Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Information Networks Department

Marine Oil Spill Detection using Deep Learning Approaches Based on Cloud Environment

A Thesis

Submitted to the Council of the College of Information Technology for the Postgraduate Studies/University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology/Information Networks

Alaa Akram Jawad Muhammad Huby

Supervised by

Assoc. Prof. Dr. Rafid Sagban Abbood Kareem Al Jubouri

Asst. Prof. Dr. Raaid Nasur Kadham Khalil Al-Obaidi

2023 A.D **1444 A.H**

Supervisor Certification

I certify that the thesis entitled (**Marine Oil Spill Detection using Deep Learning Approaches Based on Cloud Environment**) was prepared under my supervision Asst. Prof. Dr. Rafid Sakban and Asst. Prof. Dr. Raaid N. Alubady at the department of Information Networks/ College of Information Technology/the University of Babylon as partial fulfillment of the requirements of the degree of master's in information technology-Information Networks.

Signature:
Supervisor Name: Asst. Prof. Dr. Rafid Sagban Abbood Kareem
Date: / /2023

Signature:
Supervisor Name: Asst. Prof. Dr. Raaid Nasur Kadham Khalil
Date: / /2023

The Head of the Department Certification

Given the available recommendations, I forward the thesis entitled (**Marine Oil Spill Detection using Deep Learning Approaches Based on Cloud Environment**) for debate by the examination committee.

Signature:
Prof. Dr. Saad Talib Hasson
Head of Information Networks Department
Date: / /2023

Certification of the Examination Committee

We hereby certify that we have studied the thesis entitled (**Marine Oil Spill Detection using Deep Learning Approaches Based on Cloud Environment**) presented by the student (**Alaa Akram Jawad Mohammed**) and examined him/her in its content and what is related to it, and that, in our opinion, it is adequate with **(Excellent)** standing as a thesis for the degree of master's in information technology-Information Networks.

Signature:
Name: Mahdi S. Almhanna
Title: Asst. Prof. Dr
Date: / / 2023
(Chairman)

Signature:
Name: Sarmad K. Dhiab
Title: Asst. Prof. Dr
Date: / / 2023
(Member)

Signature:
Name: Saba M. Hussain
Title: Lecturer
Date: / / 2023
(Member)

Signature:
Name: Rafid S. Abbood
Title: Asst. Prof. Dr
Date: / / 2023
(**Member and Supervisor**)

Signature: :
Name: Raaid N. Alubady
Title: Asst. Prof. Dr
Date: / / 2023
(**Member and Supervisor**)

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature:
Name: Dr. Hussein Atiya Lafta
Title: Professor
Date: / / 2023
(Dean of Collage of Information Technology)

Dedication

I dedicate this effort:

For my parenets who helped me accomplish my goals by putting me in thier place.

Those who have helped me complete the thesis are greatly appreciated.

Acknowledgements

First I would like to thank our GOD “Allah”, Who gave me strength, confidence, and courage to complete this task successfully. as well as All thanks and appreciation are to the Messenger Muhammad, may God bless him and his family.

I would like to express my sincere appreciation and respect to my supervisor **Assoc. Prof. Dr. Rafid Sagban Abbood and Asst. Prof. Dr. Raaid N. Alubady** for their continuous support, patience, motivation, and immense knowledge. They give me many interesting, valuable, and sincere feedbacks throughout their supervision to complete this thesis. I could not have imagined having a better supervisor for my master's study.

I would also like to say a big thank to all the professors and lecturers of the University of Babylon, especially whose shared their words of advice and encouragement with me.

Abstract

Oil leaks onto water surfaces from big tankers, ships, and pipeline cracks cause considerable damage and harm to the marine environment. Synthetic Aperture Radar (SAR) images provide an approximate representation of target scenes, including sea and land surfaces, ships, oil spills, and look-alikes. Detection and segmentation of oil spills from SAR images are crucial to leak cleanups and protect the environment. This thesis introduces deep-learning framework for the detect oil spill used U-Net semantic segmentation technique. The Encoder part of Unet is replaced with pre-train model (transform learning) using (Densnet201, Inception V3, efficientNetb3, ReseNet101) that models are previously trained on Imagnet dataset. In contrast, the Decoder part uses U-Net structure. The dataset classes to five groups with 512 × 512 spatial dimensions and their respective annotations (800 for training and 202 for valdation and 110 for testing) so the total will be 1112. The U-net with the Densenet201 backbone presented slightly better results (92% accuracy, 56% precision, 80% recall, 74% F-score, and 69% IoU). Moreover, the best models performence save in demo and upload to VPS to test on real image with full size (1250*650) and calculate pixel area for each class. The results of this study are very promising and provide a comparable improved IoU compared to related work.

Declaration Associated with this Thesis

i. Alaa Akram Huby, Rafid Sagban, Raaid Alubady. "Oil Spill Detection based on Machine Learning and Deep Learning: A Review", *2022 5th International Conference on Engineering Technology and its Applications (IICETA)*, 2022.

ii. Alaa Akram Huby, Rafid Sagban, Raaid Alubady. "Oil Spill Segmentation from SAR Images Using Deep Neural Networks", *2022 5th International Conference on Engineering Technology and its Applications (IICETA)*, 2022.

Table of Content

Dedication ……	**i**
Acknowledgements…….	**ii**
Abstract ……..	**iii**
Declaration Associated with this Thesis ……..	**iv**
Table of Contents ……..……..	**vi**
List of Figures …….	**x**
List of Tables ……..	**xii**
List of Abbreviations …….	**xiii**

Chapter One: Introduction

1.1 Introduction ……	**1**
1.2 Related Works ……	**2**
1.3 Problem Statement ……	**8**
1.4 Research Objectives ……	**9**
1.5 Organization Of The Thesis ……..	**9**

Chapter Two: Theoretical and Background

2.1 Introduction ……..	**11**
2.2 Oil Spill Problem ……	**11**
2.2.1 The History Of Oil Spill ……	**11**
2.2.2 The Impact Of Oil Spill ……...	**12**
2.3 Oil Spill Dataset ……	**14**
2.3.1 Synthetic Aperture Radars (SAR) ……	**14**
2.3.2 Drone Images ……..	**15**
2.3.3 Advantages And Limitations of SAR and Drone……	**16**
2.4 Deep Learning Algorithms (DL) ……..……	**16**
2.4.1 Recurrent Neural Network (RNN) ……...	**17**

2.4.2 Convolutional Neural Network (CNN) ……	18
2.4.3 Classification Vs Segmentation …….	20
2.5 U-Net Semantic Segmentation Technique ……	21
2.5.1 U-Net Backbone ……..	22
2.5.1.1 ResNet ……	22
2.5.1.2 DenseNet ……	23
2.5.1.3 EfficientNet ……...	24
2.5.1.4 BN-Inception And Inception-V3 ……	25
2.5.2 Transform Learning ……..	26
2.5.3 Activation Functions ……..	26
2.5.3.1 Loss Functions (Cost Function) ……	27
2.5.4 Optimizers ……...	28
2.6 Technologies For Image Preprocessing ……	29
2.6.1 Data Augmentation …….	29
2.7 Cloud Computing ……..	31
2.7.1 Deployment Cloud Models …….	32
2.7.2 Service Models …….	34
2.8 Network Protocols ……..	35
2.8.1 Transmission Control Protocol/Internet Protocol (TCP/IP) …….	35
2.8.2 File Transfer Protocol (FTP) ……..	36
2.8.3 Remote Desktop Connection (RDP) ……..	36
2.9 Performance Evaluation ……	37
2.9.1 Intersection Over Union (IOU) ……..	37
2.9.2 Pixel Accuracy …….	37
2.9.3 Precision ……...	37
2.9.4 Recall ……..	37
2.9.5 Dice Coefficient (F1 Score) ……..	38

Chapter Three: Research Methodology & Proposed Method

3.1 Introduction ……	39
3.2 Proposed System ……..…….	39
3.3 M4D Dataset ……..	42

3.4 Preprocessing Phase ……..	**43**
3.4.1 Split Image ……...	**43**
3.4.2 Augmentation ……..…….	**44**
3.5 U-Net Semantic Segmentation Model …….……	**45**
3.5.1 Encoder Part …….……	**48**
3.5.2 Decoder Part ……..	**48**
3.5.2.1 Densenet-201 Backbone …….…….	**49**
3.5.2.2 Inception V3 ……	**50**
3.5.2.3 Efficientnet-B3 Backbone ……..	**51**
3.6 Virtual Private Server (VPS) Cloud ……	**53**
3.6.1 Remote Desktop Protocol (RDP) …….……	**55**
3.6.2 VPS Setup ……	**55**
3.6.3 File Transfer Protocol (FTP) ……	**55**
3.6.4 Connect Client With VPS …….……	**56**
3.6.4.1 Client Side ……	**56**
3.6.4.2 Server Side ……..….	**57**

Chapter Four: Results And Discussion

4.1 Introduction …….……	**58**
4.2 SAR Images Dataset Preparations ……..	**59**
4.3 Results Of The Implementation Of The Proposed System ……	**62**
4.3.1 Training Phase Outputs ……	**62**
4.3.1.1 U-Net ……	**63**
4.3.1.2 Densenet201 ……..	**63**
4.3.1.3 Efficientnetb3 ……...	**64**
4.3.1.4 Inception -V3 ……..……..	**65**
4.3.1.5 Resnet101 …….……	**65**
4.3.2 Testing Phase Results ……	**66**
4.4 Comparison Of The Proposed System Results With Related Works …….	**71**
4.5 Deploy Models On VPS ……	**72**

Chapter Five: Conclusion And Future Works

5.1 Conclusion …….	**75**
5.3 Future Works ……...	**76**
REFERENCES ……	**77**
APPENDIX ……...	**86**

List of Figures

FIGURE 2.1 MAJOR INPUTS OF PETROLEUM INTO THE WORLD……..…12
FIGURE 2.2 BIRD DEATHS FROM THE EXXON VALDEZ AND THE BRAER OIL SPILLS [3]……..13
FIGURE 2.3 IMAGES FROM FOUR SEPARATE SAR OIL SPILLS [28]…..…..15
FIGURE 2.4 MQ-4C 'TRITON' LARGE UAV [32]…….……...……..…15
FIGURE 2.5 TYPICAL UNFOLDED RNN DIAGRAM [35]……..……17
FIGURE 2.6 AN EXAMPLE OF CNN ARCHITECTURE FOR IMAGE CLASSIFICATIO[42]……..……..19
FIGURE 2.7 TYPES OF OBJECT DETECTION [44] …….……..……...20
FIGURE 2.8 U-NET: CONVOLUTIONAL NETWORKS [47] …...…..……..…….22
FIGURE 2.9 RESNET [48] ……..…...……..…….23
FIGURE 2.10 DENSENET [48] …….……..…..24
FIGURE 2.11 EFFICIENTNET [48] ……...……25
FIGURE 2.12 ACTIVATION FUNCTIONS [54] …….……..27
FIGURE 2.13 APPLICATION AND OUTCOME OF DEEP LEARNING ALGORITHMS [56] ……...…29
FIGURE 2.14 SUMMARY OF THE VARIOUS FEATURES OF CLOUD DEPLOYMENT MODELS [63] ……...33
FIGURE 2.15 CLOUD DEPLOYMENT MODELS AND INFRASTRUCTURE [65] …….35
FIGURE 3.1: THE MACRO VIEW FOR THE PROPOSED SYSTEM ……40
FIGURE 3.2: MICRO VIEW OF THE PROPOSED SYSTEM ……..……41
FIGURE 3.3 U-NET ARCHITECTURE …….……..……47
FIGURE 3.4 ARCHITECTURE DENSENET-201 BACKBONE WITH U-NET ……..….50
FIGURE 3.5 ARCHITECTURE INCEPTION V3 BACKBONE WITH U-NET ...51
FIGURE 3.6 ARCHITECTURE EFFECTIONNET-B3 BACKBONE WITH U-NET……..52
FIGURE 3.7 VPS ENVIRONMENT …….……..54
FIGURE 3.8 REMOTE DESKTOP CONNECTION ……...…….55
FIGURE 4.1 SOME SAMPLES OF THE DATASET WITH MASK (BEFORE AND AFTER SPLIT) ……..…60

FIGURE 4.2 IMAGES WITH MASKS (AFTER AUGMENTATION) ……..61
FIGURE 4.3 THE IOU AND ACCURACY LEARNING CURVES OF TRAIN AND VALIDATION (U-NET) ……..…….63
FIGURE 4.4 THE IOU AND ACCURACY LEARNING CURVES OF TRAINING AND VALIDATION (DENSENET201) ……..……64
FIGURE 4.5 THE IOU AND ACCURACY LEARNING CURVES OF TRAINING AND VALIDATION (EFFICIENTNETB3) ……..…….64
FIGURE 4.6 THE IOU AND ACCURACY LEARNING CURVES OF TRAIN AND VALIDATION (INCEPTION V3) ……..…….65
FIGURE 4.7 THE IOU AND ACCURACY LEARNING CURVES OF TRAIN AND VALIDATION (RESNET101) ……..65
FIGURE 4.8 MODELS RESULT IN THE TEST PHASE ……67
FIGURE 4.9 EXAMPLE 1 ……...…….69
FIGURE 4.10 EXAMPLE 2 ……..……70
FIGURE 4.11 VPS SIDE ……..……73
FIGURE 4.12 APPLICATION ON THE CLIENT SIDE …….……..73

List of Tables

TABLE 1.1 SUMMARY OF RELATED WORKS …….7
TABLE 2.1 SUMMARY OF ADVANTAGES AND LIMITATIONS OF SAR AND DRONE …….16
TABLE 3.1 COLORS FOR DATASET CLASSES..... 42
TABLE 3.2 AUGMENTATION TECHNIQUES..... 45
TABLE 3.3 U-NET STRUCTURE (LAYER TYPE, OUTPUT SIZES, THE PARAMETER, CONNECT TO WHICH LAYER)..... 46
TABLE 3.4 THE ADVANTAGES AND DISADVANTAGES OF BACKBONE 53
TABLE 4.1 THE STATISTICS OF THE SAR IMAGES DIVIDE..... 59
TABLE 4.2 STATISTICS OF THE SAR IMAGES DIVIDE AFTER SPLIT 61
TABLE 4.3 SUMMARY BACKBONES RESULT IN THE TRAINING PHASE..... 66
TABLE 4.4 SUMMARY OF BACKBONES RESULT IN THE TEST PHASE 68
TABLE 4.5 SHOWS A COMPARISON OF THE PROPOSED METHOD WITH OTHER METHODS. 71
TABLE 4.6 INFORMATION ABOUT VPS..... 72
TABLE 4.7 VPS PIXEL AREA PREDICTION..... 74

List of Abbreviations

Abbreviation	Description
AI	Artificial Intelligence
CNN	Convolutional Neural Network
DCNN	Deep Convolutional Neural Networks
DenseNet CNN	Densely Connected Convolutional Networks
DL	Deep Learning
DNN	Deep Neural Networks
FCN	Fully Convolutional Network
FMN	Feature Merge Network
FTP	File Transfer Protocol
IaaS	Infrastructure as a Service
IOU	Intersection over Union
Mask RCNN	Mask Region-based Convolutional Neural Networks
ML	Machine Learning
NGL	Natural Gas Liquids
VGG	Visual Geometry Group
OSCNet	Oil Spill Convolutional Network
PaaS	Platform as a Service
RDP	Remote Desktop Connection
RNN	Recurrent Neural Network
SaaS	Software as a Service
SAR	Synthetic Aperture Radar
TCP/IP	Transmission Control Protocol/Internet Protocol
TOE	Tonnes Oil Equivalent
VPS	Virtual Private Server
UAS	Unmanned Aircraft Systems
UAV	Unmanned Aerial Vehicles

Chapter One

Introduction

1.1 Introduction

The term crude oil production refers to the amount of oil extracted from the ground after inert matter or impurities have been removed from it. It is a mineral oil made up of a mixture hydrocarbons of natural origin that vary in density and viscosity from yellow to black in color [1]. In contrast, base oil can be used in manufacturing products such as lubricating greases, motor oil, and metal processing fluids. It has become increasingly apparent that oil is becoming more and more necessary with the development of life. But oil spill at sea pose serious risks and damage marine ecosystems, making them one of the biggest challenges facing the oil and gas industries [2].

Marine environments as well as coastal communities may be seriously damaged by oil pollution. The environmental impact of an oil spill is not only immediate, but it also has a long-term impact [3]. As a result of rapid development in the field of remote sensing instruments, in recent years, this has become one of the most important methods in the detection of a marine oil spill includes Synthetic Aperture Radar (SAR), which can provide high-resolution images with wide area coverage in day and night [4].

In recent years, Artificial Intelligence (AI) has seen rapid growth, with Deep Neural Networks (DNN) being the first tool to solve problems like object detection, speech recognition, and image classification [5]. As a subfield of Machine Learning (ML), Deep Learning (DL) can provide exceptional results. It utilizes Artificial Neural Networks (ANN) to extract progressively higher-level features from a dataset using multiple layers of processing. Furthermore, it can automatically learn feature representations from raw data and generate conclusions based on these representations. In addition, it is reliable and practical [6].

The Convolutional Neural Network (CNN) is one of the most common types of DL. Generally, it is used for classification tasks, where the result is the output

of a single class label. There is no doubt that many visual tasks require, especially those involving biomedical imaging, require the input of localization. As far as assigning a class to each pixel is concerned in an image [7]. To accomplish this task, researchers have developed a new type of CNN, called U-Net segmentation, for use in biomedical image segmentation. As recently as a few years ago, Olag Ronneberger and his colleagues came up with a method for segmenting biomedical images using the U-Net architecture. The main components of the architecture are two: an encoder and a decoder [8].

Whenever a developer plans to design a model, he must choose a platform for executing the model. One of the best choices is cloud computing. It considers a powerful method of managing information and applications on demand. Because of its reliability and consistency, the organization is not required to build or maintain its system to run its business. Furthermore, cloud computing provides their customers with resources like software, applications, services, and cloud computing resources that can help them reduce costs for all types and sizes of businesses and organizations, just like a power bill, based on the amount of electricity they consume from the cloud computing resources [9]. Therefore, after emphasizing the problem of oil spills in general, it is imperative to implement an automated detection system in order to prevent oil spills from spreading to marine environments.

1.2 Related Works

In related works, several studies are proposed regarding the oil spill problem. These studies are mainly related to extracting oil spill features from different dataset images. As for the techniques that have been used to extract the features, most of the researchers used CNN models for classification or segmentation.

Li et al., [17] modified an FCN to create a DL classification model for detecting oil spills in Lanset7 and Lanset8 images using GoogleNet and ResNet. In this paper, they evaluate the strengths and weaknesses of the current SVM classification method in comparison to other methods that have been developed. The results of the experiments show that the FCN-ResNet model achieves an accuracy index of 89.64% and the FCN-GoogleNet model achieves an accuracy index of 88.25% on experimental data, respectively.

Krestenitis et al., [10] were proposed an effective method for semantic segmentation using Deep Convolutional Neural Networks (DCNNs). In addition, a dataset of publicly available SAR images is presented to serve as a standard for future oil spill detection techniques. With five types of objects (oil spills, lookalikes, ships, land, and sea surface, with the latter always considered as background) were found in 1112 images extracted from raw SAR data. In this paper, they use a defined dataset to compare the effectiveness of various well-known DCNN segmentation models. The best performance was shown by DeepLabv3+, which achieved an inference time of 117 ms on the test set and an IoU of 65.06 percent.

The Densely Connected Convolutional Networks (DenseNet CNN), an image segmentation model developed by Yaohua and Xudong [18] enhanced capability to acquire subtle visual features based on the consolidation of multistage image characteristics; features of each convolutional layer are recovered, used, and then reused. Accuracy in both recognition and classification is an important part of this. The original oil spill SAR image must be denoised and filtered before it can be fed to the CNN network. Afterwards, the CNN model is applied to the image in order to extract the SAR feature. As a final step, Soft-max is used to assign a classification to the feature. Both

experiments and calculations indicate that the recognition rates for oil slicks and oil like slicks are 95.35% and 96.77%, respectively.

Xiong and Zhou [15] suggested SAR as an approach that identifies oil spills using a CNN that extracts category characteristics automatically, thereby eliminating the non-standard manual extraction methods that are currently used to detect oil spills. SAR images of the original oil spill are filtered and denoted before they are fed into a CNN in order to extract features from the image. Finally, Softmax is used in order to classify the characteristics on the basis of their features, which is the final step in the process. As part of the experiments, the European Remote-Sensing Satellite-2 (ERS-2) SAR image data was used. Based on the results of the identification, it can be concluded that the proposed approach is very effective at distinguishing "look-alike" oil slick images as well as "oil slick" images in the image search. It has been determined through experimentation and statistical analysis that the target recognition rate for oil slicks and their near-identical counterparts is 94.25% and 95.24%, respectively, with a Kappa coefficient of 89.2%.

Zeng and Wang [13] proposed a method of dark area classification in SAR using the VGG-16, a relatively DCNN. The Oil Spill Convolutional Network (OSCNet) was utilized, which consists of 12 weight layers. The massive data set of 23,768 dark patches taken from 336 SAR images is augmented with the data augmentation method to enhance the quality of analysis. Specifically, the features that can be learned from the dataset to differentiate between them. When comparing OSCNet to custom-built features, there is simply no contest. Therefore, compared to the MLP, AAMLP, A Standard Advanced ML classifier performed better. OSCNet's classification performance is vastly enhanced. Accuracy, precision, and recall of items have all been improved from

80% to 85%, while accuracy has gone up to 94% from 80% and recall from 81% to 83%.

Yekeen and Balogun [14] created a DL model for image classification. Following several stages of preprocessing, a total of 2882 images were divided into training and testing sets, with 88% going into training and 12% going into testing. High-precision segmentation, recognition, and detection of land areas, ships, look-alikes, and oil spills is a primary goal of this research project. This will be accomplished by creating a model that can analyze the texture and shape of a target. The accuracy of a model built with Mask Region-based Convolutional Neural Networks (Mask RCNN) was measured using precision, recall, and F1-measure, yielding values of 96.4%, 96.9% and 96.8%, respectively.

The Visual Geometry Group (VGG16) model proposed by Ghorbani and Behzadan [16]. Nafta, a custom image created in this study via web mining, included 1292 images captured from three views (first-person perspective, drone, and satellite) and was used to test, validate, and train CNN models for visual recognition with an accuracy rate of 92%. In the following, oil spill segmentation (such as detecting oil spill boundaries at the pixel level) was accomplished using PSPNet and mask-R-CNN models, with a mean overlap across the Union (IoU) of 49% and 68%, respectively.

Shaban et al., [11] created a two-stage DL framework, the first of which used a novel 23-layer CNN to classify the oil spills in the SAR dataset. Therefore, the next step would be to apply semantic segmentation with a U-Net structure to the dataset. The recommended process begins with pre-processing the input images by splitting them into 64-64-3-pixel patches, followed by

applying a frost filter to lower the image's background speckle noise. The images are then sent to a high-end, 23-layer CNN that has been trained with data from 80% of the patches in the dataset. The U-Net serves as the final component. Results obtained during the validation phase showed an accuracy of 92%, recall of 76%, precision of 84%, and dice accuracy of 80% when compared to similar work.

Fan et al., [12] developed a Feature Merge Network (FMNet)-based semantic segmentation model to improve marine oil spill accuracy. Based on the original picture's data, a threshold segmentation method was originally used to extract high-frequency information in the frequency domain. To improve the segmentation model's decision-making process, they first used CNN to capture the high-dimensional features embedded in a global feature, and then they integrated the high-dimensional features embedded in a segmentation model to complement the benefits. When compared to the U-Net, the proposed method's accuracy is 1.82% points higher at 61.90%. There was a 3% boost in accuracy, bringing the total percentage of correctly identified oil spill and oil-like areas to 56.33%. This paper proposes a new approach to marine oil spill monitoring that not only improves recognition accuracy but also addresses the overfitting issue that plagued the original model.

As a summary, Table 1.1 highlights the main points for each related study.

Table 1.1 Summary of Related Works

References	Dataset	DL (Included developed versions)	Metrics and Results	Advantages	Disadvantages	
Li et al., [17]	Yantai - Bohai bay	DL/CNN	Accuracy	The method has no restriction on the size of the input image	Small dataset.	
Krestenitis et al., [10]	SAR	DL/ DCNN	Accuracy, IoU	Make a collection of SAR images available online.	Backbone used with models has been fixed	
Yaohua and Xudong [18]	ERS-2 SAR	DL/DCNN	Accuracy	The model extracts multi-scale features.	Mixing of oil film and oil-like film is not considered in this experiment and accuracy of classification and recognition may be reduced in this case.	
Xiong and Zhou [15]	SAR	DL/CNN	Accuracy	Reduce the manual operation of feature extraction.	The sample was limited.	
Zeng and Wang [13]	AAMLP	DL/ DCNN	Accuracy, Recall, and Precision	A three-step processing framework, which briefly includes image segmentation, feature extraction, and target classification implemented latter two steps.	Most pay more attention to the pixel-level DL classification algorithms.	
Yekeen and Balogun [14]	SAR	DL/ Mask RCNN	Precision, Recall, and F1-measure	The model can learn from the shape and texture for localization and target detection of oil spill, lookalikes, and ship and land areas.	There is a need for the development of a large database for oil spill SAR imagery data this is with the view to further improve the accuracy of the model.	

References	Dataset	DL (Included developed versions)	Metrics and Results	Advantages	Disadvantages
Ghorbani and Behzadan [19]	An in-house image called Nafta	DL/CNN	Accuracy	The used dataset was created by web mining.	Fail to differentiate oil spill from the background when there is not enough contrast.
Shaban et al., [11]	SAR	DL/CNN	Accuracy, Recall, Precision and Dice	Provide a comparable improvement in precision and dice score compared to related work.	It cannot be applied to a problem with multiple classes.
Fan et al., [12]	SAR	DL/CNN	Accuracy	Enhances the original model's recognition accuracy.	Accuracy of recognition still needs to be improved.

After explained the previous studies in detail and summarized them in a table, it is clear that despite several studies interesting in oil spill issue most of them focused on solving the problem through classification, although the problem of oil spill needs to specify the exact location. Others reserchers designed models to predict the actual location of the occurrence of oil spill but did not focus on employing pre-trained backbone models to improve results. Moreover, some of them trained the model on small dataset and other study fixed the backbone type and the two point effect on enhancement of accuracy.

1.3 Problem Statement

CNN algorithms consider subfield of DL and it have been applied directly to classify oil spills, but they are unsatisfactory due to the unbalanced nature of pixels-wise classes, even when preprocessing is considered. The fact there is usually only one label assigned to the entire SAR image makes it difficult to manipulate complex scenarios or extract

additional information from the depicted content. Besides that, if the existing DL frameworks are designed for binary class problems, they do not apply to multi-class problems. Although Krestenitis et al., [10] proposed to use U-Net-Segmentation to get multi-class image classifications. But there was a limit in the types of Backbone users with U-Net segmentation.

i. In literature including Krestenitis [10], there is no criteria for selecting the right backbone.

ii. In Krestenitis's model [10], there is a segmentation problem, where the U-Net segmentation model is based on fixed backbone architecture.

iii. In SAR images [20], object detection is difficult due to the irregular shapes of oil spill on the ocean surface.

1.4 Research Objectives

Initially, this research aimed to develop an oil spill detection model that can detect all the classes in image to increase its performance. This aim can be reached even more with the following research objectives:

i. To investigate the best backbones for the U-Net segmentation model

ii. To enhance the U-Net segmentation model using the best backbone variants for SAR images

iii. To design a multi-model architecture with different backbones.

1.5 Organization of the Thesis

There are five chapters in the thesis. Each chapter starts with a short introduction that says what the chapter is about and what its main points are. The following is a summary of each chapter:

Chapter One gives a general overview of the research area. It shows the problems with this study and emphasizes how important the study is.

