Master Thesis

submitted within the UNIGIS MSc programme
Interfaculty Department of Geoinformatics - Z_GIS
University of Salzburg

Automated Feature Extraction and Object-Based Detection from High Resolution Aerial Photos Based on ML and AI

Flowlines Oil Leaks Detection

by

MOHAMMED SALIM SAIF AL SULAIMANI
105085

A thesis submitted in partial fulfilment of the requirements of
the degree of
Master of Science (Geographical Information Science & Systems) – MSc (GISc)

Advisor:

Prof Dr. Gudrun Wallentin

Salzburg, Austria, 05/09/2020

Automated Feature Extraction and Object-Based Detection from High Resolution Aerial Photos based on ML and AI

Case: Flowlines Oil Leak Detection

MOHAMMED SALIM SAIF AL SULAIMANI

University of Salzburg

Thesis Committee Members

Thesis Title: Automated Feature Extraction and Object-Based Detection from High Resolution Aerial Photos based on ML and AI - Case: Flowlines Oil Leaks Detection

1- Student Name: Mr. Mohammed Salim Saif AL Sulaimani
Student ID: 105085
Department of Geoinformatics
University of Salzburg

2- University Supervisor Name: Prof. Dr. Gudrun Wallentin
Assoc. Prof. for Geoinformatics and Ecology
Department of Geoinformatics
University of Salzburg

Science Pledge and Declaration

I hereby declare that the MSc Thesis "Automated Feature Extraction and Object-Based Detection from High Resolution Aerial Photos based on ML and AI - Case: Flowlines Oil Leak Detection " is presenting my entire work which I submitted to a partial fulfilment of the requirements of the degree of Master of Science in Geographical Information Science & Systems as part of UNIGIS Program from Salzburg University. During my journey to complete this work I have obtained guidance, advises and a rich information from my university supervisor, my external guiders and my colleagues at work which I have stated at "Acknowledgement Section".

So, by my signature below, I certify that my thesis is entirely the result of my own work. I have cited all sources I have used in my thesis and I have always indicated their origin.

University of Salzburg, 05/09/2020 Signature

Dedication

I dedicate my dissertation “MSc Thesis Work” to all my family members. And a special dedication goes to my wife and my parents who were always close to support me and encourage me to pursue my dreams and finish my dissertation. Also, to my wonderful university advisor who guided me through my journey of this dissertation with her valuable advises, comments, feedbacks and for keeping me on the right path and the right track.

Acknowledgement

Here, I'm taking the opportunity to thank my university supervisor Prof. Dr. Gudrun Wallentin who supported me through the entire process to accomplish my Thesis work, her valuable advises, comments and feedbacks which kept me on the right path. She, provided me with a professional guidance and taught me a great about scientific research and academic works.

Also, I would like to thank my external guiders Dr. Rachid Rahmoune and Ms. Khadija Al Aisari, both have contributed a lot to this work by providing an extensive support, ideas, solutions and facilitate the time to pursue the goals to this work.

Not to forget, that my special thanks and appreciation goes to my Work Manager Mr. Salim Al Harthy who treated me like a son and was the first one who encouraged me to take the first step in my further studies (Post- Graduate) in my own initiative and my own financial support. This person believed on me and made a lot of changes in my study and my future career and whatever I say, I will not be able to give back to him what he gave to me.

I would like to extend my appreciation to my best friend Dr. Chituru Obowu who was with me from the beginning of my study with his continuous motivation and kind support.

Another appreciation goes to my colleagues at work Mr. Ahmed Al Azri and Maadh Al Mamari, both were there with me to complete this work which would not be possible to complete without their support.

Finally, I express my gratitude to my wife, my parents, brothers and sisters for providing me with the required support throughout the entire MSc program. And through the process of researching and writing my Thesis Document which would not be possible without them.

Abstract

With the development of Remote Sensing technology, the resolution of optical Remote Sensing images has greatly improved and images have become largely available. Numerous detectors have been developed for detecting different types of objects. In the past few years, Remote Sensing has benefited a lot from deep learning, particularly Deep Convolution Neural Network (CNNs). Deep learning holds a great promise to fulfill the challenging needs of Remote Sensing and solving various problems within different fields and applications.

The use of Unmanned Aerial Systems in acquiring Aerial Photos becomes highly use and preferred by most of the organizations to support their activities because of its high resolution and accuracy which make the identification and the detection of very small features much easier than the Satellite Images. This has opened an extreme era of Deep Learning in different applications not only in feature extraction and prediction but also in analysis.

This Thesis, addresses the capacity of Machine Learning and Deep Learning in detecting and extracting Oil Leaks from Flowlines (Onshore) using High Resolution Aerial Photos which been acquired by UAS fixed with RGB Sensor to support an early detection of these Oil Leaks and prevent the company from the leak's losses and the most important thing environmental damage.

Here, there are two different approaches and different methods of DL have been demonstrated. The first approach focusing in detecting the Oil Leaks from the RAW Aerial Photos (not processed) using a Deep Learning called Single Shoot Detector (SSD), the model draws bounding boxes around the leaks and the results were extremely good. The second approach focusing in detecting the Oil Leaks from the Ortho-mosaiced Images (Georeferenced Images) by developing three Deep Learning Models using (MaskRCNN, U-Net and PSP-Net Classifier). Then a Post-Processing is performed to combine the results of these three Deep Learning Models to achieve a better detection results and improved the accuracy.

Although, there are relatively small amount of dataset available for the training purpose, the Trained DL Models have showed good results in extracting the extent of the Oil Leaks and obtains an excellent and accurate detections.

List of Tables

List of Figures

List of Acronyms and Abbreviation

MSc	Master of Science
AI	Artificial Inelegance
ML	Machine Learning
DL	Deep Learning
CNNs	Convolution Neural Network
DCNN	Deep Convolution Neural Network
FRCNN	Faster Region Convolution Neural Network
STDCNN	Semi-Transfer Deep Convolutional Neural Network
SSD	Single Shoot Detector
MaskRCNN	Mask Regional Convolution Neural Network
U-Net	U Shape Network
PSP-Net Classifier	Pyramids Scene Parsing Net Classifier
UAS	Unmanned Aerial System
PDO	Petroleum Development Oman
RGB	Red, Green, Blue
GCP	Ground Control Points
GSD	Ground Sampling Distance
ANN	Artificial Neural Network
SNARC	Stochastic Neural Analog Reinforcement Computer
R-CNN	Regional-based Convolutional Neural Network
SPP-net	Spatial Pyramid Pooling
F-RCNN	Fast Regional-based Convolutional Neural Network
R-FCNN	Regional-based Fully Convolutional Network
FPN	Feature Pyramid Network
Mask R-CNN	Mask Regional-based Convolutional Neural Network
SSD	Single Shoot Detector
YOLO	You Only Looks Once
CPU	Central Processing Unit
GPU	Graphics Processing Unit
RAM	Random Access Memory
GHz	Gigahertz
VoTT	Visual Object Tagging Tool
PASCAL VOC	PASCAL Visual Object Classes

XML	Extensible Markup Language
TFRecords	TensorFlow Records
COCO	Common Objects in Contents
ReLU	Rectified Linear Unit
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
AP	Average Precision
IoU	Intersection of Union
mAP	Mean Average Precision
AR	Average Recall

Table of Contents

1. Introduction

Within the last few years, Unnamed Aerial Systems or as it is known as UAS, UAV or Drones has developed a lot and became widely used. Nowadays UAS can take High Resolution Aerial Photos using High Definition RGB Camera. Also, it can take Thermal, Lidar and Hyperspectral data depending on the payload (sensor) which is fixed with the unit. By using Aerial Photos, a lot of Machine Learning and Deep Learning applications like Object Detection and Feature Extraction becomes more possible and has achieved better accuracy than using the Satellite Imagery, especially when trying to identify very small and complex objects like (trees, building, roads, pipelines, flowlines…).

In this thesis, UAS has been utilized as a survey method to collect the required dataset over our area of interest. A thousand of High-Resolution Aerial Photos have been acquired with around 3cm ground sample resolution. These large datasets have been used in detecting and extracting the Oil Leaks from the Flowlines (onshore) using state-of-the-art Machine Learning and Deep Learning architectures.

1.1 Literature

With the development of Remote Sensing technology, the resolution of optical remote sensing images has greatly improved and images have become largely available. Numerous detectors have been developed for detecting different types of objects. Remote Sensing has benefited a lot from deep learning, particularly deep Convolution Neural Network (CNNs) in the past few years, mainly thanks to progress achieved in the computer vision community on natural RGB/Multispectral images. This literature review seeks to clarify the challenges/problems and discusses specific and general solutions of different Deep Learning approaches applied in feature extraction and object detection, and concludes that a more initiatives are required to further improve the network's detection ability.

In recent years, Deep Convolutional Neural Networks (DCNN), have achieved an astonishing level of performance in object detection and feature extraction. However, object detection is one of the most fundamental and challenging problems. Huang, addressed the traditional DCNN methods which don't focus on multispectral remote sensing images with more than three channels, and they are limited by their training samples [1]. Deng, studied the challenges limit the applications of Faster Region Based Convolution Neural Network (FRCNN). Deng stated that FRCNN has poor localization performance with small objects and the manual annotation is generally expensive and not sufficient in number [2]. Same as Ding who also addressed the FRCNN framework which does not yield a suitably high precision [3] .While Audebert investigated various methods to deal with multi-modal and multi-scale remote sensing data for semantic labeling [4]. Binaghi, investigated the use of contextual classification procedures and built adaptive recognition model to integrate feature extraction and classification over a range of multi – scale approach [5].

A variety of specific techniques have been suggested and validated to address the problem of DCNN in feature extraction and object detection. For example, Huang proposed Semi-Transfer Deep Convolutional Neural Network (STDCNN) to overcome the weaknesses of DCNN [1]. And Audebert showed Fully Convolutional Networks can be adopted, are well-suited to the task and obtain excellent results [4]. Deng, proposed a unified and effective Deep CNN based approach for simultaneously detecting multi-class objects [2]. While Ding adopted Dense Convoluted Network to improve the precision, reduce the detection time and memory requirements [3]. Binaghi developed a novel contextual classification strategy for object recognition and proved that recognition benefits from the combined use of scale–space techniques and neural classification [5].

The literature reviewed different application of deep CNNs to feature extraction and object detection in remote sensing images. The task of feature extraction and object detection has attracted considerable research attention in recent years. At the same time, significant progress has been made in object detection by the application of deep learning. More initiatives are needed to develop an effective and well – suited DCNNs approach to obtain excellent and accurate results.

1.2 Problem Statement

Most of Oil and Gas Industries are suffering from too many issues and Oil Leaks is one of these critical issues. Oil Leaks can be at offshore from vessels while transporting the Oil or even from Flowlines and Exploration and Production Platforms. Also, it can be at onshore either from Tanks, Wells, Flowlines or Pipelines. These Oil Leaks can cause an environmental damage if it lately discovered and left for days or even weeks without taken and immediate action. It can be translated to money and consider as loss if it happens more frequent in weekly or monthly basis.

Petroleum Development Oman with its huge operations not far from this issue same as other Oil and Gas Industries. PDO operates in a concession area of about 90,000 km2 (one third of Oman's geographical area), has around 130 producing oil fields, 14 producing gas fields, around 12,000 active wells. With this large coverage and huge operations, Oil Leaks most likely to happen. Currently, the process used for detecting Oil Leaks is by driving along the corridor of the Flowlines and report back any observed leaks. This current process is consuming a lot of time, cost, requires more staff and has a potential of harm people and it is not good enough for early spill containment where the quick response and the action is required.

PDO, is looking for a proactive way to do a visual inspection along these Flowlines in a short period of time and in a weekly basis or even in a daily basis in some area where the potential of having the Oil Leaks is high. One of these proactive ways is to use Drone and utilize the Unmanned Aerial System which is fixed with High Definition RGB Camera and fly over these Flowlines in frequent basis. Due to the large area coverage, there will be a large number of Aerial Photos which will be acquired in a daily basis and will require someone to go through each individual photo and inspect which will take a lot of time and consume manpower. So, this thesis is addressing this challenge and proposing Deep Learning as a way of speeding up the process of detecting the Oil Leaks from these Aerial Photos immediately after the UAS survey along the Flowlines is completed and help the industry to take a quick action.

1.3 Document Thesis Structure

The thesis document begins with Chapter 1 as an introduction of the importance of utilizing Unnamed Aerial System and the Aerial Photos for data collection and its largely uses in Deep Learning to solve several problems. Chapter 1 also, stated the problem which trying to solve here and the thesis document structure. Chapter 2, illustrated the study area, the dataset used as an input to this study, the aims and objectives.

Chapter 3, focused more in background and theoretical topics like Machine Learning and Deep Learning, what it is, what it is for, the different between them and its applications. Chapter 3 also, covered the Convolution Neural Network and the different types of learning and layers e.g. Convolutional Layer, Pooling Layer, ReLU layer, Fully Connected layer and Loss layer.

Chapter 4, included the proposed detection solution, the general framework for any Deep Learning model. It also, explained the methodology implemented here in this thesis. Chapter 5, covered the first implemented approach for the single raw images using Single Shoot Detector while Chapter 6 covered the second approach for the georeferencing orth-mosaiced images using MaskRCNN, U-Net and PSP-Net Classifier.

Chapter 7, described the validation and the evaluation strategy for both the first approach and the second approach including an explanation of the mean average precision and the confidence threshold which have been taken into account to evaluate the accuracy of the trained models. While Chapter 8 showed the final results of the deployment and the inference of all the trained models and their performance. It also, illustrated the Post-Processing to combine between the derived results from the trained models MaskRCNN, U-Net and PSP-Net Classifier to have a better result and improve the accuracy.

Chapter 9, addressed the challenges which been faced during the development of the DL models and the recommendations to overcome these challenges. Chapter 10, included a summary of the thesis on what have been achieved, some concluding remakes and future work.

2. Case Study

It is somehow difficult and a time consuming to visually go through a thousand and thousands of images trying to identify and detect small object. As a human been this may include errors vary from person to another. Machine Learning and Deep Learning technology is here to solve such problems with a high efficiency and accuracy. The case study here, is to build a Deep Learning models which are able to detect Oil Leaks from only Flowlines (onshore) from High Resolution RGB Aerial Photos which have been taken from Unmanned Aerial System (UAS). And this will help the industries to response quickly to take an action and maintain the leaks which will reduce the harm and the damage to the environment and to the asset as well.

2.1 Aim and Objectives

The aim of this thesis is to build a Proactive Monitoring System to detect Oil Leaks from Flowlines (Onshore) from High Resolution Aerial Photos in an automated way based on Machine Learning and Deep Learning and report any observations quickly to the response team and the decisions makers.

The objectives of this project are:

- Develop and implement an efficient ML / AI approaches to automate the process of detecting oil leaks and other anomalies from images.
- Build a Geo Spatial-database capturing any observation of oil spills and other anomalies
- Highlight and map the outlines of any leaks which been detected. Monitor the status of the cleanup process and ensure the process is completed.
- Publish the results into ESRI Portal and develop a Dashboard System to monitor the current status of the discovered anomalies.

2.2 Study Area

There are four areas have been considered while collecting the data using UAS for this study purpose. All of the areas are located within Sultanate of Oman which lies between latitudes 16° and 28° N and longitudes 52° and 60° E. These areas are within PDO and belonging to its concession boundary. The nature of the land of Area 1 and 4 is sandy as it is very close to the sand dunes while Area 2 and 3 is consider as rocky and stone land. The type of surface is mostly flat except some of sand dunes hills within Area 1.

The first area is located north east of Sultanate of Oman. The below figure (Figure 1) shows the location of the first study area.

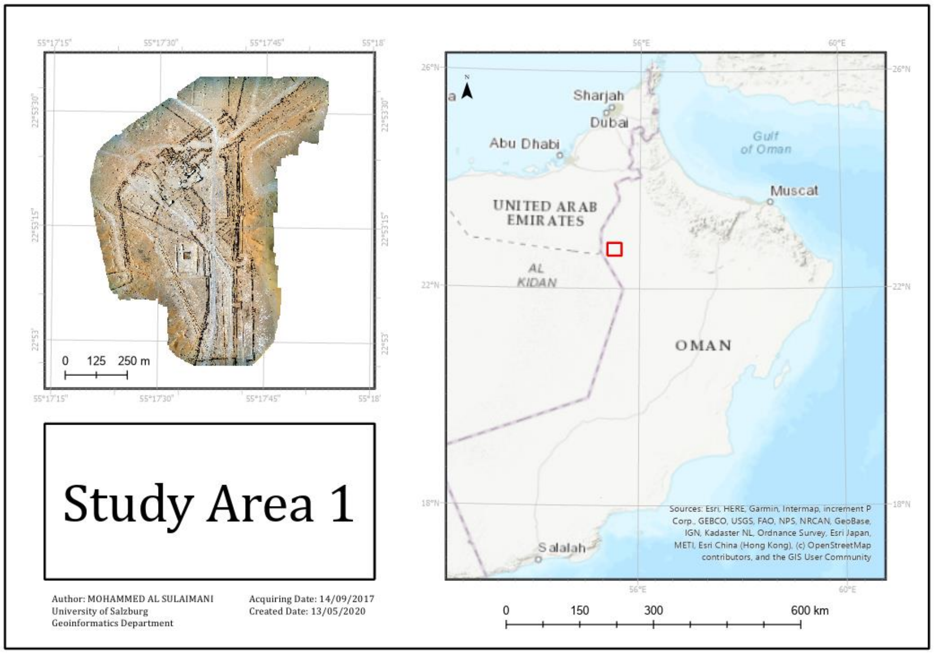


Figure 1. Map of Study Area 1, showing the location of this area within Sultanate of Oman

The second area is located to the south west of Sultanate of Oman. The below figure (Figure 2) shows the location of the second study area.

