Automatic Detection of Storage Oil Tanks using Deep Learning Models with Satellite Data

By

Vardhan Raj Modi

2023

Abstract

Fossil fuels are the bedrock of the industrialisation economy, and the oil storage information such as their geographical location is strategically and economically beneficial for the industry. Potential applications derived from object detection are imperative for carbon regulators. With the development of cutting-edge deep learning models and the rapid availability of pre-trained models, it is computationally and timely feasible to develop an end-to-end object detection methodology pipeline with the availability of sufficient training data. Two sets of training data have been utilised to perform training on two different object detection algorithms with data augmentation and transfer learning techniques. Cascade RCNN is a region-based object detection algorithm that performs detections multiple times to get optimal results, whereas YOLO performs detection and classification only once on the whole image. 88.7 mAP (Mean Average Precision) has been achieved with the CNN model, and 33.7 mAP for the YOLOX model. Planet Labs' proprietary data, which has been pre-processed, is used for the inference of the detection models to test the accuracy. >95% accurate predictions of oil storage tanks have been achieved in the inference stage of the YOLOX model with Planet data.

Acknowledgments

This study would not have been possible without the help of the Computer Vision (CV) Open-Source community. The ML tools developed by OpenMMLab© and Megvii© had been a significant part of this study. Further, thanks to the numerous contributors to the online tutorials, blogs such as StackOverflow©, and forums. And special thanks to **Prof. Kevin Tansey** for his immense support throughout my MSc course and industrial placement.

Declaration

I hereby declare that the work presented in this study is my own and all the necessary literature and adapted methods have been properly referenced.

Software Engineering

This project is set up in GitHub (github.com/vrym2) with the usage of CI/CD methods, continuous integration, and continuous development have been achieved. The codebase is constantly updated with new repositories and modules, and each codebase repository related to this study can be found at github.com/vrym2/gy7702. The primary programming language used for the project development is Python from version 3.6 to 3.9. Python3.10 and Python3.11 have been avoided in use, as the project came across some bugs and due to compatibility issues faced due to the software and applications used in the project. Microsoft's Visual Studio Code is used as the integrated development environment (IDE). As a project that involves high data processing, machine learning, and deep learning models, there is a need for a system with high performance, agility, memory, and a GPU. Google is offering credit worth three hundred dollars for every Google account with a billing status. So, the required project environments are set up in Google's Virtual machines with Ubuntu as their operating system. Multiple VMs have been set as per the project requirements, and necessary software and applications have been installed. To run DCNN model training and inference, a system with Nvidia T4 series GPU has been set and CUDA drivers' versions 10.1 and 10.2 were installed in those systems. A summary of the system specifications can be found below.

Table 1 System specifications

Python version	Python 3.6, 3.7, 3.8, 3.9
Integrated Development Environment (IDE)	Visual Studio Code 22.04
Operating System (OS)	Ubuntu 18.04, 22.04
Graphical Processing Unit (GPU)	Nvidia T4
Drivers' toolkit	CUDA 10.1, 10.2

Table of Contents

List of Figures

List of Abbreviations

Abbreviation	Description
CHT	Circular Hough Transform
SGS	Shape Guide Saliency
CSC	Contour Shape Cue
Bbox	Bounding Box
SAR	Synthetic-Aperture Radar
CV	Computer-Vision
DNN	Deep-Neural Network
HR	High-resolution
LR	Low-resolution
ROI	Region of Interest
YOLO	You-Only-Look-Once
DenseNet	Densely Connected Network
COCO	Common Objects in Context dataset
IDE	Integrated Development Environment
GPU	Graphical Processing Unit
VGG	Visual Geometry Group
R-CNN	Region-based Convolutional neural network.
ResNet	Residual Network
mAP	Mean Average Precision
AR	Average Recall

Chapter-1 Introduction

Crude oil is a significant driver of the world's economy since the late 18th century and as well as a major contributor to the world's greenhouse gas emissions. Oil tank storage inventory and their analytics are crucial in the fossil fuel market and accurate updates on these analytics on a constant timely basis is imperative and an essential factor for the daily traders and world's governments. Due to its nature of high sensitivity and low transparency, fossil fuel production houses and governments are not transparent in providing information. Counting the number of tanks in a desired oil storage facility gives the clients a strategic advantage. Hence, accurate updates on oil reserve locations are of huge importance for both carbon watchers and regulators. Remotely sensed and satellite images can provide frequent information by utilising object detection algorithms. Potential applications derived from object detection tasks have huge commercial and research value. This can be complimentary to the studies and applications of vessel tracking and monitoring, and oil spill detection both offshore and onshore. With the inception of the Copernicus program, ESA has brought forth publicly available high-resolution data of the optical bands which have been crucial in object detection tasks. Ciocarlan, A. and Stoian, A. (2021) performed ship detection on multi-spectral Sentinel-2 data by developing a deep learning network architecture which utilises self-supervised learning. Annotated data is scarcely available for remote sensing object detection tasks. They used transfer learning techniques by pretraining a neural network that can extract feature patterns and learning invariances. They argue that features learned through their development of an unsupervised pipeline of self-supervised learning work well with a lack of available annotated data.

Classification is an important computer vision task when dealing with satellite images. There are two types of classification techniques in computer vision tasks: Image-level classification and pixel-level classification. Image-level classification assigns individual labels for each image, whereas pixel-level classification extract features in each image and assigns labels for each pixel. The latter is beneficial if we are dealing with multi-class, multi-labeled satellite images. There are typically three types of commercial storage tanks that are observed in satellite images. fixed roof (Yellow, Figure-1), and floating head storage tanks (Red, Figure-1). The difference between a floating head and to fixed roof is that as the name suggests, the roof of the storage tanks constantly changes based on the volume occupancy of the crude oil inside. A significant advantage of the floating head storage tanks is their ability of vapor loss reduction, meaning it reduces any flammable vapor that sits on top of the liquid inside the

tanks, which in turn reduces safety risks and environmental air pollution. This study attempts to perform pixel-level classification and assign labels for the different types of detected tanks in satellite images.

Object detection through Computer vision has come a long way and making tremendous strides in progress through the easy access and availability of a wide range of deep neural network learning algorithms which are easily reproducible and properly adaptable. Since 2016, after YOLO, computer vision researchers and remote sensing scientists had taken a very keen interest to make a precise detection of an oil storage tank and there is a vast amount of research with publicly available datasets and algorithms in the research and open-source community. Kushwaha, N. K. *et al*. (2013) proposed one of the first methods that can detect a bright circular object with the help of its morphological characters and segmentation. The Circular Hough transform algorithm (CHT) has been widely used in the previous literature to extract circular features in images. A satellite image can comprise many objects with circular features such as roundabouts, crop circles, etc. Moreover, to contain an oil storage tank and its features in a bounding box, a model shall consider the shadow that has been left by the incident angle of the sun. This inclusion helps further in estimating the volume of the tank. Cui, Z. *et al*. (2020) proposed, in their paper, a CNN called Ellipse-FCN to accurately localise the outline of the oil tank that includes both the top head and bottom ground contact. There is a methodology based on contour shape and saliency characteristics extracted by using the shape guide saliency (SGS) model and contour shape cue (CSC) by Jing, M. *et al*. (2018). Cui, Z. *et al*. (2020) compared their model with Jing, M. *et al*. (2018), and their model yielded better accuracy, precision, and IOU scores than traditional shape-guided methods. This is evident that utilising CNNs and DNNs is advantageous in object detection tasks.

Most of the literature used high-quality data from satellites such as Gafoen-3, Airbus SPOT imagery, etc. The availability of high-quality data is irrelevant in performing object detection of storage oil tanks according to Tadros, A. *et al.* (2020). Most of the publicly available data either through ESA's Copernicus program or NASA's LANDSAT, provides medium to low resolution relative to this study. But Tadros, A. *et al.* (2020), along with their subsequent paper Tadros, A. *et al.* (2021) have developed a methodology that detects objects-based clustering and patch matching on low-res Sentinel-2 images. This is highly efficient for a group of oil tanks, but storage facilities comprise both tank clusters and distributed tanks. They did not account for the classification of the oil tank detection into floating and fixed heads due to the low-resolution nature. Moreover, patch matching has significant disadvantages, as there is an overfitting problem of similar features for distinct objects through low-res patches. Transfer learning

techniques could fill the gap when detecting storage tanks in medium to low-res satellite data. Abba, A. *et al.,* (2020) performed transfer learning on a pre-trained ResNet-50 CNN. They utilised a pre-trained resnet backbone network which was trained on a benchmark ImageNet dataset and fine-turned the model with the benchmark AID (Aerial Image Dataset). AID does not have photographed images, unlike ImageNet which has millions of natural images. Tran Pires de Lima, R. and Marfurt, K. (2019) also utilised transfer learning on VGG19Net and Inception V3 models and evaluated how the specialisation of CNN affected the transfer learning process by splitting the original models into different nodes. Hyperparameters have a significant influence on the performance of an object detection model, and they discovered transfer learning on a more generic dataset outperforms a model trained on a smaller remotely sensed dataset. It is because an object detection backbone such as ResNet, a pre-trained model that can adapt the knowledge of extracting generic features that can be transferable across different image domains such as satellite images. In these pre-trained models, initial layers work as feature extractors, extracting low-level features such as edges, textures, contours, and basic shapes, hence the pre-trained models can be utilised for image classification in remote sensing images.

Optical images are superseded with some challenges such as cloud cover, where it would be difficult to detect an oil tank in the presence of a cloud cover. R. Zhang *et al*. (2022) has resolved this issue by enhancing the detection process in high-res SAR image with optical image. As mentioned earlier, cloud cover can have negative impacts on observing an object, so they proposed a comparative “multistage framework for oil tank detection in SAR images using optical image enhancement” (R. Zhang et al. 2022). As can be observed, this is a form of transfer learning between different modalities rather than spatial resolutions. This framework is pre-trained to extract features as the initialisation of the training. As they dubbed it a “Teacher-Student” network where the network designed for optical data will guide the SAR network. In their work, each detected oil tank will fall into two distinct categories (Tank and Floating head) and one mutual category (Tank cluster) (Figure-1). As common in every object detection deep learning framework, a bounding box (Bbox) forms around the detected object. In the final inference stage, the model is used to detect oil tanks only in SAR images.