Chapter Two explains the theoretical side. It gives an overall look at the oil spill problem, DL models, and cloud computing. It gives lists of examples and diagrams to help the reader fully understand the subject of the thesis and how it works.

Chapter Three gives a general overview of the details of the methodology and research design that are used to reach the research goals. In addition, this chapter presents several different methods that, when combined, may be used to develop and build the suggested method.

In **Chapter Four**, simulations are used to show how the whole proposed method is evaluated. In this chapter, the method is looked at in different situations. This chapter also shows how the proposed method compares to the current solution from a theoretical and graphical point of view.

Chapter Five is the last. It talks about how the main research goals are met and what the main contributions of the thesis are. Based on what this study found, it also talks about some possible directions for future studies.

Chapter Two

Theoretical Background

2.1 Introduction

This chapter represents the theoretical part of thesis. In light of what has been discussed up to this point, the following is the structure of this chapter: the history of oil spill and their impact on the environment is discussed in Section 2.2. Section 2.3 goes into detail oil spill dataset, including its creation via SAR image, Drone and other technologies with advantages and limitations of them. The ways to preprocess dataset are presented in Section 2.4. The DL in AI that is related to this research scope is described in Section 2.5, including it is identification, advantages, backbone and techniques to detect oil spill. The cloud computing platform is discussed in Section 2.6. Finally, the chapter ends with Section 2.7, which goes into detail about the performance evaluation that has been used in this work.

2.2 Oil Spill Problem

Oil, a fossil fuel that has been around for a long time, is what drives the economy and provides people with heat and electricity. In contrast, when oil accidentally spill into the ocean, it can result in serious consequences. In addition to harming sea creatures, oil spill can ruin a day at the beach and make seafood unfit for consumption. To clean up the oil, measure the impact of pollution, and aid in the recovery of the ocean, sound scientific principles are required [21]. Taking a look at the history and the impact of the oil spill will provide a good foundation for understanding the scientific principles that will be used to prevent and detect the oil spill.

2.2.1 The History of Oil Spill

Although oil production, transportation, and consumption have all increased over the past decade, problem decreased during this time. Despite this, Human error is blamed for 30%-50% of oil spill, while equipment failure accounts for another 20%-40%. Consider the oil spill that have occurred in various parts of the world. several examples include the spill of 2,100,000-

2,400,000 barrels caused by the Canada Atlantic Empress, a fire that caused 1,850,000 barrels of oil in a ship tank in South Africa, combined effects of the 3,300,000-barrel spill in Mexico's Ixtoc field and the Persian Gulf War between Iran and Iraq (1,900,000 barrels) [22]. In April 2010, an estimated 4.9 million barrels of oil were spilled in the Gulf of Mexico, making it one of the largest oil spills in history. [23].

2.2.2 The Impact of Oil Spill

There are several sources of marine oil pollution, all activities related to petroleum, whether they be natural seepage, consumption, transportation, or extraction [24] as shoen in Figure 2.1.

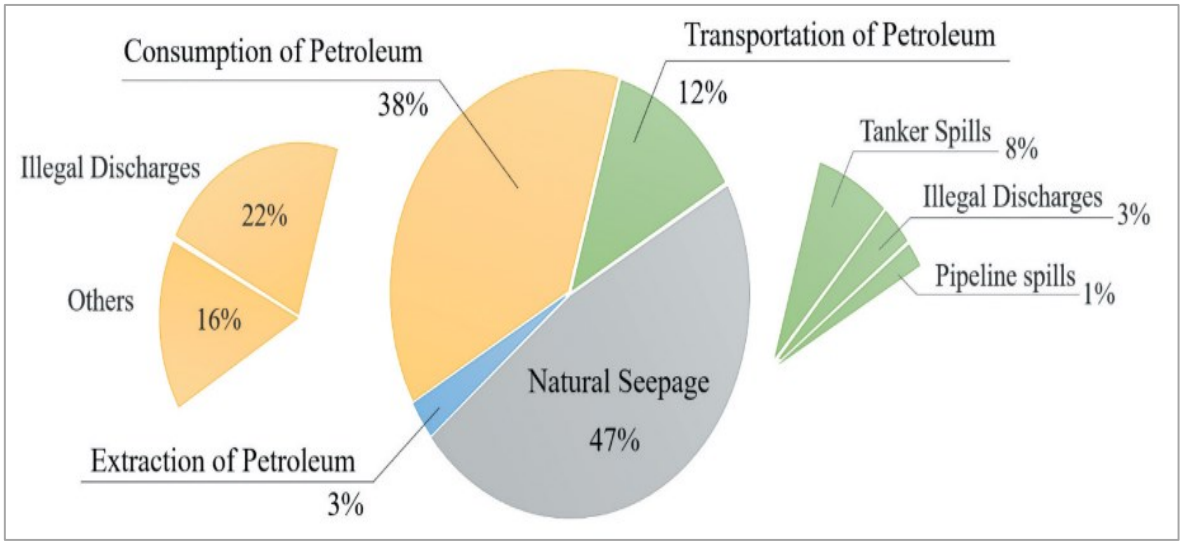


Figure 2.1 Major Inputs of Petroleum into the World [24].

More oil has spilled into the environment over the past years than has ever been stored in reservoirs. Oil can be easily broken down through either chemical oxidation or biological degradation because it is a naturally occurring substance. Oil thickness, light intensity, aeration, and nutrients are just some of the factors that affect the rate at which these processes take place. Tilling oiled shores exposes the oil to sunlight and air, and providing nutrients helps oil-degrading microorganisms flourish during the cleanup process. After a spill, if the substratum is adequately ventilated, oil degradation can typically continue

if the bulk of the oil is removed from the shore. After eight years, most of the sites affected by the Amoco Cadiz oil spill had returned to pre-spill conditions. [25].

There is no doubt that all sources of marine oil pollution affect marine life. The impact of oil spill on wildlife is probably the most public concern. Estimates of how many seabirds may have died are highly speculative. The search effort, the proximity of the shoreline to recorders, and the sea conditions at the time all have a significant impact on the accuracy of beach carcass counts. There is no correlation between the size of an oil spill and the number of seabirds killed by it. Over 35,000 dead seabirds were found after the Exxon Valdez oil spill. Despite the fact that the Braer spill (85,000 tonnes) was nearly twice as large as the Exxon Valdez spill (40,000 tonnes), only 1,500 dead birds were found. [3]. As shown in Table 2.1

Figure 2.2 Bird Deaths from the Exxon Valdez and the Braer Oil Spills [3]

Species group	Alaskan spill	Shetland spill
Sea ducks (eiders, etc.)	1440	167
Mergansers	121	1
Loons	395	14
Grebes	462	0
Heron	1	3
Geese/swans	9	0
Gulls	696	74
Kittiwakes	1225	133
Cormorants/shags	836	864
Shearwaters	3399	0
Fulmars	870	31
Guillemots/murres	20 562	220
Other auks	2174	29
Bald eagles	125	n/a
Other birds	3152	0
Total	35 467	1536

2.3 Oil Spill Dataset

There is a serious issue of oil spill affecting the marine ecosystem, which is causing concern on both a political and scientific level due to their serious impact on marine and coastal ecosystems. When evaluating seawater quality, pollution discharges and associated effects on the marine environment are important parameters to consider [26]. Research communities are facing the issue of the lack of publicly available datasets for such applications as one of the main challenges they have to deal with [27]. Several methods are commonly used to collect data such as the following:

2.3.1 Synthetic Aperture Radars

Microwave sensors are frequently used for oil spill detection and monitoring because of their wide field of view and ability to collect data at any time of day or night, regardless of weather. Because of this, they are a widely used remote sensing system for keeping tabs on oil spill. It can be detected and monitored with radar imaging technology, specifically Synthetic Aperture Radars (SAR) and Side-Looking Airborne Radars (SLAR). Both SAR (satellite-based) and SLAR (airborne) utilize backscattered radio waves reflected from a target to create two-dimensional images of the environment [28].

In the early 1950s, the SAR principle was discovered. After that, rapid development has been taking place all over the world, and a few airborne and spaceborne systems are now operational all over the world. In recent years, technological advancements and digital signal processing have resulted in flexible systems that can be used in a variety of different environments, both military and civilian, for a variety of applications. The capability of radar to see through clouds and rain has made it a valuable tool for detecting objects [29].

Despite deep learning's impressive track record in oil spill detection, there are some important caveats to keep in mind. Unfortunately, there are only a

small number of publicly available SAR oil spill datasets, and DL training requires massive datasets. Second, the shapes and boundaries of oil films that form on the ocean's surface are highly irregular and complex [20] as shown in Figure 2.2 below.

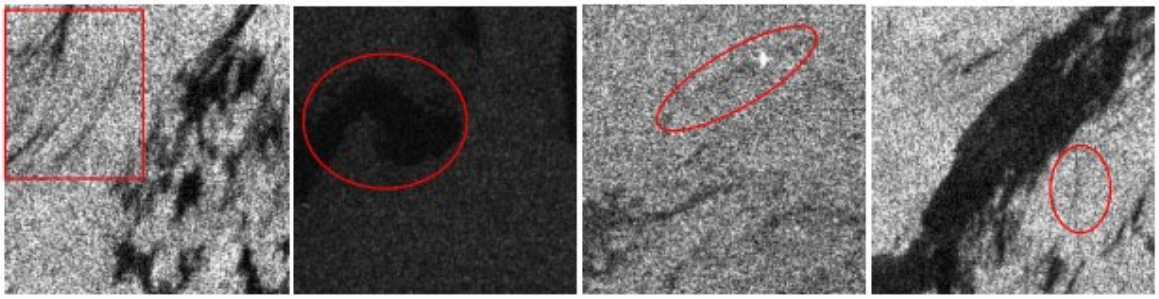


Figure 2.3 Images from Four Separate SAR Oil Spills [28].

2.3.2 Drone Images

The term "drone" can refer to any vehicle that operates autonomously or remotely, whether it operates on land, sea, or in the air. Drones come in as many varieties as there are different types of terrain, with land drones, sea drones, underwater drones, and underground drones all existing side by side. Drone aircraft, also called Unmanned Aerial Vehicles (UAVs) or Unmanned Aircraft Systems (UAS), are increasingly useful for environmental monitoring and oil spill response [30]. UAVs, also known as drones, are now being put to a wide range of practical uses. Their applications are diverse, ranging from basic freight transportation to emergency situations like fires, floods, and tsunamis. [31]. Figure 2.3 illustratize type of drone device.

Figure 2.4 MQ-4C 'Triton' Large UAV [32].

Drone aircraft can provide information quickly and economically in remote areas, filling an important surveillance gap. For oil spill response, there are a wide range of UAS that can be matched according to operational needs, forming a hierarchy of observational scales. It is possible for UAS to fly at low altitudes below clouds in order to ensure the best possible cloud effects when imaging and evaluating targeted objects. In this way, drones can overcome some of the limitations of other methods of aerial and in-situ observation. Therefore, they are suitable for responding to, detecting, and confirming potential oil spills. Satellite observations are limited by factors such as sensors' spatial and spectral resolutions, atmospheric conditions, revisit times, and costs [33].

2.3.3 Advantages and Limitations of SAR and Drone

After highlighting some of the techniques used to compile images of oil spills, we can be summarized in a Table 2.2.

Table 2.2 Summary of Advantages and Limitations of SAR and Drone

Dataset type	Advantage	Limitation
SAR	- the capability to collect data day and night under all weather conditions. - cover a wide area	- cost device
Drone	- in areas with limited access, drone aircraft can provide information rapidly and economically	- cover a limited area

2.4 Deep Learning Algorithms

A Deep Learning (DL) algorithm is a method of analyzing data by modelling complex abstractions with a multiple-level architecture which is usually based on neural networks. By using these techniques, the goal is to

develop 'true' artificial intelligence in which machines can perform complex tasks in a similar way to humans. It is very computationally intensive to build and train these systems, but GPU-based implementations have increased deep learning's popularity and success. CNNs and recurrent neural networks are the two most common approaches to DL [34]. The following section explains the most common DL algorithms:

2.4.1 Recurrent Neural Network

Recurrent Neural Network (RNNs) are a well-known and widely-used algorithm in the field of DL [35]. The primary applications of RNN are in the fields of speech processing and NLP [36]. In contrast to traditional networks, the RNN processes data in sequential order. The data's embedded structure allows for the encoding of valuable information, and this property can be put to use in a variety of contexts. Determining the meaning of a word in a sentence, for instance, requires an understanding of the sentence as a whole. Since x represents the input layer, y represents the output layer, and s represents the hidden state layer, the RNN can be thought of as a short-term memory unit. . Figure 2.4 shows an RNN diagram unfolded to fit a specific input sequence.

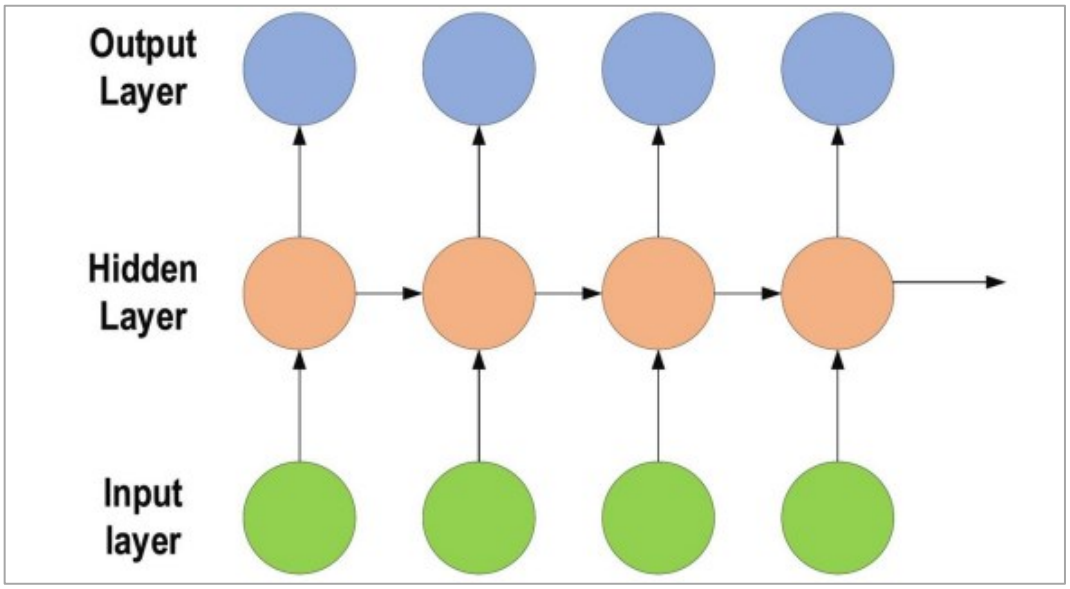


Figure 2.5 Typical unfolded RNN Diagram [35]

According to Pascanu et al., [37] All deep RNN methods can be broken down into one of three categories: hidden-to-hidden, hidden-to-output, or input-to-hidden. The learning difficulty is decreased and the benefits of a deeper RNN are made available through the use of these three methods, which introduce a deep RNN.

2.4.2 Convolutional Neural Network

A Convolutional Neural Network (CNN) is the most popular DL algorithm. [38]. In comparison to its predecessors, CNN has the advantage of automatically identifying the relevant features without human intervention [39]. In addition to its original purpose in computer vision, CNN has found widespread use in other areas such as speech processing, face recognition, etc. CNNs, like traditional neural networks, are inspired by the neurons found in the brains of humans and other animals. The CNN models the intricate network of neurons found in a cat's visual cortex [40].

Goodfellow et al., [41] the three main benefits of CNNs were identified as equivalent representations, sparse interactions, and shared parameters. To make the most of 2D input data structures like image signals, CNNs use shared weights and local connections instead of the standard FC. This procedure streamlines the network's training and improves its efficiency by only requiring a few parameters. This is exactly what happens in the brain's visual cortex. These cells can only sense local changes, rather than the entire environment. Similar to the MLP, CNN is made up of many convolution layers, then subsampling (pooling) layers, and finally FC layers [42]. The CNN architecture for image classification is illustrated in the Figure 2.5 below.

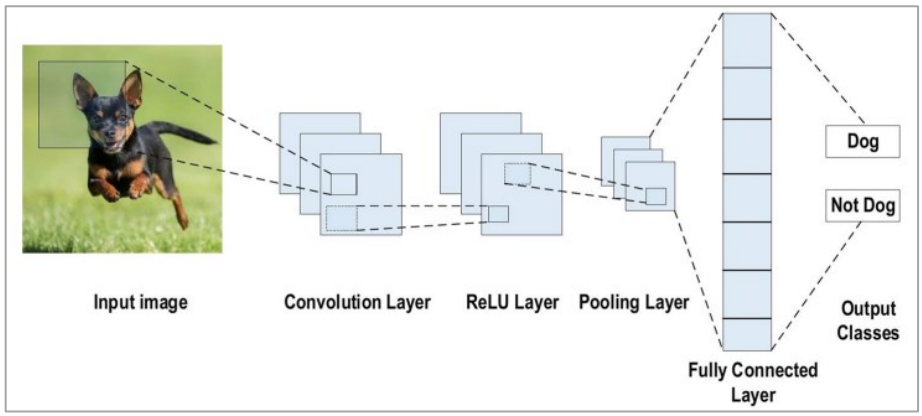


Figure 2.6 An example of CNN Architecture for Image Classification [42]

Multiple layers make up the CNN architecture (or so-called multi-building blocks). What each layer of the CNN architecture does:

- **Convolutional Layer:** The convolutional layer is the most crucial part of a CNN's architecture. A set of convolutional filters makes up the whole thing (so-called kernels). These filters are convolved with the input image (represented as N-dimensional metrics) to produce a feature map [42].
- **Pooling Layer:** The pooling layer's primary responsibility is to perform subsampling on the feature maps. Following the convolutional operations leads to the creation of these maps. In other words, this method reduces large-scale feature maps to finer-grained versions. At the same time, it keeps the vast majority of the most important data (or features) throughout the whole pooling process [42].
- **Fully Connected Layer:** This layer is typically the last one in a CNN's structure. This is the "Fully Connected" (FC) layer, where each neuron communicates with every other neuron in the preceding layer. It's what the CNN classifier uses [42].

2.4.3 Classification vs Segmentation

Classification of images involves the organization of the image into a specific category of information, and the image is described using a previously determined category or instance ID. There is no doubt that this is the easiest and most basic task of image understanding, but it is also the first breakthrough that the deep learning model has to achieve to achieve wide-scale application. To effectively classify images, it is not sufficient to simply classify them. Classification is the foundation of computer vision. The detection of key points, object localization, object recognition, semantic segmentation, instance segmentation, and semantic segmentation of objects are more difficult and meaningful tasks. The classification task is concerned with the whole picture, given the content description of the entire picture, whereas the detection task focuses on a specific object target, and it is necessary to obtain information on both the target's category and location simultaneously. In contrast to classification, target detection allows you to determine the foreground and background of an image [43] as shown in the Figure 2.6.

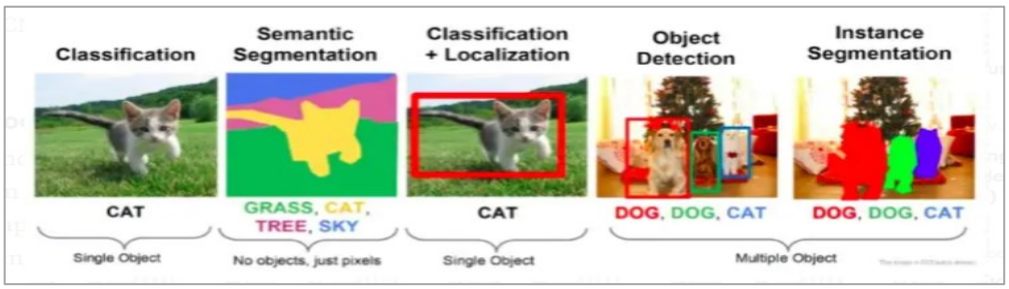


Figure 2.7 Types of Object Detection [44]

2.5 U-Net Semantic Segmentation Technique

Many visual understanding systems depend on segmentation. It includes medical image analysis (e.g., extraction of tumour boundaries and measurement of tissue volume), autonomous vehicles (e.g., detecting pedestrians and navigable surfaces), video surveillance, and augmented reality [45]. This technique is used to understand geometric shape, spatial texture, as

well as colour and grey correlation features. Furthermore, it is capable of segmenting semantic objects and extracting related features. Image segmentation is concerned with locating the target and extracting features of the regions of interest within the image [46]. Several models have been proposed for segmentation [47]:

- Fully Convolutional Network.
- Mask R-CNN (region-based convolutional neural network).
- Encoder-Decoder-Baseded Models.
- DeepLab.

The U-Net is one of the Encoder-Decoder-Based Models. It was developed to segment biomedical images as illustrated in Figure 2.7. The architecture of the system consists of three parts:

1. The contracting path or downsampling path.
2. Bottleneck.
3. The expanding path or upsampling path.

Using a contracting path with 3x3 convolutions allows for the capture of the image's context. The ratio of feature maps in a growing path improves while the total number of feature maps decreases. At last, a 1x1 convolution is applied to the feature maps, yielding a segmentation map that labels each pixel in the input image according to its attributes [47].

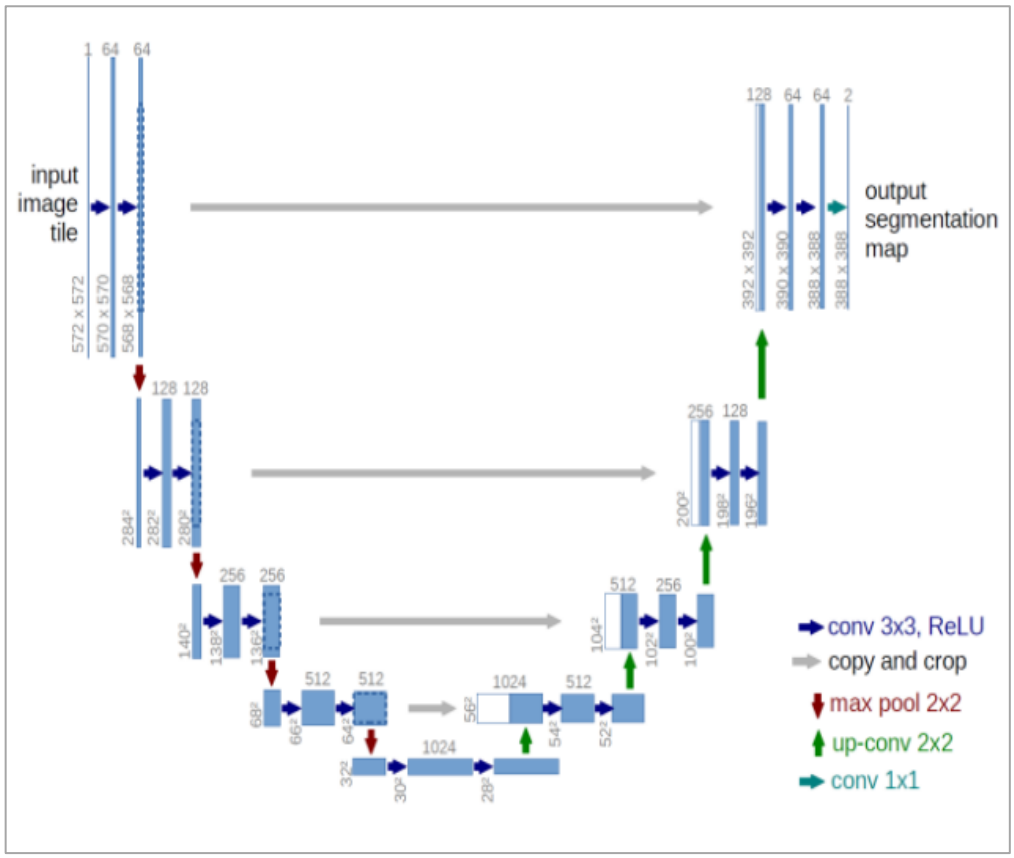


Figure 2.8 U-Net: Convolutional Networks [47]

2.5.1 U-Net Backbone

The backbone is a well-established architecture or network for feature extraction that has been proven effective in previous testing across a variety of other tasks (see, for example, VGGs, ResNets, and DenseNet). Here, we describe in depth the various backbones employed by DL models [48].

2.5.1.1 ResNet

ResNet, short for residual neural network, is a type of neural network that uses convolutional and pooling layer blocks with some skip-connections or recurrent units between them. Additionally, batch normalization occurs after the block [23]. ResNet, like the VGG family, has many variants, such as the 26M-parameter ResNet-34 and ResNet-50, the 44M-parameter ResNet-101, and the 152-layer ResNet-152. Both the ResNet-50 and ResNet-101 are popular

models for semantic segmentation and object detection. ResNet is also used for other deep learning architectures like Faster R-CNN [12] and R-FCN [25], etc. As illustrated in Figure 2.8

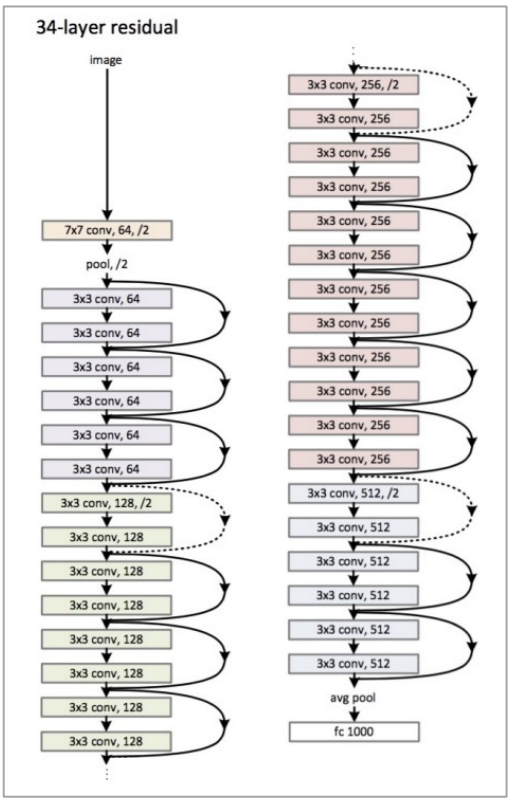


Figure 2.9 ResNet [48].

2.5.1.2 DenseNet

When it comes to CNNs, L = n, where n = the number of connections. Learning may be affected by the interlayer connection. For that, the authors in [18] DenseNet, a novel convolutional neural network architecture with L(L+1)2 connections, was introduced. All previous layers' outputs (feature maps) are fed into the current layer. The network could function with relatively shallow output channel depths (for example, 12 filters per layer), thus requiring fewer parameters. Each convolutional layer's filter count begins at an initial value, and subsequent layers use a "growth rate" of L to increase their filter count. This makes L a determinant in the total number of parameters. DenseNet is a popular deep neural network architecture, and many variants have been

proposed with varying numbers of layers. Network input images are 224 by 224 pixels. As illustrated in Figure 2.9

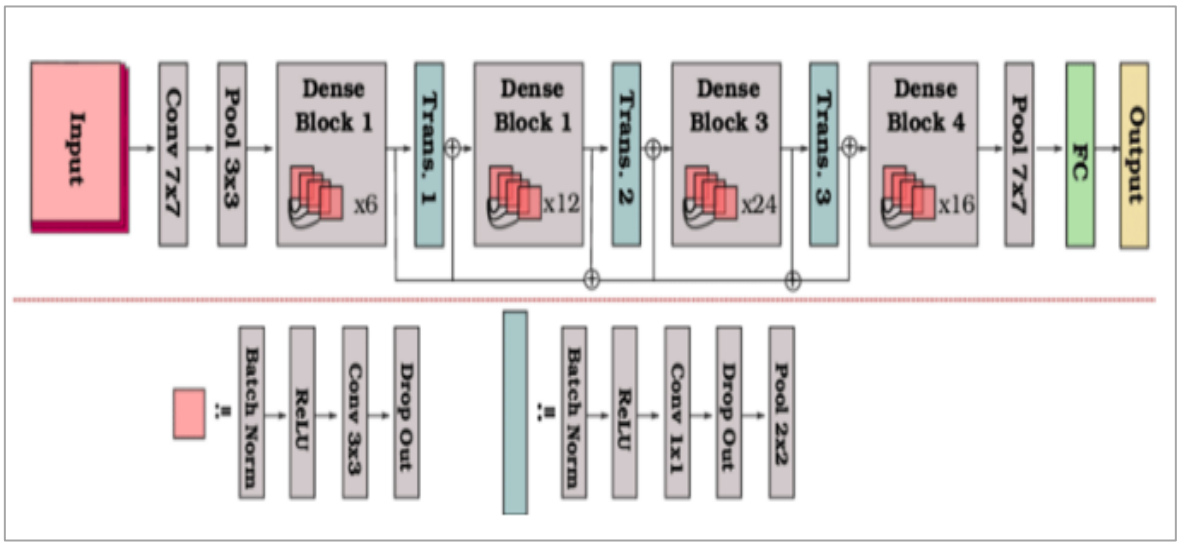


Figure 2.10 DenseNet [48].

