THESIS

AUTOMATED IDENTIFICATION OF OBJECTS IN AERIAL IMAGERY USING A CNN : OIL/GAS SITES TO MARTIAN VOLCANOES

Submitted by
Sonu Dileep
Department of Electrical and Computer Engineering

In partial fulfillment of the requirements
For the Degree of Master of Science
Colorado State University
Fort Collins, Colorado
Summer 2021

Master's Committee:

Advisor: Ross Beveridge

Mahmood R Azimi-Sadjadi
Michael Kirby

ABSTRACT

AUTOMATED IDENTIFICATION OF OBJECTS IN AERIAL IMAGERY USING A CNN : OIL/GAS SITES TO MARTIAN VOLCANOES

The recent advancements in Deep Learning techniques have revolutionized the field of computer vision. Deep Learning has received massive attention in remote sensing. The availability of open-source satellite imagery has opened up lots of remote sensing applications, from object detection to disaster assessment. In this work, I explore the application of deep learning for automated identification of oil/gas sites in DJ Basin, Colorado[1] and detection of volcanoes on Mars[2]. Oil and gas production sites are one of the significant sources of methane emissions all over the world. Methane emission studies from oil/gas sites require a count of major equipment in a site. However, these counts are not properly documented, and manual annotation of each piece of equipment in a site takes a lot of time and effort. To solve this challenge, an end-to-end deep learning model is developed that finds the well sites from satellite imagery and returns a count of major equipment at each site.

Second, an end-to-end deep learning approach is used to detect volcanoes on Mars. Volcanic constructs are fundamental in studying the potential for past and future habitable environment on Mars. Even though large volcanic constructs are well documented, there is no proper documentation for smaller volcanoes. Manually finding all the smaller volcanoes will be a tedious task. In the second part of my work, I explore the potential of deep learning approaches for Martian volcano detection.

ACKNOWLEDGEMENTS

I would like to thank my advisor Dr.Ross Beveridge and Daniel Zimmerle, CSU Energy Institute for their deep insights and guidance throughout the research. Thank you both for being patient and giving me time to explore. Also I would like to thank Dr.Mahmood R Azimi and Dr.Michael Kirby for graciously serving on my thesis committee. Last, I would also like to acknowledge my parents, Dr.Dileep Kumar and Girija M, and my brother Deepak Dileep for their support and encouragement throughout my years of study and during research. This work would not have been possible without them. Thank you, all.

DEDICATION

To my loving parents and brother

Without their support, understanding, encouragement, and love this work would not have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Remote sensing has many applications in Weather Monitoring, Climate Change, Agriculture, Forest Monitoring, Road Analyzes, Disaster Assessment, etc. Currently, around 700 satellites have been deployed for earth observation[3]. The type of imaging sensors used in a satellite can capture images at different electromagnetic spectrums. The quality of satellite images has improved consistently over the last few years. Satellite imageries support different studies and research on Earth. Not only on Earth, but remote sensing techniques are also used to study Mars. The sensors used on Mars rovers, landers, and orbiters have helped us study and explore the surface of Mars and the atmospheric conditions. Satellite imageries are generally high-resolution images, and it will be a time-consuming task to manually collect information from this vast dataset. Recent breakthroughs in machine learning have opened up much scope in remote sensing analysis. The introduction of Deep Learning techniques has helped us automate the analysis and better understand the data captured from these satellite images. Also, the recent advancements in hardware units such as CPUs and GPUs have helped reduce the processing time of large satellite images. My work on deep learning in remote sensing is split into two parts. The first work is part of a more extensive methane emission study by CSU Energy Institute. I applied deep learning techniques to identify oil and gas well sites in Colorado from Google satellite imagery. The second work is focused on using deep learning techniques to build an inventory of small to midsize volcanic edifices on Mars.

Methane, the main component of natural gas, is a potent greenhouse gas with a global warming potential over 80 times that of CO2 on a 20-year basis[4]. Reducing methane emissions has been identified as a critical requirement to limit global warming[5,6,7,8,9]. Active natural gas production sites account for 67% of the total methane emission from the oil and gas industry[10]. A study was conducted by nine cooperating U.S natural gas companies to study the methane emissions from well pads and production sites. The study showed that methane leaks from equipment and during transportation are much higher than previously thought[11].

The work described in first section is part of a top-down, bottom-up(BU), coordinated campaign to be conducted in the Denver-Julesburg (DJ) basin in northeastern Colorado in 2021. Methods planned for this study mirror those utilized in the Fayetteville coordinated study [12,13,14,15] and the Barnett coordinated campaign[17]. Methods utilized for these studies involve the development of 'bottom-up' (BU) inventories of emissions. Inventories utilize prior emissions distributions, resolved to the component level to represent emissions from individual leaks. These emissions data are multiplied, using Monte Carlo methods, with activity data consisting of counts of major equipment, components per major equipment unit, and other operational data. These calculations result in an empirical distribution of emissions expected from all oil and gas facilities in a basin. When additional information is available to resolve the timing of emissions, BU estimates may be extended by modeling the timing of emissions events, and, potentially, the downwind dispersion of pollutants.

Since BU studies rely on counts of equipment and components, the accuracy of the counts has a substantial impact on the modelled emissions. The location of wells is generally well known, as the location of the borehole, drilling direction, and production is reported to state authorities. While traditional field development integrated wells directly onto well pads containing initial processing equipment, such as separators and tanks, in recent development, wellheads are often at a substantial distance from the well pad containing that equipment. Further, reporting of well pad locations is typically not required, and few location-specific data sets exist. To fill this data gap for an upcoming methane emission study campaign by Dan Zimmerle and his team at CSU Energy Institute, I worked on developing a machine learning model to identify well pads and it's features in the DJ basin from recent google satellite imagery with 70 cm resolution[1].

My second work is in collaboration with Dr. Susan Sakimoto, The University at Buffalo and her team, to apply deep learning techniques to identify small to midsize volcanic edifices on Mars[2]. Volcanoes are crucial for studying the geographical evolution of Mars and also to study the past and future habitable environment in Mars. Even though few databases already exist for large volcanoes, there is no proper documentation for small volcanoes[18,19]. There are few image datasets to study the surface of Mars but manually finding all the small volcanoes in these high resolution images will be a tedious task. So in this work I also explore the potential of deep learning and image processing technique to detect and create a database of the small volcanoes in Mars.

The thesis is organized as follows. Chapter-1 gives a short introduction to the two topics that I worked on. Chapter-2 explains the recent works in the field of object detection from satellite imagery on earth and mars. It also gives an overview of recent object detection and object segmentation models. Chapter-3 is focused on the topic Oil/Gas site detection from satellite imagery. Chapter-3 gives a detailed explanation about the methods used, image collection, training, and finally, the results are discussed. Chapter-4 is focused on the topic Mars Volcano Detection using Deep Learning, and I discuss the models used, data collection methods, training, and results.

Chapter 2

RELATED WORKS

In this section I briefly discuss the previous works in the field of object detection, image segmentation, object detection in satellite imagery on Earth and Mars.

2.1 Object Detection models

The emergence of deep learning has improved the performance of object detection algorithms. Object detection models predict a bounding box around the object. Most CNN based object detection models run in two stages - a backbone and a head. Backbone is usually trained on ImageNet for feature extraction, and head helps in bounding box detection and class prediction. The most commonly used backbones are VGG[68], ResNet[69] or DenseNet[70]. But if these neural nets run on embedded devices the backbone could be smaller architectures such as MobileNet[71,72,73,74] or SqueezeNet[75]. The head is usually classified into two types - a one-stage detector and a two-stage detector. Few examples of the one-stage detector are YOLO, SSD, and CenterNet[76]. The most commonly used two-stage detectors are Fast R-CNN[77], Faster R-CNN[78], and R-FCN[79]. Apart from the backbone and head, most object detection models has a layer in between head and backbone. This layer, also called a Neck, helps to get features from intermediate stages. Few examples of models with Neck are Feature Pyramid Network[80], Path Aggregation Network[81], and BiFPN[82].

2.2 Image Segmentation models

Image segmentation models have developed considerably over the last few years. In segmentation problems, the model tries to assign a label for each pixel in an image. Most image segmentation models are built on the success of Region-based CNN(RCNN)[83]. Models like DeepMask[84] and other works described in these papers[85,86] learn to propose segment candidates and then use a Fast R-CNN to classify the segments. Most of these methods follow segmentation followed by recognition. However, these methods are slow and less accurate. A new method for segmentation from bounding box proposals has been developed by Dai et al. [87]. He proposed a multi-stage segmentation model using bounding box predictions followed by classification. Li et al. [88] proposed a model that simultaneously predicts classes, boxes, and segmentation masks. However, these methods were error-prone in cases with overlapping objects. Mask R-CNN[63] model solved most of these problems using their instance-first strategy. Mask R-CNN is faster and more flexible than other segmentation models.

2.3 Object Detection in Satellite Imagery using Deep Learning

Much work has been done on object detection from satellite imageries. In the paper Satellite Image Classification with Deep Learning[89], the authors presented a deep learning approach to classify objects from the IARPA Functional Map of the world. Their dataset consisted of 63 classes of objects, and they trained a CNN with these images. The authors achieved a total average accuracy of 83% and achieved an accuracy of 95% for 15 classes. The paper showed the possibility of using deep learning to achieve high object detection accuracy on satellite images. However, the paper did not mention how to extend the method for real-time object detection on satellite imagery. In the paper, You Only Look Twice: Rapid Multi-Scale Object Detection In Satellite Imagery[21], the authors developed a pipeline to approach object detection on low-resolution satellite imagery. The small objects are hard to find on satellite images using traditional CNNs. They changed the Yolo architecture to accommodate this problem. Their model can detect objects at very different scales and even objects around 5 pixels in size. The paper "The Effects of Super-Resolution on Object Detection Performance in Satellite Imagery"[22] explored how super-resolution can be applied to satellite imagery. They enhanced the low-resolution satellite imagery and tested whether it is possible to detect small objects. They tested different enhancement levels of 2x, 4x and 8x, and trained YOLO and SSD for object detection tasks. However, this method showed only a tiny improvement in detection performance. Stanford Machine Learning group developed OGNet[23], a deep learning framework to detect large oil and gas infrastructure from satellite imagery to study methane emissions. Their work was focused only on large oil/gas infrastructure and did not include well pads. Their work also did not focus on major equipment on site. They used NAIP satellite imagery and found many major sites not included in the public datasets of oil/gas infrastructure.

2.4 Object Detection on Mars using Deep Learning

Few works were also done prior for feature detection on Mars. Modern ML techniques, mostly using convolutional neural networks (CNNs), have been recently applied to the task of detecting features on Mars. Few prior works has used data from numerous instruments, including THEMIS and CTX [24,25,26], Mars Express High-Resolution Stereo Camera (HRSC) [Cohen, 2016; Wang, 2018], Lunar Orbiter Laser Altimeter (LOLA) [Silburt, 2019], MOLA [Lee, 2019], MESSENGER Mercury Laser Altimeter (MLA) [Silburt, 2019], and LRO Narrow Angle Camera (NAC) [Glaude, 2017; Mahanti, 2018; Emami 2018]. In addition to impact craters, CNN detection algorithms have been applied to rootless volcanic cones and transverse aeolian ridges [Palafox et al., 2017], rockfalls [Bickel et al., 2019], terrain types [Balme et al., 2019], and other landforms used for image indexing [Wagstaff et al., 2018]. Identifying and mapping volcanoes are difficult compared to impact craters or transverse aeolian ridges because of the extreme morphological variations observed in volcanoes. Few works have been done prior for the detection and mapping of rocks on Mars using Deep learning techniques[27] and also automating the detection of geological landforms on Mars using Convolutional Neural Networks[28].

2.5 Challenges Involved

There are a lot of challenges in object detection from satellite imagery. Large variation in visual appearances caused by sunlight, shadows, occlusions make satellite object detection a really challenging task. Some of these variations can be recreated by applying different data augmentation techniques. Also unless we don't have high resolution images, it is difficult to find the small objects. Processing and storage of high resolution satellite images takes a lot of space and computation. It is not always possible to save the images of an entire state or a country on a single machine. So, the method that we propose in this work could be a solution for detecting objects in satellite imageries without actually saving the datasets on a machine.

Chapter 3

OIL AND GAS FIELD DETECTION

The anthropogenic methane emissions play a major role in the accelerating global warming that the Earth is facing. Studying and analyzing the methane emissions are very critical in reducing global warming in the future. Several satellites have been launched to detect and study these emissions. For efficient methane emission studies, we must know the location of the emission source and the characteristics of each source. In this work, I focus on detecting and characterizing one of the major sources of methane emissions in Colorado, Oil, and gas sites. The oil and gas sector has emerged as a significant research area for government and industry to reduce methane emissions[29,30,31,32]. For a detailed study of methane emission at these sites, we need to understand the size of oil and gas sites and knowledge of major equipment on each site. Most of this information are not properly documented, and manual annotation of each site will take a lot of effort and time. So, we adapted a Deep Learning method to create a database of well site locations in DJ Basin, Colorado, and find the number of major equipment on each site. This work is in collaboration with Dan Zimmerle and his team at CSU Energy Institute and has been done as part of methane emission studies in DJ Basin, Colorado. The work has been published at the NIPS 2020 workshop tackling climate change with machine learning[1], and a detailed report of the work is provided in this thesis.

