Indian Institute of Technology Gandhinagar

Framework Development for Automated Crater Detection from Lunar Images

by

Preet Khaturia

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Master of Technology
in Electrical Engineering,
Indian Institute of Technology Gandhinagar
July 2018

Advisory Committee:

Prof. Nitin Khanna (Supervisor)
Prof. Nithin V. George
Prof. Vikrant Jain

CERTIFICATE

It is certified that the work contained in the thesis titled **Framework Development for Automated Crater Detection from Lunar Images** by **PREET KHATURIA** (**16210065**), has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Prof. Nitin Khanna
Assistant Professor
Indian Institute of Technology Gandhinagar
Gandhinagar - 382355, India

DEDICATION

Dedicated to my parents

ACKNOWLEDGMENTS

To begin with, I would like to express my sincere gratitude to my advisor, Professor Nitin Khanna, for his guidance and support during my research beginning from reading a research paper to debugging a piece of code and developing my interest towards image processing.

I would like to thank Nikita Agarwal, a Junior Research Fellow at IIT Gandhinagar and my lab mates for their honest feedback and moral support which made me work in a better way.

I would like to thank my dissertation committee members: Prof. Nithin V. George and Prof. Vikrant Jain for taking out their time to serve on my committee and providing with their valuable suggestions.

Last but not the least, I would like to express deep gratitude to my family members, with their never-ending support and patience, this journey has been a “wonderful” one.

This research was partially supported by a grant from Indian Space Research Organisation (ISRO), Department of Space, Government of India, under the project titled ‘Study of Lunar Geomorphological and Impact Cratering Processes through Chandrayaan-1 Data Sets’ and Award Number ISRO/SSPO/Ch-1/2016-17. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the funding agencies.

TABLE OF CONTENTS

Chapter

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CDA	Crater Detection Algorithm
CG	Camera Generated
CNN	Convolutional Neural Network
DEM	Digital Elevation Model
DCT	Discrete Cosine Transform
DTM	Digital Terrain Model
DWT	Discrete Wavelet Transform
FLD	Fisher Linear Discriminant
FSD	First Significant Digit
GPU	Graphics Processing Unit
HVS	Human Visual System
HiRISE	High Resolution Imaging Science Experiment
IOU	Intersection Over Union
JPEG	Joint Photography Expert Group
LRO	Lunar Reconnaissance Orbiter
LDIM	Lunar Digital Image Model
LOLA	Lunar Orbiter Laser Altimeter
MDCT	Modified Discrete Cosine Transform
NMS	Non-Maximal suppression
PCM	Pulse Code Modulation
ReLU	Rectified Linear Unit
SOM	Self-Organizing Map
RANSAC	Random sample consensus

SVM	Support Vector Machines
VRC	Volcanic Rootless Cones
UCID	Uncompressed Color Image Database

ABSTRACT

Impact craters are the most prominent and studied features in planetary remote sensing studies. These circular features are important to determine the age and geology of the planetary surface. Traditionally, crater counting has been done by visual inspection, limiting the scope and accuracy of retrieval. With the advent of lunar and planetary exploration, a large amount of data is available from various satellites which can be used to study the morphological characteristics of the landscape. We propose an algorithm to automatically detect impact craters on the surface of the Moon by using object detection framework inspired from R-CNN network. Crater detection algorithm can be considered as an object detection technique where we have only one object to detect, i.e., crater. The algorithm consists of three steps; firstly, generation of region proposals from ortho images, DEM and slope images using a graph-based segmentation method to limit the search space for a candidate region. Secondly, features are extracted corresponding to each region proposal using 2-D Convolutional Neural Networks (CNN), following VGGNet. Finally, we propose a post-processing algorithm to eliminate overlapped boxes detected as objects and to optimize the boxes based on their probability score. The algorithm results a mAP@0.5 of 87.14% and mAP@$[0.5, 0.05, 0.9]$ of 48.41% trained on images from LRO (Lunar Reconnaissance Orbiter) dataset with a resolution of 118m/pixel for DEM and 100 m/pixel for ortho images.

Chapter 1

Introduction

Craters are the most abundant topographic elements present on the surface of planetary bodies. Its nomenclature is dependent upon its formations as Impact, Volcanic or Subsidence craters. On the surface of Moon, the most dominated types of craters are the Impact craters, which are bowl-shaped depressions caused by asteroids and meteorites colliding with the surface of planetary bodies (Figure 1.1).

Some large impacts like Copernicus or Tycho craters on lunar surface can even cause secondary craters or series of craters as ejecta falls back to the surface. The impact is also affected by the slope of the surrounding terrain, which results in complex crater structures. The transformation from simple to complex crater on any planetary body is dependent on acceleration due to gravity [3]. The study of complex crater formation gives scientist an insight into the geological and morphological processes varying over time [4]. These craters have varied shape, size, and structure which is thought to be the result of surface processes, its properties, and impact parameters like size, mass, speed, and angle of the falling object. Lunar surface has abundant of such craters ranging in diameter from a few meters to hundreds of kilometers.

On the surface of Earth, the process of erosion and gravity reduces or removes the impact of craters. The absence of atmosphere and water on lunar surface preserves these

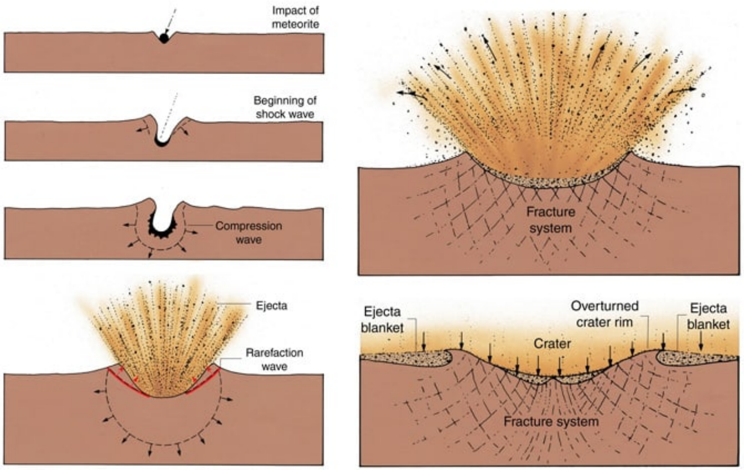


Figure 1.1: Formation of Impact Craters [2]

craters and their structure which is why they provide an excellent tool to understand the surface processes that may affect the structure of lunar craters. Craters are also classified depending upon their texture as simple and complex craters. Figure 1.2 shows the cross-sectional profile of a simple and complex crater on the Moon's surface. Simple craters show crescent bright and shadow structure with a smooth profile. Whereas, complex craters have multiple central peaks at the crater bed due to surrounding terrain or any seismic activity. The flatness of the crater floor is measured by the flatness in the center of the profile. The other parameter most frequently used to distinguish craters is the d/D or depth to diameter ratio. Depth to diameter ratio is inversely related to the size of the crater.

Lunar craters have been long studied to understand the effects of surface properties

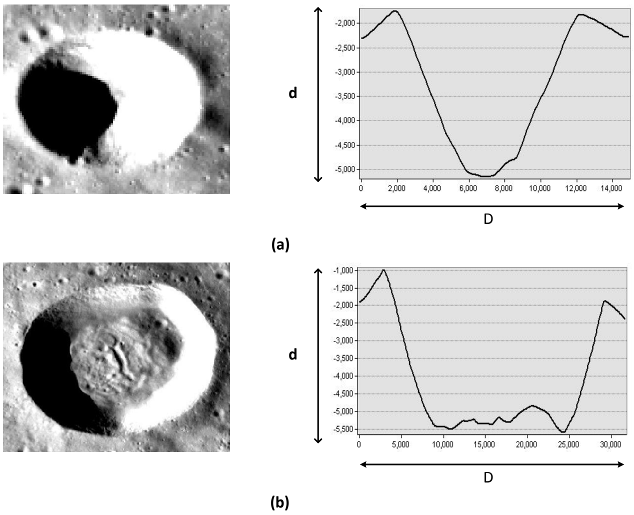


Figure 1.2: Cross sectional profile of (a) Simple and (b) Complex crater

and processes by studying their degradation process. Moreover, numerous planetary missions on Moon requires the knowledge of these depressions. The first step, however, in any such study is to locate the exact position of craters on the lunar surface with the help of lunar images. This marking of craters has been addressed using numerous approaches including manual marking, image processing techniques to detect circular shapes, use of DEM to identify and locate bowl-shaped depressions, etc. With new and high-resolution data coming in by various lunar missions such as SELENE and Chandrayaan-1, the need to develop a robust technique for crater detection has also increased and become challenging [5, 6, 7]. Moreover, mapping craters on lunar surface can also serve as a tool to estimate the landing sites, route lunar probes and rovers effectively without manual efforts.

Big data and advances in deep learning technology have improved the state of the art performance in many computer vision related tasks especially, object detection. The advancement in GPU technology has reduced the time required to converge the deeper and larger networks. Various CNN architectures have improved the classification task and brought the accuracy very close to human accuracy. With all these advances in the direction of deep learning, few researchers, in recent years, have attempted to solve the crater detection problem with the help of deep learning. This dissertation presents an automatic method for detecting impact craters on the surface of Moon using images from LRO database. Chapter 1 gives a brief review of the crater detection algorithms based on ortho images and DEM. An introduction to convolutional neural networks (CNN) has been given in Chapter 2. Chapter 3 and Chapter 4 gives the description of the proposed methodology and the experimental results. Have you ever wondered how do natural numbers occur in nature? Do they occur with same probability and if so, why their study is so important? A survey on Benford's law which explains this phenomenon and its applications to forensics techniques has been provided in Chapter 5. Conclusion and future work is given in Chapter 6.

1.1 A review of Crater Detection Algorithms

According to the Giant-impact hypothesis, Moon was formed from the remnant debris of a collision between the early Earth and a Mars-sized body, Theia. The models of the hypothesis are; a) A standard model: Earth collided with a Mars-sized body, b) Fast-Spinning Earth Model: A fast-spinning Earth impacted with a small body, c) Collision of two half-earth sized planets. These facts suggested that the composition of materials on the surface of Moon and Earth are relatively similar. To further strengthen the belief and to know more about the lunar surface and its origin, the first mission to the Moon was send by Soviet Union's Luna programme known as Luna 2 on September 13, 1959. However, the first space probe to reach the vicinity of the lunar surface was Luna 1, in 1959. The images from the missions show that the surface of the Moon is abundantly covered with crater structures showing its prominence on the Moon's surface. Since then, they have been studied giving scientists insight about the relative age and geology of the surface.

In the early twentieth century, many of the crater detection tasks were carried out manually [8, 6, 9] by marking circles on images. Even the most comprehensive catalog survey [10, 11, 12, 13] on planetary bodies was prepared with manual intervention. Taking into account that lunar surface harbors numerous craters with varied sizes, shapes, and structures, manual detection of all these craters is infeasible, laborious and prone to error task. Also, crater counting can yield disagreements up to 40% [14, 15]. However, due to the recent advances in imaging and satellite technology, ample amount of high-resolution data has become available. Many studies have focused on designing algorithms for automating crater detection from these datasets. A framework for 112 publications for crater detection algorithms using DEM/ optical images can be found in [16]. This section presents a brief review of the CDA algorithms based on data source point of view, ortho images and DEM. The review on ortho images is further divided according to the techniques as supervised and unsupervised learning techniques.

1.1.1 Crater Detection from Ortho Images

Unsupervised learning methods rely on inferring the structure of data from the labeled data. Optical images from satellites reflect craters as circular features, differentiated from the background by crater rim and ejecta material. Hence, many crater detection tasks seek motivation from various image processing algorithms which can detect circular features. Hough transform [17], based on a voting technique to detect shapes, was used to integrate these features resulting from local variations. It is the transformation from image space to the parameter space or the Hough space. The dimensions depend on the search parameters, three (position of x, position of y, center point) in the case of fitting a circle. Honda et al. [18] grouped the images from Lunar Digital Image Model (LDIM) according to spatial frequency patterns and detected circular features using a combination of Hough transform and genetic algorithms. Removal of false candidates was done by a self-organizing map (SOM) approach.

Similarly, a combinational approach using various image processing techniques were proposed in [19, 20, 21]. Using Multispectral images from Clementine space mission, the authors in [22] were able to improve their accuracy to 80% from [20]. The flowchart of the proposed algorithm is shown in Figure 1.3.

The prominence of the crater is not uniform on the lunar mare and highland region after edge detection techniques due to illumination conditions. Authors in [23] adopted an extensive preprocessing to enhance edges on Chang'E-1 CCD images followed by detection using Hough transform. Kang et al. [24] proposed the best fit model technique using RANSAC, another circle detection algorithm, on high pass filtered images using a bilateral filter. Additionally, crater classification such as simple and complex is also proposed using various 3-D features calculated from DEM images. Authors in [25] used template matching followed by evaluating probability volume to detect candidate craters. Further, image segmentation techniques for delineating craters from texture measures such as mean, variance, entropy, etc. was proposed in [26]. An object-oriented technique to classify image pixel into similar objects on lunar reflectance images was used in [27].

Illumination conditions on lunar surface show craters as a pair of crescent shaped bright and dark regions. Sawabe et al. [20] used this observation to detect bright and shadow part of the crater at different sun angles and further improved these techniques using multi spectral images [22]. Authors in [28, 29] used a boosting approach previously proposed by viola and jones (2004) [30] to detect craters from Mars Orbiter

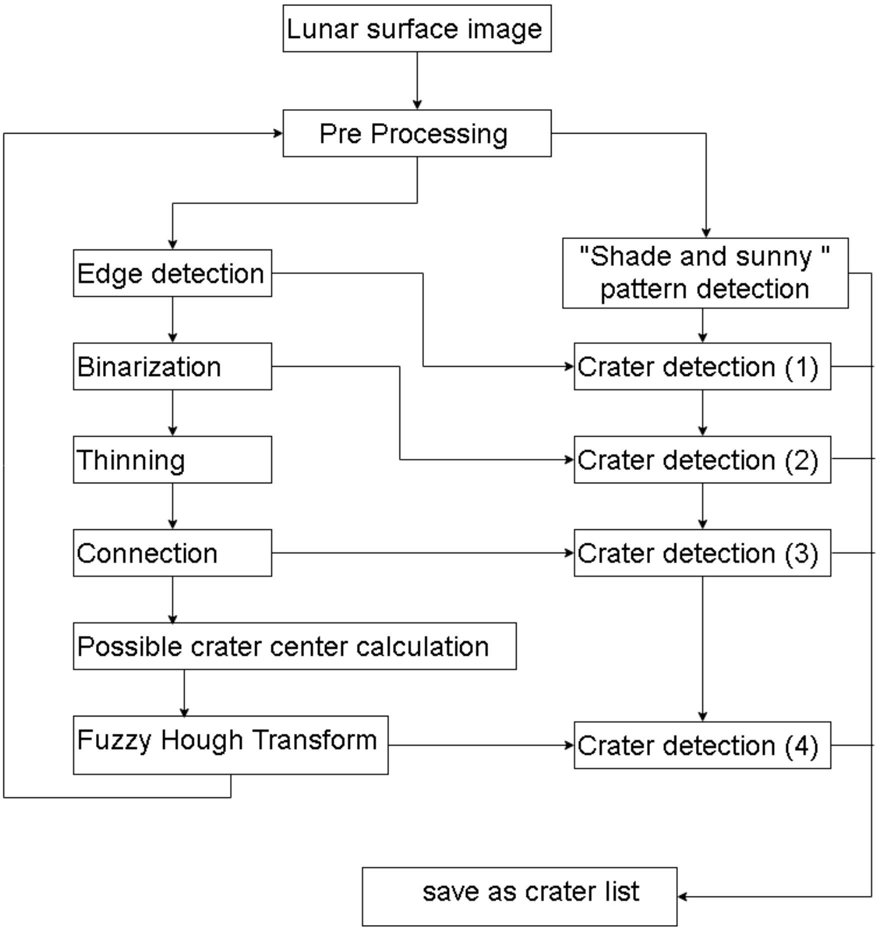


Figure 1.3: Flow diagram of the proposed algorithm [22]

Camera. Most of the crater detection algorithms detected large craters which were even identifiable through low resolution images. Algorithms on sub-km crater detection was first proposed by [31] using mathematical morphological techniques of thresholding image into regions of bright and dark areas, processing these regions using shape based features and classification using Support Vector Machines (Figure 1.4). Using a large set of texture features (1089) in [32] with boosting ensemble learning algorithm and transfer learning, better accuracy was achieved as compared to the previous one.