2.5.1.3 EfficientNet

EfficientNet [44] networks, a relatively new family of architectures, are superior to older networks in classification tasks despite having fewer parameters and fewer floating-point operations per second (FLOPS). To efficiently scale the network's width, depth, and resolution uniformly, it makes use of compound scaling. When compared to state-of-the-art networks, EfficientNet's parameters are 8.4x smaller and inference time is 6.1x faster. There are a wide variety of EfficientNet releases between B0 and B7. Depending on the available resources and the computational cost, this can be easily swapped out for one of the other EfficientNet models. The original version, EfficientNet-B0, has only 5.3 million parameters, while the most recent version, EfficientNet-B7, has 66M. As illustrated in Figure 2.11

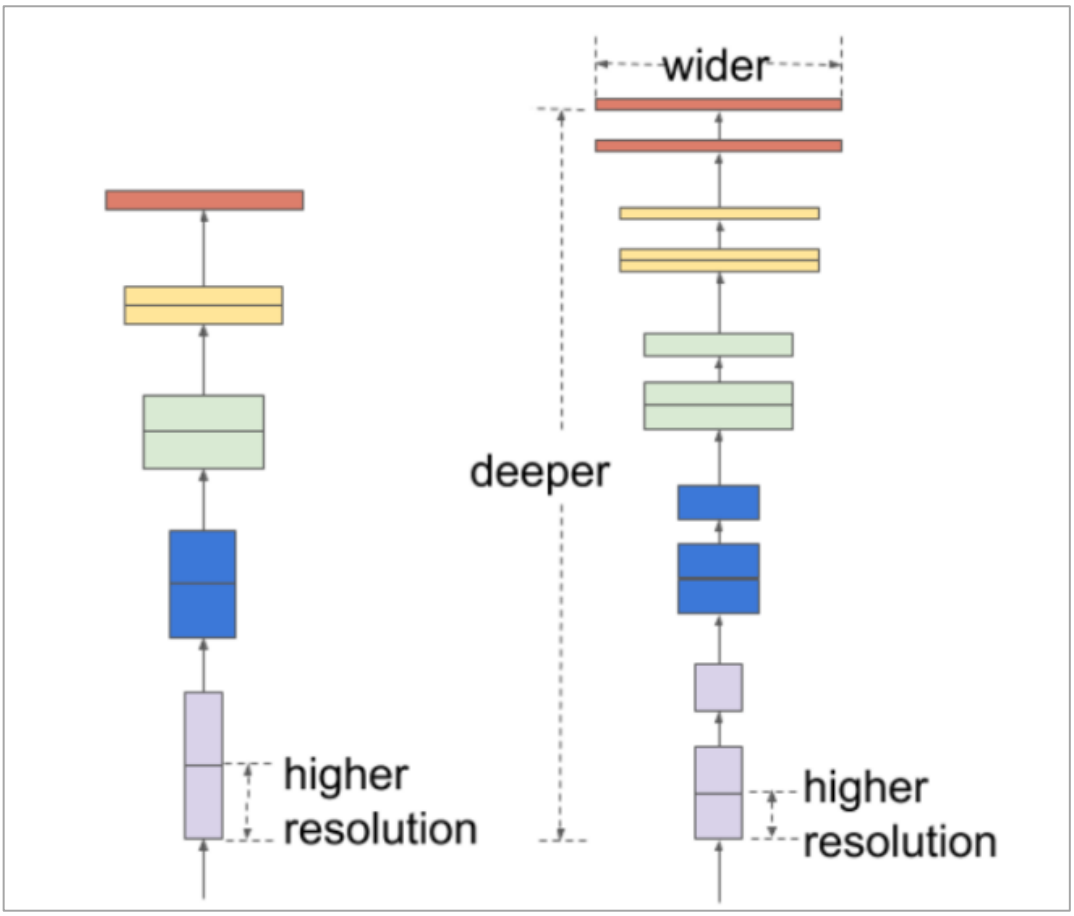


Figure 2.11 EfficientNet [48].

2.5.1.4 BN-Inception and Inception-V3

The biggest obstacle for any deep learning model is the computational expense. Training takes a long time due to the need to adjust parameters between layers, initialize the network, and choose appropriate learning rates. The researchers attempted a solution by standardizing the network's input layers across the board. This process, known as Batch Normalization (BN), reduces the effect of initializing parameters and enables the use of faster learning rates [19]. For the purpose of image classification, they apply BN to the Inception network (BN-Inception6) and run tests on ImageNet [19]. The obtained results are comparable to those of Inception on the same dataset, but they require significantly less processing time.

2.5.2 Transform Learning

By using transfer learning, knowledge can be transferred from related sources and domains to related domains. In the field of computer vision, some researchers have attempted to overcome the shortage of training samples for certain categories by adapting classifiers that were previously trained for others. Other methods aim to cope with different data distributions in the source and target domains for the same categories, such as lighting, background and viewpoint variations. It is possible to adapt classifiers or kernels based on standard features of images by using these and other related methods [49]. In most situations, transfer learning is a useful method of improving network performance directly, specifically when there is a limited amount of training data to work with. Using ImageNet pre-trained weights to initialize encoders has shown promising results in the semantic segmentation field [50].

2.5.3 Activation Functions

A neuron's activation function is developed based on the study of a biological neuron. An activation potential is a level at which the neuron becomes active. It usually keeps the output within a limited range. As activation functions introduce nonlinearity into neural networks, neural network models can learn complex operations by inserting them into neural networks [51].

1. **Sigmoid Function**: A sigmoid function is a nonlinear activation function that is used in neural networks to feedforward information [52].

2. **ReLU Function**: this is a linear activation function. Consider the most common activation function used with deep learning. It is fast-learning and easy to optimize [52].

3. **Softmax Function:** An activation function in neural networks is usually used as an output layer. The output of this activation function is a probability distribution value from a vector of real numbers. It has an output range of 0 to 1, and the sum of its output probability values equals one [53].

4. **Tanh Function**: In recurrent neural networks, the activation function is commonly used to enhance speech recognition and natural language processing. Its output ranges from -1 to 1 [54].

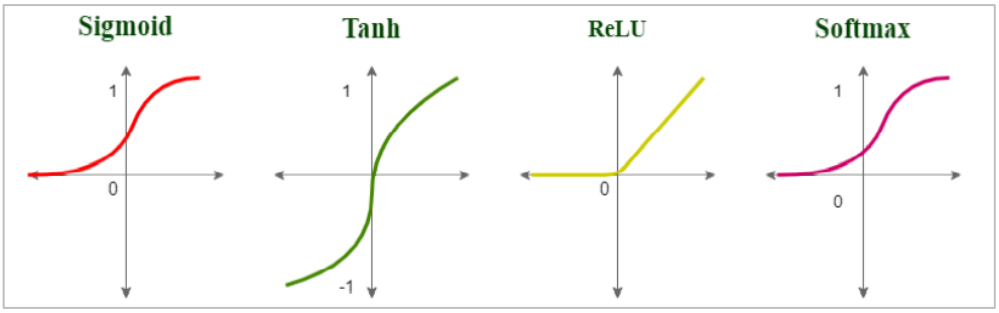


Figure 2.12 Activation Functions [54]

2.5.3.1 Loss Functions (Cost Function)

A loss function (also called a cost function) calculates the difference between a neural network's predictions and its actual output (training label data). Several methods are used to calculate the Loss function. The most common are:

1- Mean Squared Error Loss (MSE).
2- Mean Absolute Error Loss(MAE).
3- Binary Cross-Entropy.
4- Multi-class Cross-Entropy Loss.
5- Dice Loss [55].

2.5.4 Optimizers

To obtain an ideal model (high accuracy), the loss function must have a minimal rate of variation between the predictor results and the actual values, as explained earlier. During backpropagation, optimizers modify the parameters of the deep learning algorithm (weights and biases) after many iterations to minimize the loss of function values. Initially, the weight and bias values suggest zero (or one). In the next iteration, the optimizer will assume whether every parameter will increase or decrease. Several optimization algorithms are used in deep learning. The following subsections discuss some of them:

1- Gradient Descent (GD).
2- Stochastic Gradient Descent (SGD) with momentum.
3- Adagrad (Adaptive Gradient Algorithm).
4- Adadelta.
5- Adam optimizer [55].

2.6 Technologies for Image Preprocessing

To analyze the data, the DL algorithm requires input. In the same way as human vision, several analysis applications take into consideration the images in analyzing the data. In the same manner, numerical data, voice, a digital signal, and sensor outputs are considered to be inputs for deep learning algorithms that use neural networks. It can be seen in Figure 2.13 that there are different ways to classify input data according to the applications used in deep learning which requires an additional preprocessing step to remove any noise from the inputs. As a result, the accuracy of the algorithm is increased. In addition to noise removal, the preprocessing step is additionally utilized for marking the area of interest, so that the work can be concentrated on a particular region without any disturbances [56].

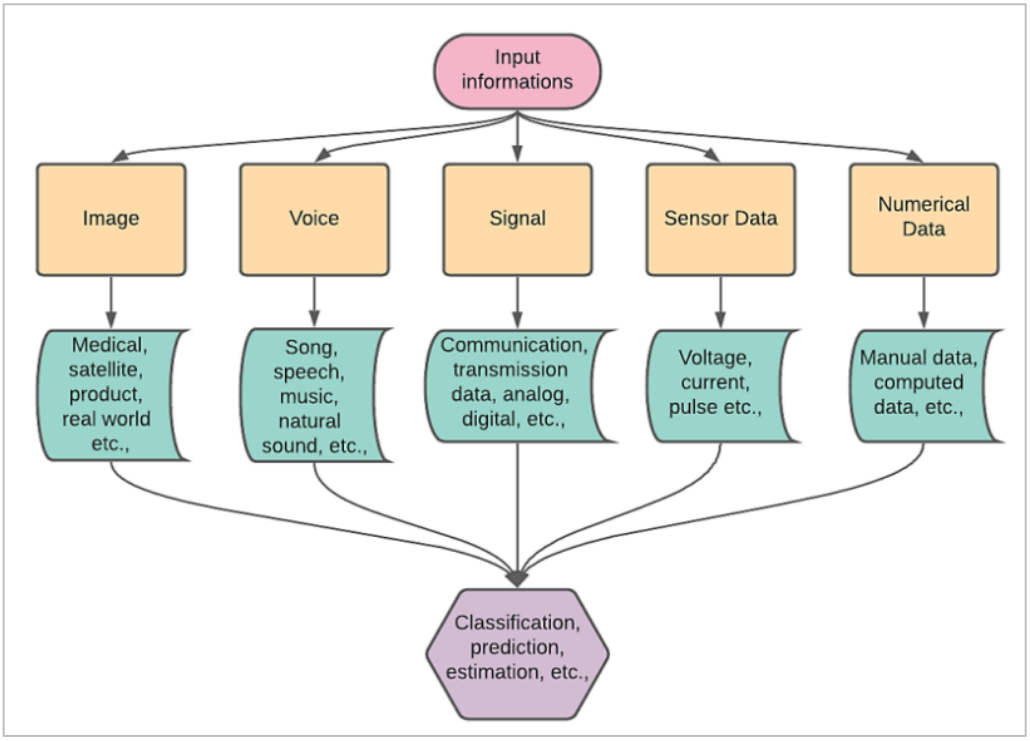


Figure 2.13 Application and Outcome of Deep Learning Algorithms [56]

2.6.1 Data Augmentation

DL models require a large amount of training data and have been shown to perform better when the training data is larger [57]. Data augmentation

involves adding new images to datasets that have fewer numbers in them in order to increase the number of images that can be used to train a neural network. During this process, the data are constrained to prevent an asymmetric representation and to prevent overfitting complications. Deep learning models can be made more robust by using appropriate data augmentation techniques [58].

A variety of techniques can be implemented to augment the original data such as zooming, flipping, shifting, rotating and adding noise to the original image to augment the original data:

- **Horizontal and Vertical Flip:** There are both vertical flipping and horizontal flipping methods which are very effective and popular when it comes to enhancing data. It has been proven that augmented data can be one of the simplest applications to handle in terms of executing its operations, which is a real benefit for users of the dataset. Furthermore, the flipping of the horizontal axis is considered to be more common than the flipping of the vertical axis, as flipping the horizontal axis is more convenient for the majority of the projects, owing to the possibility of the existence of images that are flipped horizontally being higher than the vertical axis flipping [59].
- **Rotation:** Is another method of data augmentation, and it is considered to be one of the most efficient methods of data augmentation. A rotation is performed by rotating the right or left image on an axis between (1 and 359) to change its orientation. Depending on the degree of rotation, it is possible to adjust the maximal angle of rotation. It works especially well whenever it is slight because the network must be able to recognize the object regardless of which direction it is appearing in the image. Certain applications may have problems if the image is rotated precisely in this way. It has been observed that as the rotation angle increases, the data

label is no longer retained after conversion, and background noise is introduced as a consequence. When the background noise of an image is too different from the other areas of the image, then the networks may learn incorrect features if the background noise is too diverse [34].

- **Brightness:**It is important to note that brightness is a method of enhancement that cannot easily be classified into the previous categories since it is neither a geometric nor a synthetic transformation. As the brightness of the images changes, the underlying structure of the images will also change. It is possible to alter the appearance of an image entirely by using these settings with extreme values, such as setting brightness so low that the image becomes black and the original contents can no longer be seen. Since it is easy to use and has a logical explanation, many authors have been able to use it when training deep-learning models, utilizing brightness in both a supervised and an unsupervised manner, which is due to its ease of use and logical explanation. Because brightness augmentation is based on representations, the self-supervised algorithm can have a more severe effect on classification [60].

2.7 Cloud Computing

Information and applications can be managed on-demand with Cloud Computing, a powerful technology. In this way, organizations can provide resources like software, applications and services to their customers without the need to build or maintain their in-house IT infrastructure. It is reliable and consistent. As with electricity bills, cloud computing resources are billed on a based-on-consumption basis, so cloud computing is a cost-saving technology for any organization or business. In cloud computing, you can access a shared tank of constructive computing resources, mainly servers, storage, and the Internet on demand. Cloud computing indeed has some advantages, but it also has several disadvantages, the most important of which is security. Cloud

computing suffers from many security problems such as data access, reliability, integrity, identity management, and many others [61].

Cloud computing is practical and accessible to end clients because various types of services and models work behind the scenes. There are generally two types of working models for distributed computing: Different types of administrations and models working behind the scenes make distributed computing possible and accessible to users. There are two distinct models of operation in distributed computing [62]: deployment models and service models.

2.7.1 Deployment Cloud Models

How does the cloud environment get found? Cloud divides accessibility into four categories: Private, Public, Hybrid, and Managed.

- **Public Cloud:** a cloud hosted by a specific service provider that offers the elasticity and accountability/utility of cloud computing to a single tenant (dedicated) or multiple tenants (shared) operating environment. In most cases, the off-site data center's physical infrastructure is owned and managed by the contracted service provider. Customers share infrastructure pools with minimal control over configurations, security, and availability. Since public clouds are potentially more expansive than private enterprise clouds, they can easily expand to meet growing demands.
- **Private Cloud:** includes the benefits and features of elasticity, accountability, and utility computing, but is instead a single-tenant (dedicated) operating environment provided by a company or its designated services. Private clouds, unlike public ones, give users more privacy and security options for their data. (Hosted internally, hosted remotely, and private clouds in the cloud) This private cloud model has the advantage of standardization and security but is not as scalable as

others. On-premise clouds are those that a business or individual sets up in its own data center. The IT department would have to spend money on hardware in addition to the usual operating and startup expenses. This is perfect for programs that need absolute command over their environment and security settings. When referring to a private cloud, this term refers to a cloud that is hosted off-site by a cloud provider.

- **Hybrid Clouds:** is combine the best features of both public and private clouds. The use of these clouds enables the sharing of data and, in some cases, the portability of applications between different cloud service providers and offerings, regardless of ownership or location, by employing either standard or proprietary methods. Hybrid clouds expand service providers' computing options by wholly or partially relying on external cloud service providers. Hybrid cloud models can offer elastic, externally provisioned scalability by combining the advantages of both public and private clouds.
- **Managed Clouds:** one advantage of managed clouds is that a specific service provider handles the cloud's scalability and accountability/utility model. Managed clouds can provide either a dedicated or shared platform for their users. The infrastructure is owned and/or located by the organization, but the designated service provider handles management and security [63].

Figure 2.14: Summary of the Various Features of Cloud Deployment Models [63]

Deployment Model	Managed By	Infrastructure Owned By	Infrastructure Located At	Accessible and Consumed By
Public	Third party provider	Third party provider	Off-premise	Untrusted
Private	Organization	Organization	On-premise Off-premise	Trusted
	Third party provider	Third party provider	On-premise Off-premise	
Managed	Third party provider	Third party provider	On-premise	Trusted or Untrusted
Hybrid	Both organization and third party provider	Both organization and third party provider	Both on-premise and off-premise	Trusted or Untrusted

2.7.2 Service Models

As illustrated in Figure 2.14, cloud services can be classified into three types:

- **Infrastructure as a Service (IaaS):** this service model is delivered by means of a computing infrastructure comprising servers (typically virtualized) with designated computing and storage resources. The user has complete authority over all deployed software, including operating systems, applications, and data storage, and some authority over network settings. For instance, Amazon provides the Elastic Compute Cloud (EC2) for the creation and management of virtual machines, and the Amazon Simple Storage Service (S3) for the storage and web-based retrieval of data.
- **Platform as a Service (PaaS):** is a method for building, deploying, and running software applications (PaaS). Applications must be built using the languages, libraries, services, and tools supported by the provider as part of the development platform offered as a service. Google App Engine supports Java and Python application development via a provided software development kit (SDK) and an Eclipse plugin, respectively.
- **Software as a Service (SaaS):** access to applications is provided to customers by the cloud provider's infrastructure. In the cloud, users are not responsible for managing the underlying system but may have limited control over certain aspects of individual applications and their associated settings. The applications have their own interface through which they can be accessed [64].

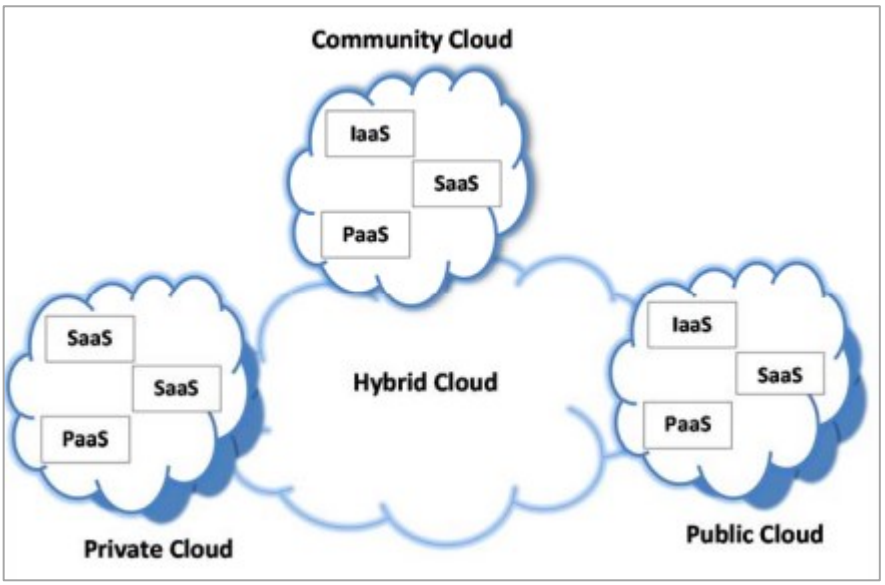


Figure 2.15 Cloud Deployment Models and Infrastructure [65].

2.8 Network Protocols

The network protocol determines how data is transmitted between devices within a network by establishing a set of rules. As a result, connected devices can communicate with each other regardless of their differences in internal processes, structures, or designs. In this thesis there are many protocols used:

2.8.1 Transmission Control Protocol/Internet Protocol (TCP/IP)

TCP/IP is a protocol that connects network devices over the internet. TCP/IP is also used in private computer networks (internet and extranets) as a communications technology. TCP/IP refers to the entire IP suite, which is a collection of rules and processes. IP and TCP are the most important protocols, but others are also included. Through TCP/IP, data can be shared over the internet by providing end-to-end communications that define how packets should be divided, addressed, sent, routed, and received. By automatically recovering from the loss of any network component, TCP/IP makes networks dependable by requiring little central control [66].

2.8.2 File Transfer Protocol (FTP)

By making it easy for users to move files between computers on a network, FTP was developed to allow the indirect use of computers on a network. In the same way, as most TCP/IP protocols, FTP uses a client/server model, which means that a user's machine creates a connection to an FTP server to send and receive files. With FTP, the user is shielded from implementation details of how files are transferred from one place to another, and the process of moving them is made simple. Therefore, FTP is designed to automatically resolve many of the problems that may arise from file formats that differ from file formats stored on different systems [67].

2.8.3 Remote Desktop Connection (RDP)

Maintaining staff can interact with the remote server system more easily through RDP's graphical interface. With the remote desktop client, we can connect any server that supports remote control and decrease the workload. As a result, operations become easier and workers' operations become more efficient. Users can use remote desktops to share their mouse, keyboard, and other devices with the remote server system, but they are not involved in any data processing, so the hardware requirements are usually low. Currently, RDP is the most popular graphical remote access protocol on the market. RDP was originally used in Windows Ins Interior and was not publicly disclosed details about themselves, and its implementation procedure has been kept secret from the public. RDP is a kind of terminal services network protocol that adopts the typical C/S architecture. Fortunately, the Ins has disclosed the details about it now. Clients run on local machines, while servers run on remote servers. Since the RDP protocol is widely used, it does not only work on Windows-based server machines but also Linux computers. Winconnect, Linrdp, and Rdesktop are some of the Linux clients that provide remote desktop control [68].

2.9 Performance Evaluation

The efficiency of different segmentation models is measured using several evaluation measures. When an image segmentation model is trained, it produces a prediction. To determine the effectiveness of the model, it is necessary to evaluate it. There are several evaluation measures commonly used for image segmentation tasks, and the following are just some of the ones that I would like to mention:

2.9.1 Intersection Over Union (IOU)

The IOU measures performance by calculating the intersection and union between the Prediction and Ground Truth values. The range of IoU is 0-1 (0 - 100%), where 0 signifies that there is no overlap and 1 signifies a perfectly overlapping segmentation [69].

$$IoU = J(A,B) = |A \cup B| / |A \cap B| \quad \text{..... (2.1)}$$

2.9.2 Pixel Accuracy

There is an equation that defines pixel accuracy by taking the ratio of correctly classified pixels to the total number of pixels. As shown in the following equation [69].

$$\text{Accuracy} = \text{Correct predictions} / \text{All predictions} \quad \text{..... (2.2)}$$

2.9.3 Precision

The precision can be defined as the ratio between the True Positives and the total number of Positives. As shown in the following equation [70].

$$\text{Precision} = TP_{\text{class A}} / (TP_{\text{class A}} + FP_{\text{class A}}) \quad \text{..... (2.3)}$$

2.9.4 Recall

The recall is defined as the number of True Positives that are correctly identified. The following is the equation for defining recall [70].

$$Recall = TP_{class\ A} / (TP_{class\ A} + FN_{class\ A}) \text{ …….. (2.4)}$$

2.9.5 Dice Coefficient (F1 Score)

Dice Coefficient (F1 Score) Simply put, the Dice Coefficient is 2 * the Area of Overlap divided by the total number of pixels in both images. The weighted average of Precision and Recall is known as the F1 score. The F1 score is useful when a dataset contains an uneven distribution of classes. The following is the equation for defining the F1-Score [45].

$$F1 = 2* ((Precision * Recall)/(Precision + Recall)) \text{ …… (2.5)}$$

- **True Positives (TP)** - It means correct values in actual and prediction for positive values which means actual value yes and predicate value yes.
- **True Negatives (TN)** - It means correct values in actual and prediction for negative values which means the actual value is no and the predicate value is no.
- **False Positives (FP)** - It means incorrect values in actual and prediction for positive values which means the actual value is no and the predicate value is yes.
- **False Negatives (FN)** - It means incorrect values in actual and prediction for positive values which means the actual value is no and the predicate value is yes [69].

Chapter Three

Research Methodology and Proposed Method

3.1 Introduction

In this chapter, all techniques used to build the proposed system for oil spill detection with SAR images will be described. The proposed system contains three phases. The first phase is to split the SAR images from 1250*650 to 512*512 instead of resizing them to keep the images' feature. As part of the second phase, an augmentation technique was used to extract deep features and to clarify the SAR image dataset as well. In the third phase, the U-Net segmentation model was deployed with a variety of backbones to segment the dataset images into five classes include (sea, oil spill, look-alike, ship and land). The best three models save as demo files and upload to VPS on the cloud. Within the VPS environment, the client can send the whole image without splitting it onto Patchify. When VPS receives the image, it applies the preparation process, divides it into pieces, inserts them in the model, and then collects and sends them back to the client. In this chapter, it will be explained in detail .As described above, 70% of the dataset was used to train the model and 20% to validate the model. As for the remaining dataset 10%, it was used to test the model after being fully trained.

3.2 Proposed System

Detecting oil spill based on SAR images is the main purpose of this thesis. A U-Net algorithm is used to process images and create mask prediction. In this thesis at the beginning, The U-Net algorithm has been implemented, and the results obtained were recorded. After that to improve the model's performance, the encoder part of the algorithm was replaced with a set of backbones. The pre-trained backbone refers to the convolutional layers used in the encoder portion of the U-Net segmentation model. These layers are typically pre-trained on a large dataset, such as ImageNet, and the weights of each layer are then used as the starting point for the U-Net segmentation model. In this thesis, we deploy (Densenet-201, Resnet101, EffectionNet-b3, and Inception V3) each of

them was tested individually with the U-Net, and the results were recorded. Then we chose the most effective three to be the models whose results will be accepted, and each of them is employed on its server so that users can examine images of the oil spill in the future. The overall methodology adopted in this thesis can be summarized (see Figure 3.1). It consists of train and test phases implemented in google colab and deploying models on the VPS cloud. Each VPS has public IP and deploys a unique model to process SAR images.