3.1 Dataset

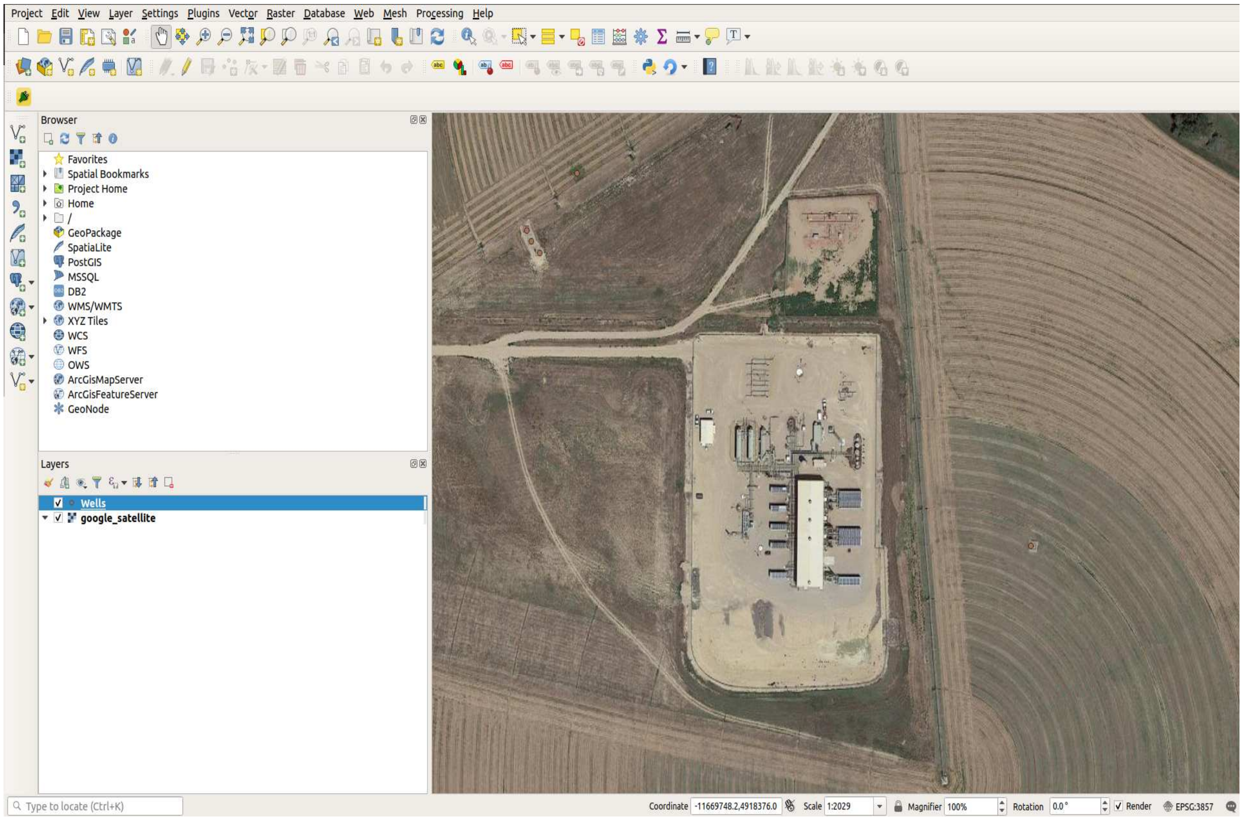
Due to the lack of publicly available datasets of well pads and equipment, we manually collected and annotated images. We created a dataset containing well pads and major well site equipment in Colorado using Google satellite imagery. To ease the process of image annotation, we developed an annotation plugin using QGIS. We collected and annotated over 1650 images of well pads in Colorado and over 500 images of major equipment in each well site. There were multiple well-pads and equipment in a few of these images. We annotated over 2800 well pads and 1700 equipment across entire Colorado. We collected images from 3 different parts of the state: northwest, southwest, and central Colorado. We choose images from 3 different parts of the state to generalize the model and better predict the unseen dataset. The well sites in these three parts of the state have different features. In the northwest part of Colorado, well sites are surrounded mainly by deserted land. Most well pads are around buildings and roads in central Colorado, and most well pads in southwest Colorado are surrounded by forest/vegetation.

The major equipment on each site looks similar, and we do not necessarily have to collect too many training samples of them. They vary due to factors such as a change in sunlight, seasonal changes, size, shadows, and few other variations due to their surroundings. So we did not include too many training samples of them, and we annotated only 500 images of the site equipment to train our model. One other aspect of applying ML to recognizing equipment became apparent as the dataset was being collected. A variety of other objects near and in gas production fields look similar to some site equipment, such as grain and agricultural water storage tanks. Therefore, we also collected over 50 negative samples of storage tank-like structures to better train the ML model and to avoid false positives, i.e., mistaking these other distractors for site equipments.

3.1.1 Image Collection and Annotation Plugin

Image annotation is the task of labeling images with a label for training a deep learning model. While annotating, we draw bounding boxes around the objects of interest in an image. Then the information or location of these bounding boxes is saved and passed to the deep learning model for training. The quality of labeling depends on the individual who does the annotation and how they handle different factors like occlusion, blurriness, and other external factors. Proper data collection and annotation is an essential step in all machine learning and computer vision models. It determines the overall precision and accuracy of the model. We focused on detecting Well pads, Well tanks, Separators, Flares, and Pumpjack in this project. Annotation includes a lot of manual human labor. The first step for training any machine learning model is data. There is no straightforward approach to collect and annotate satellite imagery. Most of the methods require downloading large datasets of satellite imagery as tiles which takes up a lot of memory and time. Even after image collection, annotating the multiple images from each tile is a time-consuming task, and there are no known easy ways to do it.

In this work, we introduce a new annotation plugin which helps to do annotation and image collection in a single step. This method also does not require downloading and storing large satellite imagery dataset. The entire plugin is implemented in a GIS software QGIS. QGIS is an open-source GIS software which helps in viewing and analyzing geospatial data. Figure 3.1 shows the Google Satellite Imagery layer in QGIS. QGIS helps to solve the problem of downloading large satellite imagery tiles, but the only downside is that it requires an active internet connection while data collection and annotation. The User interface of the developed plugin is shown in Figure 3.2.



☑ Wells
☑ google_satellite

Coordinate -11669748.2,4918376.0 Scale 1:2029 Magnifier 100% Rotation 0.0° ☑ Render EPSG:3857

Figure 3.1: Google Satellite Imagery in QGIS

The plugin is developed using PyQt and written in Python. The different push buttons in the plugin are the labels of different objects that need to be annotated. We start with specifying the

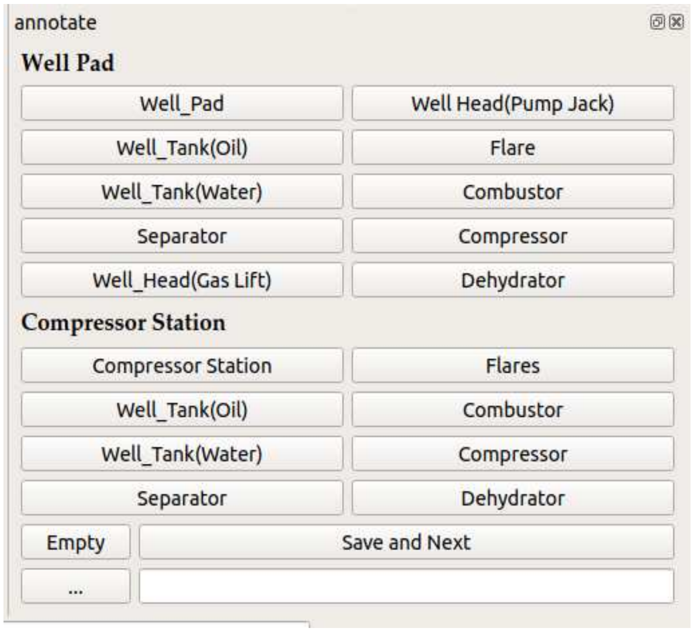


Figure 3.2: User Interface of Annotation Plugin

path where we want to save the images and labeled data. Then use the corresponding push buttons to draw bounding boxes around each object in map canvas in QGIS. Once we are done labeling at a particular well site, press Save and Next to save the image and corresponding labels. Figures 3.3 and 3.4 show a few well pad and equipment annotations using the plugin.

3.1.2 Data Augmentation

Data Augmentation is the method of increasing the size of the dataset by applying different techniques to the available dataset. It includes modifying the existing dataset by applying different transformation techniques or by creating new synthetic data from existing ones. This method also helps to reduce the overfitting of the model and has the robustness to images from different environments. Data Augmentation is also useful when we have only a limited amount of dataset to train a model. Few Geometric data augmentation techniques used are rotation, scale, transformation, crop, flip, etc. Few photometric distortion methods include changing color, brightness, contrast, hue, saturation, and adding noise. MixUP [92] is an augmentation technique in YoloV4 where we do a weighted multiplication of 2 images to create a new one. The labels are then adjusted with the weights. MixUp helps to improve the performance of the model on adversarial examples. CutMix is another augmentation technique where patches from different images are pasted on the training images. This helps to improve the robustness of the model against corruptions in the image and detect occluded objects.

Figure 3.3: Well Pad Annotation [33]

Figure 3.4: Equipment Annotation [33]

3.2 CNN Architecture

Computer Vision is the field of artificial intelligence that deals with analyzing images and video. The emergence of deep learning has facilitated advancement in the field of computer vision tasks like object detection[35,36] and segmentation[34]. One primary reason behind the success of deep learning techniques was the introduction of Convolutional Neural Networks. CNN's are a class of neural networks that are used in image classification, detection, and segmentation tasks. They are made up of artificial neurons with learnable parameters. They are commonly used in different deep learning tasks such as face detection, self-driving cars, traffic sign detection, medical diagnosis, etc. CNN has powered most of the recent advancements in the field of deep learning.

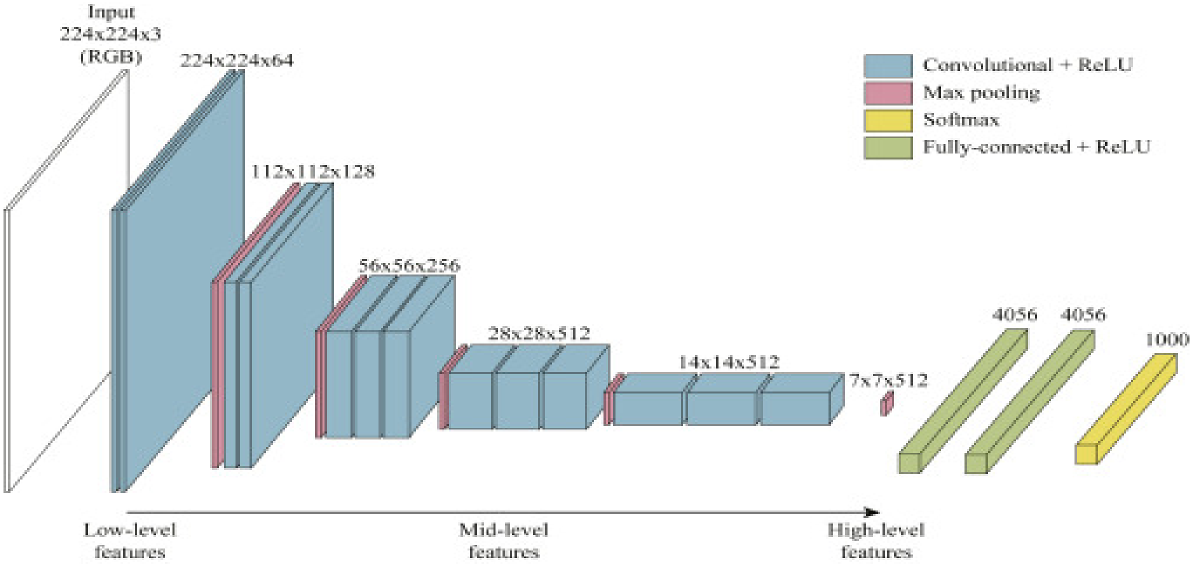


Figure 3.5: VGG16, Convolutional Neural Net Architecture[67]

One thing that makes CNNs different from other neural networks is the presence of convolution kernels in the architecture. These convolution kernel helps to identify the features which are important for the different tasks they are doing. These layers work similarly to the way the human brain analyzes images. The neural network architectures are often deep, and the initial few layers learn the small and local features in an image, and as the layers go deeper, it starts learning more complex features about the image, which are hard to interpret by humans. This deep learning method helps to understand the images much better and learn complex features. Figure 3.5 shows the architecture or layers in a VGG16 CNN model.

