



NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF

Reserve volume estimation of oil storage tanks based on remote sensing images

DIPLOMA THESIS

of

ATHANASIOS PANTOS

Supervisor: Konstantinos Karantzas
Associate Professor

Athens, July 2022



NATIONAL TECHNICAL UNIVERSITY OF ATHENS
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING
DIVISION OF

Reserve volume estimation of oil storage tanks based on remote sensing images

DIPLOMA THESIS
of
ATHANASIOS PANTOS

Supervisor: Konstantinos Karantzas
Associate Professor

Approved by the examination committee on 27th July 2022.

(Signature)

(Signature)

(Signature)

.....
Konstantinos Karantzas
Associate Professor

.....
Lazaros Grammatikopoulos
Associate Professor

.....
Emmanuel Stamatakis
Assistant Researcher

Athens, July 2022



Copyright © - All rights reserved.

Athanasios Pantos, 2022.

The copying, storage and distribution of this diploma thesis, exall or part of it, is prohibited for commercial purposes. Reprinting, storage and distribution for non - profit, educational or of a research nature is allowed, provided that the source is indicated and that this message is retained.

The content of this thesis does not necessarily reflect the views of the Department, the Supervisor, or the committee that approved it.

DISCLAIMER ON ACADEMIC ETHICS AND INTELLECTUAL PROPERTY RIGHTS

Being fully aware of the implications of copyright laws, I expressly state that this diploma thesis, as well as the electronic files and source codes developed or modified in the course of this thesis, are solely the product of my personal work and do not infringe any rights of intellectual property, personality and personal data of third parties, do not contain work / contributions of third parties for which the permission of the authors / beneficiaries is required and are not a product of partial or complete plagiarism, while the sources used are limited to the bibliographic references only and meet the rules of scientific citing. The points where I have used ideas, text, files and / or sources of other authors are clearly mentioned in the text with the appropriate citation and the relevant complete reference is included in the bibliographic references section. I fully, individually and personally undertake all legal and administrative consequences that may arise in the event that it is proven, in the course of time, that this thesis or part of it does not belong to me because it is a product of plagiarism.

(Signature)

.....

Athanasios Pantos

27th July 2022

Abstract

This work aims to derive oil tank volumes from high-resolution satellite data in order to satisfy the growing demand for accurate measurements of oil tank volumes around the world. Using Otsu thresholding, shadow area thresholding, and morphological closing, the shadow of the oil tank is retrieved from a remote sensing HSV image. The shadow of the oil tank is in the shape of a crescent. The height and the radius of the tank are used in the calculation of the total volume of the tank. In order to assess the viability of the suggested method, an optical remote sensing image with a high resolution was obtained from the Motoroil refinery located in Agioi Theodoroi, Greece.

On the test set, the object detection algorithm known as yolov3 was able to get an AP score of 0.84, while on the train set it earned a score of 0.942. In addition, the amount of crude oil barrels that were projected to be contained in the floating head tanks that were discovered was compared to the actual number of tanks. According to the findings, the method that was suggested is effective at calculating the capacity of oil storage tanks to a high degree of precision and with an accuracy that is appropriate for use in actual applications.

Keywords

high-resolution remote sensing image; shadow extraction; shadow length calculation; tank volume calculation; object detection; yolov3

to my father, my mother, my sister and to my friends

Table of Contents

Abstract	1
1 Introduction	8
1.1 Crude oil	9
1.2 How's crude oil stored	9
1.2.1 Fixed Roof Storage	9
1.2.2 Floating Roof Tanks	10
1.3 Problem Statement & Methodology	11
2 Background and Related Work	14
2.1 Machine Learning	14
2.1.1 Supervised Learning	14
2.1.2 Unsupervised Learning	15
2.1.3 Reinforcement Learning	15
2.2 Artificial Neural Networks	16
2.2.1 Backpropagation	18
2.2.2 Convolutional Neural Networks	18
2.2.3 Convolutional Layer	19
2.2.4 Pooling Layer	20
2.3 YOLOv3	21
2.3.1 Architecture	22
2.3.2 Anchor Boxes	23
2.3.3 Yolov3 Tiny Architecture	24
2.3.4 Related Work	24
2.4 Metrics	26
2.4.1 Accuracy	26
2.4.2 F_1 Score	27
2.4.3 Intersection over Union (IoU)	27
2.4.4 Yolo Loss	31
3 Methodology	32
3.1 Research Questions	32
3.2 Dataset	32
3.3 Data Augmentation	34
3.3.1 What is augmentation of data?	34

3.3.2	Why is it critical now?	34
3.3.3	What are the advantages of augmented data?	34
3.3.4	What are the difficulties associated with data augmentation?	36
3.4	Training	37
3.5	Calculating Volume via Shadow Extraction	38
3.5.1	An overview of available methods	38
3.5.2	Detecting shadows in the HSV space	39
3.5.3	Proposed method for shadow extraction	39
3.5.4	Calculating Volume and Number of Barrels	44
4	Software engineering	45
4.1	Back-end	45
4.2	Front-end	45
4.3	The app	46
5	Experimental Evaluation	48
5.1	Motor Oil's Refinery	48
5.1.1	The Refinery	48
5.1.2	Characteristics of the Refinery	49
5.1.3	Refinery Units	49
5.2	Results of the application	51
5.3	Conclusion	52
	Bibliography	54
	List of Abbreviations	55

List of Figures

1.1	Fixed roof tanks	10
1.2	A Floating Roof Tank	11
2.1	A Simple Artificial Neural Network	16
2.2	Structure of a single perceptron or neuron	17
2.3	Multi-layer Artificial Neural Network	18
2.4	Filters of a Neural Network	19
2.5	Pooling Layer	20
2.6	Example of 2x2 Max-pooling	20
2.7	Image Splitting and Bounding-boxes Prediction	21
2.8	Image Splitting and Bounding-boxes Prediction	22
2.9	IoU	27
2.10	Identification of TP, FP and FN through IoU thesholding.	28
2.11	This is a detection made using Mask R-CNN. It shows a bonding box(dotted), segmentation mask, class (fruit) and the confidence score(0.999). Object detection model typically outputs the bounding box, confidence score and clas. Confidence value is the confidence of the model on the detection and it ranges between 0 and 1.	29
2.12	Definition of Average Precision	29
2.13	A model detecting objects of the same class. There are 12 detections and 9 ground-truths.	30
2.14	Yolo loss function	31
3.1	Annotated Image Example	33
3.2	Annotated Image Example	33
3.3	Image Augmentation	36
3.4	Transfer Learning	37
3.5	Tensorboard Training	38
3.6	Enhance Shadow Portions	40
3.7	Filtered enhanced Image	40
3.8	Thresholded image	41
3.9	More Filteration	41
3.10	Highlighted Tank Contours	42
3.11	Two Contours of a tank side by side	43
3.12	Two Contours of a tank	43

4.1	Browse through the files you want to upload	46
4.2	YoloV3 is running in the back and detecting the floating head tanks	47
4.3	Number of floating head tanks and barels of oil are returned	47
5.1	Motor Oil refinery from space	48
5.2	A floating head tank detected among regular tanks	51
5.3	A FHT that appears to be full is correctly captured by the algorithm	51
5.4	The algorithm missed a floating head tank but correctly captured the other two present and successfully estimated the volume	52

List of Tables

3.1	Conversion factors for liquid (oil, condensate and NGL)	44
-----	---	----

Chapter **1**

Introduction

Sputnik 1, the world's first artificial satellite, was launched by the Soviet Union on October 4, 1957. Approximately 8,900 satellites [1] from more than 40 countries have been launched since then. These satellites assist us in a variety of ways, including surveillance, communication, and navigation. These countries also employ satellites to track another country's land and movements in order to assess its economy and power. All countries, on the other hand, keep their information hidden from one another.

In the same way, the global oil market isn't entirely transparent. Almost every oil-producing country tries to conceal its total output, consumption, and storage. Countries do so to keep their true economy hidden from the rest of the world and to strengthen their national defense systems. Other countries may be threatened by such an approach. As a result, numerous start-ups, such as planet and orbital insight, monitor similar actions in multiple countries via satellite photos [2]. Satellite photos of oil storage tanks and estimated reserves were gathered.

However, how can the volume of oil storage tanks be estimated just from satellite images? Because oil is frequently stored in floating head tanks, tank capacity estimation is achievable. This tank type features a head that lies directly on top of the crude oil, preventing odors from accumulating. As a result, the tank head's height fluctuates in proportion to the amount of oil in the tank. The volume of the tank can be estimated using the relative sizes of the outer shadow cast by the tank and the interior shadow cast by the height of the tank head. Our issue description covers two tasks: detecting floating roof tanks and extracting shadows and estimating the volume of recognized tanks. The first task utilizes target detection technology, whereas the second task utilizes computer vision technology. Let us now discuss how to do each assignment..

Tank inspection: The purpose of this exercise is to determine the volume of the floating roof tank. We can develop a target detection model for a single class, but to avoid confusion between one model and another (i.e., various types of oil storage tanks) and to make the model more robust, we suggest three categories of target detection models. For target detection, Yolov3 with transfer learning is employed because it is easier to train on the machine. Additionally, the process of data enhancement is used to improve the measurement score. Shadow extraction entails the use of a variety of computer vision methods. Due to the insensitivity of RGB color schemes to shadows, they must first be transformed to HSV [3] and lab color spaces. The threshold image is then processed

morphologically (i.e. removing noise, clear contour, etc.).

1.1 Crude oil

Petroleum, often known as crude oil or oil [4], is a naturally occurring yellowish-black liquid found deep beneath the Earth's surface in geological formations. It is frequently processed into a variety of different sorts of fuels. Petroleum components are separated via a process called fractional distillation, which involves the distillation of a liquid mixture into fractions with varying boiling points, often using a fractional column. It is composed of naturally occurring hydrocarbons with a range of molecular weights and may include other chemical molecules. Petroleum is a generic term that refers to both naturally occurring unprocessed crude oil and petroleum products composed of refined crude oil. Petroleum is a fossil fuel that is generated when enormous amounts of dead animals, primarily zooplankton and algae, are buried beneath sedimentary rock and exposed to extreme heat and pressure.

Petroleum has been extracted mostly through oil drilling. Drilling occurs following structural geological research, sedimentary basin study, and reservoir description. Recent technological advancements have enabled the extraction of additional unconventional resources such as oil sands and oil shale. Once extracted, oil is refined and separated, most simply through distillation, into a variety of products for immediate consumption or manufacturing, ranging from gasoline (petrol), diesel, and kerosene to asphalt and chemical reagents used to manufacture plastics, insecticides, and pharmaceuticals. Petroleum is used in the manufacture of a wide variety of materials and the globe is projected to utilize approximately 100 million barrels (16 million cubic metres) of petroleum every day. Petroleum production can be enormously profitable and was critical to economic progress in the twentieth century, with certain countries gaining significant economic and international power as a result of their control of oil production.

1.2 How's crude oil stored

Oil, petroleum refining, and the petrochemical industries use storage tanks extensively. There are various types of storage tanks, including those with a fixed roof, an open roof, an exterior and an internal floating roof, etc. The roof of a floating roof tank floats directly on top of the product, avoiding the potential of an ignitable environment and eliminating the need for a vapor space. The two most often utilized crude oil storage tanks are the fixed roof tank and the floating roof tank.

1.2.1 Fixed Roof Storage

These are the most frequently used oil storage containers. During storage, hydrocarbons such as liquids, volatile organic compounds, hazardous air pollutants, and some inert gases evaporate and collect between the liquid level and the fixed roof tanks.

As the liquid level in the tank changes, the gases are gradually released into the atmosphere. Easily avoid this by installing vapor recovery units. Another option is to utilize foam chambers. These protect flammable hydrocarbon or water-miscible liquids by using low expansion foam or extinguishing agents.

The foam fills the space previously occupied by air. These devices, according to experts, eliminate hazards by directing all foam directly onto the flammable liquid surface regardless of the weather conditions.



Figure 1.1. *Fixed roof tanks*

1.2.2 Floating Roof Tanks

Floating roof tanks are preferable to fixed roof tanks because they help to prevent vapor emissions. According to experts, the chances of a floating roof tank catching fire or exploding internally are also reduced.

Additionally, they state that these crude oil tanks are the best option for storing stable liquids with near-zero dynamic loads. However, adverse environmental conditions can have an adverse effect on floating roofs, as an accumulation of snow and rainwater can submerge the roof in the stored liquid.

A potential source of concern with floating roof oil tanks is the dynamic loads imposed on the roof by the constant splashing of water, which can result in roof compartments flooding. This can be remedied, however, by providing adequate stiffness in the circumferential direction near the roof.

Additionally, as the liquid exits the tank, the floating liquid gradually sinks, leaving behind liquid droplets. As a result, liquid droplets evaporate in the atmosphere.



Figure 1.2. *A Floating Roof Tank*

1.3 Problem Statement & Methodology

Oil powers the entire world; it controls the global economy; every commodity price is dependent on oil because commodities must be transported, and the majority of modes of transportation rely on oil; this is why every country attempts to keep their oil production secret for a variety of reasons. Oil is stocked in storage tanks, and data on oil production and consumption are opaque. Various countries that are the largest oil producers constantly attempt to fix oil prices to meet their needs; at the moment, the price of a barrel of oil is at an all-time high, and as a result, various countries are constantly in conflict and in a state of war. Oil does not merely fuel various nations; it also creates numerous tensions between nations, which is why they never want to share all of the information about this resource. That is why various companies such as Planet and Orbital Insight have begun collecting satellite images and estimating the location of oil storage tanks. Floating head tanks aid in estimating the volume of oil tanks; when oil comes into direct contact with air, it fumes, preventing a head from sitting on top of the storage tanks. We can calculate the volume of the tank by calculating the relative inner and exterior

shadows.

