Project report on

Deep Learning Approach to Detect Ocean Oil Spills using SAR Images

Submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

in

Computer Science and Engineering

By

Niranjan S (U2103158)

Saurav Krishnan (U2103191)

Sradha Shajan (U2103201)

Wivin Winny (U2103216)

Under the guidance of

Dr. Preetha K.G

Department of Computer Science and Engineering
Rajagiri School of Engineering & Technology (Autonomous)
(Parent University: APJ Abdul Kalam Technological University)
Rajagiri Valley, Kakkanad, Kochi, 682039

April 2025

CERTIFICATE

This is to certify that the project report entitled ***"Deep learning approach to detect ocean oil spills using SAR images"*** *is a bonafide record of the work done by* ***Niranjan S (U2103158)****,* ***Saurav Krishnan (U2103191)****,* ***Sradha Shajan (U2103201)****,* ***Wivin Winny (U2103216)*** *submitted to the Rajagiri School of Engineering & Technology (RSET) (Autonomous) in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology (B. Tech.) in "Computer Science and Engineering during the academic year 2024-2025.*

Dr. Preetha K.G
Project Guide
Professor & HOD
Dept. of CSE
RSET

Ms. Sangeetha Jamal
Project Co-ordinator
Assistant Professor
Dept. of CSE
RSET

Dr. Preetha K.G
Professor & HOD
Dept. of CSE
RSET

ACKNOWLEDGMENT

We wish to express our sincere gratitude towards **Rev. Dr. Jaison Paul Mulerikkal CMI**, Principal of RSET, and **Dr. Preetha KG**, Head of the Department of Computer Science and Engineering for providing us with the opportunity to undertake our project, "Deep learning approach for detecting ocean oil spills from SAR images".

We are highly indebted to our project coordinators, **Ms. Sangeetha Jamal**, Assistant Professor, Department of Computer Science and Engineering and **Dr. Preetha K.G** , Professor & HOD, Department of Computer Science and Engineering for their valuable support.

It is indeed our pleasure and a moment of satisfaction for us to express our sincere gratitude to our project guide **Dr. Preetha K.G** for her patience and all the priceless advice and wisdom she has shared with us.

Last but not the least, We would like to express our sincere gratitude towards all other teachers and friends for their continuous support and constructive ideas.

Niranjan S
Saurav Krishnan
Sradha Shajan
Wivin Winny

Abstract

Ocean ecosystems are extremely vulnerable to oil spills, necessitating the use of accurate and efficient detection techniques. A deep learning-based model for the autonomous identification of oil spills in the ocean using Synthetic Aperture Radar (SAR) pictures is proposed in this research. Using Precision, Recall, and F1-Score as performance metrics, We compare three models: ResNet50, RCNN with VGG16 backbone, and RCNN with InceptionV3 backbone. 5,630 SAR images were preprocessed using greyscale conversion, normalisation, histogram equalisation and data augmentation to improve the robustness of the model. The data set was gathered from CSIRO (Commonwealth Scientific and Industrial Research Organisation).

According to our research, out of all the models, RCNN with VGG16 has the best precision and F1-Score for detecting oil spills with the fewest false positives. Our deep learning models exhibit superior accuracy and responsiveness when compared to traditional machine learning techniques like Random Forest (RF) and Support Vector Machines (SVM). By providing an automatic, interpretable, and effective oil spill detection approach, our work contributes to real-time crisis management. Future research will focus on expanding model generalisation with larger datasets and using explainability techniques to improve decision-making transparency.