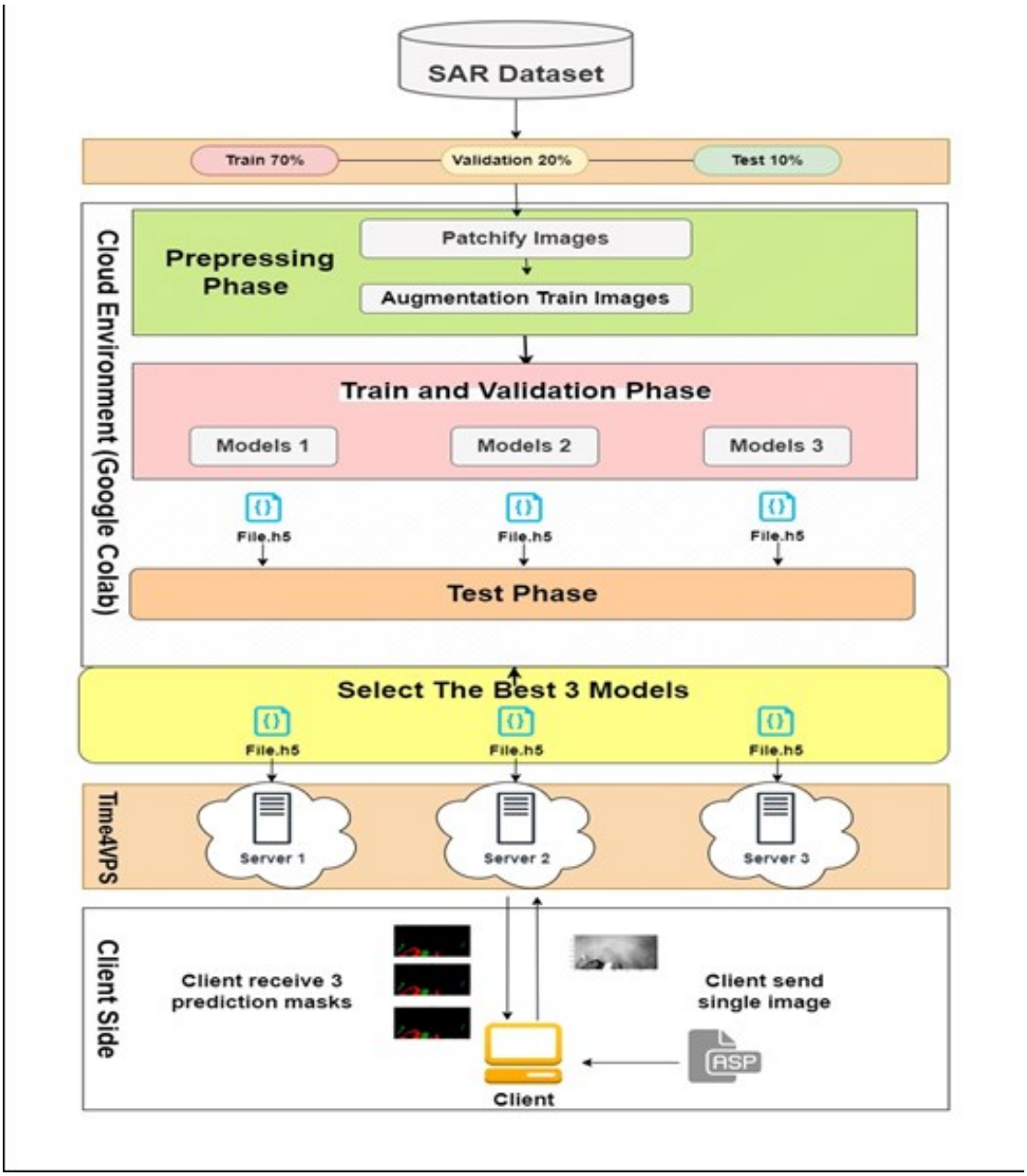


Figure 3.1: The Macro View for the Proposed System

To understand the train and test phases implemented in google colab with more details (see Figure 3.2) illustrate the micro view, which contains deeply all resource steps and it will present later.

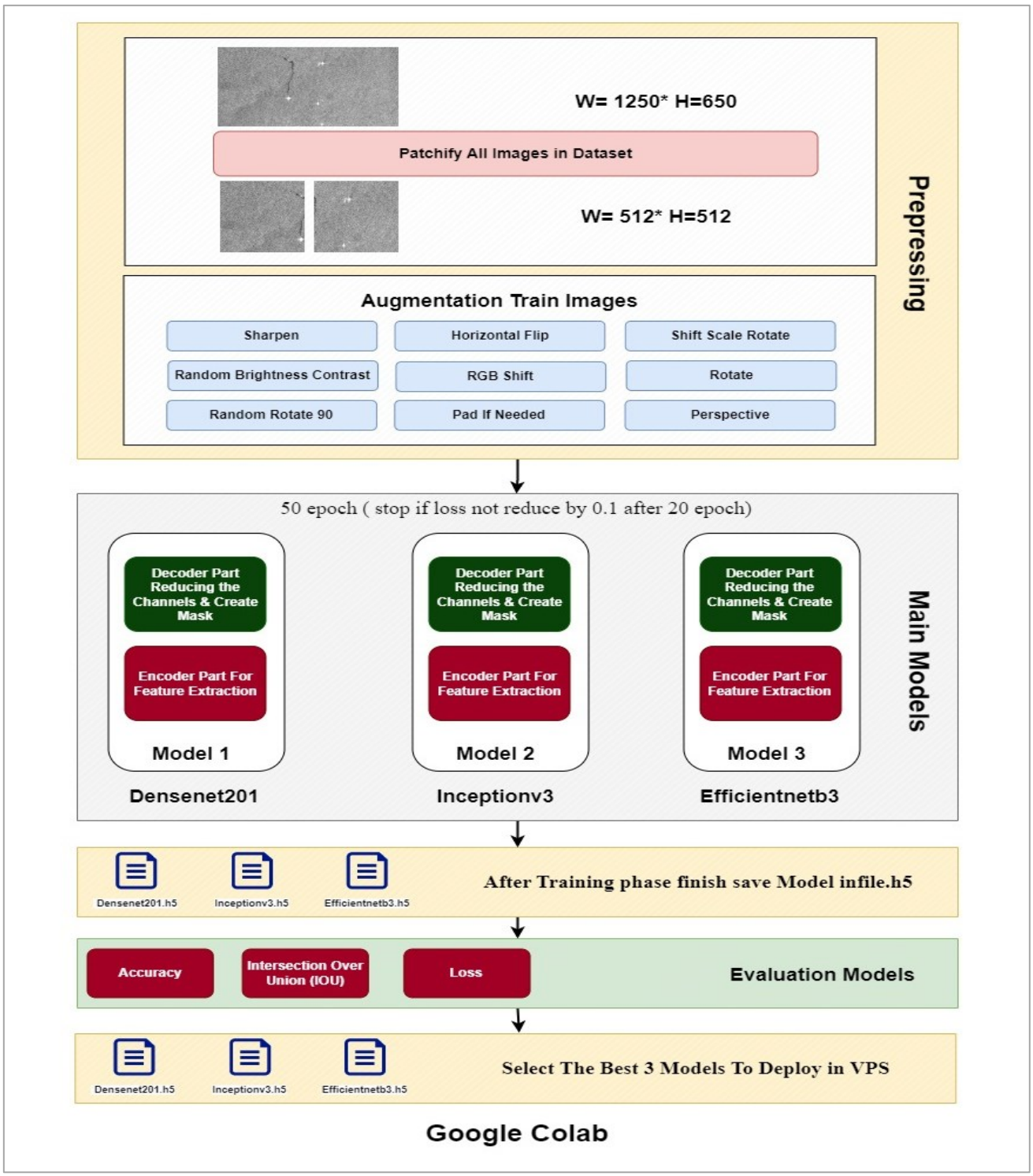


Figure 3.2: Micro View of the Proposed System

The steps in the figure above can be summarized as follows:

Step 1: Download the SAR image from M4D https://m4d.iti.gr/oil-spill-detection-dataset/

Step 2: Subdivide the dataset to (train, validate and test).

Step 3: Split images from (1250*650) to (512*512).

Step 4: Apply augmentation on training images.

Step 5: Performs semantic segmentation U-Net with a variety of backbones and selects the best three models (there are three models deployed in this thesis).

Step 6: save each model in the demo file (demo.h5).

Step 7: Evaluation models.

3.3 M4D Dataset

This thesis used the M4D dataset. EMSA has provided an oil spill dataset based on the CleanSeaNet service records, which covers eighteen months from 28/09/2015 up until 31/10/2017, covering oil spills that have taken place on the European Union's territory. There are 5 classes with specific colors for each one of them as shown in (Table 3.1) below:

Table 3.1 Colors for Dataset Classes

Class name	RGB Mask	RGB Values	1_D Labels	Color
Sea Surface	Black	(0,0,0)	0	
Oil Spill	Cyan	(0, 255, 255)	1	
Look_alike	Red	(255, 0, 0)	2	
Ship	Brown	(153, 76, 0)	3	
Land	Green	(0, 153, 0)	4	

As a result of the nature of the oil spill detection problem over SAR images, there is a high imbalance between the instances of different classes in the dataset. In particular, sample data of sea surfaces or land surfaces should dominate. In contrast, oil spills and look-alikes are generally spread over smaller regions of SAR images, whereas look-alikes usually cover a larger area due to the nature of natural phenomena such as low wind speeds and wave shadows close to the ground. It is expected that the "ship" class samples will be enumerated at a significantly lower number because their presence is not guaranteed on the scene of an oil pollution incident.

3.4 Preprocessing Phase

Pre-processing is a series of fundamental steps in the oil spill detection system that initializes the data for the next stage (the semantic segmentation stage) and clarifies image features to make the model more robust. Two steps are involved: splitting all the images and augmentation of only the train dataset images. The section below explaintion each one in detail:

3.4.1 Split Image

Segmentation algorithms differ from classification algorithms in that they require two inputs instead of one. This is a serious challenge when preparing the data. It is necessary to apply the same operations to the image and its mask, otherwise, a defect will occur during the model training. Since the size of the original image is not compatible with the U-Net algorithm, the size of both the image and its mask must be changed. To accomplish this, either the overall image size is adjusted or the image is fragmented into small enough pieces so that the U-Net algorithm can work with them.

SAR images are originally 1250 * 650, which is too large for the process, It is necessary to resize or split images with their masks in order to be able to use them. The split technique used in this thesis is shown in the steps below:

1. Input SAR image (1250 * 650).
2. Read the path in a directory.
3. List all images.
4. Read each image.
5. Returns image in RGB format.
6. Crop image.
7. Save in array.
8. Save each patch in a directory.
9. Patch SAR image (512 * 512).

3.4.2 Augmentation

In DL and computer vision, image augmentation is used to increase the amount of data available for training models. It can also improve model accuracy. With Albumentations, you can augment images in a variety of ways, including random cropping, random scaling, rotation, hue/saturation/value adjustments, etc. Additionally, the library includes visualization tools to help developers understand the results of the augmentation techniques. It also provides an easy-to-use API that makes it easy for developers to apply these techniques. There are nine kinds of augmentation techniques will be employed in this thesis on training images to reduce overfitting exposures and enhance the efficiency of the proposed system for precise assessment commonly used augmentation techniques have been observed to influence the original dataset. These techniques include see (Table 3.2).

Table 3.2 Augmentation Techniques

Augmentation Techniques	Purpose
Sharpen	Sharp the image pixel
Horizontal Flip	Flips both rows and columns of such a matrix horizontally.
Shift Scale Rotate	Randomly apply affine transformations, such as shifting, scaling, and rotating.
Random Brightness Contrast	Change the brightness of the image
RGB Shift	Change RGB image value
Rotate	Rotate image randomly
RandomRotate90	Rotate an image 90 degrees
Pad If Needed	Add padding (border) to image
Perspective	Change image viewpoints

After the data is prepared and stored into three categories (train, validation and test), the U-Net algorithm will be implemented independently, and then the U-Net algorithm will be applied with different backbones, as will be explained in the next sections.

3.5 U-Net Semantic Segmentation Model

The U-Net algorithm which is used for semantic segmentation depends on pixel image and represent all the pixel belonging to a class with the same color. It consists of a pair of paths, encoder and decoder, linked by a skip connection. The encoder part of the U-Net consists of a series of convolutional and max pooling layers, which extract features from the input image (512* 512 * 3).

Then, the decoder takes these features and up scales they back to the original image size. The skip connection allows the decoder to incorporate information from the encoder, which results in a more accurate segmentation. In the middle there is a part of the U-Net algorithm called the bottleneck that is responsible for connecting the encoder and decoder. This represents the middle point of the algorithm where the maximum depth of the image can be calculated. The maximum depth get from this model is 512. Table (3.3) below show the U-Net details according to SAR dataset images.

Table 3.3 U-Net structure (Layer Type, Output Sizes, The Parameter, Connect to Which Layer)

Layer (Type)	Output Shape	Param #	Connected To
Inputlayer	[(None, 512, 512,3)]	0	[]
Encoder_Bloak_1	[(None, 256, 256,64)]	10336	Input_1
Encoder_Bloak_2	[(None, 256, 256,128)] [(None, 128, 128,128)]	55808	Encoder_Bloak_1
Encoder_Bloak_3	[(None, 128, 128,256)] [(None, 64, 64,256)]	222208	Encoder_Bloak_2
Encoder_Bloak_4	[(None, 64, 64,512)] [(None, 32, 32,512)]	886784	Encoder_Bloak_3
Encoder_Bloak_5	[(None, 32, 32,512)] [(None, 16, 16,512)]	3543040	Encoder_Bloak_4
Conv_Block_5 (Bottleneck)	[(None, 32, 32,512)]	14256304	Encoder_Bloak_5
Decoder_Block_1	[(None, 32, 32,1024)]	9187840	Conv_Block_5 Encoder_Bloak_5
Decoder_Block_2	[(None, 64, 64,768)]	2296064	Decoder_Block Encoder_Bloak_4
Decoder_Block_3	[(None, 128, 128,384)]	574592	Decoder_Block_1 Encoder_Bloak_3
Decoder_Block_4	[(None, 256, 256,192)]	143936	Decoder_Block_2 Encoder_Bloak_2
Decoder_Block_5	[(None, 512, 512,32)]	36128	Decoder_Block_3 Encoder_Bloak_1

Conv2D	[(None, 512, 512,5)]	165	Decoder_Block_4
Softmax	[(None, 512, 512,5)]	0	Conv2D
Total Params: 31,213,205			
Trainable Params: 31,201,157			
Non-Trainable Params: 12,048			

One of the most important features in U-Net is concatenation between two parts that make U-Net able to extract (high, medium, low) image features. To combine a downward layer with an upward layer, the height and width must match (see Figure 3.3).

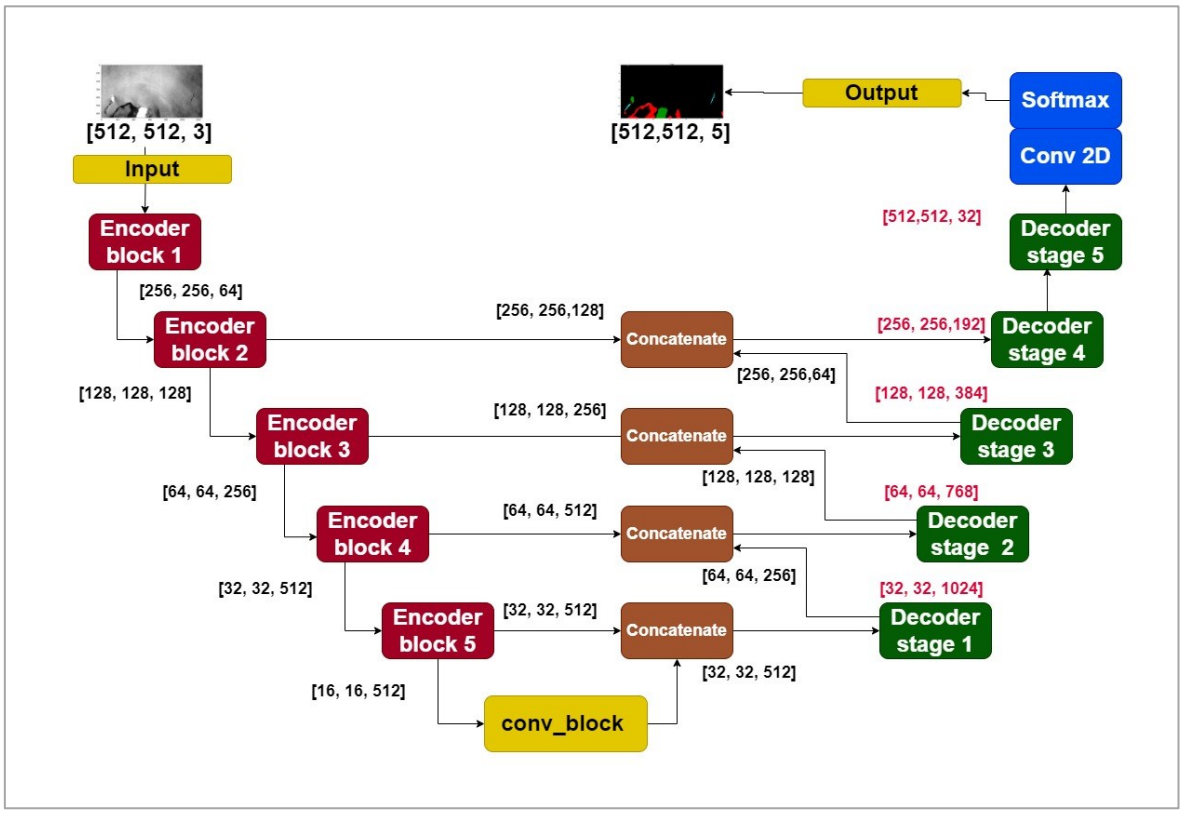


Figure 3.3 U-Net Architecture

There are four concatenate blocks in the U-Net algorithm to link the outputs of the encoder directly connected to the decoder to ensure that they are not lost

in the downsampling operations. In addition, mitigating the issue of vanishing gradients.

3.5.1 Encoder Part

It is essential to take note that the encoder is the first half of the architecture and acts as the feature extractor. It is usually a pre-trained classification network, such as the DenseNet /ResNet. As the encoder network in order to extract features from an input image, it first learns an abstract representation of the image using a series of encoder blocks.

The steps below explain encoder block:

1. Input: image (512*512*3)
2. There are two 3x3 convolutions in each encoder block.
3. After every convolution, an activation function known as a Rectified Linear Unit (ReLU) is applied.
4. 2x2 max pooling layer with stride 2.
5. Output: Input Bottleneck (16, 16,512)

3.5.2 Decoder Part

The Decoder block represents the second half of the U-Net model. In the beginning, a transpose convolution of 2x2 is performed. Afterwards, it is concatenated with the skip connection feature map from the encoder block, which corresponds to the skip connection feature map. There are also skip connections between layers in the network that enable features from earlier layers to be retrieved that can be lost due to the depth of the network.

The Steps below explain decoder block:

1. Input: Output Bottleneck (16, 16,512)
2. Each decoder block consists of 2x2 convolutions to half the channel features.
3. Concatenation from the contracting path (Encoder).

4. Each decoding unit is made up of a pair of 3x3 convolutions.
5. After every convolution, an activation function known as a Rectified Linear Unit (ReLU) is applied.
6. Output: Prediction mask (512*512*5).

The basic U-Net model is being explained and how to process SAR images. Now it has been suggested by enhancing U-Net performance by replacing the encoder part with a group of backbone models that pre-trained weights on the ImageNet dataset consisting of 1000 classes of images to extract more features from SAR images. In this thesis, four models are implemented and selected the best three backbone models (DenseNet-201, Inception V3, and Efficientnet-b3).

3.5.2.1 DenseNet-201 Backbone

Dense Blocks are used to create the dense connections between layers that give DenseNet its name as a type of CNN. There are connections between all layers (with the same feature-map size) directly with each other. Input image (512*512*3) enter to DenseNet backbone start from add ZeroPadding2D [3*3][3*3] so the input resizes to become (518*518*3) and pass it to conv1_Conv2D with filter size 7*7 after that apply BatchNormalization and use Relu activation function the output concatenate to stage 3 upsampling side. Other ZeroPadding2D [1*1][1*1] apply to the same output before entering conv2 (stage2 consists of 6 blocks). Each block in the same downsampling stage (consisting of two Conv, BN and Relu) has two links the first link connects each block with the next one and the second link skips the block and connects the directory with concatenate that became after block as shown in Figure 3.4

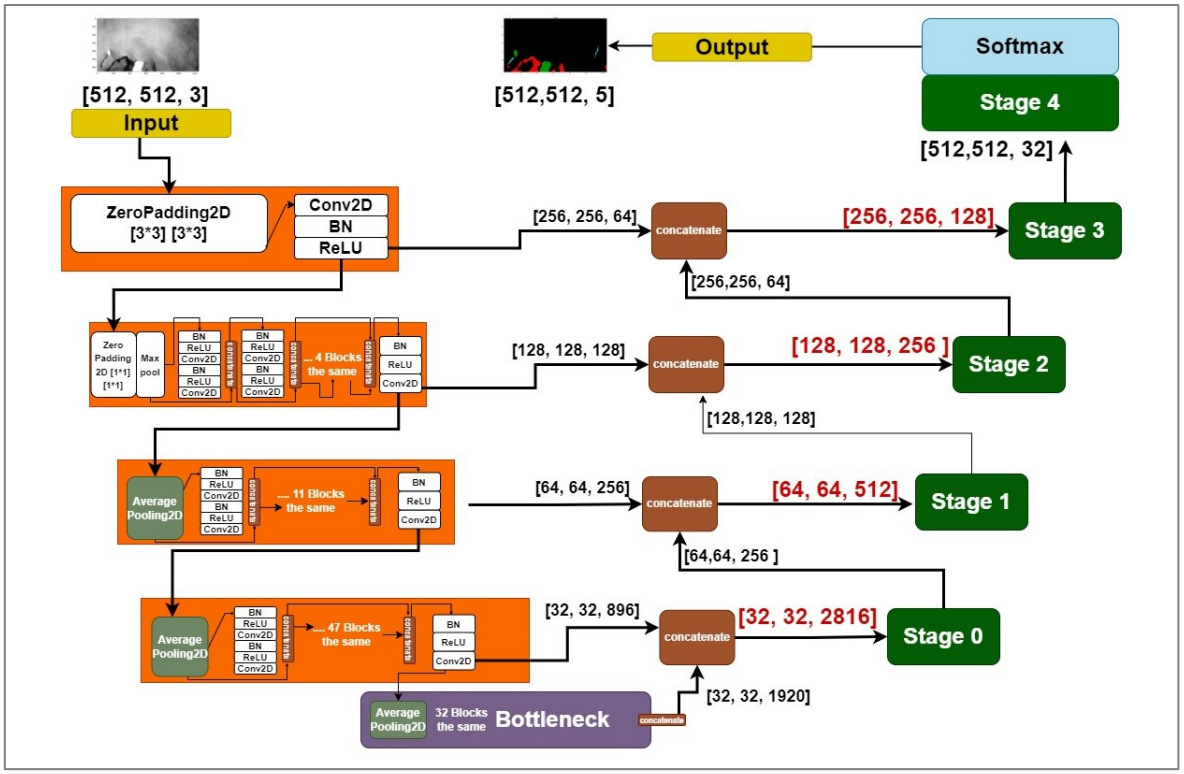


Figure 3.4 Architecture DenseNet-201 Backbone with U-Net

In stage 3 downsampling there are 12 Blocks and in stage 4 downsampling there are 48 blocks. Finally, in the bottleneck, there are 32 blocks all of which have the same block structure (consisting of Conv, BN, Relu, Conv, BN and Relu) but vary in conv kernel size and absolutely the number of blocks. At each end downsampling stage, there are two links one goes directly to upsampling and the other one connects to the next downsampling stage. The maximum channel gets from this backbone is 2816.

3.5.2.2 Inception V3

The Inception V3 is a CNN architecture that incorporates several updates, such as Label Smoothing and Factorized 7 x 7 convolutions, and is a member of the Inception family of neural network designs. This architecture starts with 3 blocks consisting of (Conv, BN, and Relu) to resize input (512, 512, 3) to become (256, 256, 64) and increase channel size as show in Figure 3.5 after that output connect with the max pool in the next downsampling stage

to decrease the size of the vector to half (128,128, 64) at the same time another link connects directory to upsampling stage 3. Stage 2 downsampling has two blocks consisting of (Conv, BN, and ReLU) and the output become (128, 128, 192). The other downsampling stages have a combination of layers that work in parallel at the same time and collect their outputs into a unit layer called the mixed. There are 10 mixed-in Inception V3. To obtain the output in a mixed layer, we combine the results from four parallel connections that are all connected to the same input. Strides and pooling are used to ensure that all parallel feature maps share the same first two dimensions (the number of feature maps can differ). The maximum channel gets from this backbone is 2048.

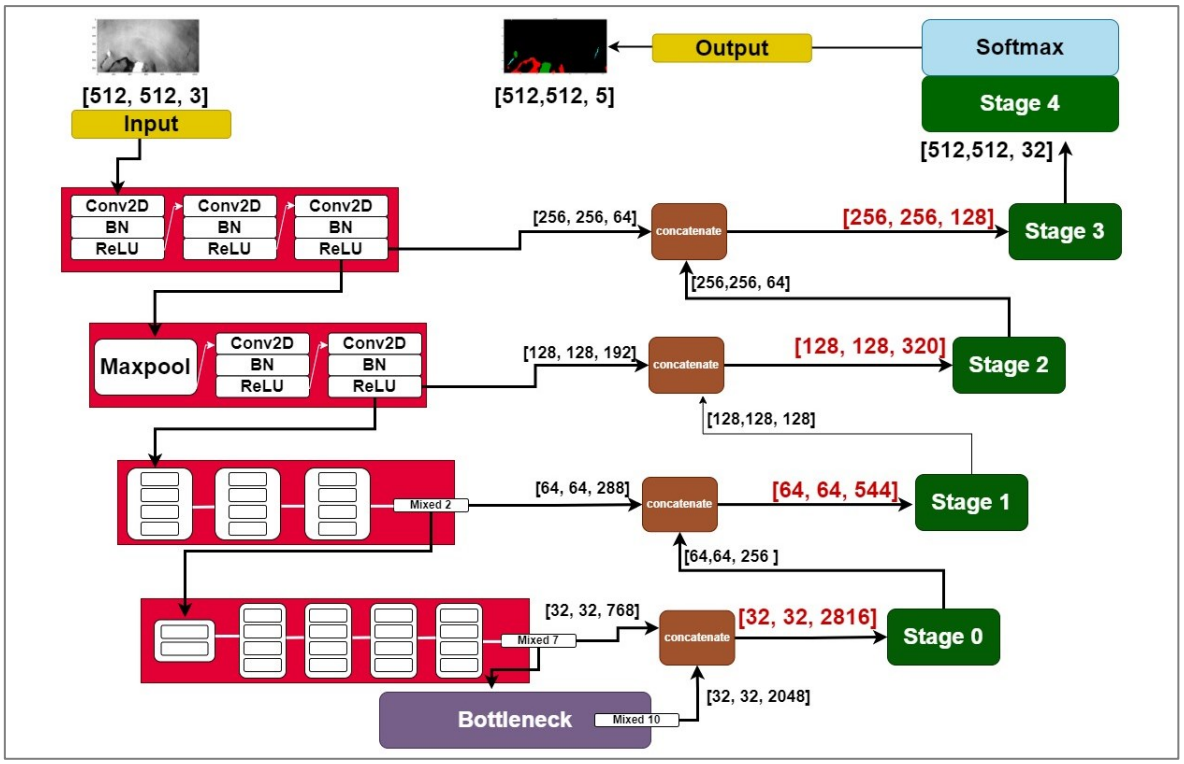


Figure 3.5 Architecture Inception V3 Backbone with U-Net

3.5.2.3 EfficientNet-b3 Backbone

To achieve a uniform scaling in terms of depth, width, and resolution across all dimensions of the image, Efficientnet-b3 uses a compound

coefficient to design the CNN architecture with a scaling method. The first stage of the process primarily applied 3*3 convolution with BN followed by Swish activation as shown in Figure 3.6 below the output became (256, 256, 40). Then applied convolution DepthwiseConv2D with BN followed by Swish activation output still (256, 256, 40) but in this block each channel gets a specific kernel. The benefit of DepthwiseConv2D using a different kernel for each input channel during the convolution process. All the other stages in downsampling begin with the same startup block that consists of (DConv2D, BN and Swish). The maximum channel gets from this backbone is 2304.

