y., Di, K., Xin, X., & Wan, W. (2017). Automatic detection of Martian dark slope streaks by machine learning using HiRISE images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *129*, 12–20. http://doi.org/10.1016/j.isprsjprs.2017.04.014