This problem can be divided into three stages.

- Object Detection : Detect whether the image is floating-head tank.
- Shadow Extraction: Extract the image of the floating-head tank.
- Volume Calculation: Calculate the volume using the extracted shadow.

Volume is estimated in some research papers by calculating the volume of the cylinder, by calculating the height of the storage tanks using the shadow, and by calculating the radius of the storage tanks using the Hough Transform, but in some later papers, volume is calculated using the interior and exterior shadow lengths.

The term "object recognition" refers to a collection of closely related computer vision tasks that involve recognizing objects in digital photographs. Classification of images entails predicting the class of an individual object contained within an image. The term "object localization" refers to the process of locating and drawing a bounding box around one or more objects within an image. Object detection is a combination of these two tasks that enables the localization and classification of one or more objects within an image.

When a user or practitioner says "object recognition," they frequently mean "object detection." The term object recognition will be used broadly to refer to both image classification (the task of determining which object classes are present in an image) and object detection (the task of localizing all objects present in an image).

As such, we can group these three computer vision tasks into the following categories:

- Classification of Images: Ascertain the type or class of an object contained within an image. A single-object image, such as a photograph, as an input. As a result, a label for the class is generated (e.g. one or more integers that are mapped to class labels).
- Object Localization: Using a bounding box, determine the presence and location of objects in an image. As an input, use an image containing one or more objects, such as a photograph. As output, one or more bounding boxes (e.g. defined by a point, width, and height).
- Object Detection: Using a bounding box and the types or classes of the detected objects, determine the presence of objects in an image. As an input, use an image containing one or more objects, such as a photograph. Output: One or more bounding boxes (e.g. defined by a point, width, and height), each labeled with a unique class. Another extension of this decomposition of computer vision tasks is object segmentation, also called "object instance segmentation" or "semantic segmentation," in which instances of recognized objects are indicated by highlighting their specific pixels rather than by using a coarse bounding box.

As can be seen from this breakdown, object recognition refers to a collection of difficult computer vision tasks. Our objective is to determine the volume of occupied space in

floating head tanks. We could create an object detection model for a single class, but to avoid confusion with other types of tanks (i.e. Tank/Fixed head tank and Tank Cluster), as well as to make the model more robust, we created a three-class object detection model. YoloV3 with transfer learning is used for object detection because it is simple to train on non-specialized machines. Additionally, Data Augmentation is used to boost the metric score.

Chapter **2**

Background and Related Work

This chapter provides the theoretical background necessary to comprehend the methods discussed in Chapter 3. After discussing machine learning, neural networks, and computer vision in detail, the YOLOv3 and Tiny-YOLOv3 were explained. At the conclusion of this chapter, related work by other researchers in this field of study is also discussed.

2.1 Machine Learning

Machine learning [5] is one of the applications of Artificial Intelligence (AI) which enables the computers to learn on their own and perform tasks without human intervention. There are numerous applications of machine learning algorithms in the field of computer vision. With the help of machine learning, formulation of some of the most complex problems have been performed easily. Various computer programs which were previously programmed by humans, sometimes by-hand, are now being programmed without any human contribution with the help of machine learning. In the recent years, due to remarkable increase in the availability of humongous sources of data and feasibility of computational resources, machine learning has become predominant with wide range of applications in our daily lives.

Types

These are the three types of machine learning:

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

2.1.1 Supervised Learning

Supervised learning is the most fundamental type of machine learning algorithm. These algorithms, as their name implies, require direct supervision. In this type of learning, the algorithm is fed data that has been labeled/annotated by humans. This data contains information about the objects of interest's classes and locations. After the training process is complete, the algorithm learns from the annotated data and predicts the

annotations of new data that was previously unknown to the algorithm. Several popular supervised learning algorithms include the following:

- Neural Networks
- Decision Trees
- Random Forest
- K-Nearest Neighbors
- Linear Regression
- Logistic Regression
- Support Vector Machines

2.1.2 Unsupervised Learning

Unsupervised learning [6] is a subset of machine learning that identifies patterns in untagged data. The hope is that by forcing the machine to build a compact internal representation of its world and then generate imaginative content from it via mimicry, which is a critical mode of learning in humans. In contrast to supervised learning, which requires an expert to label data, such as "ball" or "fish," unsupervised methods exhibit self-organization, capturing patterns as probability densities or a combination of neural feature preferences [1]. The other two levels of supervision are reinforcement learning, in which the machine is guided solely by a numerical performance score, and semi-supervised learning, in which only a subset of the data is tagged. Unsupervised Learning employs two broad methods: Neural Networks and Probabilistic Methods. Unsupervised learning algorithms such as the apriori algorithm and K-means clustering are quite common.

2.1.3 Reinforcement Learning

Reinforcement learning (RL) [7] is a subfield of machine learning that studies how intelligent agents should behave in order to maximize the concept of cumulative reward. Along with supervised and unsupervised learning, reinforcement learning is one of the three fundamental machine learning paradigms.

Reinforcement learning differs from supervised learning in that it does not require the presentation of labeled input/output pairs and does not require the explicit correction of suboptimal actions. Rather than that, the emphasis is on striking a balance between exploration (of previously unexplored territory) and exploitation (of current knowledge). Partially supervised reinforcement learning algorithms can combine the advantages of supervised and unsupervised learning.

Because many reinforcement learning algorithms for this context employ dynamic programming techniques, the environment is typically expressed as a Markov decision process (MDP).

The primary distinction between classical dynamic programming methods and reinforcement learning algorithms is that the latter do not require knowledge of an exact mathematical model of the MDP and are therefore applicable to large MDPs for which exact methods are impractical.

2.2 Artificial Neural Networks

Artificial neural networks are a frequently used class of supervised learning models. This thesis is primarily concerned with a subclass of neural networks known as convolutional neural networks (CNNs). This model is referred to as 'Artificial Neural Networks' because it was developed to mimic the neural function of the human brain. An artificial neural network is made up of a collection of neurons that are connected to one another and grouped into layers to mimic the neural function of the human brain. Similar to neurons in the human brain, neurons in an artificial neural network serve as calculation units (see Figure 2.1). Synaptic connections between neurons are referred to as 'synapses,' and they are nothing more than weighted values. Thus, when an input value is applied to a neuron (x_1, x_2, \dots, x_n), it traverses the synapse by multiplying its value by the synapse's weighted value (w_1, w_2, \dots, w_n), as illustrated in Figure 2.1.

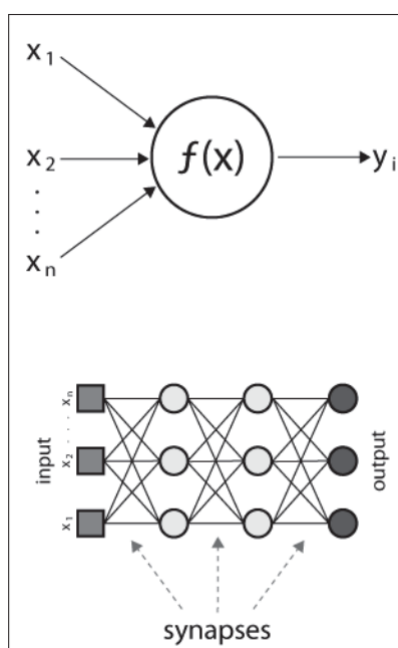


Figure 2.1. A Simple Artificial Neural Network

The bias 'b' is then added to the sum of these values to give us the final result. This will be the result of the neuron's activity. In order for a neuron to function properly, it must be equipped with a mapping mechanism that maps the inputs to the outputs. This mapping mechanism is referred to as the "Activation function". In a fully connected feed-forward multi-layer network, all of the outputs from one layer of neurons are fed as inputs to each and every neuron in the following layer of neurons. As a result, some

layers are responsible for processing the original input data, while others are responsible for processing the data obtained from neurons in the previous layer (see Figure 2.3). The number of weights assigned to any neuron in the network is therefore equal to the number of neurons in the layer immediately preceding the layer in which the neuron in question is located.

$$y = \sum_{i=1}^n (w_n * x_n) + b$$

In the above equation, "x" represents the input value given to the neuron, "w" represents the weighted value of the synaptic synapse, "n" represents the number of neurons in the network, "b" represents the bias, and "y" represents the output of the network. As a result, according to equation (2.1), the value of output 'y' is equal to the summation of the product of the values of 'x' with their corresponding weights and bias 'b' and the sum of the product of the values of 'x' with their corresponding weights and bias 'b'. A multi-layered artificial neural network, such as the one depicted in Figure 2.3, is composed of three types of layers: an input layer, one or more hidden layers, and an output layer, to name a few examples. The input layer is typically responsible for simply passing data along without altering it. The hidden layers are where the majority of the computation takes place. This layer converts the hidden layer activation to an output, such as a classification, by converting it to a classification. In each hidden layer, the outputs are used as the inputs for the next hidden layer, and so on. The number of neurons in the output layer is the same as the number of classes that were used to train the neural network in the first place.

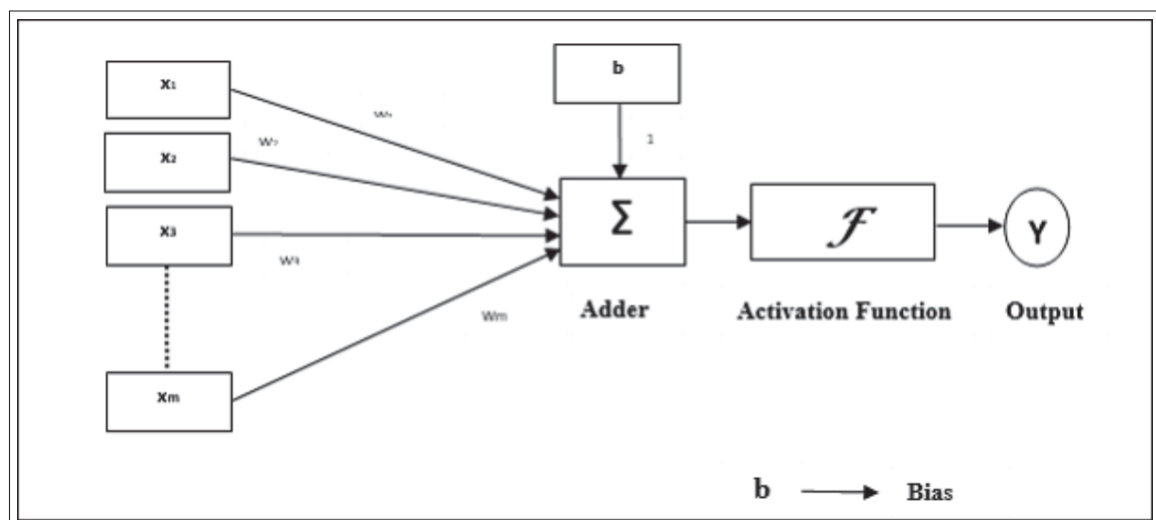


Figure 2.2. Structure of a single perceptron or neuron

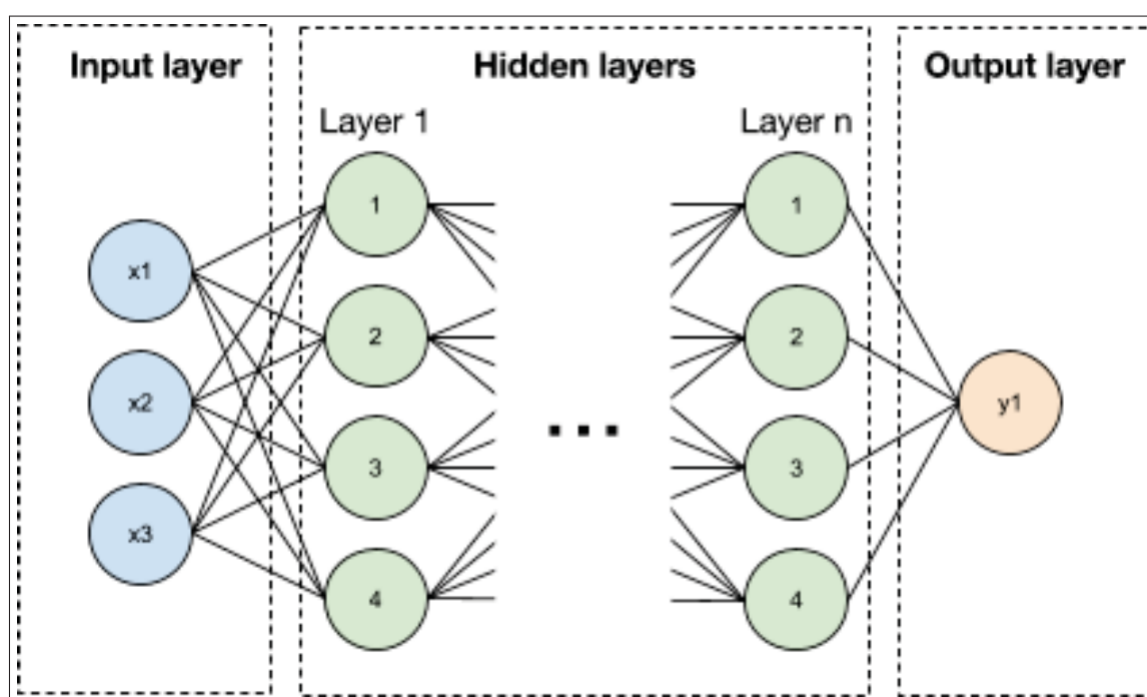


Figure 2.3. Multi-layer Artificial Neural Network

2.2.1 Backpropagation

Despite the fact that artificial neural networks have demonstrated widespread application in a variety of fields and have assisted in the achievement of ground-breaking innovations in recent years, the concept of neural networks is not new. The neural networks, formerly known as 'perceptrons,' have been in use since the 1940s and have become increasingly sophisticated. They were not as popular as they are now due to the fact that they were single-layered and required a large amount of computational power and data, both of which were difficult to come by at the time of their development. They have gained prominence primarily as a result of the invention of a technique known as 'Backpropagation.' Rumelhart and colleagues introduced the technique for the first time in the year 1986 . When the output of a network differs from the expected output, networks can use this technique to rearrange the weights of hidden layers to compensate. To ensure that the weights are adjusted appropriately, the error is calculated and backpropagated through all layers of the network to ensure that they are adjusted appropriately.