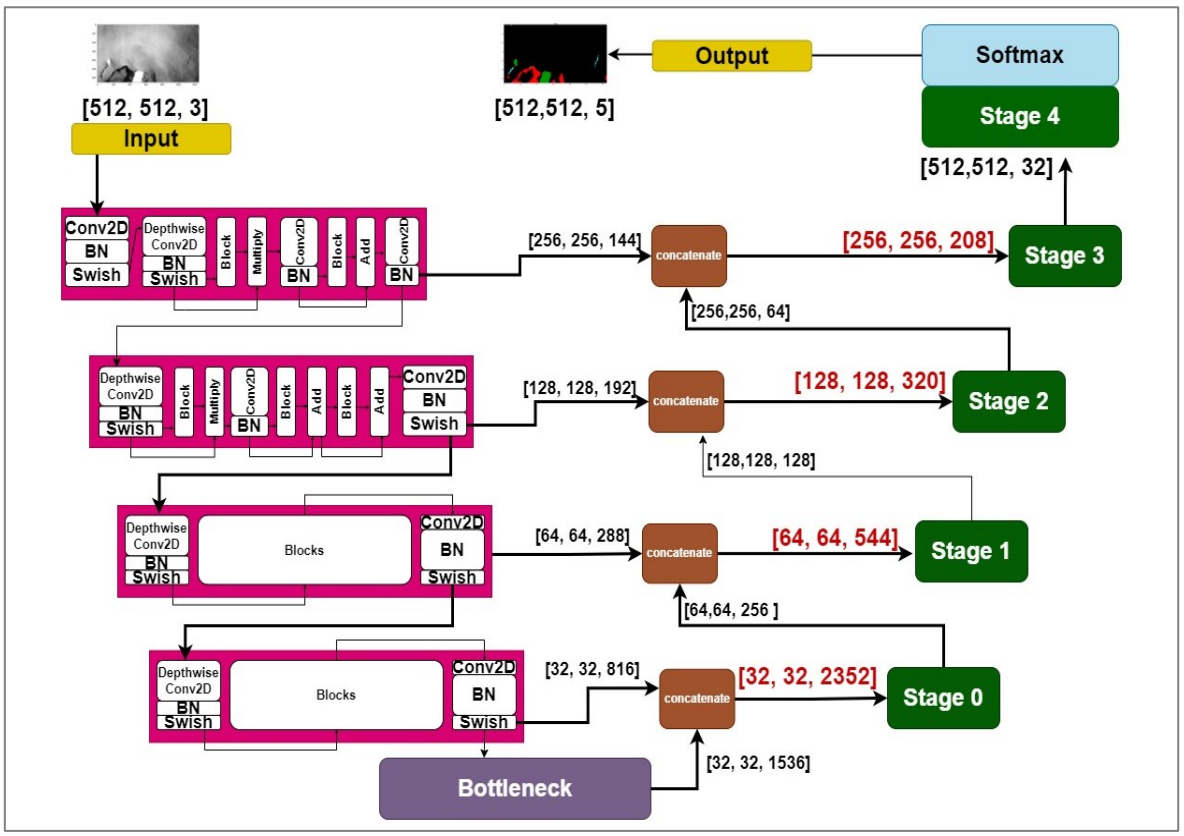


Figure 3.6 Architecture EffectionNet-b3 Backbone with U-Net

After explaining the backbones in detail we can summarize the advantage and disadvantages of each one all illustrate in Table 3.4.

Table 3.4 The Advantages and Disadvantages of Backbone

Backbone Type	Advantage	Disadvantage
DenseNet-201	-Relies on dense connections between layers.	-Complex architecture. -Need high GPU to implement.
Inception V3	Have a combination of layers that work in parallel.	-Complex architecture. -Need high GPU to implement.
Efficientnet-b3	-Fast in implementation. -Each channel gets a specific kernel when applying DepthwiseConv2D.	-When implementing it on SAR images get IOU less than DenseNet-201 and Inception V3.

3.6 Virtual Private Server (VPS) Cloud

A Virtual Private Server (VPS) is a dedicated server hosted by a third-party provider such as a web host or cloud storage provider. By splitting a physical machine into multiple private server environments that share resources, virtualization technology is used to host virtual private servers. VPS is a good option to gain access to models from everywhere. In this thesis, Time4VPS is used to host the model on it and get the IP public for each VPS as shown in Figure 3.7. The servers were used to host the best models on it. The three servers are reserved, and each one has its IP so that can be accessed via the Internet. After completing the model training phase and storing a special demo for each model, it was evaluated and the best among them was selected based on the value of IOU as it is the most important metric in the field of semantics segmentation. The best result is being got when deploying U-Net with (Densenet-201, Inception V3, and EffectionNet-b3) backbones.

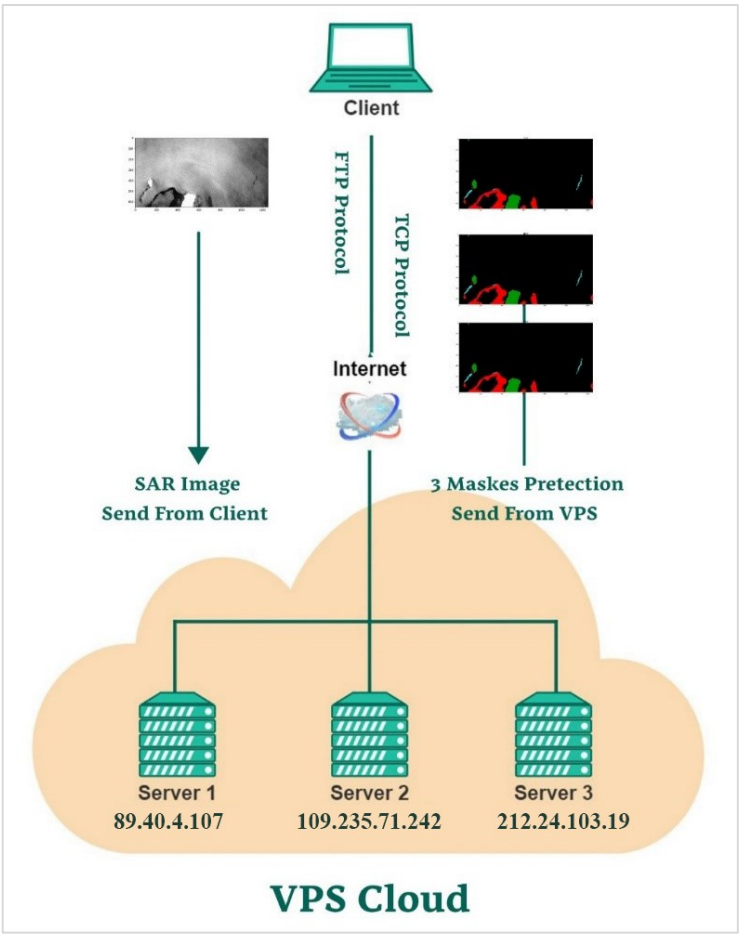


Figure 3.7 VPS Environment

The steps of cloud configuration and getting the public IP are summarized as follows:

1. Go to the official website of time4vps at https://www.time4vps.com/
2. Create a new account on the website by clicking on the signup tab and creating an account.
3. Fill in the required information, agree to the rules and policies, and click on the register button.
4. From the new order select the VPS OS and its resource that is compatible with the model requirement.
5. After choosing the billing cycle and paying it the VPS begins to install OS.
6. From the dashboard you can access VPS information to get (username, password and public IP).

3.6.1 Remote Desktop Protocol (RDP)

The configuration of a VPS can be initiated by connecting to it via Remote Desktop Protocol (RDP) after reserving a VPS and obtaninig the public IP address this is shown in Figure 3.8

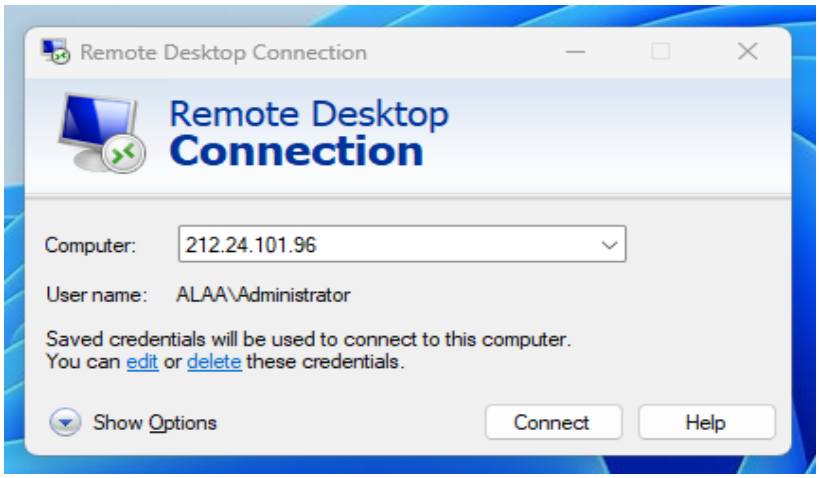


Figure 3.8 Remote Desktop Connection

3.6.2 VPS Setup

There are important steps that must deploy to enable the client to connect to the VPS server:

1. Install Python 3.10.
2. Install important DL Libraries (TensorFlow, Keras, C#).
3. Open port 5000 from inbound rules.

3.6.3 File Transfer Protocol (FTP)

With FTP, the image is transformed to the VPS for processing with U-Net mode and the prediction mask is reset from the VPS. So there are many important steps to configure FTP on the VPS side before deploying the model:

1. from Server Manager select add roles and feature
2. From Server Roles select Web Server (IIS)
3. From Role Services select FTP Serve

4. from Administrative Tools select IIS manager
5. Add a new FTP site
6. Enter the FTP name and Select the content directory (when image and mask save)
7. Choose port protocol (21) and SSL type
8. Detect how allow to access to the server (Enter the user account you previously created).
9. Open 20 -21 ports from inbound rules.
10. From the client side open file explorer and write ftp://ftpuser:FtP20202323@176.223.131.48/ to check the access to VPS.

3.6.4 Connect Client with VPS

A client initiates the communication by connecting to a VPS via public IP and a specific port (5000) using TCP as a trigger to deploy the model. The client sends requests to the server, and the server returns replies. A client can send three servers at the same time. There are multi-steps on each side to implement a connection. It will explain in the next section.

3.6.4.1 Client Side

A client-side connection is established by running a C# form that contains all public IP addresses. Below are the steps that summarize the process:

1. Run the C# form.
2. Create a thread to enable a multi-server connection.
3. From the (Choose File) button select an image.
4. Create TCPClient (act as a trigger) to start the connection.
5. Sent image to FTP server

3.6.4.2 Server Side

These steps summarize the procedure on the server side after the connection is established between them:

1. Open a connection with the client.
2. Read the image from the FTP folder.
3. Run a prediction function on the server side that includes.
    - Preprocessing backbone.
    - Split image to patch.
    - Process image with the main model.
    - Return the prediction mask to its original size.
    - Assign a color to the prediction mask.
    - Calculate pixels for each class in the image.
4. Resent mask prediction to the client folder.
5. Show mask in C# form .

Chapter Four

Results and Discussion

4.1 Introduction

The previous chapter described the work of the oil spill detection system and its sequence of operations. The system's performance results will be discussed in this chapter, which is divided into two phases: training, validation and testing in google colab and the deployment phase in VPS. There are three steps in the first phase: results of the preprocessing steps, evaluation of each model and results of the best three models. The major topics that will be covered in this section include the SAR image database and its partitioning, the proposed method's outputs in detail, and the evaluation of the system through performance calculations using SAR images. Hardware and Software Requirements

The proposed oil spill detection System operates by using the following devices:

- **Personal Computer MSI**: (Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz 2.59 GHz, RAM 16 GB, windows 11, and 64-bit OS).
- **VPS:** Three VPS from time4VPS (OS: Windows Server 2016 (64-bit) connecting to KVM Windows server using RDP, Processor: 3 x 2.6 GHz, Memory: 8192 MB, Storage: 80 GB, Bandwidth: 100 Mbps (Monthly limit: 16 TB).
- **Google Colab Pro**: is an open Jupyter notebook environment that works completely in the cloud (RAM 52 GB, Disk 166.8 GB, and GPU 26 GB).

The proposed system is implemented by using GPUs from Google Colab. It is a Google application that enables programmers to run Python code directly from their web browser. It is an ideal platform for DL tasks as it allows us to train a large database and build complex models very quickly on Google computing resources (GPU).Python is used to implement the CNN code, which is based on the TensorFlow open-source framework (a Google open-source

programming library focused on supporting tensors to produce successful work), Keras (a neural network library based on TensorFlow written in Python), and the Python standard library. The software uses free, publicly available libraries like Open, Scikit Learn, and Pandas. These libraries are specifically geared toward machine learning and data science tasks.

4.2 SAR Images Dataset Preparations

Due to the limited number of images and the requirement for a large number of training images for DL models, the dataset was split into three samples: 70% (or 800 SAR images) for training, 20% (or 202 SAR images) for validation, and 10% (or 200 SAR images) for testing (110 SAR images) as shown in Table 4.1.

Table 4.1 The Statistics of the SAR Images Divide

Train	800
Validation	202
Test	110
Total	1112

The SAR image size (1250 * 650) is not compatible with the models being proposed because the models require a maximum image size (512 * 512) to fit onto U-Net-based segmentation. Hence, the Patchify library is used to splits an image into multiple patches of the same size. The goal of this step is to make sure that all data sets can be used successfully. Because U-Net segmentation will be used, it must deploy the same procedure on the image and mask as well. Figure 4.1 shows the dataset image with a mask.

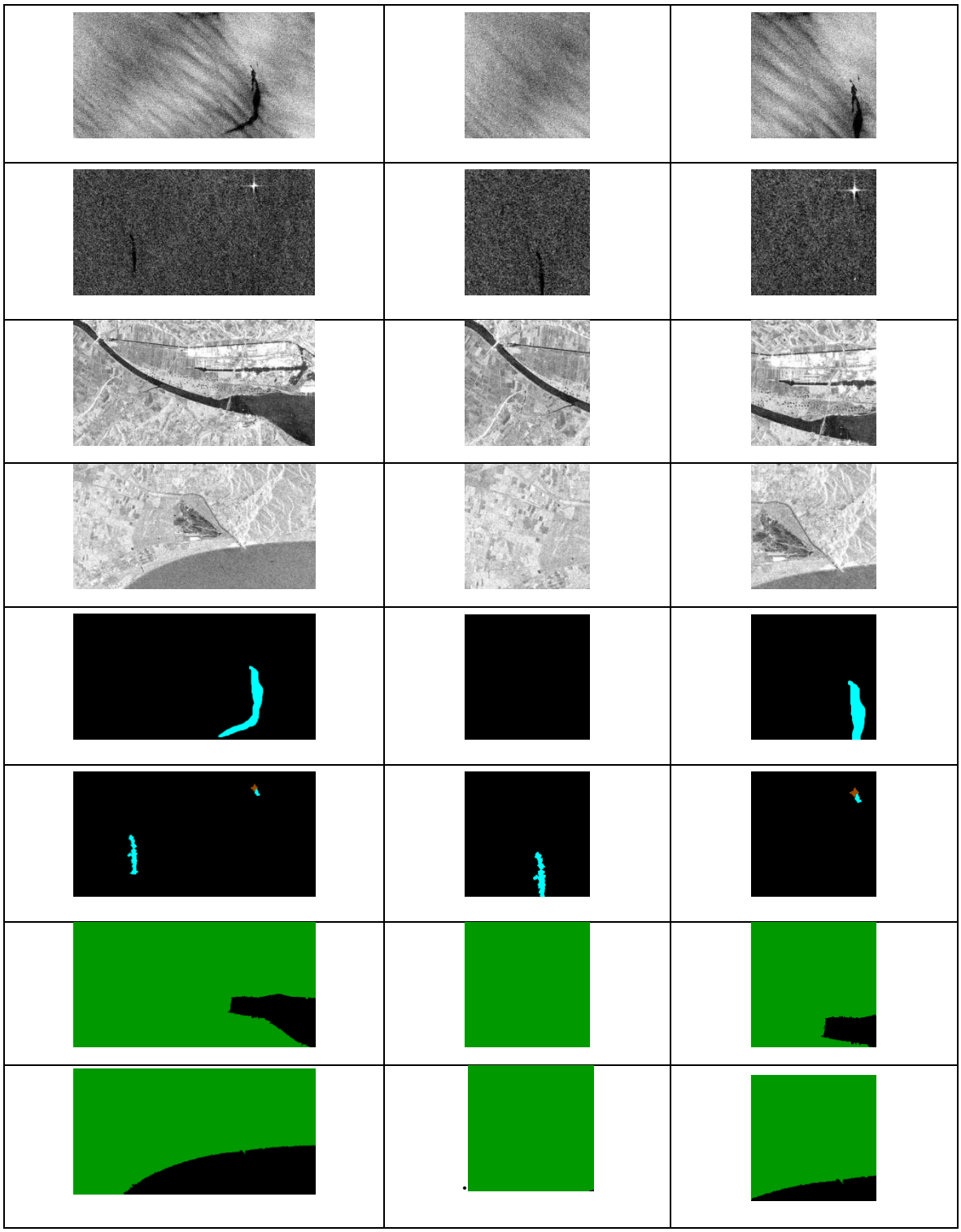


Figure 4.1 Some Samples of the Dataset with Mask (Before and After Split).

As a result of splitting images with their mask, the size of the dataset is doubled as shown in Table 4.2.

Table 4.2 Statistics of the SAR Images Divide After Split

Train	1600
validation	404
Test	202
Total	2206

After the patchy dataset now applies data augmentation on the training dataset only. Data augmentation is an effective and significant method for training an algorithm. The albuminization library was used to apply data augmentation on training images only. Nine techniques will be implemented in this thesis that explain in chapter 3 (Table 3.2). Figure 4.2 shown some sample of images after augmentation with their masks.

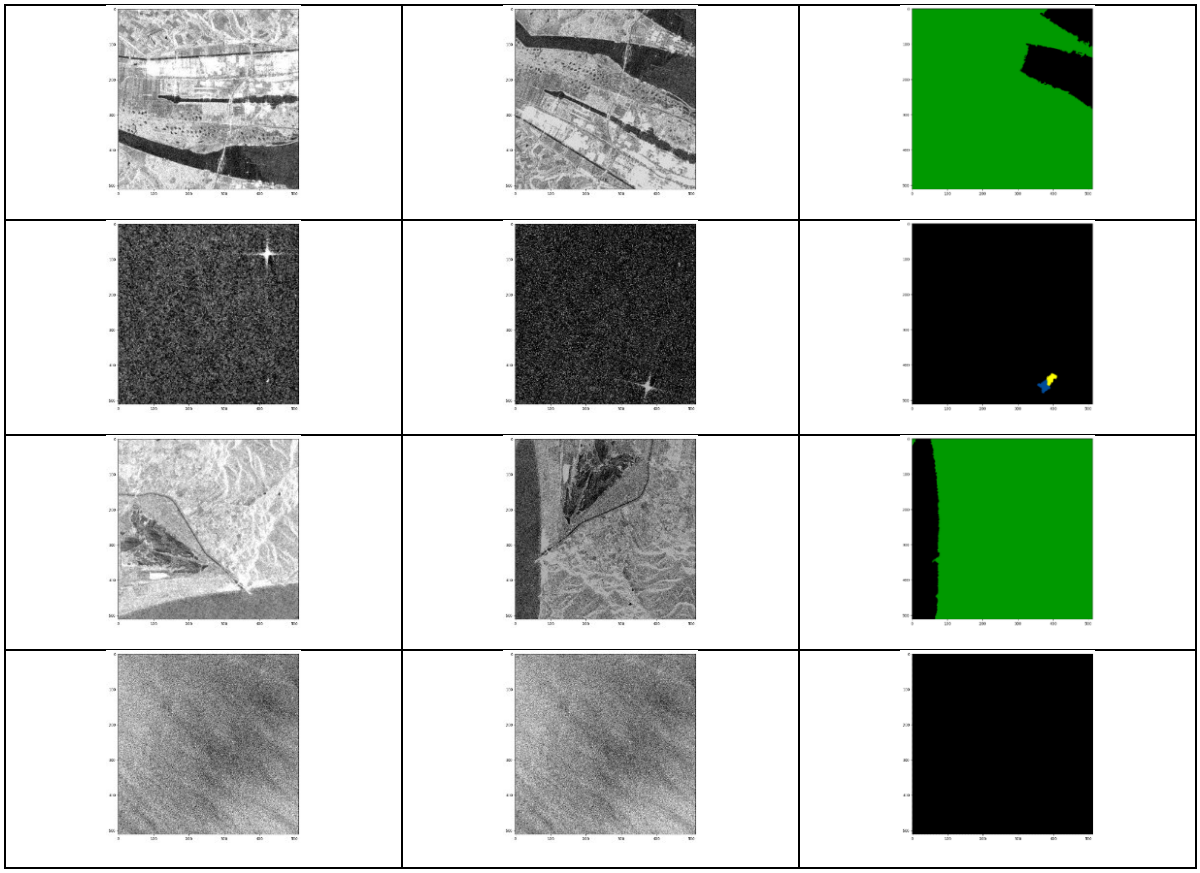


Figure 4.2 Images with Masks (After Augmentation)

The green wide area in the samples below represents the land and the star represents the ship. As shown there is a variety in pixel area between the two classes and that makes M4D dataset imbalanced. Through used albumentations library, the developer can determine the probability of implementing each technology independently. By determining the value of (P) that is within the library parameters, between 0 and 1, the higher the decimal value, the greater probability of occurrence, and vice versa.

4.3 Results of the Implementation of the Proposed System

During this section, a detailed description will be present of the outputs generated while following the steps in the proposed oil spill system. The outputs are classified into two phases: those generated in the training phase, and those generated during the test phase.

4.3.1 Training Phase Outputs

In the training phase, all training samples (1600 SAR images) with their mask must pass through the system simultaneously to be trained. Meanwhile, the validation samples (404 SAR images) are passed along to the model to be compared between them and to check the performance of the model.

In the training phase, SAR image segmentation uses the Adam optimization, and learning rate reduction to improve the value of the validation loss when model performance stops increasing (reduce on the plateau). Dice loss with focal loss apply in this thesis with batch size = 12 for training and batch size = 4 for validation and early stopping to find the best number of epochs (maximum = 50) with the condition if val-loss does not decrease after 20 epochs than stop training. Also, callback applies with model checkpoint to save the best result on file in h5 format.

In this thesis, the original U-Net is deployed and compared the results with four backbones after that select the best three backbones according to IOU. In

the following a result and output SAR images of all backbones in the training phase:

4.3.1.1 U-Net

It has been started with the learning curve of the original U-Net algorithm to represent IOU and accuracy in the training phase as shown in Figure 4.3 below, there is instability in the training process, whether in learning curve accuracy or learning curve IOU.

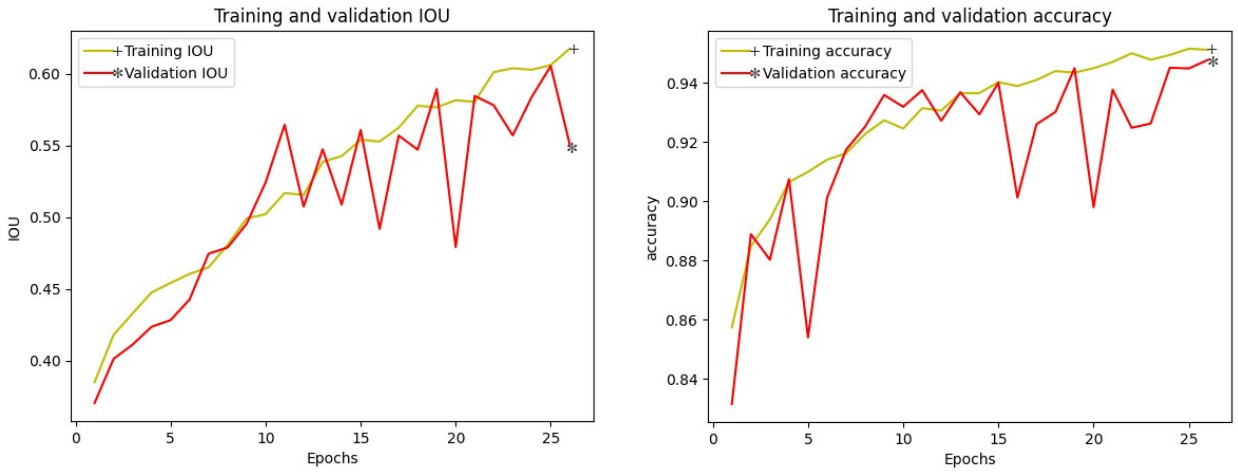


Figure 4.3 The IOU and Accuracy Learning Curves of Train And Validation (U-Net)

After the train U-Net is completed. Now the backbones are deployed with U-Net and it has been noted as shown in the next learning curves (IOU and accuracy of training and validation) plot enhance fit condition when the model is trained with backbones, because stability is increased in the validation line (red line), and the gap was decreased between the two-IOU values. It is potentially persistent training of a good fit that would produce a problem of overfitting so an early stopping was used when training the model.

4.3.1.2 DenseNet201

The DenseNet201 is considered the most complex architecture among the backbones used with the U-Net, so it was noted that the model was unstable at the beginning of the training. However, the training and validation curves

were synchronized in terms of ups and downs, and this indicates the validity of the training on the data used with the convergence of the curves in the last stages of training. As show in Figure 4.4 below.

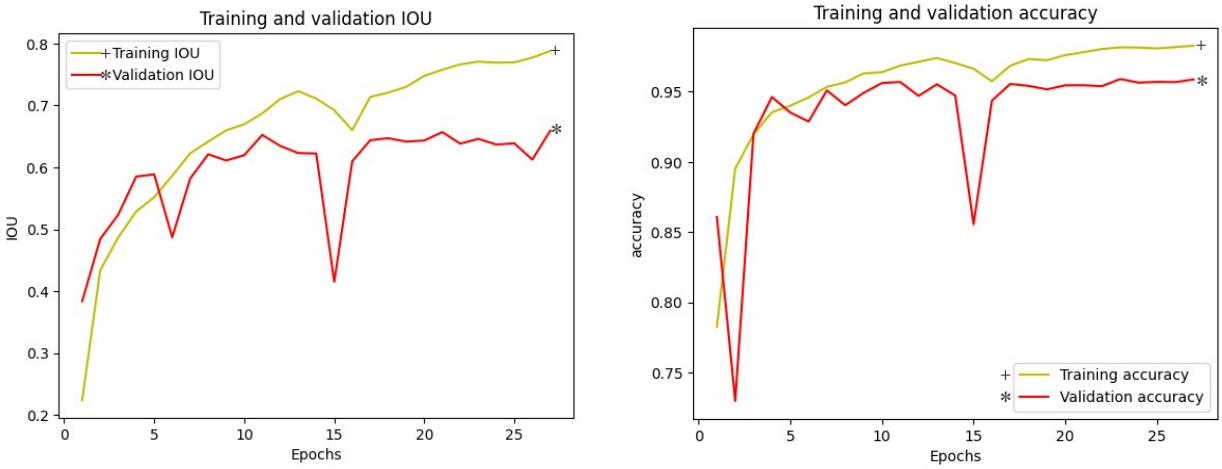


Figure 4.4 The IOU and Accuracy Learning Curves of Training and Validation (DenseNet201)

4.3.1.3 EfficientNetb3

This model is of medium complexity in terms of architecture, and as shown in the Figure 4.5, it achieved stable training and a and the gap is very little between the training and validation curves.