With the availability of more advanced and bleeding edge deep learning frameworks, along with publicly available datasets, it has been feasible to build a detection pipeline from the ground up, with the help of pre-trained models, train and test on a specific dataset for object detection, and fine-tuning of the high-resolution model on a dataset of low-resolution allows the model in an adaption of the characteristics and features specific to the low-res images.

Airbus has provided its SPOT image series on the Kaggle website for research to the open-source community (Airbus-Kaggle). Along with that, there is a collection of 1000 high-res Google Earth images of multiple oil storage tank locations worldwide with annotations. This study focused on adapting the studies mentioned above, open-access models, and datasets and performing experiments that evaluate the performance metrics and feasibility studies on different CV techniques. So, the research questions tried to be answered in this study are as follows:

1. How do correlating characteristics of two different spatial resolutions enhance feature learning in an optical satellite image?
2. In the process, how to develop an end-to-end machine learning data pipeline that can be scaled?
3. Is there a way to use transfer learning with the help of fine-tuned high-resolution trained model and be able to adapt to medium to low-resolution datasets?

Figure.1 - Image patch of Airbus SPOT image, Red-Floating roof, Yellow-Fixed roof, Blue-Tank cluster: Resolution-1.5m (Source-Airbus Oil Storage Detection, Kaggle.com)

This report follows a conventional academic paper structure, with the current introduction section for introducing the study and its elements, followed by background and previous work where the previous literature and methodologies tangible to the report are explained. It is followed by the materials and methods section which details the datasets and adapted

methodologies for the study. The results and discussion section is followed by a conclusion of the report.

Chapter-2 Background and Previous Work

A satellite image with a very coarse spatial resolution of ~1.5m/pixel contains several detailed geometrically identical objects, and these objects are typically in complex backgrounds with similar colour or share patterns. Some of these features in a satellite image make object detection a challenging task. There have been several variations of methodologies for oil tank detection in the past decade. Most of them fall under three categories. 1) traditional methodologies which detect circular objects in an image, 2) saliency-based methods, and 3) object detection from machine learning and DNN models.

2.1 - Traditional Methods vs Deep Neural Networks

Naveen, K., *et al*. (2013) only focused on very well and brightly lit objects that can be easily separated from the background, and in a low contrast area. In their adopted approach, the authors emphasised circular shape oil tank targets by image enhancement on a panchromatic image. Segmentation was performed on an optical image which has been enhanced using the "split and merge" technique. Introducing a knowledge base strategy which is like the annotating part of test batch preparation in the DNN model training gave an advantage in their methodology. In the final stage, a supervised classifier is used to perform object detection and eliminate false positives. They employed small patches of images for shadow feature extraction which is not suitable for large-scale applications. Ok, A. O. (2014) focused on storage tanks with simple and desirable shape outlines based on the shadows caused by the illumination angles which are easy to detect. Considering oil tanks with a wide range of dimensions and structures, their method would fail to detect the targets if the structure is not visible properly. Traditional methods often employ manual work of false target elimination due to the inaccuracies and low precision rate which is time costly. Zerman, E. *et al.,* (2014) proposed, in their paper, to use a "gradient-based Fast Radial Symmetry Transform" algorithm which they modified to find the circular objects and eliminated false positives in the post-processing stage.

Traditional methods have significant disadvantages in terms of accurate predictions and manual cost of time. DNNs can overcome those disadvantages through a progressive and iterative learning process. Moein, Z. *et al*. (2020) used region of interest (ROI) extraction as the starting point of the model pipeline. Faster R-CNN is highly efficient with low-res data and able to group oil depot reserves with a bounding box, and the authors employed a fast circle extraction method to select some suitable ROIs for object localisation. Convolutions have the

inherent advantage of learning from hidden layers. That's what's lacking in the traditional and salient. Yu, B. *et al.,* (2021) proposed an end-to-end deep convolution network called Res2-Unet+ in which they demonstrated detecting oil tanks in large-scale images. Their model was lagging in the detection of the small oil tanks with undesirable dimensions. Xu, D., and Wu, Y. (2020) proposed a methodology combination of YOLO-V3 and DenseNet. The main advantage of their approach is that the model combination can detect oil depots that are either densely or uniformly distributed and be able to detect even in bad weather conditions.

2.2 - You Only Look Once

YOLO has been making tremendous strides since its inception in 2016. It approaches the task of detection as a regression problem which is based on the DarkNet architecture, and YOLO can predict bounding boxes and class probabilities within a single network. The concept behind, if it can be put simply is that a user can define the grid size cell in which if the targeted object fell inside the box, defines the confidence rate of the detected object. The researchers have developed YOLO with 53 convolutional layers in combination with skip connections, and they named it DarkNet-53. DarkNet-53 or YOLOv3 performs detection in 3 different scales using a feature pyramid network. Bakirman, T. (2023) implemented several versions of YOLO

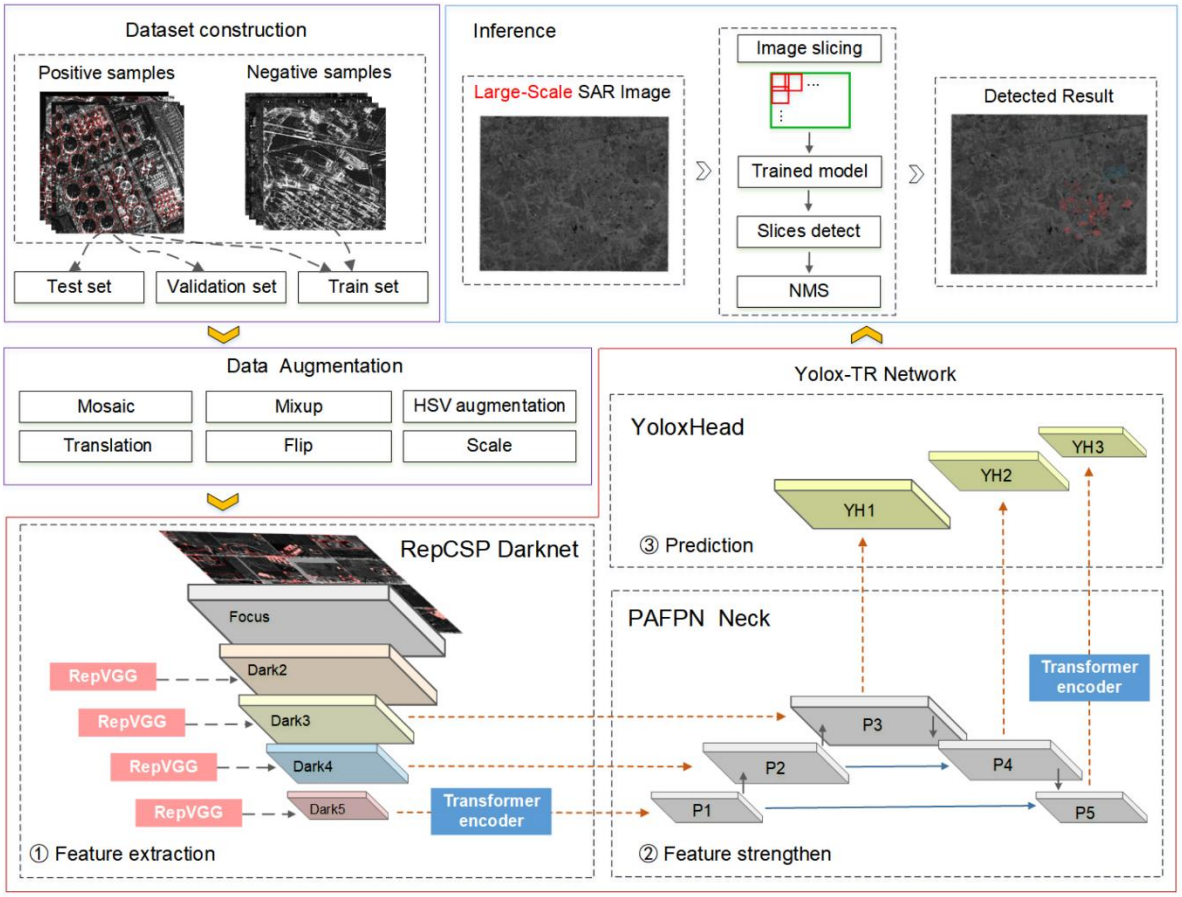

Figure.2- YOLOX-TR architecture (Wu, Q. et al., 2022)

architectures on the oil tank detection Airbus SPOT dataset to evaluate the performance metrics of individual architectures. With each iteration of YOLO, researchers have been optimising in many ways. One of the main differences between traditional object detection models such as RCNN, Fast-RCNN, and Faster-RCNN to the YOLO is the number of times the input image is passed through the networks. Traditional CNNs are two stages whereas YOLO is one stage detector, meaning the detection or prediction of an object and its location is done in a single pass, but there are drawbacks to this method that they cannot detect small objects, hence they are more useful in object detection real-time in a resource-constrained environment. Among the architectures, Bakirman, T. (2023) experimented on, YOLOv5 and YOLOv7 seem to avoid this problem as visualised in their paper.

Each YOLO architecture is built with three parts (Backbone, Neck, and Head). The backbone is responsible for the extraction of features from an input image and is called a feature extractor. YOLO backbone is a CNN where it pools all the pixels of an input image to form features at multiple granularities. Typically, a backbone of YOLO is already pre-trained on a classification dataset such as ImageNet. Most of the backbone models are classification models. These features collected in the backbone are ingested into the neck where multiscale feature aggregation happens. Head in YOLO is where the object localisation with a bounding box and class prediction happens.

