Lunar Crater Detection via Deep Learning and its Application for Age Estimation

Thesis
Submitted to the
Institute of Geodesy and Geoinformation Science
Technische Universität Berlin
Faculty VI - Planen Bauen Umwelt

In Partial Fulfillment of the
Requirements for the Degree of

Master of Science

Submitted by
Waqas Tahir
Matriculation No.: 364181

Examiners
Prof. Dr. Martin Kada
Dr. Philipp Gläser
Valentina Schmidt

March 30, 2020

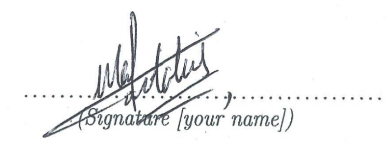
First of all I would like to thank Prof. Dr. Martin Kada at the Institute of Geodesy and Geoinformation science for giving me the opportunity to carry out state of the art research in this field.

Special thanks to Dr. Philipp Gläser and M.Sc. Amgad Agoub for their guidance.

Furthermore, I would also like to thank the support of my family during the this course of study and research.

Hereby I declare that I wrote this thesis myself with the help of no more than the mentioned literature and auxiliary means.

Berlin, 30.03.2020



Abstract

Einschlagkrater sind eines der wichtigsten Merkmale eines Planeten, das Informationen über die Geologie einer Planetenoberfläche liefert. Die Zählung von Kratern bietet auch eine wirtschaftliche Möglichkeit, das Alter einer Planetenoberfläche abzuschätzen. In der Vergangenheit wurden Krater mit Durchmessern von wenigen Metern bis zu Kilometern mit vielen Techniken, die hauptsächlich auf Regeln basieren, entdeckt. Diese Methoden erreichten keine nennenswerte Leistung, weshalb die Kratererkennung und -zählung weitgehend ein manueller Prozess blieb. In den letzten Jahren haben deep learning methoden. die früheren Techniken bei der Objektdetektion übertroffen. Der Schwerpunkt lag jedoch weiterhin weitgehend auf digitalen Höhenkarten (DEMs) der Planetenoberfläche, um Schatten- und Hintergrund-Terrains zu vermeiden. Mit solchen Karten sind kleine metergroße Krater meist unentdeckt. Darüber hinaus ist ein zusätzlicher Schritt der Vorverarbeitung erforderlich, um optische Bilder in DEMs zu konvertieren. In dieser Arbeit wird eine Deep Learning Architektur für die Kraterdetektion vorgeschlagen, die in der Lage ist, metergroße Krater zu erkennen. Das Modell basiert auf Supervised Learning, d.h. es wird ein markierter Datensatz benötigt, um den Algorithmus zu trainieren. Ein relativ kleiner Datensatz, der aus einigen hundert Bildern besteht, wurde für das Netzwerktraining ohne den Einsatz von Transfer-Learning verwendet. Das trainierte Modell wird sowohl auf einen Testsatz aus dem Trainingssplit,als auch auf einen anderen manuell annotierten Datensatz angewendet. Dieser Datensatz gehört zum Landeplatz von Apollo 17 und hat einen anderen Sonnenbeleuchtungswinkel, auf dem der Algorithmus nicht trainiert wird. Die Vorhersagebilder liegen in Form von Wahrscheinlichkeitskarten vor. Zur Extraktion der Krater werden mehrere Schwellenwertverfahren angewendet und ihre Leistung miteinander verglichen. Die besten Ergebnisse wurden mit der Methode von Otsu mit einem F1-Score von 73,8% auf dem Trainingssplit und 64,6% auf dem Bild des Apollo 17-Landeplatzes erzielt. Das Alter wird ebenfalls auf dem Apollo-17-Landestellendatensatz eingezeichnet und mit den manuell gezählten Kratern verglichen, welches die gleiche Altersschätzung ergibt.

Abstract

Impact craters are one of the most important features of a planet that provides information about the geology of a planetary surface. Counting craters also provide an economical way to estimate the age of a planetary surface. In the past detection of craters with diameter range from few meters to kilometers was performed with many techniques which are mainly rule based. These methods did not achieve a notable performance that is why crater detection and counting largely remained a manual process. In recent years deep learning methods have outperformed previous techniques in object detection. However, the focus largely remained on digital elevation maps (DEMs) of planetary surface to avoid complexion of shadows and background terrain. Using such maps small meter sized craters are mostly undetected. Moreover, it also requires an additional step of pre-processing to convert optical images into DEMs. In this thesis, a deep learning architecture is proposed for crater detection which is able to detect meter sized craters. The model is based on supervised learning, that means a labeled dataset is required to train the algorithm. A relatively small dataset composed of few hundred images was used for network training without the use of transfer learning. The trained model is applied on testset taken from training split as well as a different manually annotated dataset. This dataset belongs to Apollo 17 landing site and has different sun illumination angle on which the algorithm is not trained on. The prediction images are in the form of probability maps. Several binarization methods are applied to extract craters and their performance is compared with one another. Best results were achieved by Otsu's method with an F1-score of 73.8% on training split and 64.6% on Apollo 17 landing site image. The age is also plotted on Apollo 17 landing site dataset and compared with the manually counted craters which provides the same age estimation.

Contents

1 Introduction **6**

2 Related Work **10**
2.1 Object Detection 10
2.2 Traditional Crater Detection Methods 12
2.3 Crater Detection via Deep Learning 15
2.4 Planetary Surface Age Dating 17
2.4.1 Hartmann Production Function (HPF) 17
2.4.2 Neukum Production Function (NPF) 18

3 Methods **21**
3.1 Semantic Segmentation 21
3.2 Hough Circle Transformation 21
3.3 Instance Segmentation 21
3.4 Contrast Limited Adaptive Histogram Equalization (CLAHE) 22
3.5 Binarization Methods 22
3.6 Evaluation Method 23

4 Model Architecture **25**
4.1 Convolutional Layer 25
4.2 Pooling Layer 26
4.3 Activation Functions 27
4.3.1 Sigmoid Activation 27
4.3.2 ReLU Activation 28
4.4 Backpropagation 29
4.5 Loss Function 30
4.6 Binary Cross-Entropy 30

5 Experiments **32**
5.1 Dataset 32
5.2 Data Preparation 33
5.2.1 Data Augmentation 34
5.3 Image Normalization 34
5.4 Training 35
5.5 Crater Detection Pipeline 35
5.6 Prediction 36
5.7 Post-processing 36
5.8 Results 36

6 Discussion **47**

7 Limitations and Future Work **50**

8 Conclusion **51**

References **52**

1 Introduction

Analysis of the surface of other planets is a source of learning about our own planet. Studying impact craters present a valuable information about the geology of planets and characteristics of the surface. How crater impacts can affect the life on Earth, climatic change globally and possible consequences of extreme environmental disaster. It is a known fact that solid planet surfaces are covered with tremendous amount of craters of various sizes and shapes [Melosh, 1988]. Largely they form as a result of meteoroid impacts and also considered as windows into the interiors of solid planets [Honda and Azuma, 2000]. Computation of crater density has provided a gateway for the establishment of evolution of planet surfaces chronologically [Martins et al., 2009]. This study provides not only the information about geological processes of the planet but also the history of our solar system. Knowledge of the lunar surface features including craters is also vital for safe landing on Moon [Ivanov et al., 2015]. According to geologists, impact craters are one of the major modifying and surface forming processes for other planets and moons of all the planets [Koeberl, 1994]. Impact craters have also been a useful source of information for planetary scientists in providing the relative age of surfaces. Finding age from crater-counts is a commonly accepted method. Therefore, crater-counts provide a much less expensive way to find out the age of surface as compared to radioactive age-dating. As for the second, a rock sample is needed and age can only be determined for a specific area. Whereas by crater-counts, it is possible to find out the age of a larger area. If the relationship between the crater size and the impact energy is known then craters with larger densities indicate older surfaces [Ivanov, 2002]. Despite the importance of crater and tremendous amount of data available, for decades crater analysis remained dependent on human vision and manual operation [Honda and Azuma, 2000].

Various processes alter crater populations especially craters with smaller diameters of only few meters. This phenomena occurs more often in planets which have dense atmosphere. These processes frequently alter crater size-frequency distributions(CSFD) [Opik, 1965]. Such processes are: deceleration, ablation, fragmentation of meteors while passing through the atmosphere prior to surface impact and postformation modification of craters by erosion and deposition. Therefore, crater counts with relatively smaller diameter (i.e. diameter $<$ 0.01 km) are at a larger risk of representing an age which could be misinterpreted if the modeled production function does not take into account the factors responsible for altering CSFD in an observed range of diameter [Hartmann, 1981]. Moreover, several factors may lead to surface age as well as statistical uncertainties because identification of small craters is prone to certain biases such as resolution limits, illumination effects, compact crater count areas or limited number of craters [Soderblom, 1970].

In case of Earth's Moon, modeling of impact craters depends on the knowledge of age-dated lunar samples with correlation to observed CSFDs [Williams et al., 2018]. Chronology is a system composed of two elements such as a production function describing the shape of CSFD and a chronology function which is related to the accumulation of crater densities to absolute time. Both of these functions collectively provide a predicted CSFD or isochron for a given time length a surface has been under crater strikes. This is valid for the lunar surface since the Moon has no atmosphere. This chronology can also be applied to other solar system objects but with an addition of factor such as surface gravity [Ivanov, 2002]. Two of such production functions are given by Hartmann and Neukum which provide an approximate surface age of the Moon. These functions are given in 'Related Work' section. Crater counts provide the frequency of crater distribution of a certain area. This is usually performed by counting craters manually. This task is not only time consuming but also expensive and labor intensive depending upon the number of images and amount of craters present in images. Moreover, this approach is not practical when dealing with a large amount of images with craters of various sizes on either Moon or any other planet.

Satellites carrying cameras are the source for images which provide opportunity to count craters. Since early 2000 many missions and satellites have been deployed to the Moon for research purposes including crater studies. Lunar Reconnaissance Orbiter (LRO) is a robotic mission operated by the USA that is set out to map lunar surface. It was launched in 2009 and since then it has collected a treasure trove of data. The Lunar Reconnaissance Orbiter Camera (LROC) was specifically designed for the assessment of meter-scale and smaller features to help carry out safety analysis for future lunar landing sites on the Moon including polar region. The Narrow Angle Cameras (NACs) mounted on LRO are able to detect craters with diameters of 2.5 m or greater [Robinson et al., 2010]. It is common to find craters less than 100 m in diameter on the lunar surface [Robinson et al., 2010]. These images can be utilized to count craters.

The LRO image data is available publicly and can be downloaded or viewed online [1]. A single image taken by the LRO requires more than 150 MB of hard disk space. Such images cannot be processed without a competitive machine. There are methods developed to perform image processing tasks which are highly dependent on a machine's memory and performance. Therefore, developing techniques without advancement in computer technology was simply not enough. This problem was already known during the 1960s and therefore it leads to the development of Graphical Processing Unit (GPU). A GPU performs quick mathematical calculations and frees the space for CPU to do other tasks. Unlike CPU, it has thousands of cores designed for multi-tasking. A much needed hardware to perform computationally expensive calculations without exhaustion.

In 2000s many companies like Intel, Nvidia and AMD/ATI stepped into the race of manufacturing faster GPUs and dominated the market. This competition continues till today and GPUs are becoming more and more powerful. Meanwhile, the need for GPUs is also increasing because of big data processing needs. Nvidia introduced a chip (the Ge-Force 3) capable of programmable pixel shaders i.e. compute color, position, depth or stencil of a pixel. A short program could now process each pixel before projecting it to the screen, this processing included but was not limited to the addition of image textures. By 2002 ATI Radeon 9700 the world's first Direct3D 9.0 accelerator was introduced by Microsoft containing support for multiple render targets, floating point texture formats, multiple-element textures and stencil buffer methods. In 2010 Nvidia began a partnership with Audi to power car dashboards. This mainly increased the functionality of navigation and entertainment systems. In 2014 the PS4 and Xbox One were released powered by GPUs based on AMD's Radeon HD 7850 and 7790. Lately RTX was released by Nvidia with the aim of enabling real time ray tracing. This was a new development in computer graphics for generation of interactive images reacting to lighting, reflections and shadows. RTX also includes Artificial Intelligence (AI) integration like enhancing video processing, graphics and images directly into the applications. Today, parallel GPUs are making complex computations in the fields of oil exploration, machine learning, image processing, 3D reconstruction, statistics and even in the stock market for

[1] http://wms.lroc.asu.edu/lroc/

stock rate determinations.

With this advancement in computer processing ability, complex computation tasks are now possible to perform. That means a human labor intensive task of counting objects can possibly be automated but the problem of detecting objects could be complicated specially when object background is complex. It was no longer optimum to rely on manual detection of objects. With big data and challenge of solving tasks in limited time, it was required to develop a mechanism that could solve such tasks faster as compared to expensive manual operations and meanwhile limit man power. This gave rise to research in deep learning approaches to address this problem.

In the past, automatic detection of craters turned out to be a difficult task in cases where rims were overlapped, not clear or if image was noisy [Sawabe et al., 2006]. Multiple automated methods were presented by [Sawabe et al., 2006] using the data acquired by Clementine and Apollo. One of the methods was to thin down a set of edge pixels to one pixel using Hilditch's thinning algorithm. The lines were then connected depending upon the direction and length. If the resultant lines were closed with roundness more than 0.8 then lines were regarded as crater. However, in the last decade, new techniques have been developed in object detection, classification, localization and semantic segmentation. These methods can now be put into application because of state-of-the-art GPUs, which are not only capable of performing computationally intensive tasks but also shorten the amount of processing time as compared to previous versions of GPUs.

Based on satellite images characteristics of a planetary surface can be very complex which makes craters difficult to identify. Satellite images can have noise and that poses a challenge for the detection of degraded features on a planetary surface. High resolution satellite images (like LRO) opened new horizons for research and in the last decade many approaches have been developed to detect craters. These methods range from Generalized Hough Transformation (GHT), Crater Development Algorithms (CDAs) to deep learning. Later was mainly applied from other areas (such as medical imaging and computer vision) for craters detection.

In the past, main focus remained on medium to large sized craters (usually with a minimum diameter of few hundred meters) on digital elevation models (DEMs). This was mainly because in DEMs craters appear in the form of a dark oval or circular shaped object with no additional features such as shadow. Therefore, in this thesis the goal is to detect craters on the actual images with the original resolution. That means small craters which are less than the diameter of 5 m will also be a target. Moreover, considering actual images for crater detection will also eliminate the need for DEMs which makes this method more robust. There has been always a need for an algorithm that is able to detect craters of any size located on any topographic features of a planet. However, detection of small craters has more significance because of their shear number. Large craters (usually diameter of few kilometers) are easier to detect by visual inspection without overwhelming human effort.

Optical images have complexity of shades and background terrain features [Ali-Dib et al., 2019], therefore usage of DEM for crater detection became more popular. Performing this task directly on optical images was largely neglected in deep learning applications [Finkelstein et al., 2019]. In the scope of this thesis, a deep learning method is presented which is capable of performing semantic segmentation of small or large craters directly on actual images. That means to automatically label each pixel of an image to the correct corresponding class. In order to count number of craters and to measure their diameter, various binarization filters are tested in post-processing step which gives the final outcome of detected craters along with their diameters. The performance of these filters is measured in an F-score metric. The CSFD and diameter is then plotted in a logarithmic scale to compare graph with the lunar chronology function for the age estimation.

2 Related Work

2.1 Object Detection

Before wide use of deep learning approaches, different machine learning methods were developed to perform the tasks of object detection. [Viola et al., 2001] proposed a method which was composed of three key components. First one was to introduce a different image representation called Integral Image to compute rectangle features of the object. The second was an algorithm based on AdaBoost with a purpose to select critical visual features from large amount of data to get a set of classifiers. The third was to combine those classifiers in a cascade to discard the background regions and only compute object-like regions. This algorithm used Haar basis features and it was developed to detect faces.