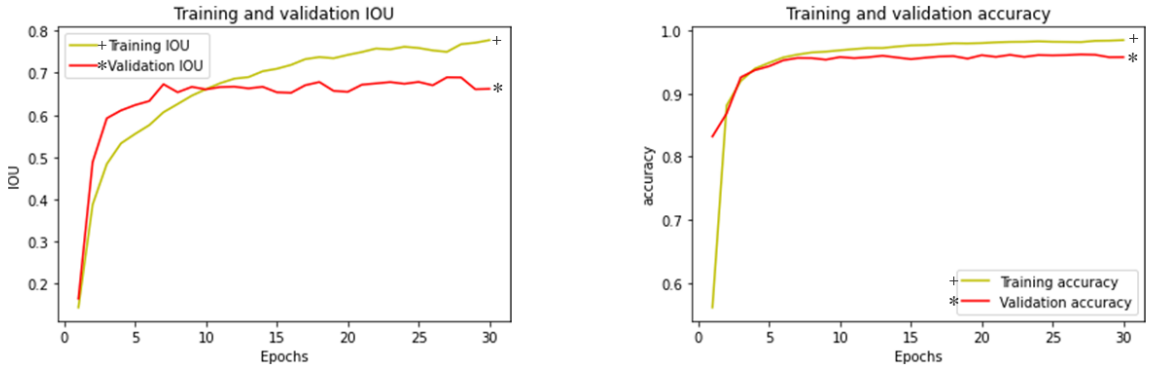


Figure 4.5 The IOU and Accuracy Learning Curves of Training And Validation (EfficientNetb3)

4.3.1.4 Inception V3

The learning curves plot in Figure 4.6 illustrates the good fit condition when the model is trained with Inception V3, because the training and the validation depreciate to a point of equilibrium, and the gap is very little between the curves.

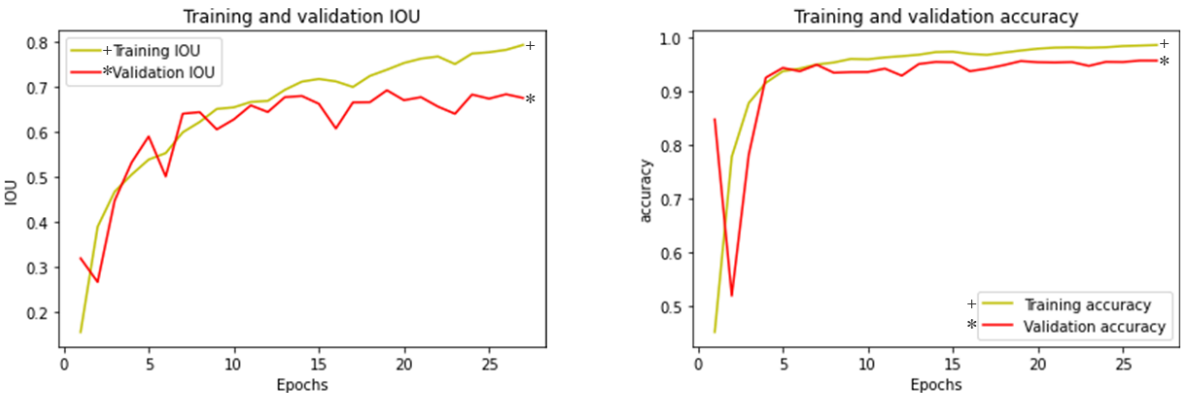


Figure 4.6 The IOU and Accuracy Learning Curves of Train and Validation (Inception V3)

4.3.1.5 Resnet101

The training in this model was unstable between the training and validation curves and recorded the lowest results compared to the rest of the models. As show in Figure 4.7.

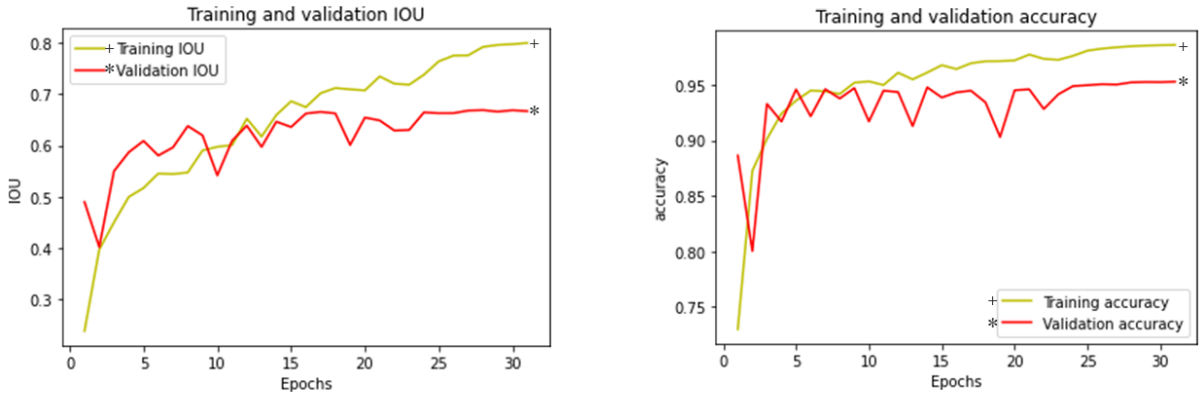


Figure 4.7 The IOU and Accuracy Learning Curves of Train and Validation (Resnet101)

In the Table 4.3, we summarized all results and we noticed the highest IOU value was detected in Densenet201 because there are L (L + 1) / 2 connections. The outputs (feature maps) from each layer are used as inputs for the next layer. It is important to consider how the connection between layers can impact learning. And the lowest value was detected in U-Net without a backbone. Because the architecture is simple and does not extract a sufficient amount of image features.

Table 4.3 Summary Backbones Result in the Training Phase

Backbone Type	IOU	Accuracy	Learning Rate	Early Stopping
U-Net	55.07	94.77	LR: 1.0000e-04	Epoch 2
Densenet201	**68.32**	95.60	LR: 1.0000e-04	Epoch 21
Inception V3	67.55	95.72	LR: 1.0000e-04	Epoch 27
EfficientNetb3	66.22	95.76	LR: 1.0000e-04	Epoch 30
Resnet101	65.72	95.29	LR: 1.0000e-05	Epoch 31

LR = 10^{-4} was determined automatically during training, and if results didn't improve, it was lowered to LR = 10^{-5} so that the model could learn more effectively, just like in Resnet101. Each model is then exported to a separate h5 file in Google Drive following the training phase.

4.3.2 Testing Phase Results

In the testing phase, all test samples that are 10% of the database (202 SAR images) will be passed to the system without their mask to be tested. The

preprocessing results in the test phase do not differ from the preprocessing results in the training phase that were explained in section (4.3), except for the data augmentation technique that is not applied at this phase because the test must be on real data. To be able to compare the performance of the models, the results are summarized in a statistical chart Figure 4.8

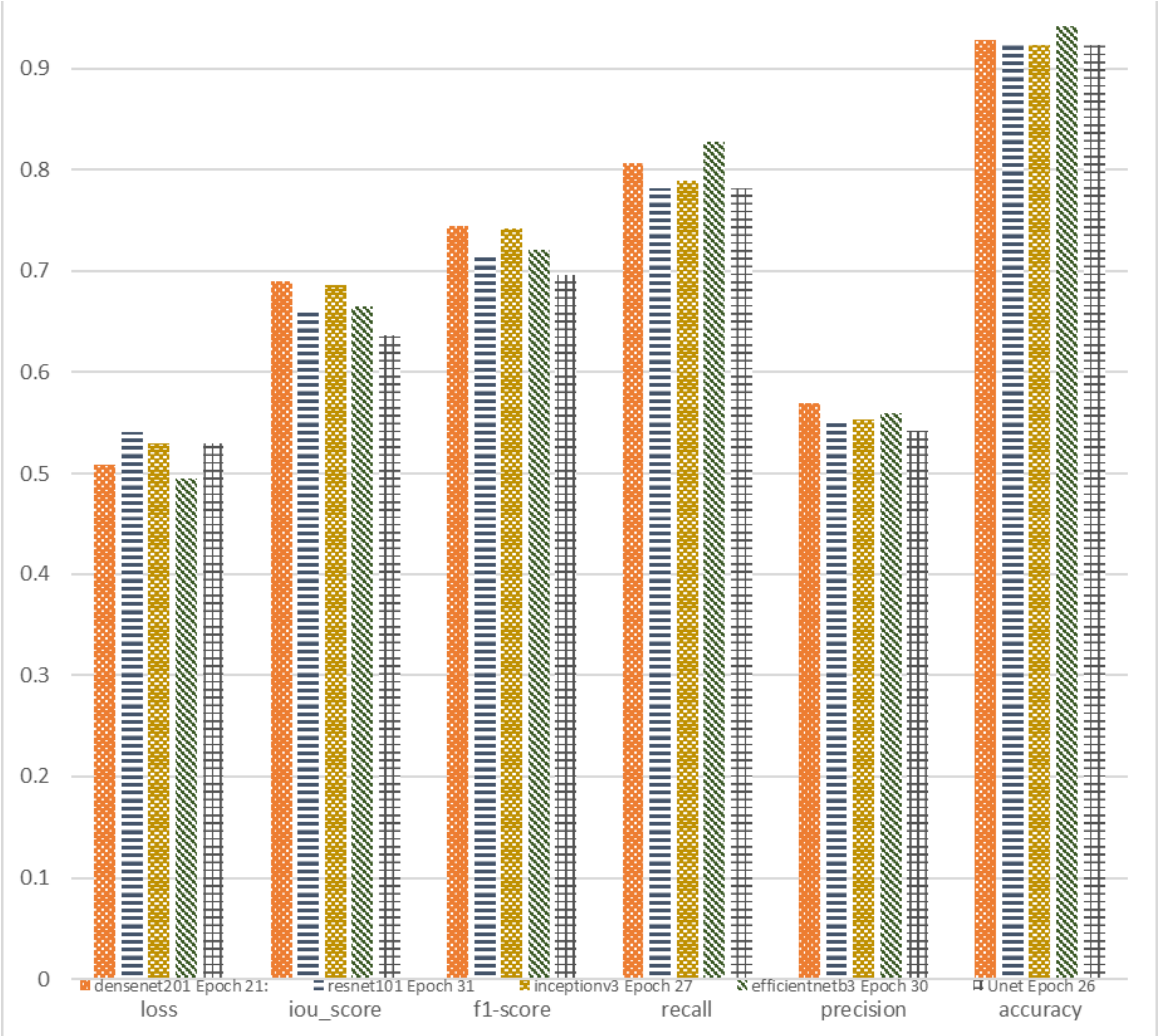


Figure 4.8 Models Result in the Test Phase

Table 4.4 contains all the values calculated in the test phase to provide more details about each model's performance.

Table 4.4 Summary of Backbones Result in the Test Phase

Backbone Type	Unet	Densenet201	Inception V3	Efficientnetb3	Resnet101
Total Params	31 M	26 M	29 M	17 M	51 M
Trainable Params	31 M	26 M	29 M	17 M	51 M
Non-Trainable Params	12 K	231 K	36 K	89 K	99 K
Loss	52.94	50.84	53.04	49.51	53.04
IOU	63.69	68.95	68.6	66.51	65.99
F1-Score	69.62	74.46	74.15	72.10	71.54
Recall	78.12	80.66	78.89	82.75	78.18
Precision	54.27	56.91	55.33	55.97	54.96
Accuracy	92.32	92.79	92.25	94.12	92.45

On reviewing the table, it also noticed the highest value for IOU in the test stage was recorded by a Densenet201model. Next, you will find two examples showing the predictive efficiency of each model. The original image and original mask for each class can be found on the first line called ground truth mask which shows each class mask individually by using the squeeze, followed by the results for each model from highest result to lowest result.

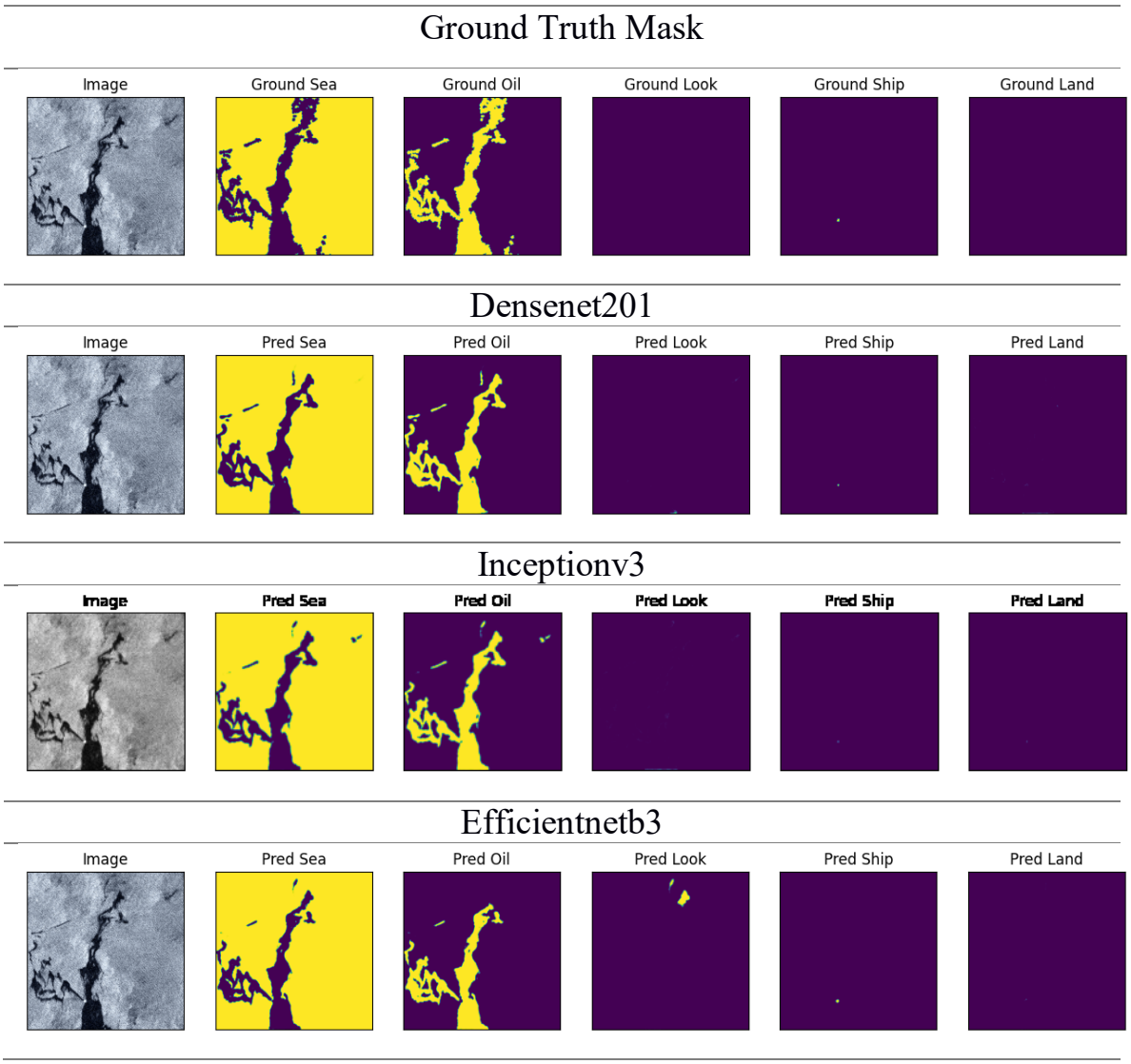


Figure 4.9 Example 1

in Figure 4.9 is noted that the Efficientnet3 model predicted a look-alike class. Both Densenet201 and Inception V3 predicted larger areas of the oil spill. This prediction is consistent with the original mask. Ship class is considered the smallest class within Dataset, and all models were able to predict it. Inception V3 recorded the lowest predicted for ship class.

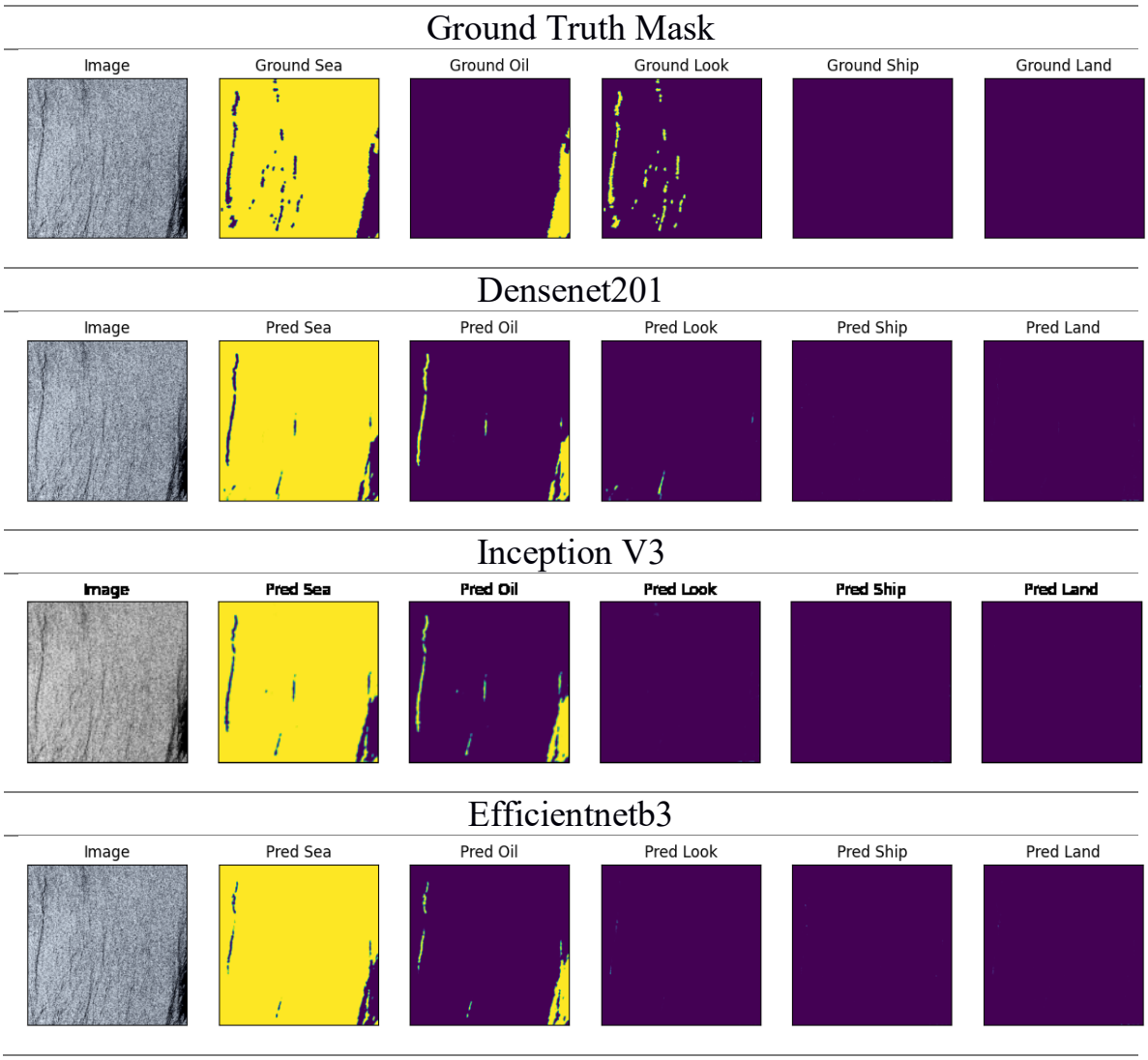


Figure 4.10 Example 2

In the Figure 4.10, it was found that the original mask had two classes: oil spill and look-alike. It is considered the most difficult class to predict. This is because the feature of its image is very similar to the oil spill, so it is difficult for models to predict it.

4.4 Comparison of the Proposed System Results with Related Works

Using the same dataset, Table (4.5) compares oil spill detection methods with the proposed system. It should be noted that the proposed system achieved higher performance than the other existing systems. Select complex structure backbones as the SAR image considers composite data, so it requires more layers to extract features. The main reason for obtaining these results was to first select complex structure backbones. Furthermore, during the preprocessing phase, we split the image in half to guarantee high quality.

Table 4.5 Shows a Comparison of the Proposed Method with Other Methods.

Batch Size	Epoch	IOU	Num. Of Parameter	Backbone	Model
Proposed System Result					
12	**50**	**68.94**	**26,379,157**	**Densenet201**	**U-Net**
Note: the model stop in epoch 21 because it is the best result					
Krestenitis et al. (2019)					
12	**50**	**65.06**	**51,606,046**	**MobileNetV2**	**DeepLapV3+**

4.5 Deploy Models on VPS

When a multi-model classifier has been trained and compared to each other, the best model is selected, and deploy the demo on VPS is to test real images with full size and calculate the pixel area of each class in images. The three VPS are rented from Time4VPS and upload a demo.h5 file for three best models to VPS. The models are distributed as shown in Table 4.6.

Table 4.6 Information about VPS

Server num	IP address	Backbone type
Server 1	Inception V3	89.40.4.107
Server 2	Densenet 201	109.235.71.242
Server 3	EffectionNetb3	212.24.103.19

The image that is sent from the client side with the original size process via all VPS simultaneously as shown in Figure 4.11, they predict masks and calculate the pixel area for each object in the image. During the modelling process, grey masks are created, so it is important to color map all classes according to the color they were assigned from the M4D team dataset. Once all these steps are complete, the VPS will send the mask to the client.

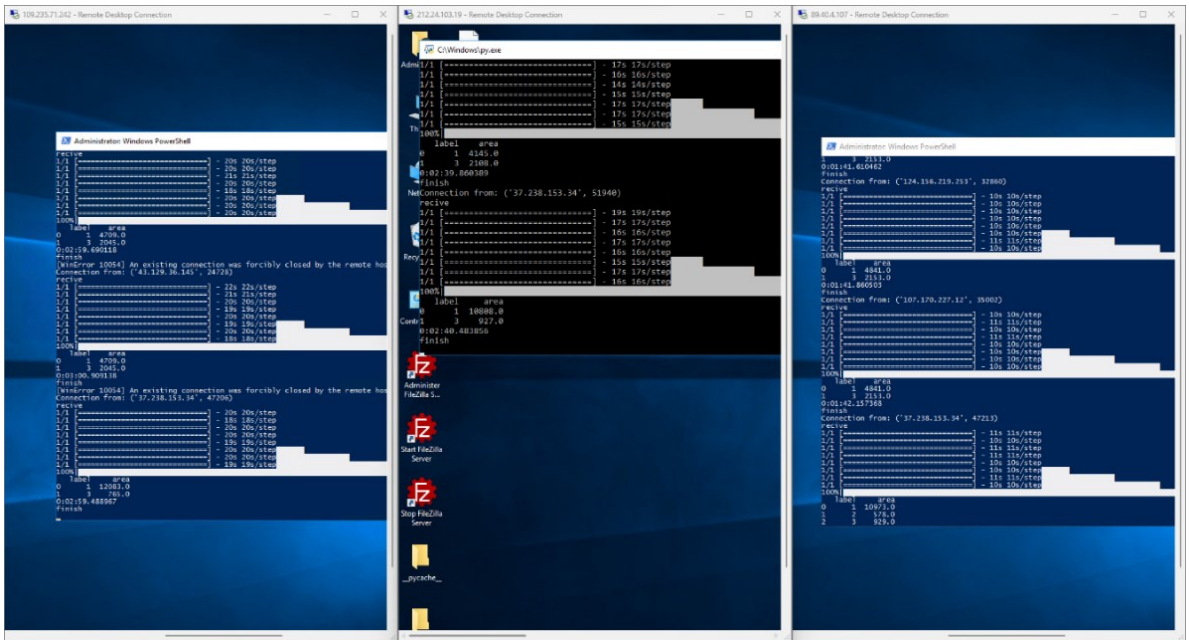


Figure 4.11 VPS Side

On the client side, there are five images shown in the App as shown in Figure 4.12 the original image with mask and three prediction masks.

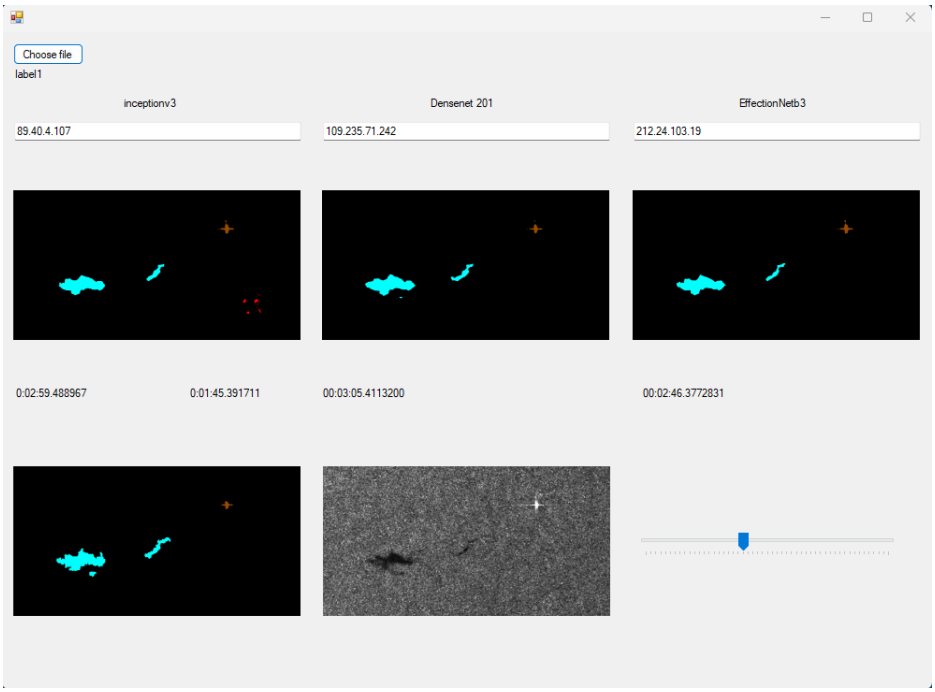


Figure 4.12 Application on the Client Side

Table 4.7 shows model predictions for the transmitted image in more detail and show the performance of each model. The highest prediction of ships

presence was achieved by the first model. However, it predicted the look-alike even if it was not present in the original image. On the other hand, the other models' prediction was correct, and the second model achieved the highest value in the oil spill class.

Table 4.7 VPS Pixel Area Prediction

Server num	Label 0 Sea	Label 1 oil_spill	Label 2 Look_alike	Label 3 Ship	Label 4 Land
Server 1	--	10973.0	578.0	929.0	--
Server 2	--	12083.0	--	765.0	--
Server 3	--	10808.0	--	927.0	--

Chapter Five

Conclusions and Future Works

5.1 Conclusion

These are the essential conclusions of the results obtained from segmentation SAR images for oil spills using the suggested method:

- In this thesis, split image technique has been used in preprocessing step instead of resizing image to get the appropriate size to fit onto U-Net segmentation model without loss image features.
- When the images are divided in half, they become twice as many, providing more data and allowing the model to be trained on more photos.
- Each dataset has properties and requires a special model to process them. Since the oil spill data is complex dataset we used pre-trained and architecturally complex models within special criteria to be able to extract as many image characteristics as possible.
- The U-Net model was implemented independently without changing its architecture. In order to enhance the original model's performance and compare its final results, the encoder was removed and replaced with backbones (DenseNet-201, Resnet101, EffectionNet-b3, and Inception V3) pre-trained on massive data.
- Since the oil spill images are irregular in shape, more than one model has been trained, and their work has been synchronized to show the user more than one result at the same time. Therefore, it depends on the opinion of the majority.
- Hosting the best models according to results on private servers so that the customer can examine the images at any time and calculate the number of pixels for each class.
- Previously, different versions of the backbone and model resulted in IOU values less than 65%. This thesis used variety models and record the highest result is an IOU value of 69%.

5.2 Future work

The proposed work can be extended in future as illustrated in the following points:

- **Real Time:** Developing the model to receive images from satellites or cameras directly from anywhere and then process them in real time.
- **Iraq Oil Spill Dataset:** Creating the Iraq oil spill dataset (pipeline or marine images) and then testing them based on oil spill detection model.
- **Distributed Nodes:** Training phase can be separated across multi nodes cloud environment.
- **Backbones Models:** Deploy more backbone models for the purpose of training data thus provides a large base for evaluating the performance of different backbones in training phase.