2.2.2 Convolutional Neural Networks

A variety of artificial neural networks are considered to be very important, including the Radial basis function neural network, the Feed-forward neural network, the Convolutional neural network, the Recurrent neural network, and the Modular neural network, among others. Convolutional neural networks (CNNs) are the most effective of these types of networks in applications such as image/video recognition [34], semantic parsing, natural language processing, and paraphrase detection [35]. A convolutional neural network

is composed of three layers, which are the Convolutional layer, the Pooling layer, and the Fully-connected layer, in most cases.

2.2.3 Convolutional Layer

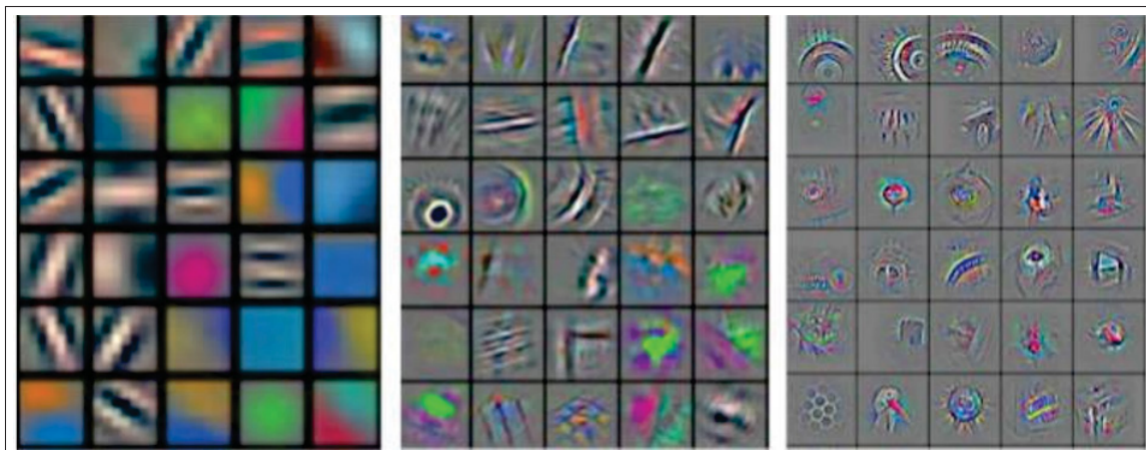


Figure 2.4. *Filters of a Neural Network*

A convolutional neural network is made up of one or more convolutional layers, which are connected together. These layers can either be pooled or fully connected, depending on the situation. A convolutional layer is typically used to perform tasks that require a lot of computation. It is made up of a collection of filters that have the ability to learn over time. Despite the fact that the filters are small in size, they are capable of filtering the entire depth of the input. Generally, the filter's dimensions are represented by the formula $l \times w \times d$, where 'l' denotes the height of the filter's length, 'w' denotes the width, and 'd' denotes the depth of the feature filter, which is equal to the number of color channels present. In general, the convolution process is carried out by a feature filter after it has been slid onto the input layer of the neural network, with the result being the generation of a feature map. A convolutional layer is the layer that is responsible for carrying out the convolution process. As a result, convolutional neural networks are referred to as networks that are composed of convolutional layers rather than simply neural networks. As depicted in Figure 2.4, the filter searches for specific patterns in the input layer during the initial stages of the search process. As part of the algorithm's training process, the filter searches for patterns for the purpose of learning to recognize them, which eventually transforms into a search to validate the existence of a specific pattern during the testing stages. In reality, there are numerous feature filters that can be trained to recognize various patterns.

2.2.4 Pooling Layer

In addition, pooling layers play an important role in the operation of a convolutional neural network. The primary function of a pooling layer is to reduce the number of parameters and computations present in the network by gradually and continuously shrinking the spatial size of the network's elements. This action is required in order to reduce the number of features that the filter has learned and to remove the requirement for the location of their location. There are numerous advantages to using a pooling layer, including the reduction of over-fitting, which is a state that occurs when the algorithm fits the data very closely by exhibiting low bias and high variance, among other things.

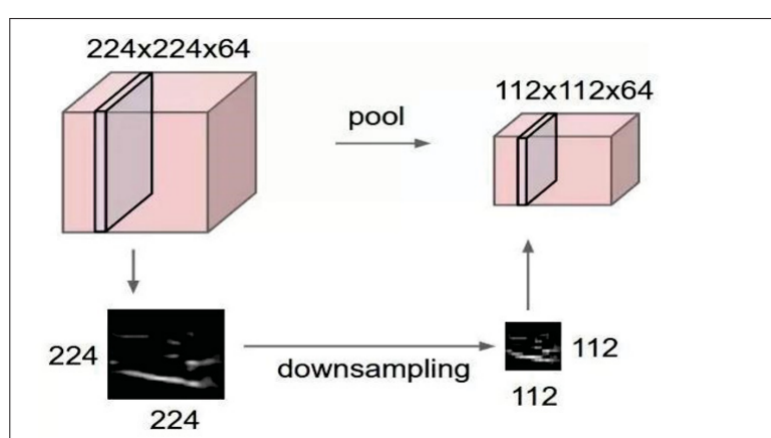


Figure 2.5. Pooling Layer

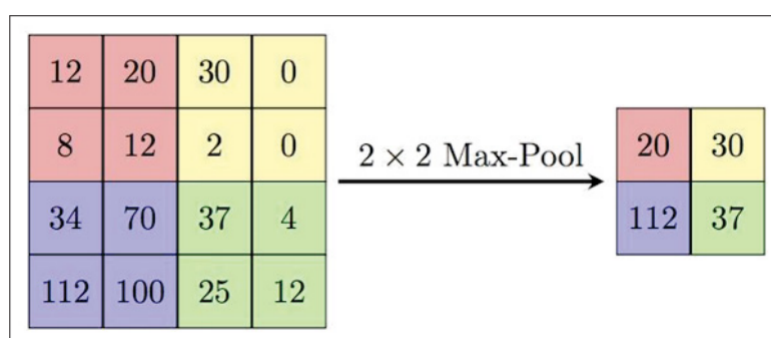


Figure 2.6. Example of 2x2 Max-pooling

Despite the fact that there are numerous types of pooling, maximum pooling is one of the most widely used in practice. This type of pooling is particularly convenient because it down-samples the layer while maintaining the depth. Pooling layers are depicted in Figure 2.5, and 2x2 map pooling is illustrated in Figure 2.6.

2.3 YOLOv3

The cutting-edge object detector YOLOv3 [8] is optimized for high accuracy and real-time performance. YOLOv3 is an enhancement to the previous YOLO version. It employs a single neural network to predict the position and class score of an object in a single iteration. This is accomplished by treating object detection as a regression problem, which transforms the input images into their class probabilities and positions. YOLO results in Numerous $S \times S$ grids are predicted from the input image and boundary boxes B , which include height, width, box center x and y coordinates. Each of these boxes has its own P (object probability) value, predicts the number of classes contained within it as C , and contains a conditional class probability P_{class} in the $S \times S$ containing an object. The network's overall prediction is $S \times S \times (B \times 5 + C)$, where the digit 5 represents each box coordinate as 4 and 1 represents the object probability.

During the test, the network uses the equation to determine the number of classes present in each grid (2.2). P_{min} is defined at the start of the test, and the system detects only objects with a P_{class} greater than or equal to P_{min} . Using non-maximal suppression, the duplicate detection of the same object is omitted during the post-processing stage.

$P(class_i) = P(class_i | object)$. $P(object) = P(class_i | object) \cdot P(object)$ (2.2) $P(class_i)$ denotes the probability of the i th class. $P(Object)$ is the probability that the grid contains the object, and $P(class_i | object)$ is the conditional class probability that the object is present in the i th class.

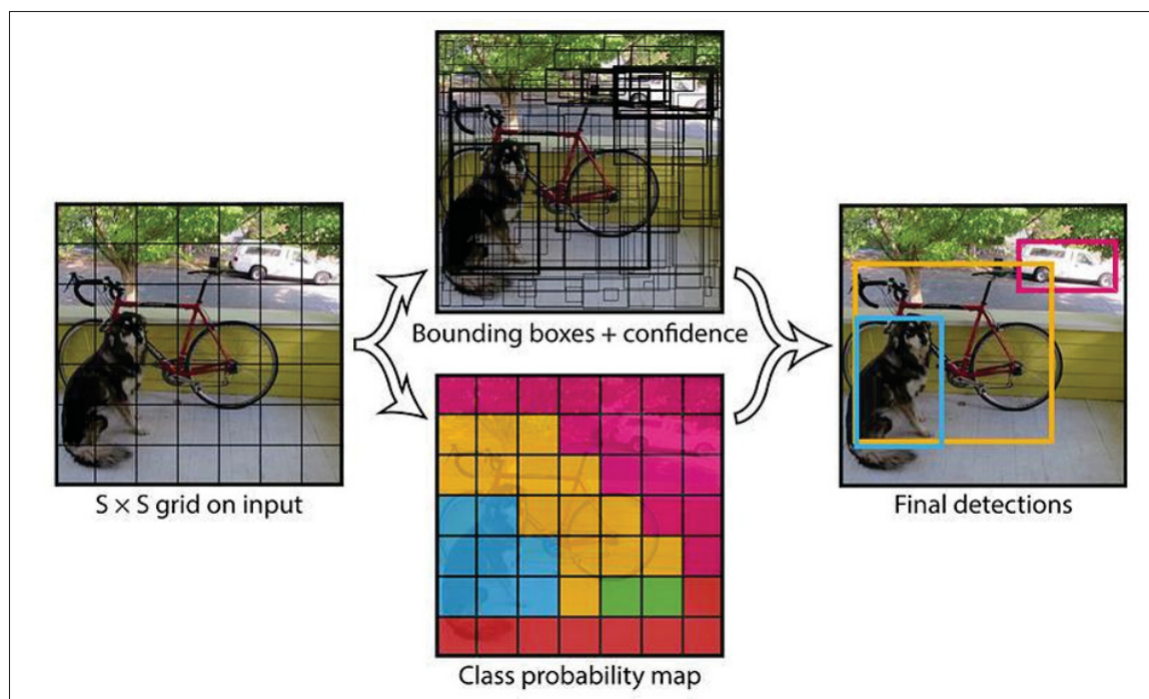


Figure 2.7. Image Splitting and Bounding-boxes Prediction

$$P(class_i) = P(class_i | object) * P(object)$$

In YOLO, only the bounding boxes with the greatest value of confidence are selected

since every grid-cell is predicting multiple bounding boxes. Therefore, YOLO generates a tensor as an output whose value is equal to $S \times S \times (B + C)$ [8]. In YOLOv3, the bounding boxes have been replaced by ‘Anchors’ which resolve the unstable gradient issue that used to occur while training of the algorithm. Therefore, YOLOv3 predicts outputs with confidence scores by generating a vector of bounding boxes whenever an input is given to the algorithm in the form of an image or a video.

2.3.1 Architecture

YOLOv2 made use of the Darknet-19 feature extractor, which consisted of 19 convolutional layers. The more recent version of this algorithm, YOLOv3, employs a new feature extractor called Darknet-53, which employs 53 convolutional layers, while the overall algorithm employs 75 convolutional layers and 31 other layers for a total of 106 layers. To facilitate downsampling, pooling layers have been removed from the architecture and replaced with another convolutional layer with stride ‘2’. This critical change was made to avoid feature loss during the pooling process. The architecture of the YOLOv3 algorithm is depicted in detail in Figure 2.8, which was created by ‘CyberailAB’.