3.2.1 Types of Layers

These are some of the crucial layers common to all Convolutional Neural Net architecture.

1. Input Layers - This layer takes the input image/data and passes it to the next layer.

2. Convolution layer - The convolution layer performs convolution on small tiles of the input image. The input image is often split into multiple smaller tiles, and convolution operation is performed on each of these tiles. Each tile is connected to the neurons in convolutional layers and computes a dot product between the weights and input pixel values. Most times, we stack multiple convolution filters to learn more complex features.

3. Pooling Layer - The pooling layer performs down-sampling operations on the previous layer. This helps to reduce the dimension of each feature passing through different layers. The two commonly used pooling techniques are Max pooling and Average Pooling. In max-pooling, only the maximum value in each filter is passed to the next layer. In average pooling, average values are taken for each filter and are then passed to the following layers. Figure 3.6 shows the operation of Max Pooling and Average Pooling layers on an input of size 4 x 4.

4. Fully-Connected Layers - After Convolutional and Max Pooling layers the output results are flattened and then passed into Fully Connected Layers. In Fully connected layers the input is connected to all neurons from the previous layers. These are usually added at a much deeper layer of the CNN model.

3.2.2 Hyperparameters in Convolution Filters

Convolution filters are the most important component in a convolutional neural network. So finding the correct hyperparameters for the filters affect the performance of the model. Three hyperparameters that are used to describe a filter are:-

1. Dimensions of Filter - In convolutional networks, filters are of size N x N x C. Here C is the filter's depth, which is always the same as the depth of input to that filter. N x N refers to the width

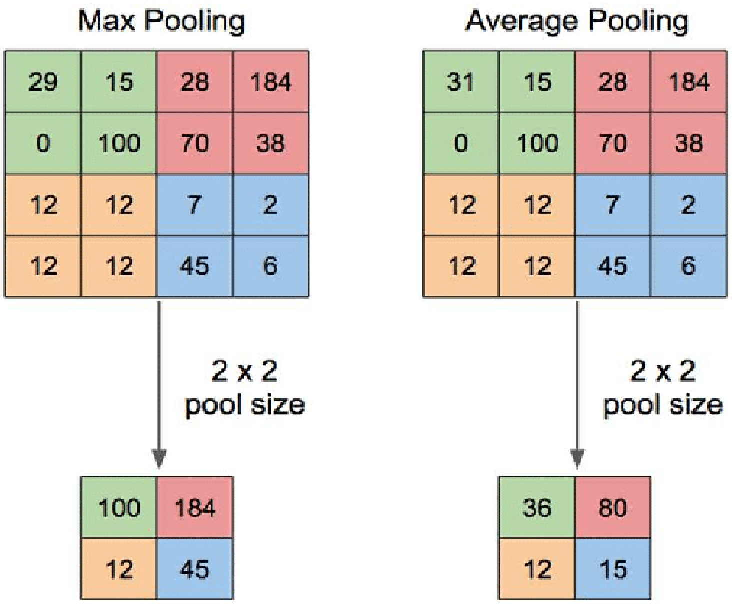


Figure 3.6: An illustration of Max Pooling and Average Pooling operations on a input of size 4 x 4

and height of the filter. N decides the number of pixels the filter should convolve over at one time. A filter of size N x N x C can perform convolution on an input of size W x H x C and return an output feature map of the size I x I x 1, where I is the size of output features.

2. Stride - Stride controls the number of pixels the filter should move after each convolution operation. Figure 3.7 shows a stride of 2 on one-dimensional data and shifts the center of the window by two on each pass. A stride of values other than 1 is used either for computational efficiency or to perform some downsampling operation, by skipping multiple elements in between.

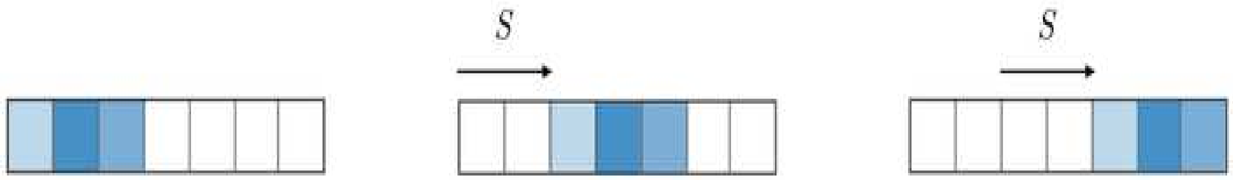


Figure 3.7: Stride=2 on a 1D vector

3. Zero-Padding - In convolutional layers, convolution operations result in losing the pixel values on the boundary of the image or data. One solution to this problem is by applying zero paddings. In zero-padding, we add one or more rows and columns of zeros to the boundary of an image. This helps against the data loss at the boundary during convolution. Figure 3.8 shows zero-padding of input and convolution operation followed by it.

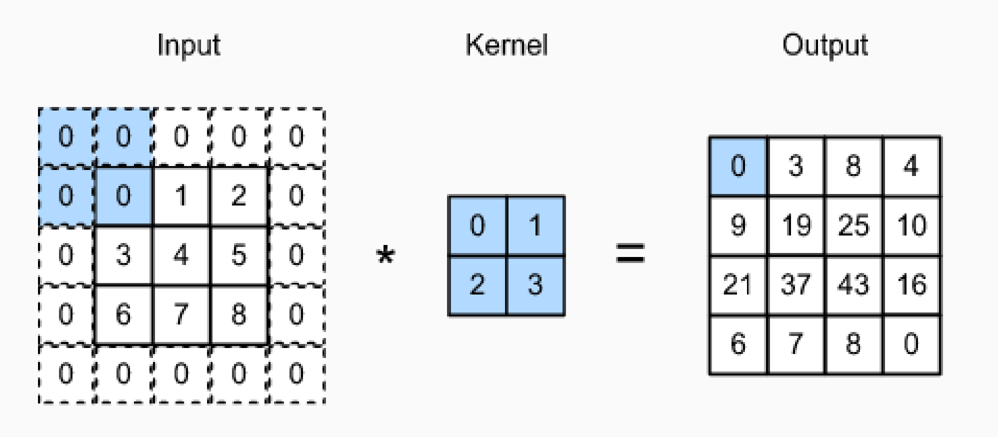


Figure 3.8: Zero padding of inputs

3.2.3 Activation Functions

The activation function transforms the weighted sum of input from a node or multiple nodes. The most common classes of activation functions are ridge, radial, and fold functions.

1. ReLU(Rectified Linear Unit) - ReLU[39] is a piecewise linear function which outputs the maximum of zero and the input. If the input is positive, then the output of ReLU will be the same as the input. If the input is zero or less than zero, then the output will be zero. ReLU helps to avoid the vanishing gradient problem when the model is learning. Figure 3.9(a) shows the plot of a ReLU function.

2. Leaky ReLU - Even though ReLU solves the vanishing gradient problem, it faces Dying ReLU issue. In Dying ReLU, the neurons won't be activated no matter what the input is. This affects the gradient flows and affects the training process. This can be rectified using leakyReLU[38].

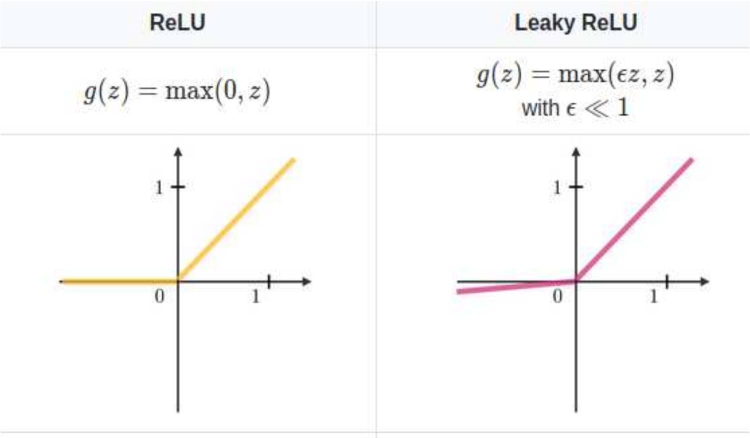


Figure 3.9: Activation Functions - ReLU and Leaky ReLU

In LeakyReLU, instead of setting negative values to zero, we set the output to a*(input). Usually, the value of a will be 0.01, and when a is not 0.01, it is called randomized ReLU. Figure 3.9(b) shows the plot of LeakyReLU.

3. Mish Activation - Mish is a smooth, non-monotonic activation function that combines tanh and softplus activation function[37]. It can be written as: f(x) = x * tanh(sig(x)) sig(x) = ln(1 + exp(x)). Figure 3.10 shows the input vs output plot of a mish activation function

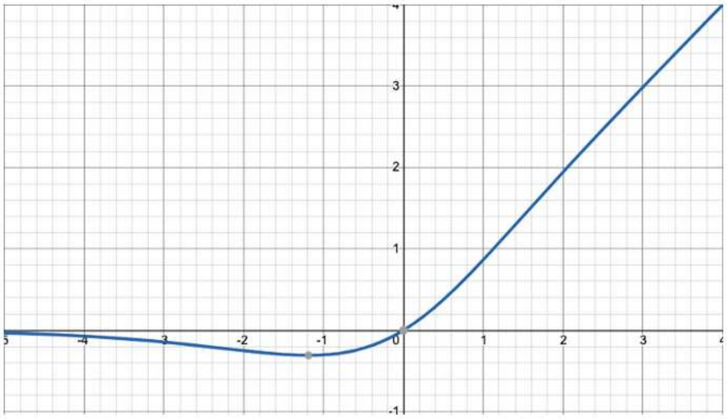


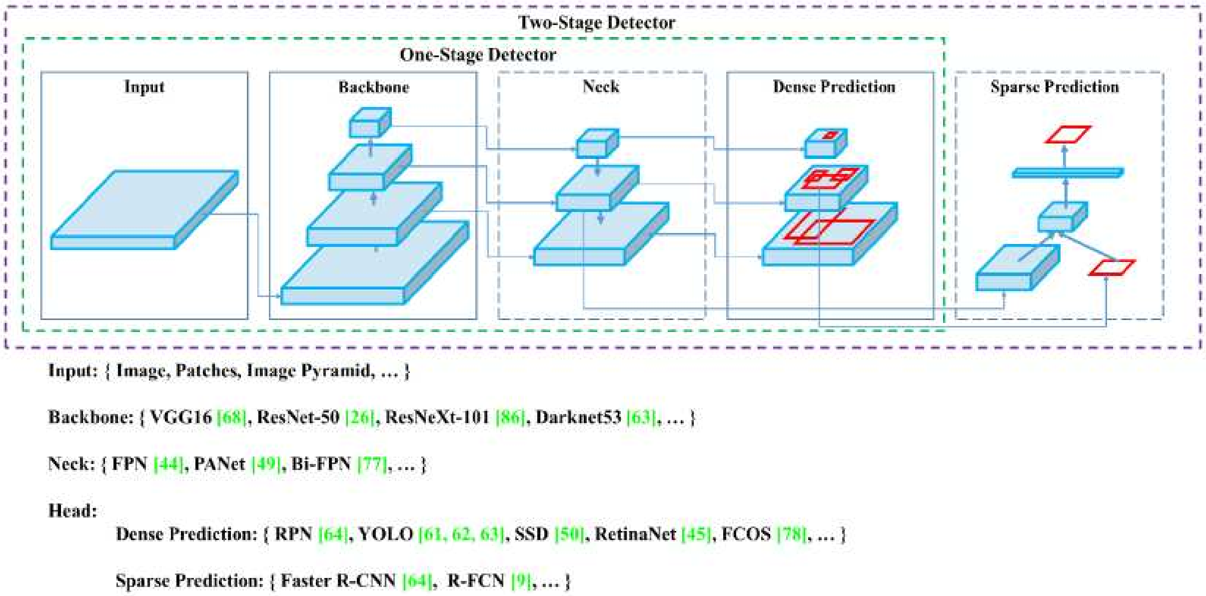
Figure 3.10: Activation Function - Mish

3.2.4 Loss Functions

Loss functions are an evaluation metric that tells us how the model/algorithm performs. Loss functions explain how much the prediction deviates from the actual values. There are different types of loss functions used, such as Mean Squared Error, Mean Absolute error, Mean Bias Error, Hinge Loss, Cross-Entropy loss etc. The most commonly used loss function for object detection tasks are Intersection over Union(IoU)[40] and its variants. Intersection over Union(IoU) is a metric that explains how much the actual bounding box and the predicted bounding box overlaps. If the predicted output completely overlaps the true bounding box, then the IoU value is 1. The IoU loss is defined as the difference between 1 and IoU. IoU Loss = 1 - IoU(A,B), where A and B are predicted bounding box and actual bounding box. A major drawback of IoU is that it works only when the bounding boxes overlap. When it doesn't, the gradient will not be propagated back. To tackle this issue, improved variants of IoU loss like CIoU[41], DIoU[41], and GIoU[42] have been developed. CIoU has shown the best performance on object detection tasks, and it considers three geometric factors in calculating the loss. It considers overlapping area, distance from center point, and aspect ratio in finding the loss value. CIoU loss can be defined as the sum of S(A,B) + D(A,B) + V(A,B), where S(A,B) is the overlap area, D(A,B) is the normalized central point distance, and V(A,B) is the aspect ratio.