Human detection methods were reviewed by [Dalal and Triggs, 2005] and they came up with the Histograms of Oriented Gradient (HOG) descriptors which experimentally outperformed then existing feature sets including wavelets for human detection. A linear Support Vector Machine (SVM) was used as a baseline classifier. A test was conducted on the MIT pedestrian dataset with mostly upright pose. Later, a more challenging set of 1800 images with different poses and background was introduced to test the performance of algorithm. From the results it was concluded that using locally normalized HOG features in a dense overlapping grid provides better results than Haar-like feature approach for human detection. Another method for training SVM with a latent structure was proposed by [Felzenszwalb et al., 2008]. This method learns the relationships between the HOG features of object parts via a latent SVM. This is a semi-convex training problem but once latent information is specified for positive examples then it turns into a convex training. This approach was tested on the PASCAL dataset and ranked first in 10 out of 20 classes that entered in the PASCAL VOC 2007 competition.

A selective search methodology was introduced by [Uijlings et al., 2013] which combines segmentation and exhaustive search. In this method, the image structure was used to guide the sampling process and the objective was to detect all the possible locations of the object. In this paper, a variety of complementary image partitioning was introduced to deal with as many image conditions as possible. The resultant search method yielded 99% recall and a mean average overlap of 0.879 over 10,000 locations in the testset. Results of this methodology were also evaluated on the PASCAL VOC 2010 and the PASCAL VOC 2012 which shows the mean Average Precision (mAP) of 0.350 and 0.351 respectively.

These algorithms had a noticeable performance but such machine learning techniques had a major requirement which was an involvement of a domain expert to extract features. Moreover, state of the art machine learning techniques also requires a problem to be broken down into different parts and then combine all the results at the final stage i.e. in the SVM a bounding box detection algorithm (localizing by drawing a box on the object) is needed to identify all the objects to have HOGs. Then they are used as an input to the algorithm that will learn to recognize target objects. These limitations gave rise to deep learning algorithms. Later are capable of not only handling large amount of data unlike traditional machine learning techniques but also provide an end to end solution without involvement of a domain expert.

Availability of a large labeled dataset (several thousands of annotated images) and competitive GPUs made it possible to implement a deep learning network and test its performance. A notable development in this field took place when [Krizhevsky et al., 2012] showed that a large deep Convolutional Neural Network (CNN) was able to classify 1.2 million high-resolution images in the contest of ImageNet LSVRC-2010. This network was trained into 1000 different classes. On the test data top-1 and top-5 error rates of 37.5% and 17.0% were achieved which outperformed the previous state of the art results [Krizhevsky et al., 2012]. This neural network had 60 million parameters with 65,000 neurons and it consisted of five convolutional layers. Some of them are followed by max-pooling operation. Then there are three fully connected layers with a final 1000-way softmax activation. A 1000-way softmax because of 1000 classes. Non-saturating neurons were used to optimize training time. A drop-out regularization was used to reduce over-fitting in the fully connecting layers. A variant of this model was also introduced in ILSVRC-2012 competition and a winning top-5 test error rate of 15.3% was achieved compared to 26.2% by the second-best entry.

The best performing methods used to be complex assembled systems that typically combine low-level image features with a high-level context [Girshick et al., 2014]. In 2014, an approach presented by [Girshick et al., 2014] aimed to get better performance on semantic segmentation as well as object detection than other networks at that time. The method is called Regional Convolutional Neural Network (R-CNN) by its authors because it combines regions with CNN features. Detection and segmentation requires localization of objects within an image unlike image classification. This proposed scalable detection algorithm achieved a mAP of 53.3% on the PASCAL VOC 2012. R-CNN was also compared to OverFeat network (a sliding window detector based on a similar CNN architecture) and R-CNN outperformed OverFeat by a margin of 7% mAP on the 200-class Large Scale Visual Recognition Challenge 2013 (ILSVRC2013). OverFeat network had the best result with 24.3% mAP before introduction of R-CNN (achieving mAP of 31.4%).

R-CNN finds a Region of Interest (RoI) from an image, creating a warped image region for all the RoIs and forward it to the convolutional network. Once each of the region is forwarded, bounding box regressors are applied and classification is done by SVM. These processes are done in three separate models in R-CNN. This resulted in slow training. Fast R-CNN introduces several innovations to improve training and testing speed while also improving detection accuracy [Girshick, 2015]. Fast R-CNN takes in an entire image and forwards it to convolutional network to create a feature map. Then it determines RoI and on top of that it applies a single layer of RoI max pooling followed by a fully connected layer. Then softmax classifiers and bounding box regressors are applied. This procedure makes the layer below RoI max pooling trainable. This also makes Fast R-CNN training a single-stage process. It also does not require any disk storage for feature caching. Unlike R-CNN, Fast R-CNN uses a single model for feature extraction from regions then dividing them into different classes and returns boundary boxes simultaneously. Fast R-CNN trains very deep VGG16 (a deep neural network model) about nine times faster than R-CNN. It is also more than two hundred times faster at test-time as compared to R-CNN [Girshick, 2015]. Fast R-CNN achieved 66.1% mAP on the PASCAL VOC 2012.

Fast R-CNN used selective search as a proposal to extract RoI which also had a room for optimization. That lead to further development of Fast R-CNN. The next version of this network is called Faster R-CNN. It uses Regional Proposal Network (RPN) which takes feature maps of an image as an input and generates a set of object proposals [Ren et al., 2015]. Then it gives each one of them the objectness score as output. This process takes place simultaneously. An alternating training procedure was introduced so that RPN and Fast R-CNN can be trained to share convolutional features. Faster R-CNN achieved 70.4% mAP on PASCAL VOC 2012.

The possibility of locating different objects with bounding boxes lead to a question that if this could be extended to locate exact pixels of each object. This problem is known as instance segmentation and was addressed by the framework introduced by [He et al., 2017]. The method is called Mask R-CNN. It extends Faster R-CNN by adding a branch of predicting an object mask in parallel along with existing branch for bounding box recognition [He et al., 2017]. This branch takes input as CNN feature map and outputs matrix with ones and zeros. One, if a pixel belongs to the object and zero otherwise. This output is known as binary mask. Mask R-CNN adds only a small overhead to Faster R-CNN running at five frames per second. It does not loose detection accuracy and has been widely used in computer vision for instance segmentation tasks.

Another architecture which mainly focused on speed rather than accuracy was presented by [Redmon et al., 2016]. It is called YOLO (you look only once) and is capable to process images in real time at 45 frames per second. However, as compare to other state of the art it makes more errors. According to the author it is also less likely to predict false positives on background. Furthermore, YOLO showed better performance than R-CNN. There are two more versions released after first YOLO which mainly focus on accuracy.

2.2 Traditional Crater Detection Methods

Experimentation with pattern recognition algorithms showed that it was possible to automate crater detection task. These algorithms originally developed for particle physics, medical imaging or optical character recognition were applied on planetary science problems such as crater detection. Types of identified features were ellipses, circles, ridges and lines etc. One of these methods is Hough Transformation (HT). It was developed originally for high energy physics by Hough in 1959. Later, Generalized Hough Transformation (GHT) methods were developed and applied for crater detection. One of the application of GHT based method was made by [Honda and Azuma, 2000] for selected lunar images taken by Clementine but this method was not validated for scientific use.

Crater detection and categorization process through data-mining from large scale scientific image database was proposed by [Honda and Azuma, 2000]. The detection module was based on state-of-the-art image processing methods. These techniques included binarization, circular object detection using Genetic Algorithm and HT. The author's method took normalized image vectors, Discrete Cosine Transform (DCT) components and intensity histograms as input vectors. An ellipse/edge detection method based on GHT was developed by [Leroy et al., 2001]. The efficiency of this method was measured as ratio of detected craters to true craters, which was about 20% and was considered acceptable for addressing landing problems. The idea was to detect craters using a multi-scale approach based on voting, and tensors as a representation. This method infer curvature estimation from noisy sparse data. The idea was to obtain the oriented normals of the edge curves by application on the edge images. This system was applied on Phobos to compute a dense saliency map corresponding to the position and shape of the craters. Geological feature cataloging could be performed by labeling images manually but only for limited number of features, handling massive datasets and high resolution images i.e. Mars Global Surveyor, it is required to automate feature identification [Vinogradova et al., 2002]. The authors in this paper used a Continuously Scalable Template Matching (CSTM) algorithm which is an image-matching algorithm. It detected 88% craters with no false alarms. The author used synthetic terrain which was used for systematic validation of this CDA. [Wetzler et al., 2005] applied various machine learning algorithms for cataloging impact craters including bagging, AdaBoost, SVM and CSTM. It was found that all the approaches, in particular SVM with normalized image patches provides detection and localization performance better as compared to boundary based approaches such as HT [Wetzler et al., 2005].

Impact craters remains some of the most studied features in lunar and planetary science. This lead to development of CDAs based on inspiration from pattern recognition methods. CDAs are an important subject of scientific research, as evident form the amount of publications regarding crater detection in the past. This research range from dating of planetary surfaces, searching of still unknown crater impacts on surface of the Earth and safe landing sites on other planets and asteroids. A notable application of CDAs was introduced by [Salamuniccar and Loncaric, 2010] which was based on edge detection and elevation data. The CDA presented was an improvement to previous CDAs and the purpose was to contribute to Martian crater catalog. With each new planetary and lunar mission, the volume of data increases significantly. This justifies research on automatic feature detection methods and importance of their role in processing, archiving, retrieving and interpreting large amounts of image data. [Salamuniccar and Loncaric, 2010] provided an overview of 73 CDA-related publications from numerous authors. Working on CDAs has been a challenging task for many reasons, e.g. multiple possible applications of CDA could provide a good solution for one specific problem on the particular planetary surface i.e. dating [Sawabe et al., 2006] but not necessarily good for other surfaces depending on the terrain features or problems such as autonomous landing on other planets or asteroids [Leroy et al., 2001]. Another challenge of CDA is to distinguish between crater and non-crater objects. Set of features that separates these two, depends on the type of surface, properties of illumination and the shapes and sizes of craters. Existing algorithms mainly focus on large craters (diameter of few kilometers) located on simple surfaces which dictates a specific choice of image features [Bandeira et al., 2010]. Therefore, it is not surprising that still no CDA available is robust as scientific community would like and that fits several research groups. Also CDA not only detects craters but also finds crater candidates that needs to be manually rejected, or corrected with diameter by computer-assistance.

Despite of extensive research in CDAs, no algorithm became a standard tool for planetary science practitioners and crater counts continued to be done via visual inspection even as high resolution datasets keep on increasing [Bandeira et al., 2010]. Crater appearance in an optical image depends on their level of degradation, inter morphologies (such as presence of central peaks, central pits, peak rings and wall terraces etc.), amount of overlap on other craters, quality of image (which includes illumination angle and surface properties) and on size that might differ by the order of magnitude. A notable advancement in CDA was proposed by [Bandeira et al., 2010] which used the approach of applying series of filters for background noise removal and creates a set of features that look for the characteristics of crescent-shaped shadow of a crater. In addition to noise removal, texture recognition was added to improve algorithm precision.

Another approach begins with template matching in which image array pixels are rotated, translated or transformed to match pieces of an image. This establishes an image-based paradigm which can be extended to matched spatial features, principle components methods and Artificial Neural Networks [Brunelli and Poggio, 1993]. An image matching algorithm known as based on CSTM was implemented and tested by [Burl et al., 2001] on selected lunar images. The algorithm uses templates provided by scientists for generation of a model to detect the target in a user specified continuous range of scales. Statistical efficiency of implementation of algorithm was measured on regions of the lunar maria (images provided by Clementine), which was about 80% with 12% false detection rate. Craters less than 5 pixels (20 m) were neglected. However, the reduction in performance was caused by complexity in background terrain for crater detection on the surface of Europa (moon of Jupiter).

For crater detection automation, large focus remained on looking for circular or elliptical shape of edges on crater boundary i.e. HT [Honda and Azuma, 2000]. Boundary based approaches seemed to perform well (relative to image resolution) under certain conditions such as detection of the medium to large craters with limited texture in background. However the performance of these approaches deteriorated with complexity in background terrain or when craters are small. Alternating to boundary based detection methods [Wetzler et al., 2005] proposed to look directly at the pixel-level pattern in an image. The idea was to automate crater detection process by looking for adjacent bright-dark regions of proper relative size. The authors achieved a detection of 60% true craters. There were also 300 false positives.

[Sawabe et al., 2006] proposed an algorithm that did not depend on cameras, spatial resolution or sun illumination. It also did not require to tune any parameters either. This algorithm was improvement to their own previously proposed algorithm. Their previous method did not include pyramid representation of the image which was included in later method and thus improving accuracy and reducing processing time. The algorithm was applied on images acquired by Clementine and Apollo under different solar elevation. This approach was able to detect craters with diameter larger than 240 m. Accuracy rate was validated by comparing with crater count results of [Neukum et al., 1975] which was 80% extraction of craters with multiple interpreters [Sawabe et al., 2006].

Changeability in appearance of craters and surrounding terrain makes imaging based autonomous crater detection difficult to be applied. By experimentation it is proven that unsupervised machine learning methods work well with relatively large craters having clear edge information but their efficiency declines with increase in terrain complexity [Meng et al., 2009]. In 2009 [Martins et al., 2009] performed Viola and Jones (2004) algorithm on Mars dataset gathered by the Mars Orbiter Camera onboard Mars Global Surveyor probe. In this paper author claims that no such method existed that is satisfactory for craters detection. Using this approach craters with diameter larger than 7 pixels were detected. Images of 240 m/pixel were used as a dataset, hence craters larger than 1680 m of diameter were detected. The source of images used are taken from Mars Orbiter Camera.

An automated system for cataloging impact craters was presented by [Stepinski et al., 2009]. The system used DEM of Mars. The process of crater identification consists of two steps, in first step it identifies round and symmetric topographic depressions as crater candidate and second step identifies crater using a machine learning method. This process is AutoCrad system and applicable to any surface represented by a DEM [Stepinski et al., 2009]. A CDA was presented by [Salamuniccar and Loncaric, 2010] which was based on fuzzy edge detectors and it takes input as digital topography data instead of image data. This algorithm claimed to have more correct detections compared to previous CDAs. There were also false positives which were removed manually. The data was taken from Mars Orbiter Laser Altimeter.

2.3 Crater Detection via Deep Learning

Existing approaches to detection techniques can be divided into two categories: supervised (requiring a labeled input data) and unsupervised (fully autonomous) techniques. [Stepinski et al., 2009] has discussed both of these approaches and their usage. Unsupervised techniques are based on pattern recognition approaches to identify crater rims in an image as circular objects. Supervised techniques are the machine learning methods to train a classifier which is then used to distinguish between craters and the other objects. However, in both approaches features are detected by narrowing down to the set of potential candidates. In a supervised method the narrowing is achieved by thresholding a probability of positive detection by a classifier. In an unsupervised approach narrowing is achieved by thresholding a parameter that measure how well the object fits a circle. In general unsupervised approaches tend to be fast and more appropriate for detection of crater with diameter of few kilometers with no worn out edges; however their performance reduces when dealing challenging terrains and smaller sub-kilometer sized craters. This makes unsupervised not a general purpose crater detection technique [Emami et al., 2015]. On the other hand, supervised approaches are more robust but usually slower and requires labeled input data, the performance depends on the quality and number of labeled data samples.