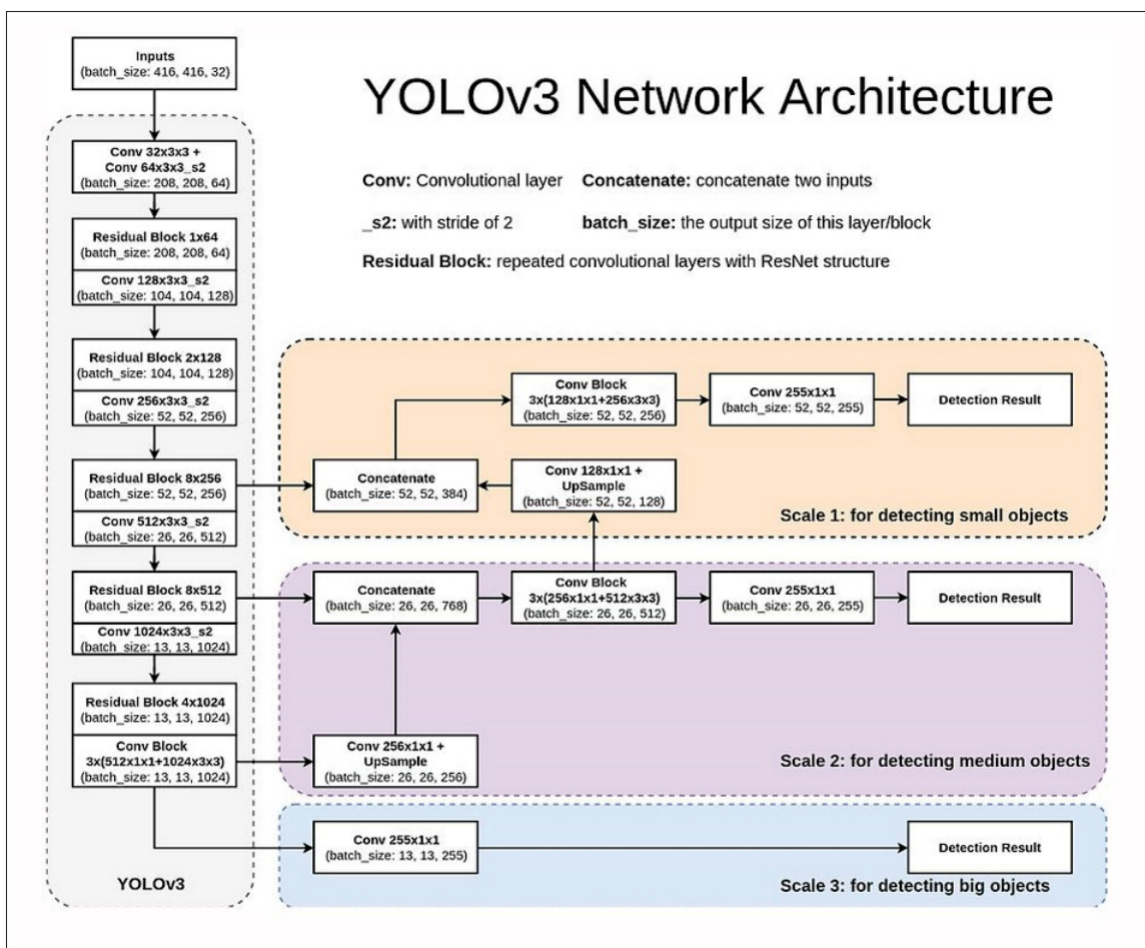


Figure 2.8. Image Splitting and Bounding-boxes Prediction

YOLOv3 detects objects at three different scales, as illustrated in Figure 2.8. On

feature maps of three distinct sizes located in three distinct locations throughout the network, 1×1 detection kernels are applied. The detection kernel has the shape $1 \times 1 \times (B(4 + 1 + C))$, where 'B' is the number of bounding boxes that a cell on the feature map can predict, '4' is the number of bounding box attributes, '1' is the object confidence, and 'C' is the number of classes. Figure 2.7 illustrates the splitting of an image and bounding-box prediction in YOLOv3, while Figure 2.8 illustrates the architecture of the YOLOv3 algorithm trained on the COCO dataset, which contains 80 classes and has three bounding boxes. As a result, the kernel would have a size of $1 \times 1 \times 255$. YOLOv3 down samples the dimensions of the input image by 32, 16 and 8 to make predictions at scales 3, 2, and 1. The input image in Figure 2.8 is 416×416 pixels in size. As mentioned previously, YOLOv3 has a total of 106 layers. As illustrated in Figure 2.8, the network down samples the input image for the first 81 layers. Because the 81st layer has a stride of 32, the 82nd layer performs the initial detection using a 13×13 feature map. Due to the use of a 1×1 kernel for detection, the resulting detection feature map is $13 \times 13 \times 255$ in size, which is sufficient for object detection at scale 3. After applying a few convolutional layers, the feature map from the 79th layer is up sampled by 2x, resulting in the dimensions 26×26 . This is then concatenated with the 61st layer's feature map. The features are fused by applying a few more 1×1 convolutional layers to the concatenated feature map. As a result, the 94th layer performs the second detection using a $26 \times 26 \times 255$ feature map, which is responsible for object detection at scale 2. Following the second detection, the feature map from the 91st layer is up sampled by 2x and convolutional layers are applied, resulting in the dimensions 52×52 . This is then concatenated with the 36th layer's feature map. The features are fused by applying a few more 1×1 convolutional layers to the concatenated feature map. As a result, the 106th layer performs the final detection using a $52 \times 52 \times 255$ feature map that is responsible for detecting objects at scale 1. As a result, YOLOv3 is more adept at detecting smaller objects than YOLOv2 and YOLO.

2.3.2 Anchor Boxes

Region proposals are computationally nearly costless when shared convolutional layers are used. The additional benefit of computing region proposals on a CNN is that it is GPU-capable. Traditional methods for calculating the return on investment, such as Selective Search, are implemented on a CPU. To deal with the detection window's varying shapes and sizes, the method employs special anchor boxes rather than a pyramid of scaled images or a pyramid of different filter sizes. The anchor boxes serve as points of reference for various region proposals centered on the same pixel.

A mechanism referred to as anchor boxes is used to generate a number of bounding boxes in this case. Anchors are simply the pixels that make up the feature image. For each anchor, nine boxes of varying shapes and sizes are generated with the anchor as their center. This layer is a data source for the classification (detection) and bounding box regression layers. Non-Maximum Suppression (NMS) is used to reduce the number of boxes by removing those that overlap with other boxes with a high probability of con-

taining an object. Following that, the probability scores are normalized using the SoftMax function. The resulting bounding boxes (ROIs) are combined with the output of the Feature Network and fed into the Detection Network. Given the variable size of the resulting feature maps, a ROI pooling layer is introduced to crop and scale the features to 14 x 14. These features are then aggregated to a maximum of 7 x 7 and fed into the Detection Network in batches. Figure 2.10 illustrates the pair of stacked common fully connected layers, as well as the classification layer and bounding box regression layer. This network generates the final class and its bounding box.

2.3.3 Yolov3 Tiny Architecture

Tiny-YOLOv3 is a condensed and simplified version of the original YOLOv3. In spite of the fact that the number of layers in Tiny-YOLOv3 is significantly less than that of YOLOv3, when high frame rates are taken into consideration, the accuracy of the model is almost as good as that of its larger counterpart. Tiny-YOLOv3 is composed of only 13 convolutional layers and 8 max-pool layers, and as a result, it requires only a small amount of memory to operate, significantly less than the layers in YOLOv3. There is a significant difference between YOLOv3 and TinyYOLOv3 in that the former is designed to detect objects at three different scales, whereas the latter is only capable of detecting objects at two distinct scales. With the exception of these distinctions, the operation of both variants is nearly identical.

YOLOv3 has many more convolutional layers than Tiny-YOLOv3, but the number of convolutional layers in Tiny-YOLOv3 is significantly lower. Tiny-YOLOv3 has only 13 convolutional layers in its primary structure, whereas the total number of layers in the overall structure is 23. Tiny-YOLOv3 extracts its features from a small number of 1 x 1 and 3 x 3 kernels, which are divided into three groups of three. For down sampling, the Tiny-YOLOv3 employs the pooling layer, as opposed to the YOLOv3, which employs convolutional layers from stride 2 for this purpose. TinyYOLOv3's convolutional layer structure is similar to that of YOLOv3's convolutional layer structure.

2.3.4 Related Work

Yukui Luo et al. presented an OpenCL implementation of the Deep Convolutional Neural Network, one of the most advanced frameworks for deep learning. Their framework made three significant contributions: a real-time object recognition system, a framework with low power consumption that can be used on portable devices, and a framework that can be used on a variety of compute devices. The framework's performance was compared to that of the CUDA framework using the YOLO V2 benchmark. Alpaydin proposed an adaptive fuzzy network topology to be used in conjunction with Deep Convolutional Neural Networks in order to achieve highly efficient object recognition for long range images with low contrast and variable, noisy backgrounds. Daniel et al. proposed the 'VoxNet' 3D Convolutional Neural Network architecture for accurate and efficient object detection using LiDAR data and RGBD point clouds. They evaluated their approach against publicly available state-of-the-art benchmarks and discovered that it outperformed these

benchmarks in terms of accuracy while classifying objects in real-time. Lewis proposed a do-it-yourself network called SimpleNet in his paper that performs deep object recognition without the need for pre-processing or expensive deep evaluations. Though the accuracy is significantly lower than the state-of-the-art, SimpleNet derives its power from appropriate loss functions with a finite number of parameters, whereas other networks derive their power from the layer depth. The author compared various CNN models such as OverFeat, VGG16, Fast R-CNN, and YOLO to SimpleNet in order to provide the audience with a comprehensive understanding of all these CNN models' performance.

Girshick et al. introduced the 'Fast R-CNN,' a region-based convolutional neural network. This network is capable of high-precision object detection at the expense of computational speed. As a result, the network is deemed unsuitable for real-time object detection and recognition, despite its high accuracy.

Ren et al. presented an updated version of 'Fast R-CNN' in their paper, dubbed 'Faster R-CNN'. As the name implies, this is an updated version of the Region-based Convolutional Neural Network, which demonstrated increased computational speed and accuracy in comparison to its predecessor and many other state-of-the-art networks. A Region Proposal Network (RPN) has been added to the network, which accelerates computation by generating features and sharing them with the Detection Network, which performs the final detection. While faster R-CNN models are capable of real-time detection, they struggle to detect smaller objects.

Though the Faster R-CNN is orders of magnitude faster than the Fast R-CNN, the first and second stages of the Faster R-CNN network, CNN feature extraction and an expensive per-region computation, respectively, slow the network down. Kim et al. addressed this issue by modifying the feature extraction stage using cutting-edge technical innovations and presenting a newer network called PVANET. This network is capable of detecting objects belonging to multiple categories with the same accuracy as its competitors while incurring a lower computational cost.

Dai et al. developed a fully convolutional network called R-FCN by adapting the existing ResNet network, which is state-of-the-art for object detection. To improve object detection accuracy, the fully connected layers in Fast R-CNN were replaced with a set of position-sensitive score maps that are also capable of encoding spatial information. As a result, R-FCN achieved the same level of accuracy as Faster R-CNN but at a faster computational speed. Kong et al. introduced the HyperNet network, which is capable of detecting objects at multiple scales via detection at multiple output layers. This network is similar to the MS-CNN, which was proposed by and provides an efficient framework for object detection at multiple scales.

Liu et al. presented a straightforward network dubbed the Single Shot multi-box Detector (SSD) that is capable of delivering high-accuracy real-time performance. This network does not make use of the regional proposal technique. The object localization and classification are carried out in a single forward pass of the network using a technique called 'multi-box' bounding box regression. As a result, the SSD is capable of performing complete computations.

Redmon et al. presented YOLOv3, an updated version of their groundbreaking net-

work YOLO, in their paper. This model outperformed all other state-of-the-art networks in terms of computational speed and accuracy, making it an ideal network for performing real-time detections and tracking while maintaining high accuracy, something that the other networks have failed to do. Additionally, the YOLOv3 is capable of detecting small objects due to its ability to detect objects at three different scales. According to the aforementioned papers, CNNs are the optimal deep learning algorithm for real-time object detection and recognition. As a result of the knowledge gathered, it is evident that the majority of research and development for autonomous driving systems is focused on transportation vehicles such as cars, with only a small amount of research focused on evaluating existing state-of-the-art deep learning models and identifying the best deep learning model for object detection and tracking in construction/excavation environments. As such, this thesis will employ CNN models to recognize small scale vehicles in real time in order to assess the performance of these algorithms and to pave the way for future innovations.

2.4 Metrics

The following metrics [9] are used to evaluate the classification performance of the algorithm:

2.4.1 Accuracy

It is defined as the ratio of the model's correct predictions to the total number of predictions. This is a useful metric, particularly when the target variable classes are well-balanced in the data. This can be written as -

$$\text{No.of correct predictions (CP)} = \text{True Positives} + \text{True Negatives}$$

$$\text{Total no.of predictions (TP)} = \text{True Positives} + \text{True Negatives} + \text{False Positives} + \text{False Negatives}$$

$$\text{Accuracy} = \frac{CP}{TP}$$

Where a True Positive is defined as a detection of an object class that has been trained correctly. A True Negative is defined as a correct missdetection, which occurs when there is no object to detect. A False Positive is defined as an incorrect detection, which occurs when there is no object to detect. A False Negative is defined as an object that is not detected as a ground truth, indicating that the algorithm failed to detect an object that should have been detected.

2.4.2 F₁ Score

The balanced F-measure is used to measure a test's accuracy. The F1 score is considered to be good if the overall number of false positives and false negatives is low. It is defined as the harmonic mean of Precision and Recall.

$$F_1 \text{ Score} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})}$$

Where Precision and Recall are defined as follows: i. Precision: It is defined as the number of true positive results divided by total number of positive results predicted by the classifier.

$$\text{Precision} = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Positives})}$$

ii. Recall: : It is defined as the number of true positive results divided by the sum of true positives and false negatives.

$$\text{Recall} = \frac{\text{True Positives}}{(\text{True Positives} + \text{False Negatives})}$$

2.4.3 Intersection over Union (IoU)

$$\text{IoU} = \frac{\text{area}(gt \cap pd)}{\text{area}(gt \cup pd)}$$

In object detection, the IoU metric [10] determines the degree of overlap between the ground(gt) truth and the prediction (pd). The ground truth and prediction can take any shape (rectangular box, circle, or even irregular shape). It is calculated in the following manner:

IoU is defined diagrammatically as (area of intersection minus area of union between ground-truth and predicted box).