A similar approach was proposed in [33]. Further in [34], using texture features in addition to shape-based features the authors were able to increase the efficiency of the algorithm to 81%. The method was to use Adaboosting by replacing the single threshold classifier by a dual threshold weak classifier based on the characteristics of the distribution of features. The detection rate of 85% was achieved on High-Resolution Stereo Camera (HRSC) images with the image resolution of 12.5 m/pixel. Gist features were calculated on candidate craters from ortho images by edge extraction techniques and further random forest algorithm was used as a classifier [35] . A similar methodology, separating bright and dark regions, tested on SELENE images, was followed by [36] and classified the craters as round or flat craters by calculating the crater profile

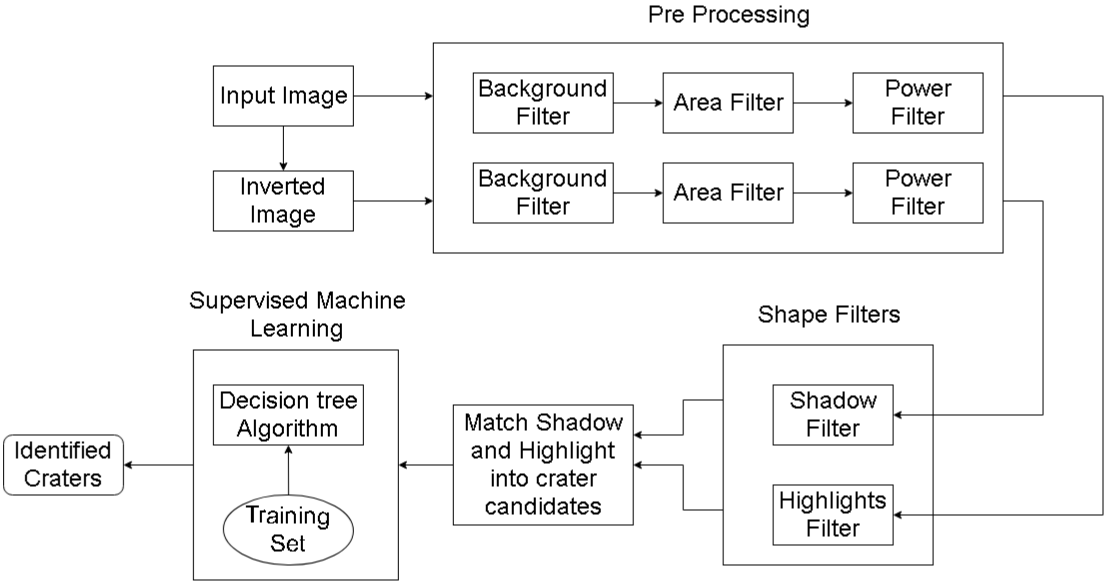


Figure 1.4: Identification of craters [31]

through DEM images. The algorithm also estimated the degradation rate by estimating the variation in intensities at the edge of the craters.

1.1.2 DEM based Techniques

Optical images are sensitive to elevation and azimuth angles of the sun; hence at different times, the shadow patterns may interfere with the shadow and bright parts of images. As a result, a few of the CDAs are based on non-optical images, in particular, DEM (Digital Elevation Model) data for crater detection. Authors proposed an approach for unsupervised crater detection algorithm using Digital topography data in [37]. The first step was crater rim indication from profile curvature which is based on the change in slope angle. To reduce the computational cost, the crater site is divided into segments using flooding algorithm, resulting in a binary image. Finally, Hough transform is used to detect the impact craters. Authors in [38] proposed an automated method for cataloging impact craters, by using similar approach in [37]. Zuo et al. in [39] generated contours on coarsened DEM for automatic crater detection. Authors in [40], proposed Rotational Pixel Swapping Method (RPSW) [41] which enhances rotationally symmetric patterns like circles and removes the unsymmetrical patterns from a binary image. The candidate regions are extracted by thresholding DEM/DTM images based on slope and angle values.

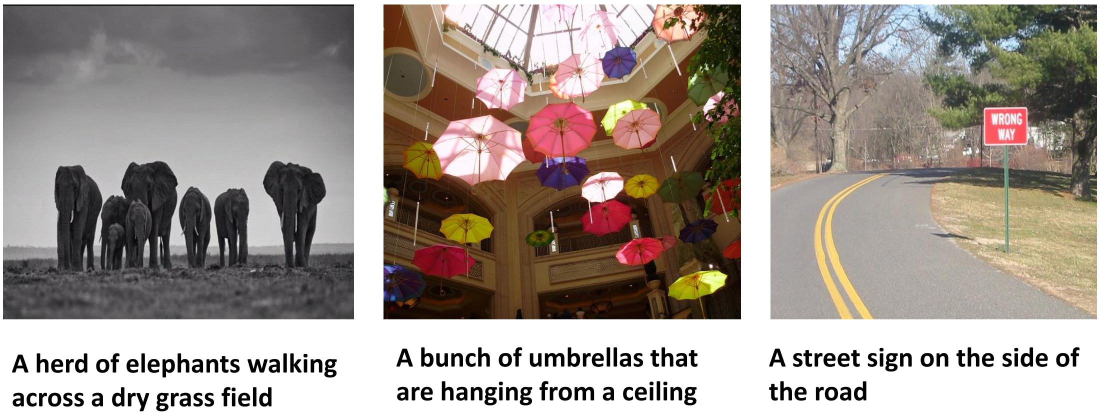
1.1.3 Data Driven Approaches

Availability of a large amount of data and advances in deep learning technology has improved the state of the art performance in many computer vision related tasks with the help of data-driven approaches. Deep Learning approach and CNN's have made it possible to get the accuracy as close to human accuracy in classification related tasks. Some of the very recent studies have also attempted to solve the crater detection problem using CNN. Authors in [42, 43], compared CNN model, named MarsNet which comprised of 5 networks trained to detect objects of various size and SVM using Histogram of Oriented Features (HOG) for classification of impact craters and Volcanic Rootless Cones (VRCs) from HiRISE images of Martian surface, concluding CNN generalizes well on the training data. Using Bandeira's dataset [44], a comparative study is performed between CNN features and previously generated handcrafted features [45, 31, 32]. The proposed algorithm outperformed the existing algorithms resulting in an average accuracy of 89 %. State of the art Faster RCNN model for real-time object detection was used in [46]. The dataset comprises 300 manually marked images of terrain features and 100 images of objects of different classes such as mountains, geysers, and craters.

Chapter 2

Convolutional Neural Networks

Convolution neural network (CNN) is one of the finest and the most popular invention in the field of deep learning and is of great importance in the field of computer vision and Natural Language Processing. The prominence of CNN's grew in the year 2012 when Alex krizhevsky [47] won the IMAGENET challenge. The challenge was to classify the IMAGENET dataset consisting of 1000 classes containing 1.3 million images. Krizhevsky used AlexNet, a CNN based network to train the dataset and the classification error was reduced from 26% to 15%. Since then, CNN's have been widely used in applications such as face recognition, scene labeling, human pose estimation, speech recognition, medical imaging, etc.



A herd of elephants walking across a dry grass field

A bunch of umbrellas that are hanging from a ceiling

A street sign on the side of the road

Figure 2.1: Scene classification from CNN [48]

In Figure 2.1, a neural network can caption the scene such as "A bunch of umbrellas that are hanging from a ceiling," whereas in Figure 2.2 CNN can detect objects in an image such as human, dogs, bicycle, etc.

Here, we give an intuitive understanding of the need for Convolutional Neural Networks. The task of image classification is to input an image to the network, and the output can be a probability of classifying an object into one of the classes or the classified pixels as in the case of semantic segmentation. Humans learn from the task of pattern recognition from birth, such as color and texture differences. Thus, it is not

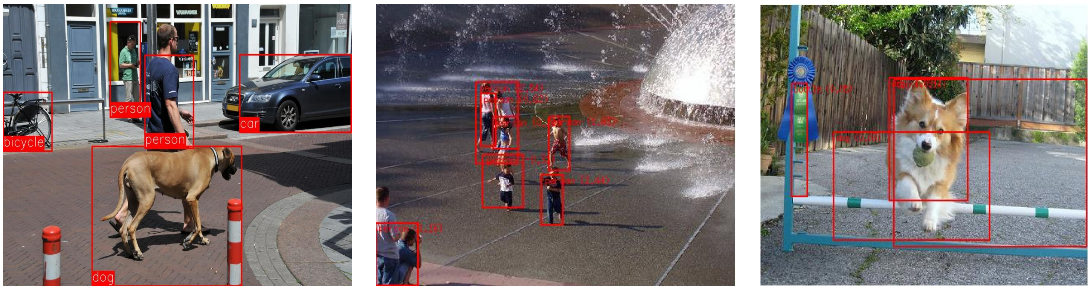
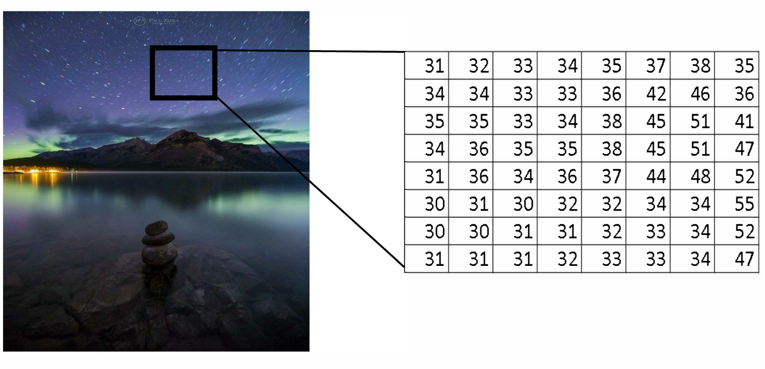


Figure 2.2: Object detection from CNN [49]

difficult for humans to differentiate between a dog or a cat or different types of cats. But when a network sees an image (Figure 2.3), what it sees is an array of pixel values. For an RGB image of size 100x100x3, each pixel value will indicate the intensity value at the particular point. We want the system to mimic how humans visualize and learn to differentiate the things around them. The idea is to make the network learn various patterns in the images and how the pixel values are related.



31	32	33	34	35	37	38	35
34	34	33	33	36	42	46	36
35	35	33	34	38	45	51	41
34	36	35	35	38	45	51	47
31	36	34	36	37	44	48	52
30	31	30	32	32	34	34	55
30	30	31	31	32	33	34	52
31	31	31	32	33	33	34	47

Figure 2.3: Image as seen by the network

Convolutional Neural Networks are inspired from animal visual cortex and hold great importance for the machine learning practitioners. Hubel and Weisel in 1959 experimented [50] on visual cortex of cats which gave some fascinating results. Cortex has a complex arrangement of cells which respond to the stimuli only in a restricted region of the visual field known as the receptive field. These small regions overlap to form the complete visual field. The first network on this finding to be simulated on a computer was proposed by Fukushima [51]. But the first fully connected CNN was LeNet, which was developed for the classification of the handwritten digits and also implemented backpropagation algorithm. However, due to unavailability of GPU technology during that time, it failed to work on complex problems.

Artificial Neural Networks (ANNs), are a type of feedforward network, analogous to the human nervous system or “neurons” and hold great importance for the machine learning practitioners. A simple ANN consists of series of layers, each consisting of multiple neurons connected to each other forming the network. The input layer feeds the data to the hidden layers, or the intermediate layer, where the training takes place and the output layer which generates the trained targets. The output of a neuron be represented as follows,

$$y_i = f(w^T x_i + b)$$

where x_{ij} is the input to the neuron, w is the training weights, and b is the offset to the network.

The weights are randomized initially, and the network is trained to best approximate the target function, by computing the loss function and backpropagating it through the network. The loss function can be the mean square error, mean approximation error, etc. The output is bounded by an activation function f such as sigmoid between [0,1] for binary class classification or a softmax activation is used for multi-class classification. The complexity of the network increases with the increase in the number of the hidden layer which in turn increases the number of parameters. Convolutional Neural Networks or CNN's differ from the traditional Nets in terms that the former exploits the spatial correlation in the input data be it to time series or image data, and has fewer parameters from ANN due to parameter sharing and sparse connection in the model. CNN's extracts feature at the different layer where the complexity of the features increase as we move from the initial hidden layer to the output layer. A typical CNN architecture is shown in the Figure 2.4.

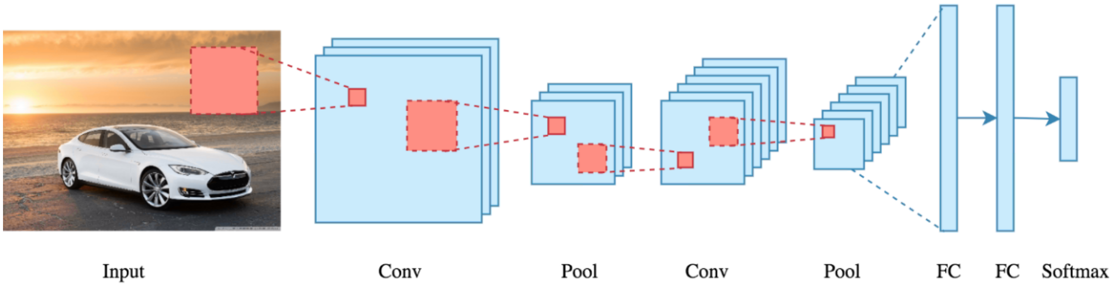


Figure 2.4: CNN architecture

CNN architecture consists of mainly three layers viz the Convolutional layer, pooling layer, and the fully connected layer.

2.1 Convolutional Layer

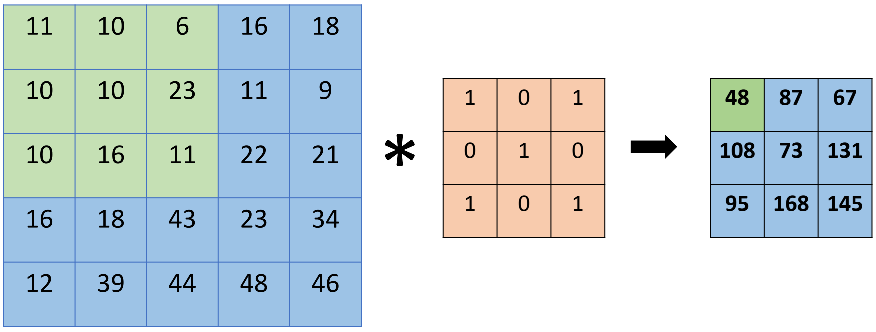
The convolutional layer, or the core building block of the CNN, outputs a function $S(x, y)$ on a 2-Dimensional input I, such as images in our case, with a kernel k of size $(2m + 1)$x$(2n + 1)$ as,

$$S(x, y) = (I * k)(x, y) = \sum_{i=-m}^{m} \sum_{j=-n}^{n} I(x + i, y + j)k(i, j)$$

where x, y are the image coordinates. The layer consists of multiple filters which slide over the image to generate a feature map. The output size of each convolutional layer can be calculated as,

$$W = (W - F + 2P)/S + 1$$

where W is the size of the image, F is the filter size, P is the padding size and S is the stride. Keeping stride and padding greater than 1 is a way to reduce the dimensionality of the feature map generated. Consider the 5x5 image in the Figure 2.5, convolved by a 3x3 filter which generates an output activation map of size 3x3 without any padding and stride of 1.



$$\begin{bmatrix} 11 & 10 & 6 & 16 & 18 \\ 10 & 10 & 23 & 11 & 9 \\ 10 & 16 & 11 & 22 & 21 \\ 16 & 18 & 43 & 23 & 34 \\ 12 & 39 & 44 & 48 & 46 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \Rightarrow \begin{bmatrix} 48 & 87 & 67 \\ 108 & 73 & 131 \\ 95 & 168 & 145 \end{bmatrix}$$

Figure 2.5: Feature map from convolutional layer

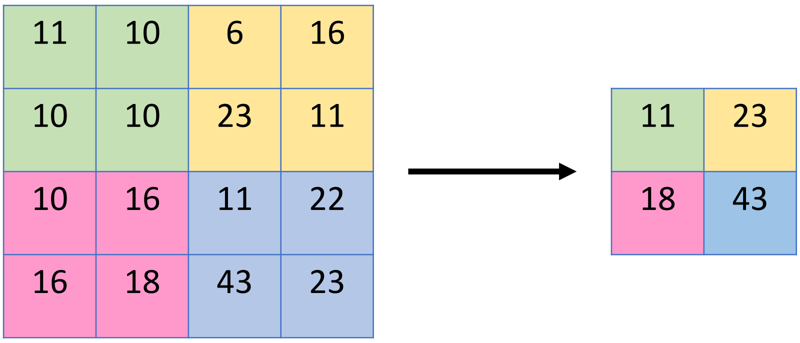
Multiple convolution operations are performed in a single layer to produce a series of activation map.

2.2 Non- Linearity Layer

The convolution layer is followed by a non-linear activation function, to increase nonlinearities in the network, such as ReLU (Rectified Linear Unit) which is also called the detector stage [52]. The intuitive idea behind introducing the non-linearity is that the real-world systems are non-linear and if we avoid this step, it is merely a scaled version of the input. This stage reduces the number of parameters and enhances the computations.

2.3 Pooling Layers

Pooling layers which can be the max pool or average pool, take the maximum or average inside a small rectangular window, to make the network translational invariant. It reduces network parameters and increases the computational power of the system. Here in this case (Figure 2.6), the max pooling is done on the feature map with a grid size of 2x2.



11	10	6	16
10	10	23	11
10	16	11	22
16	18	43	23

11	23
18	43

Figure 2.6: Max pooling

2.4 Fully Connected Layer

The last pooling layer is usually followed by a single or multiple fully connected layers which are similar to traditional Nets. The neurons in fully connected layers are connected to all the activations in the previous layers. The output fully connected layer contains as many neurons as the number of classes to be classified. For example, in case of CIFAR-10, the output fully connected layer has ten output.

Chapter 3

Methodology

The problem of crater detection can be treated as an object detection problem where there is only one object, crater. However, conventionally object detection is performed on mostly RGB images where objects are distinguished based on their colors, intensity, shadow, and association. Craters, on the other hand, are discriminated based on different properties, e.g., shadow-brightness pattern based on the lighting direction, their elevation profiles from DEM, slope values, etc. Also, the popular COCO dataset for object detection contains 7.7 objects per image, on an average [53], while the lunar surface is dominated by impact craters of varying sizes which ranges from a few hundred meters to hundred kilometers in diameter. For crater detection, the main aim is to reduce the false positives thereby increasing precision even if the recall may suffer. Given the similarity between object detection and crater detection, we adopted a methodology inspired by R-CNN [1]. The first step is to use a category independent algorithm to generate region proposals. In our case, the best results were generated from a graph-based algorithm by Felzenswalb & Huttenlocher [54]. The second step is to train all the proposals using a CNN model generating class labels and classification scores. Finally, we use a graph-based post-processing method to improve localization precision.