Contents

List of Abbreviations

AI - Artificial Intelligence

ASPP - Atrous Spatial Pyramid Pooling

AUC - Area Under the Curve

CNN - Convolutional Neural Network

CSIRO - Commonwealth Scientific and Industrial Research Organisation

DCNN - Deep Convolutional Neural Network

DL - Deep Learning

ESA - European Space Agency

FCN - Fully Convolutional Network

GAP - Global Average Pooling

GDL - General Dice Loss

GIS - Geographic Information System

IoU - Intersection over Union

NMS - Non-Maximum Suppression

OA - Overall Accuracy

OFCN - Oil Fully Convolutional Network

OSCN - Orientation-Shared Convolutional Network

PR curve - Precision Recall curve

R-CNN - Region-based Convolutional Neural Network

ResNet - Residual Neural Network

RNN - Recurrent Neural Network

ROC - Receiver Operating Characteristic

RoI - Region Of Interest

RPN - Region Proposal Network

RS - Remote Sensing

SAR - Synthetic Aperture Radar

SegNet - Segmented Neural Network

SGD - Stochastic Gradient Descent

SVM - Support Vector Machine

SWIR - Shortwave Infrared

UAV - Unmanned Aerial Vehicle

VGG - Visual Geometry Group

VV - Vertical Transmit-Vertical Receive Polarisation

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

This study presents a novel deep learning approach for identifying oil spills and emergencies in SAR data. Oil spill detection is essential for marine habitats, environmental monitoring, and maritime security, and SAR imaging is a reliable technique because it can function in any cloud and weather circumstances. Conventional detection methods frequently have high false positive rates and poor generalisation over a wide range of settings. This study explores deep learning-based techniques to identify oil spills in SAR data in order to get over these restrictions by using convolutional neural networks (CNNs) to understand complex spatial patterns, intensity changes, and texture changes.This study evaluates three deep learning models: R-CNN with VGG-16 backbone, R-CNN with Inception V3 backbone, and ResNet-50. In a supervised learning setting, they were trained on labelled SAR datasets, and their F1-score, accuracy, and precision were evaluated. By developing an even more automatic and precise detection method that facilitates the real-time identification of spills and reduces their impact on marine ecosystems, our work contributes to better environmental monitoring.

1.2 Problem Definition

The project's goal is to create a reliable deep learning-based technique for detecting oil spills using SAR pictures. Since traditional approaches often have significant false positives and low generalisation, automated detection is required. This study compares ResNet-50, R-CNN with VGG-16, and R-CNN with Inception V3 for oil spill detection based on important performance parameters. The goal is to improve environmental monitoring and response efforts by finding the optimal model to match large-scale real-time detection.

1.3 Scope and Motivation

Developing an autonomous deep learning system that can recognise oil spills in SAR photos is the goal of this study. It evaluates the accuracy, precision, and F1-score of the three deep learning models—ResNet-50, R-CNN using VGG-16, and R-CNN using Inception V3. The study aims to increase detection reliability, reduce false positives, and enhance generalisation from different water environments. Because the system uses SAR imaging, it can operate in any weather, making it suitable for real-time marine surveillance. The findings can help with further environmental monitoring and prompt reactions to oil spills.

This endeavour was spurred by the severe economic and environmental effects of oil spills, including the destruction of marine habitats and coastal towns. Despite best efforts, the present detection technologies' frequent unreliability and laborious manual intervention hinder response. Deep learning provides a desirable option that streamlines detection into a speedier and more dependable procedure by making it possible to recognise distinctive patterns in oil spills. By identifying oil spills more thoroughly, this endeavour reduces the long-term impacts of oil contamination and protects marine areas. International environmental protection and marine security protocols could both be strengthened by the installation of a functional automation system.

1.4 Objectives

1. To automatically identify ocean oil spills from SAR photos by designing, implementing, and evaluating deep learning models.

2. To use crucial evaluation criteria to compare the performance of three models: ResNet50, RCNN with VGG16 backbone, and RCNN with InceptionV3 backbone.

3. Greyscale conversion, speckle noise reduction, normalisation, histogram equalisation, binary masking, and data augmentation are effective preprocessing techniques for SAR images that enhance model performance.

4. Using F1-Score, Precision, and Recall to determine the best performing model for precise oil spill detection.

5. To combine explainability techniques for transparent decision-making in order to make the deep learning models easier to understand.