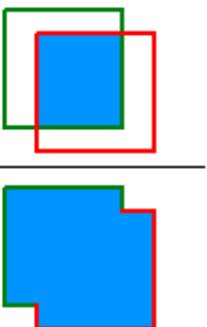
$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{Diagram 1}}{\text{Diagram 2}}$$


Figure 2.9. IoU

IoU values range from 0 to 1, with 0 indicating no overlap and 1 indicating perfect overlap between gt and pd. IoU is beneficial because it allows for thresholding; that is,

we need a threshold (say) and we can use this threshold to determine whether or not a detection is correct. True Positive (TP) is a detection for which $\text{IoU}(\text{gt}, \text{pd})$ and False Positive is a detection for which $\text{IoU}(\text{gt}, \text{pd})$ False Negative is a ground-truth that is overlooked in conjunction with gt for which $\text{IoU}(\text{gt}, \text{pd})$. Is that understood? If not, the following Figure should assist in clarifying the definitions. If the IoU threshold is set to 0.5, TP, FP, and FNs can be identified as illustrated in Fig 4 below.

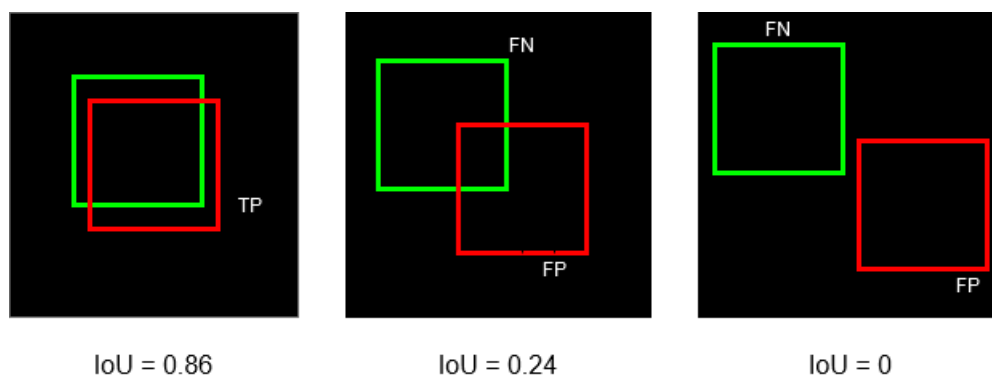


Figure 2.10. Identification of TP, FP and FN through IoU thresholding.

Note: If we increase the IoU threshold above 0.86, the first instance becomes FP, while lowering it below 0.24 results in TP. Remark: The decision to classify a detection as TP or FP and the ground truth as FN is entirely dependent on the IoU threshold chosen. For instance, in the preceding Figure, lowering the threshold below 0.24 results in TP detection in the second image, while increasing the IoU threshold above 0.86 results in FP detection in the first image.

Precision x Recall Curve (PR Curve)

As with IoU, confidence scores are based on a threshold. Increased confidence score threshold results in the model missing more objects (more FNs and thus low recall and precision), whereas a low confidence score results in the model receiving more FPs (hence low precision and high recall). This implies that we must devise a trade-off between precision and recall. The precision-recall (PR) curve represents the relationship between precision and recall at various levels of confidence. Precision and recall remain high for a good model regardless of the confidence score.

Average Precision

$\text{AP}@$ is Area Under the Precision-Recall Curve (AUC-PR) [11] evaluated at IoU threshold. Formally, it is defined as follows

Notation: $\text{AP}@$ or AP means that AP precision is evaluated at IoU threshold. If you see metrics like AP_{50} and AP_{75} then they just mean AP calculated at $\text{IoU}=0.5$ and $\text{IoU}=0.75$, respectively. A high Area Under PR Curve means high recall and high precision. Naturally, PR curve is a zig-zag like plot. That means that it is not monotonically decreasing. We can remove this property using interpolation methods. We will discuss two of those interpolation methods below:

- 11-point interpolation method
- All-point interpolation approach



Figure 2.11. This is a detection made using Mask R-CNN. It shows a bonding box(dotted), segmentation mask, class (fruit) and the confidence score(0.999). Object detection model typically outputs the bounding box, confidence score and clas. Confidence value is the confidence of the model on the detection and it ranges between 0 and 1.

$$AP@α = \int_0^1 p(r) dr$$

Figure 2.12. Definition of Average Precision

- 11-point interpolation method

A 11-point AP is a plot of interpolated precision scores for a model results at 11 equally spaced standard recall levels, namely, 0.0, 0.1, 0.2, . . . 1.0. It is defined as

Mean Average Precision (mAP)

Remark (AP and class count): AP is calculated separately for each class. This means that the number of AP values is equal to the number of classes (loosely). The mean of these AP values is used to calculate the metric: mean Average Precision (mAP). To be more precise, mean Average Precision (mAP) is the sum of all AP values across all classes.

$$mAP@α = \frac{1}{n} \sum_{i=1}^n AP_i \quad \text{for } n \text{ classes.}$$

Remark (AP and IoU): As previously stated, AP is calculated at a predetermined IoU threshold. AP can be calculated using this logic over a range of threshold values. Microsoft COCO calculated the AP of a given category/class at ten different IoUs ranging from 50 to illustrate all of this, let us look at an example.

Example Take a look at the three images in Figure 2.13 below. They contain a total of 12 detections (red boxes) and nine ground truths (green). Each detection is assigned a class denoted by a letter and the model confidence is indicated by a number. Consider the following example, where all detections are of the same object class and the IoU threshold is set to 50

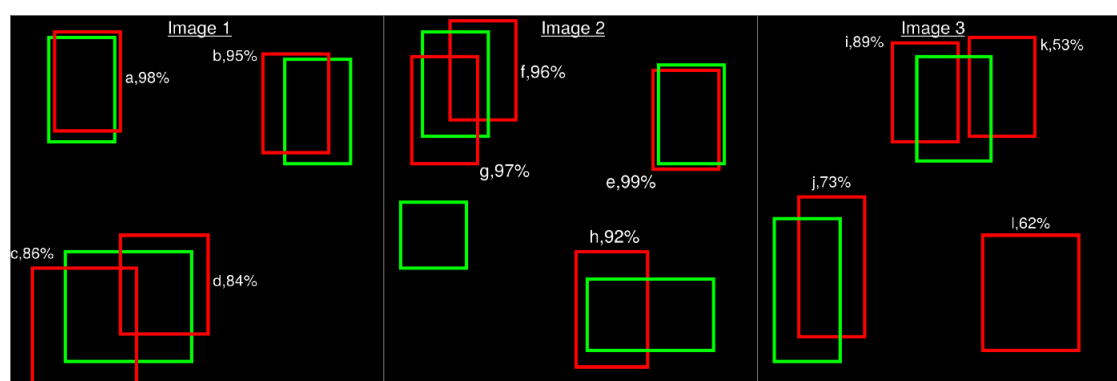


Figure 2.13. A model detecting objects of the same class. There are 12 detections and 9 ground-truths.

Observation (Multiple detections): For example, c,d in image 1, g,f in image 2, and i,k in image 3. When multiple detections are made in this manner, the detection with the highest confidence is labeled TP, while the remaining detections are labeled FPs, provided the detection has an IoU threshold with the truth box. This implies the following:

- c and d becomes FPs because none of them meets the threshold requirement. c and d have 47
- g is a TP and f is FP. Both have IoUs greater than 50
- What about i and k?

Multiple detections of the same object in an image were considered false detections; for instance, five detections of the same object were considered one correct detection and four false detections — Source: PASCAL VOC 2012 paper. Certain detectors can generate multiple detections that overlap with a single ground truth. The detection with the highest degree of confidence is considered a TP in the PASCAL VOC 2012 challenge, while the others are considered FP. - Extrapolated from the article A Survey of Object-Detection Algorithm Performance Metrics

2.4.4 Yolo Loss

The loss function for the Yolov3 model is quite complicated. Yolo calculates three distinct losses at three distinct scales and combines them for backpropagation purposes (As you can see in the above code cell, final loss is the list of three different losses). Using four subfunctions, each loss determines both the localization and classification losses.

- Mean Squared Error(MSE) of Centre (x,y).
- Mean Squared Error(MSE) of Width and Height of the bounding box.
- Binary CrossEntropy objectness score and no objectness score of a bounding box
- Binary CrossEntropy or Sparse Categorical CrossEntropy of multi-class predictions of a bounding box.

$$\begin{aligned}
& \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
& + \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
& + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
& + \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
& + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (3)
\end{aligned}$$

Figure 2.14. Yolo loss function

The final three terms of Yolov2 are squared errors, whereas the final three terms of Yolov3 are cross-entropy errors. In other words, Yolov3 now uses logistic regression to predict object confidence and class predictions.

Chapter **3**

Methodology

This chapter introduces the experimental portion of the thesis. To begin, we will discuss method and dataset selection criteria. Then we'll describe the methods we've chosen, their parameters, and the datasets we've chosen. Finally, postprocessing and evaluation will be discussed. The following chapter discusses the methods' implementation in detail. However, certain implementation details are discussed in this chapter as well, as they have an effect on method selection.

3.1 Research Questions

As discussed in the earlier sections, the overall goal of this research is to identify suitable and highly efficient deep learning models for real-time object recognition and tracking of floating head tanks and finally to evaluate the classification performance of the selected deep learning model. The following research questions have been formulated to fulfill these research objectives.

3.2 Dataset

The dataset contains satellite images taken from Google Earth of tank-containing industrial areas all over the world, which were collected from various sources. Image annotations include bounding box information for floating head tanks that are visible in the image. Fixed-head tanks do not have annotations.

- `large_images`: This is a folder/directory that contains 100 satellite raw images of size 4800x4800 each. All the images are named in `id_large.jpg` format.
- `Image_patches`: The `image_patches` directory contains 512x512 patches generated from the large image. Each large image is split into 100, 512x512 patches with an overlap of 37 pixels between patches on both axes. Image patches are named following an `id_row_column.jpg` format
- `labels_coco.json`: It contains the same labels as the previous file, converted into COCO label format. Here bounding boxes are formatted as `[x_min, y_min, width, height]`.

- `large_image_data.csv`: It contains metadata about the large image files, including coordinates of the center of each image and the altitude.



Figure 3.1. *Annotated Image Example*



Figure 3.2. *Annotated Image Example*

3.3 Data Augmentation

The quality, quantity, and relevancy of training data are all important factors in the performance of most machine learning models, and deep learning models in particular. Insufficient data, on the other hand, is one of the most common problems encountered when attempting to implement machine learning in the enterprise. This is due to the fact that gathering such information can be both expensive and time-consuming in many instances.

The performance of the majority of machine learning models, and particularly deep learning models, is dependent on the quality, quantity, and relevance of training data. Inadequate data, on the other hand, is one of the most common obstacles to implementing machine learning in the enterprise. This is because obtaining such data can often be costly and time consuming.

Businesses can use data augmentation to reduce their reliance on training data collection and preparation and to accelerate the development of more accurate machine learning models.

3.3.1 What is augmentation of data?

The term "data augmentation" [12] refers to a collection of techniques for artificially increasing the volume of data by generating new data points from existing data. This can be accomplished by making minor adjustments to existing data or by using deep learning models to generate new data points.

3.3.2 Why is it critical now?

Machine learning applications, particularly in the domain of deep learning, continue to diversify and grow at a rapid pace. Techniques for data augmentation may be an effective tool in combating the challenges that the artificial intelligence world faces.

By generating new and unique examples for training datasets, data augmentation can help improve the performance and outcomes of machine learning models. When a machine learning model's dataset is sufficiently large and diverse, the model performs better and more accurately.

Collecting and labeling data for machine learning models can be time-consuming and costly. Businesses can reduce these operational costs by transforming datasets using data augmentation techniques.

Cleaning data is one of the steps in developing a data model, which is necessary for high-accuracy models. However, if data cleaning reduces its representability, the model will be unable to make accurate predictions for real-world inputs. By introducing variations that the model might encounter in the real world, data augmentation techniques make machine learning models more robust.

3.3.3 What are the advantages of augmented data?

Among the benefits of data augmentation are the following:

- Enhancing the predictive accuracy of models incorporating additional training data into the models avoiding data scarcity in order to develop more accurate models
- minimizing data overfitting (a statistical error that occurs when a function is too closely related to a small number of data points) and increasing data variability
- Increasing the generalizability of models contributes to the resolution of class imbalance issues in classification. Cost savings associated with data collection and labeling
- Allows for the prediction of rare events
- Prevents issues with data privacy

3.3.4 What are the difficulties associated with data augmentation?

Businesses must develop evaluation systems for the augmented datasets' quality. As the use of data augmentation methods grows, it will be necessary to assess the output's quality. The domain of data augmentation requires new research and studies to generate new/synthetic data for advanced applications. For instance, generating high-resolution images with GANs can be difficult. If a real-world dataset contains biases, augmented data will also contain biases. As a result, determining the optimal data augmentation strategy is critical. What are some of the applications/examples of data augmentation? In general, image recognition and natural language processing models make use of data augmentation techniques. Additionally, the medical imaging domain makes use of data augmentation to apply transformations to images and infuse datasets with diversity. The reasons for healthcare's interest in data augmentation are as follows:

A small collection of medical images Due to patient data privacy regulations, sharing data is not easy. There are only a few patients whose data can be used as training data for the diagnosis of rare diseases.