Wu, Q. *et al.,* (2022) performed oil tank object detection with their modified YOLOX model they named it YOLOX-TR (Figure 2). YOLOX is different from its predecessors as the researchers switched YOLOX to an anchor-free model. Anchors are essential elements that are pre-set bounding boxes in early object detection models. A detector is trained to perform classification of whether the anchor boxes overlap with the ground truth. As the location of a target and its scale is unknown, for a given image, multiple anchor boxes with varying sizes and aspect ratios are created. As anchors are pre-set, they have many hyper-parameters and computational requirements. Anchor-free methods localise objects with the help of **centers** or **key points** (Figure 3). These key points are pre-defined or self-learning in nature, the spatial

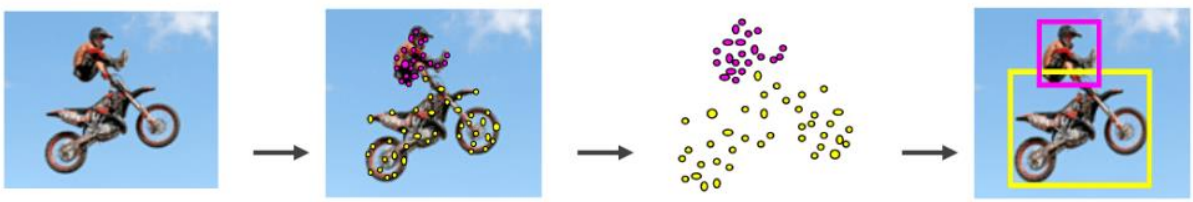

Figure.3- Key-points anchor free detection (learnopencv.com., 2022*)*

extent of an object is derived through these clusters of points. And the center-based approach finds the center of the object and predicts distances to the boundary of the localised object.

Deep neural network methodologies, in this study, pre-defined sets of images with known locations of oil storage tanks in the training data, technically, they are called “annotations”. Wu, Q. *et al.,* (2022) constructed their data with Gafoen-3 High-res SAR images, having a combination of both positive and negative samples, and split them into training, testing, and validation sets. After that, data augmentation (Figure 4) is an optional step in any DNN methodology pipeline. Data augmentation is highly beneficial if there was a limited training dataset. Data augmentation could be as simple as rotating a given image 45 deg. on its axis to produce three other distinct images. Some of the DNN models incorporate data augmentation methods by default.

Figure.4- Satellite data augmentation (Athanosis, P., 2022)

Wu, Q. *et al.,* (2022) modified the original YOLOX model to improve the feature extraction in a very dense oil tank area by including a transformer encoder in the backbone and the neck of the architecture. This will enhance the model for a better feature map representation and increases its capability to find the ROI of oil tanks. They had replaced the original convolutional layers in YOLOX with RepVGG blocks (re-parametrised) which decouples the training and inference time architecture. RepVGG is a feed-forward single topological network without any branches. They called their new backbone architecture a “RepCSP” network. After constructing a model, a DNN pipeline always has an inference module, where the testing data is fed to visualise the model performance. Slicing large-scale images is an ideal method to feed the model as most of the models outperform with small image sizes.

2.3 - Transfer Learning

There are several limitations for DNNs, and one such limitation in the context of the current study is the availability of annotated datasets and high-resolution images. Typically, a commercial satellite tends to have a very coarse resolution that can help DNNs to learn the individual features, but the publicly available data with low resolution relative to the intended target (storage oil tanks) becomes obsolete when it comes to tank detection. As mentioned earlier, oil tanks have been successfully detected in the sentinel-2 images using the patch-matching technique (Tadros, A. *et al.,* 2020), but the drawback is that the model can interpret anything that resembles the patch without any additional context. To overcome the limitation of high-resolution images availability, transfer learning techniques have been employed by the researchers where the methodology starts with a pre-trained model on a specific high-res dataset and fine-tune for object detection on a different low-res dataset.

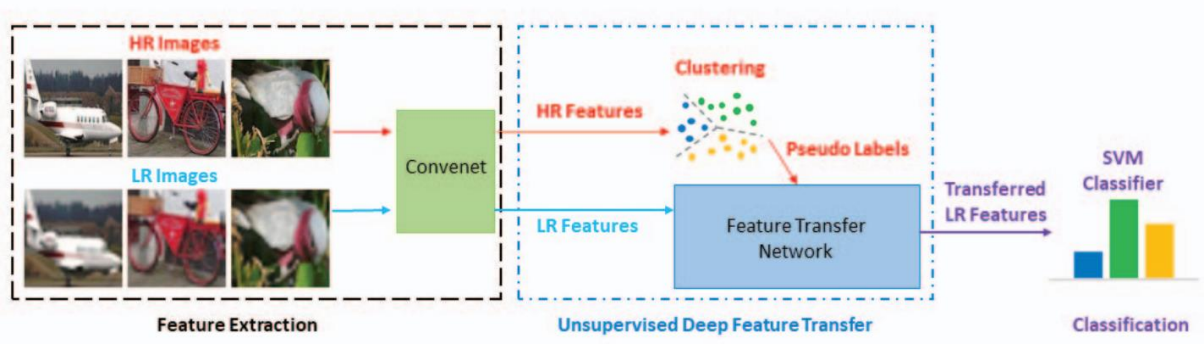

Figure.5- Transfer learning methodolugy used in Wu, Y. *et al*. (n.d.)

Wu, Y *et al*. (n.d.) utilised some of the transfer learning techniques to classify low-resolution images in their paper. Their methodology involves three modules, in which the first module is comprised of feature extractors from both HR and LR images (Figure 5). As shown in the figure, they have used the same dataset for both resolution feature extraction. With down-sampling the original high-res image and up-sampling gives the desired set of data that could be used in the feature extractor. In this stage, they used 'Convenet' as their backbone with pre-trained`Resnet-101' on the PASCAL VOC2007 dataset (Everingham, M. *et al.* 2014) and extracted N-dimensional features in both. The HR features are clustered using the classical k-mean algorithm and assigned pseudo labels, which are utilised to assign labels for LR features that sit at the nearest k-centroid. And finally using SVM, a classifier to classify the results. In testing, they had run the pipeline on LR images, where the features are extracted, and they are fed into the feature transfer network to derive transferred LR features and perform classification.

Zou, M. and Zhong, Y. (2018) used transfer learning techniques in their paper to classify objects in optical satellite images. For any DCNN training, a great deal of annotated and labeled data is required, one of the datasets used in this study for model training, such as Airbus SPOT imagery provided by the official Airbus account on Kaggle, has annotations and labels included in the dataset. Even with the availability, the model will run into the problem of overfitting. Hence, in the transfer learning methodology by Zou, M. and Zhong, Y. (2018), the model was pre-trained on the ImageNet dataset, which is an industry standard, where the weights extracted, and optical satellite pictures are utilised in fine-tuning the weights/parameters. A DCNN has many advantages in its usage of image processing. Supervised learning in DCNNs does not require interference in its training stage, and DCNNs share weights among and within layers. With the pre-trained model like AlexNet, weights of all layers are transferred to initialise the network. Fine-tuning is a machine learning process where the parameters of a model are re-adjusted accurately and precisely for better performance (Maggiori, E. *et.al.,* 2017). After that, they fine-tuned the weights of the model with a backpropagation algorithm.

This study encapsulates all the necessary research and implementation of data pre-processing, end-to-end deep learning architecture building, training the machine learning and deep learning models, inference, and experimentation. Data processing was the initial and crucial part of any object detection methodology pipeline. The datasets and the data processing steps have been explained in the later section, with the adapted deep learning methodologies.

Chapter-3 Materials and Methods

Dataset construction or preparation is a crucial aspect of computer vision and object detection methodologies. As mentioned earlier, annotated data is crucial for object detection model training. In remote sensing images, data regarding the presence of an object, and the location of the object in terms of pixel coordinates, typically bounded with a box and a count are annotated. Annotations can be defined as metadata for the image, and the machine learning models can map the metadata to the objects annotated and the models can detect objects in an unlabelled dataset in inference. Oil terminal locations around the United Kingdom have been selected as the study area. Geographic coordinate locations have been manually collected and these locations have been mapped for visualisation (Figure 6). This report has considered optical datasets of oil storage tanks that have been collected from the Kaggle website for training, validation, and testing purposes in its study. Planet scenes collected from the locations on the below map have been utilised for the inference part of the methodology.