3.1 Region Proposal

Among the various methods for traditional object detection, one common method is to test multiple locations in the image for the presence of a given object [55, 56]. The biggest disadvantage of this approach is that there are various windows sizes, image locations to try, making the computational complexity of algorithm too high and infeasible. One solution proposed by [1] was to test only in certain regions of the image, known as region proposals, instead of trying out all possible locations. There are various algorithms to extract region proposals for an RGB image. For crater detection, however, we are using ortho images, DEM and slope images hence, we found the graph based algorithm [54] most suitable for our task. The method is to group pixels taking into account the fact that similar regions can have texture variations and also considering the global image characteristics. The idea is to merge regions only if the intensity variation across the boundary of two regions is greater than variation in any one of the regions.

Let us define a few terminologies related to graphs. Consider a undirected graph (Figure 3.1) $G = (V, E)$, with vertices or nodes $v_i \in V$ and E be the edges connecting the two vertices (v_i, v_j). Here all the edges are bidirectional. The minimum spanning tree (MST) of a graph is a tree in the graph where all nodes are connected with minimum weight.

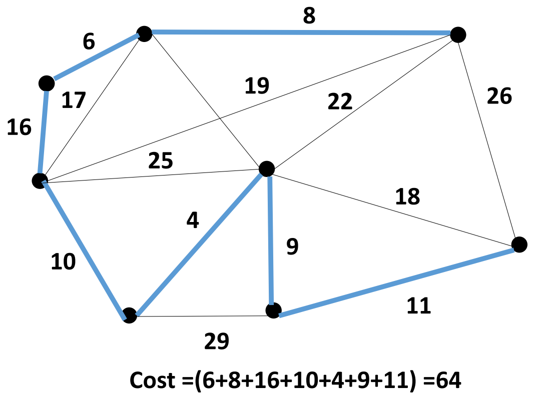


Figure 3.1: Minimum Spanning Tree (MST)

In case of image segmentation, the graph is defined with vertices as image pixels and edges as the weight which measures the dissimilarity between the pixels (which can be color, location, etc.). Segmentation is defined as partitioning the image pixels into components based on the similarity or dissimilarity of the pixels.

3.1.1 Predicate for a Boundary

To evaluate for whether there is a boundary between two components a predicate, D is defined. The predicate is defined such that it measures the dissimilarity between the two regions or components to the variation or dissimilarity between the elements of the components. The *'internal difference'* of a region is defined as maximum weight $w(e)$ of a Minimum Spanning Tree (MST),

$$Int(R) = \max_{e \in MST(R,E)} w(e) \tag{3.1}$$

The intuition is that since MST spans the vertices with lowest edge weight, $w(e)$ will define the maximum variation of the region. The difference between two regions R_1, R_2 or components is defined as the minimum weight edge linking the two components,

$$Dif(R_1, R_2) = \min_{v_i \in R_1, v_j \in R_2, (v_i, v_j) \in E} w((v_i, v_j)) \tag{3.2}$$

The difference Dif is ∞ if no edge is linking the components. We want that the difference Dif between the two components should be greater than the internal difference of either component, $Int(R_1)$ and $Int(R_2)$. That is, the predicate is true only if the above condition is satisfied.

$$D(R_1, R_2) = \begin{cases} \text{true} & \text{if } Dif(R_1, R_2) > MInt(R_1, R_2) \\ \text{false} & \text{otherwise} \end{cases} \tag{3.3}$$

where $MInt(R_1, R_2)$ is the minimum internal difference, defined as,

$$MInt(R_1, R_2) = min(Int(R_1) + \tau(R_1), Int(R_2) + \tau(R_2)) \tag{3.4}$$

Consider the case when the component size is 1, then $Int(R) = 0$. Thus, $Int(R)$ is not a good estimate for boundary for small region. A threshold function is defined as which depends on the size of the component as,

$$\tau(R) = k/|R| \tag{3.5}$$

where R is the size of the component. If the value of k is large, then larger components are preferred in the segmented output.

3.1.2 Case 1: Grid Based Segmentation

Here monochrome images are considered, and for color images, the algorithm is run for three different channels, and the resulting components are the intersection of the components for each channel. Edges or pixels are connected to the neighbors which satisfy 8-connectivity. The weight w between two edges is defined as the absolute intensity difference between two-pixel values,

$$w(v_i, v_j) = |I_{(x_i,y_i)} - I_{(x_j,y_j)}| \tag{3.6}$$

where $I_{(x_i,y_i)}$ is the pixel value of i^{th} pixel. The Figure 3.2 shows the result for Grid based segmentation.

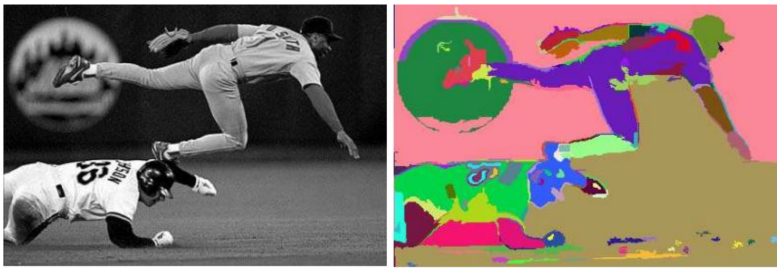


Figure 3.2: Result for grid-graph segmentation [54]

Here, as we can see the algorithm is not able to capture the non-local properties in the image. The grass beneath the player is segmented into a different region since the algorithm is capturing only the local variation in the intensity.

3.1.3 Case 2: Nearest Neighbor Graphs

The algorithm for the nearest neighbor graph is based on mapping a pixel to 5-D feature space. The weights of the edges are computed as the distance between (x, y, r, g, b) vectors for two pixels, where (x, y) is the pixel coordinate and (r, g, b) is the color value at that pixel location. The vertices are connected by finding m nearest neighbor using ANN algorithm [57]. Here, the value of m is set as 10. The nearest neighbor algorithm performs better than the grid graph and can capture global characteristics in the image. Considering the same image used in grid graph, we can see in the Figure 3.3, the grass is segmented into a single region although some part of the grass areas are disconnected in the image. We have used the latter strategy for generating region proposals in our framework.

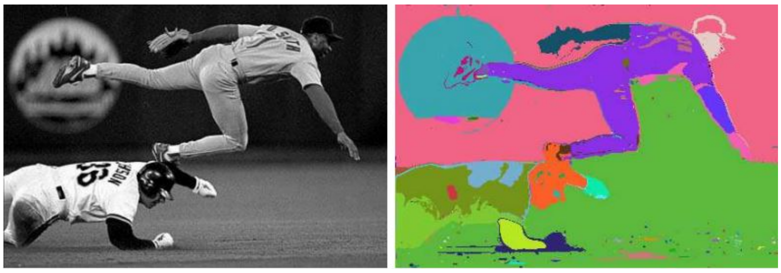


Figure 3.3: Result for nearest neighbor segmentation [54]

3.1.4 Extraction of Region Proposal

Since the lunar images are panchromatic images, there is no meaning of r,g,b values here so instead we have used the intensity (I), elevation (E) and slope values (S), consequently, the weights will be computed as distance between vectors (x,y,i,e,s) for two pixels. For this, we have constructed 3-channel RGB images, by supplying the ortho (intensity) images in the red channel, DEM (elevation) images in the green channel and slope raster in the blue channel. The slope images were calculated from DEM images, using ArcGIS software. The red channel in our images corresponds to the high intensity in the bright part of the crater and around the rim near the ejecta. Similarly, the green channel will correspond to the high elevation in the surrounding region of the crater and crater rim, and the blue channel will correspond to the high slope of the crater walls or in other words steep changes in adjacent pixels of the green channel. The region proposals were generated from these images varying k from $100 - 400$ with an interval of 50 and sigma varying in the range of $0.5 - 0.9$. Here, sigma (σ) is the amount of Gaussian smoothing to reduce digitization effect. The images for crater class were generated from these region proposals for which the IOU with ground truth was at least 0.6 and rest of the images were categorized into the non-crater class or the background class. We can have multiple region proposals for a single crater which will satisfy the IOU threshold. IOU between two regions b_i and b_j is defined as,

$$IOU(b_i, b_j) = \frac{area(b_i \cap b_j)}{area(b_i \cup b_j)} \tag{3.7}$$

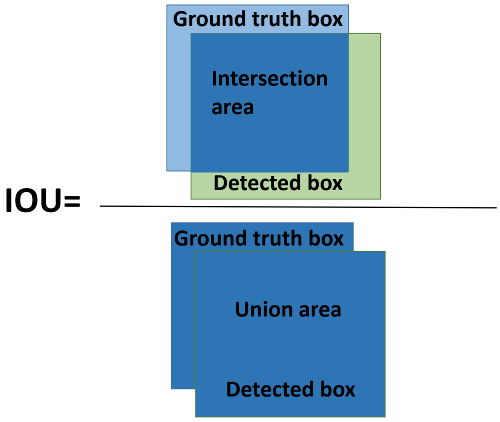


Figure 3.4: Intersection Over Union (IOU)

where, $area(b_i \cap b_j)$ is the area of overlap between the boxes b_i, b_j and $area(b_i \cup b_j)$ is the area of union of the boxes (Figure 3.4). To encounter this, we have performed non-maximal suppression (NMS) with the ground truth boxes and only those region proposals which had maximum intersection over union (IOU) with ground truth are selected. The images were warped to a pixel size of 227x227 following the procedure by [1]. The non-crater class has a much larger number of images compared to the crater class. To maintain the class balance, we selected as many numbers of non-crater images as crater images, randomly. It should be noted that the before saving the images of size 227x227, all the channels are scaled between $0 - 255$ separately and converted to unsigned-integer with 8-bit.

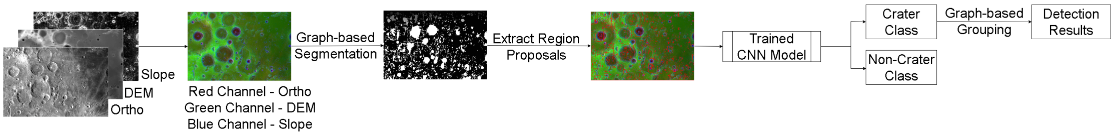


Figure 3.5: Extracting Region Proposals

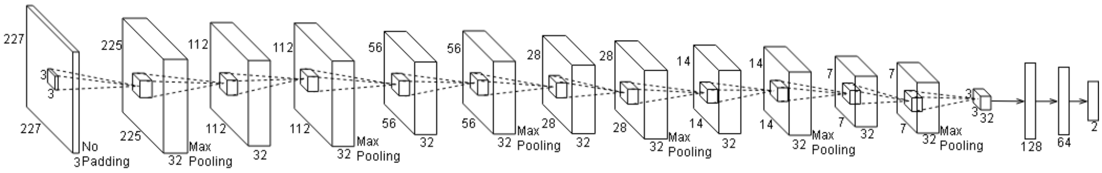


Figure 3.6: CNN architecture

3.2 CNN Architecture

The CNN architecture (Figure 3.6) used to classify crater vs. non-crater images is inspired from the VGGNet [58]. Our CNN consists of twelve convolution layers, each followed by batch normalization and ReLU activation layers. There are three fully connected layers after the last convolution layer. Max-pooling layers are added after the two consecutive convolution layers. To reduce the overfitting in the network, we have used the dropout of 0.2, and the kernel regularization strength is set as $1e^{-3}$. The first convolution layer takes the input of size 227x227x3. There are total 32 filters in each layer of size 3x3 with stride 1. The first fully connected layer has 128 neurons and the second fully connected layer has 64 neurons. The output of this layer goes to the softmax with 2 neurons which provides the distribution into crater vs. non-crater classes. Adadelta [59] with batch size 32 is used for optimizing weights and the binary cross-entropy loss is used to penalize the wrong predictions. The learning rate is initialized to be 1.0 and decayed as the function $lr/epochs$, where lr is learning rate and epochs are set to be 100. The experiments are performed on NVIDIA GeForce GTX 1080 GPU with 8 GB memory. The model with best validation accuracy is chosen as the best model and used on test data.

3.3 Post Processing

Often a single crater may be detected by multiple boxes from which we would like to derive the best-fit bounding box. In literature, people have used Non-Maximum Suppression (NMS) to handle such cases. NMS performs suppression by selecting box with the highest probability. Instead of this, we have designed an algorithm to group the boxes based on the graph and derive the best fit from these grouped boxes.

3.3.1 Non-Maximal Suppression

In traditional object detection, Non-Maximal Suppression is one of the popular approach to remove redundant boxes. NMS is a class-specific approach for selecting the box with highest probability and suppress all other boxes which have very high IOU with the selected box. Our implementation for NMS at test time is as follows:

Suppose there are N total boxes, each with probability p_i where $i \in \{1, 2, ..., N\}$ for craters. Select the box b_h with highest probability p_h. Compute the IOU of b_h with b_i where $i \in \{1, 2, ..., N\} \setminus \{h\}$.

All the boxes, b_i, with $IOU \geq \delta$ are suppressed. This process is repeated until all the boxes are either selected or suppressed. We have set the value of δ as 0.6 to perform NMS at test time.

3.3.2 Graph-based Approach

We have adopted the graph-based approach to group the craters. The craters which have high overlap with each other are out in the same group. We construct an undirected acyclic graph, $G = (V, E)$ where V corresponds to each box, and E corresponds to edges between the craters depending on whether they are connected or not. Given N vertices, theoretically, there are a total of $N(N-1)/2$ pairs of vertices possible. For a given pair (v_i, v_j) we check whether they are connected or not. If the path between v_i and v_j exists, we skip the pair otherwise we compute the IOU between the two. If the $IOU \geq 0.3$, an edge is added between v_i and v_j. Once the graph is computed, we extract the connected components $\{c_1, c_2, ..., c_n\}$ from the graph which corresponds to groups $\{g_1, g_2, ..., g_n\}$. The worst case complexity of this grouping is $O(n^2)$ when each component is a single vertex. However, as the graph construction progresses, we skip the pairs for which the path already exists, thereby reducing the time required to construct the graph. Since for each image we get such cases where grouping is required, our complexity is always $\leq O(n^2)$. Further, since our object is crater here, the connectivity checks on boxes which are disjoint can be ignored thus reducing the time required.

Once the crater groups are derived, we derive the new box for that group using weighted averaging. We use the probability scores for each box as their weight. The region proposals give the rectangles as output hence, our detection boxes may also be rectangles. We force these rectangles to be square by simply taking the average of width and height. Therefore, given a box b with dimensions (x, y, w, h) which are coordinates of top-left point and width and height if the box b, the crater dimensions are derived as $(x + (w+h)/4, y + (w+h)/4, (w+h)/4)$ corresponding to (x_c, y_c, rad). Now, let us assume that the true crater $\hat{c}$ is described as $(\hat{x}, \hat{y}, \hat{r})$, where $\hat{x}$ and $\hat{y}$ are the center coordinates and $\hat{r}$ is the radius. Let there be a group g which contains boxes around the true crater $\hat{c}$. The dimension $\hat{x}$ of the true crater $\hat{c}$ is computed as,

$$\hat{x} = \frac{\sum_{i=1}^{n} p_i x_i}{\sum p_i} \tag{3.8}$$

where p_i is the probability of the i^{th} crater in group g, x_i is the center coordinate x of the i^{th} crater in group g, and n is the number of craters in group g. Similarly the center coordinate $\hat{y}$ and radius $\hat{r}$ is calculated as,

$$\hat{y} = \frac{\sum_{i=1}^{n} p_i y_i}{\sum p_i} \tag{3.9}$$

$$\hat{r} = \frac{\sum_{i=1}^{n} p_i r_i}{\sum p_i} \tag{3.10}$$

Chapter 4

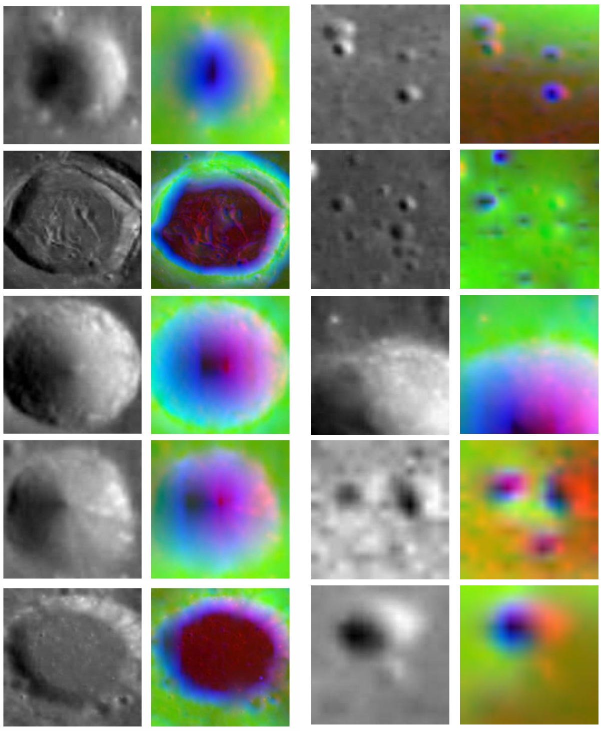
Experimental Results

4.1 Dataset Generation

Ortho images and DEM from Lunar Orbiter Laser Altimeter (LOLA) camera of LRO (Lunar Reconnaissance Orbiter) spacecraft were used in this work. The resolution for ortho images and DEM is 100 pixels/m and 118 m/pixels, respectively. The regions above ±35° latitude were excluded to reduce projection errors. Currently, no public dataset is available which can be used for classification of craters, hence, we have generated our own dataset from the LRO images.