In 2015, a CDA was proposed by [Emami et al., 2015] which took into account the optical images and the algorithm worked by employing a multi-scale candidate region detection step which was based on convexity cues and candidate region was verified via machine learning. In this paper a CNN classifier was tested against SVM and it was concluded that CNN classifier outperforms SVM both in terms of recall and precision [Emami et al., 2015]. The data used for both training and testing was partially labeled by NASA scientists. Each image consists of 600x400 pixels and taken from LRO. The result of their experiment is shown in Fig. 1.

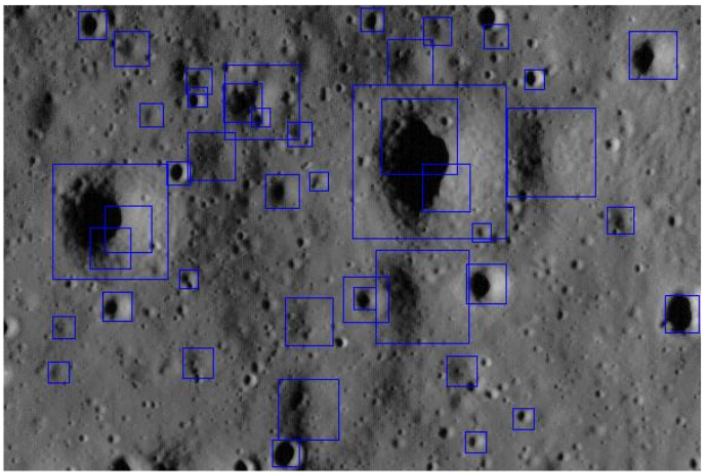


Figure 1: Verified regions in boxes on a test image [Wetzler et al., 2005].

According to [Palafox et al., 2017], Mars Reconnaissance Orbiter opened another frontier to automate landforms detection process. Authors came up with two different approaches. These are detection of volcanic rootless cones and transverse aeolian ridges using CNNs and also using SVM with HOG features. They showed that CNNs can detect a wide range of landforms and has a better accuracy and recall than traditional classifiers based on SVMs. [Wang et al., 2018] presented a method composed of CNN architecture and named it as CraterIDNet which takes remotely sensed planetary images as input and outputs detected craters, their apparent diameters and their positions. The experiments shows that CNN based architecture has the advantages of high robustness, detection and identification accuracy over other methods used by [Urbach and Stepinski, 2009], [Bandeira et al., 2010] and [Ding et al., 2011].

A deep learning model based on CNN architecture was introduced by [Silburt et al., 2019] for crater identification on lunar DEMs. The author also applied transfer learning to craters on Mercury. In this paper random DEMs from LRO and Kaguya were taken as an input data. Thus it was a global gray scale map. The proposed CNN detects only half of the caters per target. The post-CNN recall is lower at 57% $\pm$ 20%. Detections for craters less than 15 pixels of diameter largely improved the post-CNN test recall to 83% $\pm$ 16%. The sample of DEM from [Silburt et al., 2019] is shown in Fig. 2.

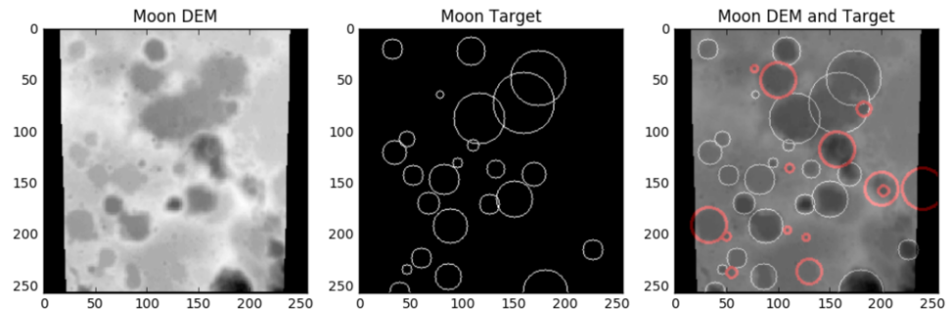


Figure 2: Left is the Moon DEM sample and middle image shows prediction of craters and right one shows the missing classifications which are marked in red circles [Silburt et al., 2019].

[Cohen et al., 2016] demonstrated that ConvNet for crater detection outperformed previously tested methods including CDAs presented by [Stepinski et al., 2009], [Bandeira et al., 2010] and [Ding et al., 2011] on the same dataset. An algorithm based on Fast-RCNN was proposed by [Emami et al., 2015] for lunar crater detection on LRO dataset and it showed great potential in CNN applications for this task. ConvNets are becoming more common in crater detection problems and these are specially of more importance because of the fact that major part of CDAs is not accepted by planetary science community as general purpose crater detection tools [Emami et al., 2015].

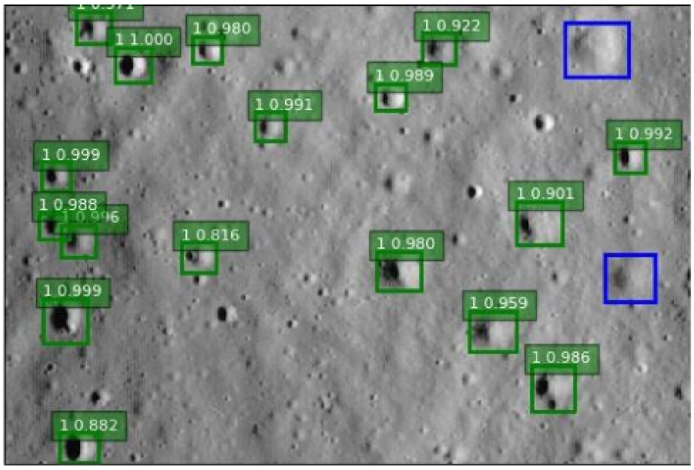


Figure 3: Crater detection results on LRO dataset by using Fast R-CNN based network. Blue squares represent missing craters by the author [Emami et al., 2015].

Recently [Ali-Dib et al., 2019] experimented with instance segmentation method on crater detection problem. The algorithm is well known Mask-RCNN presented by [He et al., 2017]. Model was trained on lunar DEMs to detect craters in an image and simultaneously producing a mask for each crater that also traces the outer rim of it. Then a post-processing pipeline was introduced to find the closest fitting ellipse to these masks. Using this architecture, authors claimed to identify 87% of the craters correctly. However, optical images were not taken into account because the target diameter was ≥ 5 km which justifies the use of DEMs. Moreover, as largely the work in lunar crater detection is performed on DEMs, therefore significant amount of annotations for lunar optical images are also difficult to find.

2.4 Planetary Surface Age Dating

Moon provides an ideal test site for studying crater records, particularly since almost all of the lunar endogenic activities ended more than around 3 Giga years (Ga) with some exceptions [Hiesinger et al., 2000]. Therefore, in last 3 Ga crater impacts have dominated to change lunar landscape. Space missions have also studied the Moon extensively and collected samples from the Moon have provided a unique opportunity to assign age to craters and areas where accumulated crater are counted [Stöffler and Ryder, 2001]. Therefore, on the Moon it can be estimated that cratering rate is the number of craters of a given diameter that are accumulated at given surface during a given time interval.

CSFD reflects the standard distribution of crater impacts. By plotting CSFD against diameter, the age is estimated by the help of lunar production functions. These well known methods are proposed by W. Hartmann and G. Neukum.

2.4.1 Hartmann Production Function (HPF)

Hartmann uses a log-incremental SFD representation with a standard bin size for diameter to represent the CSFD of terrestrial planets. The obtained results are referred as Hartmann production function or HPF. The number of craters per kilometers squared are calculated for a certain diameter range which is $D_L < D < D_R$, this range represents a bin where D_L and D_R are the left and right boundaries of diameter range respectively. The standard bin width is $D_R/D_L = 2^{1/2}$. Findings from most of the lunar mare basalt samples suggests a narrow range of ages between 3.2 to 3.5 Ga [Stöffler and Ryder, 2001]. However, lunar production functions takes into account that crater populations are not effected by any processes. Therefore, if such processes have occurred that changed crater formations then age estimation will be different than the actual. In case of lunar surface, some lava flows on the surface could be younger [Hiesinger et al., 2000] therefore the age variation is represented by a factor of 1.1.

The tabulated HPF is considered reliable for the projectile of a production function because the crater counts from different areas of the Moon are combined and averaged. The incremental form of HPF takes form of a piece-wise three segment power law [Ivanov, 2002].

$$\log N_{2^{1/2}} = -2.616 - 3.82 \log D_L, (D < 1.41\,\text{km}) \tag{1}$$

$$\log N_{2^{1/2}} = -2.920 - 1.80 \log D_L, (1.41\,\text{km} < D < 64\,\text{km}) \tag{2}$$

$$\log N_{2^{1/2}} = -2.616 - 3.82 \log D_L, (D > 64\,\text{km}) \tag{3}$$

The function is represented in Fig. 4. Hartmann chose power law segments in 1960s when this work started. Some of the selections were on the basis of historical reasoning that only the craters branch with diameter range between 1.41 km and 64 km was well established. At that time there were already existing laws of meteoroids and asteroids and Hartmann's attempt was to relate those laws to lunar data.

2.4.2 Neukum Production Function (NPF)

Neukum proposed an analytical function describing the CSFD of lunar impact craters. He wrote a series of publications in description of his function. For summaries, see [Neukum and Ivanov, 1994] and [Neukum, 1983]. Neukum showed that the production function is stable since 4 Ga. The time Neukum proposed this function, a full size crater spectrum was known. His approach was different from Hartmann in a way that he computed a polynomial fit to the cumulative number of craters. Whereas Hartmann proposed a piece-wise exponential equations for his production function. For the time period of 1 Ga, Neukum's production function can be represented as:

$$\log_{10}(N) = a_0 + \sum_{n=1}^{11} a_n [\log_{10}(D)]^n. \tag{4}$$

In above equation D is in km, N is the number of craters with diameters greater than D per squared kilometers per Ga and for values of coefficients a_n see [Ivanov, 2002]. The above equation is valid for crater diameters from 0.01 km to 300 km.

On the basis of age assumption NPF was fit to the crater count. It is notable that both HPF and NPF are a good match for the crater diameter data under 1 km range. However, with D > 1 km, HPF has higher values than NPF and both functions meet again at diameter of approx. 40 km. In Fig. 4 it can be seen that the maximum variation between the two functions is of the factor 3 around the diameter of approx. 6 km. Note that below the diameter of 1 km and between 30-100 km, both of the functions are same. After 100 km both started to decline but HPF declines more rapid as compared to NPF.

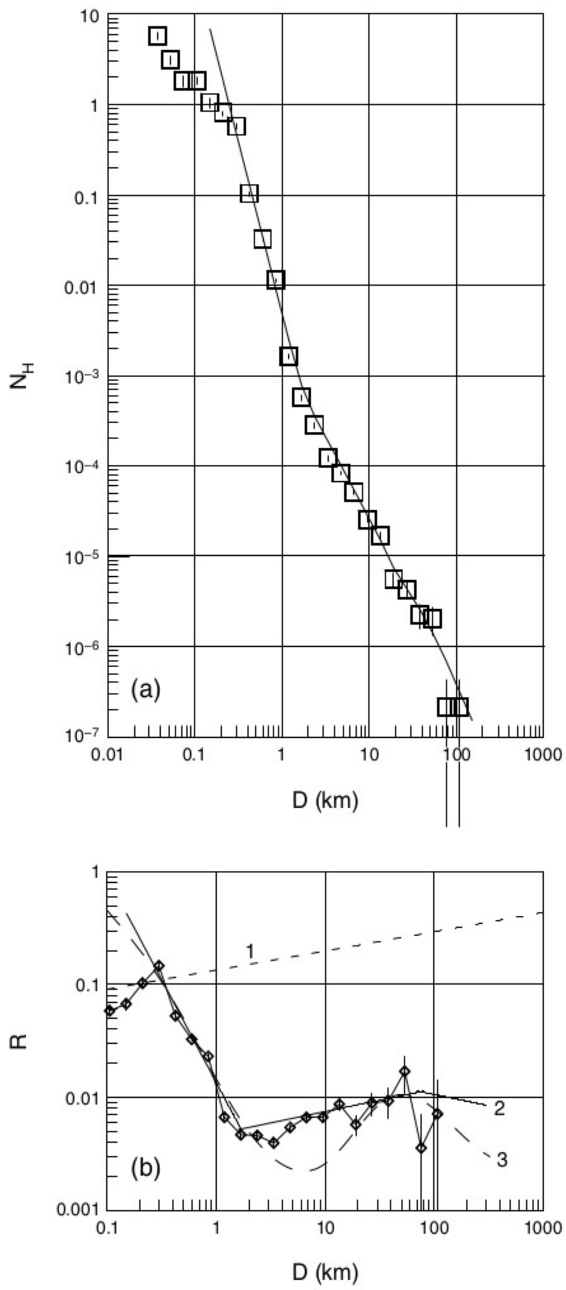


Figure 4: These plots are taken from [Ivanov, 2002]. It shows the representation of the Hartmann production function (HPF). HPF is a function composed of set of points shown in the plot. Straight lines represent the piece-wise power law fitting to the data (Eq. 1). Lower figure shows the comparison of Neukum (NPF) and Hartmann (HPF) in the R plot representation. The maximum discrepancy between HPF (2) and NPF (3) (roughly a factor of 3) is observed in the diameter bins around D close to 6 km. D less than 1 km and diameter range between 30-100 km, both production functions outputs similar results. The dashed line 1 represents the approximate saturation level estimated by Hartmann (1995).

3 Methods

With multiple lunar missions there are millions of images taken by various satellites which are important source of data for crater detection. Deep learning methods require labeled input data for training so that an algorithm is able to detect objects. Performance of these algorithms increases with increase in training samples. According to size and distribution of lunar craters there can be thousands of them in one NAC image of LRO which represents the area of $125\,\mathrm{km}^2$. Generating a large labeled or groundtruth dataset is expensive even though a domain expert is not necessarily required for image annotation. Deep neural networks typically require thousands of images for training such as Mask-RCNN. Often transfer learning is also helpful for loss convergence when training on a new dataset. On the other hand, there are deep learning models that have proven to perform well on small dataset with only few hundred images. These shallow networks can also be trained from scratch on a dataset.

Below are the methods that were applied in an attempt to detect lunar craters on part of one of the NAC images which are part of the training split. The model is also applied for the detection on LRO image of Apollo 17 landing site.

3.1 Semantic Segmentation

Image level classification means treating each image as an identical category. In object detection an object is localized and recognized with respect to type of its class. Whereas semantic segmentation is also called pixel-level classification. This is a task of clustering parts of an image together which belong to the same object class [Thoma, 2016]. It can also be treated as pixel-level prediction because it classifies each pixel of an object into its category. However, this type of segmentation does not distinguish between different instances of an object.

3.2 Hough Circle Transformation

Hough proposed a procedure for line detection in images. This method was extended to GHT for detection of curves in a picture and detailed procedure is provided by [Duda and Hart, 1972]. This has remained one of the widely used methods for circle detection in the field of computer vision.

3.3 Instance Segmentation

Instance segmentation provides not only a pixel-level segmentation to an object but also determines the different instances of objects in an image. This addition makes instance segmentation more challenging than semantic segmentation. In last 3 years neural networks have been introduced that are able to perform instance segmentation and also keep better accuracy as compared to state of the art methods. But a successful training of such deep networks requires many thousands of labeled samples.