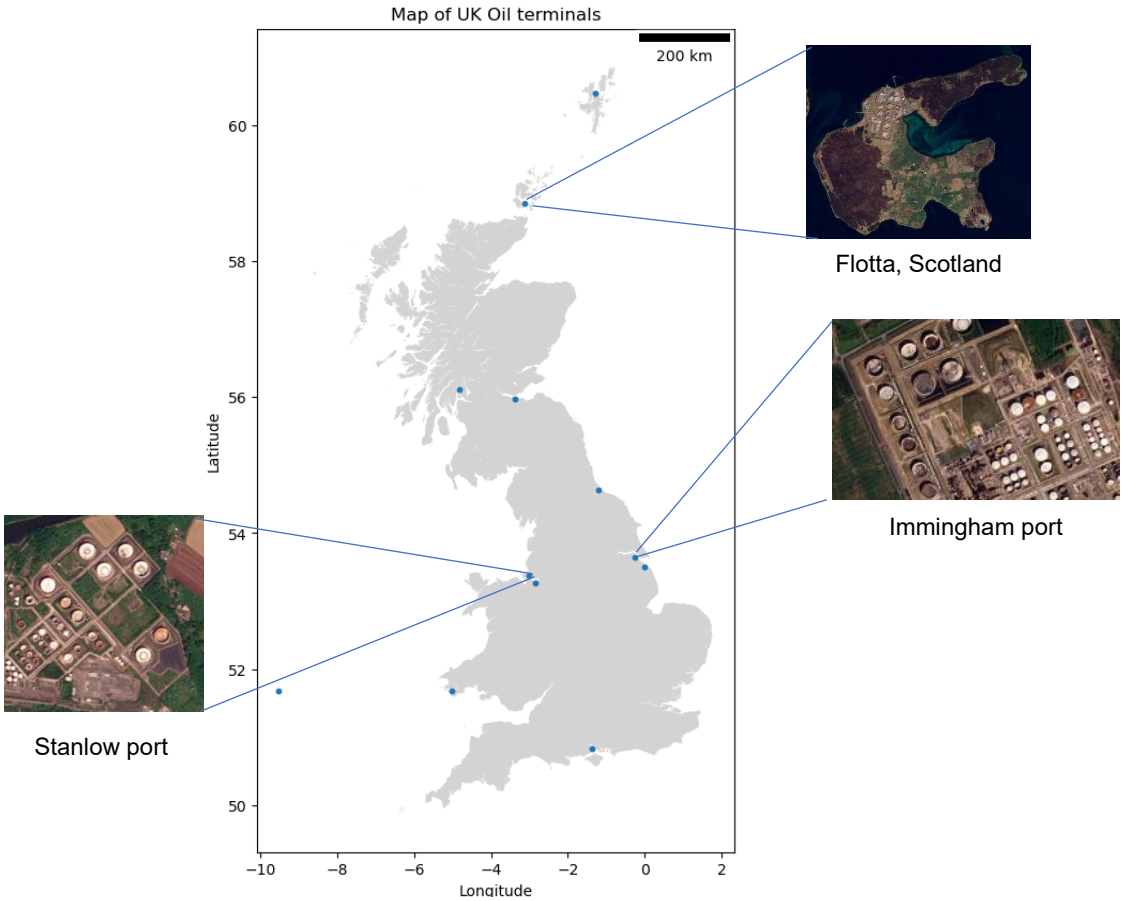

Figure.6- Geographic locations of UK oil terminals

Optical data comes in different resolutions and scales, three different optical datasets have been considered from three different sources in this report, in which two of them (Google Earth and Airbus) are available publicly on the Kaggle website. These two datasets are annotated

with metadata regarding the location of oil storage tanks and classification between different oil tank types and the resolution (Google Earth and Airbus SPOT = ~<1m^2 and 1.3m^2/pixel respectively) of these two datasets are relatively high compared to the third optical dataset. The third dataset has been produced with Planet API, downloading the data from their "PS Scene" catalog, which are from the Planet Scope sensors. The resolution (3m^2/pixel) of these images are relatively low, but the idea was to use the transfer learning techniques to train the model of both HR and LR images and inference the model with LR unlabeled images.

3.1 - Datasets

3.1.1 - Google Earth Oil tanks dataset - Kaggle

A Kaggle dataset (Karl Heyer, 2019), comprises several high-resolution industrial location images taken on Google Earth from all around the world. These individual images are annotated with bounding box information of floating head tanks and saved in a JSON file. The dataset comprises files in four different categories, which are as follows (Karl Heyer, 2019).

1. Large images (Figure 7): A folder that contains 100 large images with a pixel width and height of 4800 x 4800 pixels. These files are in the '.jpg` format with the naming convention as 'id_large.jpg'. 'id' is the number of the image (1-100)
2. Image patches (Figure 7): A folder that contains image patches of the large images that were split into chunks with the size of 512*512 and the overlap between each image patch is 37 pixels on both axes. The naming convention for image patches is 'id_row_column.jpg'. The row and the column of the image patch represent the row and column of the chuck of the large image.
3. labels.json: This JSON file consists of the labeled information of each image in a typical dictionary format. Each image is labeled "Skip" If there are no tanks present in the image, and the images with the tanks are annotated with an appropriate label, there are three labels in this dataset that corresponds to the tanks, "tanks", "floating head tank", "tank cluster."
4. labels_coco.json: This JSON file has the same information as the above JSON file, but it is converted into the COCO standard dataset format. The bounding box information is represented as [x_min, y_min, width, height].

In the process of selecting data for training, a 'seed' value has been set so that the data split is identical whenever the detector data pipeline is run. As there are images with and without oil tanks in them, they are divided into two sets, data with annotations, and data without annotations. Among these two datasets, 350 random samples are selected, 300 for training

and 50 for validating, respectively in each dataset (with annotations, without annotations). These datasets are now converted into COCO format. The COCO dataset is the benchmark in the object detection community which was developed by Microsoft. The annotations used to store image data set by COCO have been followed for the construction of this dataset. If defined, COCO format is a JSON structure that dictates the format of saving image labels and metadata. Both the training and validation datasets have been converted into COCO format in this study.

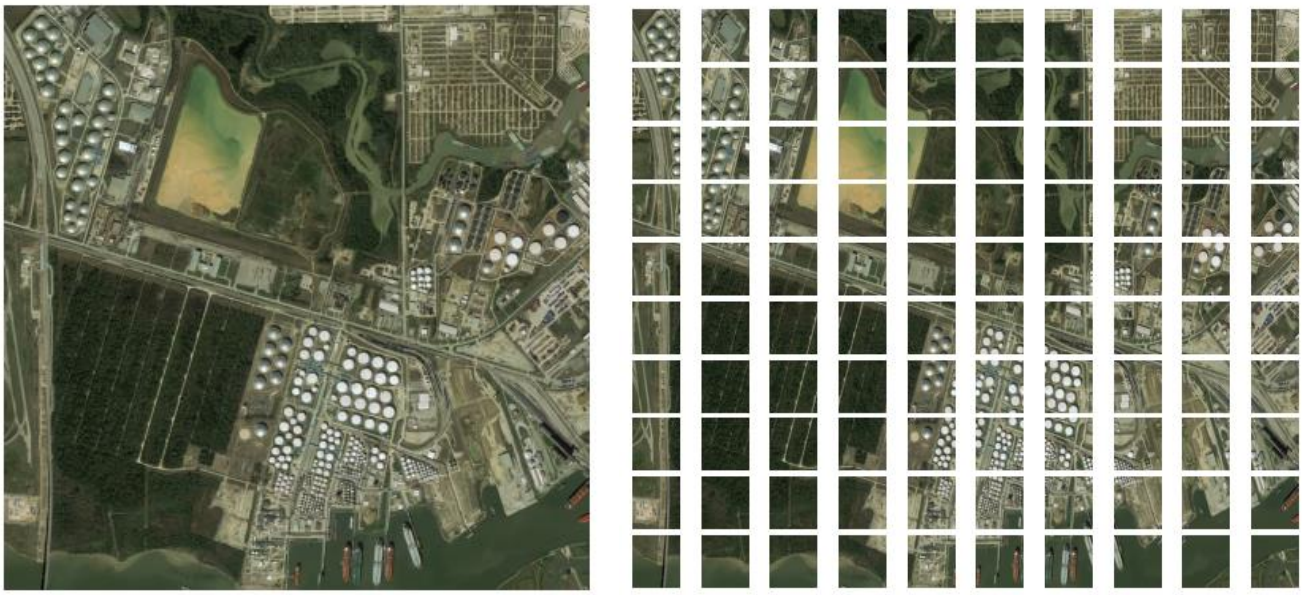

Figure.7- Image patch construction (Google Earth Images)

3.1.2 - Airbus SPOT imagery - Kaggle

Airbus (Airbusgeo, 2021) SPOT images were provided on Airbus's official Kaggle page. It comprises 98 high-resolution (1.3m^2/pixel) data in JPEG format. Each image has the spatial dimensions of 2560 x 2560 pixels, which is equivalent to a three-square kilometer of ground surface area. These images cover oil random oil storage locations across different parts of the world. They have also provided annotations of storage oil tanks in each high-res image. These annotations are stored in a CSV file which comprises the image ID, only one class of the object in the image which is a 'storage oil tank', and the pixel coordinates as bounding boxes where those objects are situated on the image. Annotated bound values consist of pixel coordinate (xmin, ymin, xmax, ymax) values of each bounding box. Calculating the difference between each axes gives the dimensional values (width, height, aspect ratio) of each bounding box, and basic statistical analysis has shown that there are bounding boxes with less than 5 pixels of height and width, and an aspect ratio greater than 2.5. Data has been cleansed by removing such sort of anomalies to prepare for model training.

3.1.3 - Planet

Planet Labs provides its data through its flagship API services which have a Pythonic configuration for data download and processing. Planet Scope products have two versions that can be downloaded, Basic and Ortho scenes. The difference between those scenes, as the name suggests, ortho scenes are orthorectified. Planet assets have three different ranges of band orders. A basic asset will be having three bands in the visible spectrum. This study concerns itself with the image files for computer vision detection, so only the basic assets have been considered for this study. All the scenes that correspond to the UK oil terminal locations have been collected for the inference of the adapted models.

3.2 - Methodology

This study utilised two different object detection algorithms to conduct experiments on the different datasets mentioned in the datasets section. The algorithms are Cascade R-CNN (Region-based Convolutional Neural Network) and YOLO. Architecturally, these two algorithms are different from each other as the prior has typically two stages, where it generates region proposals for the following stages to perform classification and bound box regressions. Latter is famous for its one-stage detector where the model divides the input into a series of grids and attempts to predict bounding boxes and class probabilities on those grid cells. The Cascade R-CNN algorithm which uses weights from pre-trained ResNet-101 architecture (explained in the later section), is trained on the Google Earth High-resolution image. Data augmentation techniques have been employed during the training process. With YOLO, YOLOX model has been utilised for training and inference of Airbus SPOT imagery data. Both models have been used to perform testing on opposite datasets that they have not been trained on, and YOLO is used for the inference of Planet Labs data. Compared to the YOLO, Cascade R-CNN yields better accuracy as the experiment shows on both datasets, but the inference of yolo model is computationally highly efficient and less prone to system hangups with low GPU memory. YOLO works well with large-scale images with complex backgrounds, and Cascade R-CNN achieves better performance and accuracy by progressively refining the detection results.