Ortho, DEM, and slope raster were used simultaneously in our work. We used the catalog by [60] to serve as the ground truth dataset (Figure 4.1). This catalog has been generated by using modified DEM and a CDA based on Hough transform on LRO dataset and two Chandrayaan-1 images. This is the most complete catalog for craters $\geq$ 8 km in diameter. Ortho images and DEM from Lunar Orbiter Laser Altimeter (LOLA) camera of LRO (Lunar Reconnaissance Orbiter) spacecraft were used in this work. The resolution for ortho images and DEM is 100 pixels/m and 118 m/pixels, respectively. The regions above ±35° latitude were excluded to reduce projection errors. Currently, no public dataset is available which can be used for classification of craters. Hence, we have generated our dataset from the LRO images.

The classification database corresponding to crater vs. non-crater classes was generated in our work with the help of this ground truth catalog. It should be noted here that the catalog by [60] had many craters which were not marked or whose marking were shifted from the actual crater position. The shift in crater markings was fixed manually before generating the classification dataset of image size. We have restricted our crater ground truth to 1 km in diameter. So any crater or region proposal with size less than 1 km in diameter was discarded. The sample set of our dataset has been shown in the Figure 4.1. The images in the left show only the ortho part or the red channel. Partially overlapped craters were categorized into negative class. Images on the left show only the ortho part or the red channel. Partially overlapped craters were categorized into negative class.



(a) Crater Class (b) Background Class

Figure 4.1: Examples from our dataset (The images in the left are the ortho images or the red channel and to the right side are the 3 channel- Ortho images, DEM, and slope images.)

Each sample image for training is generated by randomly cropping input images, each having a size ranging from 1857x4283 to 5274x5264 pixels. From the entire lunar surface mosaic, we have taken only the portion which had the lighting direction from left to right, i.e., the crater intensity increased from left to right (green and the red shaded area in Figure 4.2). The area shaded with green color is used for training and area shaded with red color is used for test purpose. The yellow shaded area in Figure 4.2 is not used in this experiment as the intensity direction in that part of the image is different than the rest. The area outside the shaded region has been discarded to decrease the projection error.

We have first generated region proposal as explained in the methodology by varying the sigma from 0.5 to 0.9 with the steps of 0.1 and varying minSize (minimum size of the region proposal to be generated) and parameter k from 100 to 400 in the steps of 50. Sigma, minSize, and k are the parameters required by the segmentation algorithm of [54]. MinSize controls the minimum size of the component to be segmented. From the list of generated region proposals, the proposals which have IOU at least 0.6 with the ground truth, are considered for crater class, and rest of the proposals are considered for non-crater class. It may happen that for a single crater, multiple region proposals might be generated which would satisfy the IOU threshold, in such cases, we have selected

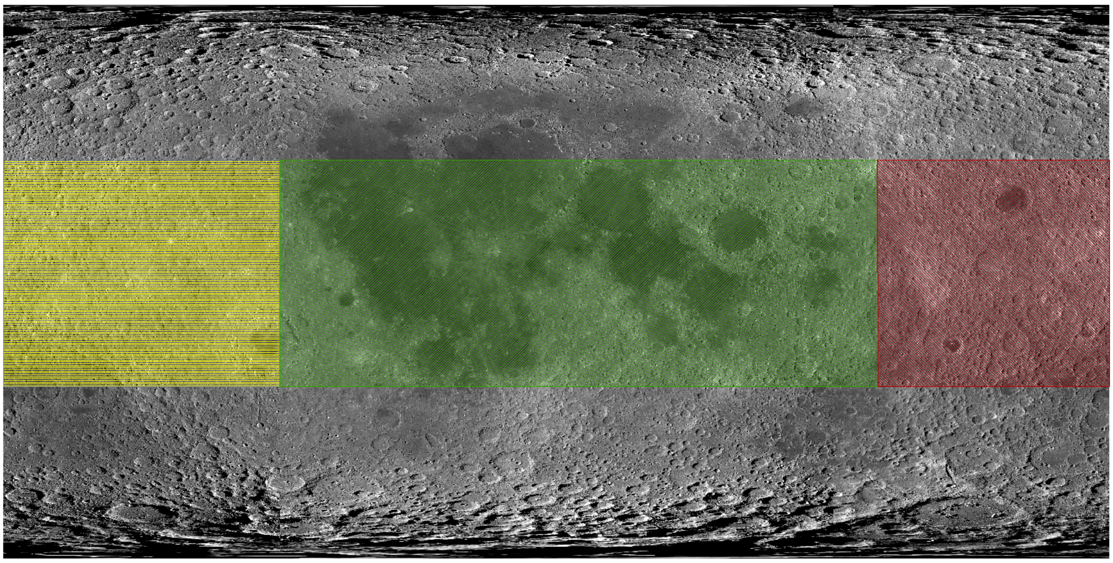


Figure 4.2: Devision of LRO mosaic in train and test dataset (green and red shaded area corresponds to the training and test area)

only those region proposals which had maximum IOU with ground truth and rest of the proposals for that particular crater are discarded. The images were warped to the pixel size of 227x227 following the procedure by [1]. The non-crater class has a much larger number of images compared to the crater class. To maintain the class balance, we selected as many numbers of non-crater images as crater images, randomly. It should be noted that before saving the images of size 227x227, all the channels are scaled between 0-255 separately and converted to unsigned-integer with 8-bit.

4.2 Evaluation of Trained Model

- To evaluate the goodness of predicted crater boxes, we compute mAP (mean Average Precision), a standard widely used for evaluating object detection algorithms [61].

- For a set of boxes, $B_o = \{b_n | 1 \leq n \leq N\}$, where N is total number of detected boxes in an image after post processing, and $B_g = \{b_m | 1 \leq m \leq M\}$, where M is the number of ground truth boxes for the same image, mAP is defined as,

$$mAP@U = \frac{1}{|U|} \sum_{i \in U} AP_i \tag{4.1}$$

where, $U = \left\{ \Delta_{min} + s\delta | \ \forall \ s \in \{0, 1, 2, ..., \frac{\Delta_{max} - \Delta_{min}}{\delta}\} \right\}$

$$AP_i = \frac{|TP_i|}{|TP_i| + |FP_i|} \tag{4.2}$$

where,

$$TP_i = \{\forall b_j \in B_o | IOU(b_j, B_g) > i\}, IOU(b_j, B_g) = \max_{b_m \in B_g} \{IOU(b_j, b_m)\}$$

and $FP_i = B_o - TP_i$

We evaluate the mAP@0.5 defined as,

$$mAP@0.5 = AP_{0.5} \tag{4.3}$$

where, $U = \{0.5\}$ and $TP_{0.5}$ is the set of all the detected boxes having IOU with any of the ground truth box greater than 0.5.
mAP@[0.5,0.9] with δ=0.05, $\Delta_{min} = 0.5$, and $\Delta_{max} = 0.9$, is defined as,

$$mAP@[0.5, 0.9] = \frac{1}{9} \sum_{i \in U} AP_i \tag{4.4}$$

where, $U = \{0.5 + 0.05s| \ \forall \ s \in \{0, 1, 2, ..., 8\}\}$.

4.3 Crater Identification on the Lunar Surface

To assess the performance of our algorithm, we test it on 26 images which are not used in the entire training process. These images come from the area shown by red shading in Figure 4.2 and are of average size 2727x5097. We have used the same setting for k, minSize, and sigma, as used for region proposals as used for train dataset and the proposals, are warped to size 227x227 and score them using the trained CNN. We apply the post-processing strategies NMS and graph-based grouping to those proposal boxes which are predicted as craters by our CNN. Ground truth boxes for craters have been manually annotated using crater tools.

The baseline detection results on LRO dataset are shown in the Table 4.1. Figure 4.3 shows the results of graph-based grouping on a test image. The strict measure $mAP@[0.5, 0.9]$ is very low as the even though our algorithm performs well for differentiating the crater with non-crater class, accurate localization of crater box is a highly challenging problem and not truly solved. Hence, as we increase the IOU above 0.75, the AP decreases significantly and thereby decreases the mAP as well.

Considering the catalogue [60] as the ground truth, our algorithm is able to detect nearly $\sim$ 1000 craters per image in the test area (red shaded area in Figure 4.2).

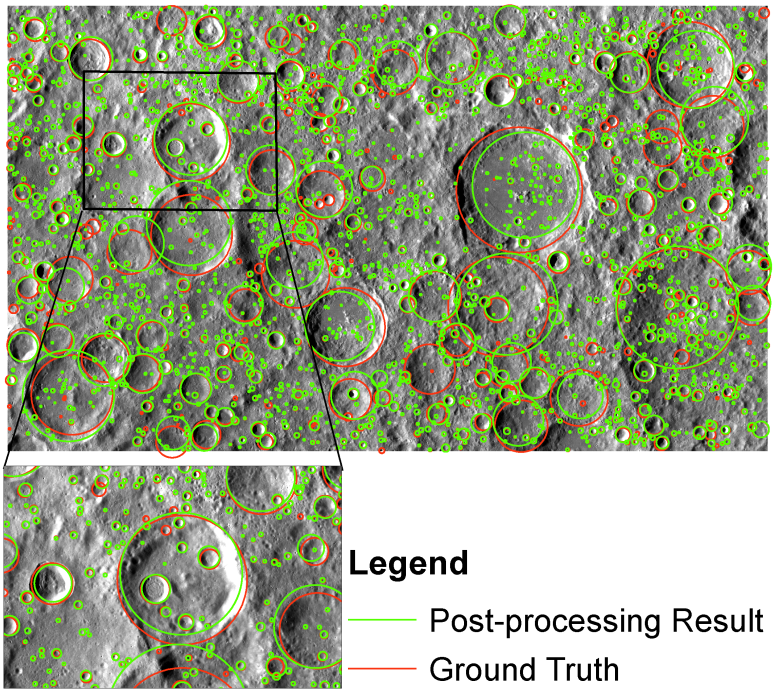


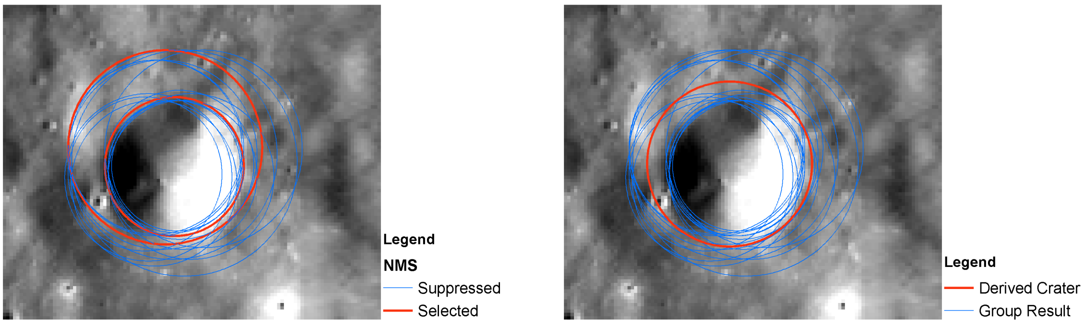
Figure 4.3: Detection result on LRO dataset (green circles are the results of the proposed algorithm, and red circles are ground truth which are manually annotated)

Table 4.1: Detection results (%) on LRO dataset

Training Data (LRO)	Post Processing	LRO test	
		mAP@.5	mAP@[.5 , .9]
LRO Train	NMS	74.78	37.40
LRO Train	Graph Based Approach	**87.48**	**48.84**

4.4 Comparing NMS with Graph-based Grouping

NMS is a popular strategy to remove the duplicate detection of one object [55]. However, in our case even after filtering the results with NMS threshold at 0.6 IOU, there were many duplicates per crater. (Figure 4.4(a)). We have also noticed that for large craters, detections with part of the crater may have a higher probability than the actual crater. For this reason, we have developed a grouping strategy to derive crater dimensions from all the redundant detection based on their probability (Figure 4.4(b)). It can be observed from Table 4.1 that using the graph-based grouping has increased the accuracy by ∼12 points. In further discussion, we have reported the accuracy of graph-based grouping, unless explicitly stated.



(a) (b)

Figure 4.4: (a) Result After NMS where multiple rings for single craters can be seen in red markings (b) Result of Graph based grouping

4.5 Effect of Resolution

India's first lunar mission, Chandrayaan-1, has collected the surface and elevation data at a very high resolution of 5 m/pixel and 10 m/pixel for ortho images and DEM, respectively. We have tested our model on these images as well to analyze how the model behaves with different resolution data. The Chandrayaan-1 dataset has increased the resolution by 20 fold than LRO, however entire lunar surface mosaic is not available, and discontinuity in adjacent strips restricted us from using the Chandrayaan-1 data for training itself.

Applying the same model, without altering the parameters, the $mAP@0.5$ is 48.06% and the $mAP@[0.5, 0.9]$ is 27.49%. We observed that due to the very high resolution of the Chandrayaan-1 dataset, edges of the image have multiple craters which are not fully contained in the image and this results into less accurate detection near the edges by our algorithm. Once we remove such craters which are not fully contained in the image, our accuracy, $mAP@0.5$ changes to 62.23% and $mAP@[0.5, 0.9]$ changes to 35.60% (Table 4.2).

Table 4.2: Detection results (%) on Chandrayaan-1 dataset

Training Data (LRO)	Post Processing	Chandrayaan-1 test	
		mAP@.5	mAP@[0.5 ,0.9]
LRO Train	NMS	56.01	29.15
LRO Train	Graph Based Approch	**62.23**	**35.60**

4.6 Testing on Martian Terrain

In this section, we discuss the generalization capability of our model. Compared to Moon, Mars surface has a different composition, gravitational effects, and the history of impacts. The major difference between Moon and Mars is the presence of the atmosphere on mars which would also play a significant role in the degradation of craters.

Table 4.3: Detection results (%) on Martian dataset

Training Data (LRO)	Post Processing	h0905_0000 HRSC	
		mAP@.5	mAP@[.5 , .9]
LRO Train	NMS	74.69	42.67
LRO Train	Graph Based Approch	**84.78**	**59.18**

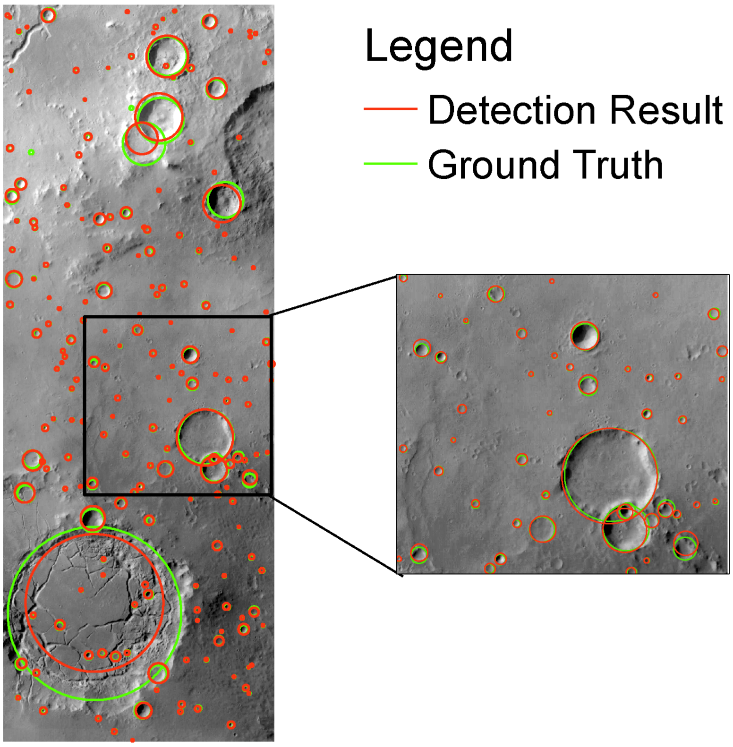


Figure 4.5: Detection results on Martian data. (The Green circles represent the Ground Truth, and Red circles are the detection result)

The Martian surface shows fewer impact craters compared to the lunar surface. Therefore, the performance of our algorithm on Martian surface would be a good measure to show generalization capability of our model. Thus, we test our developed model on Martian image collected by HRSC (High-Resolution Stereo Camera).

In particular we have taken the panchromatic image, $h0905\_0000$ widely used by other researchers for crater detection [31, 25, 34], with resolution of 50m/pixel for ortho image and DEM. In Table 4.3, we report the results of transfer learning on Martian dataset, achieving 84.78% $mAP@0.5$ and 59.18% $mAP@[0.5, 0.95]$ using our graph-based approach as the post-processing step. We can see that our post-processing strategy of graph-based grouping works better compared to NMS giving $\sim$10 and $\sim$16 points higher accuracy than the later for $mAP@0.5$ and $mAP@[0.5 : 0.95]$, respectively. The results are shown in Figure 4.5, where it can be noticed that our model has successfully differentiated the "river" from the craters in the bottom left crater.

Chapter 5

Benford's Law and its Applications in Forensics

A lot of discoveries that have happened has resulted from the analysis of data. J Snow in 1854 noticed that the people suffering from Cholera are drinking water from the same tap, which led to the discovery that Cholera is spread by drinking contaminated water. The study was conducted long back before the knowledge of bacterias and viruses during that time, which further led to the development of theories in modern microbiology.

In today's world, technological improvement has led to low-cost sensors and higher storage capacity which has eventually led to unbridled installations of the sensor to acquire data. According to a study, researchers have predicted that 40 Zettabytes (40,000 Exabytes) of data will be generated by 2020, which is almost 300 times compared to 2005. The connection with data requires an ongoing effort to transform the data into something insightful. Examples include detection of Gravitational Waves, observation of the "God Particle," new drugs to prevent cancer, etc. One such way to analyze the pattern in data is through Benford's Law [62], which was originally introduced in 1881 by Simon Newcomb but acknowledged in 1938. One would typically assume the probability of occurrence of first digits in large datasets to be uniformly distributed. That is the probability of occurrence of 1's is same as of 9's. But the first digits of a large dataset followed some different observation. The law suggested that the numbers starting with the digit one will occur 30% of the time and follows a logarithmic curve. Considering it to be true, let us consider the population of top 232 countries in the year 2017 [63] as shown in the table 5.2. We tried to plot the first digit for the population data (as shown in the Figure 5.1). The frequency of occurrence of 1 is the highest which is about 29.7% (table 5.1), whereas actual law predicts 30.10%. A similar trend is followed with the other digits. This empirical study on the population data explains the Benford's law.