6. To develop a scalable, effective, and automated framework for oil leak detection in order to facilitate real-time crisis management.

1.5 Challenges

The project's biggest issue was controlling noise and unpredictability in SAR images since imaging artefacts and ocean surface conditions can result in false positives when signalling oil spills. Additionally, because deep learning models were trained on a small dataset of 5,630 pictures, massive quantities of data augmentation were required to improve generalisation. The second goal was to balance computational frugality with model performance in order to deploy the high-performing model for online monitoring without being overly resource-demanding.

1.6 Assumptions

1. Oil spills are reliably captured by SAR pictures, and reliable ground truth labels are provided by the CSIRO dataset.

2. Preprocessing methods successfully improve image quality, denoising and enhancing feature extraction for deep learning algorithms.

3. Oil spills may be distinguished from other marine elements using the chosen deep learning architectures (ResNet50, RCNN with VGG16, and RCNN with InceptionV3).

4. The model's performance for oil spill detection can be determined using evaluation criteria such as precision, recall, and F1-score.

5. When applied to unseen SAR images outside of the training dataset, the trained models perform admirably.

1.7 Societal / Industrial Relevance

Finding oil spills in the ocean has significant effects on the environment, society, and industry. This project provides environmental authorities with an automated and dependable monitoring system for real-time oil spill identification, enabling prompt action and minimising ecological damage. By enabling prompt mitigation actions, early oil spill detection helps shield the petroleum and maritime sectors from financial losses, fines from the authorities, and damage to their reputation. Better spill management also benefits fisheries and coastal populations, reducing the long-term impact on marine ecosystems and livelihoods.

This project contributes to the development of a robust, scalable, and cost-effective oil spill detection framework by applying deep learning to SAR images. This framework can be used in marine control centres, UAV surveillance, and satellite-based monitoring systems for environmental protection worldwide.

1.8 Organization of the Report

This report is structured into seven chapters, each detailing a critical aspect of the research:

- Chapter 1: Introduction provides a summary of the research, outlining its goals and obstacles as well as the importance of using deep learning to detect oil spills using SAR photos.
- Chapter 2: Literature Survey highlights research gaps and compares deep learning with traditional machine learning techniques to illustrate earlier findings on oil spill detection.
- Chapter 3: Requirements outlines the hardware and software requirements, such as compute power, development software, and dataset needs, that are necessary to complete the project.
- Chapter 4: System Architecture, which also includes workflow charts, model architectures (ResNet50, RCNN with VGG16, and RCNN with InceptionV3), and data preprocessing activities

- Chapter 5: System Implementation describes how the models are actually put into practice, including preprocessing the dataset, training the model, optimising the hyperparameters, and evaluation techniques.
- Chapter 6: Findings and Discussion, which also compares the model's performances on F1-Score, Precision, and Recall and discusses its advantages and disadvantages.
- Chapter 7: Conclusion and Future Scope outlines the main conclusions, selects the top-performing model, and talks about possible advancements in deep learning-based oil spill detection in the future. .

This chapter provided a summary of the report's format, including the key sections that guide the reader through the investigation. The technical specifics, experimental findings, and broad ramifications of this investigation are examined in the ensuing chapters.

Chapter 2

Literature Survey

2.1 Oil spill classification based on satellite image using deep learning techniques

2.1.1 Introduction

In the past few years, deep learning has shown progress, especially in image classification and multi-pattern detection. At the moment, the network As ResNet has the ability to learn complex model features in the network space, it has evolved as a practical tool for the processing of intricate image data for applications. Specifically, deep learning models can find application in satellite surveillance. For instance, ResNet is capable of analyzing tremendous amounts of oil spill data such as those reported to satellites, and employing powerful features for vital image research. Resnet extract abilities can scatter satellite images into various locations victimized by oil spills in different regions through continued methods that effectively perform well performance model so classification accuracy is maximized.