REFERENCES

[1] R. P.Patel, "International Journal of Business & Management," *IOSR J. Bus. Manag.*, vol. 18, no. 08, pp. 139–142, 2016, doi: 10.9790/487x-180803139142.

[2] G. Melillos and D. H. Hadjimitsis, "Oil Spill Detection Using Sentinel 1 SAR Data at Cyprus's Coasts", *melillos2021oil.*, vol. 11729, pp. 191-196, 2021, doi: 10.1117/12.2585886.

[3] P. F. Kingston, "Long-Term Environmental Impact of Oil Spills," *Spill Sci. Technol. Bull.*, vol. 7, no. 3–4, pp. 53–61, 2002, doi: 10.1016/S1353-2561(02)00051-8.

[4] P. Liu, C. Zhao, X. Li, M. He, and W. Pichel, "Identification of Ocean Oil Spills in SAR Imagery Based on Fuzzy Logic Algorithm," *Int. J. Remote Sens.*, vol. 31, no. 17, pp. 4819–4833, 2010, doi: 10.1080/01431161.2010.485147.

[5] W. Ding, J. Nayak, H. Swapnarekha, A. Abraham, B. Naik and D. Pelusi, "Fusion of Intelligent Learning for COVID-19: A State-Of-The-Art Review and Analysis on Real Medical Data," *Psychiatry Res.*, vol. 14(4), no. January, p. 293, 2020.

[6] H. Liu and B. Lang, "Machine learning and Deep Learning Methods for Intrusion Detection Systems: a Survey," *Appl. Sci.*, vol. 9, no. 20, 2019, doi: 10.3390/app9204396.

[7] D. C. Cireşan, A. Giusti, L. M. Gambardella, and J. Schmidhuber, "Deep Neural Networks Segment Neuronal Membranes in Electron Microscopy Images," *Adv. Neural Inf. Process. Syst.*, vol. 4, pp. 2843–2851, 2012.

[8] H. Su, F. Xing, X. Kong, Y. Xie, S. Zhang, and L. Yang, "Robust Cell Detection and Segmentation in Histopathological Images using Sparse Reconstruction and Stacked Denoising Autoencoders," *Advances in Computer Vision and Pattern Recognition*, no. 9783319429984. pp. 257–

278, 2017, doi: 10.1007/978-3-319-42999-1_15.

[9] S. Garg, K. Kaur, N. Kumar, G. Kaddoum, A. Y. Zomaya, and R. Ranjan, "A Hybrid Deep Learning-Based Model for Anomaly Detection in Cloud Datacenter Networks," *IEEE Trans. Netw. Serv. Manag.*, vol. 16, no. 3, pp. 924–935, 2019, doi: 10.1109/TNSM.2019.2927886.

[10] M. Krestenitis, G. Orfanidis, K. Ioannidis, K. Avgerinakis, S. Vrochidis, and I. Kompatsiaris, "Oil Spill Identification from Satellite Images using Deep Neural Networks," *Remote Sens.*, vol. 11, no. 15, pp. 1–22, 2019, doi: 10.3390/rs11151762.

[11] M. Shaban, R. Salim, H. Abu Khalieh, A. Khelifi, A. Shalaby, S. El-Mashad, A. Mahmoud, M. Ghazal and A. El-Baz, "A Deep-Learning Framework for the Detection of Oil Spills from SAR Data," *Sensors*, vol. 21, no. 7, 2021, doi: 10.3390/s21072351.

[12] Y. Fan, X. Rui, G. Zhang, T. Yu, X. Xu, and S. Poslad, "Feature Merged Network for Oil Spill Detection using SAR Images," *Remote Sens.*, vol. 13, no. 16, 2021, doi: 10.3390/rs13163174.

[13] K. Zeng and Y. Wang, "A Deep Convolutional Neural Network for Oil Spill Detection from Spaceborne SAR Images," *Remote Sens.*, vol. 12, no. 6, 2020, doi: 10.3390/rs12061015.

[14] S. T. Yekeen and A. L. Balogun, "Automated Marine Oil Spill Detection using Deep Learning Instance Segmentation Model," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. - ISPRS Arch.*, vol. 43, no. B3, pp. 1271–1276, 2020, doi: 10.5194/isprs-archives-XLIII-B3-2020-1271-2020.

[15] Y. Xiong and H. Zhou, "Oil Spills Identification in SAR Image Based on Convolutional Neural Network," *14th Int. Conf. Comput. Sci. Educ. ICCSE 2019*, vol. 1, no. Iccse, pp. 667–670, 2019, doi: 10.1109/ICCSE.2019.8845383.

[16] Z. Ghorbani and A. H. Behzadan, “Monitoring Offshore Oil Pollution Using Multi-Class Convolutional Neural Networks,” *Environ. Pollut.*, vol. 289, no. July, p. 117884, 2021, doi: 10.1016/j.envpol.2021.117884.

[17] Li, Yan Yang, Xiaofei Ye, Yunming Cui, Lunan Jia, Binfeng Jiang, Zhongming Wang and Shaokai, “Detection of Oil Spill Through Fully Convolutional Network,” *Commun. Comput. Inf. Sci.*, vol. 848, no. June, pp. 353–362, 2018, doi: 10.1007/978-981-13-0893-2_38.

[18] X. Yaohua and M. Xudong, “A SAR Oil Spill Image Recognition Method Based on Densenet Convolutional Neural Network,” *Proc. - 2019 Int. Conf. Robot. Intell. Syst. ICRIS 2019*, pp. 78–81, 2019, doi: 10.1109/ICRIS.2019.00028.

[19] Z. Ghorbani and A. H. Behzadan, “Monitoring Offshore Oil Pollution using Multi-Class Convolutional Neural Networks,” *Environ. Pollut.*, vol. 289, no. September 2020, p. 117884, 2021, doi: 10.1016/j.envpol.2021.117884.

[20] Q. Zhu, Y. Zhang, Z. Li, X. Yan, Q. Guan, Y. Zhong, L. Zhany and D. Li, “Oil Spill Contextual and Boundary-Supervised Detection Network Based on Marine SAR Images,” *IEEE Trans. Geosci. Remote Sens.*, vol. 60, no. December, 2022, doi: 10.1109/TGRS.2021.3115492.

[21] E. E. & E. Indiana Department of Education, “Overview of Unit 3 The Issue of Renewable Energy,” *Renew. Energy*.

[22] Michel, J. Fingas and Merv, “Oil Spills: Causes, Consequences, Prevention, and Countermeasures,” *World Scientific.,* pp. 159-201, 2016, doi: 10.1142/9789814699983_0007.

[23] S. T. Hussan, “Oil Spillage Problem and A Solution,” *Int. Conf. Mech. Ind. Mater. Eng.*, vol. 238, no. 2, pp. 1–7, 2019.

[24] Y. J. Yang, S. Singha, and R. Mayerle, “A Deep Learning Based Oil Spill Detector Using Sentinel-1 SAR Imagery,” *Int. J. Remote Sens.*, vol. 43, no. 11, pp. 4287–4314, 2022, doi: 10.1080/01431161.2022.2109445.

[25] D. S. Page, J. C. Foster, P. M. Fickett, and E. S. Gilfillan, “Long-term Weathering of Amoco Cadiz Oil in Soft Intertidal Sediments,” *2005 Int. Oil Spill Conf. IOSC 2005*, p. 4427, 2005, doi: 10.7901/2169-3358-1989-1-401.

[26] K. N. Topouzelis, “Oil Spill Detection by SAR Images: Dark Formation Detection, Feature Extraction and Classification Algorithms,” *Sensors*, vol. 8, no. 10, pp. 6642–6659, 2008, doi: 10.3390/s8106642.

[27] G. Orfanidis, K. Ioannidis, K. Avgerinakis, S. Vrochidis, and I. Kompatsiaris, “A Deep Neural Network for Oil Spill Semantic Segmentation in SAR Images,” Georgios Orfanidis , Konstantinos Ioannidis , Konstantinos Avgerinakis , Stefanos Vrochidis , Ioannis Kompatsiaris Centre for Research and Technology Hellas (CERTH) - Information Tec,” pp. 3773–3777, 2018.

[28] R. Al-Ruzouq, M. Gibri, S.A Barakat, A. Shanableh, A. Kais, O. Hamed, S. Al-Mansoori and M. A. Khalil. “Sensors, Features, and Machine Learning for Oil Spill Detection and Monitoring: A Review,” *Remote Sens.*, vol. 12, no. 20, pp. 1–42, 2020, doi: 10.3390/rs12203338.

[29] A. Doerry, “Introduction to Synthetic Aperture Radar,” *2019 IEEE Radar Conf. RadarConf 2019*, pp. 1–14, 2019, doi: 10.1109/RADAR.2019.8835560.

[30] T. R. Fuchi, L. Mokam, and T. Kombe, “Drone Equipment and Configuration for Crude Oil Spill Detection in Water,” *Eur. J. Technol.*, vol. 6, no. 3, pp. 1–14, 2022, doi: 10.47672/ejt.1127.

[31] M. Papoutsidakis, A. Chatzopoulos, and D. Piromalis, “UAV Flight Control Based on Arduino Board Implementations,” *Int. J. Eng. Appl. Sci. Technol.*, vol. 04, no. 05, pp. 438–443, 2019, doi: 10.33564/ijeast.2019.v04i05.064.

[32] G. Martinic, “A Practical Instruction Manual for Training Cadets,” no. April, 2016, doi: 10.13140/RG.2.2.33313.61286.

[33] S. Domaille and D. Campion, “Droning On : A Review of UAV Use in Recent Spills Attended by ITOPF and Considerations for the Future,” *Insterspill*, 2018, [Online]. Available: http://www.itopf.org/knowledge-resources/documents-guides/document/droning-on-a-review-of-uav-use-in-recent-spills-attended-by-itopf-and-considerations-for-the-future/.

[34] J. Heidari, “Classifying Material Defects with Convolutional Neural Networks and Image Processing, MSC Thesis,” 2019, [Online]. Available:http://urn.kb.se/resolve?urn=urn:nbn:se:uu:diva387797%0AR efrence66%0A.

[35] H. Hewamalage, C. Bergmeir, and K. Bandara, “Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions,” *Int. J. Forecast.*, vol. 37, no. 1, pp. 388–427, 2021, doi: 10.1016/j.ijforecast.2020.06.008.

[36] A. N. Jagannatha and H. Yu, “Structured Prediction Models for RNN Based Sequence Labeling in Clinical Text,” *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 856–865, 2016, doi: 10.18653/v1/d16-1082.

[37] R. Pascanu, C. Gulcehre, K. Cho, and Y. Bengio, “How to Construct Deep Recurrent Neural Networks,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Conf. Track Proc.*, pp. 1–13, 2014.

[38] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Commun. ACM*, vol. 60, no. 6, pp. 84–90, 2017, doi: 10.1145/3065386.

[39] Q. Zhang, M. Zhang, T. Chen, Z. Sun, Y. Ma, and B. Yu, “Recent Advances in Convolutional Neural Network Acceleration,” *Neurocomputing*, vol. 323, pp. 37–51, 2019, doi: 10.1016/j.neucom.2018.09.038.

[40] D. H. Hubel and T. N. Wiesel, “Receptive Fields, Binocular Interaction and Functional Architecture in the Cat’s Visual Cortex,” *J. Physiol.*, vol.

160, no. 1, pp. 106–154, 1962, doi: 10.1113/jphysiol.1962.sp006837.

[41] R. Won, "Deep-learning boost," *Nat. Photonics*, vol. 12, no. 8, p. 443, 2018, doi: 10.1038/s41566-018-0231-3.

[42] L. Alzubaidi *et al.*, "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions, " vol. 8, no. 1. Springer International Publishing, 2021, doi: 10.1186/s40537-021-00444-8.

[43] H. Wu, Q. Liu, and X. Liu, "A Review on Deep Learning Approaches to Image Classification and Object Segmentation," *Comput. Mater. Contin.*, vol. 60, no. 2, pp. 575–597, 2019, doi: 10.32604/cmc.2019.03595.

[44] F. Li, J. Johnson, and S. Yeung, "Lecture 11 : Detection and Segmentation," 2017.

[45] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image Segmentation Using Deep Learning: A Survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 7. pp. 3523–3542, 2022, doi: 10.1109/TPAMI.2021.3059968.

[46] E. K. Wang, C. M. Chen, M. M. Hassan, and A. Almogren, "A Deep Learning Based Medical Image Segmentation Technique in Internet-Of-Medical-Things Domain," *Futur. Gener. Comput. Syst.*, vol. 108, pp. 135–144, 2020, doi: 10.1016/j.future.2020.02.054.

[47] S. Patel, "Deep learning Models for Image Segmentation," *Proc. 2021 8th Int. Conf. Comput. Sustain. Glob. Dev. INDIACom 2021*, no. August, pp. 149–154, 2021, doi:10.1109/INDIACom51348.2021.00027.

[48] O. Elharrouss, Y. Akbari, N. Almaadeed, and S. Al-Maadeed, "Backbones-Review: Feature Extraction Networks for Deep Learning and Deep Reinforcement Learning Approaches," 2022, [Online]. Available: http://arxiv.org/abs/2206.08016.

[49] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and Transferring Mid-Level Image Representations using Convolutional Neural

Networks," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1717–1724, 2014, doi: 10.1109/CVPR.2014.222.

[50] L. Zhou, C. Zhang, and M. Wu, "D-linknet: Linknet with Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction," *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2018-June, pp. 192–196, 2018, doi: 10.1109/CVPRW.2018.00034.

[51] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the Classification Effectiveness of Intrusion Detection by using Improved Conditional Variational Autoencoder and Deep Neural Network," *Sensors (Switzerland)*, vol. 19, no. 11, 2019, doi: 10.3390/s19112528.

[52] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation Functions: Comparison of Trends in Practice and Research for Deep Learning," pp. 1–20, 2018, [Online]. Available: http://arxiv.org/abs/1811.03378.

[53] M. Wang, S. Lu, D. Zhu, J. Lin, and Z. Wang, "A High-Speed and Low-Complexity Architecture for Softmax Function in Deep Learning," *2018 IEEE Asia Pacific Conf. Circuits Syst. APCCAS 2018*, no. June 2020, pp. 223–226, 2019, doi: 10.1109/APCCAS.2018.8605654.

[54] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks," *J. Phys. Conf. Ser.*, vol. 1237, no. 2, pp. 111–122, 2019, doi: 10.1088/1742-6596/1237/2/022030.

[55] N. K. Manaswi, "Deep Learning with Applications Using Python: Chatbots and Face, Object, and Speech Recognition With TensorFlow and Keras," *Springer,* 2018.

[56] R. G, "A Study to Find Facts Behind Preprocessing on Deep Learning Algorithms," *J. Innov. Image Process.*, vol. 3, no. 1, pp. 66–74, 2021, doi: 10.36548/jiip.2021.1.006.

[57] S. R. Nayak, D. R. Nayak, U. Sinha, V. Arora, and R. B. Pachori,

"Application of Deep Learning Techniques for Detection of COVID-19 Cases using Chest X-ray Images: A Comprehensive Study," *Biomed. Signal Process. Control*, vol. 64, no. June 2020, 2021, doi: 10.1016/j.bspc.2020.102365.

[58] M. Nisa, H. J. Shah, S. Kanwal, M. Raza and D. Attique, "Hybrid Malware Classification Method using Segmentation-based Fractal Texture Analysis and Deep Convolution Neural Network Features," *Appl. Sci.*, vol. 10, no. 14, 2020, doi: 10.3390/app10144966.

[59] C. Shorten and T. M. Khoshgoftaar, "A Survey on Image Data Augmentation for Deep Learning," *Journal of Big Data*, vol. 6, no. 1. 2019, doi: 10.1186/s40537-019-0197-0.

[60] I. Kandel, M. Castelli, and L. Manzoni, "Brightness as an Augmentation Technique for Image Classification," *Emerg. Sci. J.*, vol. 6, no. 4, pp. 881–892, 2022, doi: 10.28991/ESJ-2022-06-04-015.

[61] D. P. Timothy and A. K. Santra, "A Hybrid Cryptography Algorithm for Cloud Computing Security," *2017 International Conference on Microelectronic Devices, Circuits and Systems, ICMDCS 2017*, vol. 2017-Janua. pp. 1–5, 2017, doi: 10.1109/ICMDCS.2017.8211728.

[62] M. I. Malik, "Cloud Computing-Technologies," *Int. J. Adv. Res. Comput. Sci.*, vol. 9, no. 2, pp. 379–384, 2018, doi: 10.26483/ijarcs.v9i2.5760.

[63] J. Sen, "Security and Privacy Issues in Cloud Computing," *Archit. Protoc. Secur. Inf. Technol. Infrastructures*, no. iv, pp. 1–45, 2013, doi: 10.4018/978-1-4666-4514-1.ch001.

[64] B. Calabrese and M. Cannataro, "Cloud Computing in Healthcare and Biomedicine," *Scalable Comput.*, vol. 16, no. 1, pp. 1–18, 2015, doi: 10.12694/scpe.v16i1.1057.

[65] I. M. Khalil, A. Khreishah, and M. Azeem, "Cloud Computing Security: A Survey," *Computers*, vol. 3, no. 1, pp. 1–35, 2014, doi: 10.3390/computers3010001.

[66] G. Olmedo, R. Lara-Cueva, D. Martínez, and C. de Almeida, “Performance Analysis of a Novel TCP Protocol Algorithm Adapted to Wireless Networks,” *Futur. Internet*, vol. 12, no. 6, 2020, doi: 10.3390/fi12060101.

[67] S. Kam and A. Takagi, “File Transfer And Access (FTP , TFTP , NFS) History of FTP.”

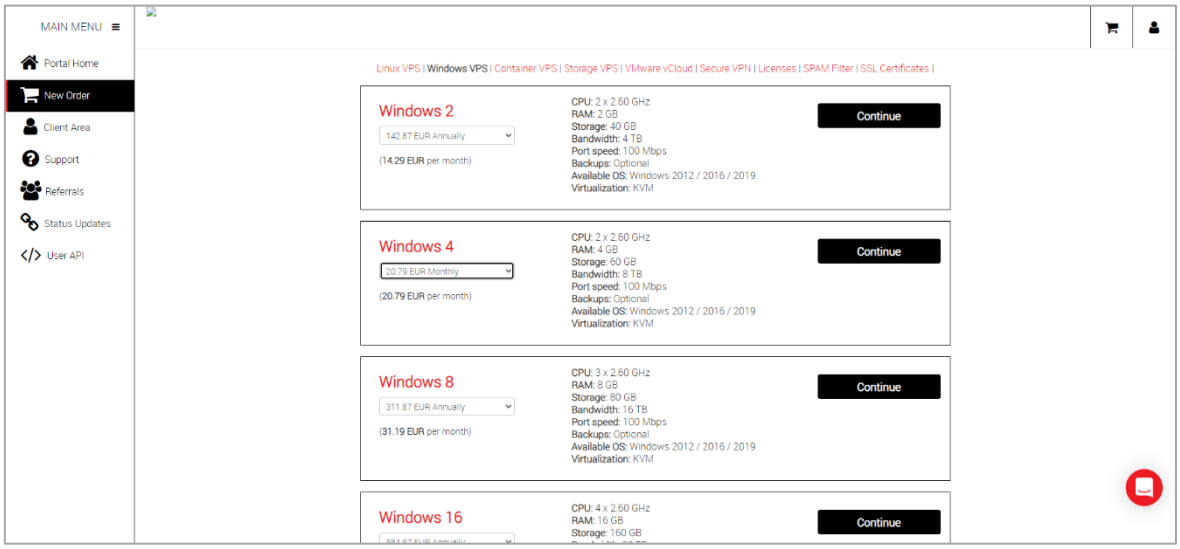
[68] X.-L. ZHANG, X.-Y. WU, and W.-X. ZHANG, “Research and Implementation of RDP Proxy Proxy-based Audit System,” vol. 54, pp. 214–220, 2017, doi: 10.2991/cnct-16.2017.29.

[69] J. Miao and W. Zhu, “Precision–Recall Curve (PRC) Classification Trees,” *Evol. Intell.*, no. December, 2021, doi: 10.1007/s12065-021-00565-2.

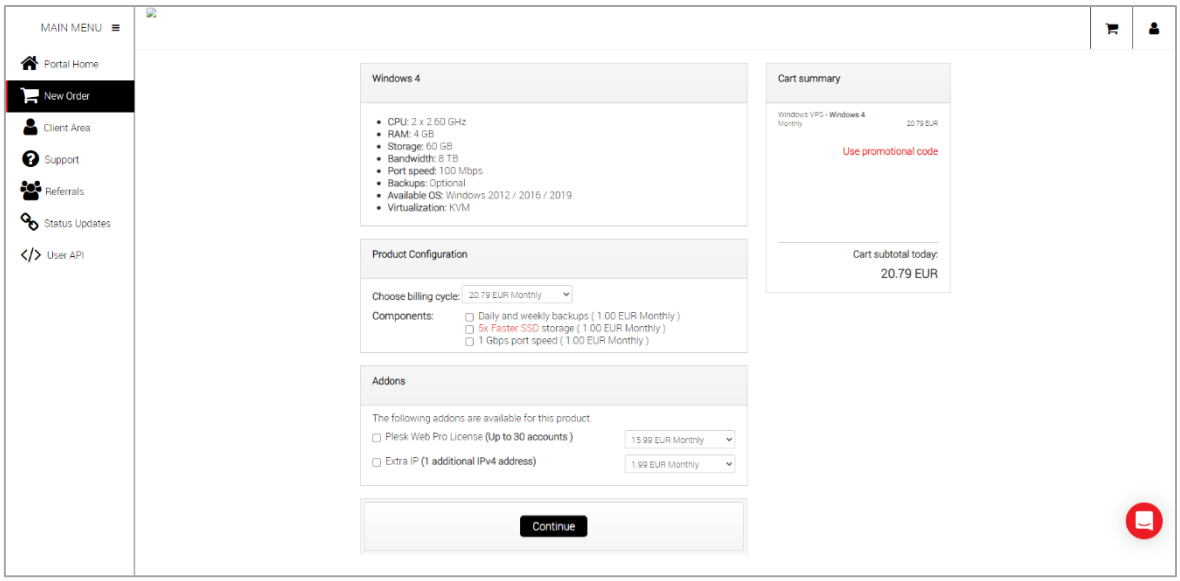
[70] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview,” pp. 1–17, 2020, [Online]. Available: http://arxiv.org/abs/2008.05756.

Appendix

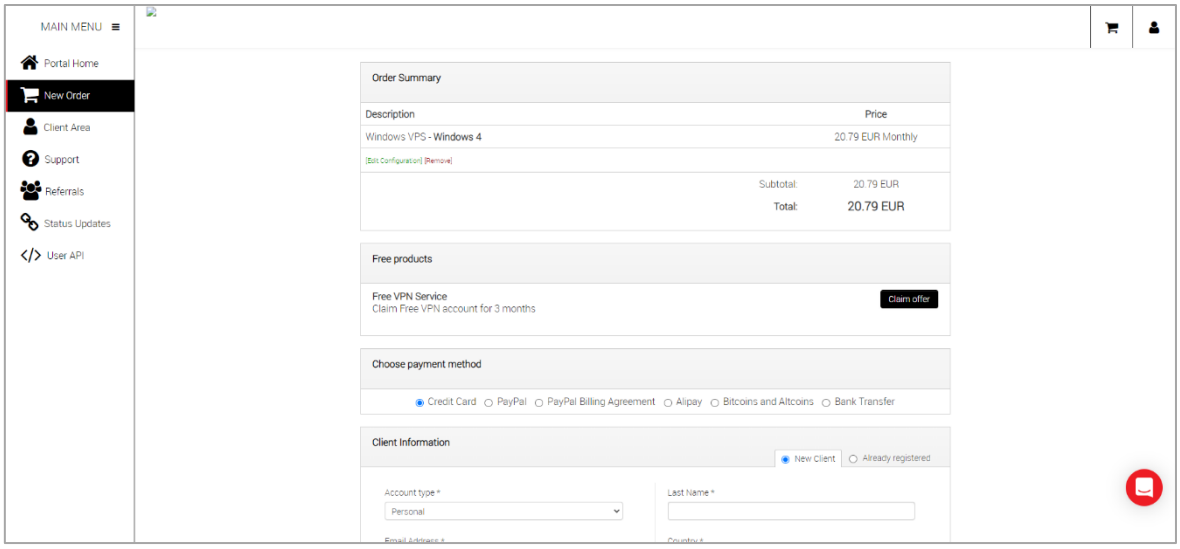
1. There are many options VPS as show in Figure:



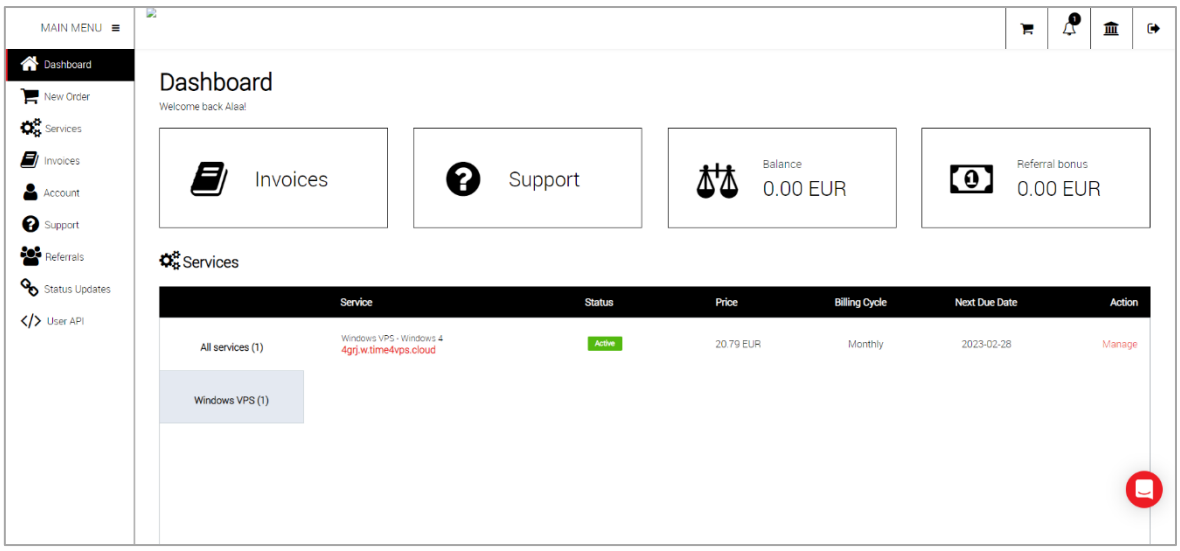
2. Windows 4 with (CPU 2 - RAM 4GB – Storage 60GB) select to deploy model on it:



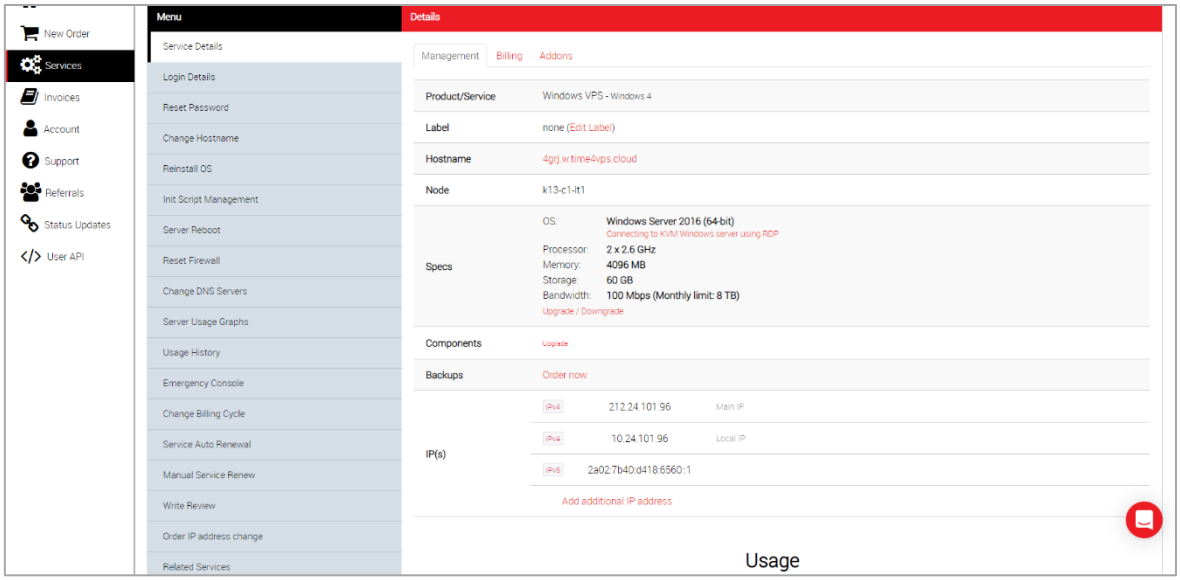
3. The price for single month is 20.79 EUR