3.2.5 YOLOv4

The introduction of deep learning has shown great improvement in the field of computer vision tasks such as object detection, segmentation, and classification. The availability of large image datasets like ImageNet, MS COCO, and PASCAL VOC has helped researchers to train better models for detection and segmentation tasks. Few different architectures available for object detection tasks are YOLO[43,44,45,46], R-CNN[47], SSD[48] and many more. We evaluated the performance of these models on our dataset. Since Yolo-V4 has a fast inference speed and accuracy compared to other deep learning models for our well pad detection task, we decided to move forward with Yolo-V4[43]. The first YOLO was developed by Joseph Redmon et al.; in 2015, and since then, three additional models have been proposed, Yolo-V2, Yolo-V3 and Yolo-V4. Yolo-V4 was released in 2020 and has shown immense potential compared to the previous models. The Yolo series are end-to-end trained, which takes in an input image and predicts the bounding box and class labels directly.



Input: { Image, Patches, Image Pyramid, ... }

Backbone: { VGG16 [68], ResNet-50 [26], ResNeXt-101 [86], Darknet53 [63], ... }

Neck: { FPN [44], PANet [49], Bi-FPN [77], ... }

Head:
- **Dense Prediction:** { RPN [64], YOLO [61, 62, 63], SSD [50], RetinaNet [45], FCOS [78], ... }
- **Sparse Prediction:** { Faster R-CNN [64], R-FCN [9], ... }

Figure 3.11: Object detector Architecture - One Stage and Two Stage Detector

Yolo-V4 detects objects in 2 steps with a backbone and a head. The backbone is a CNN model and is trained on Imagenet[49]. This stage helps to extract the features from the image. The second stage is the head which predicts classes and bounding box of objects. Figure 3.11 shows the two commonly used architectures of object detection models - One-stage detector and two-stage detector. In one stage detector, the object detection task is treated as a regression problem, and the model tries to predict the class probabilities and bounding box coordinates. Yolo is a one-stage detector. A two-stage detector uses a region proposal network to generate bounding boxes directly. Most architectures also use a layer called Neck in between backbone and head, and this helps to collect feature maps from different stages to facilitate skip connections.

YoloV4 has shown significant performance improvement compared to Yolo-V3[46]. Yolo-V4 improved AP and FPS by 10% and 12% compared to Yolo-v3. The Yolo-V4 architecture consists of mainly four steps - CSPDarknet53[50] as a backbone, Spatial Pyramid Pooling(SPP)[51] and Path Aggregation Network(PAN)[52] as the Neck and YoloV3 as head. To improve the accuracy further authors also added few methods called Bag of Freebies[43] and Bag of Specials[43] for the backbone and detector stage. These contain better data augmentation techniques, improved loss functions, better activation functions, and few methods to converge faster. Shown below is the bag of freebies and bag of specials used in Yolo-V4.

- Bag of Freebies (BoF) for backbone[43]: CutMix and Mosaic data augmentation, DropBlock regularization, Class label smoothing
- Bag of Specials (BoS) for backbone[43]:Mish activation, Cross-stage partial connections, Multi-input weighted residual connections (MiWRC)
- Bag of Freebies (BoF) for detector[43]:CIoU-loss, Cross mini-Batch Normalization(CmBN), DropBlock regularization, Mosaic data augmentation, Self-Adversarial Training, Eliminate grid sensitivity, Using multiple anchors for a single ground truth, Cosine annealing scheduler, Random training shapes
- Bag of Specials (BoS) for detector[43]: Mish activation, SPP-block, SAM-block, PAN path-aggregation block

3.3 Training

We used well pads of all different sizes and features to train the model. YOLO takes in images of size 606 x 606 as input, so we either downsampled or upsampled each image as an input to the model. The different data augmentation techniques that we applied on the dataset are scaling, blur, cut-mix and rotation. We trained the model over 10,000 iterations with a step decay learning rate strategy. We used an initial learning rate of 0.001, and it will be multiplied by a factor of 0.1 after every 5000 iterations. We used the CIoU-loss function for training with a momentum of 0.949 and decay of 0.0005. The entire training is done on a single Nvidia GTX 1070 GPU. The training took around 15 hours to finish, and the loss did not go down after 10,000 iterations.

3.4 Performance Evaluation

To evaluate the performance of the well site detection model, we used a 5-fold cross-validation strategy. To perform a five fold-cross validation, we first divide the training samples evenly into five sets. Then we used four sets for training the model and the remaining one set for validation. This is repeated until all five different combinations are used for training the model. We applied a 4-fold cross-validation strategy on equipment detection models. The loss function that we used for training both models is CIoU loss. We ran the training until the average loss no longer decreased. The performance of the model is measured using standard measures. We set an Intersection over Union(IoU) value thresholded at 0.3 for detections. Any detections with IoU above 0.3 will be classified as a true positive. To evaluate the model, we used F1 scores. F1 score is a standard measure defined to integrate True Positive, False Positive, and False Negative findings[53]. True positive tells us about the percentage of labeled examples detected by the model. False-positive rate shows the percentage of detections that are erroneous. False-negative rate tell us about the actual labeled instances not detected as a fraction of all the labeled instances.

3.5 Methods

Before training, we initialized the YOLO-V4 weights from a model trained on ImageNet[49]. We applied the transfer learning technique to train our model using only relatively limited training data of well pads and site equipment. To train the model, we used google satellite images collected from QGIS using the developed plugin. We collected over 1650 images of well pads from different parts of Colorado and over 500 images of different equipment on each site. After training, the weights generated are used for testing the model. The entire implementation of the model is in QGIS. We used the python interface of QGIS to run the model. The four coordinates(latitude, longitude) of the area of interest are the inputs to the python code. Then the program starts collecting images as patches from this area and feeds them into the Yolo model. In order to save testing time, we decided to pass the array of images directly to the model instead of saving the images. Thereby saving the load and save time. This helped us to reduce testing time by 1/3 rd. Also, since we worked with really high-resolution satellite images, we collected tiles from these large images using smaller sliding windows. These sliding windows move across the whole image and collect patches of images. Instead of using a fixed window size, we used an adaptive window approach so that the sliding window can change size if needed. We set the sliding window size to 500 x 500, which collects images of size 500 x 500 pixels. We also set the stride to 450. The window adjusts size if the well pad is huge, and the well pads' boundaries are close to the window's boundaries. Initially, the window size will be increased by 100 on both sides, and the process repeats until the well pad fits inside the window. This approach helped us to find well pads of different sizes. The adaptive sliding window approach is used only for well pad detection and not for equipment since they are all of the same sizes. A key constraint employed in this work is that equipment are most often located within well pads. Consequently, first well pads are detected and then equipment is looked for inside well pads. This technique avoids searching vast areas for unlikely smaller equipment.

3.6 RESULTS - OIL/GAS FIELD DETECTION

The 1650 labeled well pad images were split into five sets of 330 images each. Out of 5, 4 sets are used for training and the rest 1 for validation. Five different combinations of these five sets are used for finding the best performing model. Most images have multiple well pad annotations. The ML well pad detection model trained and tested using a 5-fold cross-validation strategy is summarized in Table 3.1. Since fold-2 gave the highest F1 score, we choose the weights from fold-2 for our work. We used the same cross-validation technique for training equipment detection model but with a 4-fold cross-validation strategy. So we split the 500 images into four sets of 125 images each. The results are summarized in Table 3.2. Column 2 in Table 3.1 and Table 3.2 shows the actual number of hand-labeled well pads or equipment.

Table 3.1: 5-fold cross validation result for well pads

	Well Pad	True Positive	False Positive	False Negative	F1 Score
Fold-1	603	595/603(98.6%)	30/625(4.8%)	8/603(1.3%)	0.9690
Fold-2	665	658/665(98.9%)	19/677(2.8%)	7/665(1.0%)	0.9806
Fold-3	610	607/610(99.5%)	43/650(6.6%)	3/610(0.4%)	0.9607
Fold-4	545	544/545(99.8%)	39/583(6.6%)	1/545(0.1%)	0.9644
Fold-5	403	402/403(99.7%)	21/423(4.9%)	1/403(2.4%)	0.9733
Average		2806/2826(99.2%)	152/2958(5.1%)	20/2826(0.7%)	0.9696

Table 3.2: 4-fold cross validation result for major equipments

	Equipment	True Positive	False Positive	False Negative	F1 Score
Fold-1	291	290/291(99.6%)	0/290(0%)	1/291(0.3%)	0.9982
Fold-2	356	354/356(99.4%)	12/366(3.2%)	2/356(0.5%)	0.9805
Fold-3	389	389/389(100%)	4/393(1.1%)	0/389(0%)	0.9948
Fold-4	676	667/676(98.6%)	17/684(2.4%)	9/676(1.3%)	0.9808
Average		1700/1712(99.2%)	43/1733(2.4%)	12/1712(0.7%)	0.9885

Eventhough receiver operating characteristic(ROC) curve is a common technique to evaluate machine learning models, this particular technique is not useful for our detection model. ROC curve requires calculation of number of true negatives, but yolov4 did not output true negatives for our case. So we decided to use Precision-Recall to evaluate our model. Precision and recall are calculated using true positives, false positives and false negatives. All the predicted bounding boxes has a confidence score associated with it. Bounding boxes above some particular threshold are considered positive detections and any detections below the threshold are eliminated. To find the optimal threshold, we used precision and recall. We adjusted the detection threshold of the model until there is a balance between precision and recall. The precision and recall at different thresholds for well pad and equipment detection model are plotted in Figure 3.12 and Figure 3.13. Table 3.3 and Table 3.4 shows the tabular data of precision-recall values. As you can see from graphs, a threshold of 0.2 is ideal for our well pad detection model and a threshold of 0.3 is ideal for our equipment detection model. The entire model is implemented in an open-source GIS software, QGIS, which outputs a shape file containing the latitude and longitude information of each well pad, and the location of each piece of equipment detected. We ran the model over

Denver-Julesburg (DJ) basin and checked the results. Due to the unavailability of a database of exact well pad counts, we decided to check the accuracy at five different regions of the DJ basin. We collected rectangular tiles of 5000 x 5000 from these five regions and averaged the results. The well pad and equipment detection results over these 5 test regions in DJ Basin are described in Table 3.5 and Table 3.6. For well pads, we got an average true positive rate of 95% and for equipment, we achieved an average true positive rate of 92%.

Table 3.3: Precision-Recall for different detection thresholds - Wellpad

Threshold	Precision	Recall
0.15	0.96	0.99
0.20	0.97	0.99
0.25	0.97	0.97
0.30	0.97	0.96
0.35	0.98	0.95
0.40	0.98	0.95
0.45	0.98	0.93
0.50	0.98	0.89
0.55	0.99	0.89
0.60	0.99	0.86
0.70	0.99	0.83
0.80	0.99	0.80
0.90	0.99	0.78

Even though the model managed to pick up most of the well sites and equipment in DJ Basin, there are few False Positive detections and False Negatives. Fig 3.17 shows few false positive well tank and well pad detections. The top left image shows an area that is falsely classified as a well pad. The top right and Bottom left image show a well tank-like structure detected as well tanks. These are probable agricultural sites. The bottom right image has some false negatives where the model missed few well tank detections. Using high-resolution satellite imagery and applying the knowledge of oil/gas pipeline connections to find the probable areas of well pads could help avoid many false positive detections.