Model configuration is one of the foundational steps for computer vision methodologies for model training, testing, and inference. MMdetection, which is one of the popular toolboxes for object detection tasks, has been used to construct the oil tank detection methodology pipeline based on the Cascade R-CNN model configuration. Cascade R-CNN is a variant of the Faster R-CNN object detection framework. This configuration helps to detect objects on a different level of complexity, such as detecting small and inoculate objects. It uses a cascade of multiple

stages for detection, where each stage filters out false positives by focusing on different proposal subsets. This cascaded network helps in refining predictions. Typically, it will have three stages in the network. With each stage, progressively, IOU thresholds will be increased for the positive samples which focus on the proposals that overlap with the ground truth anchor boxes. Proposals are derived from the first stage by region proposal network (RPN), where these proposals are refined by the region of interest (ROI) head by extracting features and performing object classification and bbox regression.

3.2.1 - YOLOX

YOLO is one stage detector algorithm that has a similar skeletal architecture to Cascade RCNN. For this study, the YOLOX configuration has not been manipulated as it can extract low-res features of images. In the model architecture, CSPDarknet is used as the backbone which has multiple sequential blocks. It has five stages called dark2, dark3, dark4, dark5, and stem. Each stage samples the input data from the previous stage and performs feature extraction using convolution layers with distinct configurations. Batch normalization (BN) and an activate function called Sigmoid-weighted Linear Unit (SiLU) have been used. The activation function is a variance of the sigmoid function where it is computed by the multiplication of input with the sigmoid function. Dark5 stage includes SPP bottleneck, which stands for spatial pyramid pooling. This is used to remove the constraint of fixed size on the network. SPP layers generate fixed-length representations of the input and are fed into the Fully connected layers (FCC) later in the architecture.

3.2.2 - Cascade R-CNN

Back Bone- In the current methodology which has been adapted from Kaggle (Ari, 2021), ResNet-101 is used as the backbone of the Cascade R-CNN architecture. ResNet-101 is utilised to extract features from the input image that are passed through further in the architecture. In DNNs, the residual learning framework (ResNet) increases its layers the deeper it goes, which helps in gaining accuracy from the deep network. Current ResNet architecture has 101 layers. It has four residual stages and the output feature maps from these four residual stages will be used in further processing. ‘PyTorch’ pre-trained weights from the ImageNet dataset have been initialised in the backbone. This is advantageous in catching typical image representations. Initialising weights from a pre-trained model is the initial part of the transfer learning technique employed in this methodology. Batch Normalization (BN) is also included in the current configuration. Typically, BN normalizes the input features and improves the training process. It helps in tackling a problem called the “internal covariate shift problem”. This is common when training a layer in a deep network. The problem comes up when updating the weights of the layers, as the model assumes the weights in earlier layers

are constant, updating earlier layers will change the distribution of the input to the next layer, and so on and forth. The weights in the backbone layer are assigned before the training process, and they are adjusted by a process called “Back Propagation”, which means calculating and updating weights that help in minimising the difference between ground truth labels and predicted outputs.

Deformable Convolution (DCN) is used in the later stages of ResNet-101, but regular convolution layers have been used in the first stage. DCNs are helpful in object detection tasks, where localising the objects of different sizes and ratios such as multiple storage tank sizes and their width-to-height ratio called the aspect ratio. During the pre-processing stage, the oil tanks with annotated bounding boxes with irregular aspect ratios are eliminated to avoid false sampling. They are different from regular convolution operations where DCN does flexible sampling of points rather than a fixed grid sampling (Figure 8). A network with DCN adapts adjusting itself in selecting samples based on the input data. The images fed into the network have objects with various spatial deformations, spatial transformations, and geometric variations. Each sampling point is paired with an offset vector which tells how much the sampling point is shifted from its original position. These offsets are learned in the training and convolution kernels adapt the sampling positions dynamically. Generally, these DCNs are computationally very costly, as they are used to extract fine-detailed spatial features. The earlier stage of ResNet-101 will produce a set of proposals of object locations using fixed anchor box sizes. These proposals are refined at the later stage, where DCNs are most beneficial.

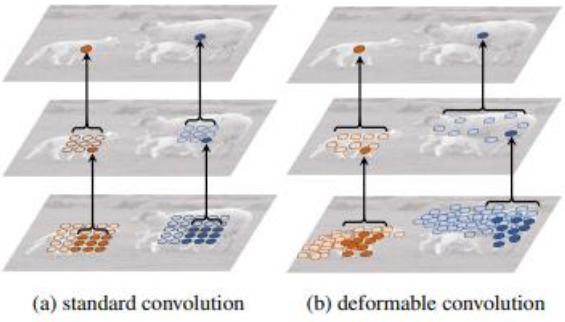

Figure.8- Comparison between regular and deformable convolution (Source- Dai, J. et.al., 2017)

Neck - A feature Pyramid Network (FPN) has been used in the neck of the model configuration. It is used to detect objects at different scales by enhancing the multi-scale feature representation of the input image (Hui, J., 2020). The backbone extract feature maps at different spatial resolutions which are fed into the FPN. These maps will have both low-level and high-level features where low-level feature maps are in fine-grained detail (high-resolution), and high-level maps have low-resolution but high semantic information. FPN

facilitates both bottom-up and top-down pathways to produce high-resolution layers from a rich semantic layer. The output of the bottom-up pathway is used to enrich the top-down pathway feature maps by using the bottom-up output as a reference (Figure-9). The FPN used in the neck of this study's model configuration takes high-resolution feature maps from the backbone and applies 1x1 convolution to reduce the number of channels. This decreases the level of complexity in the feature maps and preserves important features. Output from this top-down pathway works as an initial representation of the FPN.

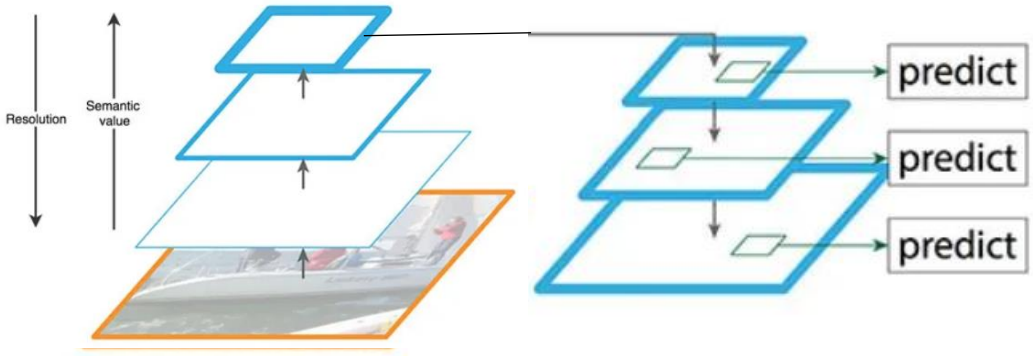

Figure.9- Feature Pyramid Network (Lin, T.-Y., *et. al.,* 2017*)*

FPN has a function called "Lateral connections" that merges low-res feature maps from the backbone to the high-res up-sampled. Lateral connections have 1x1 convolutions which transform low-res features to have the same number of channels as high-res features. Upsampling takes after the latter connections step in the network. Merge features from the latter connections step are up-sampled to have the resolution of a high-res feature map. Up-sampling follows the interpolation techniques such as bilinear up-sampling. These up-sampled features are summed element by element with their relative high-resolution features, resulting in fine-grained details. The process of 'Lateral Connections' and 'Up-sampling and Summing' is repeated iteratively by creating a feature pyramid. At each iteration, feature maps are refined through both steps, gradually incorporating multi-scale information. The output pyramid from the FPN in the neck of the model configuration provides a feature pyramid that is constructed by stacking the refined feature maps from all the levels. These feature maps contain rich information at multi scales that can enable objects to be detected at varying sizes.

Head (RPN) – Region Proposal Network (RPN) is set in the head of Cascade R-CNN model configuration. Input for the RPN is the feature maps produced from the neck (FPN) of the model configuration. The main purpose of the RPN is to generate a set of proposals, in other words, potential bounding box proposals of objects in each input image. The sliding window

technique, with a window size of 1x1, is applied to each spatial position in the input feature map. This will result in a set of anchor boxes with fixed sizes. The difference between Cascade RCNN to YOLOX discussed in the earlier sections is that YOLOX is an anchor-free based approach. Classification is then performed on each anchor box as RPN predicts two values, named “Objectness score and “Bounding box regression offsets”. The objectness score by RPN corresponds to the probability of an object of interest in the anchor box with the help of a binary classifier. BBox offsets produced by RPN are utilised to adjust the size and position of anchor boxes to align with the objects of interest in the image. With these two values, RPN can rank the anchor boxes and subset proposals from earlier stages. The total number of proposals generated is controlled through a parameter, called “*pre_nms_top_n*” provided by the model configuration. A technique called “Non-Maximum Suppression” (NMS) has been applied to these extracted proposals to suppress any redundant ones. This technique suppresses any highly overlapping proposals and sustains the proposal with the most confidence score. There is another parameter “*post_nms_top_n*” which controls the total of number proposals after the NMS step.