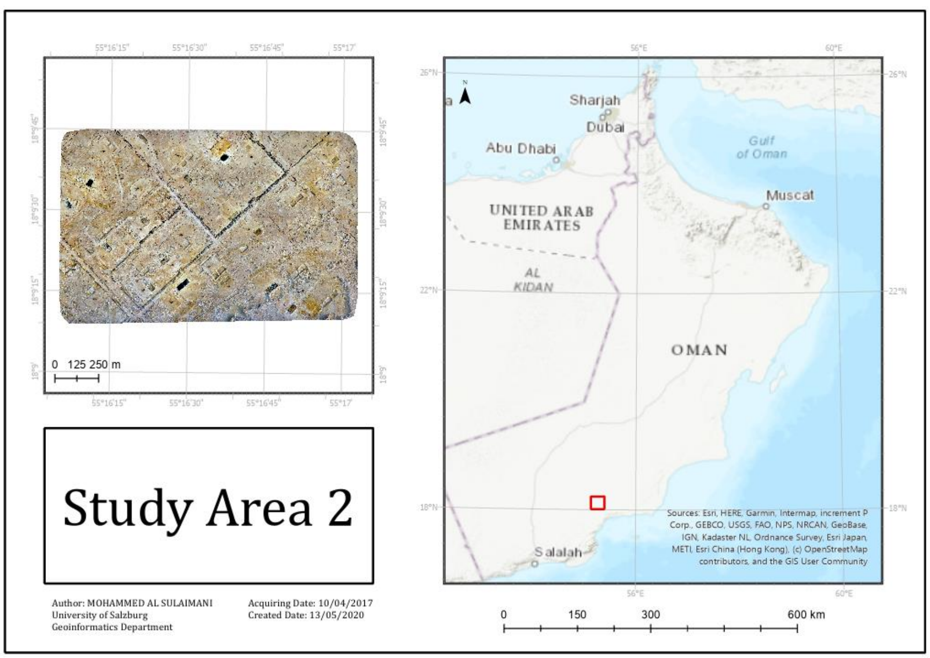


Figure 2. Map of Study Area 2, showing the location of this area within Sultanate of Oman

The third area is located to the south west of Sultanate of Oman close to Area 2. The below figure (Figure 3) shows the location of the third study area.

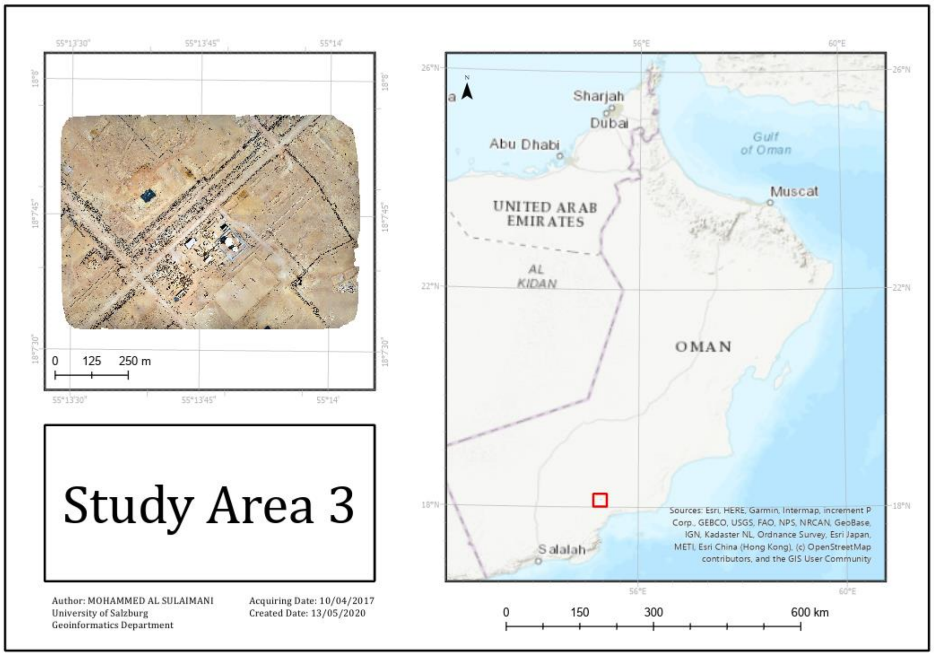


Figure 3. Map of Study Area 3, showing the location of this area within Sultanate of Oman

The fourth area is located to the north east of Sultanate of Oman close to Area 1. The below figure (Figure 4) shows the location of the fourth study area.

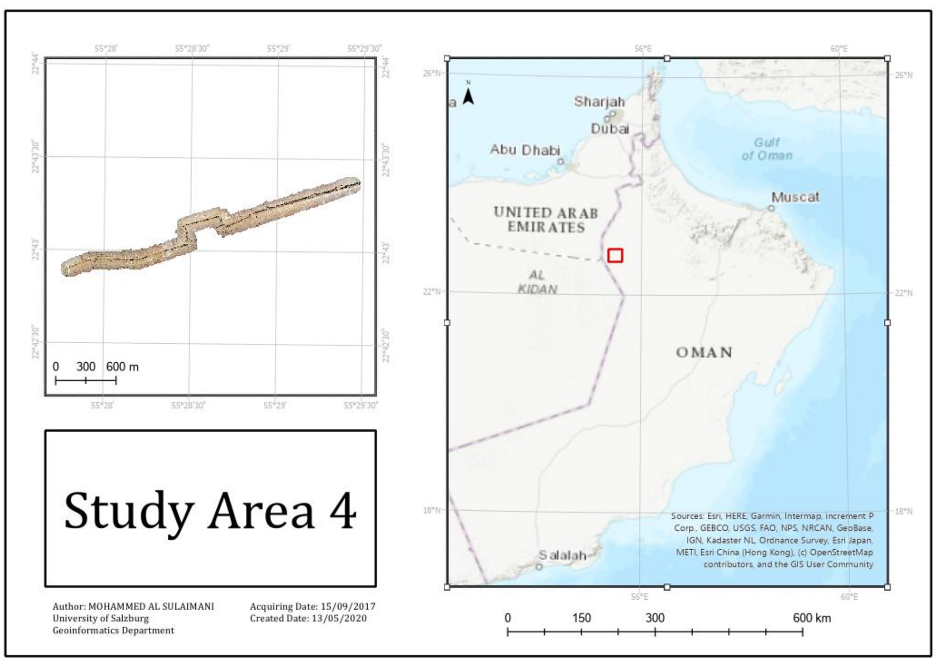


Figure 4. Map of Study Area 4, showing the location of this area within Sultanate of Oman

2.3 Data Acquisition

The data has been acquired using Unnamed Aerial System (UAS). The unit used to acquire the data is from Fix-wing type called "ebee" which can fly for longer time covering quite large area within one flight. It is considered as a professional surveying drone. The below table (Table 1) listed the specification of this unit.

Table 1. Specification of Sensefly ebee Fixing UAS

Hardware:	
Wingspan	110cm (43.3 in)
Weight	1.1kg (2.4lb)
Motor	Low-noise, brushless and electric
Radio Link Range	3km nominal (up to 8km)
Sensors	Parrot Sequoia, thermoMAP
Operations	
Automatic 3D Flight Planning	Yes
Cruise Speed	40 – 110 km/h
Maximum Flight Time	59 minutes
Ground Control Points (GCPs) required	No, RTX / PPK activated
Software	
Flight Planning & Control Software	eMotion 3
Image Processing Software	Pix4Dmapper

There are around 4651 RAW Aerial Photos been collected over the four areas. These Raw data have been processed and Ortho-mosaiced using Pix4Dmapper Software to generate a seamless Georeferenced Images. The following table (Table 2) listed more details of these images and the sensor which been used for data collection and the below figure (Figure 5) shows a picture of the drone used.

Table 2. Sensor and Aerial Photos Details

Sensor Details	
Payload	Sony (DSC-RX1R)
Sensor Type	Optical
Focal Length	35 mm
Aerial Photos Details	
Width	6000 pixels
Height	4000 pixels
Horizontal Resolution	350 dpi
Vertical Resolution	350 dpi
Color Representative	RGB
ISO Speed	ISO-100
Exposure Bias	0 step
Item Type (Format)	JPEG File
Ground Sampling Distance (GSD)	3cm

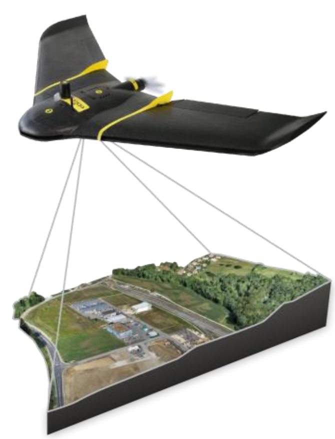


Figure 5. Sensefly ebee Fix-wing Unit

3. Background

Within the last 10 years and with the development in computer vision, humans were trying to build a technology that enables machines to solve a big task in a way it acts like humans, this is what it called Artificial Intelligence or AI. Some others are linking this to the behavior of humans. These machines are built and manufactured with a brain that can think like human and mimic their actions. Some of these techniques in computer science that used the Artificial Intelligence behind it and require human intelligence are translation between languages, speech recognition and prediction.

Machine Learning and Deep Learning are subset of the Artificial Intelligence and consider as Narrow Artificial Intelligence. Machine Learning is more related data and algorithms that allow computers to learn from examples without being explicitly programmed. While Deep Learning is a subset of Machine Learning and uses Artificial Neural Networks (ANN) as models to automatically builds a hierarchy of data representations. All of these concepts have contributed a lot to humans' life [6].

3.1 Machine Learning

Although the term of Machine Learning was presented in 1959, this technology only emerged as an era of Artificial Intelligence on 1980 [7]. Machine Learning as mentioned previously is consist of a logarithms and techniques that allow computer to learn from different examples without being explicitly programmed. One of the fundamental attributes of intelligence behavior is the ability to learn. These learnings are going to automatically improve with experience and by adding more examples to it [6].

3.1.1 Features (Pre-Processing)

A pre-processing step would be required to seeped up the computation as the computer needs to handle a huge numbers of pixels per second. So, the target is to search for useful features that are fast to compute and presenting some of these pixels directly to a complex pattern recognition algorithm might computationally be infeasible. The pre-processing should be done with a high level of careful as some of the information might be discarded and if this information is considered as highly important to the solution of the issue, then this will affect the overall accuracy of the system [8].

3.1.2 Generalization

Generalizability is the measure of the main outcome of machine learning and it represents the capability of the model to its prediction with any new data, so it is very important for the model to learns different patterns that are generalizable to achieve an accurate prediction whenever there are new data [7].

3.2 Deep Learning

Since the early 2000s, Deep Learning has become one the of the most popular approaches in Machine Learning after its high performance in solving the classical machine learning problems [7]. Deep Learning as a subset of Machine Learning techniques has showing a great successful for analyzing the data and solving complicated problems especially Convolutional Neural Networks. The first Artificial Neural Network (ANN) called the Stochastic Neural Analog Reinforcement Computer (SNARC) which consists of 40 nears has been developed by Minsky and Dean Edmonds on 1951 [6]. This technique, builds models based on ANN to automatically builds a hierarchy of data representations such powerful examples is Object Detection which is demonstrated in this thesis.

3.3 Types of Learnings

Both Machine Learning (ML) and Deep Learning (DL) involves four types of learnings. These types are Supervised Learning, Semi-Supervised Learning, Un-Supervised Learning and finally Reinforcement. The Supervised Learning is fully guided and controlled by human who are responsible to feed the model with an input (labeled) consider as a learning to the model while Un-Supervised Learning is not involving any human contribution as learnings (not labeled) while building them model and let the model learn by itself. The Semi-Supervised Learning is between the Supervised and the Un-Supervised Learnings making the use of both labeled and unlabeled data [9]. The fourth learning type is the Reinforcement which specialized in solving control or sequential decision-making problems. Its more complex than other learning types and deals with learning with feedback and via interaction [10]. the below figure (Figure 6) shows the types of learnings.

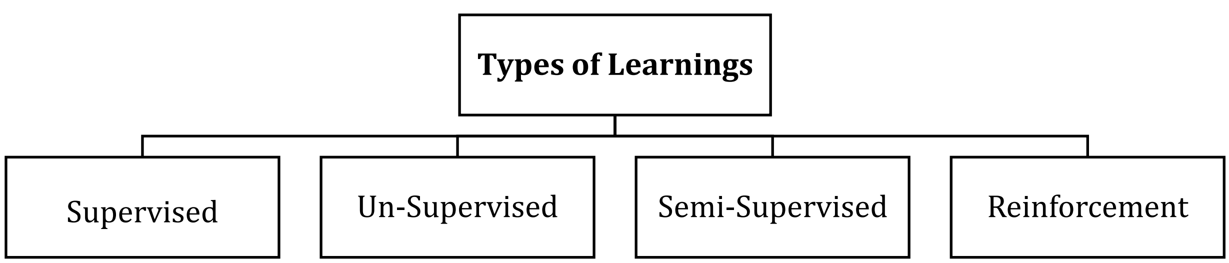


Figure 6. Types of Learnings

3.3.1 Supervised Learning

In Machine Learning and Deep Learning the typical way of learning is Supervised Learning where the learnings have been prepared and annotated by human guide. So, the process of learning here by assigning labels to some types of data while training the model [9]. The algorithms are learning by examples from already prepared information based on human observations [11]. This type of learning is involving two methods. The first one is Classification where the output values are discrete and having a defined value. The second one is Regression where the output values are continues [11]. Supervised Learnings is very powerful learning but it can be misleading by overfitting the model and if some patterns belonging to a parent population that differs from the one considered during the learning phase [12].

3.3.2 Un-Supervised Learning

Un-Supervised Learning comes at the second of learning types. Here, there the algorithm has not involvement or guidance from the human or the user. It analyzes the data (inputs). The computer will be asked to identify groups or clusters of similar patterns based of similarity of dissimilarity which can be ideally clearly distinguished from each other's. the algorithm is responsible to run the whole task [12].

3.3.3 Semi-Supervised Learning

Semi-Supervised Learning taken into account labeled and unlabeled inputs (data). It is obviously presumed that Semi-Supervised Learning is hybrid version of learning, so both humans and computer (algorithm) are involving in the process of Semi-Supervised Learning. The advantages of using this type of learning is that the fact data labeling process is often difficult, expansive and time consuming. Also, by using more precise decision boundary the model becomes more robust and the accuracy becomes much higher [9]. The below figure (Figure 7) shows the relation between data labeling process with the Supervised, Semi-Supervised and Un-Supervised Learnings.

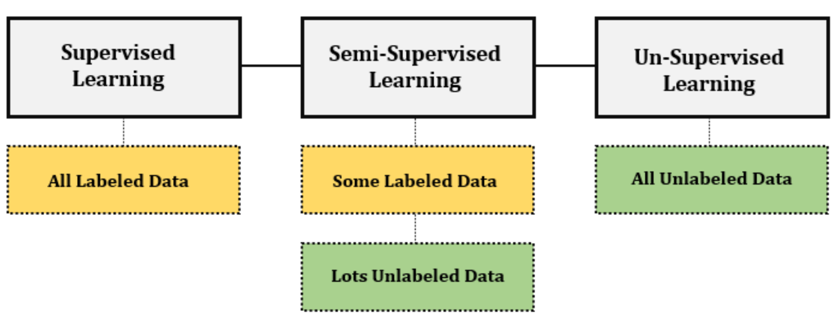


Figure 7. Classification of Supervised, Semi-Supervised and Un-Supervised Learnings [9]

3.3.4 Reinforcement Learning

The types of learnings differ from each other's in the terms of the kinds of feedback that the algorithm will receive after they make a prediction. Supervised Learning based on its ground truth labeled data the model will know how accurate its prediction. And as there is no labeled data within the Un-Supervised Learning, no feedback is provided. While the Reinforcement Learning is located between the Supervised and the Un-Supervised Learning, it will receive a delayed feedback [10]. Reinforcement which specialized in solving control or sequential decision-making problem and become more promising. The Reinforcement Learning is leveraging the recent and rapid developments of Machine Learning to make better control decisions [10].

3.4 Convolution Neural Network

One of the most effective method for image recognition and the first truly successful Deep Learning based on a multi-tiered hierarchical network is Convolutional Neural Network (CNN). It consists of mainly Convolutional Layer, Pooling Layer and Fully Connected Layer. These layers of a CNN are stacked to form a full convolutional layer. CNN is very suitable for image processing and can extract automatically rich correlation characteristics from images [13].

The below figure (Figure 8) shows the architecture of the CNN

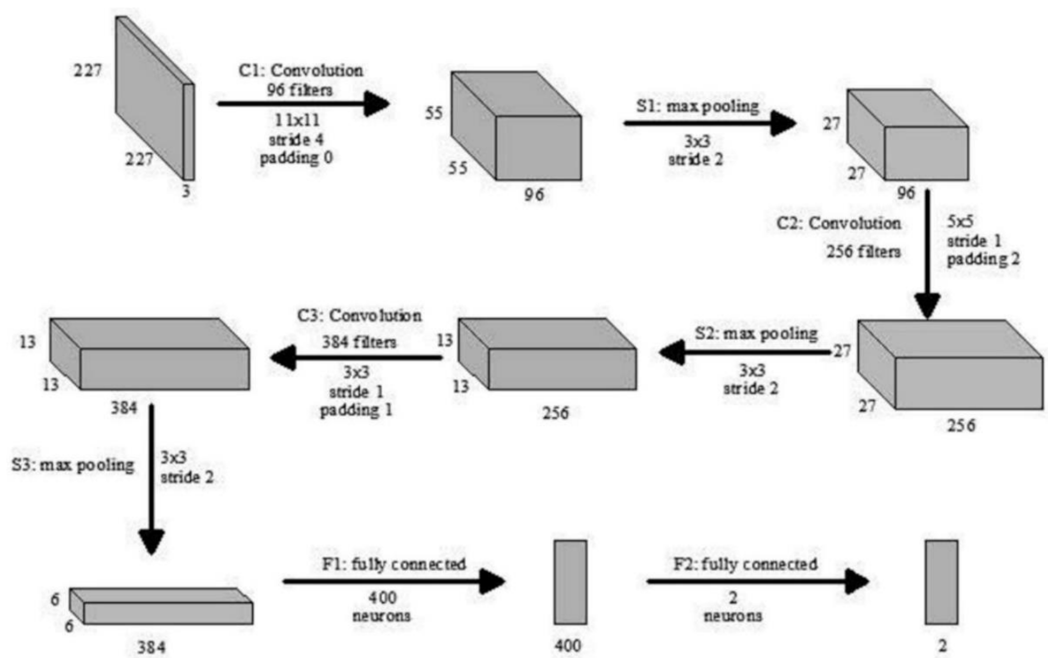


Figure 8. The Architecture of the CNN [13]

3.4.1 Convolutional Layer

The first layer in Neural Network perform convolutions instead of inner products. The most important part of the CNN is the Convolutional Layers which convolve the input data (images) with convolutional kernels to created different images. By increasing the number of the convolutional layers, the scale of the convolutional features becomes gradually coarser [14].