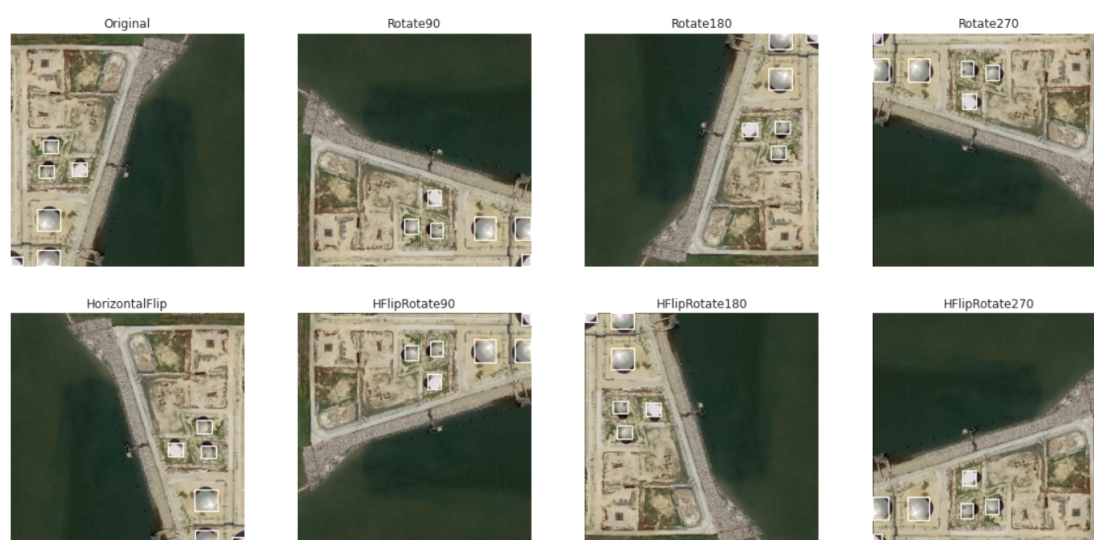


Figure 3.3. *Image Augmentation*

Below are the methods used in our experiments:

- Horizontal Flipping
- Rotating 90 degree
- Rotating 180 degree
- Rotating 270 degree
- Horizontal Flipping and 90-degree Rotating
- Horizontal Flipping and 180-degree Rotating
- Horizontal Flipping and 270-degree Rotating

3.4 Training

After finishing all the steps mentioned above and making necessary changes in the configuration file, the training process is initialized. The step count and classification loss in each step can be seen on screen, as shown in Figure 3.2. It can be noted that the classification loss starts at a really high value and gradually decreases as the algorithm learns as the iterations progress. This has been visualized in the form of a graph, with the help of TensorFlow Board shown in Figure 3.5. In Figure 3.2, the 'Globalstep' represents the iteration or batch number that is being processed. 'Loss' value given is the sum of Localization loss and Classification loss. These represent the price paid for inaccuracy of predictions. The optimization algorithm keeps reducing the loss value until a point where the network is considered to be trained by the researcher. In general, lesser loss implies better training of the model. 'Sec/step' is the time taken to process that corresponding step.

```

Pretrained Weight Loaded
Yolo DarkNet weight loaded
Frozen DarkNet layers
Model: "yolov3"

```

Layer (type)	Output Shape	Param #	Connected to
input (InputLayer)	[(None, 416, 416, 3)]	0	
yolo_darknet (Functional)	(None, None, None, 3)	40620640	input[0][0]
yolo_conv_0 (Functional)	(None, 13, 13, 512)	11024384	yolo_darknet[0][2]
yolo_conv_1 (Functional)	(None, 26, 26, 256)	2957312	yolo_conv_0[0][0] yolo_darknet[0][1]
yolo_conv_2 (Functional)	(None, 52, 52, 128)	741376	yolo_conv_1[0][0] yolo_darknet[0][0]
yolo_output_0 (Functional)	(None, None, None, 3)	4747288	yolo_conv_0[0][0]
yolo_output_1 (Functional)	(None, None, None, 3)	1194008	yolo_conv_1[0][0]
yolo_output_2 (Functional)	(None, None, None, 3)	302104	yolo_conv_2[0][0]

```

Total params: 61,587,112
Trainable params: 20,949,576
Non-trainable params: 40,637,536

```

Figure 3.4. *Transfer Learning*

We have used adam optimizer(initial learning rate=0.001) to train our model and applied cosine decay to reduce the learning rate w.r.t number of epochs. The best weight is saved using Model Checkpoint during the training, and the last weight is saved after the completion of training.

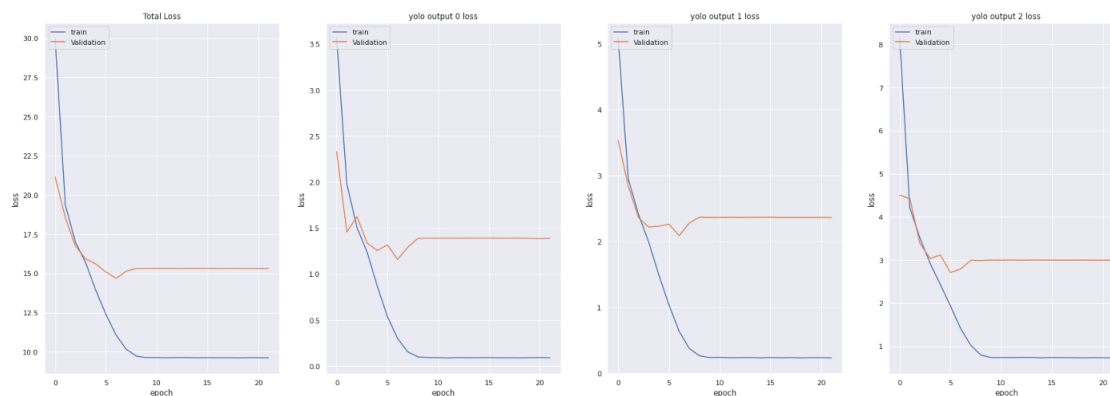


Figure 3.5. *Tensorboard Training*

3.5 Calculating Volume via Shadow Extraction

3.5.1 An overview of available methods

In the past decade, urban remote sensing has been dominated by high-resolution remote sensing. However, medium and high-resolution remote sensing images are highly influenced by shadows, especially in metropolitan settings with dense, tall structures, resulting in significant information loss in the shadow region. Accurate shadow extraction and information restoration in shadow areas are applicable not only to the three-dimensional reconstruction of structures, but also to urban feature categorization, urban planning, road network extraction, and impervious layer research among others. Shadow detection and information compensation research focuses mostly on RGB photos, single shadows, and video sequences. The remote sensing images are susceptible to aerosol and sensor noise, further complicating shadow-related research. This is the reason why shadow identification and correction are lacking from generally used methods for remote sensing photos with a complex background and many shadows.

The complete process of shadow identification and compensation is referred to as shadow removal. Shadow extraction accuracy is a crucial prerequisite for shadow compensation [13]. Numerous shadow extraction strategies, including model-based, feature-based, and machine learning-based ones, have been proposed by researchers recently. The model-based methods obtain shadow information using mathematical models generated from the sensor's position and sun azimuth, the most common of which is the digital earth model (DEM). The DEM was used in, where the author obtained a crude shadow through DEM and then utilized SVM (Support Vector Machine) to optimize the shadow feature. However, the greatest disadvantage of the model-based model is that it requires less available prior knowledge such as camera and lighting orientation information. The feature-based methods often require picture feature extraction and segmentation, in which the feature may be paired with spectral, texture, and semantic information, such as shadow indexes based on color spaces for shadow highlighting. This type of algorithm primarily converts images from the RGB model to hue and saturation intensity (HIS) space or an equivalent space, such as hue, saturation, and value (HSV), hue, chroma,

and value (HCV), luma, inphase, and quadrature (YCbCr and YIQ), where shadow areas have a higher hue and lower intensity than non-shadow areas. Tsai et al. evaluated and analyzed the attribute of $(H+1)/(I+1)$ in different color spaces and created a shadow mask using Otsu's technique based on this information. Silva et al. translated photos to the CIELCh model, presented a modified shadow index, and used multilayer thresholding to generate the shadow mask. Ma et al. later enhanced Tsai's index and presented NSVDI for the HSV space. Nonetheless, the HSV space is constrained by incorrect values in the case of equal pixel values in the R, G, and B bands, a common occurrence in photographs. Multispectral remote sensing photos typically include NIR bands with longer wavelengths, which are extremely sensitive to shadows. Consequently, NIR is typically used in conjunction with other bands to strengthen shadow features, such as the shadow feature improvement techniques established by that mix NIR bands with other bands. Since 2015, the development of deep learning networks has accelerated, bringing new ideas for shadow extraction. Numerous effective deep learning techniques have been developed for shadow detection, achieving better results than conventional techniques, yet current machine learning-related algorithms are mostly for ordinary photos. Due to illumination variations and satellite revisit cycles, it is challenging to get shadowed and unshadowed photos of the same place for shadow identification from remote sensing photographs. In conclusion, the approaches described to date are only suitable for shadow extraction under restricted situations; hence, further study is required to develop methods for shadow extraction from remote sensing photos that are efficient.

3.5.2 Detecting shadows in the HSV space

The approach uses normalized saturation-value difference index (NSVDI) in Hue-Saturation-Value (HSV) color space [14] to detect shadows and exploits histogram matching to recover the information under shadows. When working with the HSV color space, hue, saturation, and value are all used to express the colors of an image. The hue values in shadow areas are relatively consistent and high in value, whereas the hue values in non-shadow areas are relatively low. On the other hand, it is difficult to tell the difference between the saturation of the shaded area and the saturation of the surrounding ground. The value of the shadow area is less than the value of the non-shadow area, and vice versa. When the shadow region has low luminance and hue values, we can use the $(H + 1)/(V + 1)$ ratio image to enhance them. Here, H is the hue channel image and V is the value channel image, and the ratio image is the $(H + 1)/(V + 1)$ ratio image.

Many different approaches have been proposed for rationing these channels in order to improve shadows on the screen. The NSVDI algorithm proposes the SVS+V solution as a possible solution.

3.5.3 Proposed method for shadow extraction

In my research, I discovered that the $H+1V+1$ was distorted by strong artifacts in the H channel, which was most likely caused by the source images being RGB jpegs saved from Google Earth rather than true high resolution satellite photography, as opposed to

true high resolution satellite photography. Despite its impressive performance on some images, the SVS+V technique failed miserably on the majority of them.

It was discovered that the experimentally validated ratio $(l1+l3)V+1$ was effective, where $l1$ and $l3$ represent the first and third channels of the LAB color space image, respectively.

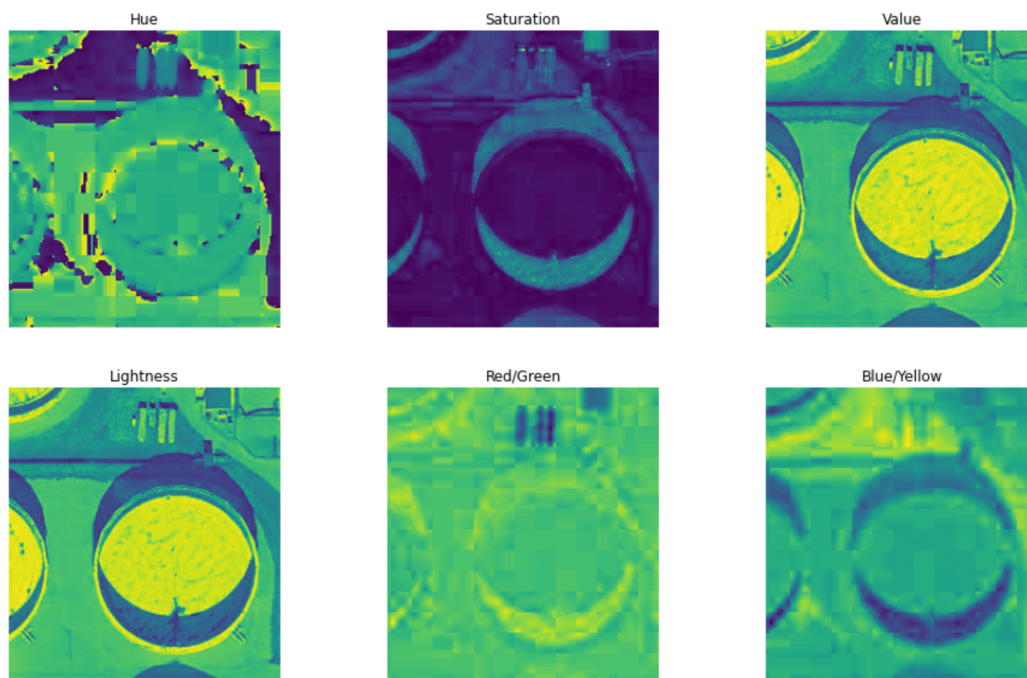


Figure 3.6. *Enhance Shadow Portions*

With the help of thresholding, we can filter the enhanced image. I discovered that the minimum threshold was frequently too strict, whereas the mean threshold was frequently too permissive. A combination of 0.5 minimum threshold and 0.4 mean threshold yields satisfactory results.