4. After payment complete the VPS active and windows OS server 2016 install on it.

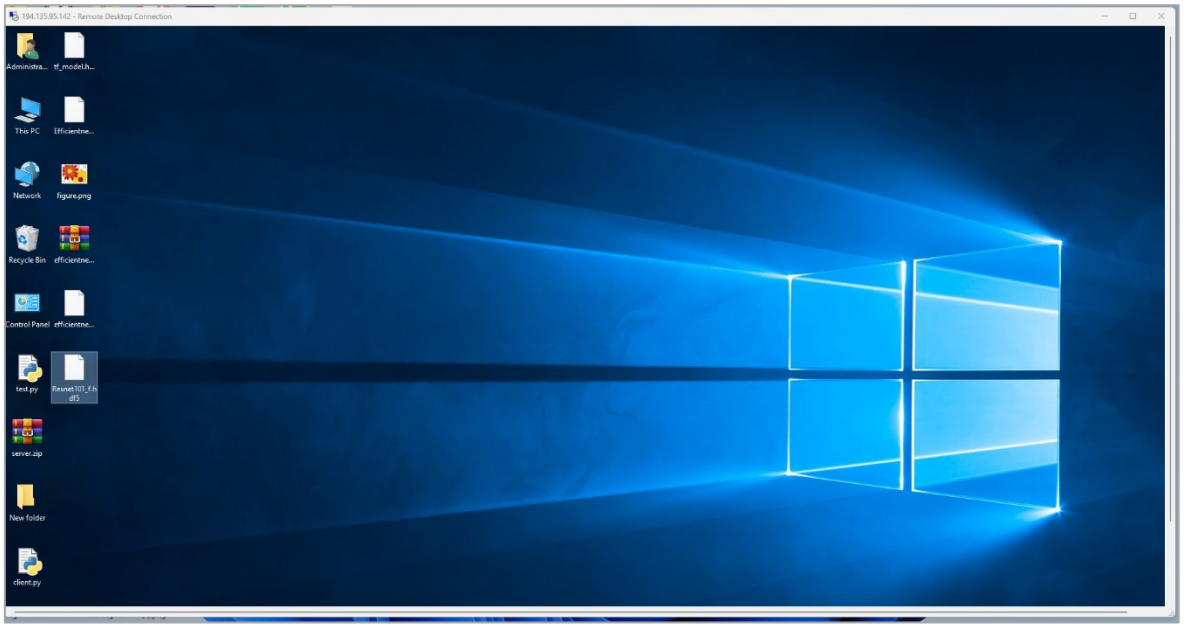


5. When go to details all information show like OS type and public IP address

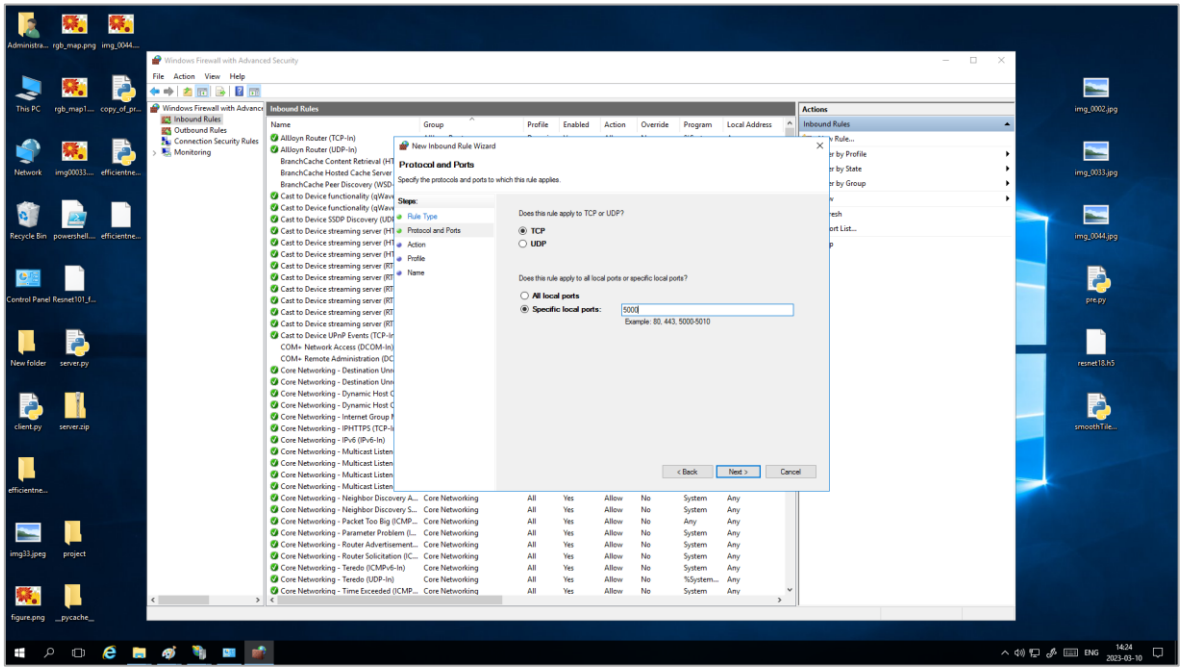


6. Remote Desktop Connection (RDP) use to connect with server

7. When VPS open important libraryes install like (tensorflow, Keras, C#) and Demo.h5 file upload.

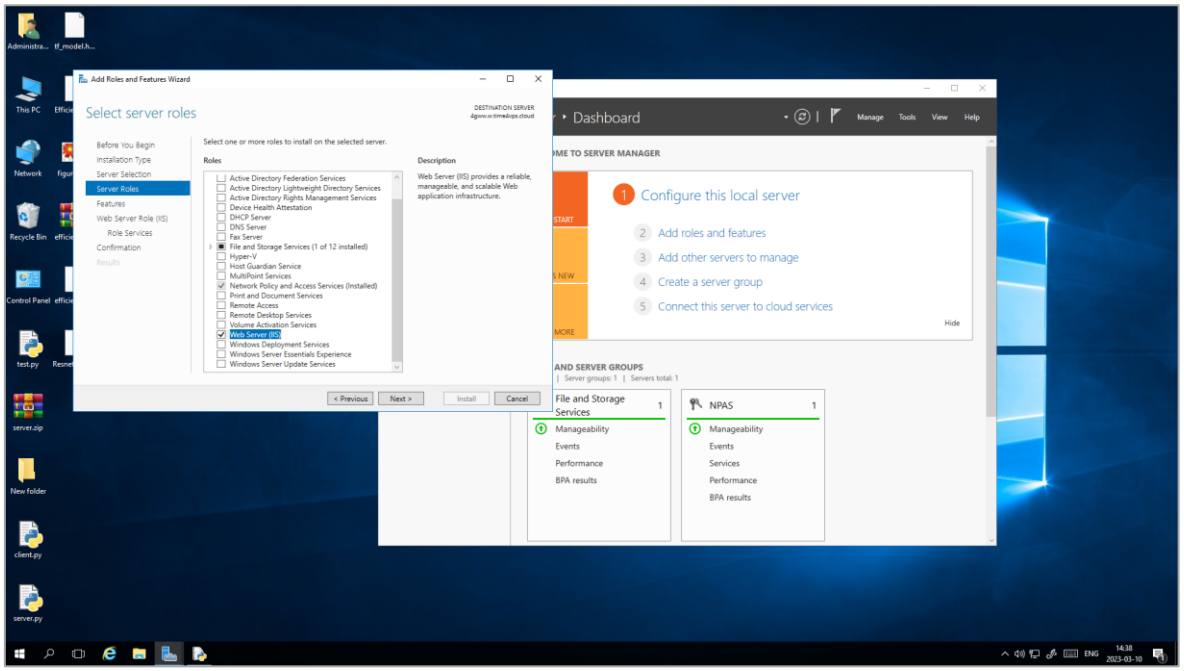


8. From inbound rules select port 5000 to enable client access to server.

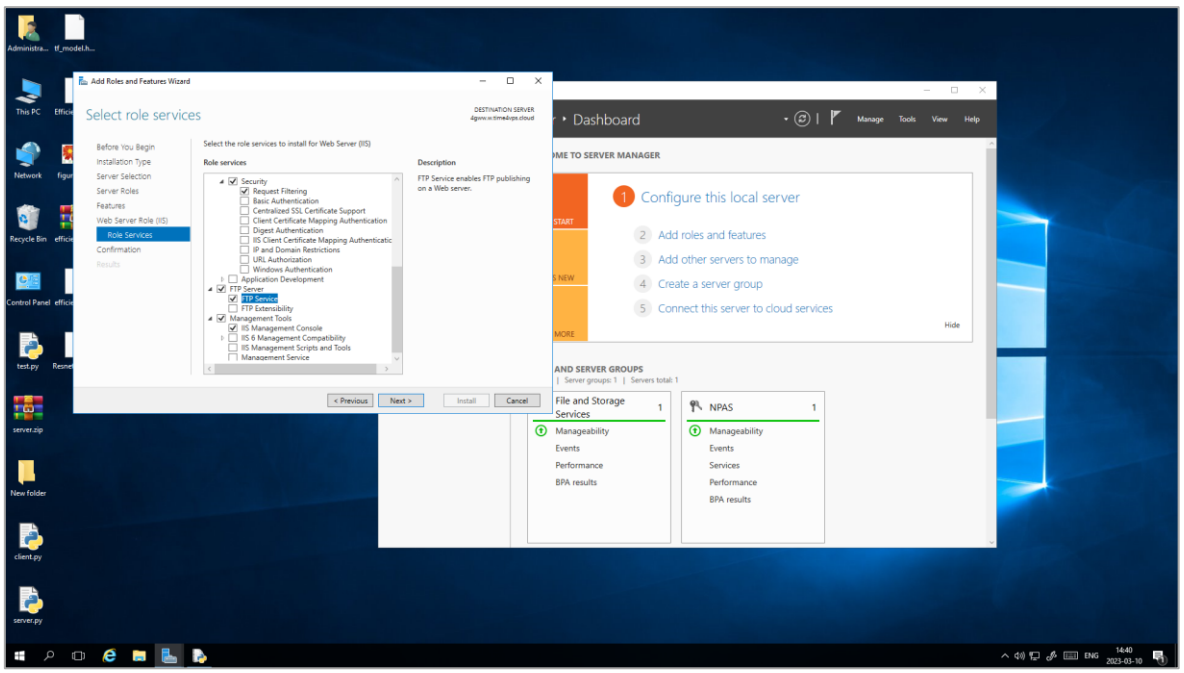


One of the challenge when deploy the models are how to transforme the image without resize. In this thesis FTP use for that task.

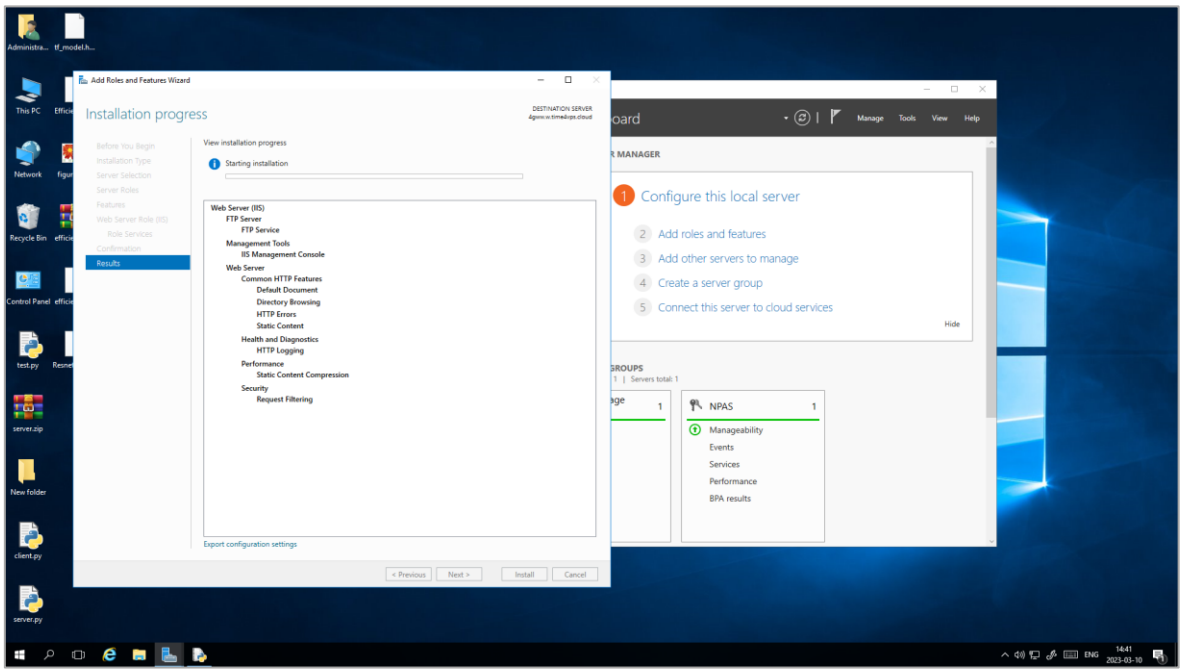
1. From server manager in VPS must install FTP feature.



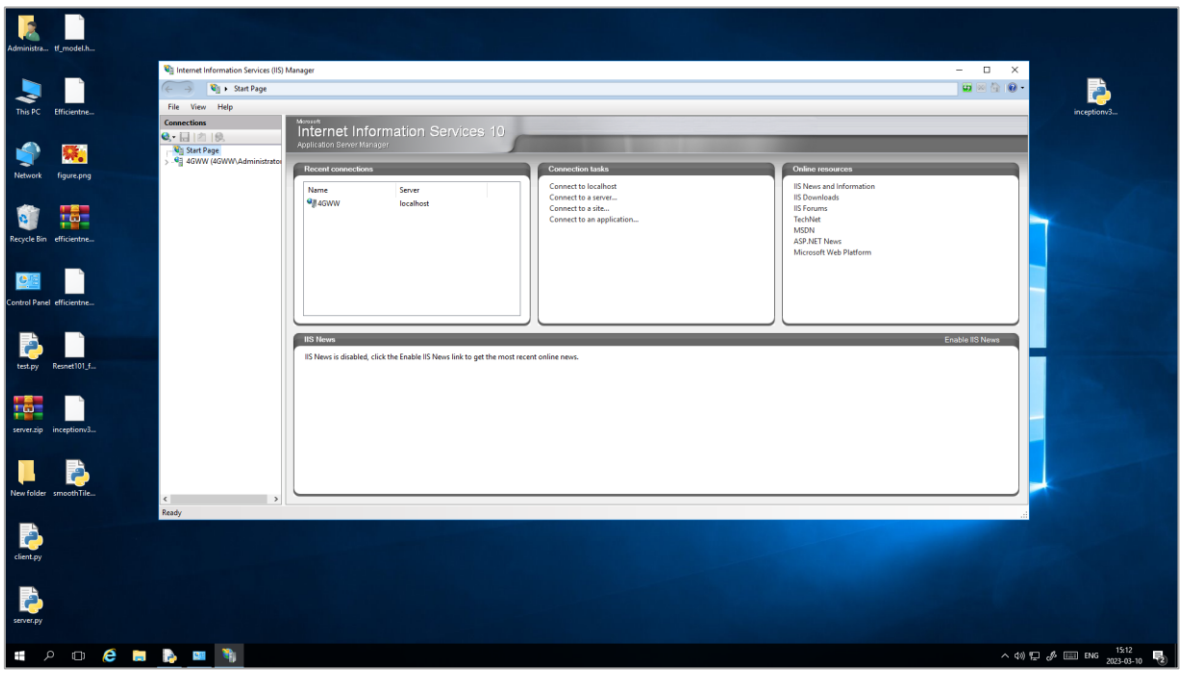
2. From Role services select FTP service



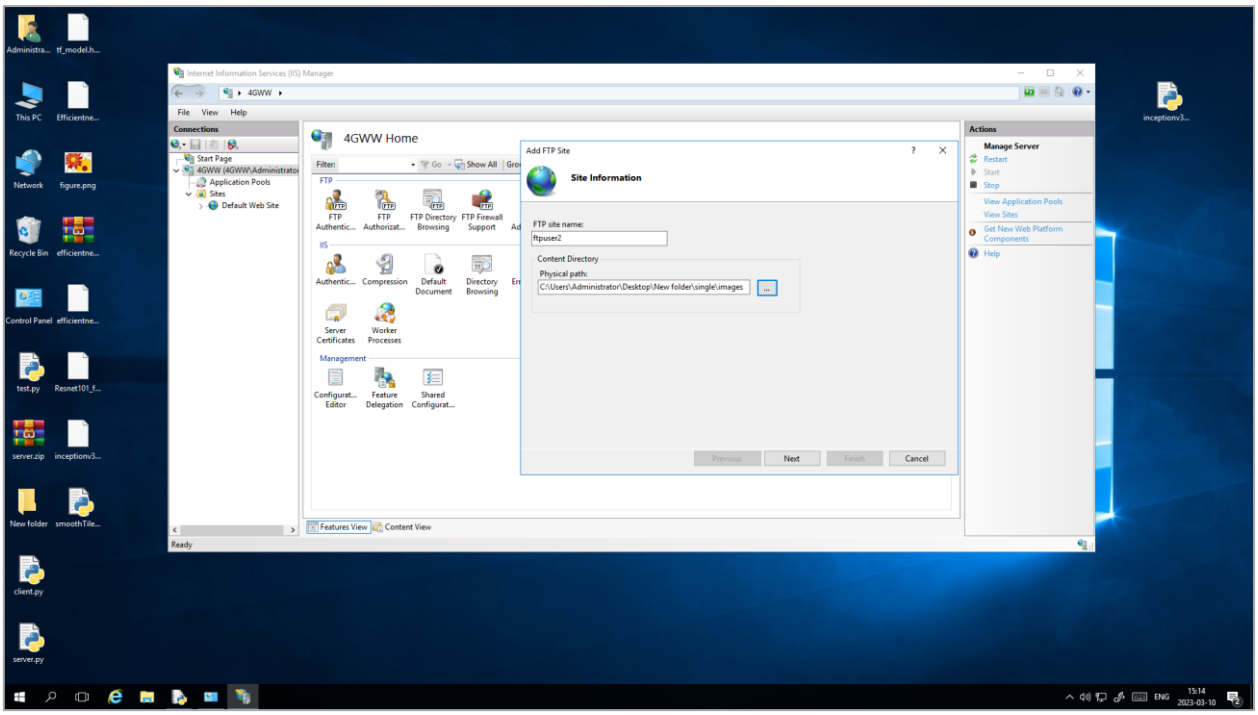
3. Install feature



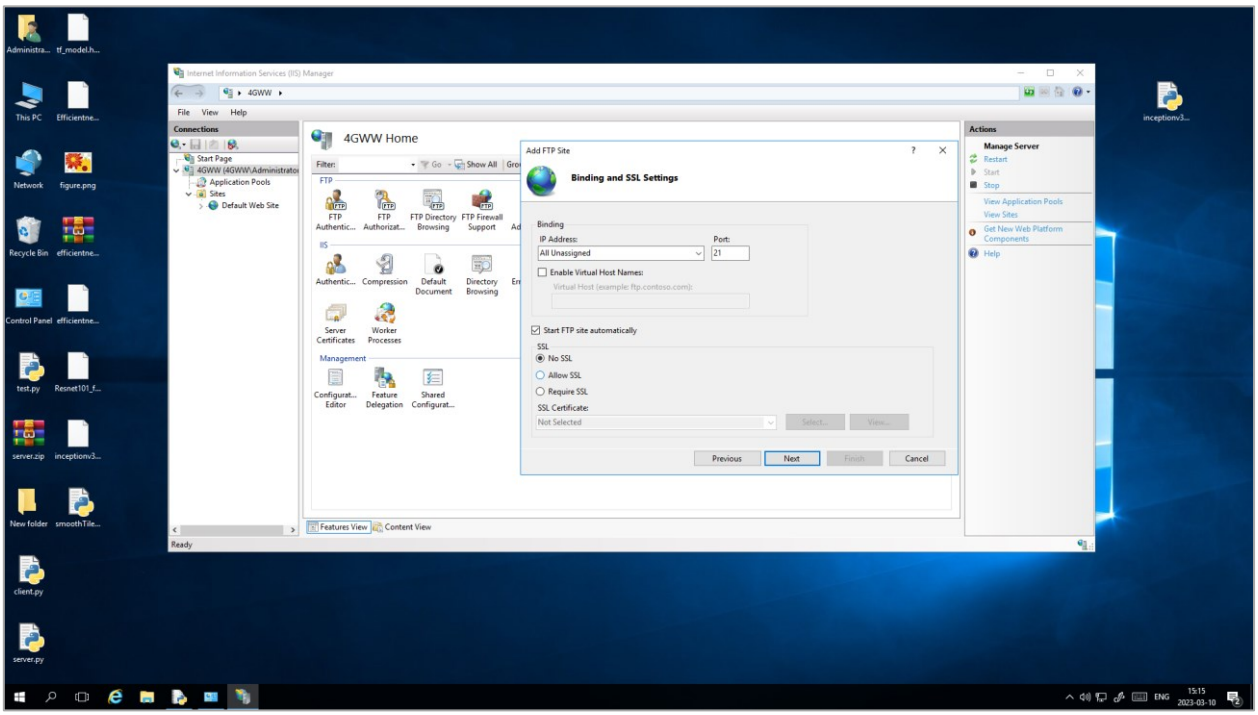
4. Setup FTP



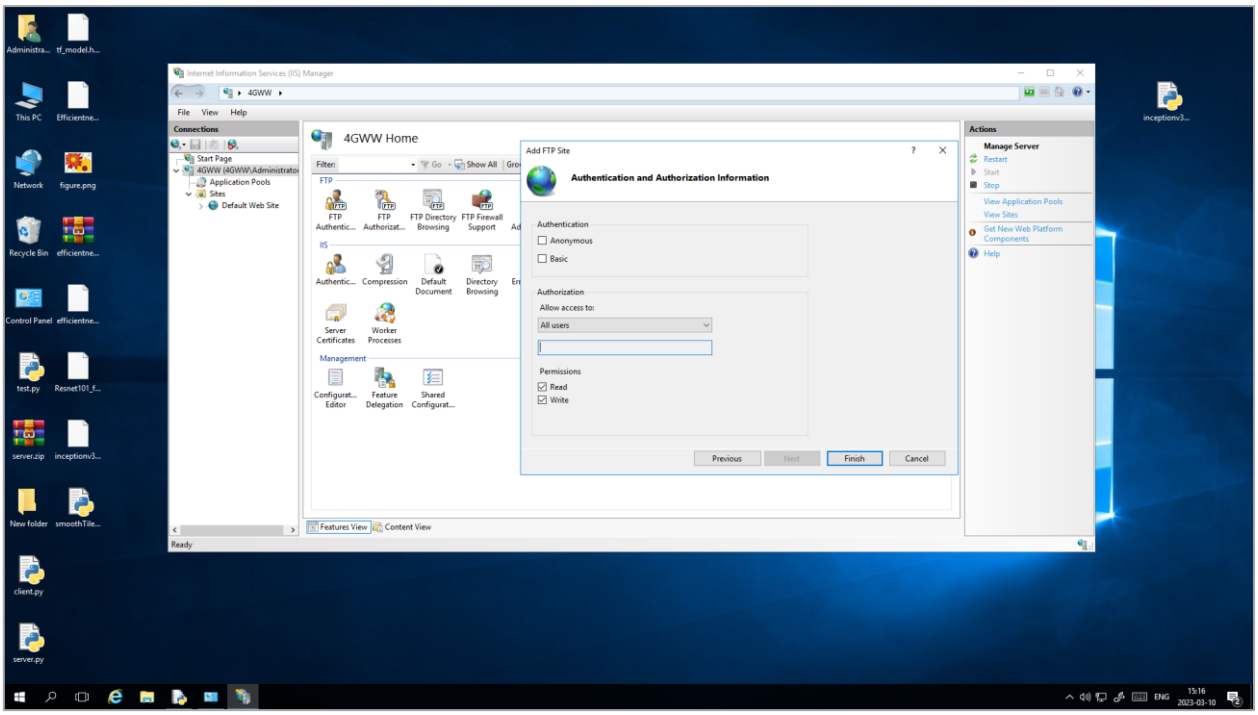
5. Select new ftp site



6. Select port 21



7. Select how can access



جمهورية العراق
وزارة التعليم العالي والبحث العلمي
جامعة بابل-كلية تكنولوجيا المعلومات
قسم شبكات المعلومات

الكشف عن تسرب النفط البحري باستخدام التعلم العميق داخل البيئة السحابية

رسالة

مقدمة إلى مجلس كلية تكنولوجيا المعلومات في جامعة بابل والتي هي جزء من متطلبات الحصول

شبكات المعلومات/على درجة الماجستير في تكنولوجيا المعلومات

آلاء اكرم جواد محمد حبي

بأشراف

أ.م.د رافد صكبان عبود كريم الجبوري

أ.م.د رائد نصر كاظم خليل العبيدي

1444هـ 2023 م

خلاصة

ان تسرب النفط على الأسطح المائية من ناقلات النفط الكبيرة والسفن اوشقوق خطوط الأنابيب يسبب حدوث أضرار جسيمة بالبيئة البحرية. توفر صور الرادار ذي الفتحة التركيبية (SAR) تمثيلًا تقريبيًا للمشاهد المستهدفة، بما في ذلك الأسطح البحرية والأرضية والسفن وانسكابات النفط والمظاهر المتشابه. يعد اكتشاف الانسكابات النفطية من عن طريق الصور وتقسيمها أمرًا بالغ الأهمية لعمليات اكتشاف ومعالجة التسرب وبالتالي حماية البيئة من هذه المخاطر. تقدم هذه الرسالة إطارًا للتعلم العميق للكشف عن تسرب النفط باستخدام تقنية التجزئة U-Net. وقد تم استبدال جزء التشفير من U-Net بنماذج مثل (Densnet201, Inception V3, EfficientNetb3 ReseNet101) التي تم تدريبها مسبقًا على مجموعة بيانات Imagnet والتي تمثل جزء Encoder في المقابل يستخدم جزء Decoder بنية U-Net. تقسم فئات مجموعة البيانات إلى خمس مجموعات ذات أبعاد مكانية 512 × 512 (800 للتدريب و 202 للتقييم و 110 للاختبار). قدمت U-net مع العمود الفقري Densenet201 افضل نتائج (دقة 92٪ ، دقة 56٪ ، استرجاع 80٪ ، 74٪ درجة F ، و 69٪ IoU). علاوة على ذلك يتم حفظ أفضل النماذج أداءً وتحميلها إلى VPS للاختبار على صورة حقيقية بالحجم الكامل (1250 * 650) وحساب مساحة البكسل لكل فئة. نتائج هذه الدراسة واعدة للغاية وتوفر IoU محسّنًا مقارنة بالأعمال ذات الصلة.