In this thesis, the proposed method for crater detection with a small dataset is based on an encoder-decoder architecture. This architecture is a slight variation of a neural network also known as u-net presented by [Ronneberger et al., 2015]. This network is chosen to perform pixel-wise segmentation because of three main reasons; it does not require several thousands of images for training, it does not need transfer learning and can be trained from scratch which makes it robust, it also has a simple network structure which can be easily altered to fit segmentation goals. U-net has also proven to be an effective network for segmenting single class in satellite data [Snuverink, 2017]. U-net was originally developed for segmentation of biomedical images and it achieved an average IoU of 77.5% on DIC-HeLa dataset in ISBI cell tracking challenge 2015 which was the best score at that time. A different version of u-net was applied for Martian craters segmentation by [DeLatte et al., 2019] and 76% accuracy was achieved along with age dating results consistent with human annotations for same geological units. This application demonstrated that CNN offers an advantageous approach to labor-intensive challenging task of satellites image analysis.

3.4 Contrast Limited Adaptive Histogram Equalization (CLAHE)

Histogram equilization considers the global contrast of an image. Therefore in cases where image has high pixel values, such regions tend to loose significant amount of information as those pixels are stretched further towards the value of 255 and object becomes too bright.

By using CLAHE, the image is divided into small blocks also called tiles (function implemented in OpenCV has default tile size of 8x8 pixels). Then histogram equalization is performed on each of these tiles. This makes histogram confined to a small area. It will be amplified if noise is there. Contrast limiting is applied to avoid this problem. The default contrast limit is 40 in OpenCV which is implemented in python library. Pixels with larger values are clipped and distributed to other bins uniformly before application of histogram equalization. After that there are artifacts in tile boarders which are removed by the application of bilinear interpolation.

3.5 Binarization Methods

Several binarization methods are applied to experiment best suited thresholding for probability maps also known as heat maps. These maps represent an image with a range of values from dark to bright pixels. Dark pixels means that it is a background class i.e. no crater whereas bright pixels means that they belong to a potential crater class. Binarization methods that are used in post-processing are given below.

1. Otsu's method.
2. Mean method.
3. Yen's method.
4. Isodata method.
5. Niblack's method.
6. Sauvola's method.
7. Adaptive mean.

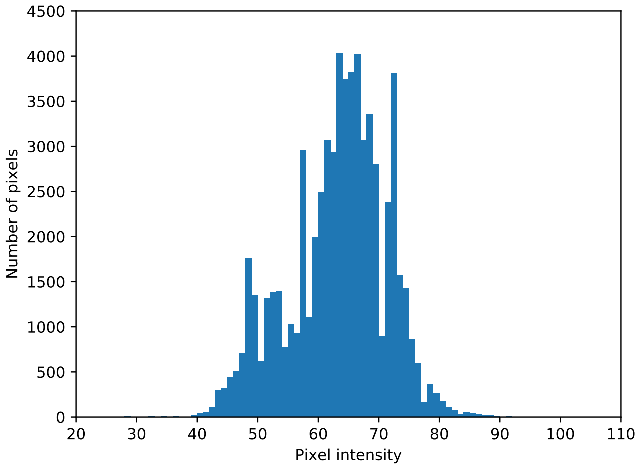
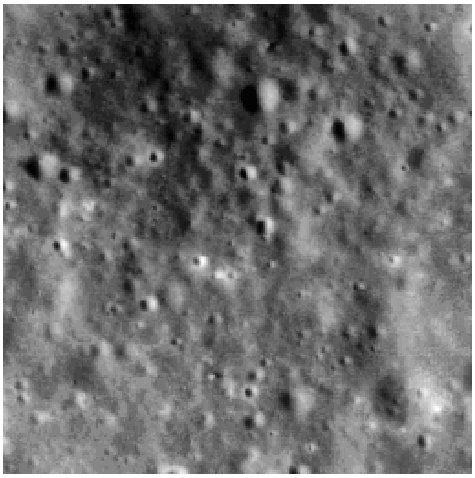


Figure 5: Left is the LRO optical image tile (256 × 256 pixels) of Apollo 17 landing site. Figure on the right is the histogram of left image.



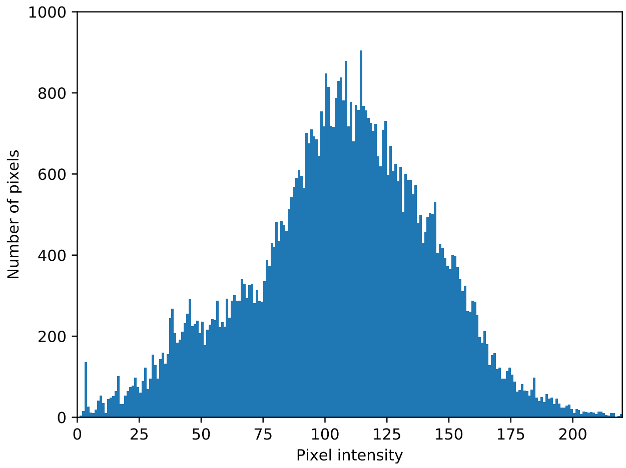


Figure 6: Left is the result of Fig. 5 after application of CLAHE. Figure on the right shows the histogram of image on the left.

8. Li's method

The above mentioned methods are applied and discussed in the 'Experiments' section.

3.6 Evaluation Method

Precision, recall and the F1-score are chosen to evaluate performance of the algorithm. These are well known evaluation metrics in object detection problem because of taking into consideration not only the true and false positives but also the false negatives. Precision is defined as the percentage of results which are relevant and mathematically written as:

$$\text{Precision} = \frac{\text{True positive}}{\text{True positive} + \text{False positive}} \tag{5}$$

True positives are correctly identified craters and false positives are craters detected by the algorithm but no crater exists on that location.

Recall is defined as the percentage of total relevant results correctly classified by the algorithm and is given by:

$$\text{Recall} = \frac{\text{True positive}}{\text{True positive} + \text{False negative}} \tag{6}$$

In the above equation false negatives are simply missed craters by the algorithms. For overall evaluation, both of these metrics are taken into account and finally F1-score has been computed by the given equation:

$$\text{F1-score} = 2 * \frac{\text{Precision} \ * \ \text{Recall}}{\text{Precision} \ + \ \text{Recall}} \tag{7}$$

Hence higher the F1-score better is the performance of algorithm and vice versa. Depending on the type of data and following the experimentation results of [Emami et al., 2015], it is chosen that a verified region is a true positive if it has more than 40% IoU with a ground truth crater; otherwise, it is a false positive. IoU is calculated by the ratio of area of intersection over union.

4 Model Architecture

The model architecture is composed of encoding and decoding paths as shown in the Fig. 7. Like in a typical CNN, the image becomes smaller as a result of convolutional operations. U-net has the same effect and thus left part of this pipeline is also referred to as encoding path. It consists of several convolutional operations such that each of two unpadded convolutional layers with an activation of Rectified Linear Unit (ReLU) are followed by a layer of max pooling of the size 2×2. This results into downsampling of the image. During each downsampling step the number of feature channels are doubled as seen in the Fig. 7. This part ends up with a dropout function before it starts expanding.

The right part of Fig. 7 illustrates the decoding path where every step consists of upsampling of feature map which is followed by a 2×2 convolutional operation (also referred to as up-convolution). This halves the number of feature channels. The gray arrow shows the concatenation of corresponding feature map from encoding path where it is cropped from the image. It is then subjected to two 3×3 convolutions and each followed up by a ReLU activation function. The cropping is required because of loss of boarder pixels during convolutions. At the final layer a 1×1 convolution maps each 64 component feature vector to the target number of classes i.e. background class and the foreground class. The final layer follows up by sigmoid activation function which determines the output of a class score between zero and one. Finally, binary cross entropy determines the loss while training of the algorithm. In this model the total number of convolutional layers amounts to 23. It is important to select the input image size so that max pooling operations are applied to a layer with the same x and y size. This allows seamless tiling of segmentation map.

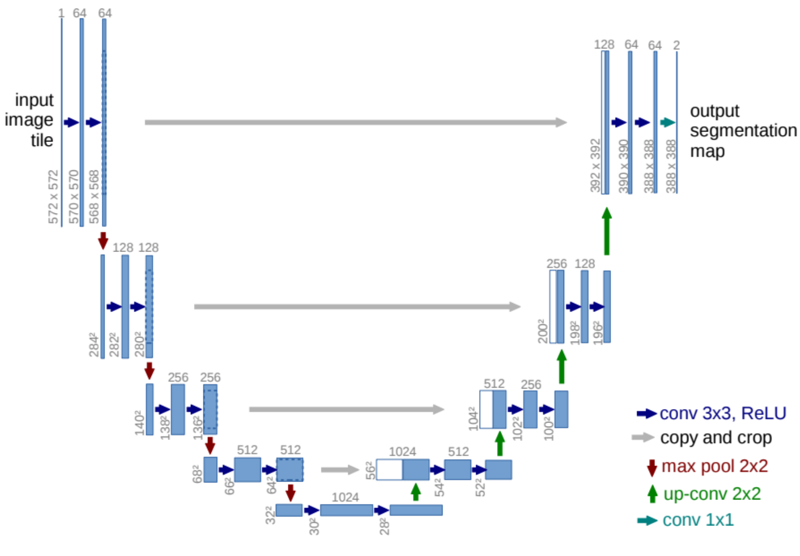


Figure 7: U-net Architecture [Ronneberger et al., 2015].

4.1 Convolutional Layer

Convolutional layers essentially extracts a feature map from images. Images are mathematically represented by matrices with three color channels as red, green and blue (RGB). Gray scale images have only single channel. Therefore an image has a size $h \times w \times d$ where d is depth represented by the number of channels. Convolutional layers includes a filter (also referred to as kernel) which is also composed of $f_h \times f_w \times f_d$. Filter has a height and width smaller than the image size. It slides over (convolves with) the image producing a feature map. This convolution is the sum of element-wise multiplication of filter with the image. It is to be noted that depth of the filter is same as that of the input image therefore it varies with network.

Filter stride is a parameter that needs to be defined before training in convolutional layers. It determines the number of pixels by which a filter shifts at a time. Convolutional layers tend to reduce the output mapping size. A larger stride size will also result in a smaller sized output. Eqs. 8 and 9 shows the relationship between output and the input size of an image with a stride s and filter f. The size of feature maps decreases by increase in convolutional layers. Row (O_x) and column (O_y) output size of convolutional layers are calculated as:

$$O_x = \frac{i_x - f_x + 2p}{s} + 1, \tag{8}$$

$$O_y = \frac{i_y - f_y + 2p}{s} + 1 \tag{9}$$

Assuming example of an input image with size (256×256) and a filter size of (3×3) having a stride s of one and zero padding p will result into an output size of (254×254). By using additional filters n, the feature map will be of size $(254 \times 254 \times n)$. So, addition of filters will increase the output depth of a convolutional layer.

Example shown in Fig. 8 represents a typical convolutional operation without padding and stride value of one. The output 429 in 4×4 matrix is obtained by addition of element wise multiplication of the filter with top left 3×3 portion of the input image. Then filter jumps to the next pixel and other values are obtained the same way.

INPUT IMAGE

18	54	51	239	244	188
55	121	75	78	95	88
35	24	204	113	109	221
3	154	104	235	25	130
15	253	225	159	78	233
68	85	180	214	245	0

FILTER

1	0	1
0	1	0
1	0	1

OUTPUT IMAGE

429	505	686	856
261	792	412	640
633	653	851	751
608	913	713	657

Figure 8: Convolutional Operation

4.2 Pooling Layer

An input image could be large which increases the amount of parameters and introducing a pooling layer helps to reduce the number of parameters. Most commonly used type of pooling is max pooling. Fig. 9 shows an example of max pooling with stride and filter size of 2. The idea is to keep the high values in each quadrant because the highest number represents a particular feature and in the example shown in Fig. 9, number six is the highest value in this quadrant. It means that the most activated pixel in this quadrant is six and same goes for other quadrants. The high values are preserved and lower ones are dropped out which are not as activated. Another pooling layer type is average pooling where averaged output of all the pixels is preserved. As it can be seen in the example below that pooling has reduced a 4×4 matrix to just 2×2 matrix, significant amount of parameters are reduced, in addition pooling may also help in reducing overfitting.

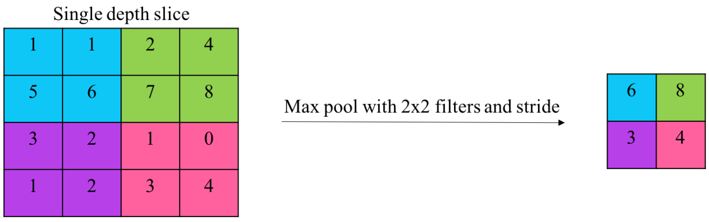


Figure 9: Pooling Operation

4.3 Activation Functions

These functions are an extremely important feature of a neural network. These functions decide which neuron (a neuron is a mathematical function) would be activated. This means whether the information received by a neuron is relevant or should it be ignored. An activation function performs a non-linear transformation on the input signal and forward it as an input to the next layer of neurons.

Without activation functions, weights and biases would be simply doing a linear transformation and a linear equation is easy to solve but very limited to the capacity of solving complex problems. Therefore, a neural network without having an activation function is simply a linear regression model which will not be capable of learning or performing complex tasks. Image classification or object detection is a complicated task and would require non-linear transformations. These functions make the process of back propagation possible because of the receiving gradients and error which are a measure to update weights and biases.

4.3.1 Sigmoid Activation

It is defines as:

$$a = \sigma(z) = \frac{1}{1 + e^{-z}} = \frac{e^z}{e^z + 1} \tag{10}$$

where

$$z = Wx_i + b \tag{11}$$

W being weight vector and x_i is image vector, b stands for bias

The maximum output of this function is one and minimum is zero. Output always lies between values zero and one. Graph of sigmoid function can be viewed in Fig. 10

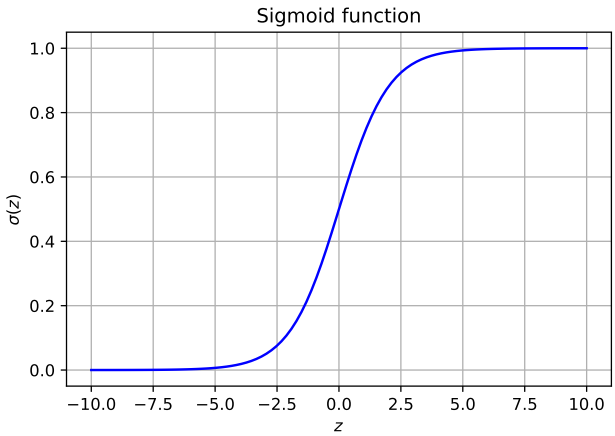


Figure 10: Sigmoid Function

From the graph it can be seen that z is zero when curve is passing though 0.5 of $\sigma(z)$. In convention, a rule can be set if i.e. z is greater than 0.5 then output is always one and if less than 0.5 then it is zero. In other words, if z is larger than 2.5 in graph then the derivative of this function is close to one and same way if it less than -2.5 then it is equal to or close to zero.

This means that if z does not fall inside both limits i.e. -2.5 and 2.5 then it can slow down gradient descent. It also becomes a source of vanishing gradient problem. If the weights are initialized with either very large or very small values then these values saturate the input to sigmoid at a limit region (i.e. close to zero or close to one). Even if weights are initialized at a value i.e. ~ 0.2, in a deep network with many layers it will also lead to vanishing gradient problem. In case of only 4 layer network $0.2^4 = 0.0016$ which is small and will get even smaller in next layers, hence it will lead to vanishing gradient problem.

Sigmoid is used for a problem where binary classification is required. In a neural network layer sigmoid must not be used in every layer because of the problems mentioned above. However, where binary classification is required, it can be useful in the last layer of the network to squash output such as $1 \geq \hat{y} \geq 0$.