3.4.2 Pooling Layer

After the Convolutional Layer a Pooling Layer is usually applied to create a high computational cost since multiple convolutional kernels will increase the number of feature map [14]. Pooling Layers are used to produce a summary statistic of its input and reduce the spatial dimensions of the feature image [15]. The effect of the different pooling methods on the classification accuracy should be compared to optimize the training results [13].

3.4.3 Fully Connected Layer

After applying several Convolutional and Pooling Layers the Fully Connected Layers comes next. This means that each one of the neurons in any layer is directly connected to every node of the previous layer [15]. These layers are used as the final layers in object detection or classification problems and it works in reshaping all features into dimensions [14].

3.4.4 Activation Layer

Activation Layers is considered as additional layers and there are a number of Activation Layers within the architecture of the CNN. These layers inject nonlinear factors into the neural network to make the data distinguishable under the nonlinear conditions. The effect of the different activation functions is compared on the classification accuracy to optimize the training results [13].

3.5 Object Detection

Generic Deep Learning based object detection, currently with the state-of-the-art can be primally divided into two main groups: Two-Stage Detectors and One-Stage Detector [16]. In other words, the methods of the object detection are categorized into two types. One follows the traditional by generating region proposal firstly and then classifying each proposal into different groups. Second, treated the object detection as regression or classification [17].

3.5.1 Two-Stage Object Detection

The two-stage detectors framework divide the task into two processes. First stage is the generation of proposals. Second stage is making prediction for these proposals. Within the first stage the detectors will try to highlight regions in the input data (image) which may potentially be objects [17]. This process is trying to associate each object within the image to at least one of the proposed regions. While the second stage is utilizing Deep Learning to classify the proposals with the correct categorical labels. So, the object must belong to one of the predefined class labels or be as background (not defined) [17]. The two-stage object detection mainly includes Regional-based Convolutional Neural Network (R-CNN), Spatial Pyramid Pooling (SPP-net), Fast Regional-based Convolutional Neural Network (F-RCNN), Faster Regional-based Convolutional Neural Network (F-RCNN), Regional-based Fully Convolutional Network (R-FCNN), Feature Pyramid Network (FPN) and Mask Regional-based Convolutional Neural Network (Mask R-CNN)[16].

3.5.2 One-Stage Object Detection

The One-Stage object detection consider all positions within the input data (image) as a potential object [16]. This method doesn't have a sperate stage for proposal generation, so it is trying to classify each region of interest within the image as either target objects or a background within one process. Mapping straightly from input data (image pixels) to bounding box coordinates and class probabilities can reduce the time consume [16]. The one-stage object detection mainly includes MultiBox, AttentionNet, G-CNN and the two significant frameworks are You Only Looks Once (YOLO) and Single Shoot Detector (SSD) [17].

4. Proposed Detection Solution

This thesis is proposing the use of the powerful Deep Learning techniques in Oil Leaks Detection from High Resolution Aerial Photos which were acquired using the Unmanned Aerial Systems (UAS). This will help in developing and implementing efficient Deep Learning models who are able to clearly, fastly and accurately identify Oil Leaks from Flowlines (Onshore). Four different types of Deep Learning Models (architectures) have been considered as an object detection solution for Oil Leaks. To achieve an efficient result, the workflow has been divided into two approaches. The first approach used the raw images as an input to train the model while the second approach focus in Georeferenced Ortho-mosaiced images as an input to the model.

For the first approach, the one-stage object detection method is considered by using only the Single Shoot Detector (SSD) technique to detect the Oil Leaks from the raw images. SSD architecture is a very light, faster and state-of-the-art object detection technique [18]. This class of algorithm is called Object Detection. The target here is to be able to detect classes along with bounding box prediction in an image [19] .

For the second approach, the two-stage object detection methods are utilized by trying three different techniques (U-Net, PSP-Net Classifier and Mask R-CNN). U-Net and PSP-Net Classifier architectures are considering as Pixel-Based Classification or Semantic Segmentation that classify each pixel in an image to a particular class [20] [21]. Mask R-CNN architecture is considered as Instance Segmentation which integrates both the object detection with the bounding box task with semantic segmentation task. This allow us to detect the object first and then precisely segmenting a mask for each object instance [22].

4.1 Remote Sensing Deep Learning Framework

The framework of any Deep Learning application is somehow similar especially for object detection applications. In Remote Sensing, the Machine Learning or the Deep Learning framework is consisting of five important steps as describes at below figure (Figure 9):

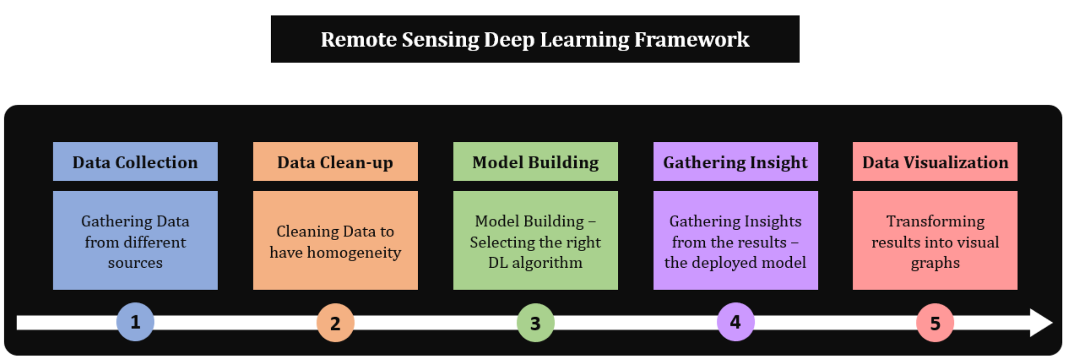


Figure 9. Remote Sensing Deep Learning Framework

4.2 Methodology

The below figure (Figure 10) illustrates the methodology which been demonstrated in this thesis for both Raw and Processed Aerial Photos for the first and the second approaches as mentioned previously.

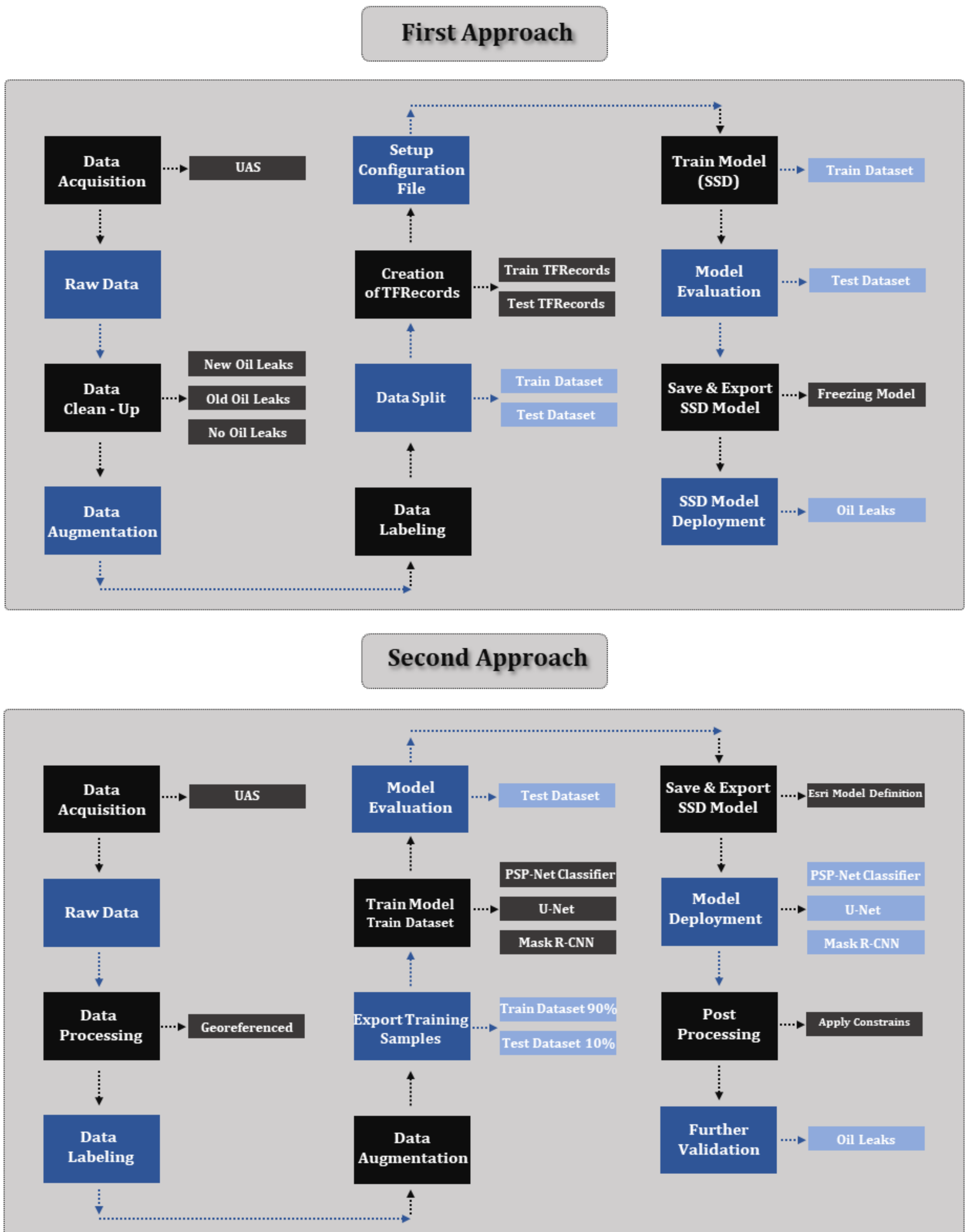


Figure 10. Methodology (RAW and Processed Input Data) [19] [20] [21] [22]

5. First Approach

The first approach is to build a Deep Learning Model using Single Shoot Multi Box Detector architecture (SSD) from Raw Images (Singles). SSD detects objects with one pass of the network. The DL Model is built by tuning and refining a pretrained model on the Common in Context dataset (COCO) [23], to work for our case Oil Leaks Detection.

5.1 Single Shoot Multi Box Detector

Most of the current state-of-the-art object detection works by either hypothesize bounding boxes, resample pixels for each box or apply quality classifier. the first Deep Learning architecture based on object detector which is not re-sample pixels achieve an accurate result as other do. This has speeded the detection process as there is no longer need for the resampling stage and by eliminating the bounding box proposals [19].

The advantages of using the Single Shoot Multi Box Detector is provided real time processing and high accuracy. It is also suited for small objects and for this case the image is sliced into some sub-images which are fed into the SSD network model. This process is enhancing the local contextual information and prevent these objects from being truncated [18].

SSD only required an input data (images) and ground truths boxes (labels) during the training process for each object. SSD can detect multiscale object high accuracy by applying default boxes with different aspect ratios [18]. In simple way Single Shoot Detector consists of two components: backbone Model and SSD head. The Backbone Model is also knowing as Transfer Learning and represents the preconfigure neural network (pre-trained image classification network) used as architecture for training the new model [14]. An example of Backbone Models is ResNet34, ResNet50 or ResNet152, these been trained on the ImageNet dataset contains more than million images. SSD head is represents one or some convolutional layers added to the Backbone [19].

The below figure (Figure 11), shows the Backbone with the first few layers (Boxes in White) and the rest few layers represents the SSD head (Boxes in Blue).

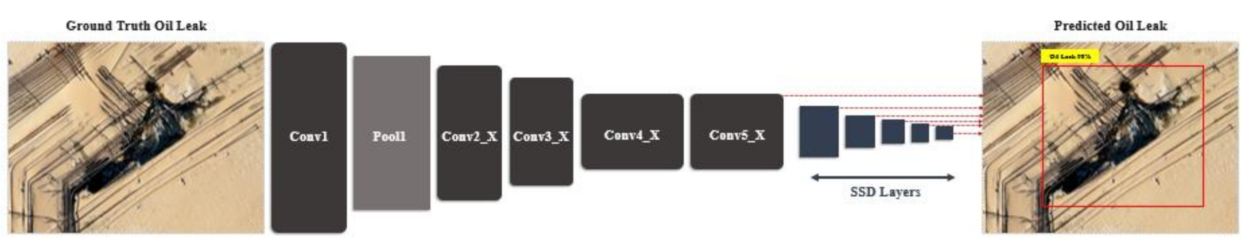


Figure 11. Architecture of a convolutional neural network with SSD Detector [19]

5.2 Hardware and Software

In this first approach the hardware and its specifications which been used to train the SSD Model, as per described on the below table (Table 4).

Table 4. First Approach Hardware Details and Specifications

Hardware:	
Workstation Model	Mobile Workstation-EliteBook 8570w
Processor	Intel (R) Core (TM) i7 -3720QM CPU 2.60GHz
Installed Memory (RAM)	24.0 GB
Windows Operating System	Windows 10
System Type	64-bit Operating System
Graphics Card (GPU)	NIVIDIA Quadro K2000M (2GB)

For the software, TensorFlow as Machine Learning Software from Google have been used in the first approach. TensorFlow is an end-to-end open source platform and it is freely available to everyone. JupyterLab is used to write the codes and train the DL Model. The below table (Table 5) describes the software's, dependencies and different required libraries along with the SSD Architecture to build Deep Learning Model which is able to detect the Oil Leaks from the Raw data.

Table 5. First Approach Software Details and Specifications

Software's	
Data Labeling Tool	Visual Object Tagging Tool (VoTT)
Environment	Conda
ML Platform	TensorFlow (GPU)
TensorFlow Model	TensorFlow Object Detection API
Dependencies	OpenCV, Python 3.6, JupyterLab
Other Libraries	Cython, Pillow, Lxml, Python-dotenv, conda-forge
Deep Learning Architecture	Single Shoot Multi Box Detector (SSD)
SSD Version	MobileNet
Pycocotools	Common Object in Context (COCO)

5.3 Data Clean-Up and Preparation

Data Clean-Up or Data Engineer is an important step for any kind of GIS or Remote Sensing applications including Deep Learning Application. In this step of the first approach, Raw data been prepared by removing any defocus or corrupted images which are not useful to be used to train the DL Model. Then, all photos been filtered and categorized based on the Oil Leaks. There were around 4651 Aerial Photos in total been surveyed over the four study areas. Around 210 images have new Oil Leaks, 104 images have an old Oil Leaks and 4337 images have no Oil Leaks. This step is needed to know and decide which images to consider as an input to the model. Here, the focus is to detect only the new Oil Leaks.

The below figure (Figure 12) shows that around 210 images from a total of 4651 images are having new Oil Leaks which repents 4.5%. This number of images is actually too low to be consider as an input (training samples), so a Data Augmentation step is required to increase the number of the training samples.

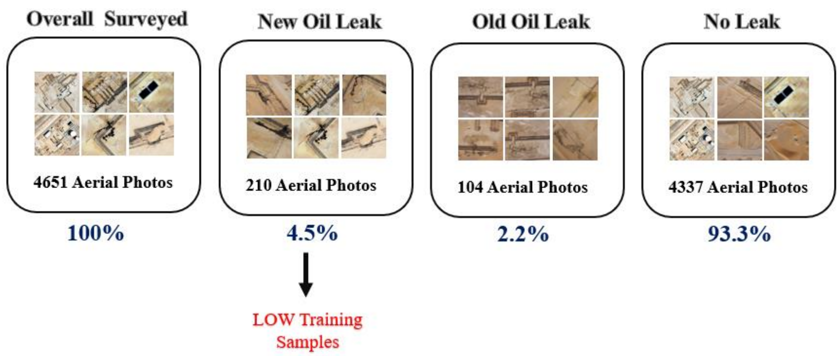


Figure 12. Data Clean-Up and Filtration of the Raw Data

5.4 Data Augmentation

Data Augmentation, is a strategy that enables practitioners to significantly increase the diversity of the data available for training models. This process is applied for each individual dataset to create newly dataset from the existing dataset which can be used as an additional input to train the DL Model. And as we do have a small number of images around 210 to be consider as a training samples, Data Augmentation is needed to increase the number of training samples. There are several techniques for the Data Augmentation like cropping, horizontal flipping, vertical flipping, rotating, zoom in or out, resize, change in contrast and others. The below figure (Figure 13) shows an example of one original image which been processed with three types of Data Augmentation techniques (Horizontal Flip, Vertical Flip and Horizontal & Vertical Flip) to create newly additional three dataset from the original image. By applying the same strategy to all the 210 images, another 630 images been created and ready to be used as an input to train the model.

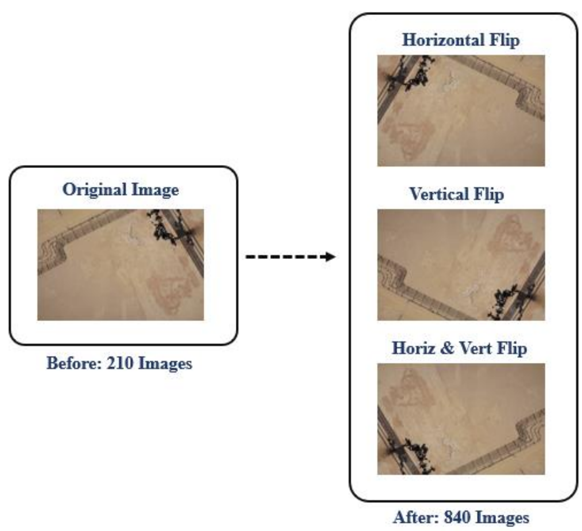


Figure 13. Data Augmentation of Raw Data

5.5 Data Labeling

Data Labeling is the most important step to build a Deep Learning model as any mistake on labeling the objects will lead to a confusion while training the model. Also, labels are considered to be a ground truth data during the evaluation process which tell whether the model is performing good or not and determine based on these ground truths the accuracy of the model and calculate the confidence rate. Thus, such step involves a lot of manual process to labels all objects. It is hard, difficult and time-consuming process. In this first approach with the SSD architecture a software called Visual Object Tagging Tool (VoTT) have been used to manually labeled all Oil Leaks present within the 840 images. The labeling process involves drawing a bounding box contain the Oil Leak and exported to an XML File Format structured as PASCAL VOC (Figure 14) which can be easily imported to TensorFlow using SSD architecture. For each imagery an associated XML file (labeling) is created. Labeling objects in the PASCAL VOC Bounding Box is obtained by assigning values of (xmin-top left, ymin-top left, xmax-bottom right, ymax-bottom right) to the location of the objects within the image.