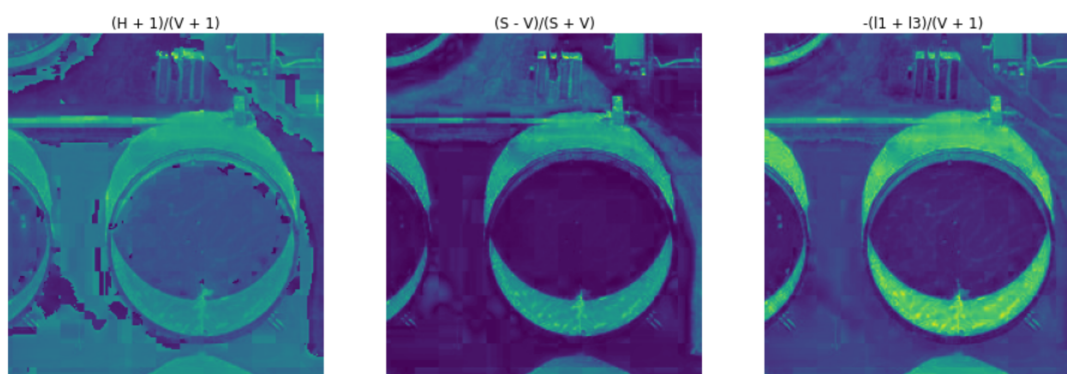


Figure 3.7. *Filtered enhanced Image*

The threshold-ed image is then processed with morphological operations. These operations are:

- Hessian Filter - cleans up noise and line artifacts from white pipes which appear in many images
- Clear Border - clears contours from surrounding tanks
- Morphological Closing - helps separate shapes
- Area Closing - fills small holes
- Morphological Labeling - labels features



Figure 3.8. *Thresholded image*

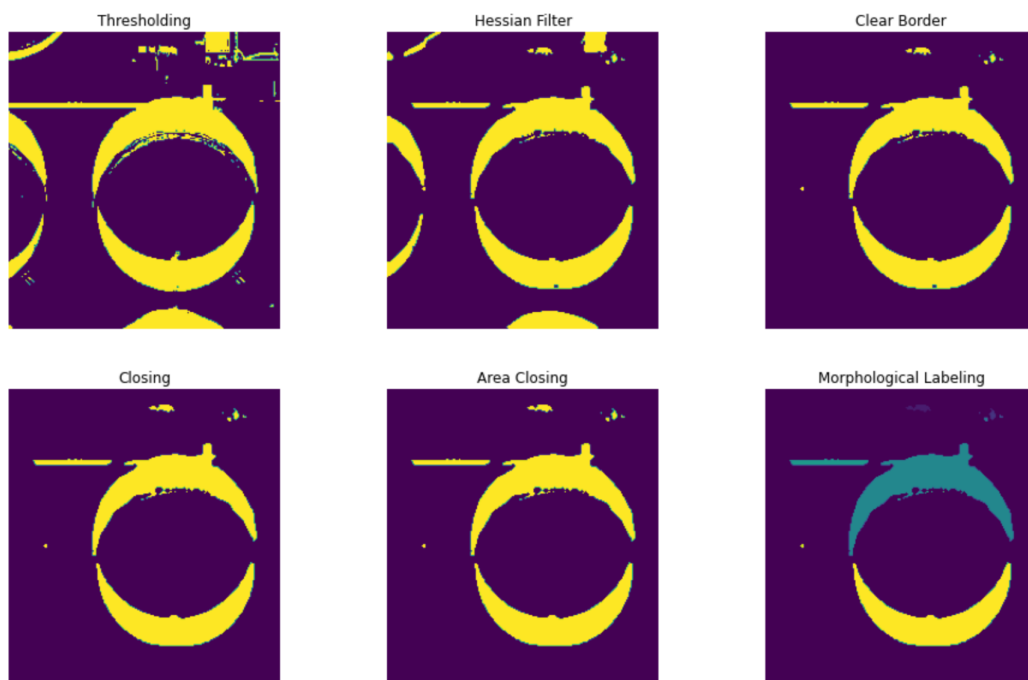


Figure 3.9. *More Filtration*

We then filter the regions present by certain heuristics. The bounding box of the feature should intersect the bounding box of the tank. The feature should have an area of more than 25 pixels. The pixel coverage of the labeled image should be approximately the same as in the threshold image.

The first two clear up small artifacts. The third deals with the fact that the Hessian filter sometimes creates regions in spaces that are otherwise empty.



Figure 3.10. *Highlighted Tank Contours*

We extract the two tank shadow contours



Figure 3.11. *Two Contours of a tank side by side*



Figure 3.12. *Two Contours of a tank*

3.5.4 Calculating Volume and Number of Barrels

Volume is estimated as 1 minus the ratio of the smaller area to the larger area. The larger area corresponds to the exterior shadow of the tank, while the smaller area corresponds to the interior shadow.

Several methods exist for measuring and quantifying volumes of petroleum [15] (oil, condensate, NGL, and gas). Quantities can be expressed in terms of mass (weight), volume, and energy density. Various units, such as cubic metres and barrels, both of which indicate volume, may be used to express the various numbers.

In the petroleum sector, older American (British) units are commonly utilized. When doing conversions, such as from volume to energy content, there is no exact conversion factor, and you must know/make assumptions about the substance's attributes. Assumptions regarding energy content per cubic meter of gas and weight per volume unit of natural gas liquids are examples. Volumes of oil and gas are often expressed in standard cubic metres (Sm³), and the temperature and pressure at which they apply must also be specified for an accurate representation of volumes. The standard conditions are 15 degrees Celsius and normal air pressure (1013.25 hPa).

Before amounts of different petroleum products (oil, gas, NGL, and condensate) can be added, they must be converted to a standardized quantity and unit. Using standard cubic metres of oil equivalents is the most prevalent technique (abbreviated as Sm³ o.e.).

When doing conversions, the Norwegian Petroleum Directorate applies a volumetric conversion of NGL to liquid and an energy-based (but not accurate) conversion factor for gas based on average shelf features. The characteristics of oil, gas, and NGL vary over time and from field to field. In resource reports and other comparisons involving oil equivalents, however, a consistent and uniform conversion factor is applied to all fields and finds.

Table 3.1. Conversion factors for liquid (oil, condensate and NGL)

Conversion factors for liquid (oil, condensate and NGL)			
1 Sm ³	=	6.2898 barrels	English barrel (bbl) or Stock Tank Barrel (STB)
1 bbl	=	0.1590 Sm ³	
1 Sm ³	=	0.84 toe	Tonne of oil equivalent (at a density of 840 kg/Sm ³)
1000 Sm ³ /y	=	17.23 bbl/d	Production rate (17.18 bbl/d in leap years)
1000 bbl/d	=	58035 Sm ³ /y	Production rate (58194 Sm ³ /y in leap year)

Chapter **4**

Software engineering

4.1 Back-end

When Python web frameworks like Flask and Django first grew to fame and prominence, the language Python itself was quite different than it is today. Examples of these frameworks include Many of the components that make up current Python, such as asynchronous execution and the standard known as ASGI (Asynchronous Server Gateway Interface), did not yet exist or were in their infancy at the time.

Python's FastAPI is a web framework that was developed from the ground up to take advantage of new capabilities that are available in Python. It connects with clients in an asynchronous and concurrent manner using the ASGI standard, and if necessary, it is also compatible with the WSGI protocol. It is possible to use asynchronous functions for both routes and endpoints. In addition, FastAPI makes it possible to write web apps quickly and effectively in Python code that is up to date and contains type hints.

4.2 Front-end

Adrien Treuille, Amanda Kelly, and Thiago Teixeira established the software company Streamlit in 2018. The company has its headquarters in San Francisco, California, and it was founded in 2018. Streamlit provides an open-source platform for machine learning and data science teams to use in order to create data applications with the programming language python.

The platform assists data scientists and machine learning engineers in the creation of python-based apps by utilizing python scripting, application programming interfaces (APIs), widgets, immediate deployment, team communication tools, and application management solutions. Programs such as face-GAN explorers, geographic data browsers, deep dream network debuggers, and real-time object identification applications are all examples of the kind of applications that may be produced with Streamlit. Frameworks such as Scikit Learn, Altair, Bokeh, latex, Keras, Plotly, OpenCV, Vega-Lite, PyTorch, NumPy, Seaborn, Deck.GL, TensorFlow, Python, Matplotlib, and Pandas are all compatible with Streamlit.

4.3 The app

The application is extremely simple to comprehend and employ. After defining the diameter and height of the tank, a request is sent to the API that is running in the background. Following the detection of floating head tanks, the shadow extract technique is applied to every one of the detected floating head tanks. After the user inputs the diameter and height, the volume is computed by multiplying those values by the percentage returned by the back-end API as a json file. Then, on streamlit, basic calculations such as the total number of barrels are performed and sent to the user so they can view the results.

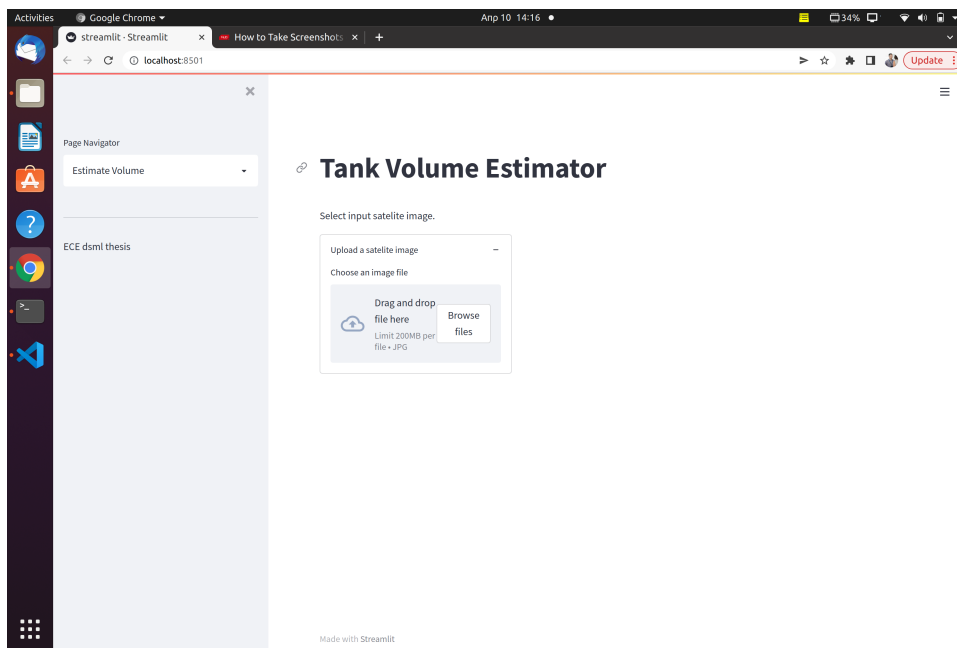


Figure 4.1. Browse through the files you want to upload

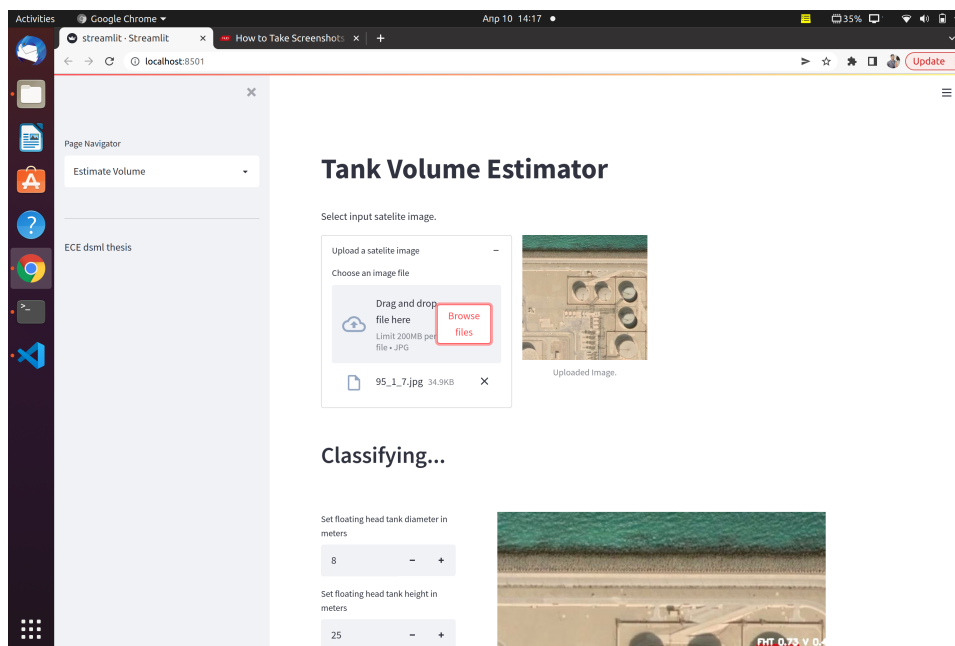


Figure 4.2. YoloV3 is running in the back and detecting the floating head tanks

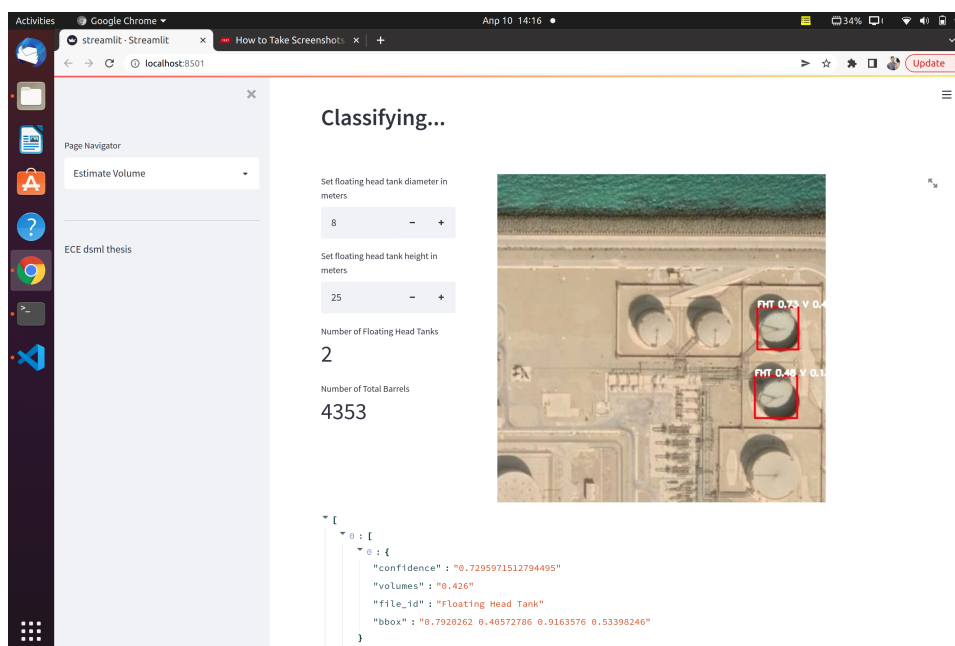


Figure 4.3. Number of floating head tanks and barrels of oil are returned

Chapter 5

Experimental Evaluation

5.1 Motor Oil's Refinery

The application was used to predict the number of crude oil barrels of certain floating head tanks of motor oil's refinery in Greece. Satellite images were given as inputs to the application as depicted below.