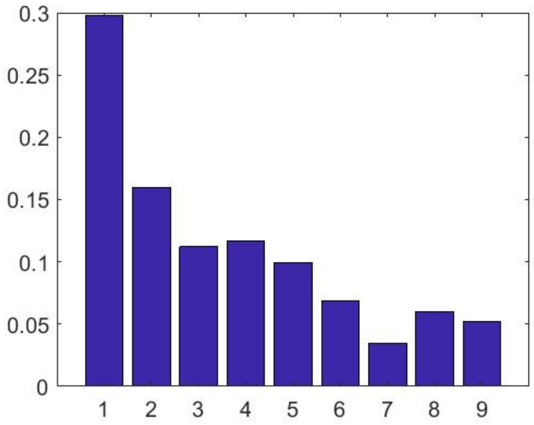


Figure 5.1: First Digit Distribution for the data in table 5.2

Digit	1	2	3	4	5	6	7	8	9
Frequency	69	37	26	27	23	16	8	14	12
Propotion	0.297	0.159	0.112	0.116	0.099	0.069	0.034	0.060	0.052

Table 5.1: Frequency of occurrence of First digit for data in table 5.2

5.1 Benford's Law

The history of Benford's Law dates back to the year 1881. Benford's Law was first observed by astronomer-mathematician Simon Newcomb, who noticed that the initial pages of his logarithmic table were more worn out than the last pages. Inspired by this observation, he published an article *Note on the Frequency of Use of the Different Digits in Natural Numbers* [65] in American Journal of Mathematics, 1881. The paper quotes,

> That the ten digits do not occur with equal frequency must be evident to anyone making much use of logarithmic tables and noticing how much faster the first pages wear out than the last ones. The first significant digit is oftener 1 than any other digit, and the frequency diminishes up to 9. The question naturally arises whether the reverse would be true of logarithms. That is, in a table of anti-logarithms, would the last part be more used than the first, or would every part be used equally ? The law of frequency in the one case may be deduced from that in the other. The question we have to consider is, what is the probability that if a natural number be taken at random its first significant digit will be n, its second n', etc.
> As natural numbers occur in nature, they are to be considered as the ratios of quantities. Therefore, instead of selecting a number at random, we must select two numbers, and inquire what is the probability that the first significant digit of their ratio is the digit n. To solve the problem we may form an indefinite number of such ratios, taken independently; and then must make

Country	Population (July ’17)	Country	Population (July ’17)
China	1409517397	Madagascar	25570895
India	1339180127	North Korea	25490965
United States	324459463	Australia	24450561
Indonesia	263991379	Ivory Coast	24294750
Brazil	209288278	Cameroon	24053727
Pakistan	197015955	Taiwan	23626456
Nigeria	190886311	Niger	21477348
Bangladesh	164669751	Sri Lanka	20876917
Russia	143989754	Romania	19679306
Mexico	129163276	Burkina Faso	19193382
Japan	127484450	Malawi	18622104
Ethiopia	104957438	Mali	18541980
Philippines	104918090	Syria	18269868
Egypt	97553151	Kazakhstan	18204499
Vietnam	95540800	Chile	18054726
Germany	82114224	Zambia	17094130
Democratic Republic of the Congo	81339988	Netherlands	17035938
Iran	81162788	Guatemala	16913503
Turkey	80745020	Ecuador	16624858
Thailand	69037513	Zimbabwe	16529904
United Kingdom	66181585	Cambodia	16005373
France	64979548	Senegal	15850567
Italy	59359900	Chad	14899994
Tanzania	57310019	Somalia	14742523
South Africa	56717156	Guinea	12717176
Myanmar	53370609	South Sudan	12575714
South Korea	50982212	Rwanda	12208407
Colombia	49065615	Tunisia	11532127
Kenya	49699862	Cuba	11484636
Spain	46354321	Belgium	11429336
Argentina	44271041	Benin	11175692
Ukraine	44222947	Greece	11159773
Uganda	42862958	Bolivia	11051600
Algeria	41318142	Haiti	10981229
Sudan	40533330	Burundi	10864245
Iraq	38274618	Dominican Republic	10766998
Poland	38170712	Czech Republic	10618303
Canada	36624199	Portugal	10329506
Morocco	35739580	Sweden	9910701
Afghanistan	35530081	Azerbaijan	9827589
Saudi Arabia	32938213	Hungary	9721559
Peru	32165485	Jordan	9702353
Venezuela	31977065	Belarus	9468338
Uzbekistan	31910641	United Arab Emirates	9400145
Malaysia	31624264	Honduras	9265067
Angola	29784193	Tajikistan	8921343
Mozambique	29668834	Serbia	8790574
Nepal	29304998	Austria	8735453
Ghana	28833629	Switzerland	8476005
Yemen	28250420	Israel	8321570

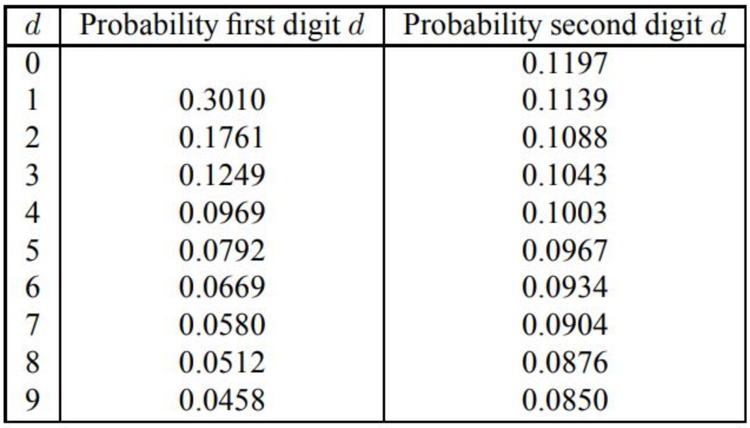
Table 5.2: Population of top 232 countries by UN Nations [64]

> the same inquiry respecting their quotients, and continue the process so as to find the limit towards which the probability approaches.

Newcomb thus concluded that all of the digits are not equally likely and that (in base 10) a number which starts with one will have a probability of occurrence of 30%, starting with two will have 17.6% and so on. He concluded that,

> The law of probability of the occurrence of numbers is such that all mantissa of their logarithms is equally probable.

Newcomb also speculated on the probabilities of the first and second digits as shown in the table 5.3.



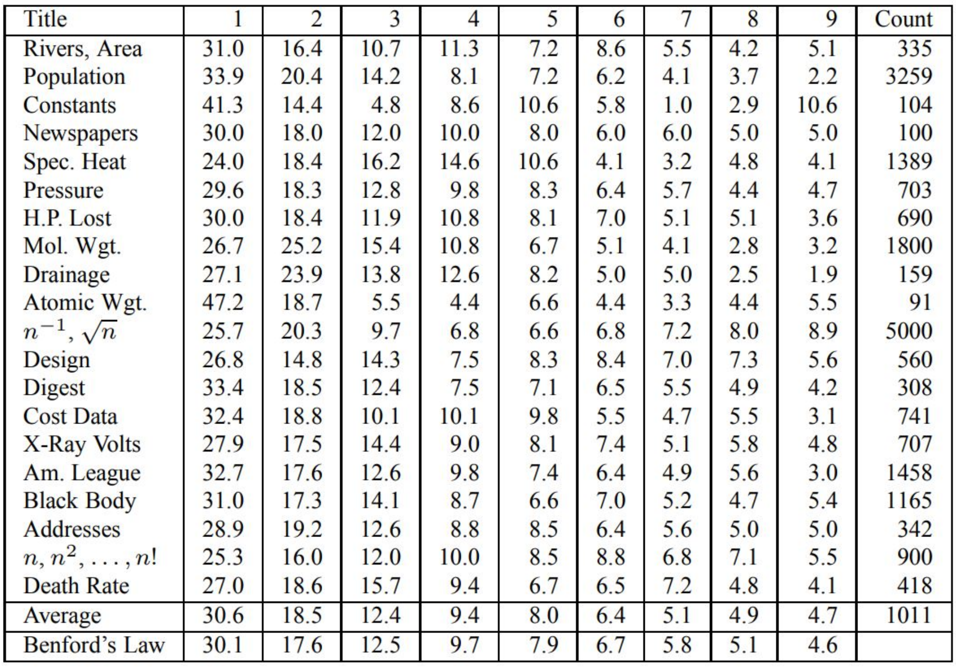
d	Probability first digit d	Probability second digit d
0		0.1197
1	0.3010	0.1139
2	0.1761	0.1088
3	0.1249	0.1043
4	0.0969	0.1003
5	0.0792	0.0967
6	0.0669	0.0934
7	0.0580	0.0904
8	0.0512	0.0876
9	0.0458	0.0850

Table 5.3: Newcomb's observations for the probabilites of getting a first digit or second digit [65]

Frank Benford, in the year 1938, studied and rediscovered Benford's law or the First Digit Law. He published his work titled *"The Law of Anomalous Numbers"* [62] in the Proceedings of the American Philosophical Society. He was an American physicist and engineer, also an expert in optics, with 109 publications and 20 patents in the same field. Benford collected data from 20 different fields such as addresses, newspapers, river data and studied the distribution of the first digit. The table 5.4 shows the frequency of leading digits.

Few observations can be studied from the last two rows of the table 5.4 showing the average percentage of occurrence for each digit and the result from the Benford's computation. One can observe that frequency of occurrence of 1's is seen to be 0.306 nearly equal to the logarithm of 2. For 2's the average is 0.185 which is close to the logarithm of $3/2$ or $\log 3 - \log 2$, which is the logarithmic interval. Similar observations can be seen for other digits too. The distribution follows the logarithmic relation given by the following equation,

$$F_a = \log\left(\frac{a+1}{a}\right) \tag{5.1}$$



Title	1	2	3	4	5	6	7	8	9	Count
Rivers, Area	31.0	16.4	10.7	11.3	7.2	8.6	5.5	4.2	5.1	335
Population	33.9	20.4	14.2	8.1	7.2	6.2	4.1	3.7	2.2	3259
Constants	41.3	14.4	4.8	8.6	10.6	5.8	1.0	2.9	10.6	104
Newspapers	30.0	18.0	12.0	10.0	8.0	6.0	6.0	5.0	5.0	100
Spec. Heat	24.0	18.4	16.2	14.6	10.6	4.1	3.2	4.8	4.1	1389
Pressure	29.6	18.3	12.8	9.8	8.3	6.4	5.7	4.4	4.7	703
H.P. Lost	30.0	18.4	11.9	10.8	8.1	7.0	5.1	5.1	3.6	690
Mol. Wgt.	26.7	25.2	15.4	10.8	6.7	5.1	4.1	2.8	3.2	1800
Drainage	27.1	23.9	13.8	12.6	8.2	5.0	5.0	2.5	1.9	159
Atomic Wgt.	47.2	18.7	5.5	4.4	6.6	4.4	3.3	4.4	5.5	91
n^{-1}, $\sqrt{n}$	25.7	20.3	9.7	6.8	6.6	6.8	7.2	8.0	8.9	5000
Design	26.8	14.8	14.3	7.5	8.3	8.4	7.0	7.3	5.6	560
Digest	33.4	18.5	12.4	7.5	7.1	6.5	5.5	4.9	4.2	308
Cost Data	32.4	18.8	10.1	10.1	9.8	5.5	4.7	5.5	3.1	741
X-Ray Volts	27.9	17.5	14.4	9.0	8.1	7.4	5.1	5.8	4.8	707
Am. League	32.7	17.6	12.6	9.8	7.4	6.4	4.9	5.6	3.0	1458
Black Body	31.0	17.3	14.1	8.7	6.6	7.0	5.2	4.7	5.4	1165
Addresses	28.9	19.2	12.6	8.8	8.5	6.4	5.6	5.0	5.0	342
$n, n^2, \ldots, n!$	25.3	16.0	12.0	10.0	8.5	8.8	6.8	7.1	5.5	900
Death Rate	27.0	18.6	15.7	9.4	6.7	6.5	7.2	4.8	4.1	418
Average	30.6	18.5	12.4	9.4	8.0	6.4	5.1	4.9	4.7	1011
Benford's Law	30.1	17.6	12.5	9.7	7.9	6.7	5.8	5.1	4.6	

Table 5.4: Distribution of first digits from Benford's dataset[62]

where F_a is frequency of the first digit a. One important thing to observe here is that the results were compiled from numbers having digits ranging between four to six. Benford also postulated the frequency of digits for the second, third place, etc., and showed that the law also holds for the reciprocals. The second digit can hold numbers from 0-9. Hence we have ten digits for the second place and so on. Considering the frequency F_b of a digit b at second place, which also depends on the digit a at the first place the logarithmic interval has to be now divided into ten parts. The logarithmic interval for number ab will be $\log(ab+1) - \log(ab)$, where $ab+1$ is the number greater than ab. Similarly, logarithmic interval for digit a at the first place is, $\log(a+1) - \log(a)$. The frequency of digit b at the second place followed by a is given by,

$$F_b = log\left(\frac{ab+1}{ab}\right) \Big/ \log\frac{a+1}{a}. \tag{5.2}$$

Lets take an example where we want to find the probability of 1 coming at the second place following 6 from Benford's law,

$$F_b = log\left(\frac{62}{61}\right) \Big/ \log\frac{7}{6}.$$

The equation can now be generalized to the digits in qth position as,

$$F_b = \frac{log\dfrac{abc\cdots p(q+1)}{abc\cdots pq}}{log\dfrac{abc\cdots o(p+1)}{abc\cdots p}} \tag{5.3}$$

Figure 5.2. shows the frequency of digits d at first and the second place.

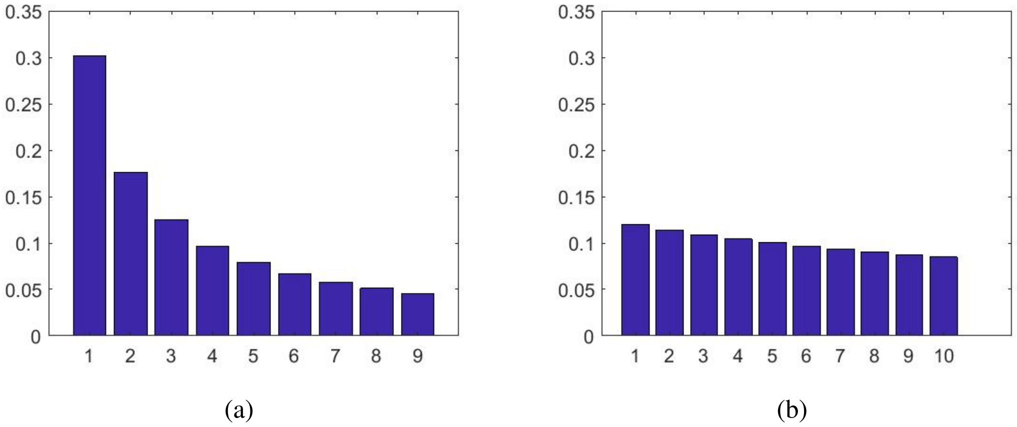


Figure 5.2: Distribution of a) First and b) Second digits from Benford's Law [62]

Benford also observed that this law was more accurate to 'anomalous' numbers, which do not follow any particular trend and are random such as street addresses or the number in newspapers whereas the first digits of molecular and atomic weight deviated from the observations.

5.2 JPEG Compression Framework

A digital image is a rectangular grid of pixels, where each pixel represents the grayness of an image for a monochrome image or the component value for a colored image such as RGB image. A 1000x1000 monochrome image with 8-bit resolution would require megabytes of space for storage which is not satisfying. In this kind of scenario, the theory of data compression comes into play. The idea is to create a compact representation of data by exploiting statistical redundancy in an image. JPEG (Joint Photographic Expert Group) is the most common lossy compression technique for still images, developed by Joint Photographic Expert Group in 1992.

JPEG is a lossy compression scheme which exploits the limitations of Human Visual System (HVS) which is more sensitive to change in brightness than color of a scene. Thus, even on losing some of the information in the image the quality of the image remains high. A trade-off can be made between the quality of the image and the size by changing the compression ratio or the quality factor varying between $1-100$, 100

having the highest quality or the least compression. JPEG compression scheme typically consists of 5 steps, a) RGB to YCbCr color space transformation b) Sub sampling of the Chroma colour space c) transformation from spatial domain to the frequency domain using DCT(Discrete Cosine Transform) d) Quantization of the resulting block DCT depending on the quality factor e) Coding the data using Huffman coding. Figure 5.3 shows the block diagram of JPEG compression chain. Since most of the literature related to Benford's Law in forensics is related to compression detection in images, we provide the detailed explanation of JPEG compression technique.