Figure 14. PASCAL VOC Format Structure

The below figure (Figure 15) shows an example of the Data Labeling process for the Oil Leak and the associated created XML file (Labeling) which includes the class name “OilLeak” and the extent for the labeled Oil Leak.

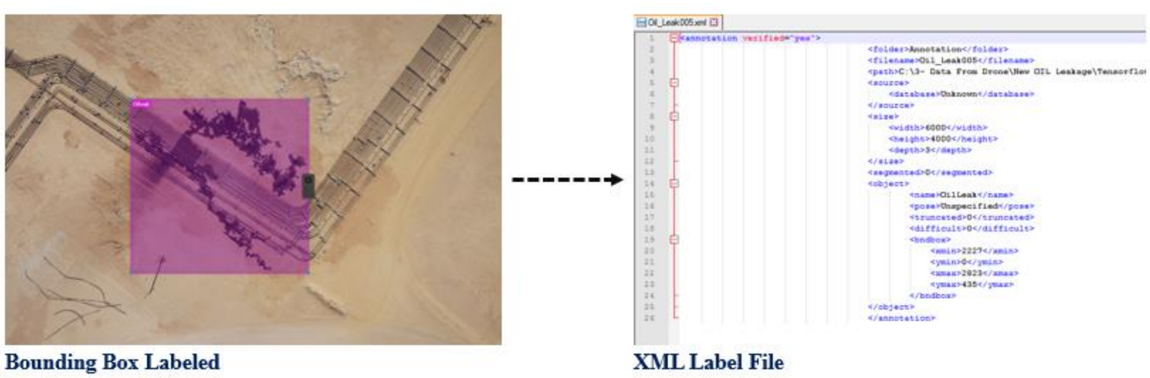


Figure 15. Data Labeling Raw Data (PASCAL VoC)

5.6 Training SSD Model

Before start training the model, it is required to divide the training sample (dataset) into two catogries, dataset to be used to train the model and another dataset to be used to test and evaluate the model. In this first appoach, around 5% of the total input data (trianing samples) are being used for testing and evaluating the performance and the accuracy of the model. This percentage can vary dependign on the input data (trainign samples), the more training samples, the high percentage to consider as a testing datsets.

While processing with the creation of Train Dataset and Test Dataset, it is required to transform these datasets into a format which can be imported by TensorFlow, for example TensorFlow Records (TFRecords). Here, a file configuration of a pretrained model on the Common Objects in Contents (COCO) is used. COCO is pretrained dataset consists of 91 objects types and includes around 2.5 million labeled instances in 328k images. The intention is tune and refine this pretrained model in our Oil Leak dataset [23].

Some setup is required on the Pipeline Configuration file of the COCO dataset so, it would work for our Oil Leak dataset. The setup and the change involve the below (Table 6) & (Figure 16):

Table 6. COCO Pipeline Configuration File [23]

COCO Pipeline Configuration File	
Number of Classes	Only one class (OilLeak)
Shape Resizer	300x300 (Height and Width). This will divide the images to a smaller chips size for a better performance.
Batch Size	6 (depending on the used GPU)
Learning Rate:	0.001 (Learning rate is a very important parameter, while training our model it will see the training data several times and adjust itself (the weights of the network). Too high learning rate will lead to the convergence of our model to a suboptimal solution and too low learning can slow down the convergence of our model.)
Steps or Epochs:	50,000 (The number of cycles of training on the data)

Figure 16. Change made on COCO Configuration File [23]

5.7 Trained DL SSD Model Deployment

The trained SSD model on TensorFlow is consists of four different files. The first file is "model-ckpt.meta" which contain the graphDef which describes the data flow, annotations, input pipliens and other information. The second file is "model-ckpt.data-0000-of-00001" which contains the values of the variables like weights, biases and other information. The third file is "model-ckpt.idex" which includes the metadata of a Tensor. The fourth file is "checkpoint" which contain all checkpoint inofrmation.
Any trained model on TensorFlow is containg all of these variables within these four files. To be able to use and deploy the model on any other softwrae or within the webserver a proccess called "Freezing" is performed to merge and save all of the required information into on single file that can be deployed easily. This process is knowing as Freezing the Graph, the single file is on ".pb extention". This file can be imported into other software (e.g. ArcGIS Pro using Python Raster Function and ESRI Defintion File) and run the inference on the Oil Leaks dataset.

6. Second Approach

The second approach is to build a Deep Learning Model using three different architecture (MaskRCNN, PSP-Net Classifier and U-Net) from Georeferenced Images (Ortho-mosaiced). PSP-Net Classifier and U-Net architectures are Pixel Based Classification or know as Sematic Segmentation that classify each pixel in an image to a specific class [21] [20]. While MaskRCNN architecture is an Instance Segmentation which consider both detecting the objects with bounding box and semantic segmentation [22]. From each architecture a Deep Learning model is built and able to detect the Oil Leaks from the Processed Imagery. Then, to increase the accuracy of the detected Oil Leaks, a Post-Processing step is performed by combining the results of the three models and by applying some constrains using other surveyed data like Flowlines and Fenced areas.

In this second approach the power of ArcGIS Pro and the ArcGIS API for Python are being utilized to build end to end Deep Learning Model using these platforms. This involves all steps from data labeling, exporting training samples, training the model and till the deployment of the trained model within ArcGIS Pro. A step by step guide with all Python Codes using Jupyter Notebook is explained at the end of this Thesis Document within the Appendix.

6.1 PSP-Net Classifier

PSP-Net Classifier stands from Pyramid Scene Parsing Network Classifier. One of the fundamental topics in Computer Vision is Scene Parsing based on Semantic Segmentation. Scene Parsing is able to predicts label, location and the shape for each object within the image [21]. Scene Parsing is based on the Fully Connected Network (FCN). This technique is facing challenge when it deals with complex input data (images). FCN based method is lacked of suitable strategy to utilize the clues of the global scene [21].

Most Semantic Segmentation models is consisting of two parts. The first one is Encoder which extracts out features from the input data (image) [24]. The second one is the Decoder which predicts the class of each pixel [25]. The Encoder in the PSP-Net Classifier comes at the first and contains the Convolutional Neural Network (CNN) with dilated convolutions along with the pyramid pooling module. While the Decoder comes at the end to extract the out feature from the input image and convert them into predictions through different layers [21].

Pyramid Scene Parsing Network is suitable for the global features. By considering both local and global clues this will make the prediction more reliable and accurate by designing global pyramids pooling. PSP-Net Classifier architecture is consisting of an Input Image, Feature Map, Pyramid Pooling Module and Final Prediction [21].

At the beginning CNN is used to get the feature map from the convolutional layer (Encoder). Then this feature map is pooled into different layer sizes followed by up sampling and concatenation layers (Decoder) to form the final prediction per pixel. the Pyramid Pooling Module is an important process as it helps to capture the global context in the image which then help in classifying the pixels. The below figure (Figure 17) shows the PSP-Net Architecture with its items [21].

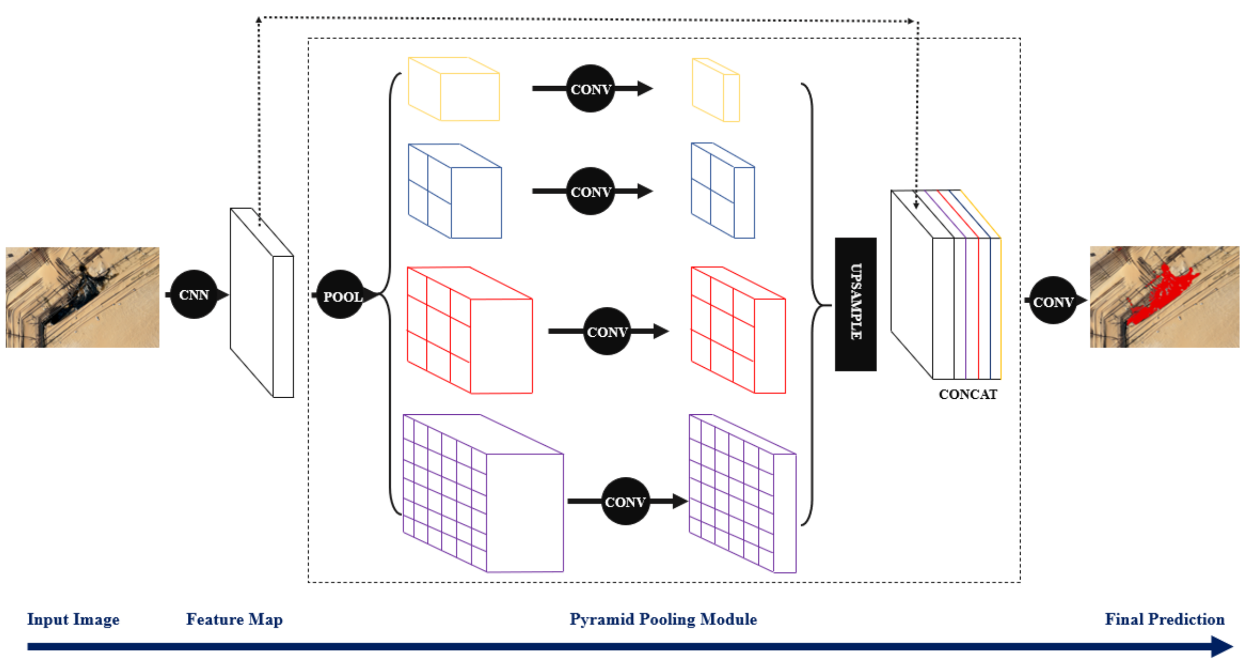


Figure 17. PSP-Net Classifier Architecture [21]

6.2 U-Net

U-Net is somehow similar to PSP-Net Classifier as it belongs to Semantic Segmentation algorithms. U-Net architecture also contains Decoder at the first and Encoder at the end same as PSP-Net Classifier. The Encoder is representing a pretrained classification network like VGG/ResNet (which is applied here) [24]. It works to encode the input image into feature representations at multiple different levels by applying the convolution blocks followed by a max pool down sampling. While the Decoder comes next to up sampling and concatenation the feature map by regular convolution operations [20].

The U-Net architecture is illustrated on the below figure. The left side includes the typical architecture of a convolutional network and consists of a contracting path. It has a repeated application of two 3x3 convolutions and each followed by a rectified linear unit (ReLU) [20]. And then a 2x2 max pooling operation with stride 2 foe down sampling and at each one of the down sampling process the number of feature channels get doubled. The right side includes an expensive path and consists of up sampling of the feature map followed by a 2x2 convolution that halves the number of feature channels, a concatenation with associated cropped feature map and two of 3x3 convolutions where each followed by a ReLU [20]. The cropping step is important and necessary duo to the loss of border pixels in every convolution. At the final layer a one of 1x1 convolutions is used to map the feature vector to the desired number of classes. The U-Net architecture has in total 23 convolutional layers. The below figure (Figure 18) describes the U-Net architecture. Blue boxes represent the multi-channel feature map. While white boxes represent copied feature map [20]. The different operations illustrated by the arrows. Red arrow for max polling operation, Green arrow for up sampling convolution and others as it been described at the legend.

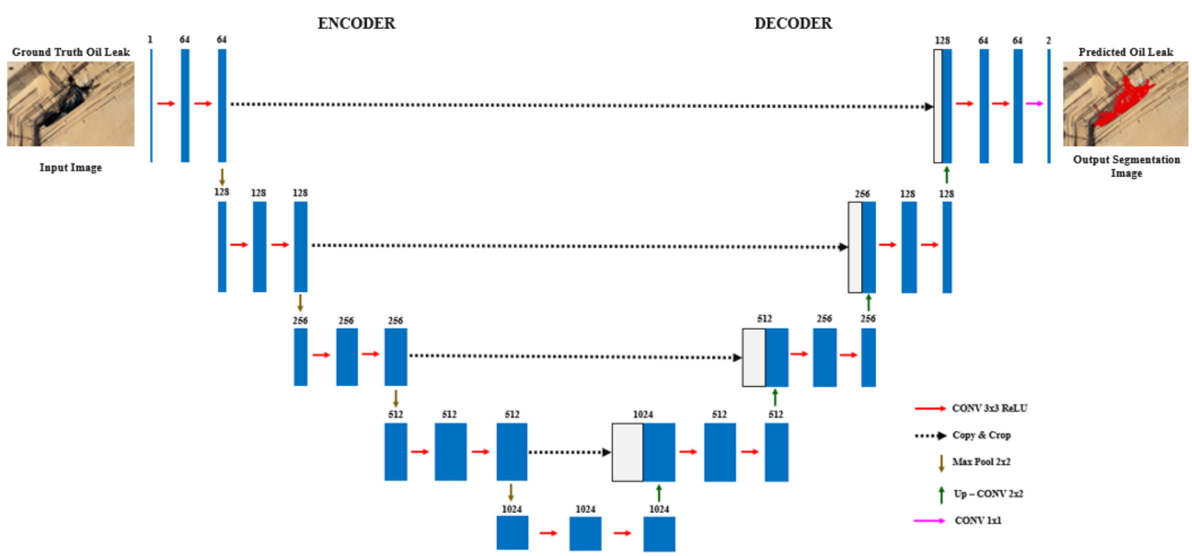


Figure 18. U-Net Architecture [20]

6.3 Mask R-CNN

Object Detection algorithm detects object within the input data (image) and find its object by bounding boxes, SSD is an example of Object Detection algorithm. Semantic Segmentation algorithm is a pixel level classification such as PSP-Net Classifier and U-Net. While Instance Segmentation algorithm integrates both Object Detection and Semantic Segmentation. It detects the object with bounding boxes and classifies each pixel into predefined classes, best example of Instance Segmentation is Mask Regional Convolutional Neural Network (Mask R-CNN) [22].

Mask R-CNN is actually an extension of Faster R-CNN which was not designed for pixel to pixel alignment between inputs and outputs. preserves the exact spatial locations using a simple quantization layer called RolAlign. Mask R-CNN uses the confidence of instance classification as the quality measurement indicator [22].

Mask R-CNN adopts two-stage procedure, it uses Regional Proposal Network (RPN) in the first stage while the second stage predict the class and the box offset. The below figure (Figure 19) shows the framework of the Mask R-CNN. It is clear that the Rol Pool in Faster R-CNN has been replaced with RolAlign in Mask R-CNN which helps to preserve the spatial information [26]. The output from the RolAlign will be faded into Mask Head which contain two convolution layers, a mask for each Rol which leads to a segmentation of an image in pixel to pixel process [22].

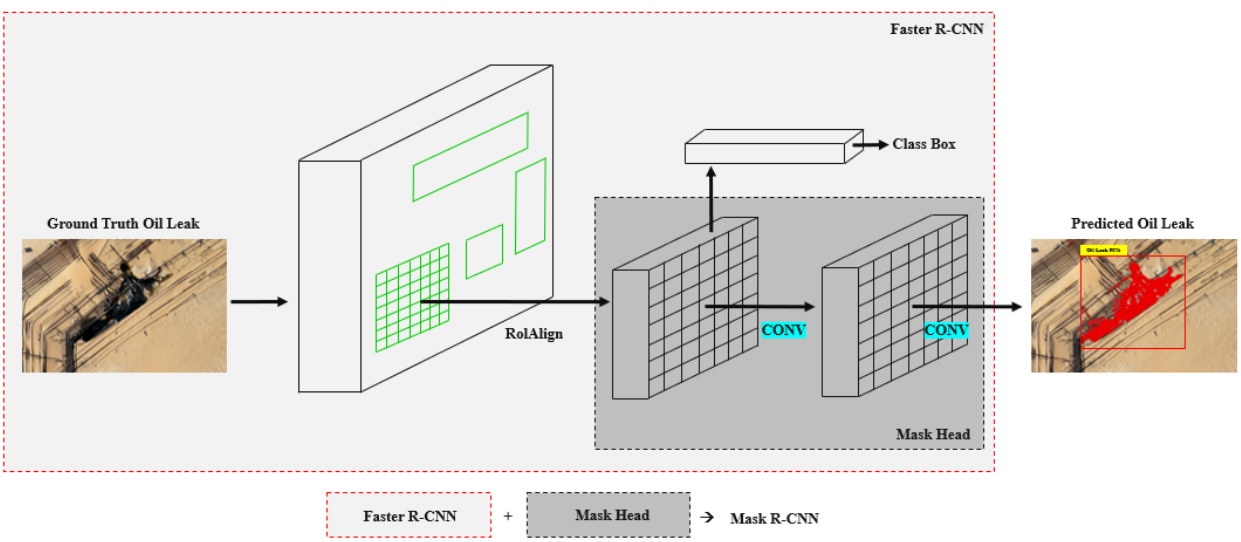


Figure 19. Mask R-CNN Architecture [22]

6.4 Hardware and Software

In the second approach the hardware and its specifications which been used to train the PSP-Net, U-Net and Mask R-CNN Models, as per described on the below table (Table 7).