Figure 5.1. Motor Oil refinery from space

5.1.1 The Refinery

The refinery is situated near Agioi Theodoroi, Corinth, approximately 70 kilometers from Athens's city center. With its auxiliary units and gasoline distribution facilities, the Refinery represents the largest privately held industrial complex in Greece and is regarded as one of the most modern refineries in Europe. The complexity of the Motor Oil refinery, as measured by Nelson's Complexity Index, is 11.54.

It has the ability to refine a variety of crude oil types and produces a vast array of oil products. It supplies Greek oil businesses, but the vast majority of its output is exported. In addition, it is the sole refinery in Greece that produces base oils. The

refinery's output conforms to European Union regulations and the strictest international standards. The refinery's technical sophistication enables Motor Oil to produce products with a high added value, tailoring the final product mix to market demands, assuring better distribution pricing and attaining higher refining margins than other composite refineries in the Mediterranean.

5.1.2 Characteristics of the Refinery

The refinery produces all sorts of fuel and is one of the most technologically modern and complicated in Europe, including Hydrocracker and Catalytic Cracking units with a Nelson Complexity Index score of 11.54. It produces refined fuels (gasoline and diesel for automobiles) according to EU specifications.

Refinery characteristics:

- Processing capacity: 185,000 barrels of crude oil per stream day (BSD).
- It is the only Greek refinery with a unit producing base oils and finished lubricants that are approved by international organizations such as the American Petroleum Institute (API), the European Automobile Manufacturers Association (ACEA), and the United States Army and Navy.
- With the recent installation of a fifth gas turbine, the electricity and steam co-generation unit now has a capacity of 85 MW.
- Natural gas is used as both a fuel and a raw material in the creation of hydrogen. It has a capacity of 2,600,000 cubic meters (Crude Oil: 1,000,000 m³, Intermediate Finished Products: 1,600,000 m³).
- It features sophisticated tanker docking facilities designed for tankers of up to 450,000 tons DWT that can accommodate more than 3,000 boats each year.
- It is equipped with a sophisticated truck loading terminal that can accommodate 220 tanker trucks per day and considerably increases Motor Oil's competitive position on the southern Greek market.

5.1.3 Refinery Units

The refinery is constituted of the following units:

- Fuels Production: Crude Oil is refined in the Crude Distillation Units (capacity of more than 186,000 barrels per day) to produce LPG, Naphtha, Kerosene, Diesel, and Fuel Oil. Kerosene and Diesel are reprocessed to make jet fuel and diesel fuel, respectively, by removing sulfur, thereby conforming to the standards (both automotive and heating grades).
- Fuels Production: The Crude Distillation Units (around 186,000 bbl/day total capacity) refine crude oil, from which LPG, Naphtha, Kerosene, Diesel, and Fuel Oil

are generated. Kerosene and Diesel are refined to generate jet fuel and diesel fuel, respectively, by removing sulfur, thereby conforming to the standards (both automotive and heating grades).

- Gasoline Production: Here, naphtha is treated in order to produce gasoline with a high octane number, hence reducing the need to add lead to fuel.
- Hydrocracker Complex: This is one of the company's major investment projects, with capital expenditures of € 350 million and completion in 2005. The functioning of the Hydrocracker enables the manufacture of new clean fuels with a low sulfur content in accordance with the 2009 criteria of the European Union (Auto Oil II). In addition, the unit significantly contributed to the refinery's environmental improvement, as the FCC's emissions were significantly decreased.
- The FCC complex is fed atmospheric Fuel Oil to make LPG, gasoline, diesel, and Fuel Oil. A portion of the LPGs are sent to downstream units where they are processed into high-quality gasoline components.
- Lubes Production: In addition to atmospheric fuel oil, the Lubes Vacuum Unit also receives atmospheric fuel oil. After a series of steps to increase the lubricants' qualities, such as viscosity index, pour point, and cloud point, the final base lubricants are manufactured and stored. In addition, asphalt can be produced from the Lubes vacuum unit, while the bottoms of both vacuum units are sent to the visbreaker for further upgrading and fuel oil production.

5.2 Results of the application

Below are the estimations of certain floating head tanks of the application:

The screenshot shows a web application interface for classifying floating head tanks. The interface includes a sidebar with a logo and navigation options, a main content area with input fields for diameter and height, and a central image of an industrial site with a red bounding box around a tank. The JSON response is also visible at the bottom.

Classifying...

Set the average floating head tank diameter in meters

50.00 - +

Set the average floating head tank height in meters

25.00 - +

Number of Floating Head Tanks

1

Number of Total Barrels

235,706

FHT 0.98 v 0.77

Floating Head Tanks detected with the YOLOv3 Model and Localized with Bounding Boxes

JSON Response

```
[
  {
    "confidence": "0.98"
  }
]
```

Figure 5.2. A floating head tank detected among regular tanks

The screenshot shows a web application interface for classifying floating head tanks. The interface includes a sidebar with a logo and navigation options, a main content area with input fields for diameter and height, and a central image of an industrial site with a red bounding box around a tank. The JSON response is also visible at the bottom.

Classifying...

Set the average floating head tank diameter in meters

50.00 - +

Set the average floating head tank height in meters

25.00 - +

Number of Floating Head Tanks

1

Number of Total Barrels

285,312

FHT 0.97 v 0.93

Floating Head Tanks detected with the YOLOv3 Model and Localized with Bounding Boxes

JSON Response

```
[
  {
    "confidence": "0.972659714874268"
  }
]
```

Figure 5.3. A FHT that appears to be full is correctly captured by the algorithm

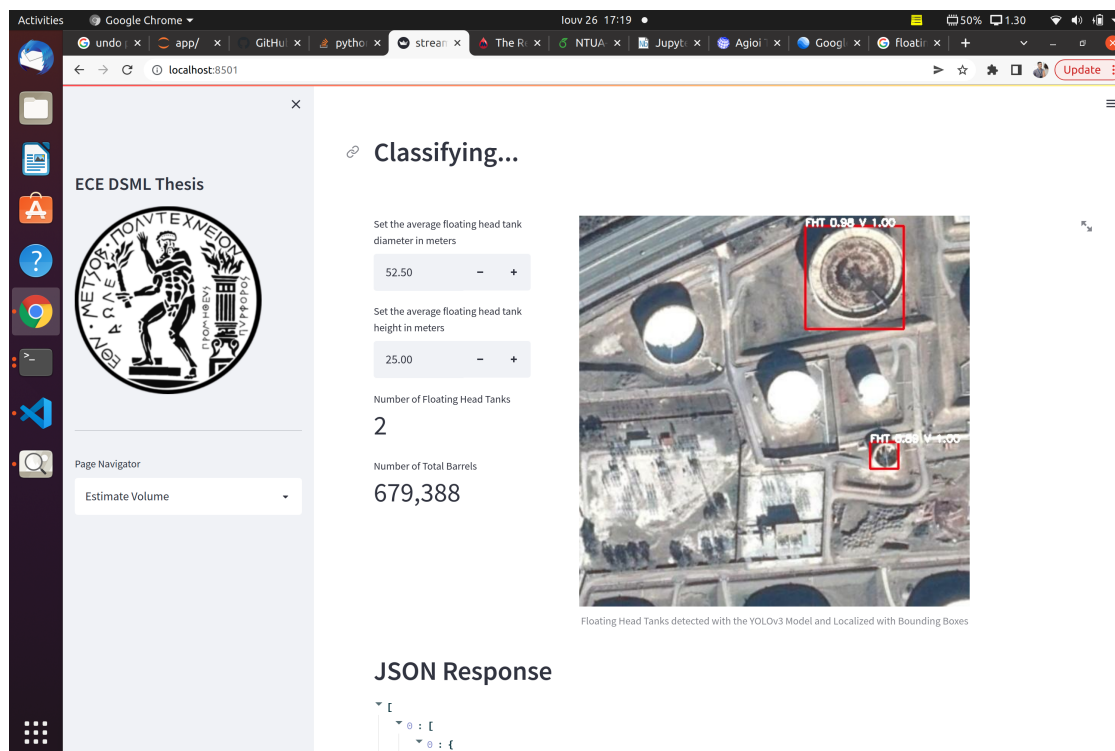


Figure 5.4. *The algorithm missed a floating head tank but correctly captured the other two present and successfully estimated the volume*

5.3 Conclusion

In the method suggested in this work, remote sensing photos were converted into HSV images, and a ratio image was created to highlight the shadows. The ratio image was thresholded using the highest inter-class variance of the levels, followed by thresholding according to area and processing using morphological operators to produce the oil tank's shadow.

In the future, Hough transform can be tested to detect the radius of the tank's top thus the tank's volume can be determined with sufficient precision, we will do additional research on the nature of shadows in various color spaces and combine the texture characteristics of shadows in order to extract tank shadows from a larger variety of photos. The Hough transform and machine-learning-related techniques should be further coupled in the future to increase the accuracy of the tank identification rate and tank volume estimation.

Bibliography

- [1] Oliver Montenbruck, Eberhard Gill και Fh Lutze. *Satellite orbits: models, methods, and applications*. *Appl. Mech. Rev.*, 55(2):B27–B28, 2002.
- [2] Kumar Navulur, Fabio Pacifici και Bill Baugh. *Trends in optical commercial remote sensing industry [Industrial profiles]*. *IEEE Geoscience and Remote Sensing Magazine*, 1(4):57–64, 2013.
- [3] Sławomir Skoneczny. *Nonlinear image sharpening in the HSV color space*. *Przegląd Elektrotechniczny (Electrotechnical Review)*, 88(2):140–144, 2012.
- [4] Morris Muskat. *Physical principles of oil production*. 1981.
- [5] Batta Mahesh. *Machine learning algorithms-a review*. *International Journal of Science and Research (IJSR).[Internet]*, 9:381–386, 2020.
- [6] Horace B Barlow. *Unsupervised learning*. *Neural computation*, 1(3):295–311, 1989.
- [7] Richard S Sutton και Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.
- [8] Joseph Redmon και Ali Farhadi. *Yolov3: An incremental improvement*. *arXiv preprint arXiv:1804.02767*, 2018.
- [9] Mohammad Hossin και Md Nasir Sulaiman. *A review on evaluation metrics for data classification evaluations*. *International journal of data mining & knowledge management process*, 5(2):1, 2015.
- [10] Hamid Rezaatofghi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid και Silvio Savarese. *Generalized intersection over union: A metric and a loss for bounding box regression*. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, σελίδες 658–666, 2019.
- [11] Rafael Padilla, Sergio L Netto και Eduardo AB Da Silva. *A survey on performance metrics for object-detection algorithms*. *2020 international conference on systems, signals and image processing (IWSSIP)*, σελίδες 237–242. IEEE, 2020.
- [12] Connor Shorten και Taghi M Khoshgoftaar. *A survey on image data augmentation for deep learning*. *Journal of big data*, 6(1):1–48, 2019.
- [13] Tarek Rashed και Carsten Jürgens. *Remote sensing of urban and suburban areas*, τόμος 10. Springer Science & Business Media, 2010.

- [14] Rita Cucchiara, Costantino Grana, Massimo Piccardi, Andrea Prati και Stefano Sirotti. *Improving shadow suppression in moving object detection with HSV color information*. *ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No. 01TH8585)*, σελίδες 334-339. IEEE, 2001.
- [15] Wieslaw Grabon, Pawel Pawlus, Waldemar Koszela και Rafal Reizer. *Proposals of methods of oil capacity calculation*. *Tribology International*, 75:117-122, 2014.

List of Abbreviations

BPF	Band Pass Filter
FHT	Floating Head Tank
HSV	hue, saturation, value; also known as HSB, for hue, saturation, brightness
YOLO	You only look once