4.3.2 ReLU Activation

Rectified Linear Unit is defined as:

$$a = \max(0, z) \tag{12}$$

The graphical form is given in Fig. 11. As seen in the graph, ReLU gives an output for a positive value otherwise the output is zero. It is to be noted that in graph the line looks linear but ReLU is non-linear in nature and it is one of most popular functions used in neural networks because of its simplicity and ability to not let all the neurons fire at once. ReLU is computationally much faster than sigmoid or tanh (written as $a = \tanh(z)$), it does not have exponent computation in such as sigmoid activation function therefore reduces training time

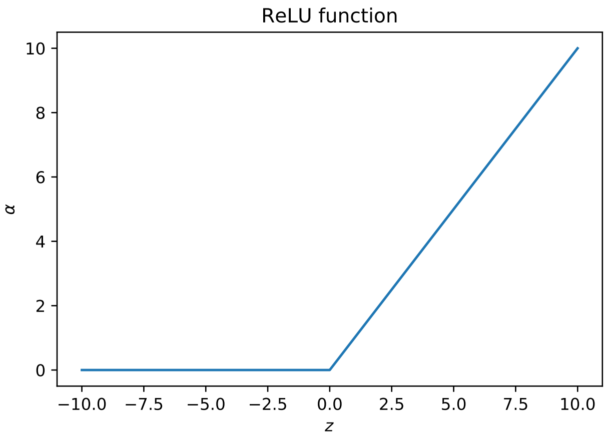


Figure 11: ReLU Function

significantly in very deep neural networks. [Krizhevsky et al., 2012] observed that training deep CNNs with ReLU is much faster as compared to sigmoid or tanh.

Unlike sigmoid when the receiving input is at the right or left plateau i.e. less than -2.5 or greater than 2.5 in Fig. 10 which makes it meaningless to pass a backward pass because of derivative being closer to zero, ReLU only saturate when the input is a negative value. ReLU allows training of larger nets at much less computational costs which means more parameters can be trained at the same computational cost.

4.4 Backpropagation

A CNN requires to update its weight for a given training data in order to reduce loss. Backpropagation is an efficient method for computation of gradients which are required for gradient based optimization of weights or kernel parameters in neural networks [Rumelhart et al., 1988]. This optimization problem refers to minimize the loss function which is performed by the specific combination of weights. Backpropagation requires the computation of loss function at each iteration therefore loss function should be differentiable and continuous.

Initial weights of an untrained network are randomly taken. Without training, a network is not able to make meaningful predictions for an input as there is no relation between an input image and its groundtruth output yet. The weights in a network are adjusted by exposing a network to training samples which are labeled according to the correct class. Before backpropagation there is a forward pass in which an image is taken into the network and first layer of network computes a feature map which is a very low level feature map. Then this activation map is fed to the next layer usually a hidden layer that computers another activation map with slightly high level features than the previous layer and this goes on till the last layer yielding a network output. This results into a feature map which is determined by a loss function that how much different it is from the labeled feature. During backpropagation step of training the aim is to adapt weights in a way that difference between network output and desired output is minimized so that correct features can be extracted from an input image. There are usually multiple epochs till weights are adjusted so that loss is minimized. Backpropogation works on derivation chain rule to minimize loss function and all weights are updated in the negative direction of gradient function. A gradient is a vector containing derivatives. It is computed using partial derivatives and produces a vector field unlike a derivative which is dependent on a single variable. A Jacobian matrix represents gradient. Optimization algorithms such as Adam optimizer minimize or maximize loss function using its gradient values with respect to parameters.

Assuming a multiplication function of two numbers i.e. $f(x, y) = xy$, it is possible to derive the partial derivative for either of input.

$$f(x, y) = xy \quad \rightarrow \quad \frac{\partial f}{\partial x} = y \quad \frac{\partial f}{\partial y} = x \tag{13}$$

The derivation function of a variable indicate the rate of change of a function with respect to that variable surrounding an infinitely small region near a specific point.

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h} \tag{14}$$

In above equation the operator $\frac{d}{dx}$ (a derivative operator) is applied to a function f. The resultant is a derivative. The sensitivity of an expression on a value is determined by the derivative. If $x = 4, y = -3$ then $f(x, y) = -12$ and this would return a derivative on $x\frac{\partial f}{\partial x} = -3$. It is clear that if value of this variable is increased by a tiny amount, it would create an impact of three times decrement because of its negative sign. It can also be seen by rearranging the above equation such that; $f(x+h) = f(x) + h\frac{df(x)}{dx}$. In other way as $\frac{\partial f}{\partial y} = 4$, it is also expected that by increasing the value of y by a tiny amount which is h would also increase the function output by $4h$ because of positive sign.

4.5 Loss Function

It determines the amount of deviation from the groundtruth or labeled data of the algorithm. Higher loss means the actual outcome is very different than expected result. A high loss function indicates poor performance of the model. If training is carried out in set of batches then a loss function is able to define the average of losses for individual training samples.

There are different loss functions which are mainly chosen according to the type of problem. For lunar crater detection, binary cross-entropy was used because it is a binary classification problem.

4.6 Binary Cross-Entropy

As the name suggests, it is a default loss function used for a binary classification problems. It is used where target values are zero or one. It calculates a score that summarizes the average difference between predicted and actual probability distributions for a predicting class. An ideal value for a cross entropy loss function is zero. Mathematically representation is given as:

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log\left(p\left(y_i\right)\right) + \left(1 - y_i\right) \cdot \log\left(1 - p\left(y_i\right)\right) \tag{15}$$

Where y is the label and $p(y)$ is predicted probability for N points.

5 Experiments

The experiments for crater detection are largely performed with u-net based architecture. However, a method of instance segmentation (Mask R-CNN) was also tested. As u-net does not provide the intended output of crater counts and diameters, images are binarized according to various thresholding methods and for final outcome regional proposal algorithm was used on binary images. HCT was also applied to find amount of craters directly from probability maps. Given below are layers of the u-net model used for training and generating predictions from the trained version of it.

Table 1: Layers in u-net

Layer type	Comments
Encoder	
Conv2D	Zero padding, ReLU activation
Dropout	Value of 0.5
MaxPooling2D	
Decoder	
UpSamplig2D	Size 2
Conv2D	
Dropout	Value of 0.5
Add	Adding the result of previous dropout with the correspondingly sized Conv2D layer from encoder part
Final Layer	
Conv2D	Zero padding with sigmoid activation

5.1 Dataset

There are two different datasets taken into account for model evaluation. First dataset which is in the form of gray scale lunar image is taken from LROC archive.The image selected from this data source is of the size (5064 × 52224 pixels) with a pixel representing a length of 1.009 m. This means processing these images would require a competitive hardware. Also each image in this data has several thousands of craters with sizes ranging from few meters to several kilometers. Keeping in view the size of an image and amount of craters present in an image, it was considered to take one of the image entitled as ”M1111897809LE” (can be easily located in LRO database) and crop it into several small images of equal size and width (256 × 256). This image has solar longitude angle of 52.81°. This cropping dimension was chosen based upon the ease for annotation process. Smaller image size would mean less craters to annotate in a single image and therefore more number of images in training set which would make data handling such as pre-processing and post-processing simpler. The amount of cropped images are 320 out of which 240 are used for training and 60 are used for validation. For testing 20 images are used with craters diameter in one image tile ranging from 1 m to 120 m.

Second dataset is taken from [Polegubic, 2020] who annotated an LRO gray scale optical lunar image with 21.36°solar longitude angle of the Apollo 17 landing site. The image is of the size 3130 × 2427 pixels with one pixel representing a length of 0.51 m. This image is cropped into several images of the size 256 × 256 for algorithm's prediction. Tiles of this size are created to align this data with the testing data from the first dataset a part of which was also used for training.

5.2 Data Preparation

For data preparation, annotation is one of the most important tasks in computer vision problem related to deep learning. This task is done manually, images are annotated and assigned a class. These are established by the annotator who can perform this task on the basis of simple instructions. No skilled labor is required to do this task. These annotations carry information to the neural network about the target features in an image. In this data. there is only one label which is "crater" and has shape of a circle, whereas rest is the background class.

Annotation or labeling could be laborious and time taking task. It is specially required when a network is not trained on the same type of class as required for predictions. The annotations for lunar optical images are performed by me for preparation of the dataset. There are several annotation tools and after performing few experiments with various tools, VGG Image Annotator was found most simple and faster to annotate lunar crater data. Once the labeling is complete, a file represents the craters with a center point as x and y coordinates and radius respectively.

In Fig. 12, images shown in a row are part of the training dataset and Fig. 13 are the respective annotations. Images marked with circles on craters are not intended to be used for the training of algorithm, those are only visualizations to labeled data so that wrong annotations could be removed or missing ones could be added. The result of these annotations is projected to get the masks of craters. These masks are binary images that represents the black and white pixel values of images. White pixels means craters and black means background class as shown in the Fig. 13. These provide learning objective to neural network that show the shape looks like and what features to learn. These binary masks have pixel values either zero or one and are of the same shape along x and y as of the respective images.

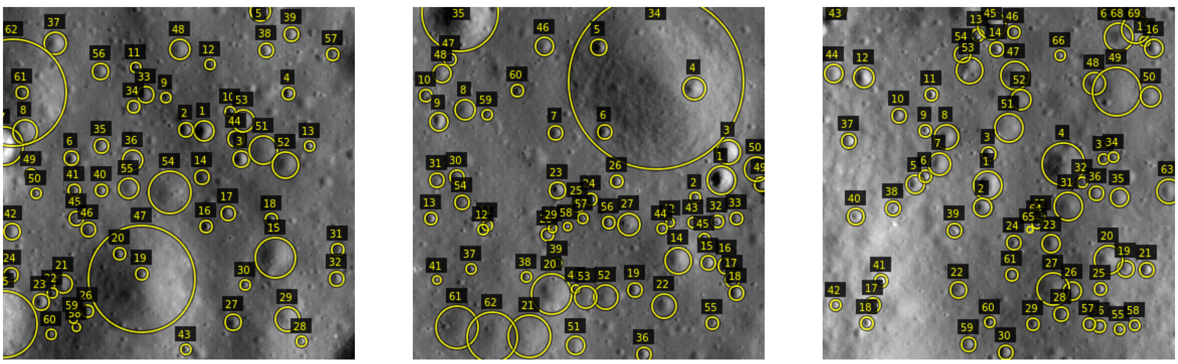


Figure 12: Example of annotations performed on cropped images. Annotation is performed using VIA tool.

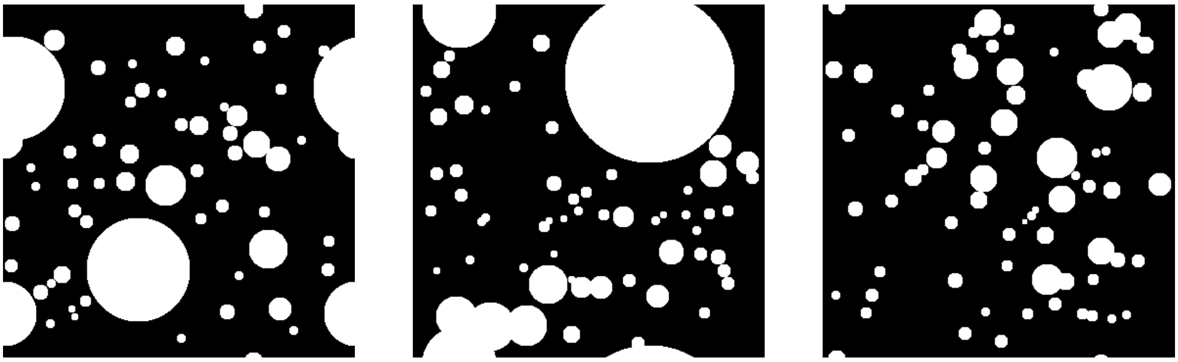


Figure 13: Visualization of binary masks after projection from annotated images that belong to Fig. 12.

5.2.1 Data Augmentation

Reducing overfitting can be achieved through data augmentation. For many of the vision problems, generic input transformations like rescaling, cropping, adjusting colors or addition of noise are often helpful and may substantially improve generalization [Dvornik et al., 2019]. Development of more elaborate strategies requires prior knowledge of the task. For instance, all categories in ImageNet or Pascal VOC datasets are invariant to horizontal flips (e.g, flipping an airplane is still an airplane). But flipping will not be meaningful for the MNIST dataset, for example a flipped '5' is not a digit. Typically augmentation is achieved by creating new images with objects placed at various positions in existing scenes and this strategy of random placements has proven surprisingly effective for object instance segmentation [Dwibedi et al., 2017] which focuses on retrieving instances of a particular object. Whereas in semantic segmentation the focus is on distinguishing categories rather than object themselves. For such tasks the random placement technique simply does not work and was proven through experimentation by [Dvornik et al., 2019].

Specifically for lunar optical image augmentation, training image is randomly augmented in following ways:

1. Rotation: Random rotation of images from 0° to 360° angle.
2. Flipping: Image is flipped with a factor of 0.5 as probability.

The images are randomly perturbed during training so that augmentation can have maximum effect. This way model have a low probability of seeing same type of training image more than once. In lunar crater dataset there are no RGB images hence color augmentation is opted out.

5.3 Image Normalization

Normalization is a key step in the pre-processing pipeline for any deep learning task. Normalization is also very important for lunar images and there are a variety of methods for doing it. The aim of normalization is to remove heavy variation in data that does not contribute to the prediction process and instead accentuate the features and differences that are of most importance. Data normalization is an important step which ensures that each input parameter (pixels in this case) has a similar data distribution. This makes convergence faster while training the network. As images are comprised of matrices with pixel values. Gray scale images are single matrix of pixels unlike colored images with three color channels. Normally pixel values range between 0 and 255. This raw format can also be used as an input to the neural network but this will increase amount of parameters that will result into slower training. Instead there is a great benefit in normalizing these pixel values to the range [0,1] to centering and even standardizing values. Lunar image data is normalized by subtracting the mean from each pixel and then dividing the result by standard deviation. The distribution of such data would resemble a Gaussian curve centered at zero. For image inputs, the pixel numbers should be positive, so the chosen scale to normalize each image pixel is in the range of [0,1].

5.4 Training

The data composed of images and their respective masks is divided into training ($\sim 80\%$) and validation ($\sim 20\%$) dataset. This percentage practice is typical in machine learning, however it does not have to be always at this rate. In case of large dataset i.e. several thousands of images, 60% or even less can be set for training.

The input images along with their corresponding segmentation maps are utilized to train network with Adaptive Moment Estimation (Adam) implementation of Keras (originally written in Cafe [Ronneberger et al., 2015]). Inside training pipeline, the convolutions are unpadded which leads to a smaller output size after every layer. This decrease in size is of the factor of constant border width. The batch size is reduced to a single image to minimize the overhead and utilize maximum GPU memory. In this configuration momentum is kept high (0.99) such that a larger number of already seen training samples determine the update in ongoing optimization step. Momentum helps to accelerate gradient vectors in the right direction. A softmax function over the final feature map combined with cross entropy loss function determines the energy function. In softmax, which is defined as $p_k(\mathbf{x}) = \exp\left(a_k(\mathbf{x})\right) / \left(\sum_{k'=1}^{K} \exp\left(a_{k'}(\mathbf{x})\right)\right)$, where $a_k(\mathbf{x})$ denotes the activation in feature channel k at the pixel position $\mathbf{x} \in \Omega$ with $\Omega \subset \mathbb{Z}^2$. K is the number of classes and $p_k(\mathbf{x})$ is the approximated maximum function. This means that $p_k(\mathbf{x})$ will be close to one for k with maximum activation and vice versa for other values of k. The cross entropy function penalizes each deviation of $p(y)$ from one as defined in Eq. 15.

The weights are saved in a checkpoint when losses are less than previous saved checkpoint. Any of these saved weights could be utilized to apply on a test dataset but definitely saved weights with best performance on validation dataset is the best choice to be deployed on a test dataset.