Table 7. Second Approach Hardware Details and Specifications

Hardware:	
Workstation Model	PC Workstation-HP Z800
Processor	Intel (R) Xeon (R) CPU 2.79GHz (2 Processors)
Installed Memory (RAM)	96.0 GB
Windows Operating System	Windows 10
System Type	64-bit Operating System
Graphics Card (GPU)	NIVIDIA GeoForce GTX 1660 (6GB)

For the software, TensorFlow and PyTorch as Machine Learning Software have been used in the second approach. TensorFlow is an end to end open source platform from Google and it is freely available to everyone. PyTorch is also an open source platform from Facebook's Artificial Intelligence research group. Jupyter Notebook is used to write the codes and train the DL Model. In this second approach, all steps from data labeling till deploying the model have been done within ESRI products, ArcGIS Pro and ArcGIS API for Python as ESRI now is providing an end to end Machine Learning and Deep Learning solutions by utilizing third-party DL libraries. The below table (Table 8) describes the software's, dependencies and different required libraries along with the PSP-Net Classifier, U-Net and Mask R-CNN architectures to build Deep Learning Models which is able to detect the Oil Leaks from the Georeferenced Ortho-mosaiced data.

Table 8. Second Approach Software Details and Specifications

Software's	
Data Labeling Tool	ArcGIS Pro 2.5
Software's	ArcGIS Pro2.5 and ArcGIS API for Python
Environment	Conda - Python
ML Platform	TensorFlow (GPU) and PyTorch
Dependencies	OpenCV, Python 3.6, Jupyter Notebook
Other Libraries	Keras, fast.ai, scikit-image, Pillow and Libtiff
Deep Learning Architecture	PSP-Net Classifier U-Net Mask R-CNN

6.5 Data Labeling

As been described before Data Labeling is the most important step to build a Deep Learning model and it consume a lot of time as it is considered to be a manual process. In this approach, “Labels Object for Deep Learning Tool” within ArcGIS Pro 2.5 has been used to label all existing new Oil Leaks within the study area (Georeferenced Ortho-mosaiced images) as an “Feature Class File”. Then, this Feature Class along with the input data are used to create the Deep Learning Datasets using “Export Training Data for Deep Learning Tool”.

The output of this process is a folder contains image chips and another folder contains metadata files (labels) in a specified format by the user. Created training samples are based on a small subimages (chips) containing the feature or the class of interest to support third-party DL application such as Google TensorFlow, PyTorch or Microsoft CNTK.

Within the four study areas, there are around 10 new Oil Leaks, all Oil Leaks are being labeled except only one Oil Leak which is going to be used as a validation to test the performance of the trained model. The below figure (Figure 20) shows and examples of these Oil Leaks and its Labels.

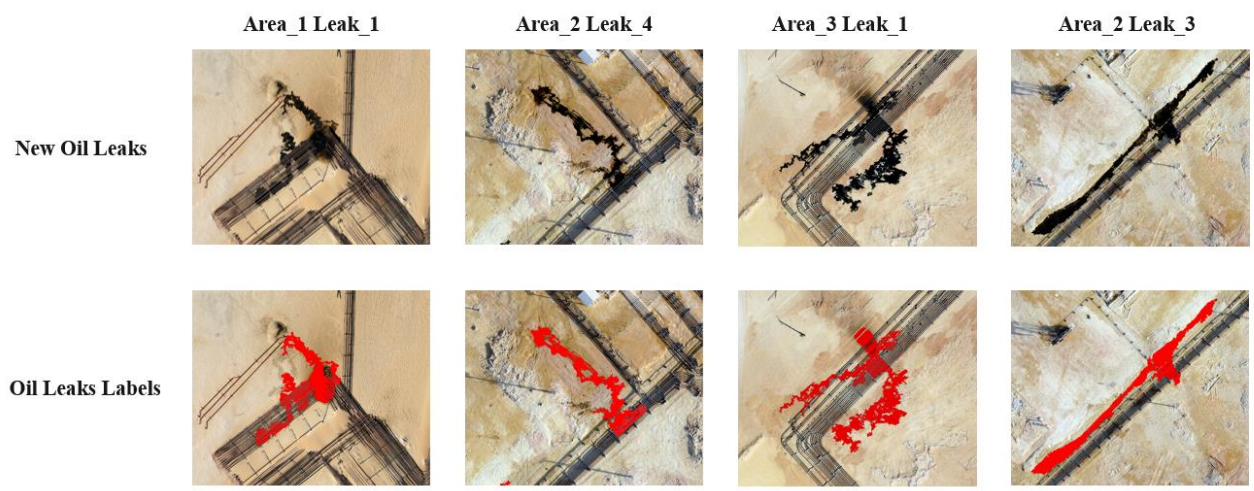


Figure 20. Data Labeling Georeferenced Images

The creation of the Training Samples is a critical step , the user have to specify some parameters like the image chip format, the size of the image chips for the X and Y dimensions, the distance to move in the X and Y direction of the next image chips, the format of the output metadata labels (KITTI_rectangles, PASCAL VOC, Classified_Tiles, RCNN_Masks or Labeld_Tiles), the type of the reference system to be used, the rotation angel that will be used to do data augmentation and generate additional image chips and specify the processing mode of all raster items either as a mosaic dataset or as an image service.

The Training Samples used for the PSP-Net Classifier and the U-Net are created with the same parameters as these architectures belonging to Semantic Segmentation and required similar format and parameters of the training samples. A total number of 32856 image chips have been created. The below table (Table 9) illustrates the training samples details.

Table 9. Training Samples Description for PSP-Net Classifier and I-Net

Training Samples for PSP-Net Classifier and U-Net	
Format	TIFF Format
Tile Size X	256 pixels
Tile Size Y	256 pixels
Stride X	128 pixels
Stride Y	128 pixels
Rotation Angle	45^0
Reference System	Map Space
Metadata Format	Classified Tiles

The below figure (Figure 21) shows an example of the Training Samples (Image Chips and Labels) created for the PSP-Net Classifier and the U-Net with 256x256 pixels for each image chip.

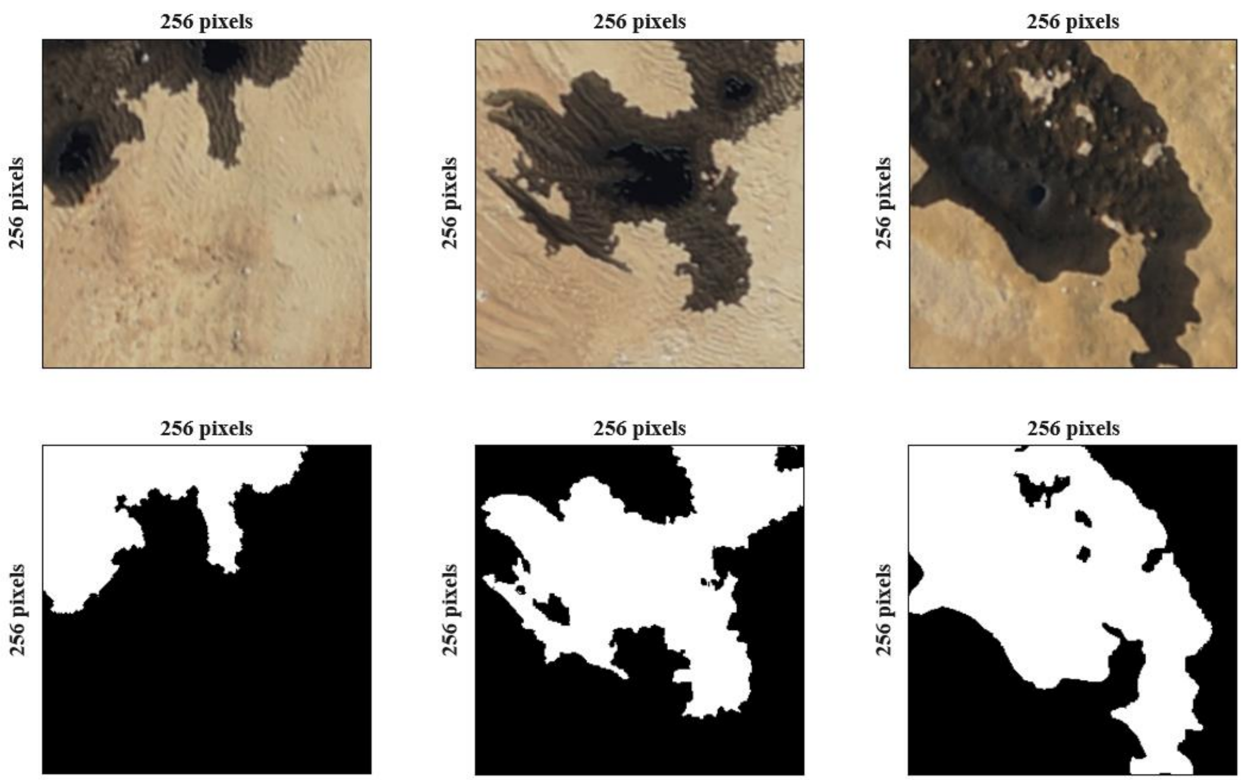


Figure 21. Image Chips for PSP-Net Classifier and U-Net

The Training Samples used for the Mask R-CNN are created with different Tile Size, Stride and Metadata Format. A total number of 23364 image chips have been created. The below table (Table 10) illustrates the training samples details.

Table 10. Training Samples Description for Mask R-CNN

Training Samples for Mask R-CNN	
Format	TIFF Format
Tile Size X	128 pixels
Tile Size Y	128 pixels
Stride X	64 pixels
Stride Y	64 pixels
Rotation Angle	45^0
Reference System	Map Space
Metadata Format	RCNN Masks

The below figure (Figure 22) shows an example of the Training Samples (Image Chips and Labels) created for the Mask R-CNN with 128x128 pixels for each image chip.

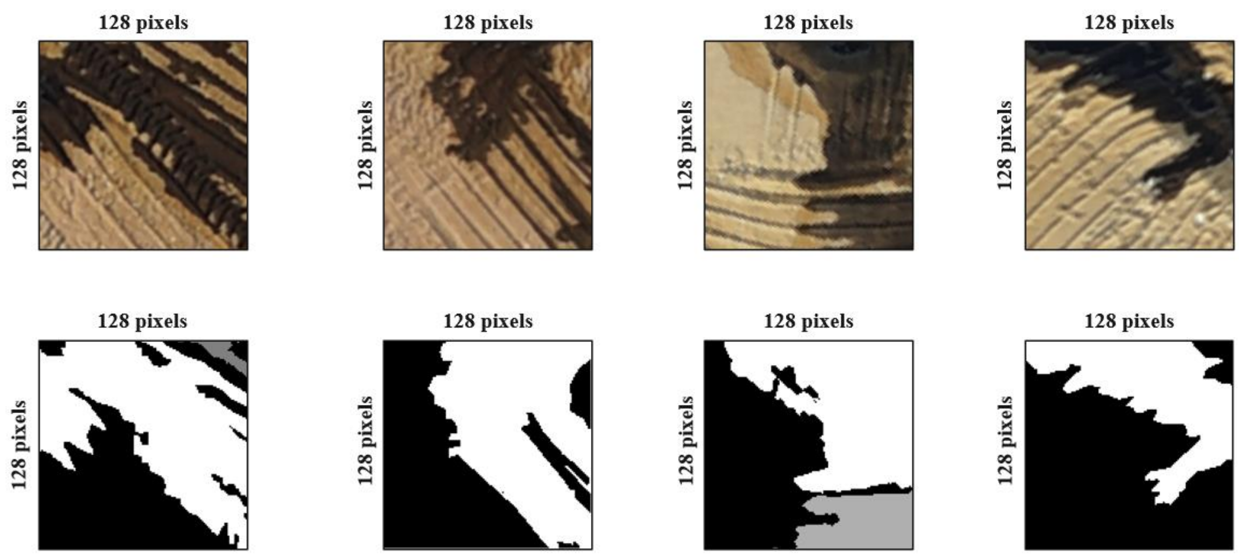


Figure 22. Image Chips for Mask R-CNN

6.6 Data Augmentation

Data Augmentation, is a strategy that enables practitioners to significantly increase the diversity of the data available for training models. And as explained before in (5.4), this process is applied to overcome the low number of the training samples. Here two ways of Data Augmentation been applied within the process of exporting the training data for the Deep Learning.

The first one is by setting the value of the Stride in both X and Y to half of what it been set for the Tile Size in both X and Y. For example, for Mask R-CNN the value of the Stride been set to 64 in X and 64 in Y, the value is half of the value of the Tile Size 128 in X and 128 in Y. When Stride value is equal to half the Tile Size value, there will be an overlap of 50% between the first image chip and the next image chip which leads into more creation of the training samples. And if the Stride value is equal to the Tile Size value, there will be no overlap. The below figure (Figure 23) shows an example of the overlap area between the first image chip and the second one as part of Data Augmentation process.

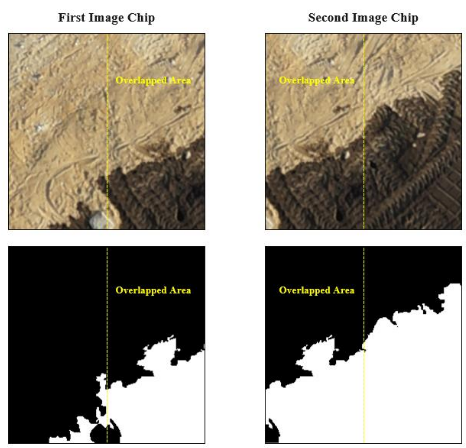


Figure 23. Data Augmentation of Georeferenced Data (50% Overlap)

The second one is by setting the Rotation Angle option to 45^0. This means that the image chip will be generated with a rotation of 45^0 to create an additional image chips and the same training samples will be captured at multiple angles of 45^0 in multiple image chips ensuring more training samples. In this case each original image chip will be rotated 7 times with 45^0 and create additional 7 image chips as it is illustrated in below figure (Figure 24).

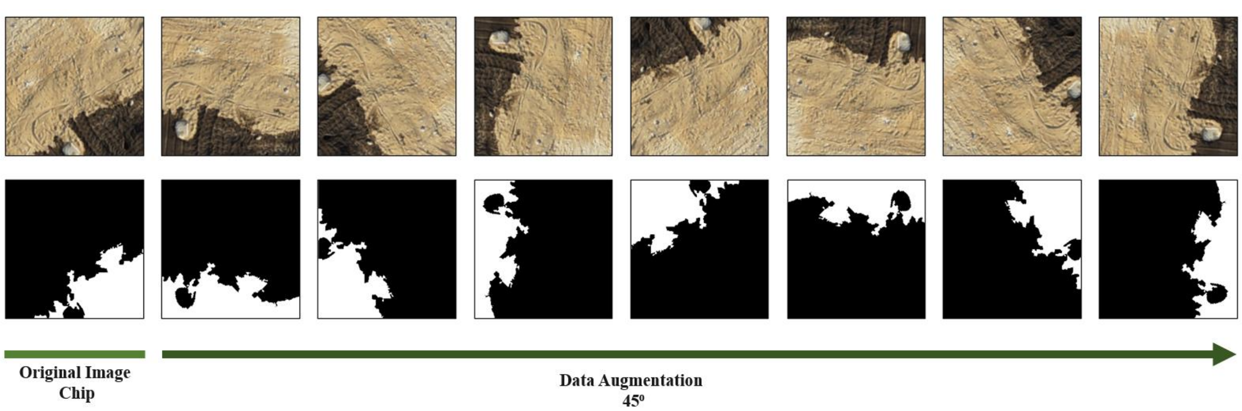


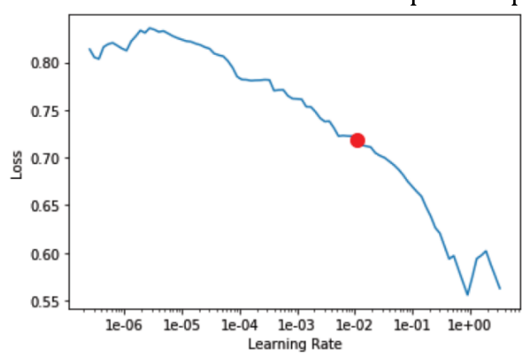
Figure 24. Data Augmentation of Georeferenced Data (45^0 Rotation)

6.7 Training the DL Model (PSP-Net, U-Net and Mask R-CNN)

Before start training the models, three different things need to be defined. Firstly, it is required to divide the training sample (dataset) into two catogries, dataset to be used to train the model and another dataset to be used to test and evaluate the model as been explained before in (5.6). In this appoach, by default around 10% of the total input data (trianing samples) are being used for testing and evaluating the performance and the accuracy of the model for all DL Models (PSP-Net Classifer, U-Net and Mask R-CNN).

Secondly, an important parameter to be defined at the beginning is the Learning Rate, this parameter is very important while training the model as it will see the training data several times and adjust itself (the weights of the network). Too high learning rate will lead to the convergence of our model to a suboptimal solution and too low learning can slow down the convergence of our model. In this second approach, a function from ArcGIS API for Python called "lr_find ()" being used to find the optimum Learning Rate. For the PSP-Net Classifier the optimum learning rate which been proposed by the function is 0.01096. the below figure shows (

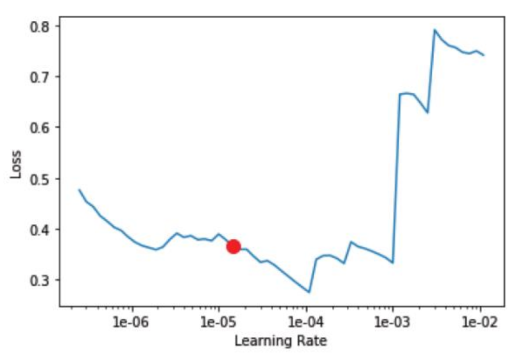
Figure 25) the output graph from this function and the red point represents the optimum learning rate.



0.01096478196143185

Figure 25. Optimum Learning Rate for PSP-Net Classifier Model

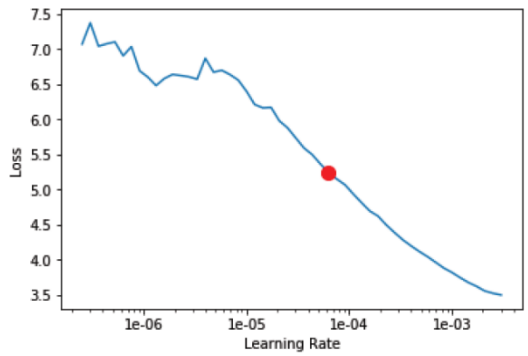
For the U-Net the optimum learning rate which been proposed by the function is 0.000001. the below figure shows (Figure 26) the output graph from this function and the red point represents the optimum learning rate.