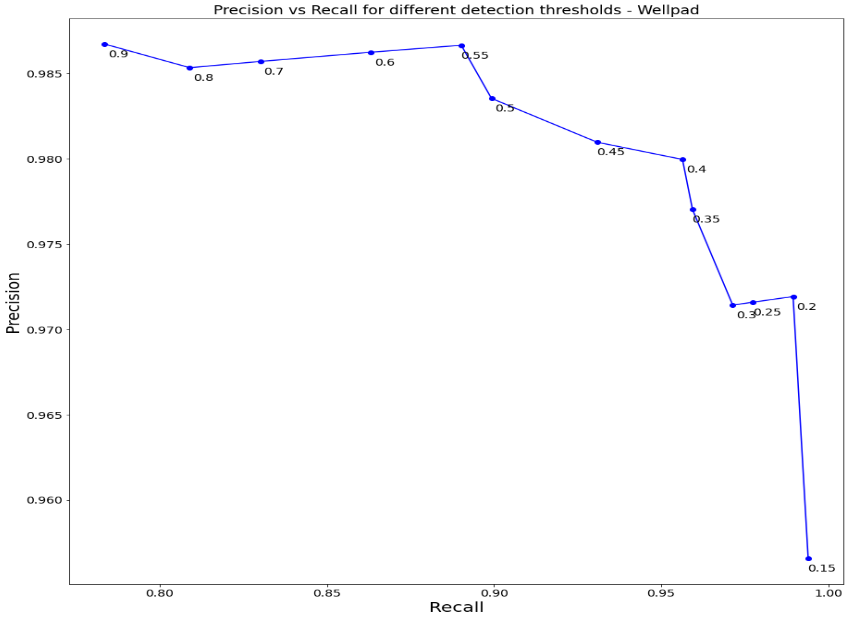


Figure 3.12: Precision-Recall Plot for different detection thresholds - Wellpad

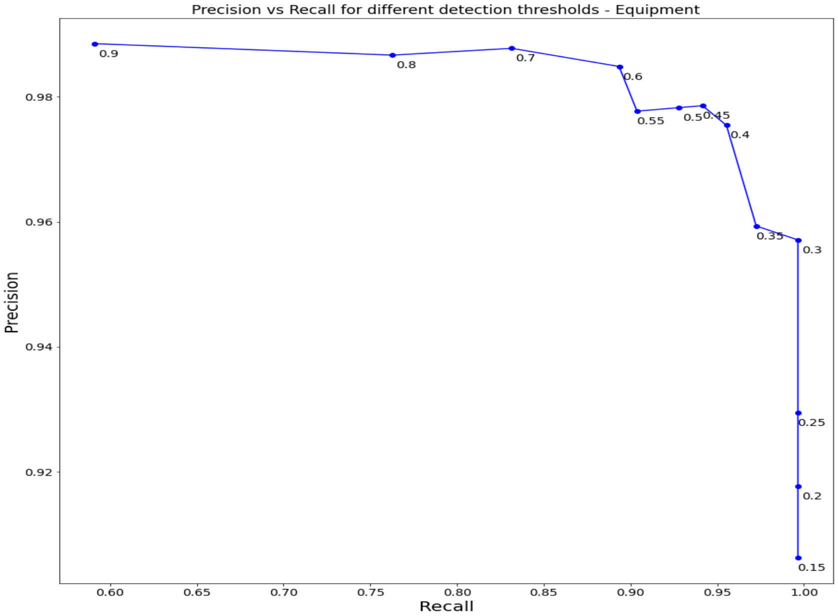


Figure 3.13: Precision-Recall Plot for different detection threshold - Equipment

Table 3.4: Precision-Recall for different detection thresholds - Equipment

Threshold	Precision	Recall
0.15	0.90	0.99
0.20	0.92	0.99
0.25	0.93	0.99
0.30	0.95	0.99
0.35	0.95	0.97
0.40	0.96	0.95
0.45	0.97	0.94
0.50	0.97	0.92
0.55	0.98	0.90
0.60	0.98	0.89
0.70	0.98	0.83
0.80	0.99	0.76
0.90	0.99	0.59

Table 3.5: Well Pad Detection Results over 5 test regions in DJ basin

	Well Pads	True Positive	False Positive	False Negative
Region-1	16	15/16(93.75%)	0/15(0.0%)	1/16(6.25%)
Region-2	11	10/11(90.90%)	1/11(9.09%)	1/11(0.0%)
Region-3	17	16/17(94.11%)	0/16(0.0%)	1/17(5.88%)
Region-4	13	13/13(100%)	1/14(7.1%)	0/13(0.0%)
Region-5	24	23/24(95.83%)	2/25(8%)	1/24(4.16%)
Average		77/81(94.91%)	4/81(4.93%)	4/91(4.39%)

Table 3.6: Equipment Detection Results over 5 test regions in DJ basin

	Equipment	True Positive	False Positive	False Negative
Region-1	163	151/163(92.63%)	2/153(1.30%)	12/163(7.36%)
Region-2	57	52/57(91.22%)	4/56(7.14%)	5/57(8.77%)
Region-3	98	91/98(92.85%)	7/98(7.14%)	7/98(7.14%)
Region-4	55	49/55(89.09%)	4/53(7.54%)	6/55(10.90%)
Region-5	137	127/137(92.70%)	9/136(6.61%)	10/137(7.29%)
Average		470/510(92.15%)	26/406(6.40%)	40/510(7.84%)

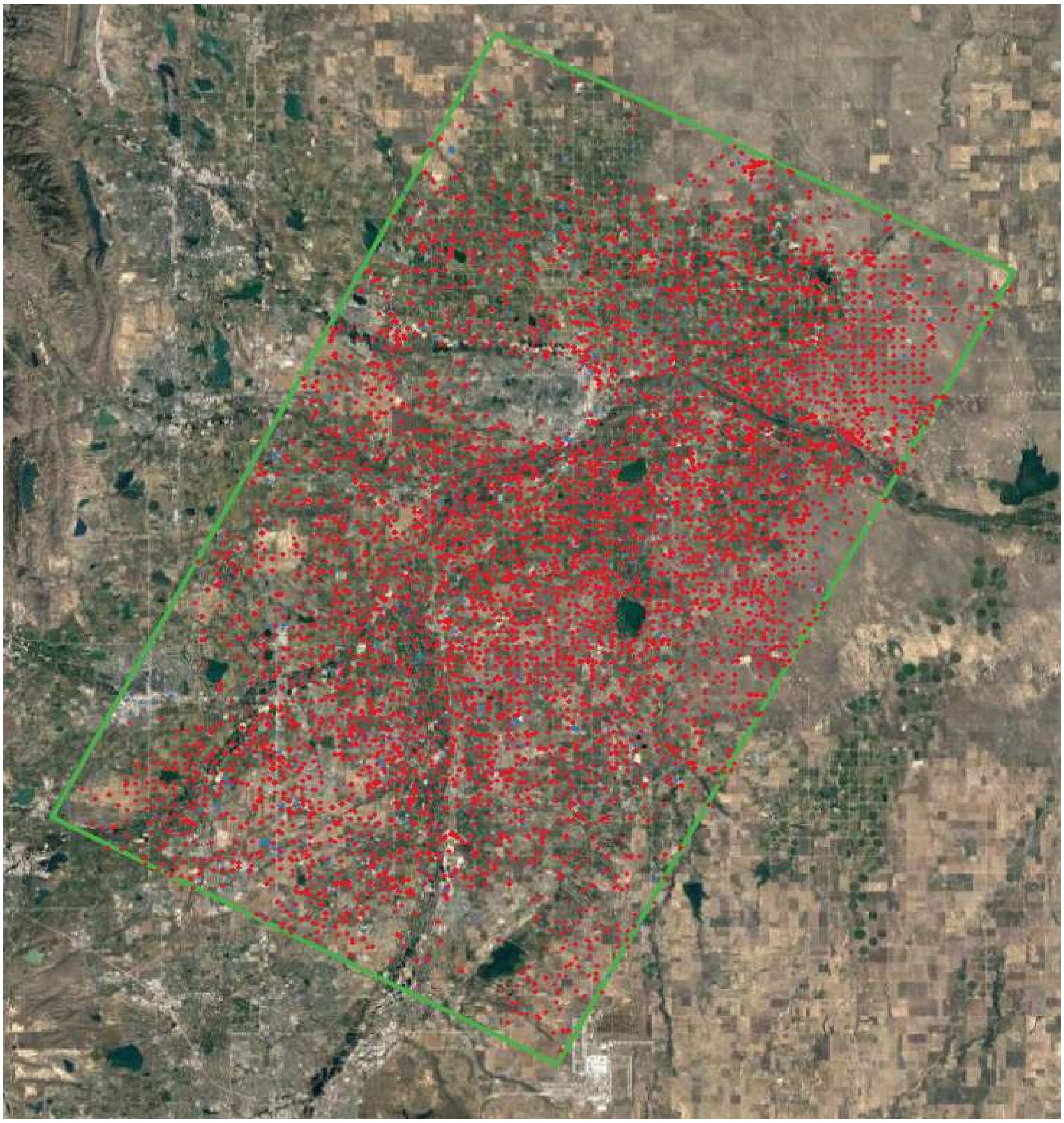
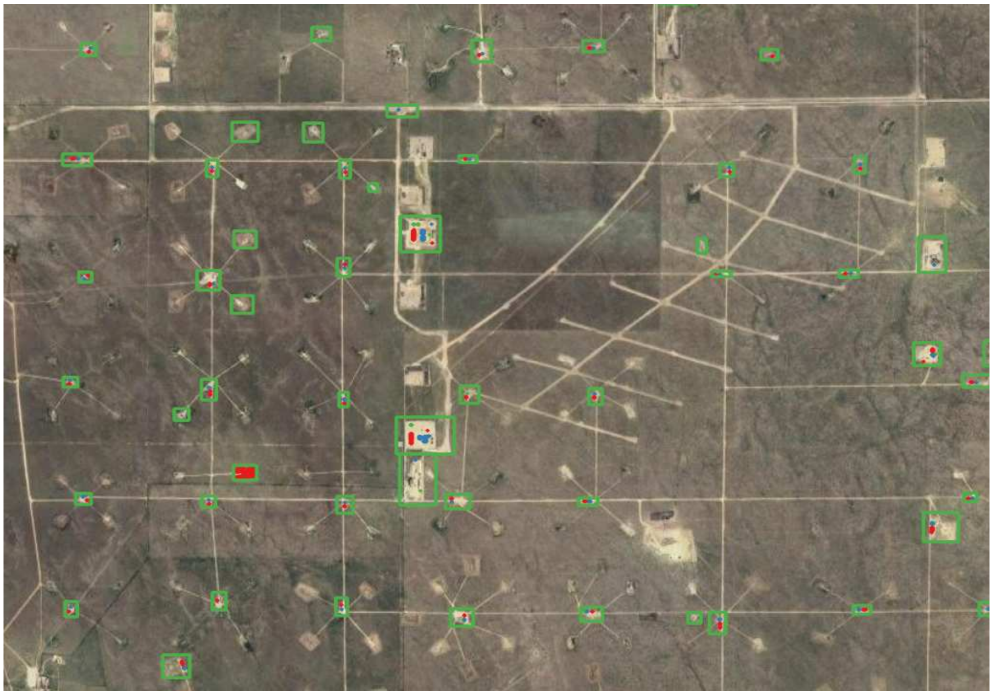


Figure 3.14: Well Pad detections across DJ Basin. DJ Basin is the green bounding box and red boxes are well pad detections[33]



(a) Well Pad Detection - 1

(b) Well Pad Detection - 2

Figure 3.15: Well Pad Detections[33]

(a) Equipment Detection - 1

(b) Equipment Detection - 2

Figure 3.16: Equipment Detections[33]

Figure 3.17: False Positives and True Negatives across DJ Basin[33]

Chapter 4

MARS VOLCANO DETECTION

The second part of my work is focused on building an inventory of small to midsize volcanoes on Mars to constrain and quantify the role of these edifices in martian volcanic processes, their morphological variations, and implications for compositional variations, thermal evolution, thermal budgets, and volatile interactions over space and time. Post-Viking Orbiter mission data revealed thousands of small (<50 km basal diameter) volcanic constructs—shields, cones, domes, and rootless volcanic cones—that were almost entirely invisible in Viking Orbiter images. These small martian volcanoes outnumber their larger siblings and have the potential to reveal valuable information about the thermal, chemical, and volatile evolution of Mars' interior through time. However, no comprehensive global survey of small and midsize Martian volcanoes exists to allow for global assessments and comparisons with more enormous volcanoes for regional or temporal comparisons. The large volcanoes are properly documented, but there is no proper documentation of the smaller volcanoes. Finding and analyzing the martian volcanoes and the associated heat is significant in studying the evolution of Mars and also in analyzing future habitable environments on Mars. There are datasets from different imaging instruments to study the surface of Mars, but a manual annotation of each of the small volcanoes will be a time-consuming task. There are already few works in the field of object detection on Mars using Deep learning. Deep learning has been used for detecting impact craters[55], rockfalls[56], and other landforms on Mars[57]. In this work, we focus on exploring the potential of deep learning methods to find the volcanoes on Mars. This work was done in collaboration with Dr.Susan Sakimoto and her team. This work was published at 52nd Lunar and Planetary Science Conference, 2021[2], and a detailed report of the work is presented in this thesis.

4.1 Data Collection

Java Mission-Planning and Analysis for Remote Sensing (JMARS)[58] is a software that can be used to view and analyze the remote sensing data from different sources. It was first developed to analyze the data of Mars from the THEMIS instrument. Later it was extended to study data from other planets such as Earth, Venus, Mercury, and moons of Jupiter and Saturn. JMARS also has additional data about MARS from different instruments such as Thermal Emission Spectrometer(TES), Infrared Thermal Mapper(IRTM), Mars Orbital Laser Altimeter(MOLA), Mars Orbiter Camera(MOC), etc. We collected images taken by THEMIS[59] and MOLA[60] using JMARS for training and testing a CNN model.