2.1.2 Methodology

1. **Dataset Preparation**: Data augmentation techniques such as rotation, noise reduction, and linear horizontal and vertical flipping were used to prepare the dataset using the material clear satellite images with various terrains, water desert green zone, and sky data. All images were resized to 224x224 pixels, and the values were comparable.

2. **Model Architecture**: It solves the vanishing gradient problem by using residual connections across 50 layers using the ResNet50 structure. By assisting the network in learning residual functions, residual blocks enhance the model's performance for

deep structures.

3. **Training Configuration**: The dataset was used with batch size 32, Adam Optimiser learning rate 0.00001, to train the extension model 70% and 30% for validation. Categorical cross entropy can be a legitimate loss function for the majority of distribution function types.

4. **Model Optimization**: By combining the early stopping layer and release mechanism to avoid overfitting, model optimisation balances training time and performance. For ten cycles, the model is visible.

5. **Performance Metrics**: The performance model employs metrics such as accuracy precision recall and f1 score to distinguish between oil spill and no spill oil spill categorisation accuracy.

2.1.3 Results

ResNet50 model performed well in classifying satellite photos with different contours after ten training epochs. With a score of 91, the reference model was the most accurate model in dry and climatic zones. Compared to distinguishing between green patches of water bodies based on colour and texture similarities in satellite images, the model is more accurate at distinguishing between green patches of water bodies and non-water surfaces. However, certain misclassifications resulted from spectral overlap, particularly between green spaces and aquatic bodies. Regardless of how deep the model is, accuracy is ensured by utilising ResNet50's connections.

2.2 A Novel Deep Learning Method for Marine Oil Spill Detection from Satellite1 Synthetic Aperture Radar Imagery

2.2.1 Introduction

Oil seepage from maritime activities, pipeline spills, and natural seepage all pose a major hazard to marine ecosystems and necessitate effective detection and monitoring techniques. Due to its capacity to image in both day and nighttime settings and independent of the weather, Synthetic Aperture Radar (SAR) has become a vital instrument for oil spill observation. However, the following are the disadvantages of the conventional SAR-based detection methods: low detection rates, imbalanced datasets, complex algorithms, and an inability to differentiate oil spills from other comparable occurrences. In order to overcome the limitations, the current study introduces a novel deep learning technique for real-time oil spill identification in SAR images utilising the Faster Region-based Convolutional Neural Network (Faster R-CNN). A sizable dataset of 15,774 labelled Sentinel-1 and RADARSAT-2 SAR image oil spill samples is used to train, validate, and test the model.

2.2.2 Methodology

1. **Dataset Preparation**: 1,786 Sentinel-1A/B and RADARSAT-2 SAR pictures were pre-processed using normalisation, masking of land areas, incidence angle correction, and speckle noise removal. Areas affected by oil spills were manually annotated and verified. For model generalisation, 8,576 photos were added to the training set via flipping and rotation.

2. **Model Architecture**: For effective oil spill detection, Faster R-CNN combines Fast R-CNN with a Region Proposal Network (RPN). VGG-16 extracts feature maps, RPN creates bounding boxes, and classification/regression refines detection. Using Non-Maximum Suppression (NMS), duplicate boxes are removed. By optimising performance through joint training, runtime can be reduced by 25–50%.

3. **Training Configuration**: Joint training of Fast R-CNN and RPN is used to train the model, halving the training time to 25–50%. Regression loss and classification are employed for precision, while non-maximum suppression is used to improve detection. To efficiently detect oil spills, a 0.0001 learning rate, 100 epochs, and data augmentation speed up convergence, accuracy, and generalisation.

4. **Model Optimization**: Training RPN and Fast R-CNN together optimises the model, resulting in a 25–50% reduction in training time. In order to successfully detect oil spills, non-maximum suppression is used to increase detections, classification and regression losses for accuracy, and a learning rate of 0.0001, 100 epochs, and data augmentation for better convergence, accuracy, and generalisation.