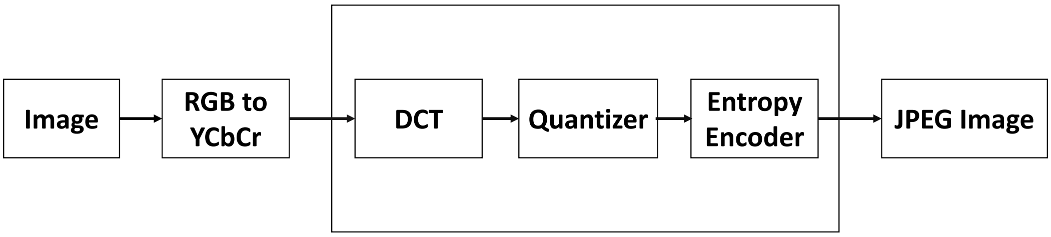


Figure 5.3: JPEG compressor block

5.2.1 RGB to YCbCr Space Transformation

The retina of human eye contains two photoreceptors, rods and cones, which are more susceptible to brightness than to color in an image. The RGB color space mixes the luma and chroma components of the light using color display. Therefore, an image is first transformed from RGB to YCbCr before compression, where Y is the luminance factor, and Cb/Cr is the chrominance factor. Further Cb/Cr channel is down sampled by a factor of 2 relative to the luminance channel for the same reasons discussed above.

$$\begin{aligned} Y &= 0.299R + 0587G + 0.114B \\ Cb &= -0.1687R - 0.3313G - 0.5B + 128 \\ Cr &= 0.5R - 0.418G - 0.0813B + 128 \end{aligned} \tag{5.4}$$

Consider you have a 1024x1024 RGB image. The first process will be to convert it into YCbCr color space. The next step will be to reduce the resolution of Cb/Cr channels by 2, i.e., 512x512 pixels. It implies that for each 2x2 area, we require only six different values, instead of 12, reducing the data by 50%.

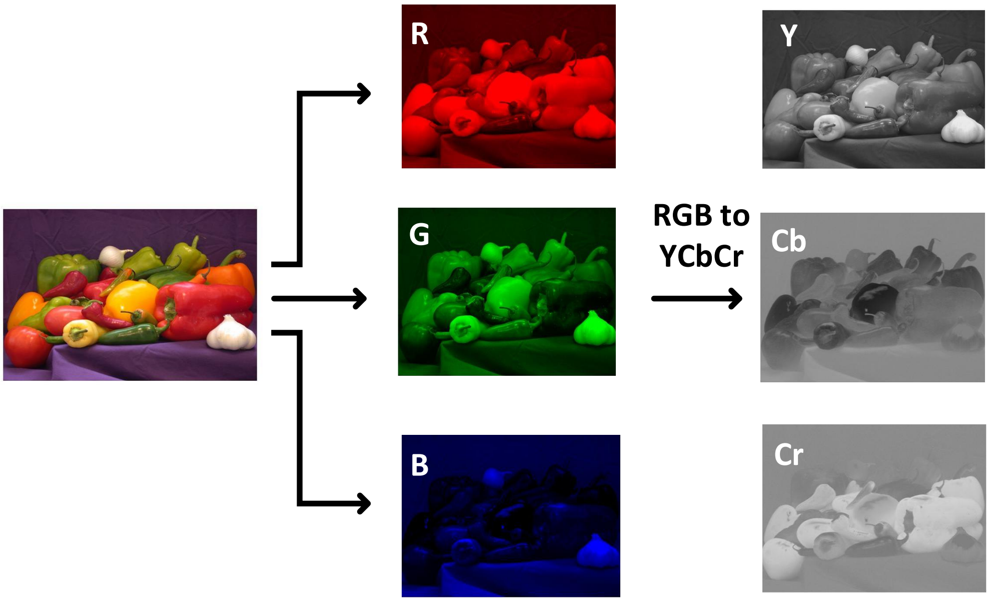


Figure 5.4: Color space transformation

5.2.2 Discrete Cosine Transform

Human eyes are more sensitive to low-frequency components in an image. Therefore, it is possible to compress the data to reduce the redundant information. To convert the spatial information to the frequency domain, Discrete Cosine Transform (DCT) is performed, requiring less number of coefficients than any other transform. The range in the image for an 8-bit image is shifted around center converting from unsigned [0,255] to signed integer [-128,128]. The image is then divided into 8x8 block, which has been selected empirically and DCT is performed. The DCT conversion is,

$$D(i,j) = \frac{1}{4}k(i)k(j)\sum_{x=0}^{7}\sum_{y=0}^{7} f[x,y]cos\left[\frac{(2x+1)i\pi}{16}\right]cos\left[\frac{(2y+1)j\pi}{16}\right] \tag{5.5}$$

where i, j are the spatial frequencies, $f(x, y)$ is the 8x8 data block and $k(i, j)$ is the normalization factor defined as,

$$k(u) = \begin{cases} \frac{1}{\sqrt{2}}, & if \;\; u = 0 \\ 1, & otherwise. \end{cases} \tag{5.6}$$

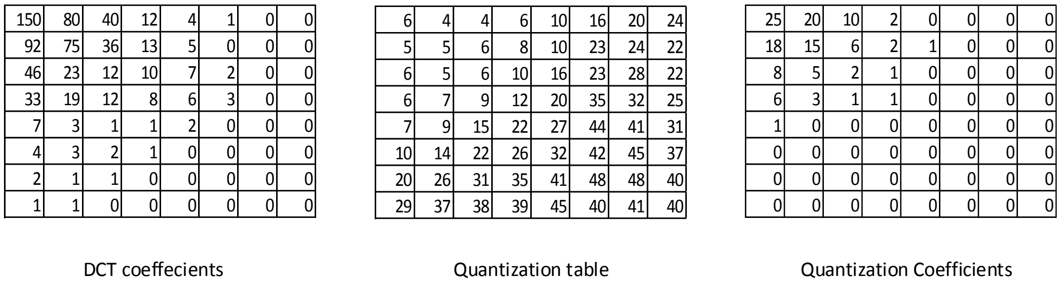
DCT separates the high and the low frequencies in the image, enabling us to save only the required frequencies depending on the quantization matrix.

5.2.3 Quantization

Next step after block DCT on image data, quantization is applied to remove the high frequency data. The quantization table is chosen depending on the amount of compression, and the chroma channel is heavily quantized compared to luma channel. The quantization is performed as follows,

$$P = round\left(\frac{D(i,j)}{Q(i,j)}\right) \tag{5.7}$$

where D is the 8x8 DCT coefficients, and Q is the quantization matrix. Figure 5.5 shows an 8x8 image block with unquantified DCT coefficients and the resulting quantized coefficients on division by the Quantization matrix.



150	80	40	12	4	1	0	0
92	75	36	13	5	0	0	0
46	23	12	10	7	2	0	0
33	19	12	8	6	3	0	0
7	3	1	1	2	0	0	0
4	3	2	1	0	0	0	0
2	1	1	0	0	0	0	0
1	1	0	0	0	0	0	0

DCT coeffecients

6	4	4	6	10	16	20	24
5	5	6	8	10	23	24	22
6	5	6	10	16	23	28	22
6	7	9	12	20	35	32	25
7	9	15	22	27	44	41	31
10	14	22	26	32	42	45	37
20	26	31	35	41	48	48	40
29	37	38	39	45	40	41	40

Quantization table

25	20	10	2	0	0	0	0
18	15	6	2	1	0	0	0
8	5	2	1	0	0	0	0
6	3	1	1	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Quantization Coefficients

Figure 5.5: Quantized DCT coefficients of an 8x8 image block

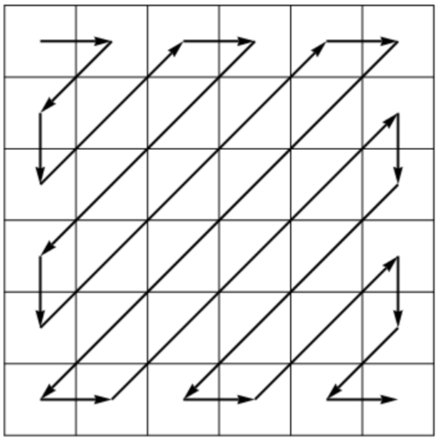


Figure 5.6: Zig Zag Order Scan

5.2.4 Huffman Coding

After quantization, zig-zag order sequence (shown in Figure 5.6) orders all the AC and DC (0,0) frequencies, from lowest to highest frequency for run-length encoding.

After this step Huffman coding is performed which assigns shorter codes for frequently appearing numbers than rare ones, to reduce the storage and transmission memory.

5.3 Benford's Law in Image Forensics

With the increase in the use of digital technologies like smart phones and cameras, the primary concern for forensics experts remains in the authenticity of the digital content in the form of images, audios or videos. Often instances arise where they can be strong evidence in the court of law. Many software is available nowadays which can manipulate images without leaving any visual traces and one need not be an expert in it to do, so. A lot many manipulations such as copy and move forgery, tampering, median filtering, double JPEG attack are possible leaving no visual artifacts but disturbing the statistical properties of the image. For example, in median filtering attack, the pixel difference alters in case of the manipulated image.

The first application to Benford's law in images was provided in [66]. Authors proved experimentally that the gradient of an image follows Benford's law. They also suggested a method for entropy coding depending on the logarithmic occurrence of digits, similar in the case as in Huffman coding where the frequently occurring digits are coded with lower bits. Similarly, in [67], the law was applied specifically to medical images such as of MRI and CT scan and also developed a denoising algorithm generating good results. Application of the law to detect fingerprints and face images has been proposed in [68]. Image splicing detection using Benford's law has been proposed in [69, 70]. Gonzalez [71] in his work claimed that images in pixel domain do not follow Benford's law since for an 8- bit image the numbers are limited in range 0-255 (since the pixel values are not anomalous) but did follow when the image is transformed from spatial domain to DCT (Discrete Cosine Transform) domain. The first significant digit can be calculated as,

$$s = \left\lfloor \frac{|X|}{10^{\lfloor \log_{10} |X| \rfloor}} \right\rfloor \tag{5.8}$$

where s is the first significant digit and X is any positive number. Transformation into DCT domain is an important step when an image is JPEG compressed. Some of the properties of Benford's law were also demonstrated in their work,

Property 1: *A random variable X follows Benford's law if the random variable $Y = log_{10}X$ mod 1 is uniform in [0, 1).* A random variable is strong Benford if it follows property 1.

Property 2 (Scale invariance): *Suppose that X follows Benford's law; then the random variable Z = αX will follow Benford's law for an arbitrary array α if only if X is strong Benford's.*

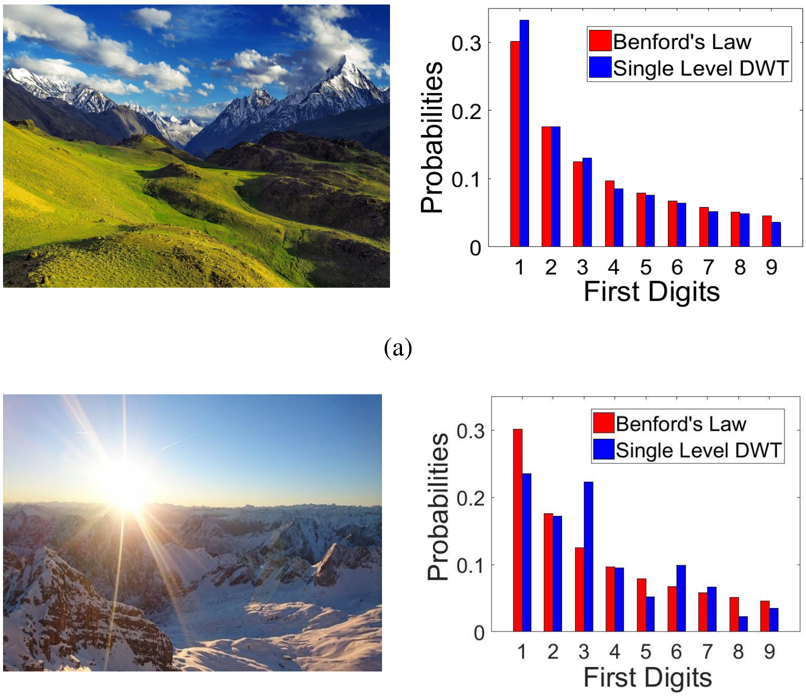
Property 3 (Product of independent random variables): *Let X be strong Benford's, and let Y be another random variable independent of X. Then, the random variable $Z = X * Y$ is strong Benford's.*

Few of the applications to image forensics from Benford's were demonstrated in work. Determining if an image saved in bitmap format has been compressed previously is not a naive task. Authors proposed a scheme in which given a test image which was a bitmap, a JPEG compression was performed with a quality factor of 100. If the image is not previously compressed, it will obey the Benford's law not otherwise, due to the JPEG artifacts from the previous compression. Using similar observations, they were also able to estimate the quality factor for the bitmap image, if previously compressed. The idea was to compress the image with different quality factors and estimate the fitting error from Benford's law. The one with least fitting error will then be the Q-factor or the quality factor for the test image.

An image if double compressed with different quantization matrix becomes a matter of concern for forensics experts because it could be the result of copy-move forgery or tampering of the image. The quantized coefficients for a double compressed JPEG image vary from a single compressed image if the quantization matrix for the second compression is not an integer multiple of the first quantization matrix. Thus on every compression, the statistics of quantized coefficients deviates from the Benford's law. Using these statistics [72], the authors generated features by extending the idea of applying Benford's Law to the quantized DCT using 20 subbands generating 180-D feature vectors and named as the Mode based first digit features. The 180 dimension features are the probabilities of the occurrence of first digits of the quantized DCT coefficients. Fisher Linear Discriminant (FLD) is used for supervised classification trained by 1138 images from UCID (Uncompressed Color Image Database), and the rest 200 is used for test purpose. Wang et al. [73] proposed a generalized Benford's Law for Gaussian and Laplacian distribution. The generalization of the law is given as,

$$B_g(h) = a_1 . \log 10 \left(a_3 + \frac{1}{h^{a_2}} \right) \tag{5.9}$$

where a_1, a_2, a_3 are the model parameters whose values depend on the distribution. Application of Benford's to watermarked images was proposed in [74]. Discrete Wavelet Transform (DWT) is used in JPEG2000 lossless compression method, which is the non-commercial version to JPEG to reduce the blocking artifacts in JPEG. Qadir et al. [75] analyzed the law on DWT coefficients and concluded that DWT coefficients more accurately followed the Benford's as compared to the DCTs, and also gave the analysis of the DWT coefficients at different compression rates. The analysis showed they followed Benford's with little deviation. An image if taken when the camera is pointing towards the light direction results in glare based problems. The retrieval of glared images by using Benford's law was proposed in [76]. The idea is to transform image to DWT domain and extract all the sub band and fit the Benford's on the significant digits of all coefficients in each band. The images whose coefficients deviated from the law were the ones which had glare (as shown in the Figure 5.7).



(a)

(b)

Figure 5.7: (a) Image 'Mountains 1' and the 1'st digit probability and Benford's law distribution (b) Image 'Mountains 2' and the 1'st digit probability and Benford's law distribution [76]

With the advent and easy availability of image editing software, distinguishing between photorealistic and camera generated images is a challenging task. Xu et al. [77] proposed a method to differentiate between photorealistic and camera generated (CG) images. The features are generated as follows (Figure 5.8). The images are separated into RGB channels, and the 8x8 block DCT coefficients are calculated over each channel, and the frequency of the first digit is computed along all AC frequencies, which sums up to 27-D feature for one image. The first digit frequency is also computed over the gradient of the image which totals up to 54-D feature for each image (for each channel the Block DCT and Gradient Magnitude will generate 18- D feature).

Multiple JPEG compression occurs when the image has undergone several compression and decompression steps, which can be due to tampering or copy-move forgery, modifying the statistics of the coefficients. With each compression, the statistics of the DCT coefficients slowly deviate from the law. The Figure 5.9 shows the probability mass function (pmf) of the First digits of the quantized DCT coefficients for different

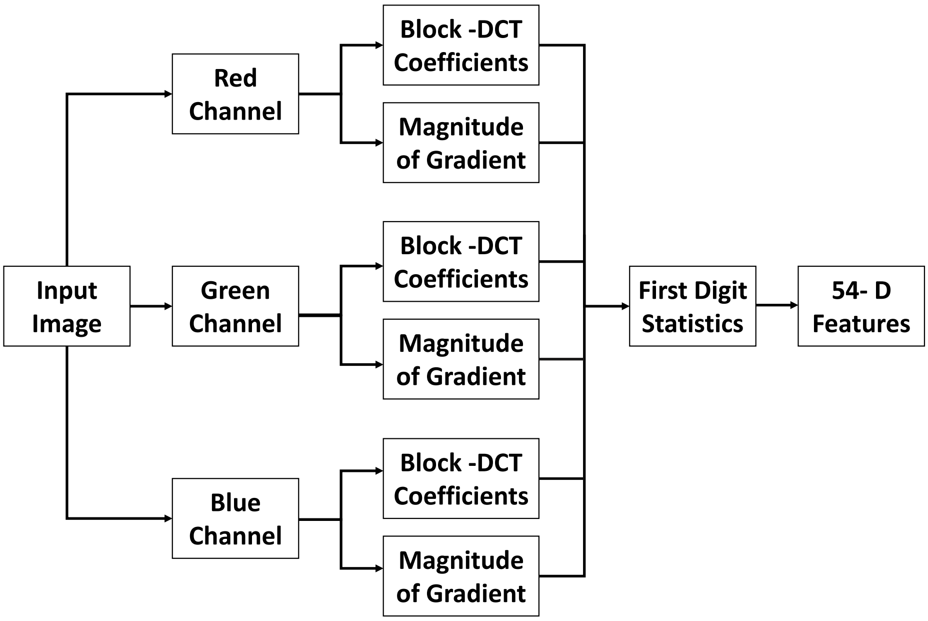


Figure 5.8: Feature generation process

compression ratios. Milani et al. [78] used first digit law to detect the number of compressions the image has undergone. The strategy was to classify the features based on a set of binary classifiers. The idea to generate features has been inherited from [72], but reducing the number of subbands in the features. The features contained subbands 2,5 and 6 resulting in a feature vector of 27-D.

Pasquini et al. [79] used Benford-Fourier (BF) coefficients to detect uncompressed JPEG images from the compressed ones. The following section explains the statistics of Benford-Fourier (BF) Coefficients.