Head (ROI) – These selected region proposals from RPN are passed through a Region of Interest (ROI) align operation, as the name suggests, aligning region proposals with the relative features from the FPN. ROI has multiple stages in which each stage has a shared two-layer fully connected head (2FC). This operation is imperative in the context of the model configuration as it helps in extracting precise features from arbitrarily shaped regions. This operation segments the ROI into spatial bins with a fixed-size grid (Figure-10). In each bin, the features in that bin are bilinearly interpolated from the feature map of the corresponding pyramid level. The input for this ROI is not the original image but the feature map with a reduced size from the previous step. Interpolation ensures that the alignment is accurate with the ROI. These interpolated features are pooled in each bin by an operation of Max pooling (Figure 10). These features are further processed by DCNN in the later stages and generate final RoI-aligned features. Later in the stages, a set of classifiers and regression heads have been used to refine the proposals and improve the accuracy. The classifier head in each stage helps in predicting the class probabilities of each proposal, and on the other hand regression head predicts refined bbox coordinates. Cascade thresholding is applied at each stage to filter proposals with low confidence. The final prediction happens in the last stage of the network where the remaining proposals are used. Both the class probabilities and refined bbox coordinates are used to make the predictions, including class labels and accurate bounding box locations.

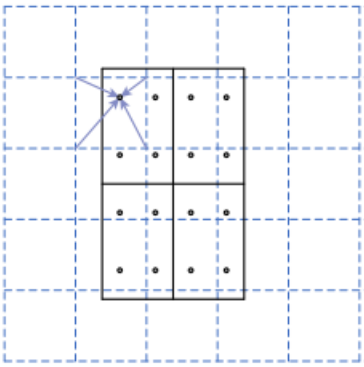

Figure 10 RoI Align operation, dashed lines – feature map, sloid line – bin, dots – sampling points (Source – He, K. et.al., 2018)

The CNN model is trained on a combination of cross-entropy loss for classification and smooth L1 loss for bounding box regression. The training pipeline utilised various data augmentation techniques such as scaling, rotation, random flipping, and contrast adjustment. The training of the model is followed by validation where the model uses data augmentation to improve the model's performance. It also applies padding and normalization steps that have been followed in the training step. Details of the model optimization are as follows:

- The model utilises stochastic gradient descent (SDG) with a learning rate of 0.0025, a momentum of 0.9, and a weight decay of 0.0001.
- The model has been trained for a total of 12 epochs utilising an epoch-based runner.
- The model is initialised at a pre-trained checkpoint, meaning, the model configuration has been started with a pre-trained model.

This methodology takes in the input of large images in the COCO format with unrestricted size. The trade-off between the two methodologies is that YOLOX has higher inference speeds and less training time with low accuracy. As it is an anchor-free based model, inference can be performed on a large-scale image with the training and validation performed on image patches.

Chapter-4 Results and Discussion

This study has begun with the idea of utilising SAR backscatter data to identify the storage tank objects in the SAR images. Sentinel-1 C band SAR data is publicly available and data acquisition and pre-processing of SAR data has been engineered and automated for this study. As it turned out that the resolution for Sentinel-1 was not optimal for the detection of objects less than 10m in diameter. There are numerous deep-learning studies on SAR object detection. As mentioned in the background and previous work section, Wu, Q. *et al.,* (2022) worked on half a meter SAR data and utilised the YOLOX model where they reconfigured by adding a Transformer encoder, self-attention modules and replaced backbone CNN layers with structural parameterised Visual Geometry Group (VGG) blocks. Ma, C, *et. al.,* (2022) has considered estimating structural projection points of oil storage tanks. They experimented on RADARSAT-2 data which has a resolution of 3m. The analysis resulted in the 3D structural estimation of the object which provides total volume occupancy and backscattered points of the side wall which provides the floating rooftop height. Both papers worked with a very coarse resolution for their analysis, which is relatively very high compared to Sentinel-1 GRD product data (Figure-11).

Figure.11-Sentinel-1 GRD processed data; Top-Optical satellite image, bottom-S1 GRD data, Immingham Port, England (processed with ESA SNAP)

Readily available satellite optical image datasets seem much more practical and feasible for this study. Two optical datasets with similar spatial resolutions have been used for training purposes. The Google Earth oil tank dataset has been trained on the above adapted Cascade RCNN object detection framework, and the robustness has been evaluated with the performance metrics of loss functions. Airbus SPOT images are trained on YOLOX architecture. There are 1000 high-resolution images of oil storage tank sites from all around the world with a combination of dense urban sites and remote rural sites in the Google dataset, whereas there are 100 large-scale images in the Airbus dataset. Each image is split into 512 x 512 x 3 for the initial convolutions. Training and validation have been performed on the split dataset between annotated and non-annotated sets of classes. During the training process, all the image patches have gone through down-sampling and up-sampling which resulted in

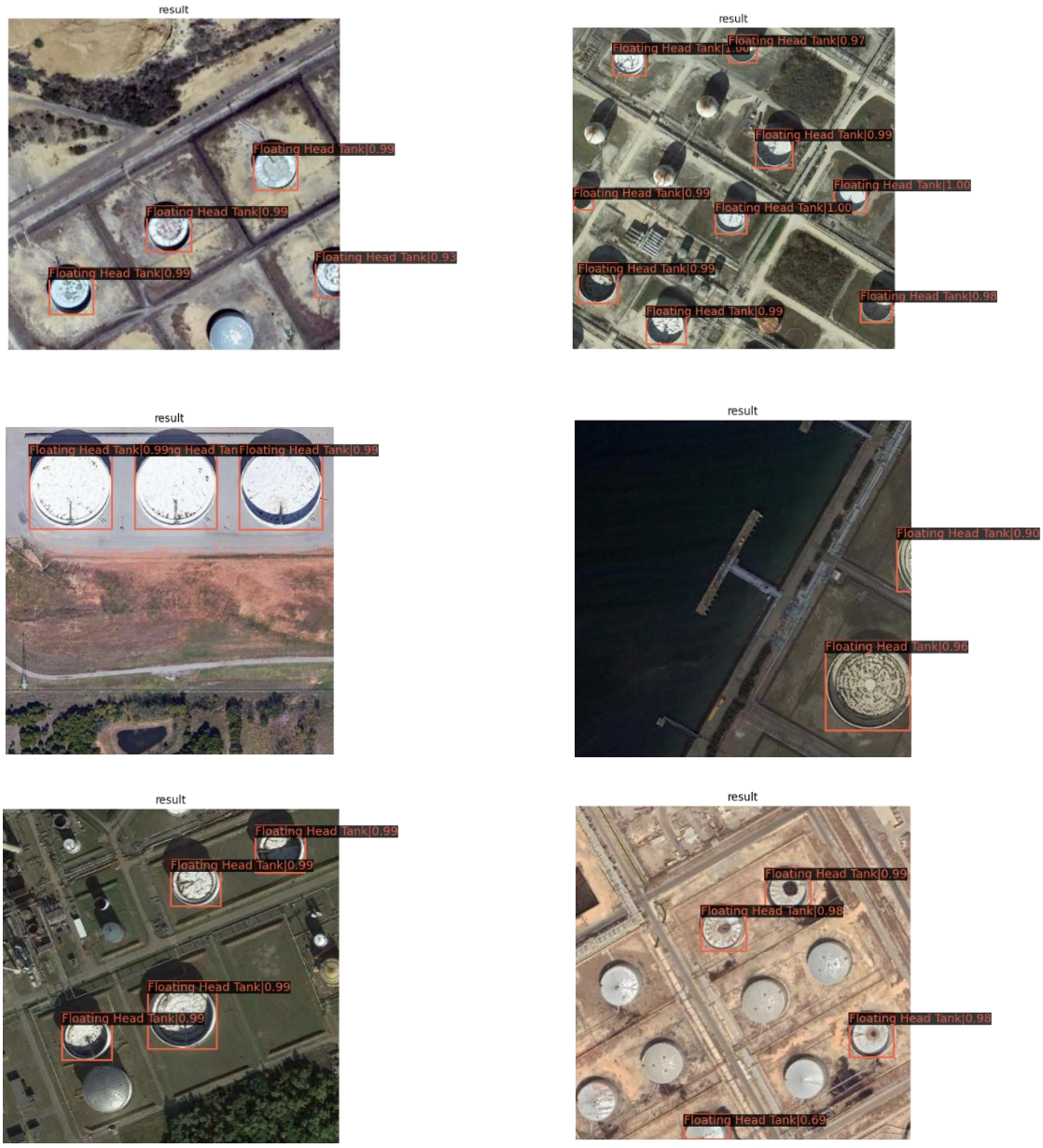

Figure.12- Cascade-RCNN: Floating Head Tanks detection

the extraction of low-level features of the high-res image. This process helps the model in identifying features in low-res images during the inference stage. Inference has been performed on random images with labelled and non-labelled annotations. CNN model mainly