1.4454397707459279e-05

Figure 26. Optimum Learning Rate for U-Net Model

For the Mask R-CNN the optimum learning rate which been proposed by the function is 0.0000006. the below figure (Figure 27) shows the output graph from this function and the red point represents the optimum learning rate.




```
6.309573444801929e-05
```

Figure 27. Optimum Learning Rate for Mask R-CNN Model

The third parameter which should be defined is the number of Epochs which represents the number of cycles of training on the input data (training samples). Here, all DL Models (PSP-Net, U-Net and Mask R-CNN) are been trained with 5 Epochs.

6.8 Deploying the DL Model (PSP-Net, U-Net and Mask R-CNN)

The trained DL Mdoels can be deployed in different ways. Firstly, it is required to save the trained DL model weights to create an ESRI Model Defintion and Deep Learning Package in a Zip format. Secondly, EMD can be used in ArcGIS Pro or ArcGIS API for Pyhon to run the inferance on the Oil Leak dataset while DLPK can be used in Image Server.

For the PSP-Net Classifer and the U-Net Models, an Image Classification python raster function is created and used to inference a PyTorch Image Classifier. The python file called “ArcGIS Image Classifier”.For the Mask R-CNN Model, an Instance Segmentation python raster function is created and used to inference a arcgis.learn deep learning model. The python file called “ArcGIS Instance Detector”.The trained models can be also be “Frozen”. This process is knowing as Freezing the Graph, the single file is on “.pb extention” as been explained before in (5.7).

In this appach all traine models been saved as an Esri Model Definiton and been deployed in ArcGIS Pro 2.5.

7. Validation and Evaluation Strategy

After training the DL model it is important to test and evaluate its performance. There are different ways to achive this depending on each DL architecure and each framework. In this theiss, several DL Models been built using several architecures (SSD, PSP-Net Classifier, U-Net and Mask R-CNN). The evaluation strategy is divided based on each approach (first and second).

7.1 First Approach

In the first apprach the training samples have been divided into two parts, Train and Test datasets. The Test dataset is equal to 5% (42 images) from the total training samples (840 images). The evalaution of the performance is based on this Test dataset. The first approach focused on tuning and refining a pretrianed model on the Common Object in Contacts dataset (COCO) [23] and the PASCAL VOC format [27] for the labels to build a new DL Model using Single Shoot Detector (SSD) which is able to detect Oil Leaks. There are 12 metrices used for chatactrizing the performance of an object detector on COCO dataset [23]. These metrices can be dvided into two groups.

The first one is related to the Precision metrices and includes 6 metries. The second one related to the Recall metrices and includes 6 metries. Before going through these metrices, it is good to understand what dose Precision, Recall and IoU means [23].

Precision measures how accurate the model is in predictions. In another word Precision means the percentage of our model prediction that are relevant. Recall is measure how good is the model in finiding all the positives from all cases, for example the model can find 85% of the positives from all positive cases. Intersection of Union or known as IoU measures the overlap area between the two bounding boxes of the ground truth and the prediction. To calcualet each one of these, four matematical defintion need to be understood (True Positive, True Negative, False Positive and False Negative) [27].

Intersection of Union (IoU), can be calculated by dividing the Area of Overlap of the ground truth and the prediction (bounding boxes) with Area of Union of the ground truth and the prediction (bounding boxes) as per below formula [27].

$$\mathbf{IoU} = \frac{\textbf{Area of Overlap}}{\textbf{Area of Union}}$$

True Positive represents a correct class which means that the model successfully detected the Oil Leak as an Oil Leak and the IoU is more than 50%. While True Negative represents also a correct class but the IoU is less than 50%. False Positive represents a wrong class which means that the model failed to detect the Oil Leak as an Oil Leak and consider it with another class but the IoU is more than 50%. While False Negative represents a wrong class and IoU is less than 50%. The below figure (Figure 28) shows how these mathematical definitions are between the ground truth and the predicted [27].

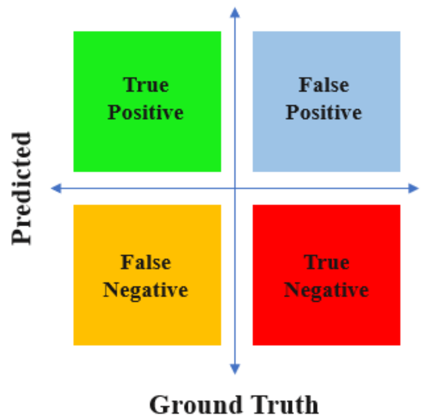


Figure 28. Predicted & Ground Truth | True & False | Positive & Negative [27]

Precision, can be calculated by dividing the True Positive with True Positive plus False Positive or in another way by dividing the True Positive with the Actual Results [27], as per below formula.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}} \qquad \text{Precision} = \frac{\text{True Positive}}{\text{Actual Results}}$$

Recall, can be calculated by dividing the True Positive with True Positive plus False Negative or in another way by dividing the True Positive with the Predicted Results [27], as per below formula.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} \qquad \text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}}$$

Then the Accuracy of the model can be calculated by dividing the True Positive + True Negative with the Total cases [27], as per below formula.

$$\text{Accuracy} = \frac{\text{True Positive + True Negative}}{\text{Total}}$$

After understanding the concept behind the mathematical definitions of the True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN) and how to use them to calculate the Precision, the Recall and the Accuracy based on the Intersection of Union (IoU), The below figure (Figure 28) illustrated this with the Oil Leak case [27].

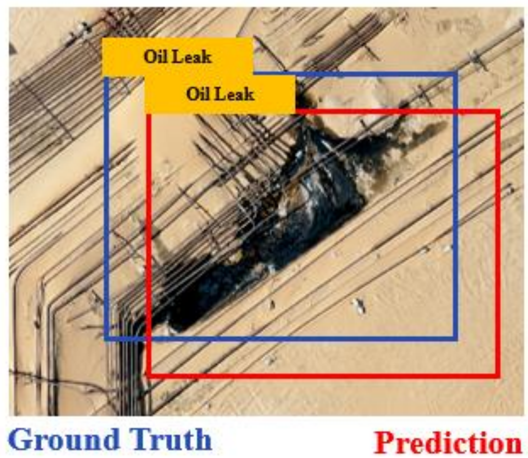


Figure.29. Ground Truth and Prediction | IoU – Oil Leak Example

The Precision and the Recall are always between 0 to 1. Thus, Average Precision (AP) or Mean Average Precision (mAP) is also between 0 to 1. The closest to 1 means the model is very accurate in predicting the objects [27].

There are 12 metrices used for characterizing the performance of an object detector in COCO dataset which been used here as pretrained model and refine it based on our training samples to build SSD model able to detect Oil Leaks. these 12 metrices corresponds to the Average Precision and the Average Recall. These 12 metrices describes at the below table (Table 11) [23].

Table 11. The 12 metrics used for characterizing the performance of an object detector on COCO dataset [23]

Average Precision (AP)	
AP	% AP at IoU=.50:.05:.95 (primary challenge metric)
$AP^{IoU=.50}$	% AP at IoU=.50 (PASCAL VOC metric)
$AP^{IoU=.75}$	% AP at IoU=.75 (strict metric)
AP Across Scales	
AP^{small}	% AP for small objects: area < 32^2
AP^{medium}	% AP for medium objects: 32^2 < area < 96^2
AP^{large}	% AP for large objects: area > 96^2
Average Recall (AR)	
$AR^{max=1}$	% AR given 1 detection per image
$AR^{max=10}$	% AR given 10 detections per image
$AR^{max=100}$	% AR given 100 detections per image
AR Across Scales	
AR^{small}	% AR for small objects: area < 32^2
AR^{medium}	% AR for medium objects: 32^2 < area < 96^2
AR^{large}	% AR for large objects: area > 96^2

In this study as PASCAL VOC is being used and for this the prediction consider a positive is Intersection of Union (IoU) is more or equal to 0.5 so while evaluating the SSD model the metric ($AP^{IoU=.50}$) will be considered for the Average Precision and as the area of the Oil Leaks consider as large the metric (AR^{large}) will be consider too [23].

7.2 Second Approach

In the second apprach for all DL architucres (PSP-Net, U-Net and Mask R-CNN) the training samples have been divided into two parts, Train and Test datasets while training the model. The Test dataset is equal to 10% from the total training samples. The evalaution of the performance is based on this Test dataset.

For this approach a graph been created showing the Training and the Validation (Test) losses plotted together to see how the model is performing during the training process till it finish. The less loss value the more accurate model. Good DL models are expected to have both Training and Validation losses somehow aligned close to each other and having the same trend.

Besides the Validation (Test) dataset, one Oil Leak been left and not been included on the creation process of the training samples and as well while training the model. This Oil Leak is used later on to further test the DL models created from (PSP-Net Classifier, U-Net and Mask R-CNN). the below figure (Figure 29) shows the Oil Leak which been not included within the training samples.

Figure 29. Area_1 Leak_2 for DL Models Evaluation

Confidence Threshold can be set while deploying the model (Mask R-CNN). Confidence Threshold works like the Mean Average Precision which been explained before in (7.1) and vary between 0 to 1. For example, by setting the Confidence Threshold to 0.85 while running the DL model, this means that the model will only detect Oil Leaks with a confidence more than 85%.

8. Results

In the first approach with the Raw Data, the trained SSD model has been deployed using the frozen graph (. pb extension) within JupyterLab. And for the second approach with Georeferenced Ortho-Mosaiced Data, the trained models (PSP-Net Classifier, U-Net and Mask R-CNN) have been deployed using Esri Model Definition (EMD) within ArcGIS Pro2.5

8.1 First Approach

The first approach includes only one Deep Learning model which is Single Shoot Detector architecture (SSD). The model been trained outside ArcGIS Pro and ArcGIS API for Python using CONDA Python environment and TensorFlow with other dependencies and necessary libraries.

8.1.1 Single Shoot MultiBox Detector (SSD)

SSD is an Object Based Detection, it generates a bounding box around the detected object within the input data (image) with the Confidence Threshold highlighted at the top of the bounding box for each detection.

Before running the SSD DL model and presenting the results, it is good to have a look at the Mean Average Precision, the Average Recall and the Loss. The below figure (Figure 30) shows the six metrices based on the Mean Average Precision. It is clear that the Average Precision at Intersection of Union with more than or equal to 50% is more than 0.85 at 50,000 steps (end of the training) but it reached close to 1 at 46,643 steps. this give us an indication on how is the trained SSD model is accurate and able to detect Oil Leaks.

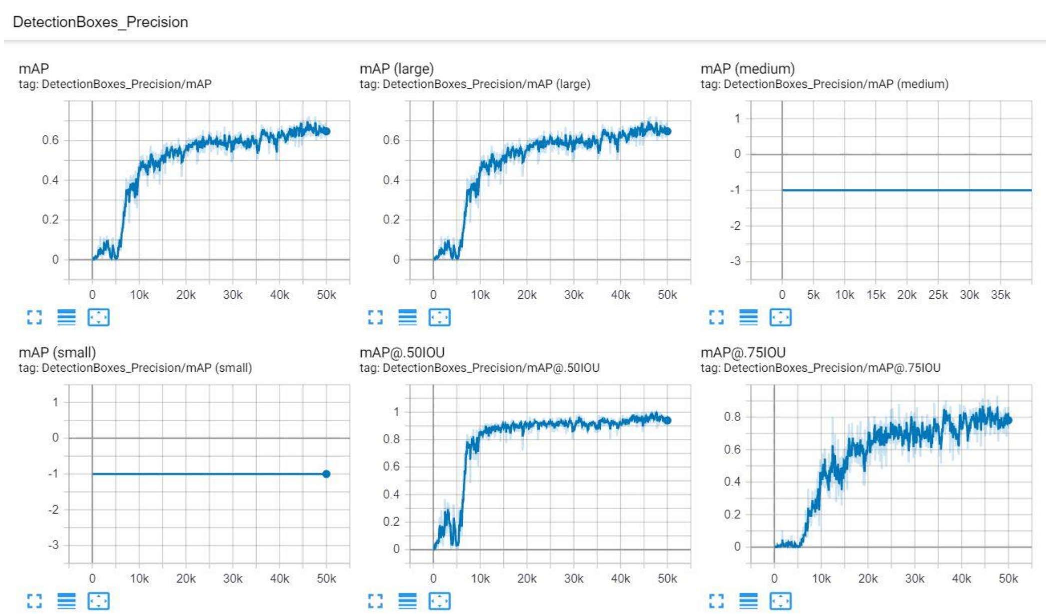


Figure 30. Mean Average Precision - SSD Model

And the below figure (Figure 31) shows the six metrices based on the Average Recall. It is clear that the Average Recall at for large object (area more than 96^2) is 0.72 at 50,000 steps and it reached to 0.76 at 46,643 steps. this give us an indication on how is the trained SSD model is accurate and able to find all positive Oil Leaks from all cases.

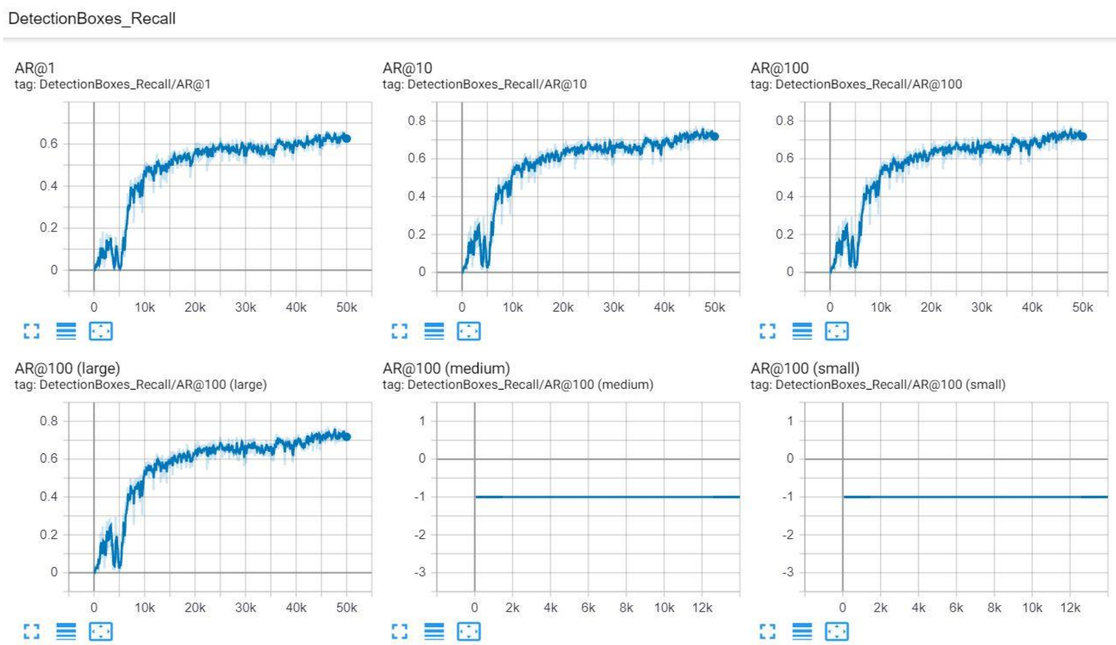


Figure 31. Average Recall - SSD Model

While the below (Figure 32) figure shows the Loss and how it decreased through training model and its learning. The lowest Loss represents the best refining model and here is 2.1 at 46,643 steps. The model at this step was saved as frozen graph and deployed to detect the Oil Leaks.

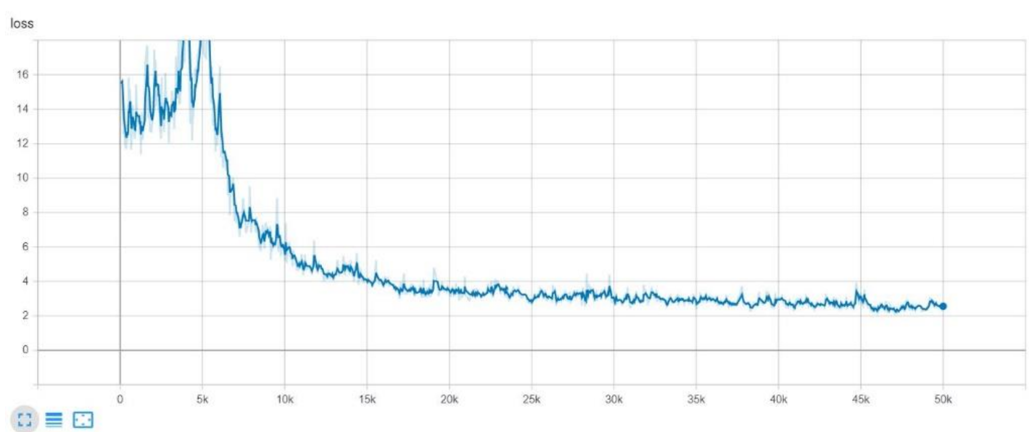


Figure 32. Loss - SSD Model

The below figure (Figure 33) shows some results of the trained DL SSD model in detecting the Oil Leaks. Here, the Confidence Threshold vary between 75% to 99% which is quite good.