Consider a random variable X, which models all the 8 x 8- block DCT coefficients with symmetric pdf $f_x(X)$ with respect to 0. Then a random variable Z determined by the probability density function,

$$f_z(z) = \begin{cases} 2f_x(z), & z > 0 \\ 0, & z \leq 0. \end{cases} \tag{5.10}$$

For a continuous random variable, Z, the pdf of $\tilde{Z}$ ($\tilde{Z} = \log_{10} Z$) can be expressed as:

$$f_{\tilde{Z}}(\tilde{z}) = 1 + 2\sum_{n=1}^{+\infty} |a_n| cos(2\pi n \tilde{z} + \phi_n), \quad \tilde{z} \in [0, 1) \tag{5.11}$$

As discussed in various literatures, the distribution of X can be modeled by a Gen-

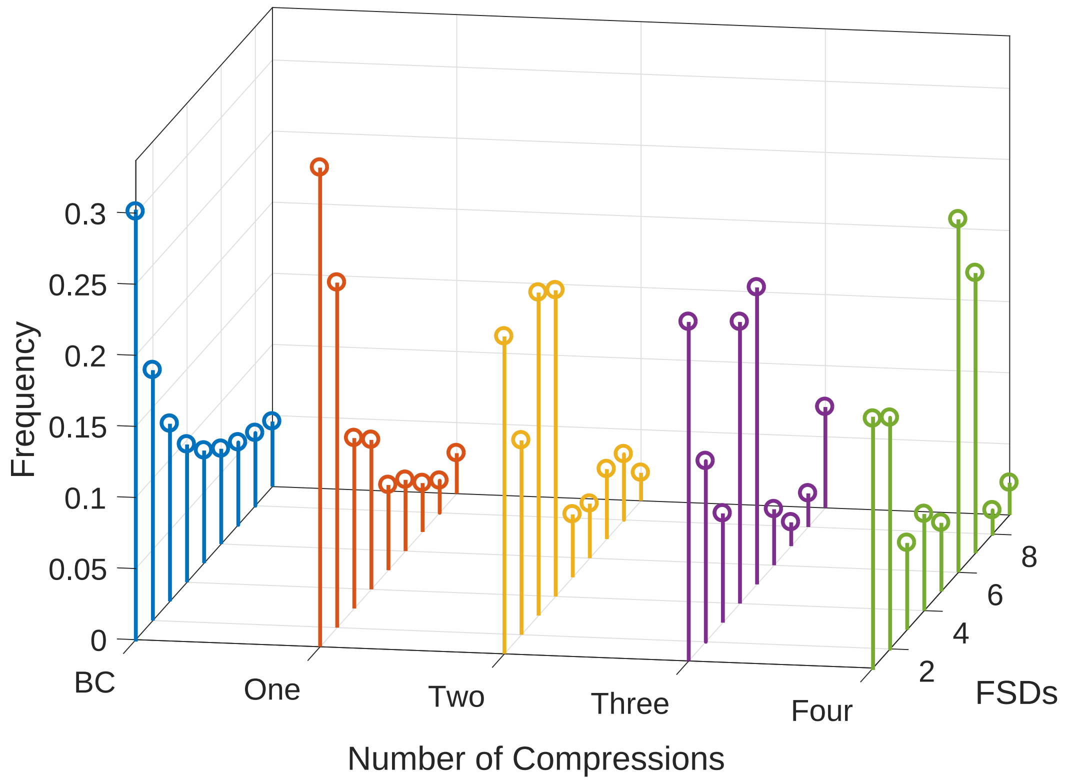


Figure 5.9: Pmfs of the first digit (FDs) with respect to the Benford's coefficients (BC) at different compressions

eralized Gaussian (GG) pdf,

$$f_x(x) = \frac{\beta c}{2\Gamma(1/c)} e^{-|\beta x|^c}, \beta = \frac{1}{\sigma}\sqrt{\frac{\Gamma(3/c)}{\Gamma(1/c)}} \tag{5.12}$$

where σ is the standard deviation and c is the shaping factor. Thus a_n and its magnitude can be written as (derived in [71])

$$a_n = \frac{2A}{\beta c} e^{j\frac{2\pi n \log \beta}{\log 10}} \Gamma\left(\frac{-j2\pi n + \log 10}{c \log 10}\right) \tag{5.13}$$

$$|a_n| = \sqrt{\prod_{k=0}^{\infty}\left[1 + \frac{(2\pi n)^2}{\log^2} 10(ck+1)^2\right]} \tag{5.14}$$

Thus it can be seen that the Benford-Fourier coefficients are dependent on c and n and can be used to differentiate between compressed and uncompressed images.

The Benford-Fourier coefficients are given by,

$$a_n = \int_{-\infty}^{+\infty} f_z(z) e^{-j2\pi n \log_{10} z} dz \tag{5.15}$$

or it can be written as the expected value of a random variable $d_n(z)$ where,

$$d_n(Z) = e^{-j2\pi n \log_{10} Z} \tag{5.16}$$

An estimation of such coefficients from M samples as,

$$\hat{a_n} = \frac{\sum_{m=1}^{M} e^{-j2\pi n \log_{10} Z_m}}{M} \quad m = 1, ...M \tag{5.17}$$

where Z_m are the random variables represents the DCT coefficients at the corresponding frequency in the block m, which are Independently and Identically distributed. On applying the Central Limit Theorem to real and imaginary parts of a_n we have,

$$\hat{a_n} = a_n + W_0 \tag{5.18}$$

where a_n is the sample mean and W_0 is the zero-mean complex normal random variable. For W_0 to be circularly symmetric $E\{W_0^2\} = 0$. That is,

$$E\{W_0^2\} = E\{(\hat{a_n} - a_n)^2\} = \frac{1}{M}(a_{2n} - a_n^2) \tag{5.19}$$

It is already known that distribution of $R := |\hat{a_n}|$ is given by the Rice distribution, with parameters $|\hat{a_n}|$ and σ, where σ is the standard deviation. Thus,

$$\sigma^2 = \frac{\{|\hat{a_n}|^2\} - |a_n|^2}{2} = \frac{1}{2M}(1 - |a_n|^2) \tag{5.20}$$

Since a_n is very small when n $\geq$ 3, which leads to issue in solution for the Rice distribution. Thus, we apply a special case of Rice distribution, the Rayleigh distribution, where $|a_n| \simeq 0$ and $\sigma = 1/\sqrt{2M}$:

$$f_R(r) = 2Mre^{-Mr^2} \tag{5.21}$$

A binary hypothesis test is then decided for uncompressed images, which gives a threshold given the upper bound to the probability of false alarm. Inspired by [79], authors in [80] proposed a method to detect multiple JPEG compressions and also identify the quality matrix with which it is compressed. Figure 5.10 shows the detection algorithm for the proposed work.

Anti-Forensics or counter-forensics aims at hiding the traces of manipulation in images such as the effect of JPEG compression. These techniques have been used in CFA forensics, median filtering detection, histogram-based methods, etc. In [81], the authors have proposed an anti-forensics technique that conceals the effect of JPEG compression, thus reproducing an uncompressed image from a single JPEG compressed image. The

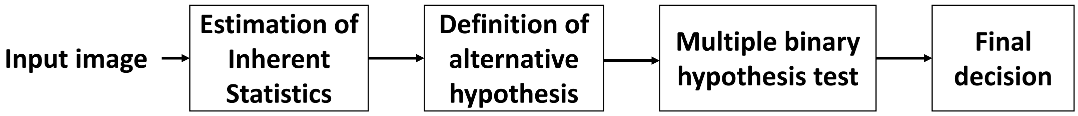

Input image
Estimation of Inherent Statistics
Definition of alternative hypothesis
Multiple binary hypothesis test
Final decision

Figure 5.10: Detection algorithm for multiple JPEG compression [80]

idea is to redistribute the First significant digits (FSDs) of the JPEG compressed image to recover the original distribution of the DCT coefficients. The statistical DCT coefficients of the uncompressed image closely follow the Benford's law as compared to the JPEG compressed image (Figure 5.11). Considering, for any sub-band (i,j) whose quantization factor is $Q(i,j)$, and the histogram contains peaks which are multiples of the quantization factor $Q(i,j)$. They then define, vectors $p_k, k \in \mathbf{Z}$ in each subband containing multiples of the QF $Q(i,j).k$. The next step is to redistribute the quantized coefficients in p_k such that to obtain a Laplacian histogram among the quantization interval,

$$I_Q = [kQ - Q/2, kQ + Q/2] \tag{5.22}$$

The aim is to uniformly distribute the values of vector p_k in the range $[L_1, L_2]$, where

$$L_1 = \log_{10}(|kQ - Q/2|) \tag{5.23}$$

$$L_2 = \log_{10}(|kQ + Q/2|) \tag{5.24}$$

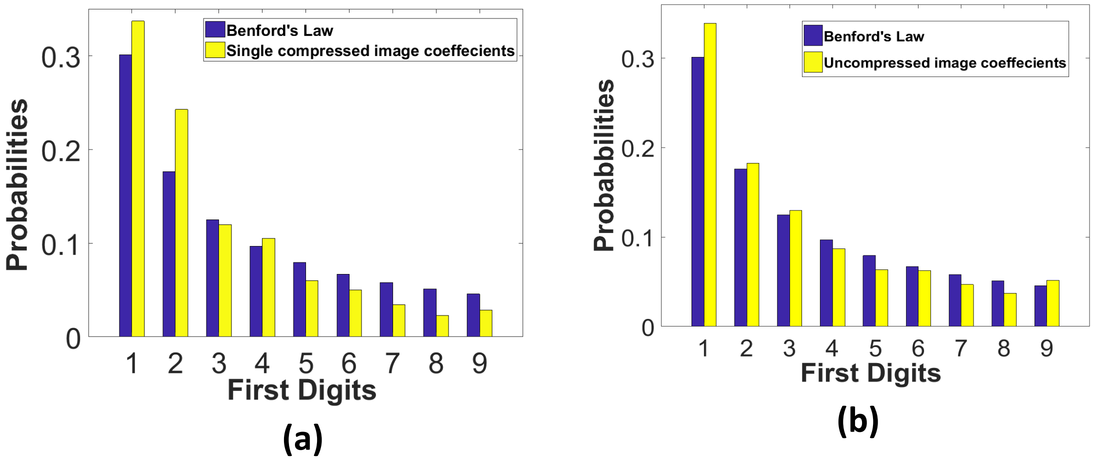


Figure 5.11: Deviation of Benford's Law for a) JPEG compressed b) and Uncompressed image

Another anti-forensics technique to hide the effects of double compression such that the compression detectors based on Benford's law are unable to detect the effect is proposed in [82]. Most of the approaches detecting frauds in double or multiple JPEG compressions relies on measuring some distance from empirical pmf $\hat{p}(m)$ of Benford's law to the actual pmf $p(m)$ of the first significant digits of the DCT coefficients. One of the techniques is to measure the difference between the pmf using Kullback-Leibler (KL) divergence D_{KL} as,

$$D_{KL}(p||\hat{p}) = \sum_{m=1}^{9} p(m) \frac{p(m)}{\hat{p}(m)} \tag{5.25}$$

An image is considered double compressed when the divergence D_{KL} is less than a given threshold T_{KL} (which is decided by a forensics expert). The proposed algorithm alters the statistics of the coefficients such that the KL divergence of Benford's pmf to a double compressed image D^2_{KL} is nearly equal to the divergence of a single compressed image D^1_{KL}.

5.4 Benford's Law in Audio and Printer Forensics

Audio Forensics, similar to image and video forensics is related to the analysis and the authentication of audio and music recordings. Just like JPEG in images, Mp3 compression is the widely used form of audio compression. Similar to image editing software, audio editing software is readily available in the market, which can convert a low bit rate mp3 audio to a high bit rate audio. A brief introduction to MP3 compression is given in the next paragraph.

The first step in MP3 compression is to separate the input signal (PCM) into multiple components, using a filter bank. Here, the size of the filter bank is 32. Further, MDCT (Modified Discrete Cosine Transform) is applied on each subband transforming it into length of (18 or 6 subband samples), which generates a total of 576 or 192 subband samples. The psychoacoustic model uses perpetual coding (based on how humans perceive sound) and eliminates information inaudible to our ears, precisely detecting conditions under which audio signals mask each other. Next step is the quantization of the spectral values decided by the previous step. Finally, the quantized spectral values are encoded using Huffman encoding resulting in the MP3 bitstream. The decoding process is as follows. The first step is to perform Huffman decoding on the MP3 bitstream, following the inverse quantization and IMDCT to get the coefficients in the subband domain. The synthesis filter generates the PCM waveform in the final step. The block diagram of MP3 compression and decompression is shown in the Figure 5.12.

Authors in [83], used Support Vector Machines (SVMs) to differentiate between double and single compressed audio using the features as the first digit distribution of the MDCT coefficients. Particularly, the feature set consists of 9 values of probability

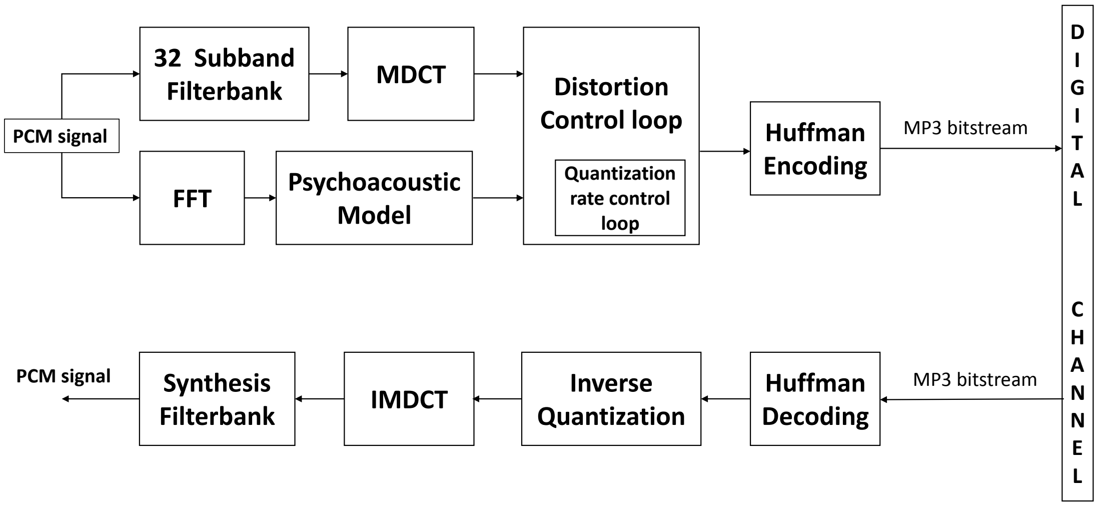


Figure 5.12: MP3 encoder and decoder block diagram

distribution corresponding to each digit and is fed to the SVM for each audio. This method proposed is the global method. Another method proposed is the Band distribution method in which the 576 coefficients are divided into 32 bands, and then the MDCT statistics on the first digits are calculated. The second method of Band distribution outperforms the global distribution method.

A method to identify printer's make and model from the printed and scanned images was proposed in [84]. Traditionally, noise feature extraction, i.e., extracting randomness from the noise pattern was used for printer classification.

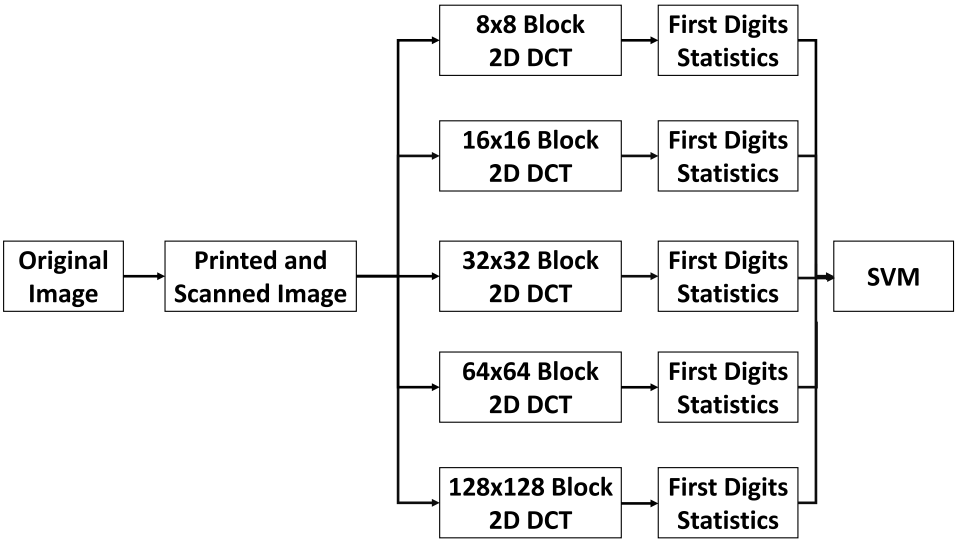


Figure 5.13: Multi-Size Block DCT Coefficients Flowchart [84]

Here, the authors proposed to use the first digits DCT coefficients at different block sizes to detect the weak noise pattern.

Finally, the resulting feature is a 9-D vector averaged over each first digit value and classified using Support Vector Machines (SVM). Figure 5.13 shows the block diagram of the proposed algorithm.

Chapter 6

Conclusion

The planet Earth, since its formation, has been pummeled by asteroids and meteorites which over time have been eroded due to rain and wind, destroying the traces of its past and origin. But Moon, the Earth's natural satellite still holds the clues to our history and origin. The surface of Moon is abundantly filled with giant impact craters, rocks, rivers, VRC's where craters constitute 80% of the surface area. Thus, studying these craters can answer the most vexing questions of all time, ***"How did the Earth form?"*** To settle this debate, the first Apollo mission brought the rocks- precisely 382kg from the cold and dry surface of the Moon, to unveil its formation and its relation to Earth. Since then, missions like SELENE, Chang-E1, Chandrayaan-1 have been send to the Moon to gather more information about its surface and formation. Many researchers have worked on the dataset from different missions on the Moon to locate craters and further analyze it to study the geological formation of the Moon. Despite numerous publications in this domain, most of the algorithms are data specific or requires a lot of manual intervention.