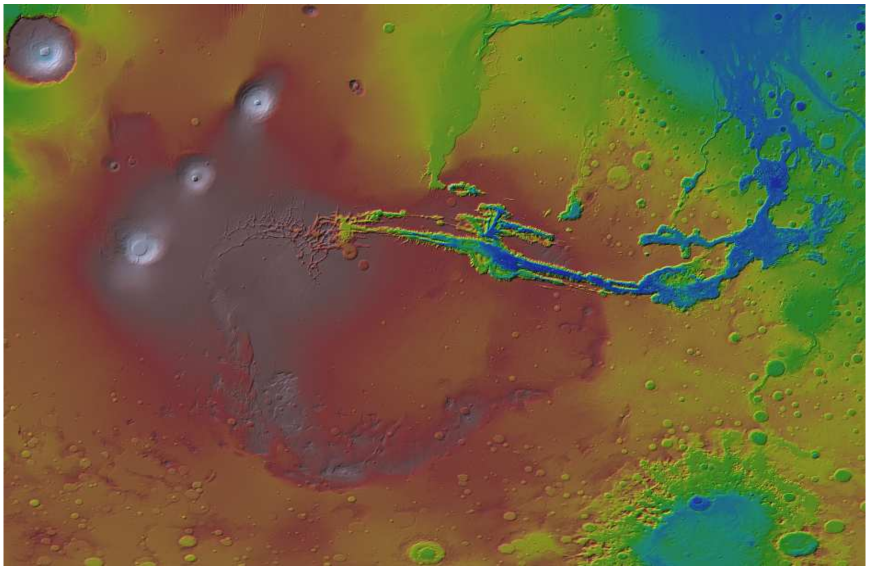


Figure 4.1: MOLA image showing a wider view of Mars

MOLA is an imaging instrument on the Mars Global Surveyor(MGS) launched on November 7, 1996, whose mission was to orbit Mars and map the surface of Mars over a 4.5 year time[61]. It collected data about the elevation of the surface of Mars until June 30, 2001. MOLA emitted infrared pulses to the surface of Mars at a rate of 10 times per second[63]. It then used the time of flight to determine the elevation of the surface of Mars. This helped the scientists to create a topographic map of Mars. We used colorized Elevation images from MOLA to train and test our

Figure 4.2: Image Taken by MOLA with small volcanoes

first image segmentation model. Figure 4.1 shows the image collected by MOLA, and Figure 4.2 shows the small volcanoes on Mars. The small holes on the surface are the openings of volcanoes. Since there is no readily available dataset of the volcanoes on Mars, we collected 10 MOLA Colorized Elevation images using JMARS. Each of the ten images contains multiple volcanoes, and we split seven images for training and the remaining 3 for testing. There were a total of 230 steep volcanic summits in the training set and 45 in the test. We also performed basic data augmentation techniques rotation and scaling to each image during training.

The second dataset used for finding volcanoes are images collected by the THEMIS instrument. THEMIS[59] was an imaging instrument on Mars Odyssey orbiter. It takes the image of the surface of Mars in the visible and infrared region of the electromagnetic spectrum. THEMIS produced high-resolution visual images at 18 meters per pixel and infrared images at 100 meters per pixel. For our task of volcano detection, we used THEMIS Day IR images. Figure 4.3 shows images collected from the THEMIS Day IR dataset, which we used for training our model. The small bumps in the image are the presence of volcanoes on the surface. We collected ten images from the THEMIS dataset and used 7 for training and the rest 3 for testing. We annotated a total of 48 steep summits in the training set and around 13 steep summits in the testing set. To increase the training size, we applied the same data augmentation technique used before.

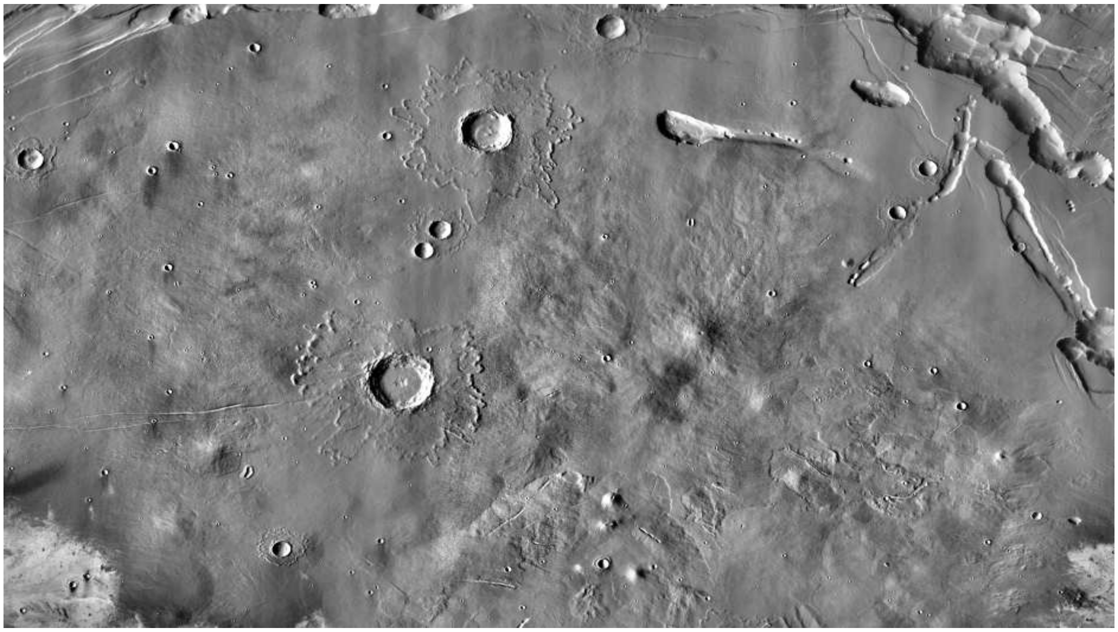


Figure 4.3: Image taken by THEMIS with small volcanoes

4.2 CNN Architecture

Modern ML techniques, using convolutional neural networks (CNNs), have been recently applied to the task of detecting features on Mars. Prior works have used images from different sources for detecting craters, ridges, and other landforms on Mars. Identifying and mapping volcanoes is difficult compared to impact craters or transverse aeolian ridges because of the extreme morphological variations observed in volcanoes.

We started our experiments using a Yolo-v4 object detection model. An object detection model tries to predict a rectangular bounding box around the object. But we observed that a bounding detection model like Yolo v4 did not perform well on volcano detection task. So we switched our focus to an image segmentation model. A segmentation model assign a label to each pixel in an image. We observed that an image segmentation model, Mask-RCNN fits well into the task of volcano detection on Mars.

Mask RCNN[63] is an instance segmentation model which finds the class, mask, and bounding box for an object of interest. Mask RCNN is an extension of Faster RCNN and uses ResNet-101 for feature extraction from images. Mask-RCNN operates in 2 stages - a region proposal network and a binary mask classifier. The region proposal network uses ResNet 101 to find multiple regions of interest(ROI) or bounding boxes where the probability of finding objects are high. These regions are then fed into the Binary Mask Classifier. The Binary mask classifier refines the predicted bounding box and predicts the object's class and creates a mask for each object. Figure 4.4 shows the basic architecture of a Mask-RCNN model.

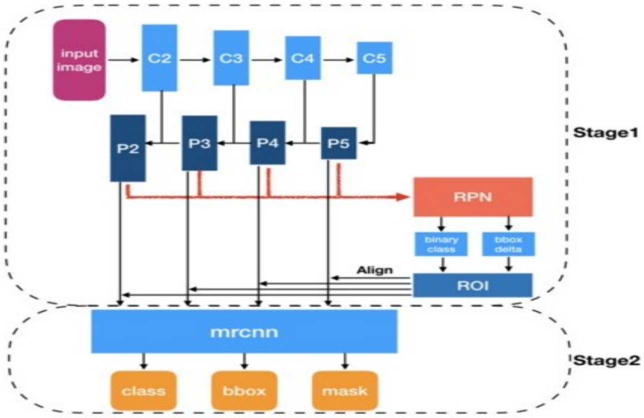


Figure 4.4: Mask-RCNN Architecture

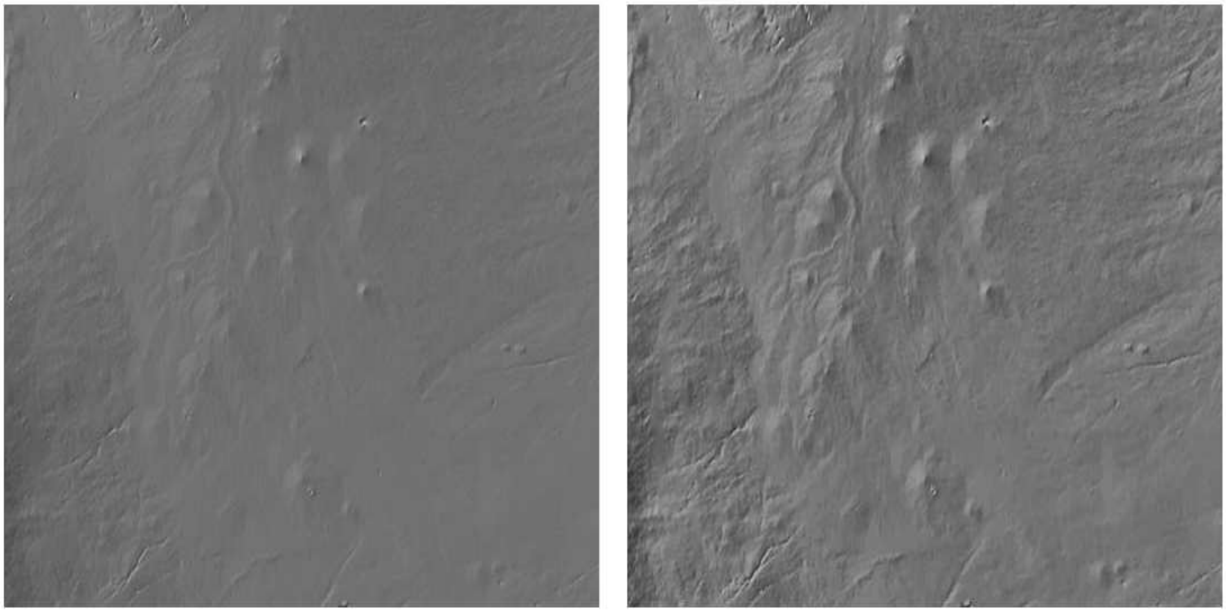
4.3 Training

Using JMARS Public 5.1.6 [13], we exported the MOLA (Mars Orbiter Laser Altimeter [14]) Colorized Elevation layer as JMARS JPEG gridded data at 128 pixels per degree (463 m/pixel at the equator) for Ten regional areas with small volcanoes in the Tempe Terra, Syria Planum, and Pavonis Mons SE areas [1, 2, 16-19]. After some preprocessing, the ten MOLA images were randomly split into seven training and three test images and annotated for steep volcanic summits

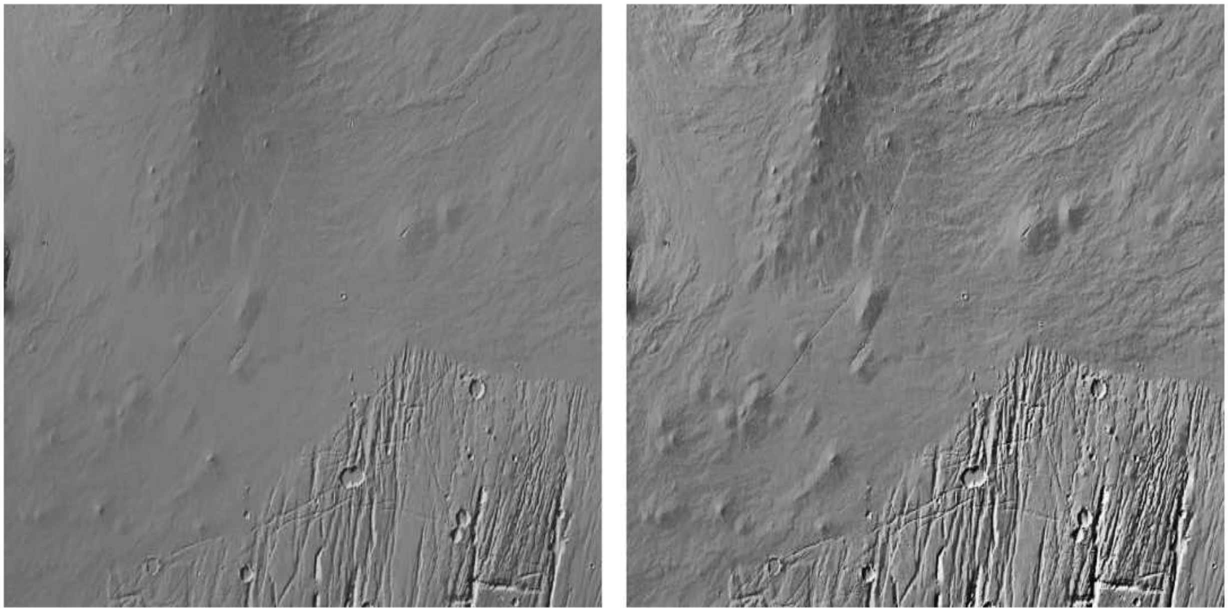
(230 in training and 45 in test). The Mask R-CNN network was trained on the seven training images and applied to the three test images with a detection probability threshold of 0.70.