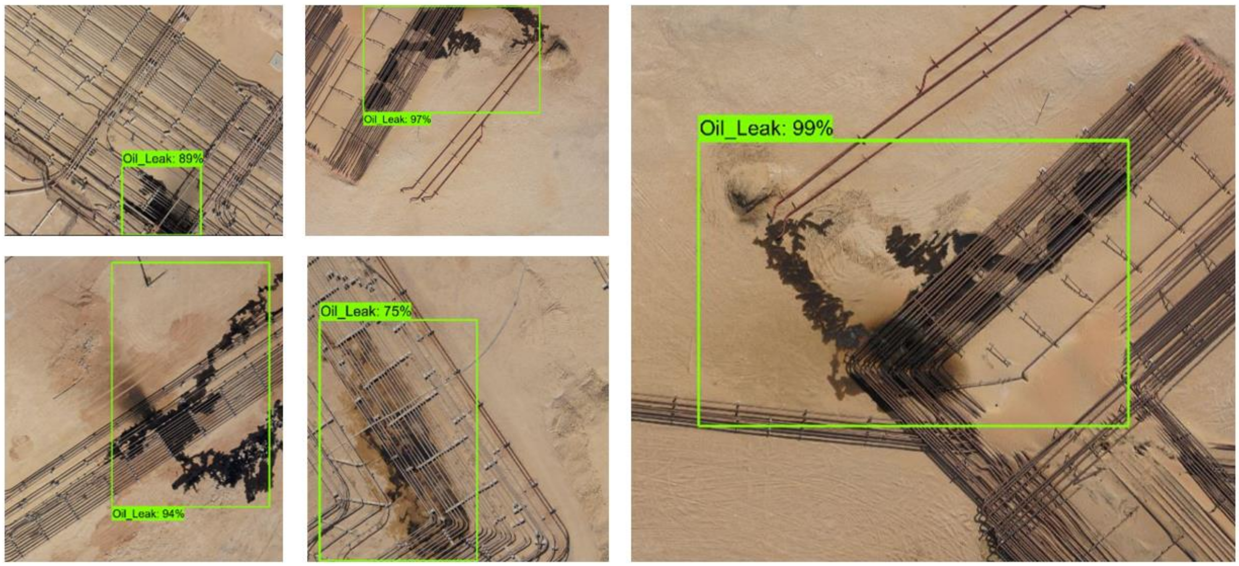


Figure 33. Detected Oil Leak - SSD Model

8.2 Second Approach

The first approach includes only three Deep Learning model which are Pyramid Scene Parsing Network (PSP-Net), U-Net and Mask Regional Convolution Network (Mask R-CNN). The models been built within ArcGIS Pro 2.5 and ArcGIS API for Python using Jupyter Notebook and PyTorch with other dependencies and necessary libraries.

8.2.1 Pyramid Scene Parsing Network (PSP-Net Classifier)

PSP-Net Classifier is an Object Detection belonging to Semantic Segmentation, it classifies each pixel within the input data (image) to a unique class [21]. PSP-Net Classifier model have been trained using Jupyter Notebook within ArcGIS API for Python (arcgis. learn) and deployed within ArcGIS Pro 2.5 using "Classify Pixel Using Deep Learning Tool" for all Oil Leaks areas.

The PSP-Net Classifier model have been trained for 5 Epochs and the below table (Table 12) illustrates the Training Loss and the Validation Loss for each epoch. Also, the time been taken to train the model for each epoch.

Table 12. Training and Validation Loss – PSP-Net Classifier

epoch	train_loss	valid_loss	accuracy	time
0	0.197607	0.174747	nan	1:37:53
1	0.156059	0.109922	nan	1:37:33
2	0.109755	0.303835	nan	1:42:26
3	0.091321	0.115462	nan	1:39:05
4	0.091815	0.120796	nan	1:37:28

And the below figure (Figure 34) plots the Training Loss and the Validation Loss within the 5 epochs. It can be observed that Loss for both Training and Validation dataset are close to each other except within epoch 2. This gives an indication on how the model is performing through the training process.

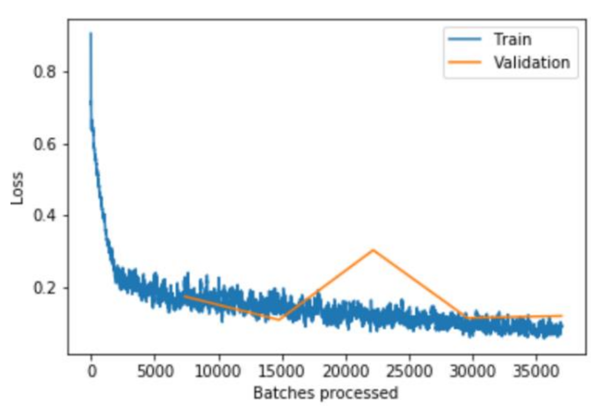


Figure 34. Training and Validation Graph Plot – PSP-Net Classifier

The first deployment of the PSP-Net Classifier is on the Oil Leak which was not been included on the creation of the training samples and the generation of the image chips. This will help in measuring the performance of the PSP-Net Classifier on a dataset which is considered as a new dataset (not seen before by the model). the below figure (Figure 35) shows that the trained PSP-Net model is able to detect the Oil Leak positively.

Figure 35. Test Oil Leak Classification Performance of the PSP-Net Classifier

Then the PSP-Net Classifier model been deployed with all existing new Oil Leaks within the study area (9 Oil Leaks). The below figure (Figure 36) shows that the PSP-Net Classifier also performed good in detecting the extent of the Oil Leaks.

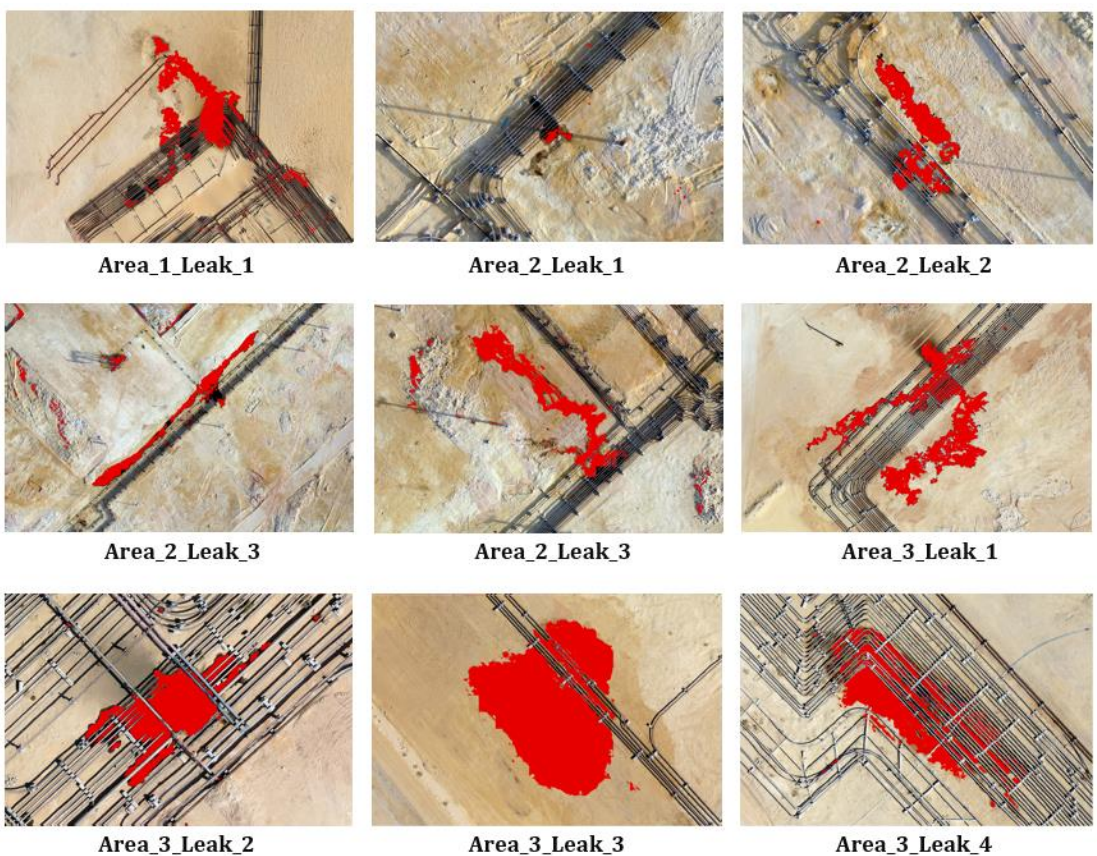


Figure 36. All Oil Leaks Classification Performance of the PSP-Net Classifier

8.2.2 U-Net

U-Net is an Object Detection belonging to Semantic Segmentation similar to PSP-Net Classifier, it classifies each pixel within the input data (image) to a unique class [20]. U-Net model also been trained using Jupyter Notebook within ArcGIS API for Python (arcgis. learn) and deployed within ArcGIS Pro 2.5 using "Classify Pixel Using Deep Learning Tool" for all Oil Leaks areas.

U-Net model have been trained for 5 Epochs and the below table (Table 13) illustrates the Training Loss and the Validation Loss for each epoch. Also, the time been taken to train the model for each epoch.

Table 13. Training and Validation Loss – U-Net

epoch	train_loss	valid_loss	accuracy	time
0	0.130735	0.134585	nan	57:22
1	0.112720	0.079924	nan	50:20
2	0.094387	0.053498	nan	50:42
3	0.097629	0.047091	nan	50:42
4	0.065857	0.049511	nan	50:59

And the below figure (Figure 37) plots the Training Loss and the Validation Loss within the 5 epochs. It can be seen that Loss for both Training and Validation dataset are close to each other and follow same trend This gives an indication on how the model is performing through the training process.

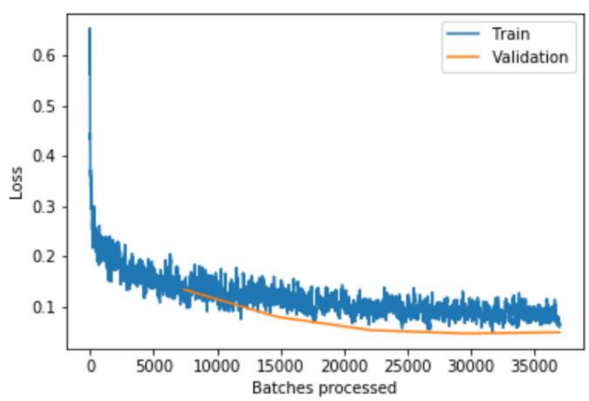


Figure 37. Training and Validation Graph Plot – U-Net

The first deployment of the U-Net is on the same Oil Leak which was not been included on the creation of the training samples and the generation of the image chips. This helps in measuring the performance of the U-Net on a dataset which is considered as a new dataset (not seen before by the model). This will also give us the opportunity to compare the classification performance of both PSP-Net and the U-net on the same dataset the below figure (Figure 38) shows that the trained U-Net model is able to detect the Oil Leak positively.

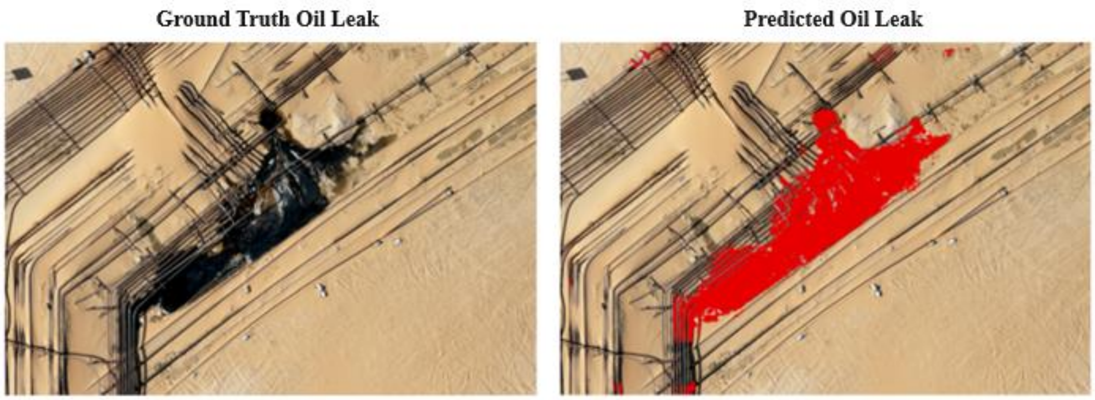


Figure 38. Test Oil Leak Classification Performance of the U-Net

Then the U-Net model been deployed with all existing new Oil Leaks within the study area (9 Oil Leaks). The below figure (Figure 39) shows that the U-Net also performed pretty good in detecting the extent of the Oil Leaks.

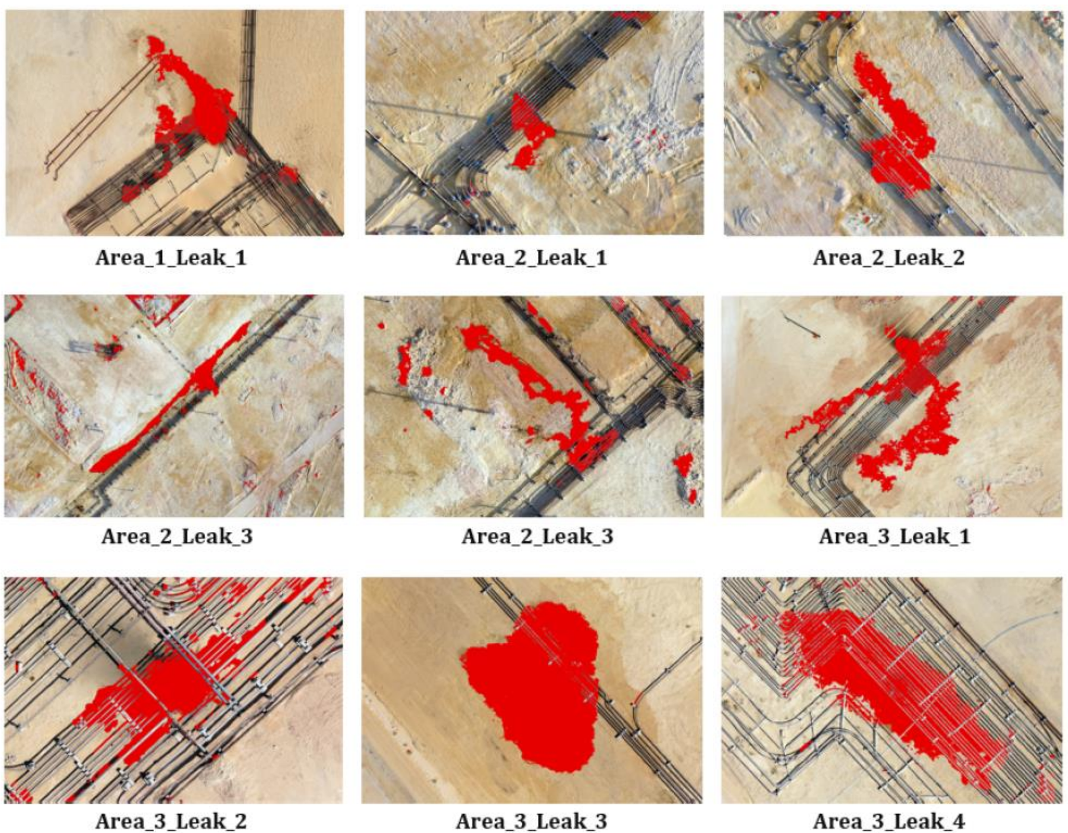


Figure 39. All Oil Leaks Classification Performance of the U-Net

8.2.3 Mask R-CNN

Mask R-CNN is an Object Detection belonging to Instance Segmentation not like to PSP-Net Classifier and U-Net which belong to Semantic Segmentation, it detects the object with bounding boxes and classifies each pixel into predefined classes. Mask R-CNN model also been trained using Jupyter Notebook within ArcGIS API for Python (arcgis. learn) and deployed within ArcGIS Pro 2.5 using "Detect Object Using Deep Learning Tool" for all Oil Leaks areas.

Mask R-CNN model been trained for 5 Epochs and the below table (Table 14) illustrates the Training Loss and the Validation Loss for each epoch. Also, the time been taken to train the model for each epoch.

Table 14. Training and Validation Loss – Mask R-CNN

epoch	train_loss	valid_loss	time
0	1.025442	1.003281	42:51
1	0.968297	0.911558	39:26
2	0.774006	0.837970	39:03
3	0.725213	0.799334	38:51
4	0.623179	0.809268	38:58

And the below figure (Figure 40) plots the Training Loss and the Validation Loss within the 5 epochs. It can be seen that Loss for both Training and Validation dataset are close to each other and follow same trend This gives an indication on how the model is performing through the training process.

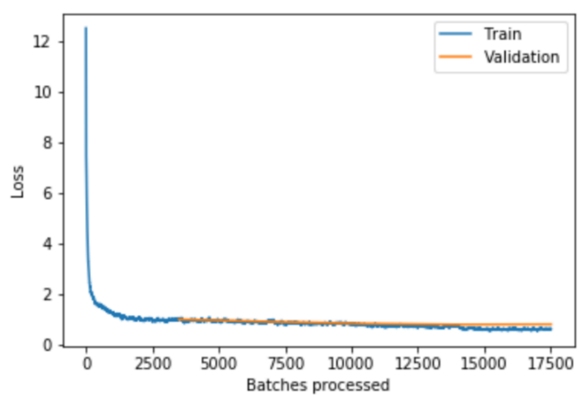


Figure 40. Training and Validation Graph Plot – Mask R-CNN

By following the same methodology has been done for PSP-Net Classifier and U-Net to test the model performance, Mask R-CNN is been tested first on the same Oil Leak which was not been included on the creation of the training samples and the generation of the image chips. This helps in measuring the performance of the Mask R-CNN on a dataset which is considered as a new dataset (not seen before by the model). This will also give us the opportunity to compare the classification performance of all trained models Mask R-CNN, PSP-Net and the U-net on the same dataset the below figure (Figure 41) shows that the trained Mask R-CNN model is able to detect the Oil Leak positively.