This dissertation presents an automated method to detect impact craters on the surface of Moon through the ortho images, DEM and slope images derived from DEM. Since no dataset is build keeping in mind all the three images, we have built our dataset consisting of images with widely varying densities and area, challenging our CDA. Most of the CDA's are data dependent and fails to work for different terrains. Thus, we have also tested our method on Chandrayaan-1 to observe the effect of resolution and on Martian data, to check if our algorithm is data dependent or not. Our algorithm performs very well when trained and tested with LRO dataset and has detected $\sim$1000 new craters per test image.

Manual observation on Chandrayaan-1 dataset has revealed that the higher resolution data of Chandrayaan-1 shows very different patterns for slope than LRO. It should also be noted that median filtering has been applied on Chandrayaan-1 DEM data to remove spurious peaks and dips. Also, we have noticed that due to very high resolution, the area covered by a single Chandrayaan-1 image is relatively very low then the LRO image resulting in incomplete craters near the edges. Such part of craters is misclassified by CNN as craters which results in low mAP because of localization inaccuracies. We believe that fine-tuning the CNN model using transfer learning on Chandrayaan -1 data may improve the final mAP. The model used for generating results on Martian dataset achieves $mAP@0.5$ of 84.78% and $mAP@[0.5, 0.9]$ of 59.18%, without parameters tuning. We hope to obtain better mAP with transfer learning on Martian dataset.

We have used LRO (Lunar Reconnaissance Orbiter) dataset with a resolution of 100 m for ortho images and 118 m for DEM in our Crater Detection Algorithm (CDA). Working with such low resolution limits our CDA to detect very small craters. Thus, our next step would be to work on Chandrayaan-1 dataset having a high resolution of 10 m for ortho and 5 m for DEM data. Currently, our model has been trained using ortho images in one lighting direction, i.e., shadow to bright region (shown in green color in Figure 4.2). For fine-tuning the model, it needs to be trained for different lighting conditions. Since DEM is independent of the sun angle conditions and in similar sense slope images, it would be interesting to observe the effect of training on only DEM and slope images.

As of now, we have used a binary model, differentiating between the crater and the non-crater class. Craters vary greatly in their shapes and size, depending upon the type of impact and the lunar surface morphology. Further, the lunar surface also harbingers other landforms such as volcanic rootless cones (VRCs) and rivers. Thus, we wish to build a multi-class dataset differentiating craters based on shape (such as round and floor craters), complexity (such as simple and complex craters), overlapped craters, VRC's. Further, crater rim is not always a perfect circle. Getting an estimate of center and radius from our CDA can help us identify the true shape of craters, on developing a rim identification algorithm. Any object detection algorithm greatly depends on the quality of region proposals used. We have used the Felzenswalb & Huttenlocher [54] method for finding regional proposals, giving a recall on an image of $\sim 75\%$. This recall means on an average the number of proposals matched with the ground truth having an IOU of at least 0.6. Using an algorithm such as Selective search and Edge boxes can probably increase this recall to improve the accuracy of our algorithm.

Bibliography

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[2] "Crater dataset," 2012. [Online]. Available: http://kdl.cs.umb.edu/w/datasets/craters/

[3] H. Melosh, *Planetary Surface Processes*, ser. Cambridge Planetary Science. Cambridge University Press, 2011. [Online]. Available: https://books.google.co.in/books?id=3bQD1DJgliIC

[4] R. B. Baldwin, "Lunar crater counts," *The Astronomical Journal*, vol. 69, p. 377, 1964.

[5] S. W. Kieffer and C. H. Simonds, "The role of volatiles and lithology in the impact cratering process," *Reviews of Geophysics*, vol. 18, no. 1, pp. 143–181, 1980.

[6] M. Ravine and R. Grieve, "An analysis of morphologic variation in simple lunar craters," *Journal of Geophysical Research: Solid Earth*, vol. 91, no. B13, 1986.

[7] A. Basilevskii, "On the evolution rate of small lunar craters," in *Lunar and Planetary Science Conference Proceedings*, vol. 7, 1976, pp. 1005–1020.

[8] R. J. Pike, "Crater dimensions from apollo data and supplemental sources," *The Moon*, vol. 15, no. 3-4, pp. 463–477, 1976.

[9] C. C. Allen, "Central peaks in lunar craters," *The moon*, vol. 12, no. 4, pp. 463–474, 1975.

[10] G. Salamunićcar and S. Lončarić, "Gt-57633 catalogue of martian impact craters developed for evaluation of crater detection algorithms," *Planetary and Space Science*, vol. 56, no. 15, pp. 1992–2008, 2008.

[11] N. G. Barlow, "Crater size-frequency distributions and a revised martian relative chronology," *Icarus*, vol. 75, no. 2, pp. 285–305, 1988.

[12] E. Kozlova, G. Michael, J. Rodionova, and V. Shevchenko, "Compilation and preliminary analysis of a catalogue of craters of mercury," in *Lunar and Planetary Science Conference*, vol. 32, 2001.

[13] L. Andersson and E. A. Whitaker, "Nasa catalogue of lunar nomenclature," 1982.

[14] R. Greeley and D. E. Gault, "Precision size-frequency distributions of craters for 12 selected areas of the lunar surface," *The Moon*, vol. 2, no. 1, pp. 10–77, 1970.

[15] M. Kirchoff, K. Sherman, and C. Chapman, "Examining lunar impactor population and evolution: additional results from crater distributions on diverse terrains," in *EPSC-DPS Joint Meeting 2011*, 2011, p. 1587.

[16] G. Salamunićcar and S. Lončarić, "Open framework for objective evaluation of crater detection algorithms with first test-field subsystem based on mola data," *Advances in Space Research*, vol. 42, no. 1, pp. 6–19, 2008.

[17] N. Suetake, E. Uchino, and K. Hirata, "Generalized fuzzy hough transform for detecting arbitrary shapes in a vague and noisy image," *Soft Computing*, vol. 10, no. 12, pp. 1161–1168, Oct 2006. [Online]. Available: https://doi.org/10.1007/s00500-005-0038-2

[18] R. Honda, Y. Iijima, and O. Konishi, "Mining of topographic feature from heterogeneous imagery and its application to lunar craters," in *Progress in Discovery Science*. Springer, 2002, pp. 395–407.

[19] M. Magee, C. Chapman, S. Dellenback, B. Enke, W. Merline, and M. Rigney, "Automated identification of martian craters using image processing," in *Lunar and Planetary Science Conference*, vol. 34, 2003.

[20] Y. Sawabe, T. Matsunaga, and S. Rokugawa, "Automatic crater detection algorithm for the lunar surface using multiple approaches," *Journal of the Remote Sensing Society of Japan*, vol. 25, no. 2, pp. 157–168, 2005, cited By :8. [Online]. Available: www.scopus.com

[21] G. Salamuniccar and S. Loncaric, "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*, vol. 48, no. 5, pp. 2317–2329, 2010.

[22] Y. Sawabe, T. Matsunaga, and S. Rokugawa, "Automated detection and classification of lunar craters using multiple approaches," *Advances in Space Research*, vol. 37, no. 1, pp. 21–27, 2006.

[23] L. Luo, X.-y. Wang, W. Ji, and C. Li, "Automated detection of lunar craters based on chang'e-1 ccd data," in *Image and Signal Processing (CISP), 2011 4th International Congress on*, vol. 2. IEEE, 2011, pp. 883–887.

[24] Z. Kang, Z. Luo, T. Hu, and P. Gamba, "Automatic extraction and identification of lunar impact craters based on optical data and dems acquired by the chang'e satellites," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 10, pp. 4751–4761, 2015.

[25] L. Bandeira, J. Saraiva, and P. Pina, "Impact crater recognition on mars based on a probability volume created by template matching," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 12, pp. 4008–4015, 2007.

[26] T. Barata, E. I. Alves, J. Saraiva, and P. Pina, "Automatic recognition of impact craters on the surface of mars," in *International Conference Image Analysis and Recognition*. Springer, 2004, pp. 489–496.

[27] Z. Yue, J. Liu, and G. Wu, “Automated detection of lunar craters based on object-oriented approach,” *Chinese Science Bulletin*, vol. 53, no. 23, pp. 3699–3704, 2008.

[28] J. R. Kim, J.-P. Muller, S. van Gasselt, J. G. Morley, and G. Neukum, “Automated crater detection, a new tool for mars cartography and chronology,” *Photogrammetric Engineering & Remote Sensing*, vol. 71, no. 10, pp. 1205–1217, 2005.

[29] R. Martins, P. Pina, J. S. Marques, and M. Silveira, “Crater detection by a boosting approach,” *IEEE Geoscience and Remote Sensing Letters*, vol. 6, no. 1, pp. 127–131, 2009.

[30] P. Viola and M. J. Jones, “Robust real-time face detection,” *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.

[31] E. R. Urbach and T. F. Stepinski, “Automatic detection of sub-km craters in high resolution planetary images,” *Planetary and Space Science*, vol. 57, no. 7, pp. 880–887, 2009.

[32] W. Ding, T. F. Stepinski, Y. Mu, L. Bandeira, R. Ricardo, Y. Wu, Z. Lu, T. Cao, and X. Wu, “Subkilometer crater discovery with boosting and transfer learning,” *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 4, p. 39, 2011.

[33] W. Ding, T. F. Stepinski, L. Bandeira, R. Vilalta, Y. Wu, Z. Lu, and T. Cao, “Automatic detection of craters in planetary images: An embedded framework using feature selection and boosting,” in *Proceedings of the 19th ACM international conference on Information and knowledge management*. ACM, 2010, pp. 749–758.

[34] L. Bandeira, W. Ding, and T. F. Stepinski, “Detection of sub-kilometer craters in high resolution planetary images using shape and texture features,” *Advances in Space Research*, vol. 49, no. 1, pp. 64–74, 2012.

[35] J. Yin, H. Li, and X. Jia, “Crater detection based on gist features,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 8, no. 1, pp. 23–29, 2015.

[36] S. Vijayan, K. Vani, and S. Sanjeevi, “Crater detection, classification and contextual information extraction in lunar images using a novel algorithm,” *Icarus*, vol. 226, no. 1, pp. 798–815, 2013.

[37] B. D. Bue and T. F. Stepinski, “Machine detection of martian impact craters from digital topography data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 45, no. 1, pp. 265–274, 2007.

[38] T. F. Stepinski, M. P. Mendenhall, and B. D. Bue, “Machine cataloging of impact craters on mars,” *Icarus*, vol. 203, no. 1, pp. 77–87, 2009.

[39] W. Zuo, Z. Zhang, C. Li, R. Wang, L. Yu, and L. Geng, “Contour-based automatic crater recognition using digital elevation models from chang’e missions,” *Computers & Geosciences*, vol. 97, pp. 79–88, 2016.

[40] S. Yamamoto, T. Matsunaga, R. Nakamura, Y. Sekine, N. Hirata, and Y. Yamaguchi, "An automated method for crater counting using rotational pixel swapping method," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 8, pp. 4384–4397, 2017.

[41] ——, "Rotational pixel swapping method for detection of circular features in binary images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 2, pp. 710–723, 2015.

[42] L. Palafox, A. Alvarez, and C. Hamilton, "Automated detection of impact craters and volcanic rootless cones in mars satellite imagery using convolutional neural networks and support vector machines," in *46th Lunar and Planetary Science Conference*, 2015, pp. 1–2.

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

[44] U. B. KDLab, "Crater dataset," 2013. [Online]. Available: http://kdl.cs.umb.edu/w/datasets/craters/

[45] L. Bandeira, W. Ding, and T. Stepinski, "Automatic detection of sub-km craters using shape and texture information," in *Lunar and Planetary Science Conference*, vol. 41, 2010, p. 1144.

[46] W. Li, B. Zhou, C.-Y. Hsu, Y. Li, and F. Ren, "Recognizing terrain features on terrestrial surface using a deep learning model: an example with crater detection," in *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*. ACM, 2017, pp. 33–36.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[48] https://cs.stanford.edu/people/karpathy/neuraltalk2/demo.html.

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

[50] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *The Journal of physiology*, vol. 148, no. 3, pp. 574–591, 1959.

[51] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*. Springer, 1982, pp. 267–285.

[52] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[53] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[54] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.

[55] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[56] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[57] S. Arya, D. M. Mount, N. S. Netanyahu, R. Silverman, and A. Y. Wu, "An optimal algorithm for approximate nearest neighbor searching fixed dimensions," *Journal of the ACM (JACM)*, vol. 45, no. 6, pp. 891–923, 1998.

[58] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[59] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[60] G. Salamuniććar, S. Lončarić, A. Grumpe, and C. Wöhler, "Hybrid method for crater detection based on topography reconstruction from optical images and the new lu78287gt catalogue of lunar impact craters," *Advances in Space Research*, vol. 53, no. 12, pp. 1783–1797, 2014.

[61] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[62] F. Benford, "The law of anomalous numbers," *Proceedings of the American philosophical society*, pp. 551–572, 1938.

[63] https://esa.un.org/unpd/wpp/.

[64] M. LLC. (1999) MS Windows NT kernel description. [Online]. Available: https://esa.un.org/unpd/wpp/Download/Standard/Population/

[65] S. Newcomb, "Note on the frequency of use of the different digits in natural numbers," *American Journal of Mathematics*, vol. 4, no. 1, pp. 39–40, 1881.

[66] J.-M. Jolion, "Images and benford's law," *Journal of Mathematical Imaging and Vision*, vol. 14, no. 1, pp. 73–81, 2001.

[67] J. Sanches and J. S. Marques, "Image reconstruction using the benford law," in *Image Processing, 2006 IEEE International Conference on*. IEEE, 2006, pp. 2029–2032.

[68] A. Iorliam, A. T. Ho, N. Poh, and Y. Q. Shi, "Do biometric images follow benford's law?" in *Biometrics and Forensics (IWBF), 2014 International Workshop on*. IEEE, 2014, pp. 1–6.

[69] I. Amerini, R. Becarelli, R. Caldelli, and A. Del Mastio, "Splicing forgeries localization through the use of first digit features," in *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on.* IEEE, 2014, pp. 143–148.

[70] S. Tong, Z. Zhang, Y. Xie, and X. Wu, "Image splicing detection based on statistical properties of benford model," in *Proceedings of the 2nd international conference on computer science and electronics engineering*, 2013, pp. 792–795.

[71] F. Pérez-González, G. L. Heileman, and C. T. Abdallah, "Benford's law in image processing," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*, vol. 1. IEEE, 2007, pp. I–405.

[72] B. Li, Y. Q. Shi, and J. Huang, "Detecting doubly compressed jpeg images by using mode based first digit features," in *Multimedia Signal Processing, 2008 IEEE 10th Workshop on.* IEEE, 2008, pp. 730–735.

[73] J. Wang, B.-H. Cha, S.-H. Cho, and C.-C. J. Kuo, "Understanding benford's law and its vulnerability in image forensics," in *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on.* IEEE, 2009, pp. 1568–1571.

[74] X. Zhao, A. T. Ho, and Y. Q. Shi, "Image forensics using generalised benford's law for accurate detection of unknown jpeg compression in watermarked images," in *Digital Signal Processing, 2009 16th International Conference on.* IEEE, 2009, pp. 1–8.

[75] G. Qadir, X. Zhao, and A. T. Ho, "Estimating jpeg2000 compression for image forensics using benford's law," in *Optics, Photonics, and Digital Technologies for Multimedia Applications*, vol. 7723. International Society for Optics and Photonics, 2010, p. 77230J.

[76] G. Qadir, X. Zhao, A. T. Ho, and M. Casey, "Image forensic of glare feature for improving image retrieval using benford's law," in *Circuits and Systems (ISCAS), 2011 IEEE International Symposium on.* IEEE, 2011, pp. 2661–2664.

[77] B. Xu, J. Wang, G. Liu, and Y. Dai, "Photorealistic computer graphics forensics based on leading digit law," *Journal of Electronics (China)*, vol. 28, no. 1, p. 95, 2011.

[78] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple jpeg compressions using first digit features," *APSIPA Transactions on Signal and Information Processing*, vol. 3, 2014.

[79] C. Pasquini, F. Pérez-González, and G. Boato, "A benford-fourier jpeg compression detector," in *Image Processing (ICIP), 2014 IEEE International Conference on.* IEEE, 2014, pp. 5322–5326.

[80] C. Pasquini, G. Boato, and F. Perez-Gonzalez, "Multiple jpeg compression detection by means of benford-fourier coefficients," in *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on.* IEEE, 2014, pp. 113–118.

[81] C. Pasquini and G. Boato, "Jpeg compression anti-forensics based on first significant digit distribution," in *Multimedia Signal Processing (MMSP), 2013 IEEE 15th International Workshop on*. IEEE, 2013, pp. 500–505.

[82] S. Milani, M. Tagliasacchi, and S. Tubaro, "Antiforensics attacks to benford's law for the detection of double compressed images," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3053–3057.

[83] R. Yang, Y. Q. Shi, and J. Huang, "Detecting double compression of audio signal," in *Media Forensics and Security II*, vol. 7541. International Society for Optics and Photonics, 2010, p. 75410K.

[84] W. Jiang, A. T. Ho, H. Treharne, and Y. Q. Shi, "A novel multi-size block benford's law scheme for printer identification," in *Pacific-Rim Conference on Multimedia*. Springer, 2010, pp. 643–652.