We conducted a second test with the same methods but using THEMIS (Thermal Emission Imaging System [19]) daytime temperature data for eleven areas on the South and East lower flanks of Pavonis Mons volcano (JMARS exported JPEG gridded data at 256 pixels per degree, 231 m/pixel at the equator, e.g., Fig. 1B). For THEMIS data, seven training images had 48 steep summits annotated, and four test images had 13 steep summits. Mask RCNN uses a loss function that combines the classification loss, bounding box regression loss, and segmentation loss. We trained the model until the loss functions do not decrease any more.

Before training, we performed some pre-processing steps on both MOLA and THEMIS images, which improved the object detection results. First, we converted each image to a gray-scale image and then performed Histogram Equalization. Histogram equalization is a method of contrast adjustment of images using an image's histogram. It is hard for normal eyes to see the smaller volcanoes on the surface of Mars, but the contrast enhancement method helped to tackle this problem up to a specific limit. Histogram equalization increases the global contrast in an image by better distributing the pixel intensities on the histogram. This method helps to view the smaller volcanoes, which have a really low contrast. Instead of using regular histogram equalization, which requires fine-tuning, we used Contrast limited adaptive histogram equalization(CLAHE)[65] method. Unlike the histogram equalization methods, adaptive histogram methods[64] computes the histogram of several patches of image and use them to distribute the intensity. This helps to improve the local contrast and hence enhances edges. CLAHE is a modified adaptive histogram equalization technique that helps to avoid the amplification of noisy pixels. Figure 4.5-a and 4.5-b show the image before and after applying CLAHE. It can be seen that after CLAHE, the edges of the volcanoes become visible.



(a) MOLA - 1



(b) MOLA - 2

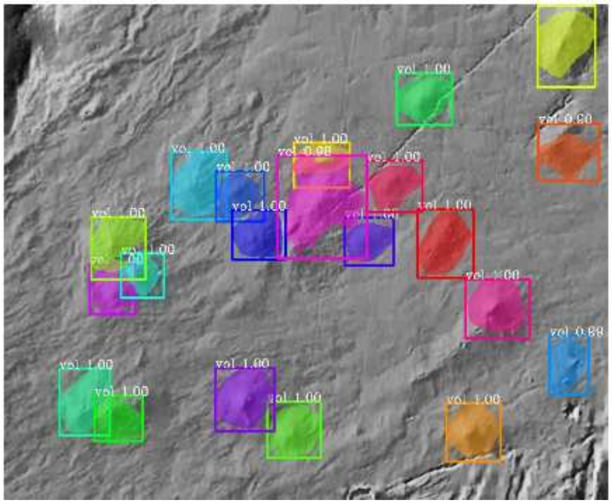
Figure 4.5: MOLA images before and after CLAHE

4.4 RESULTS - MARS VOLCANO DETECTION

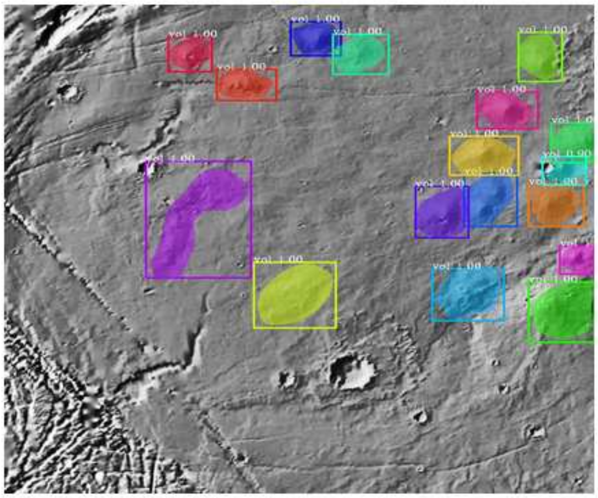
As is common practice, we started with a network with initial weights tuned to vision tasks using the COCO (Common Objects in Context) data set [66]. We then fine-tuned weights by training on the 7 MOLA images using stochastic gradient descent (SGD) for optimization and a mini-batch size of 4 images. The network was applied to the three MOLA test images using a detection probability threshold of 0.70. 43 out of the 45 steep summits were uniquely detected using a single predicted bounding box containing the summit at this threshold. The remaining two were both captured within a single predicted bounding box. An additional 17 bounding boxes contained no steep summit (false positives) and 12 of which were volcanic constructs. A second pilot study with the same methodology used 10 THEMIS images downloaded from South and East lower flanks of Pavonis Mons volcano (JPEG format, 256 pixels per degree, 231 m/pixel at the equator). For THEMIS data, the seven training images had 48 steep summits annotated, and the three test images had 35 edifices of which 13 had steep summits. At the 0.70 threshold, 12 of the 13 steep summits were detected There were 23 formal false positives and 12 of them were other volcanic constructs (shallow slope shields, lave tube ridges, rift vents, etc.).

Table 4.1: MOLA and THEMIS Volcano Detection Results

	Volcanoes	True Positive	False Positive	False Negative
MOLA	45	43/45(95.55%)	17/62(27.41%)	2/45(4.44%)
THEMIS	13	12/13(92.30%)	10/23(43.47%)	1/13(7.69%)

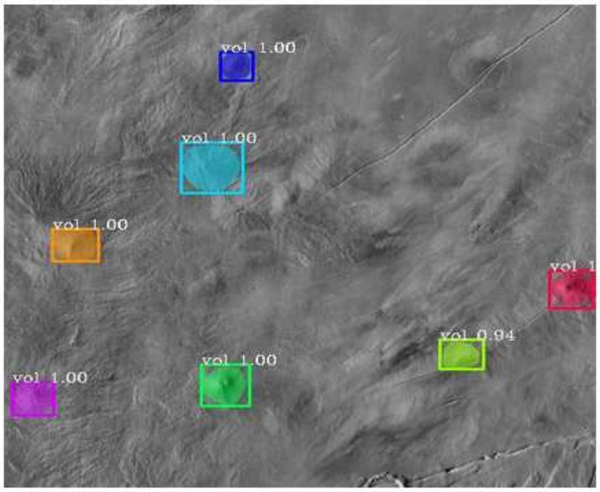


(a) MOLA - 1

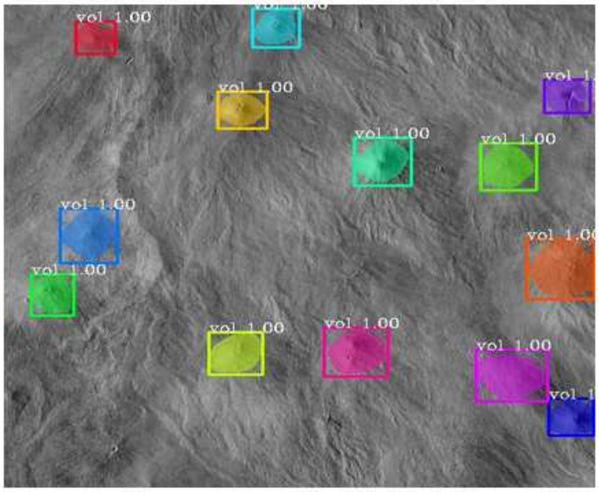


(b) MOLA - 2

Figure 4.6: Segmentation Results on MOLA dataset



(a) THEMIS - 1



(b) THEMIS - 2

Figure 4.7: Segmentation Results on THEMIS dataset

Chapter 5

CONCLUSION

This thesis presented ways to apply deep learning techniques for object detection from satellite images. I worked on two different topics on object detection from satellite imagery using deep learning. My first work was part of methane emission studies in Colorado by CSU Energy Institute. I applied deep learning methods to identify well pads and major site equipment from google satellite imagery to support their study. Our initial results from DJ Basin, Colorado, look promising and could support any future research in this field. The model managed to detect most of the well sites and equipment from satellite images. However, there were also a few false positive and false negative detections which are hard to avoid. Future work could focus on collecting high-resolution imageries and using advanced data augmentation methods. Using information about oil/gas pipelines or well locations to find potential well site areas can help to eliminate most of the false positive detections. The modeling technique utilized in this work could also be used for other applications, such as the detection of agricultural facilities (grain storage and feedlots are of interest), or other emissions sources (e.g., compressor stations, gas stations, or pipeline infrastructure).

My second work was focused on finding small to midsize volcanoes on Mars using deep learning. We used images collected by THEMIS and MOLA instruments for our work. The initial results for Martian volcano detection using deep learning looks impressive. Even though there are a lot of space for improvement, using deep learning methods for Martian volcano detection is a promising field.

In this thesis I explored two different deep learning models, a bounding box detection model and an image segmentation model. We saw how different deep learning models perform on different object detection task. Bounding Box detection model worked well for oi/gas site detection task, while an image segmentation model worked great for martian volcano detection task. Eventhough it is not clear why a model performs better on a particular task, choosing the right model is an important part of the whole setup.

One big advantage of our method is its ability to run online without downloading the whole dataset on a system. We implemented the model in a GIS software, QGIS and does not require downloading all the datasets. QGIS supports image from all major satellite imagery sources from Google, Bing, OSM, and Yahoo. Few disadvantages are that the satellite imagery should be available in a GIS server and also an active internet connection is required while running the model. But if storage spaces are available, our implementation could also be used offline.

Eventhough the model performed great on the two object detections tasks, there are a lot of spaces for improvement. More sophisticated data augmentation techniques could be used to simulate variations in sunlight, occlusion and other seasonal changes. These augmentation techniques could help in training better models. Also using classical image processing techniques along with deep learning models could result in a much better model. Currently, finding the best model and hyperparmeters is based on experiments. Future research could focus on finding the best models theoretically for each tasks.

My two works on applying deep learning to satellite imageries have shown what artificial intelligence(AI) is capable of. The results from both my works are undeniably incredible. It is pretty impressive how a machine learns to identify complex tasks like oil/gas site detection and volcano detection from few training images. Both tasks achieved a really high accuracy above 90%. The space of application of deep learning and artificial intelligence is infinite. I hope my work on deep learning for remote sensing will inspire future works in this field.

Bibliography

[1] Dileep, S., Zimmerle, D., Beveridge, R., & Timothy, V. (2020). Automated Identification of Oil Field Features using CNNs. NeurIPS 2020 Workshop Tackling Climate Change with Machine Learning. 2020 Conference on Neural Information Processing Systems (NeurIPS 2020).

[2] Sakimoto S.E.H., Lewis D., Dileep S., Memon P., Beveridge J.R., Blanchard N.T., Gregg T.K.P., "Deep Learning for an Inventory of Small to Midsize Volcanic Edifices on Mars", no. 2548, 2021.

[3] https://www.thegeospatial.in/earth-observation-satellites-in-space

[4] IPCC Core Writing Team, Pachauri R, Meyer L (2014) IPCC, 2014: Climate Change 2014: SynthesisReport. Contribution of Working Groups I. II and III to the Fifth Assessment Report of the intergovernmentalpanel on Climate Change. IPCC, Geneva, Switzerland, 151

[5] D. Allen, "Attributing Atmospheric Methane to Anthropogenic Emission Sources," Acc. Chem. Res., vol.49, no. 7, pp. 1344–1350, Jul. 2016, doi: 10.1021/acs.accounts.6b00081.

[6] R. A. Alvarez, S. W. Pacala, J. J. Winebrake, W. L. Chameides, and S. P. Hamburg, "Greater focus neededon methane leakage from natural gas infrastructure," PNAS, vol. 109, no. 17, pp. 6435–6440, Apr. 2012, doi:10.1073/pnas.1202407109.

[7] H. Schaefer et al., "A 21st-century shift from fossil-fuel to biogenic methane emissions indicated by 13CH4,"Science, vol. 352, no. 6281, pp. 80–84, Apr. 2016, doi: 10.1126/science.aad2705.

[8] S. Schwietzke, W. M. Griffin, H. S. Matthews, and L. M. P. Bruhwiler, "Natural Gas Fugitive EmissionsRates Constrained by Global Atmospheric Methane and Ethane," Environ. Sci. Technol., vol. 48, no. 14, pp.7714–7722, Jul. 2014, doi: 10.1021/es501204c.

[9] A. J. Turner et al., "A large increase in U.S. methane emissions over the past decade inferred from satellitedata and surface observations," Geophys. Res. Lett., vol. 43, no. 5, p. 2016GL067987, Mar. 2016, doi:10.1002/2016GL067987.

[10] Inventory of U.S. Greenhouse Gas Emissions and Sinks:1990-2018

[11] Allen, D. T.; Torres, V. M.; Thomas, J.; Sullivan, D. W.;Harrison, M.; Hendler, A.; Herndon, S. C.; Kolb,C. E.; Fraser, M. P.; Hill, A. D.; Lamb, B. K.; Miskimins, J.; Sawyer, R. E.; Seinfeld, J. H. Measurements ofmethane emissions at natural gas production sites in the United States. Proc. Natl. Acad. Sci. U. S. A. 2013,DOI: 10.1073/ pnas.1304880110

[12] D. T. Allen et al., "A Methane Emission Estimation Tool (MEET) for prediction of natural gas emissionswith high temporal and spatial resolution.," Environ. Sci. Technol., vol. in review, 2020.