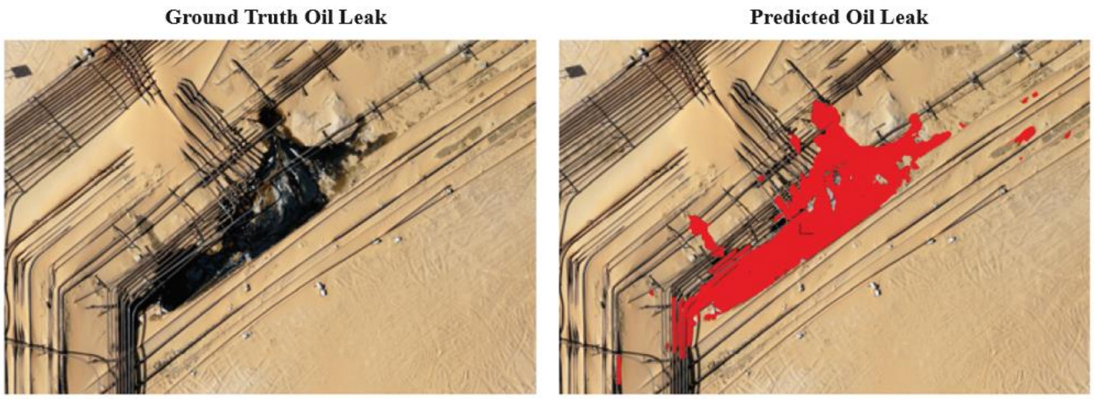


Figure 41. Test Oil Leak Classification Performance of the Mask R-CNN

Then the Mask R-CNN model been deployed with all existing new Oil Leaks within the study area (9 Oil Leaks). The below figure (Figure 42) shows that the Mask R-CNN also performed pretty good in detecting the extent of the Oil Leaks.

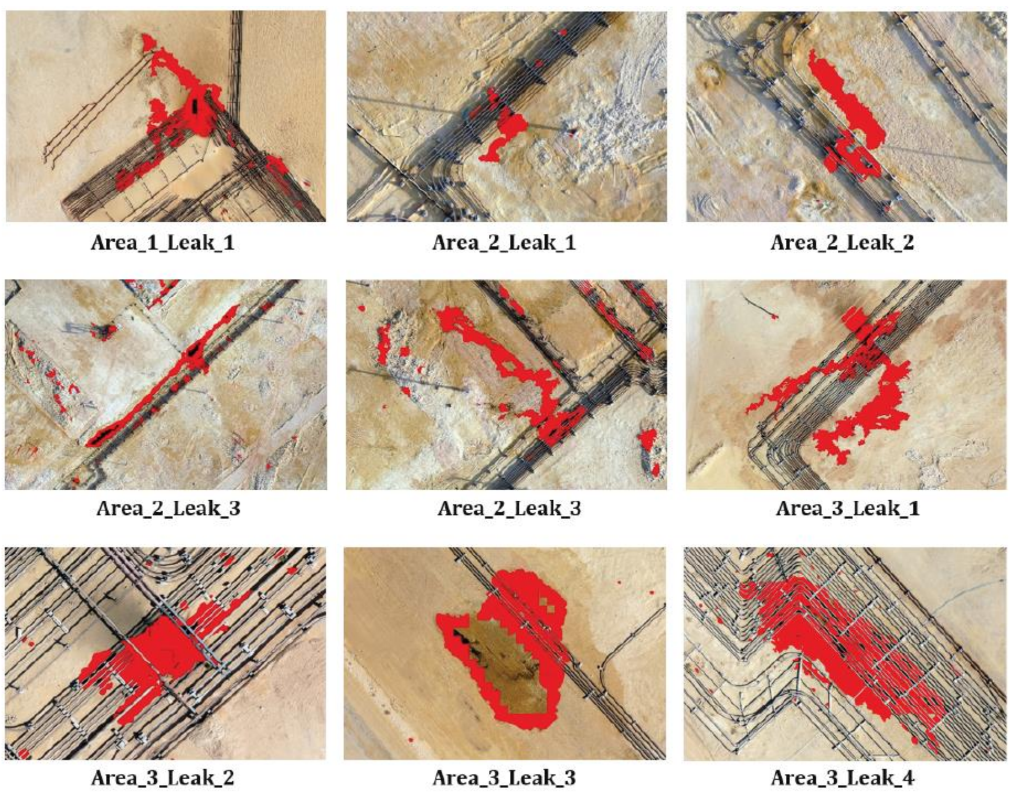


Figure 42. All Oil Leaks Classification Performance of the Mask R-CNN

8.2.4 Post Processing

It can be clearly seen from above figures (36, 37, 39, 40, 42 and 43) that there is some contribution of shadows which somehow been detected as an Oil Leak which is not correct. It is difficult for the model to distinguish between what is consider as an Oil and what is a Shadow. Thus, a further Post Processing was implemented for all trained model (PSP-Net Classifier, U-Net and Mask R-CNN) to minimize the present of shadows and eliminate bad detections. The Post Processing was implemented inside ArcGIS Pro 2.5 by using "Model Builder" and involved the below steps:

- Dissolve Boundaries and merge all areas that "shared the same boundaries"
- Remove Polygons which their area is greater than or equal to "1000-meter square" as these polygons consider as "bad detections and will be removed"
- Use the Flowlines Feature as a constrain, so any detections away from the Flowlines will be consider as a bad detection and will be removed.
- Use the Fenced Feature (Facilities) like "Stations or Buildings" as a constrain as these Facilities have contributed a lot to Shadows, so any detections within or touch Fenced Areas will be consider as a "bad detection" and will be removed.
- Apply the same previous constrains to all models (PSP-Net Classifier, U-Net and Mask R-CNN). Secondly, perform "Intersect Analysis" and consider all not matching detections from these three models as a bad detection and will be removed. Thirdly, combine all results from all models and consider them as a good detection (Oil Leaks).

The below figure (Figure 43) shows an example results from the Post-Processing and it can be clearly seeing a very good enhancement after eliminate the effect of Shadows and any other bad detections.

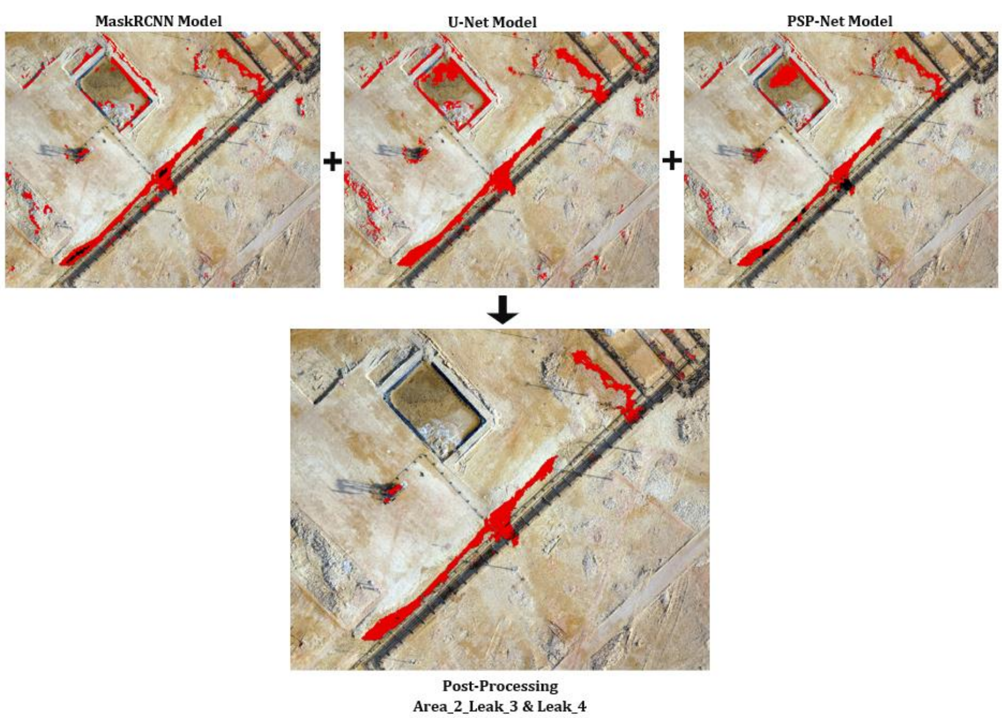


Figure 43. Post Processing (PSP-Net, U-Net and Mask R-CNN)

9. Discussion

After deploying the trained DL models (SSD, PSP-Net, U-Net and Mask R-CNN) and seen the results either for the Raw or the Georeferenced Ortho-mosaiced data, its clearly that overall detections are in a good accuracy and are able to identify Oil Leaks with high confidence rate.

9.1 Result Analysis

The Mean Average Precision at (IoU = 50%) of the SSD model has reached more than 85% in detecting the positive Oil Leaks from the total positive cases. While the Mean Average Precision and the Average Recall for the Large area (> 96^2) reached close to 70% which is also a good percentage in finding the Oil Leaks. The minimum Loss recorded was 2.1 at step 46,643 which consider as the best refining loss for the model.

PSP-Net Classifier and U-Net as Semantic Segmentation achieved a very good performance in detecting the Test Leak (Oil Leak which not been seen before by the model). Both models are also able to identify other Oil Leaks correctly except few parts of Leak 1 within Area 1. For the PSP-Net Classifier, the minimum Training Loss was around 0.09 at epoch number 5 and the minimum Validation Loss was around 0.10 at epoch number 2. Both Training and Validation Loss are somehow aligned and having the same trend except within the epoch number 2 where the Validation Loss goes a bit higher than the Training Loss. For the U-Net, both Training and Validation Loss are having the same trend which gives an indication of the model performance in detecting the Oil Leaks within the Training and the Validation dataset. The minimum Training Loss was around 0.06 at epoch number 5 and the minimum Validation Loss was around 0.047 at epoch number 3.

Mask R-CNN, by setting the Confidence Threshold to 90% it does a good detection on the Test Oil Leak similar to PSP-Net Classifier and U-Net and to all other Oil Leaks except Leak 3 in Area 3 which missed some part of it. Also, it missed small parts of Leak 3 in Area 2 but overall, the detection performance was very good in finding all the Oil Leaks.

9.2 Challenges and Recommendation

The most challenge been faced is the present of shadows within the training dataset which confused a bit the trained model and contributed to the detection of the Oil Leaks. PSP-Net Classifier and U-Net have much contribution of shadows than Mask R-CNN. So, detecting Oil Leaks from High Resolution Aerial Photos is very challenging as the color of the Leak is black which is very close to the Shadows which make it very difficult for the model to distinguish between what is consider as an Oil and what is a Shadow. Thus, a Post Processing was implemented to eliminate or minimize such effect.

The present of shadows is linked to data acquisitions and the time of acquiring the images. Thus, its recommended to acquire the images at mid-day when the sun rays be vertical at the equator to avoid such contribution of shadows.

10. Conclusion

Oil Leaks is one of the critical problems that most of the Oil and Gas industries are suffering from. Oil Leaks can be at offshore from vessels while transporting the Oil or even from Flowlines, Exploration and Production Platforms. And these Oil Leaks can cause an environmental damage if it lately discovered and left for days or even weeks without taken and immediate action. The traditional way of detecting these Oil Leaks is by driving along the corridor of the Flowlines and report back any observed leaks which consume a lot of time, cost, require more staff and has a potential of harm people and it is not good enough for early spill containment where the quick response and the action is required. Thus, the utilize of Unnamed Aerial System is been proposed to fly over these Flowlines in frequent basis. Due to the large area coverage, there will be a large number of Aerial Photos which will be acquired in a daily basis and will require someone to go through each individual photo and inspect which will take a lot of time and consume manpower, so this study addressed this challenge by proposing Deep Learning as a way of speeding up the process of detecting the Oil Leaks from these Aerial Photos immediately after the UAS Survey along the Flowlines is completed.

10.1 Summary

To summarize with, the Deep Learning models was built in two approaches considering the input data type. The first approach focused in Raw data to build Deep Learning model with SSD architecture. The model was built outside ArcGIS Pro and ArcGIS API for Python using TensorFlow as an open source Machine Learning Platform inside Jupyter Lab within CONDA Environment. The trained model showed a good performance with high Mean Average Precision (~85%). The second approach focused in Georeferenced Ortho-mosaiced images to build DL models with PSP-Net Classifier, U-Net and Mask R-CNN architectures. The models were built within ArcGIS Pro and ArcGIS API for Python using Jupyter Notebook. The trained models showed a very good performance in detecting the extent of the Oil Leaks with some contribution of shadows. Post Processing was implemented to minimize the shadow effect and eliminate any bad detection by applying some constrains.

10.2 Feature Work

To increase the accuracy and the performance of the trained models it is recommended to increase the number to training samples. More training samples will increase the detection confidence of the Oil Leaks. Another consideration is to use the Digital Surface Model which includes height information which can be used as another constrain to avoid built-up facilities while training the model.

References

1. Huang, B., B. Zhao, and Y. Song, *Urban land-use mapping using a deep convolutional neural network with high spatial resolution multispectral remote sensing imagery.* Remote Sensing of Environment, 2018. **214**: p. 73-86.
2. Deng, Z., et al., *Multi-scale object detection in remote sensing imagery with convolutional neural networks.* ISPRS Journal of Photogrammetry and Remote Sensing, 2018. **145**: p. 3-22.
3. Ding, P., et al., *A light and faster regional convolutional neural network for object detection in optical remote sensing images.* ISPRS Journal of Photogrammetry and Remote Sensing, 2018. **141**: p. 208-218.
4. Audebert, N., B. Le Saux, and S. Lefèvre, *Beyond RGB: Very high resolution urban remote sensing with multimodal deep networks.* ISPRS Journal of Photogrammetry and Remote Sensing, 2018. **140**: p. 20-32.
5. Binaghi, E., I. Gallo, and M. Pepe, *A neural adaptive model for feature extraction and recognition in high resolution remote sensing imagery.* International Journal of Remote Sensing, 2003. **24**(20): p. 3947-3959.
6. El Bouchefry, K. and R.S. de Souza, *Chapter 12 - Learning in Big Data: Introduction to Machine Learning*, in *Knowledge Discovery in Big Data from Astronomy and Earth Observation*, P. Škoda and F. Adam, Editors. 2020, Elsevier. p. 225-249.
7. Vieira, S., W.H. Lopez Pinaya, and A. Mechelli, *Chapter 1 - Introduction to machine learning*, in *Machine Learning*, A. Mechelli and S. Vieira, Editors. 2020, Academic Press. p. 1-20.
8. Bishop, C.M., *Pattern recognition and machine learning*. 2006, New York: Springer.
9. Mohanasundaram, R., et al., *Chapter 8 - Deep Learning and Semi-Supervised and Transfer Learning Algorithms for Medical Imaging*, in *Deep Learning and Parallel Computing Environment for Bioengineering Systems*, A.K. Sangaiah, Editor. 2019, Academic Press. p. 139-151.
10. Wang, Z. and T. Hong, *Reinforcement learning for building controls: The opportunities and challenges.* Applied Energy, 2020. **269**: p. 115036.
11. Langer, H., S. Falsaperla, and C. Hammer, *Chapter 2 - Supervised learning*, in *Advantages and Pitfalls of Pattern Recognition*, H. Langer, S. Falsaperla, and C. Hammer, Editors. 2020, Elsevier. p. 33-85.
12. Langer, H., S. Falsaperla, and C. Hammer, *Chapter 3 - Unsupervised learning*, in *Advantages and Pitfalls of Pattern Recognition*, H. Langer, S. Falsaperla, and C. Hammer, Editors. 2020, Elsevier. p. 87-124.
13. Li, L., et al., *Convolution Neural Network based Chemical Leakage Identification*, in *Computer Aided Chemical Engineering*, M.R. Eden, M.G. Ierapetritou, and G.P. Towler, Editors. 2018, Elsevier. p. 2329-2334.
14. Zhao, G., et al., *Multiple convolutional layers fusion framework for hyperspectral image classification.* Neurocomputing, 2019. **339**: p. 149-160.
15. Teuwen, J. and N. Moriakov, *Chapter 20 - Convolutional neural networks*, in *Handbook of Medical Image Computing and Computer Assisted Intervention*, S.K. Zhou, D. Rueckert, and G. Fichtinger, Editors. 2020, Academic Press. p. 481-501.
16. Zhao, Z., et al., *Object Detection With Deep Learning: A Review.* IEEE Transactions on Neural Networks and Learning Systems, 2019. **30**(11): p. 3212-3232.
17. Wu, X., D. Sahoo, and S.C.H. Hoi, *Recent advances in deep learning for object detection.* Neurocomputing, 2020.
18. Li, Y., et al., *Multi-block SSD based on small object detection for UAV railway scene surveillance.* Chinese Journal of Aeronautics, 2020.
19. Liu, W., et al. *SSD: Single Shot MultiBox Detector*. arXiv e-prints, 2015. arXiv:1512.02325.

20. Ronneberger, O., P. Fischer, and T. Brox *U-Net: Convolutional Networks for Biomedical Image Segmentation*. arXiv e-prints, 2015. arXiv:1505.04597.
21. Zhao, H., et al. *Pyramid Scene Parsing Network*. arXiv e-prints, 2016. arXiv:1612.01105.
22. He, K., et al. *Mask R-CNN*. arXiv e-prints, 2017. arXiv:1703.06870.
23. Lin, T.-Y., et al. *Microsoft COCO: Common Objects in Context*. arXiv e-prints, 2014. arXiv:1405.0312.
24. Agaradahalli Gurumurthy, V. *Encoder-Decoder based CNN and Fully Connected CRFs for Remote Sensed Image Segmentation*. arXiv e-prints, 2019. arXiv:1910.06041.
25. Badrinarayanan, V., A. Kendall, and R. Cipolla *SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation*. arXiv e-prints, 2015. arXiv:1511.00561.
26. Tian, Y., et al., *Instance segmentation of apple flowers using the improved mask R–CNN model.* Biosystems Engineering, 2020. **193**: p. 264-278.
27. Everingham, M., et al., *The Pascal Visual Object Classes (VOC) Challenge.* International Journal of Computer Vision, 2010. **88**(2): p. 303-338.

Appendix

1- Flowlines Oil Leaks Detection from Unmanned Aerial Photos (Drones) Using Deep Learning Mask R-CNN. All codes and step by step demonstration (Jupyter Notebook) using ArcGIS Pro 2.5 and ArcGIS API for Python can be accessible through GitHub (Here)
2- Flowlines Oil Leaks Detection from Unmanned Aerial Photos (Drones) Using Deep Learning PSP-Net Classifier. All codes and step by step demonstration (Jupyter Notebook) using ArcGIS Pro 2.5 and ArcGIS API for Python can be accessible through GitHub (Here)
3- Flowlines Oil Leaks Detection from Unmanned Aerial Photos (Drones) Using Deep Learning U-Net. All codes and step by step demonstration (Jupyter Notebook) using ArcGIS Pro 2.5 and ArcGIS API for Python can be accessible through GitHub (Here)