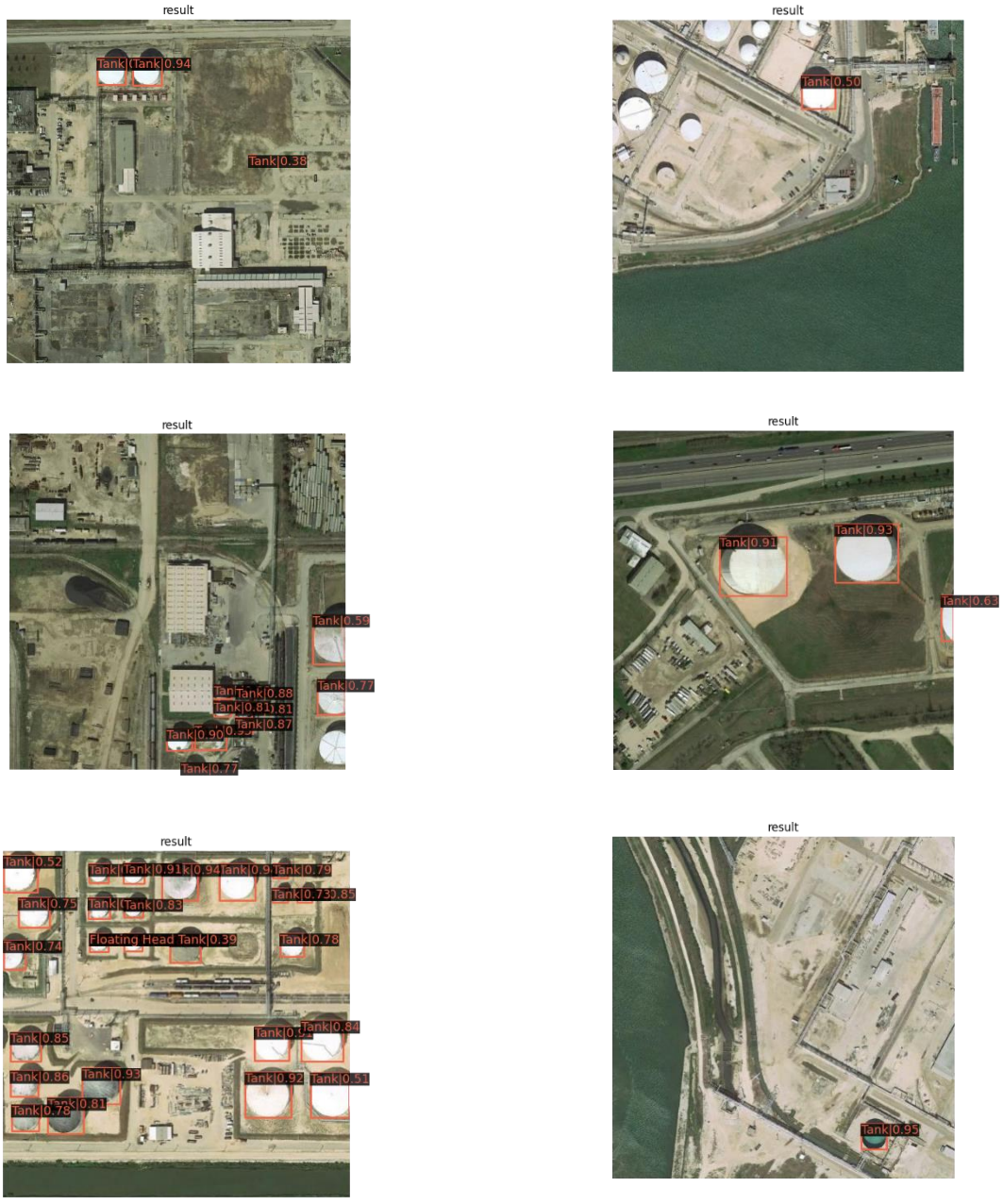

Figure.13- Cascade RCNN: Fixed head tank detection (False positive detected on the bottom right image)

consists of two objectives, classification, and detection. Performance metrics (loss functions) help in optimising an objection detection model by iteratively adjusting the parameters and improving the prediction accuracy of an object. As mentioned in the methodology, cross-entropy loss for classification and smooth L1 loss for bounding box regression (object detection) has been used to evaluate the performance and improve the Cascade RCNN model iteratively. These loss functions quantify the difference and discrepancy of model predictions from the ground truth. In other words, it calculates or measures the model's deviation from

accurate prediction. Figure-14 shows the RPN losses for the CNN model. RPN is where the classification and bbox regression takes place. RPN losses decrease iteratively, conveying that the model is minimising its loss in the RPN stage of training.

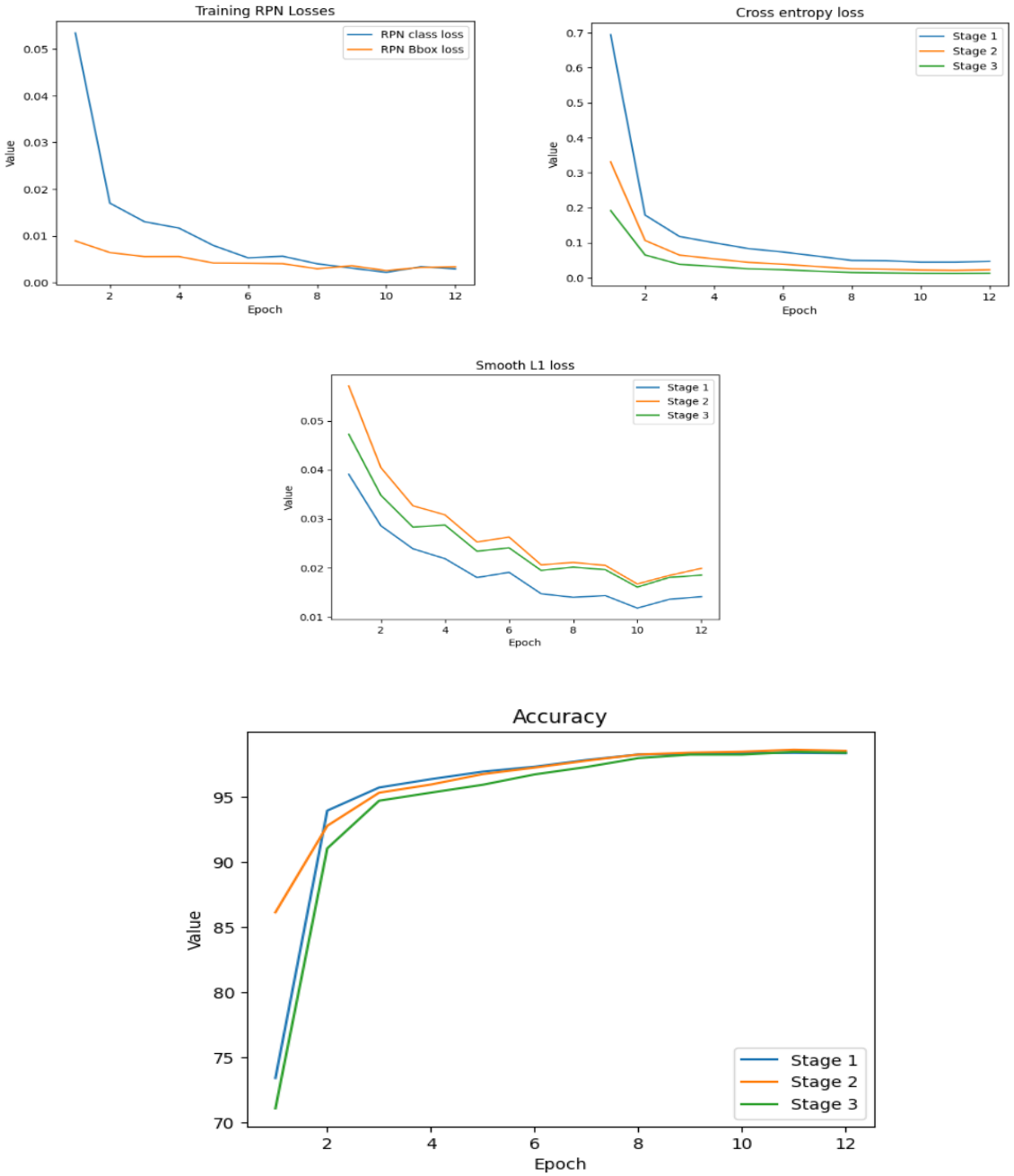

Figure.14- Cascade R-CNN model performance metrics

Cross entropy loss (Eq(1)) can be defined as the sum of the negative logarithmic values of probabilities of each class (tank, floated head tank, tank cluster) in each image, where 'y' is the binary value (0 or 1) of class containing in each image or a target prediction, 'j' is the number of each class (1,2,3), $f(s_i)$ is the activate function of inputs or weights (s_i), and 'l' is

the number of events or images fed into the network. The sigmoid activation function has been used for the conversion of inputs of weight probabilities into continuous variables. It has been used in the RPN head but is disabled in the ROI head as the ROI head is intended for multi-class classification of assigning labels to the proposed region of interests. Cross entropy loss in each stage is observed to be falling (Figure-14) as the model has been progressively improving through each stage. Classification between two of the three classes mentioned above is observed in Figures 12 and 13 where the CNN model can classify different oil storage tank types.

$$-\sum_{n=i}^{n}\sum_{j=1}^{c} y_i\ j \log\left(f(s_i)j\right) \qquad \text{Eq (1)}$$

As a deep learning object detection model has multiple objectives, in the case of this study, objectives such as the classification of the detected object and placing a bounding box around the object, performance metrics help in balancing the objectives' significance by assigning weights to different components. Gradient values are also calculated through loss functions, as these gradient values provide the direction and magnitude of the adjustments that are needed to be made for loss minimization. The other objective was to assign bounding boxes for a predicted object. Smooth L1 loss is a common loss function utilised in bounding box regression of object detection tasks. The loss is observed to be minimising progressively in the epochs (Figure 14). Smooth L1 loss is also known as Huber loss, it improves upon the limitations set by the traditional L1 loss. L1 loss which is also called mean absolute error (MAE) is the average sum of the absolute difference between actual predicted and ground truth values. Smooth L1 loss responds well with outliers and is sensitive to small errors in the bbox predictions.

Cascade RCNN and YOLO models have been trained on a total of 12 epochs and 10 epochs respectively using an epoch-based runner, with each epoch containing 50 iterations, and a validation step for every 4th epoch in the CNN model and for every epoch in the YOLO model. During the CNN model training process, the optimisation algorithm progressively updates the parameters to minimize the loss function, as each iteration takes a batch of the training data into the network, computing losses and performing backpropagation to update the model parameters. This process is repeated by a few iterations set (50) until the criteria of convergence are met. Intersection over Union (IOU) of 0.5, 0.6, and 0.7 are set for stages one, two, and three respectively. An accuracy of 97% is achieved in the CNN model at the end of the 12th epoch in the training process (Figure-14) whereas mAP (Mean Average Precision) of 88.7, 71.2, and 45.5 for large, medium, and small objects have been achieved in YOLO model as evidence in figure 15.