5.5 Crater Detection Pipeline

After training, to get detected craters from raw images, LRO images are split into 256×256 sized tiles which goes through the following.

1. Each 256×256 tile is fed to the segmentation network.

2. Prediction results are stitched back together into the full sized original image. At this stage the resultant is a probability map which is also an image and can be compared to the target input image.
3. To compute the radii and locations, binarization methods are applied to convert probability maps to binary images.
4. The region proposal algorithm described by [Reiss, 1993] is used for computation of locations and radii from binary images.
5. The model performance is evaluated by the F1 measure on test data from training split and another dataset annotated from [Polegubic, 2020].
6. Neukum production function is used to plot CSFD against crater diameters and lunar age for the test data surface is estimated.

5.6 Prediction

The best checkpoint of trained model is used for prediction on testset. In this case the checkpoint with minimum loss on validation set is the best performing version used for prediction on lunar images. First, predictions are performed on the lunar images without histogram equalization. Then lunar images are subjected to CLAHE and predictions are performed once again. The difference in performance can be viewed in results section.

Predictions yield a probability map of each image in which light pixels depict maximum likelihood and darker ones represents low probability of crater existence. Such heat map of one of the test images is shown in Fig. 14 (b). Trained model is used for prediction on two testsets. First one is similar to training dataset but test images are not seen by the algorithm during training or evaluation. Second testset is an LRO optical image of the Apollo 17 landing site and is annotated manually for the craters by [Polegubic, 2020].

5.7 Post-processing

The resulting predictions from u-net model which are in the form of probability map are post-processed, that means images are first binarized (converted to white and black pixels) by applying various binarization methods which are entitled in binarzation section. This is needed in order to count number of craters and their diameters. Region proposal algorithm proposed by [Burger et al., 2009] is used for finding x, y coordinates and diameter of craters along with the location in an image. This procedure is repeated for the LRO lunar optical image of Apollo 17 landing site. The age of this lunar region is already known. Therefore, CSFD is computed by extracting craters in combination with the u-net, binarization methods and region proposal algorithm. Then CSFD is plotted against crater diameters in a log-log plot to estimate age by comparison with Neukum's lunar production function.

5.8 Results

The trained model is used to generate predictions on both the testsets. The resultant images are in the form of probability map but craters can not be extracted directly from prediction images. One of the such image is shown in Fig. 14 (b). In order to extract craters, these images are binarized. It is a segmentation such that an image is divided into constituent objects. This process works by finding a threshold value. This value divides an image histogram into two parts. A white part represents the foreground class or craters in this case and black part presents the background class.

Otsu's method determines a threshold value automatically. In probability map of lunar images, the histogram has two peaks. Otsu in simple words takes the approximate middle value of the two peaks. This is unlike a global thresholding where an arbitrary value is normally chosen. This method works on the principle of minimizing the intra-class variance, defined as weighted sum of variances of two classes. Given below is mathematical representation of this method and Fig. 14 (c) shows the outcome.

$$\sigma_w^2(t) = \omega_0(t)\sigma_0^2(t) + \omega_1(t)\sigma_1^2(t) \tag{16}$$

Here ω_0 and ω_1 are the probabilities of two classes which are separated by a threshold t. σ_0^2 and σ_1^2 represents the variances of two classes.

Mean thresholding as shown in Fig. 14 (d) is computed by taking mean of the gray scale values of the prediction image (Fig. 14 (b)). In this method all the values higher than this mean (which is a float number) are considered to be foreground and rest is background.

Yen's thresholding method is based on the consideration of two factors. One is the variance between the thresholded and the original image. Second one is the number of bits required to represent the thresholded image. Yen proposes a maximum correlation criterion for defining variance and the cost function and takes both of these factors for multilevel thresholding. The threshold values are the identified by minimizing the cost function. For mathematical derivation and further reading of Yen's method, see [Yen et al., 1995]. Fig. 14 (e) shows the outcome on the prediction image (Fig. 14 (b)).

Isodata thresholding is histogram based threshold also known as inter-means or Ridler-Calvard method. Threshold is computed by separating the image into two groups of pixels. The resultant threshold lies in the midway between mean intensities of these two groups. That is average of the two. It was proposed by [Ridler et al., 1978]. Fig. 14 (f) shows the outcome of this method on the prediction image (Fig. 14 (b)).

Niblack's method was developed to preserve minute details while segmentation. It introduces a concept of local window. This method computes the threshold by calculating local mean and standard deviation of pixel values in a local window confined to an image. See Eq. 17 for the mathematical representation.

$$T_d = m(x, y) + k \times s(x, y) \tag{17}$$

where $m(x, y)$ is local mean and $s(x, y)$ is the standard deviation and k is an image dependent parameter selected manually by the user (normally -0.2 for dark foreground and +0.2 for dark background). The resulting segmentation performed on Fig. 14 (b) can be viewed in Fig. 14 (g).

Savoula proposed to compute a threshold of a gray scale image at each pixel using:

$$T = m \times \left[1 + k \times \left(\frac{s}{R} - 1\right)\right] \tag{18}$$

Here k is user defined parameter which was taken as default implemented in scikit-image python library, m is mean and s is the local standard deviation computed in a certain window size which is centered on current pixel and R is the dynamic range of the standard deviation. $R = 128$ with 8-bit gray scale images. This makes the method sensitive to background features. Result of this algorithm on Fig. 14 (b) can be visualized in Fig. 14 (h).

In adaptive mean thresholding, algorithm computes the threshold for a pixel on the basis of a small region around it. This results into various thresholds for different regions of the same image. This method is implemented in OpenCV python library. Outcome of this method on Fig. 14 (b) can be seen in Fig. 14 (i).

Li's method is based on minimum cross entropy. It select a threshold that minimizes cross entropy between thresholded and original image. It was presented by [Li and Lee, 1993]. Application of this method on Fig. 14 (b) can be seen in Fig. 14 (j).

The evaluation in terms of F1-score of the u-net model on the LRO image of Apollo 17 landing site is given in Table. 2. The dataset consists of a single image of the size 3130×2427 pixels which is annotated by [Polegubic, 2020]. It is cropped into several 256×256 tiles. Each pixel represents the resolution of 0.51 m. Then images are subjected to CLAHE. Prediction is performed on all the images and resulting tiles are stitched together in an order similar to the original image. The binarization methods presented in Table. 2 are applied to convert the probability maps into the binarized images. Then region proposal algorithm is applied to count the number of craters, their x, y coordinates and respective diameters. Many of the outliers are of the diameter of 1 pixel therefore, all the detected craters of 1 pixel (0.51 m) are filtered out. Then precision, recall and F1-score is calculated by using Eqs. 5, 6 and 7 respectively. A crater is considered as a polygon. These polygons are implemented by using shapely, a library of python. Then IoU is computed based on polygons. If the IoU of detected and predicted polygon (crater) is greater than 40% then it is taken as a true positive. This percentage is taken after the experiments of crater detection presented by [Emami et al., 2015] and also by visual inspection of craters. This IoU is true for all the scores shown in Tables 3, 4 and 5.

The precision, recall and F1-score on 20 test images from the training split are shown in Table. 3. The testset is not seen by the u-net and is subjected to CLAHE before performing prediction. F1 computation is performed on each image separately then score of all the images is averaged. Given filters are chosen on the basis of their performance as indicated in the Table. 2. Low performing methods according to Table. 2 are neglected.

Evaluation metrics shown in Table. 4 are computed on the LRO image of Apollo 17 landing site. The images are subjected to CLAHE before performing predictions. The annotated data for this testset is taken from [Polegubic, 2020] who annotated all the craters manually. The evaluation metrics presented in Table. 5 are calculated without CLAHE. The testset used is the same as used for calculating scores in Table. 4. The input images are however not subjected to CLAHE before predictions. The input data used in this table is the same as that of in Table. 4.

Relationship in the form of graphical representation between precision and recall of different binarization methods is shown in Fig. 18. Adaptive Mean, Adaptive Gausssian, Sauvola and Niblack are not shown because of their poor performance with respect to F1-score which are calculated in Table. 2. Precision and recall are computed using Eqs. 5 and 6 respectively.

Table 2: F1-score on applied binarization methods on lunar crater probability map (Apollo 17 landing site). First row shows the percent decrease (indicated with a negative sign) and percent increase of binarization threshold.

	-50 %	-40 %	-30 %	-20 %	-10 %	0%	10%	20%	30%	40%	50%
Otsu	51.9	53.3	53.0	51.7	52.8	51.9	50.7	50.1	47.9	46.7	45.9
Isodata	47.6	49.2	47.7	48.4	48.6	46.9	47.4	47.6	44.7	44.6	44.1
Yen	36.7	38.7	41.4	44.3	45.1	45.8	45.8	47.7	48.3	49.2	49.4
Li	36.7	38.7	44.2	44.4	45.2	45.8	45.8	48.2	48.4	49.5	49.3
Mean	32.5	35.1	36.0	36.7	38.8	40.1	42.6	44.4	44.6	46.0	45.8
Adaptive mean	2.2										
Adaptive Gaussian	9.3										
Sauvola	14.0										
Niblack	9.3										

Table 3: Patch wise mean F1-score on testset distribution from training split.

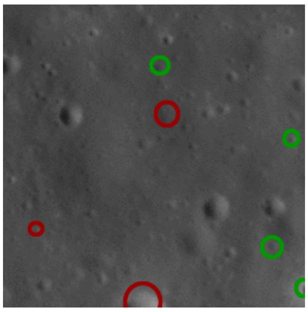
	Precision	Recall	F1-score
Otsu	77.5	70.6	73.8
Isodata	76.9	70.1	73.3
Li	56.9	80.5	66.7
Yen	51.8	72.9	60.5

Table 4: F1-score on Apollo 17 landing site LRO image, annotated by [Polegubic, 2020]. The results are obtained after using CLAHE on input images.

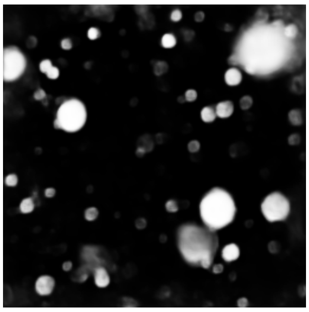
	Precision	Recall	F1-score
Otsu	77.0	55.7	64.6
Isodata	73.9	53.8	62.3
Li	60.7	59.2	59.9
Yen	51.9	53.1	52.5

Table 5: F1-score on Apollo 17 landing site LRO image, annotated by [Polegubic, 2020]. The results are obtained without application of CLAHE to input images.

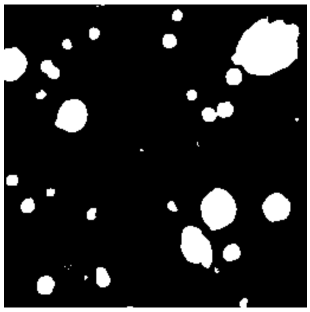
	Precision	Recall	F1-score
Otsu	50.8	42.4	46.2
Isodata	46.9	41.0	43.7
Li	35.1	44.0	39.0
Yen	25.2	39.1	30.6



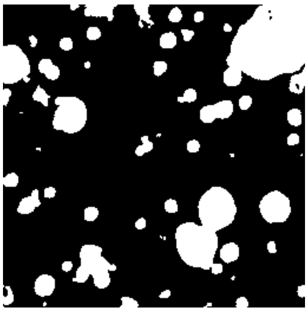
(a) Lunar optical image



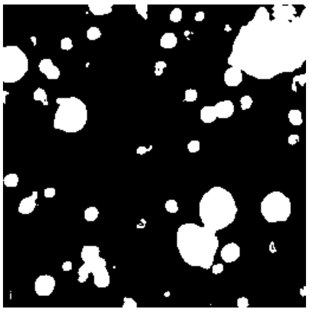
(b) Prediction of (a)



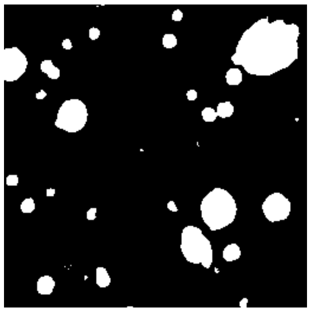
(c) Otsu



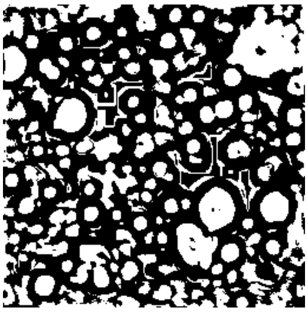
(d) Mean



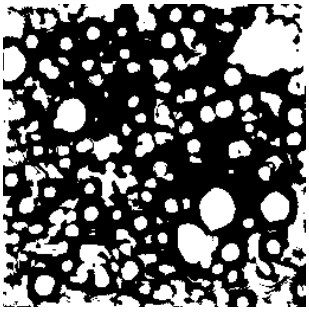
(e) Yen



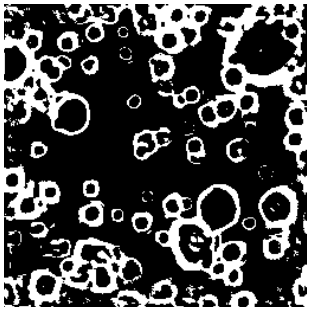
(f) Isodata



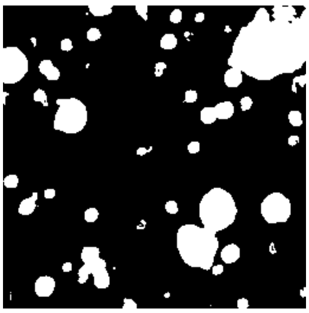
(g) Niblack



(h) Sauvola



(i) Adaptive Mean

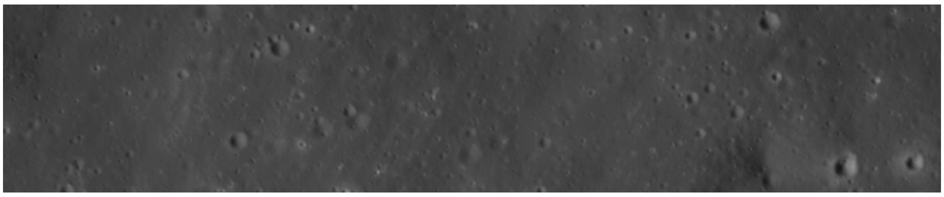


(j) Li

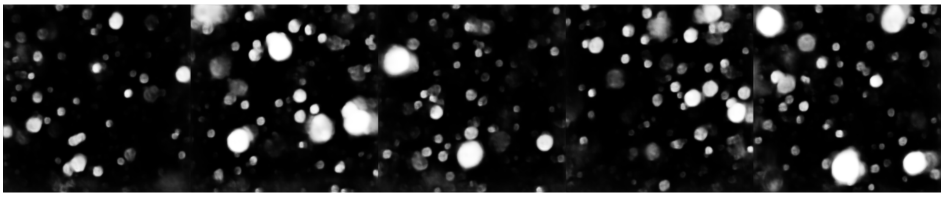
Figure 14: Image (a) is a 256×256 tile of lunar optical image taken by LRO of the Apollo 17 landing site. Craters marked in red are not detected by the u-net and craters marked in green are detected but removed in Otsu and Isodata thresholding. (b) shows the prediction (probability map) of (a) by the u-net model. Other figures show the binarization results of thresholding methods.

HCT was also applied for crater detection but the performance is very poor. Fig. 15 (c) shows extracted craters on prediction image tiles that belongs to Apollo 17 landing site. The detected craters are marked in green circles with a red dot in the center. HCT does not show a great performance in circle detection unless they have a significant round shape. This phenomena was also observed by [Wetzler et al., 2005].