[13] A. M. Robertson et al., "Variation in Methane Emission Rates from Well Pads in Four Oil and GasBasins with Contrasting Production Volumes and Compositions," Environ. Sci. Technol., Jun. 2017, doi:10.1021/acs.est.7b00571.

[14] S. Schwietzke et al., "Improved Mechanistic Understanding of Natural Gas Methane Emissions fromSpatially Resolved Aircraft Measurements," Environ. Sci. Technol., May 2017, doi: 10.1021/acs.est.7b01810.

[15] T. L. Vaughn et al., "Temporal variability largely explains top-down/bottom-up difference in methaneemission estimates from a natural gas production region," PNAS, p.201805687, Oct.2018, doi:10.1073/pnas.1805687115.

[16] D. J. Zimmerle et al., "Gathering pipeline methane emissions in Fayetteville shale pipelines and scopingguidelines for future pipeline measurement campaigns," Elem Sci Anth, vol. 5, no. 0, Nov. 2017, doi:10.1525/elementa.258.

[17] D. R. Lyon et al., "Constructing a Spatially Resolved Methane Emission Inventory for the Barnett ShaleRegion," Environ. Sci. Technol., vol. 49, no. 13, pp. 8147–8157, Jul. 2015, doi: 10.1021/es506359c.

[18] Baptistaet. al (2008) JGR, 113(E9).

[19] Bleacher,et. al(2009) JVGR, 185(1-2), 96-102.

[20] Mark Pritt, Gary Chern, "Satellite Image Classification with Deep Learning", 2020 [21] Van Etten, A., 2018. You only look twice: Rapid multi-scale object detection in satellite imagery. arXiv preprint arXiv:1805.09512.

[22] Shermeyer, J. and Van Etten, A., 2019. The effects of super-resolution on object detection

performance in satellite imagery. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (pp. 0-0).

[23] Sheng, H., Irvin, J., Munukutla, S., Zhang, S., Cross, C., Story, K., Rustowicz, R., Elsworth, C., Yang, Z., Omara, M. and Gautam, R., 2020. OGNet: Towards a Global Oil and Gas Infrastructure Database using Deep Learning on Remotely Sensed Imagery. arXiv preprint arXiv:2011.07227.

[24] Benedix G. K., Lagain A., Chai K., Meka S., Norman C., Anderson S., Bland P. A., Martian Impact Crater Database: Towards a Complete Dataset of d > 100 m and Automatic Identification of Secondary Craters Clusters

[25] S. H. Silverman, K. R. Blasius, S. J. Ferry and P. R. Christensen, "Thermal emission imaging system (THEMIS) for Mars 2001 using an uncooled microbolometer array," 1999 IEEE Aerospace Conference. Proceedings (Cat. No.99TH8403), 1999, pp. 377-389 vol.3, doi: 10.1109.

[26] III, J. , Malin, M. , Caplinger, M. , Fahle, J. , Wolff, M. , Cantor, B. , James, P. , Ghaemi, F.(2013). Calibration and performance of the Mars Reconnaissance Orbiter Context Camera (CTX). Mars. 8. 1–14. 10.1555/mars.2013.0001.

[27] Bickel, V.T., Conway, S.J., Tesson, P.A., Manconi, A., Loew, S. and Mall, U., 2020. Deep learning-driven detection and mapping of rockfalls on mars. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, 13, pp.2831-2841.

[28] Palafox, L.F., Hamilton, C.W., Scheidt, S.P. and Alvarez, A.M., 2017. Automated detection of geological landforms on Mars using Convolutional Neural Networks. Computers & geosciences, 101, pp.48-56.

[29] R. A. Alvarez, D. Zavala-Araiza, D. R. Lyon, D. T. Allen, Z. R. Barkley, A. R. Brandt, K. J. Davis, S. C.Herndon, D. J. Jacob, A. Karion,et al., “Assessment of methane emissions from the us oil and gas supplychain,”Science, vol. 361, no. 6398, pp. 186–188, 2018.

[30] “United states environmental protection agency, global methane initiative.”https://www.epa.gov/gmi

[31] Y. Zhang, R. Gautam, S. Pandey, M. Omara, J. D. Maasakkers, P. Sadavarte, D. Lyon, H. Nesser, M. P.Sulprizio, D. J. Varon,et al., “Quantifying methane emissions from the largest oil-

producing basin in theunited states from space,"Science Advances, vol. 6, no. 17, p. eaaz5120, 2020.

[32] T. R. Scarpelli, D. J. Jacob, J. D. Maasakkers, M. P. Sulprizio, J.-X. Sheng, K. Rose, L. Romeo, J. R.Worden, and G. Janssens-Maenhout, "A global gridded (0.1×0.1) inventory of methane emissions fromoil, gas, and coal exploitation based on national reports to the united nations framework convention onclimate change.,"Earth System Science Data, vol. 12, no. 1, 2020

[33] Google Earth Imagery(c) 2020

[34] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," inProceedings of the IEEE conference on computer vision and pattern recognition, pp. 3431–3440, 2015.

[35] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and AlexanderC Berg. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision(ECCV), pages 21–37, 2016

[36] 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, pp. 91–99, 2015.

[37] Diganta Misra.Mish:A self regularized non-monotonic neural activation function.arXiv pre printarXiv:1908.08681, 2019.

[38] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic mod-els. In Proceedings of International Conference on Ma-chine Learning (ICML), volume 30, page 3, 2013.

[39] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines.In Proceedings of International Conference on Machine Learning (ICML),pages 807–814, 2010

[40] Jiahui Yu, Yuning Jiang, Zhangyang Wang, Zhimin Cao,and Thomas Huang. UnitBox: An advanced object detection network. In Proceedings of the 24th ACM international conference on Multimedia, pages 516–520, 2016

[41] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, RongguangYe, and Dongwei Ren. Distance-IoU Loss: Faster and better learning for bounding box regression. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI),2020.

[42] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, AmirSadeghian, Ian Reid, and Silvio Savarese. Generalized in-tersection over union: A metric and a loss for bounding box regression. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pages658–666, 2019.

[43] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detec- tion.arXiv preprint arXiv:2004.10934, 2020.

[44] Redmon, J., Farhadi, A.: YOLO9000: better, faster, stronger. CoRR abs/1612.08242 (2016)

[45] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-timeobject de- tection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pages 779– 788, 2016.

[46] Joseph Redmonand, Ali Farhadi.YOLOv3:An incremental improvement

[47] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 580–587, 2014.

[48] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and AlexanderC Berg. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision(ECCV), pages 21–37, 2016

[49] D.Rolnick,P.L.Donti,L.H.Kaack,K.Kochanski,A.Lacoste,K.Sankaran,A.S.Ross,N.Milojevic, N.Jaques, A. Waldman-Brown, et al., “Tackling climate change with machine learning,” arXiv preprintarXiv:1906.05433, 2019.

[50] Chien-Yao Wang, Hong-Yuan Mark Liao, Yueh-Hua Wu,Ping-Yang Chen, Jun-Wei Hsieh, and I-Hau Yeh. CSPNet:A new backbone that can enhance learning capability ofcnn.Proceedings of the IEEE Conference on Computer Vi-sion and Pattern Recognition Workshop (CVPR Workshop),2020. 2, 7

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.Spatial pyramid pooling in deep convolutional networks forvisual recognition.IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 37(9):1904–1916,2015. 2, 4, 7

[52] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia.Path aggregation network for instance segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 8759–8768, 2018.1, 2, 7

[53]C. Goutte and E. Gaussier. A probabilistic interpretation of precision, recall, and F-score, with implicationfor evaluation. In ECIR '05: Proceedings of the 27th European Conference on Information Retrieval, pages345–359, 2005

[54] L. Torrey and J. Shavlik.2009. Transfer learning. In E. Soria, J. Martin,R. Magdalena, M. Martinez, and A.Serrano, editors,Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods,and Techniques. IGI Global

[55] Baptista, Ana Rita, Nicolas Mangold, Véronique Ansan, David Baratoux, Philippe Lognonné, Eduardo I. Alves, David A. Williams, Jacob E. Bleacher, Philippe Masson, and Gerhard Neukum. "A swarm of small shield volcanoes on Syria Planum, Mars." Journal of Geophysical Research: Planets 113, no. E9 (2008).

[56] Bickel,et. al(2020), IEEE J.Sel.TopicsAppl.Earth Obs.Remote Sens., 13, 2831-2841.

[57] Wagstaff, Kiri, You Lu, Alice Stanboli, Kevin Grimes, Thamme Gowda, and Jordan Padams. "Deep Mars: CNN classification of mars imagery for the PDS imaging atlas." In Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1. 2018.

[58] Christensen, Philip R., Joshua L. Bandfield, James F. Bell III, Noel Gorelick, Victoria E. Hamilton, Anton Ivanov, Bruce M. Jakosky et al. "Morphology and composition of the surface of Mars: Mars Odyssey THEMIS results." science 300, no. 5628 (2003): 2056-2061.

[59] Christensen, P.R., et al.,(2004) THEMIS Public Data Releases, PDS Geosciences node, ASU,

[60] Smith, D. E., et al. (2001). JGR, 106(E10), 23689–23722.

[61] https://attic.gsfc.nasa.gov/mola/

[62] [https://tharsis.gsfc.nasa.gov/MOLA/index.php]

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

[64] Pizer, S.M., Amburn, E.P., Austin, J.D., Cromartie, R., Geselowitz, A., Greer, T., ter Haar Romeny, B., Zimmerman, J.B. and Zuiderveld, K., 1987. Adaptive histogram equalization and its variations. Computer vision, graphics, and image processing, 39(3), pp.355-368.

[65] Zuiderveld, K., 1994. Contrast limited adaptive histogram equalization. Graphics gems, pp.474-485.

[66] Lin, Tsung-Yi et al. "Microsoft COCO: Common Objects in Context." ECCV (2014).

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

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

[69] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun.Deep residual learning for image recognition. InProceedings of the IEEE Conference on Computer Vision and Pat-tern Recognition (CVPR), pages 770–778, 2016

[70] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kil-ian Q Weinberger. Densely connected convolutional net-works. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4700–4708, 2017.

[71] Andrew Howard, Mark Sandler, Grace Chu, Liang-ChiehChen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu,Ruoming Pang, Vijay Vasudevan, et al. Searching for MobileNetV3. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2019. 2, 4

[72] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. MobileNets: Efficient convolutional neural networks for mobile vision applications.arXiv preprint arXiv:1704.04861, 2017. 2, 4

[73] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4510–4520, 2018

[74] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan,Mark Sandler, Andrew Howard, and Quoc V Le. MNAS-net: Platform-aware neural architecture search for mobile.In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2820–2828, 2019

[75] Forrest N Iandola, Song Han, Matthew W Moskewicz,Khalid Ashraf, William J Dally, and Kurt Keutzer.SqueezeNet: AlexNet level accuracy with 50x fewer parameters and¡ 0.5 MB model size.arXiv preprintarXiv:1602.07360, 2016

[76] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qing-ming Huang, and Qi Tian. CenterNet: Keypoint triplets for object detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), pages 6569–6578,2019.

[77] Ross Girshick. Fast R-CNN. InProceedings of the IEEE In-ternational Conference on Computer Vision (ICCV), pages1440–1448, 2015.

[78] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun.Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in Neural InformationProcessing Systems (NIPS), pages 91–99, 2015

[79] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN:Object detection via region-based fully convolutional net-works. InAdvances in Neural Information Processing Sys-tems (NIPS), pages 379–387, 2016.

[80] Tsung-Yi Lin, Piotr Doll ar, Ross Girshick, Kaiming He,Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR), pages 2117–2125, 2017

[81] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia.Path aggregation network for instance segmentation. InProceedings of the IEEE Conference on Computer Visionand Pattern Recognition (CVPR), pages 8759–8768, 2018

[82] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficient-Det: Scalable and efficient object detection. InProceedingsof the IEEE Conference on Computer Vision and PatternRecognition (CVPR), 2020.

[83] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. InCVPR, 2014

[84] P. O. Pinheiro, R. Collobert, and P. Dollar. Learning to segment object candidates. InNIPS, 2015

[85] P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Doll ar. Learning to refine object segments. In ECCV, 2016.

[86] J. Dai, K. He, Y. Li, S. Ren, and J. Sun. Instance sensitive fully convolutional networks. In ECCV, 2016

[87] J. Dai, K. He, and J. Sun. Instance-aware semantic segmentation via multitask network cascades. In CVPR, 2016

[88] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei. Fully convolutional instance-aware semantic segmentation. In CVPR, 2017

[89] Pritt, M., and Chern, G. (2017, October). Satellite image classification with deep learning. In 2017 IEEE Applied Imagery Pattern Recognition Workshop (AIPR) (pp. 1-7). IEEE.