Figure.15- YOLOX model inference on Planet Labs scene (Stanlow port, England. Date of capture: 20-5-2023)

Chapter-5 Conclusion

Compared to low-resolution Sentinel-1 SAR data, optical datasets with very high resolution are ideal for transfer learning. As there are pre-trained detection models available in the open-source community, features, and representations of an object are neatly derived using these pre-trained models and can be deployed in specific tasks such as classifying and detecting storage oil tanks. Kaggle provided two oil storage tanks' image datasets. These datasets, as mentioned above have been filled with oil storage tank images from all around the world. Both datasets have been provided with the labelled annotations of storage tanks in each image. These annotations include the pixel locations of each storage tank and the min and max bounding box values. These are essential in the training process as the model performs feature extractions and maps the features on a new image. Machine learning object detection models demand high computational costs when training from scratch, but with pre-trained model usage and transfer of weights, it is feasible to scale the process spatially. The performance of the models is improved with fine-tuning of the parameters during the optimization process. During this study, an end-to-end object detection pipeline of Planet data has been developed with a Planet Labs scene as an input the output displays the detected oil tanks in the scene (Figure 14).

Inference in High-resolution SAR data can be made possible with a complimentary optical data CNN network which is trained to extract features based on a “Teacher-Student” network. There has been a precedent for this methodology in the literature (Zhang, R. *et.al.*, 2022), and further research and development of this study can be progress toward CNN of different modalities.

References

1. Ciocarlan, A. and Stoian, A. (2021) 'Ship Detection in Sentinel 2 Multi-Spectral Images with Self-Supervised Learning', *Remote Sensing*, 13(21), pp.4255. Available at: https://doi.org/10.3390/rs13214255.
2. Abba, A., Musa, K., Umar, A., Saleh, N., Khamis, H. and Dauda, M. (2020), 'Transfer learning strategy for satellite image classification using deep convolutional neural network', *International Journal of Advanced Engineering and Management Research*, 5(04). Available at: https://www.ijaemr.com/uploads/pdf/archivepdf/2020/IJAEMR_411.pdf
3. Pires de Lima, R. and Marfurt, K. (2019), 'Convolutional Neural Network for Remote-Sensing Scene Classification: Transfer Learning Analysis', *Remote Sensing*, 12(1), pp.86. Available at: https://doi.org/10.3390/rs12010086.
4. Clifford, K., Stanley, R. (2020) 'Too Much Oil: How a Barrel Came to Be Worth Less Than Nothing', New York Times, 20 April. Available at: https://www.nytimes.com/2020/04/20/business/oil-prices.html (Accessed 09 May 2023)
5. Naveen, K., Chaudhuri, D., Manish, S. (2013) 'Automatic Bright Circular Type Oil Tank Detection Using Remote Sensing Images', *Defense Science Journal*, 63, pp. 298-304. Available at: https://www.researchgate.net/publication/273756854_Automatic_Bright_Circular_Type_Oil_Tank_Detection_Using_Remote_Sensing_Images
6. Cui, Z., Guo, W., Zhang, Z., Chen, H., and Yu, W. (2020) 'Ellipse-FCN: Oil Tanks Detection from Remote Sensing Images with Fully Convolution Network', *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, Waikoloa, HI, USA*. pp. 2855-2858. Available at: https://ieeexplore.ieee.org/document/9324631
7. Jing, M., Zhao, D., Zhou, M., Gao, Y., Jiang, Z., and Shi, Z. (2018) 'Unsupervised Oil Tank Detection by Shape-Guide Saliency Model', *IEEE Geoscience and Remote Sensing Letters,* 16(3), pp. 477-481. Available at: https://ieeexplore.ieee.org/document/8502883
8. Airbusgeo-Kaggle, *Airbus Oil Storage Detection*. Available at: https://www.kaggle.com/datasets/airbusgeo/airbus-oil-storage-detection-dataset (Accessed 10 May 2023).

9. Karl, H. (2019). *Oil Storage Tanks*. Available at: https://www.kaggle.com/datasets/towardsentropy/oil-storage-tanks (Accessed 12 May 2023).
10. SpaceNet (no date). *Multi-Sensor All-Weather Mapping*. Available at: https://spacenet.ai/sn6-challenge/ (Accessed 10 May 2023).
11. Wu, Y., Zhang, Z. and Wang, G. (2019) 'Unsupervised Deep Feature Transfer for Low Resolution Image Classification'. Available at: https://openaccess.thecvf.com/content_ICCVW_2019/papers/RLQ/Wu_Unsupervised_Deep_Feature_Transfer_for_Low_Resolution_Image_Classification_ICCVW_2019_paper.pdf
12. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J. and Zisserman, A. (2014). 'The Pascal Visual Object Classes Challenge: A Retrospective', *International Journal of Computer Vision*, 111(1), pp.98–136. Available at: https://doi.org/10.1007/s11263-014-0733-5.
13. Moein, Z., Gholamreza, A., and Navid, A.S. (2020) 'A new approach for oil tank detection using deep learning features with control false alarm rate in high-resolution satellite imagery', *International Journal of Remote Sensing*, 41(6), pp.2239-2262. Available at: https://doi.org/10.1080/01431161.2019.1685720
14. Ok, A. O. (2014) 'A New Approach for the Extraction of Aboveground Circular Structures from Near-Nadir VHR Satellite Imagery', *IEEE Transactions on Geoscience and Remote Sensing*, 52(6), pp. 3125-3140. Available at: https://ieeexplore.ieee.org/document/6557527
15. L. Zhang and Liu, C. (2019) 'Oil tank detection based on linear clustering saliency analysis for synthetic aperture radar images', *Proc. IEEE Int. Conf. Image Process. (ICIP)*, pp. 2981-2985, Available at: https://ieeexplore-ieee-org.ezproxy3.lib.le.ac.uk/document/8803347
16. Zerman, E., Batı, E., Akar, G. B., Başeski, E., and Düzgün, Ş. (2014) 'Circular target detection algorithm on satellite images based on radial transformation', *Proc. 22nd Signal Process. Commun. Appl. Conf. (SIU),* pp. 1790-1793, Available at: https://ieeexplore-ieee-org.ezproxy3.lib.le.ac.uk/document/6830598
17. Yu, B., Chen, F., Wang, Y., Wang, N., Yang, X., Ma, P., Zhou, C. and Zhang, Y. (2021) 'Res2-Unet+, a Practical Oil Tank Detection Network for Large-Scale High Spatial Resolution Images', *Remote Sensing* 13(23), p.4740. Available at: https://doi.org/10.3390/rs13234740
18. Xu, D. and Wu, Y. (2020) 'Improved YOLO-V3 with DenseNet for Multi-Scale Remote Sensing Target Detection', *Sensors*, 20(15), p.4276. Available at: https://doi.org/10.3390/s20154276.

19. Bakirman, T. (2023), 'An Assessment of YOLO Architectures for Oil Tank Detection from SPOT Imagery', *International Journal of Environment and Geoinformatics*, 10(1), pp.9–15. Available at: https://doi.org/10.30897/ijegeo.1196817
20. Wu, Q., Zhang, B., Xu, C., Zhang, H. and Wang, C. (2022), 'Dense Oil Tank Detection and Classification via YOLOX-TR Network in Large-Scale SAR Images', *Remote Sensing*, 14(14), p.3246. Available at: https://doi.org/10.3390/rs14143246.
21. Athanosis, P. (2022), 'Reserve volume estimation of oil storage tanks based on remote sensing images', *Diploma-Thesis National Technical University of Athens*, Available at: https://dspace.lib.ntua.gr/xmlui/bitstream/handle/123456789/56490/NTUA_ECE_Thesis_fn.pdf?sequence=1
22. Zou, M. and Zhong, Y. (2018), 'Transfer Learning for Classification of Optical Satellite Image',19(1). Available at: https://doi.org/10.1007/s11220-018-0191-1.
23. Maggiori, E., Tarabalka, Y., Charpiat, G., & Alliez, P. (2017). Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(2), 645–657. Available at: https://doi-org.ezproxy3.lib.le.ac.uk/10.1109/TGRS.2016.2612821.
24. Zhang, R., Guo, H., Xu, F., Yang, W., Yu, H., Zhang, H., Xia, G. -S., (2022), 'Optical-Enhanced Oil Tank Detection in High-Resolution SAR Images', *IEEE Transactions on Geoscience and Remote Sensing*, 60, pp. 1-12, Available at: https://10.1109/TGRS.2022.3215543.
25. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H. and Wei, Y., (2017), 'Deformable Convolutional Networks', *Computer Vision Foundation - IEEE Xplore*, Available at: https://openaccess.thecvf.com/content_ICCV_2017/papers/Dai_Deformable_Convolutional_Networks_ICCV_2017_paper.pdf
26. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S., (2017), 'Feature Pyramid Networks for Object Detection'. Available at: https://arxiv.org/pdf/1612.03144.pdf.
27. Hui, J. (2020), 'Understanding Feature Pyramid Networks for object detection (FPN), Available at: https://jonathan-hui.medium.com/understanding-feature-pyramid-networks-for-object-detection-fpn-45b227b9106c. Accessed on: 29 May 2023
28. Ma, C., Zhang, Y., Guo, J., Hu, Y., Geng, X., Li, F., Lei, B. and Ding, C., (2022), 'Structural projection points estimation and context priors for oil tank storage estimation in SAR image', *ISPRS Journal of Photogrammetry and Remote Sensing*, 194, pp.267–285. Available at https://doi.org/10.1016/j.isprsjprs.2022.10.016.

29. learnopencv.com. (2022), ‘YOLOX Object Detector Paper Explanation and Custom Training’, Available at: https://learnopencv.com/yolox-object-detector-paper-explanation-and-custom-training/ (Accessed 31 May 2023).
30. Ari, (2021), ‘Oil Tank Detection – MMDetection’, Available at: https://www.kaggle.com/code/vexxingbanana/oil-tank-detection-mmdetection/notebook (Accessed 2 June 2023).