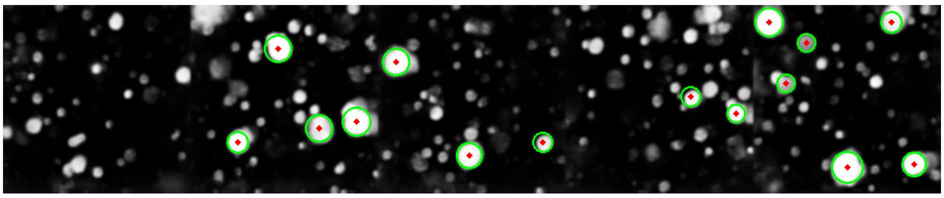
Fig. 16 (a) shows segment of the Apollo 17 landing site image which contain a large crater on left of the image and represented annotations are performed by [Polegubic, 2020]. For predictions the entire image was cropped into 256×256 pixel image tiles and Fig. 16 (b) shows the prediction on resultant tiles which are stitched together. This shows that there are border artifacts due to stitching. It is to be noted that large crater is divided into two images tiles, such that one image contains nearly 80% of this crater and other has crater rim with a low contrast value. Fig. 16 (c) shows the binarization of prediction result Fig. 16 (b) using Otsu's method.



(a)



(b)



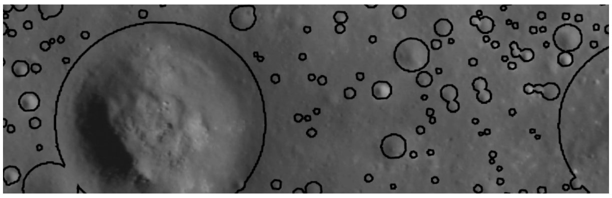
(c)

Figure 15: Image (a) shows five stitched tiles (256×256) of LRO optical images belonging to Apollo 17 landing site. (b) is the prediction of (a) using u-net and tiles are stitched together in the same order as Fig. (a). (c) shows extracted craters using HCT marked in green with a red dot in the middle.

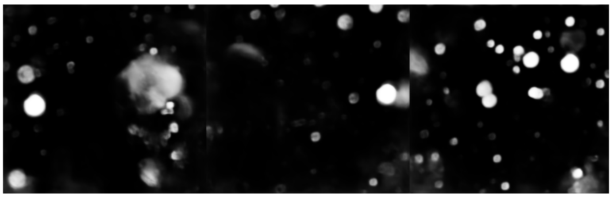
Mask R-CNN which is a deep neural network based architecture was also applied for lunar crater detection. The evaluation is quite poor as only few craters are detected on the Apollo 17 landing site image. There are multiple recognized craters by Mask-RCNN as shown in Fig. 17. Each colored rectangle show a detected crater. Multiple rectangles show multiple instances of craters which are either inside another crater on overlapping with other craters.

The detected craters by [Polegubic, 2020] and the comparison with u-net are plotted in Fig.

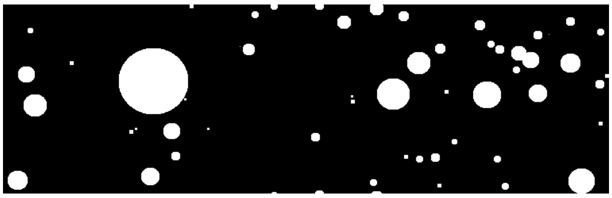
19. The plot is obtained after filtering all the craters of 0.51 m diameter as outliers. The cumulative sum of craters against their diameters are then plotted in a log-log scale. Left Fig. 19 is computed by [Polegubic, 2020] after manually annotating craters and there are two lines passing through the data. First line on the left shows that crater in this range shows the surface age of 0.1 Ga. Second line shows the age of 1 Ga. This shows that surface where the craters are counted is between 0.1 to 1 Ga old. Fig. 19 (right) shows the comparison plot where orange dots represents data points of [Polegubic, 2020] and blue plot shows the outcome of algorithm with Otsu's thresholding method. Both of the plots are are in the same age range. However, the blue plot shows that detected craters have ~ 3 pixel (1.5 m) diameter larger than the manually annotated dataset by [Polegubic, 2020] on Apollo 17 landing site image. This shows the same approximate age as determined by [Polegubic, 2020].



(a)



(b)



(c)

Figure 16: Fig. (a) shows a tile of LRO optical image annotated by [Polegubic, 2020]. This image belong to Apollo 17 landing site. Fig. (b) shows the prediction tiles of Fig. (a) stitched together (each tile of 256×256 pixels). It shows that a large crater on the left is not detected. Fig. (c) shows binarization of Fig. (a) by application of Otsu thresholding then using region proposal algorithm to extract crater diameters with local coordinates.

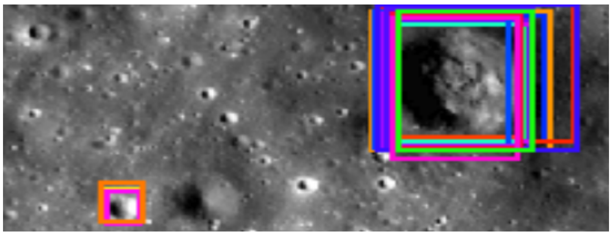
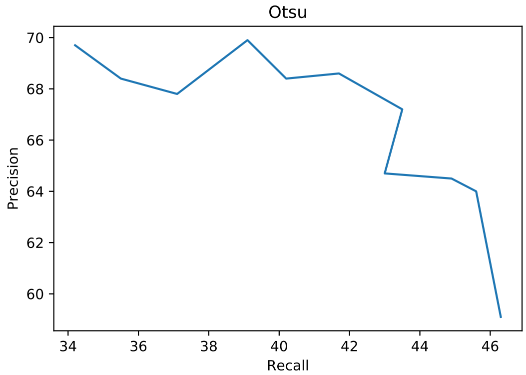
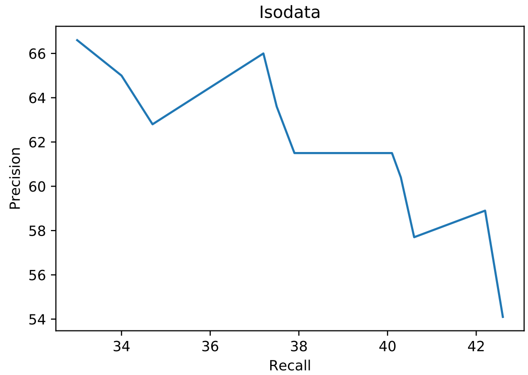


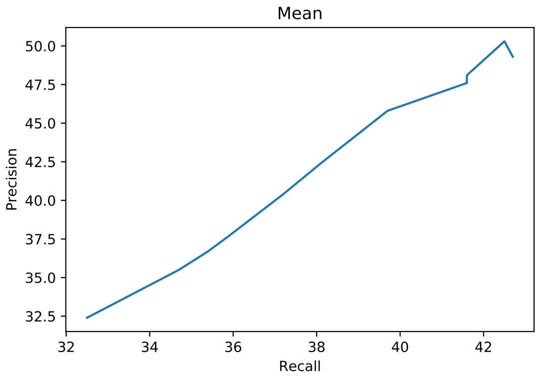
Figure 17: Prediction of Mask-RCNN. Instance segmentation on a tile of LRO optical image belonging to Apollo 17 landing site. Each rectangle show a detected crater.



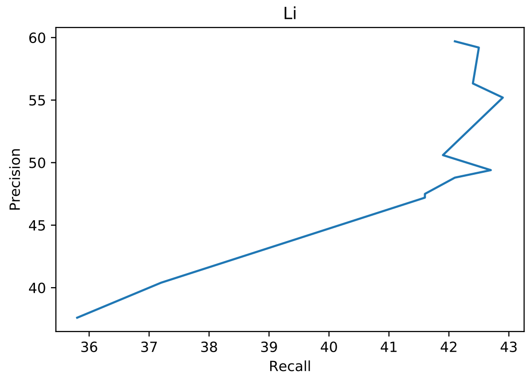
(a)



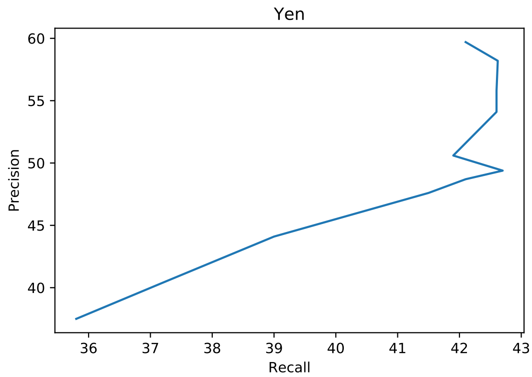
(b)



(c)



(d)



(e)

Figure 18: Relationship between precision and recall on different binarization methods. These values lead to the F1-score represented in Table. 2.

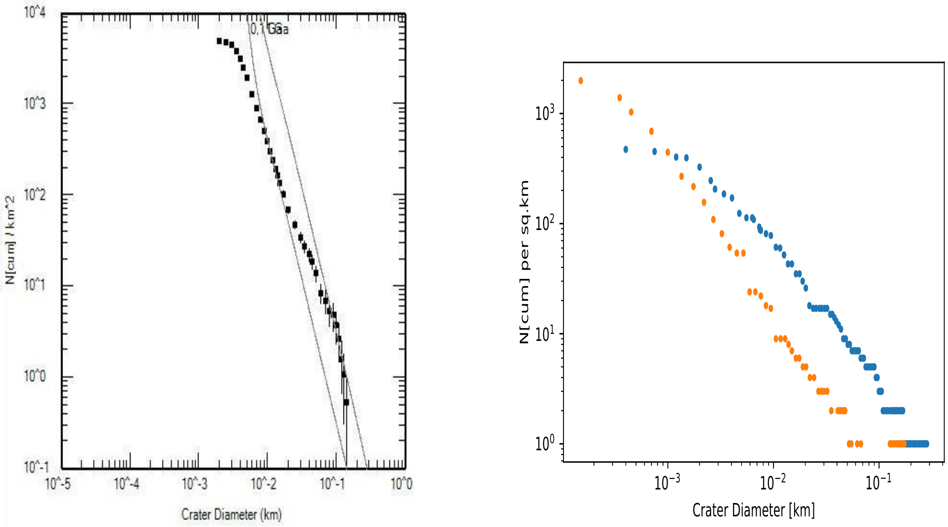


Figure 19: Plot on the left shows CSFD computed by [Polegubic, 2020] manually on LRO image of Apollo 17 landing site. Plot on the right shows comparison of CSFD of predictions belonging to the same image by [Polegubic, 2020] and detection generated by the u-net. Craters are extracted using Otsu's filter.

6 Discussion

The method of semantic segmentation based upon u-net is applied to generate probability distribution of craters on the lunar optical images taken by the LRO on two testsets which are different based upon solar angle and illumination. It is not possible to detect craters directly from the model output which are probability maps, therefore images are binarized using several binarization methods. The used binarization methods can be viewed in Fig. 14. Fig. 14 (a) represents a tile of the lunar optical image of Apollo 17 landing site. This dataset is annotated by [Polegubic, 2020] and is used as a testset for the evaluation of u-net model. The probability map of this image as shown in Fig. 14 (b) is acquired after application of CLAHE. Bright blobs in the prediction image shows the likelihood of a crater. By observing Fig. 14 (a), it is obvious that partially faded out craters have less likelihood in the prediction map. In Otsu thresholding the value is calculated by taking the two peaks of an image then taking a middle value. That is, all the dark and white pixels are grouped in two histograms and the approximate middle value is taken as the thresholding value. That is why some of the craters with little likelihood or low brightness are omitted out with this threshold. Other methods except Isodata mainly result into over segmentation because of lower threshold values with reference to Otsu. In the case of Niblack and Sauvola's method the thresholding value is 17 and 27 respectively out of the total range which is 0 to 255 pixel values. Lowering thresholding creates over segmentation problem as many blobs end up combined together and a slight noise in the prediction map is also considered as a crater as seen in Fig. 14 (g) and (h). Background terrain features contribute to noise in the prediction map which falls in the pixel values close to threshold of Niblack and Sauvola's method. That is why these methods count background features as craters which results into low precision and therefore low F1-score. Mean, Yen and Li's method have threshold values in the range of 45 to 50 pixels with mean's method having value of 45. Because of that, mean has more blobs than the other two methods. As the name suggests, mean thresholding is simply the mean of an entire probability map. As there is more background class than foreground class, hence there are more dark pixels which result into low value of 45.

The F1-score shown in Table. 2 are computed on the entire image of Apollo 17 landing site which is stitched from smaller tiles of 256×256 pixels. It can be observed that the F1-score increases by increasing threshold for methods like Yen, Li and mean but for Otsu and isodata it decreases slowly by further increasing threshold. Almost all of the thresholding methods result into slightly over or under segmentation in the case of crater detection. As Otsu and isodata have a high threshold value which is 97 and 89 respectively. Thus by decreasing this value, it is likely that F1-score would increase. Rest of the methods results into over segmentation that specially penalizes recall thus resulting lower F1-score in comparison with Otsu and isodata. It has been also observed that taking a large image and cropping it into 256×256 tiles then performing predictions and stitching predicted images back to form the actual image, results into low F1-score in comparison to F1 measured on every single image separately. This is due to the fact that many of the detected craters are split into half in two images and some are located in such a way that one image has crater shadow and other has illumination. This results into detection of crater on one image but not on the other, which means there is one true positive and one false negative so recall of the two images is 50% instead of 100%. That is why scores in Table. 2 are lower than that in Table. 4 on same dataset (both with CLAHE). This phenomena can be observed in Fig. 16 where a large crater is divided into different tiles with one having a dark edge and other with the illumination part. However, the entire crater is not detected but the smaller ones inside of it by the u-net model which is most likely because of less illumination even after the application of CLAHE.

F1-score in Table. 3 is the highest because of testset which is a training split. U-net is trained on the similar images with respect to solar angle and therefore it is expected that performance on training set is higher than on the Apollo 17 landing site testset. These scores are computed on each image separately and then an average value is taken of all the images. Both precision and recall are computed image wise as well. On the Apollo 17 landing site image the F1-score shown in Table. 4 are lower as compared to Table. 3. However, in both tables precision has not changed significantly but there is a notable decrement in recall score of methods and after analyzing images, it is concluded that this happened because of different solar angle and illumination on testset of Table. 4. Another reason for lower scores is that none of the images with similar solar angles and illumination are used in training of the algorithm. A deep learning model does not perform well like training split testset, on the images which are not used in training of the model. Otsu once again outperforms other methods including isodata by a slight margin.

Evaluation metrics shown in Table. 5 are lower than Table. 3 and 4 because of zero contrast enhancement. The input images used to compute scores are same for all these tables but without enhancing local contrast the difference between illumination and shade is little. This makes model to learn slow and with limited amount of training data, the performance is poor on images without CLAHE.

It is clear that using methods of high threshold i.e. Otsu and isodata reduces recall but increases precision. Both of these methods yield less false positives in comparison to other methods. High threshold has a drawback of missing small craters which are usually less than 6 pixels (3 m) in diameter and also those which are highly degraded and barely have a shadow. The difference between recall and precision is obvious only if the segmenter is strongly over or under splitting [Badrinarayanan et al., 2017]. In any of the case, only considering precision can be misleading as evaluation metric because it favors under segmentation. Contrary to precision, recall does not favor under or over segmentation, therefore F1-score is taken into account as the evaluation metric for segmentation accuracy.

From Fig. 18, it can be noted that Otsu and isodata provides a good trade off between precision and recall thus results into higher F1-score as compared to other methods. This makes both of these methods effective for generating segmentation mask as compared to the other used methods. For mean method the thresholding is low and by increasing this value, both precision and recall increases. This trend can also be observed in Li and Yen's method shown in Figs. 18 (d) and (e) respectively. However, recall is becoming constant at 42.5 in both methods. By further increase in precision, the recall is decreasing. This phenomena is because of under segmentation which means there are more false negatives (missed craters), hence decreasing recall. But for precision false positives decreases which increases the value of precision according to Eq. 5.

In all the methods other than Otsu and isodata, both precision and recall is low in the beginning of curves. This is because of over segmentation which results into low IoU. That in turn leads to less true positives. False negatives are computed by subtracting true positives from groundtruth craters. hence, more the false negatives and lower the recall. For precision, the value is obviously low as there are many false positives. The background noise is also considered as crater in over segmentation condition.

It is also observed that cropping a large image and performing predictions on smaller cropped images will result into artifacts when segmented maps are stitched together. These can be visualized in Figs. 15 (b) and 16 (b). Binarization of stitched large image results into addition of false positives because of these artifacts. This penalizes F1-score and has been observed in Table. 2, therefore it makes more sense to perform image wise binarization and calculate F1-score on single image basis then average the F1-score of all the images to get final evaluation.

Mask-RCNN was also used in experimentation for instance segmentation of lunar craters. Fig. 17 shows the outcome on LRO image of Apollo 17 landing site. It is observed that performance of this architecture was poor despite of huge success on several types of object classification and instance segmentation problems. Through experimentation, it is found that this happened because of little training data. Mask-RCNN is a deep neural network with many layers for features extraction which is trained on thousands of images for generating prediction with proven accuracy record. Furthermore, lunar craters does not resemble with classes the algorithm is trained on i.e. COCO or ImageNet dataset. That means Mask-RCNN has to be trained on a large dataset of lunar optical images. This requires intense amount of labeling craters on thousands of lunar images. Moreover, the complexity of craters in optical images with shades on one side makes it difficult for such a deep architecture to learn from a dataset composed of less than 300 training images.

The estimation plot as shown in Fig. 19 (right) shows that resulting detected craters from the u-net fall in the same age range as estimated by [Polegubic, 2020]. The difference of roughly around 1.5 m is largely because of the thresholding method which considers overlapping craters as one. This difference can be seen as the blue dotted line is not exactly on the orange line but a little on the right towards greater crater diameter.

7 Limitations and Future Work

A deep learning model based on the u-net with a small dataset has been proven effective as compared to instance segmentation model Mask-RCNN. Obviously u-net only performs semantic segmentation which means all the craters inside a crater or overlapping craters are counted as one. The performance of crater detection deep learning algorithm can be potentially improved in the following ways.

1. Increasing the amount of training data. This is likely to increase the performance of the u-net model for semantic segmentation task.
2. Usage of CLAHE in data augmentation will remove this step from the post-processing which will shorten the post-processing step. Moreover, it is possible that performance of model might also increase due to variation in data.
3. Usage of optical images for training with several different solar angles. Trained model with such several variations in the shadow length and sun illumination will increase the prediction ability of model on any satellite lunar optical images.
4. The precision, recall and F1-score can be improved by changing the model structure to be trained on larger images. Hence, a large image can be predicted at once rather than prediction on smaller image tiles and stitching them together.
5. For instance segmentation a larger dataset (i.e. > 300 images) can be tested on Mask-RCNN and most likely with increase in data the performance of Mask-RCNN will also improve. With a small dataset few crater instances were already detected as shown in Fig. 17. Moreover, it has already shown potential by showing on DEMs as discussed in the 'Related Work' section.

8 Conclusion

Crater automation can be challenging because of variability in appearance of craters and surrounding terrain. Traditional crater detection methods composed of CDAs, GHT and CSTM along with machine learning methods such as SVM have been in practice largely till recent years. Deep learning methods have got an attention of planetary science community following the success in object detection competitions. These techniques have outperformed other traditional crater detection methods. Largely the experimentation regarding these techniques is carried out on the DEMs as crater features are not as complex as in optical images which have a shadow on one side and sun illumination on the other.

For a deep learning algorithm it is necessary to have annotated dataset for training. In case of lunar craters there is not open source availability of such dataset which could be used for training. Therefore, a relatively small dataset is annotated and used for training of a u-net based model. This architecture was selected because of its proven success with little dataset in medical science. Also Mask-RCNN was applied for instance segmentation with the same dataset. In post processing HT was tested along with several binarization methods in terms of precision, recall and F1-score.

U-net model was trained on a high resolution optical image tiles taken by LRO. For evaluation it was applied on training split testset and another dataset annotated manually by [Polegubic, 2020] with a different solar and sun illumination angle. This image belong to Apollo 17 landing site. On the prediction images different binarization methods were applied to extract craters. HT had the worst performance on prediction images. The evaluation on binarized images was performed on images tiles which are stitched together; patch wise evaluation on images and averaging the results; applying CLAHE and without application of CLAHE. It is observed that by applying CLAHE the F1-score increases significantly. All the methods except Otsu and isodata had poor performance which was due to over segmentation. Mask-RCNN was also applied the performance was poor which is because less amount of dataset and it turns out that transfer learning does not help if the data is too different. In result, experimentation with u-net and combination of Otsu thresholding gave the best F1-score of 73% on training split and 64% on Apollo 17 landing site image. The age estimation plot also shows that by using this deep learning approach the age estimation is similar to the age predicted on manually annotated dataset. The comparison of both data points using Otsu's filter fall very close to the data points of manual annotations. However, there is a slight difference in the diameters largely because of the overlapping craters.

References

[Ali-Dib et al., 2019] Ali-Dib, M., Menou, K., Zhu, C., Hammond, N., and Jackson, A. P. (2019). Automated crater shape retrieval using weakly-supervised deep learning. *arXiv preprint arXiv:1906.08826*.

[Badrinarayanan et al., 2017] Badrinarayanan, V., Kendall, A., and Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495.

[Bandeira et al., 2010] Bandeira, L., Ding, W., and Stepinski, T. (2010). Automatic detection of sub-km craters using shape and texture information. In *Lunar and Planetary Science Conference*, volume 41, page 1144.

[Brunelli and Poggio, 1993] Brunelli, R. and Poggio, T. (1993). Face recognition: Features versus templates. *IEEE transactions on pattern analysis and machine intelligence*, 15(10):1042–1052.

[Burger et al., 2009] Burger, W., Burge, M. J., Burge, M. J., and Burge, M. J. (2009). *Principles of digital image processing*, volume 54. Springer.

[Burl et al., 2001] Burl, M. C., Stough, T., Colwell, W., Bierhaus, E., Merline, W., and Chapman, C. (2001). Automated detection of craters and other geological features.

[Cohen et al., 2016] Cohen, J. P., Lo, H. Z., Lu, T., and Ding, W. (2016). Crater detection via convolutional neural networks. *arXiv preprint arXiv:1601.00978*.

[Dalal and Triggs, 2005] Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection.

[DeLatte et al., 2019] DeLatte, D. M., Crites, S. T., Guttenberg, N., Tasker, E. J., and Yairi, T. (2019). Segmentation convolutional neural networks for automatic crater detection on mars. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(8):2944–2957.

[Ding et al., 2011] Ding, W., Stepinski, T. F., Mu, Y., Bandeira, L., Ricardo, R., Wu, Y., Lu, Z., Cao, T., and Wu, X. (2011). Subkilometer crater discovery with boosting and transfer learning. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(4):39.

[Duda and Hart, 1972] Duda, R. O. and Hart, P. E. (1972). Use of the hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15.

[Dvornik et al., 2019] Dvornik, N., Mairal, J., and Schmid, C. (2019). On the importance of visual context for data augmentation in scene understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[Dwibedi et al., 2017] Dwibedi, D., Misra, I., and Hebert, M. (2017). Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1301–1310.

[Emami et al., 2015] Emami, E., Bebis, G., Nefian, A., and Fong, T. (2015). Automatic crater detection using convex grouping and convolutional neural networks. In *International Symposium on Visual Computing*, pages 213–224. Springer.

[Felzenszwalb et al., 2008] Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *2008 IEEE conference on computer vision and pattern recognition*, pages 1–8. IEEE.

[Finkelstein et al., 2019] Finkelstein, M., Baby, S. M., and Kitano, H. (2019). Automatic lunar crater detection from optical images and elevation maps.

[Girshick, 2015] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

[Girshick et al., 2014] Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation.

[Hartmann, 1981] Hartmann, W. K. (1981). Chronology of planetary volcanism by comparative studies of planetary cratering. *Basaltic Volcanism on the Terrestrial Planets.*, pages 1049–1127.

[He et al., 2017] He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

[Hiesinger et al., 2000] Hiesinger, H., Jaumann, R., Neukum, G., and Head III, J. W. (2000). Ages of mare basalts on the lunar nearside. *Journal of Geophysical Research: Planets*, 105(E12):29239–29275.

[Honda and Azuma, 2000] Honda, R. and Azuma, R. (2000). Crater extraction and classification system for lunar images. *Mem. Fac. Sci. Kochi Univ.(Inform. Sci.)*, 21:13–22.

[Ivanov, 2002] Ivanov, B. (2002). The comparison of size-frequency distributions of impact craters and asteroids and the planetary cratering rate. *Asteroids III*, 1:89–101.

[Ivanov et al., 2015] Ivanov, T. I., Huertas, A., and Johnson, A. E. (2015). Probabilistic surface characterization for safe landing hazard detection and avoidance (hda). US Patent 9,141,113.

[Koeberl, 1994] Koeberl, C. (1994). African meteorite impact craters: Characteristics and geological importance. *Journal of African Earth Sciences*, 18(4):263–295.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105.

[Leroy et al., 2001] Leroy, B., Medioni, G., Johnson, E., and Matthies, L. (2001). Crater detection for autonomous landing on asteroids. *Image and Vision Computing*, 19(11):787–792.

[Li and Lee, 1993] Li, C. H. and Lee, C. (1993). Minimum cross entropy thresholding. *Pattern recognition*, 26(4):617–625.

[Martins et al., 2009] Martins, R., Pina, P., Marques, J. S., and Silveira, M. (2009). Crater detection by a boosting approach. *IEEE Geoscience and Remote Sensing Letters*, 6(1):127–131.

[Melosh, 1988] Melosh, H. (1988). The rocky road to panspermia. *Nature*, 332(6166):687.

[Meng et al., 2009] Meng, D., Yunfeng, C., and Qingxian, W. (2009). Method of passive image based crater autonomous detection. *Chinese Journal of Aeronautics*, 22(3):301–306.

[Neukum, 1983] Neukum, G. (1983). Meteoritenbombardement und datierung planetarer oberflächen, habilitation dissertation for faculty membership. *Ludwig-Maximilians-University, Munich*, page 186.

[Neukum and Ivanov, 1994] Neukum, G. and Ivanov, B. (1994). Crater size distributions and impact probabilities on earth from lunar, terrestrial-planet, and asteroid cratering data. *Hazards due to Comets and Asteroids*, 1:359–416.

[Neukum et al., 1975] Neukum, G., König, B., Fechtig, H., and Storzer, D. (1975). Cratering in the earth-moon system-consequences for age determination by crater counting. In *Lunar and Planetary Science Conference Proceedings*, volume 6, pages 2597–2620.

[Opik, 1965] Opik, E. (1965). Mariner iv and craters on mars.

[Palafox et al., 2017] Palafox, L. F., Hamilton, C. W., Scheidt, S. P., and Alvarez, A. M. (2017). Automated detection of geological landforms on mars using convolutional neural networks. *Computers & geosciences*, 101:48–56.

[Polegubic, 2020] Polegubic, D. (2020). Landing site characterization for the pts lunar mission to the apollo 17 area. *Master thesis at the Technical University of Berlin.*

[Redmon et al., 2016] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788.

[Reiss, 1993] Reiss, T. H. (1993). *Recognizing planar objects using invariant image features*, volume 17. Springer.

[Ren et al., 2015] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

[Ridler et al., 1978] Ridler, T., Calvard, S., et al. (1978). Picture thresholding using an iterative selection method. *IEEE transactions on Systems, Man and Cybernetics*, 8(8):630–632.

[Robinson et al., 2010] Robinson, M., Brylow, S., Tschimmel, M., Humm, D., Lawrence, S., Thomas, P., Denevi, B., Bowman-Cisneros, E., Zerr, J., Ravine, M., et al. (2010). Lunar reconnaissance orbiter camera (lroc) instrument overview. *Space science reviews*, 150(1-4):81–124.

[Ronneberger et al., 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer.

[Rumelhart et al., 1988] Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1.

[Salamuniccar and Loncaric, 2010] Salamuniccar, G. and Loncaric, S. (2010). Method for crater detection from martian digital topography data using gradient value/orientation, morphometry, vote analysis, slip tuning, and calibration. *IEEE transactions on Geoscience and Remote Sensing*, 48(5):2317–2329.

[Sawabe et al., 2006] Sawabe, Y., Matsunaga, T., and Rokugawa, S. (2006). Automated detection and classification of lunar craters using multiple approaches. 37(1):21–27.

[Silburt et al., 2019] Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., Tamayo, D., and Menou, K. (2019). Lunar crater identification via deep learning. *Icarus*, 317:27–38.

[Snuverink, 2017] Snuverink, I. (2017). Deep learning for pixelwise classification of hyperspectral images. *Master of Science Thesis*.

[Soderblom, 1970] Soderblom, L. A. (1970). *The distribution and ages of regional lithologies in the lunar maria*. PhD thesis, California Institute of Technology.

[Stepinski et al., 2009] Stepinski, T. F., Mendenhall, M. P., and Bue, B. D. (2009). Machine cataloging of impact craters on mars. *icarus*, 203(1):77–87.

[Stöffler and Ryder, 2001] Stöffler, D. and Ryder, G. (2001). Stratigraphy and isotope ages of lunar geologic units: Chronological standard for the inner solar system. In *Chronology and evolution of Mars*, pages 9–54. Springer.

[Thoma, 2016] Thoma, M. (2016). A survey of semantic segmentation. *arXiv preprint arXiv:1602.06541*.

[Uijlings et al., 2013] Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. (2013). Selective search for object recognition. *International journal of computer vision*, 104(2):154–171.

[Urbach and Stepinski, 2009] Urbach, E. R. and Stepinski, T. F. (2009). Automatic detection of sub-km craters in high resolution planetary images. *Planetary and Space Science*, 57(7):880–887.

[Vinogradova et al., 2002] Vinogradova, T., Burl, M., and Mjolsness, E. (2002). Training of a crater detection algorithm for mars crater imagery. In *Proceedings, IEEE Aerospace Conference*, volume 7, pages 7–7. IEEE.

[Viola et al., 2001] Viola, P., Jones, M., et al. (2001). Rapid object detection using a boosted cascade of simple features. *CVPR (1)*, 1(511-518):3.

[Wang et al., 2018] Wang, H., Jiang, J., and Zhang, G. (2018). Crateridnet: an end-to-end fully convolutional neural network for crater detection and identification in remotely sensed planetary images. *Remote Sensing*, 10(7):1067.

[Wetzler et al., 2005] Wetzler, P. G., Honda, R., Enke, B., Merline, W. J., Chapman, C. R., and Burl, M. C. (2005). Learning to detect small impact craters. In *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)-Volume 1*, volume 1, pages 178–184. IEEE.

[Williams et al., 2018] Williams, J.-P., van der Bogert, C. H., Pathare, A. V., Michael, G. G., Kirchoff, M. R., and Hiesinger, H. (2018). Dating very young planetary surfaces from crater statistics: A review of issues and challenges. *Meteoritics & Planetary Science*, 53(4):554–582.

[Yen et al., 1995] Yen, J.-C., Chang, F.-J., and Chang, S. (1995). A new criterion for automatic multilevel thresholding. *IEEE Transactions on Image Processing*, 4(3):370–378.