5. **Performance Metrics**: IoU, Precision, Recall, and AP are used to evaluate the effectiveness of oil spill detection. Bounding box overlap is estimated by IoU, proper detection is measured by Precision, and spills that are detected are measured by Recall. Their tradeoff is shown by the P-R curve, whereas AP averages the levels of Precision over Recall. Good model performance is revealed by high precision, recall, and AP.

2.2.3 Results

TThe Faster R-CNN model performed well in oil spill detection, according to the Precision-Recall (P-R) curve analysis, with an Average Precision (AP) of 92.56%, 89.23% Precision, and 89.14% Recall. The training and validation loss curves demonstrated a consistent reduction with the minimum validation loss of 0.0511 at epoch 37, suggesting the model's optimal convergence. Model accuracy was constant when wind speeds ranged from 3 to 10 m/s and incidence angles ranged from 21° to 45°. However, very low wind speeds (1–2 m/s) and high incidence angles (45°–47°) led to reduced detection accuracy because of weak contrast and noise interference. Precision and Recall dropped to 74.85% and 74.72%, respectively, when noisy labels were included, suggesting that the quality of the training data affected the results. The model demonstrated real-time detection capabilities by processing a single SAR image in less than 0.05 seconds on an NVIDIA GeForce RTX 3090 GPU. HY-1C/1D Coastal Zone Imager (CZI) optical images were used to confirm the accuracy of oil spill detection, even in the face of slight positional shifts caused by ocean currents. The accuracy and robustness of detection can be further improved in future studies with multi-source satellite data.

2.3 A Deep Convolutional Neural Network for Oil Spill Detection from Spaceborne SAR Images

2.3.1 Introduction

The growth of offshore oil companies and marine trade exacerbates the serious environmental risks posed by oil spills. Because marine circumstances are frequently complex, spill detection is difficult and requires a continuous monitoring arrangement. Since oil reduces surface roughness and produces a black appearance in photos, Synthetic Aperture

Radar (SAR) is useful for spill detection. Traditional machine learning (ML) techniques, such Support Vector Machines (SVMs) and Decision Trees, rely on manually constructed features and have poor performance. By automatically extracting features, Deep Learning (DL), and specifically Convolutional Neural Networks (CNNs), improves detection.The Oil Spill Convolutional Network (OSCNet), a Transfer-Learning and hyperparameter-tuned Deep CNN designed to improve SAR-based detection, is proposed in the current work. On a sizable corpus of labelled SAR pictures, OSCNet performs better than ML and current DL models. The results show that the key to effective, scalable oil spill identification from spaceborne SAR images is CNNs' advantage in feature extraction and real-time observation.

2.3.2 Methodology

1. **Dataset Preparation**: 23,768 dark patches from 336 SAR pictures that have been identified as oil spills (4,843) and lookalikes (18,925) make up the dataset. Adaptive threshold segmentation was used to create dark patches, which were subsequently softened using filtering methods. For CNN input, images were resized to 64 x 64. The data was seperated(training: 13,722, testing: 10,036) and added to it by flipping, rotating, moving, and scaling.

2. **Model Architecture**: The OFCN model is a fully convolutional oil spill segmentation network based on U-Net. It is comprised of a skip-connected encoder-decoder design. The encoder uses convolutional layers with Batch Normalisation, ReLU activation, max-pooling, Squeeze-and-Excitation, and Dropout to process hierarchical features. By using bilinear upsampling and concatenation with encoder outputs, the decoder functions as a mirror image of the encoder structure. A binary segmentation mask for detecting oil spills is created by a final sigmoid activation.

3. **Training Configuration**: To lessen the class imbalance caused by oil spills, the OFCN model is trained using class-balanced binary cross-entropy. Overfitting is prevented by data augmentation techniques like flips, shifts, rotations, zooming, and shearing. A two-phase training procedure refines on full-resolution data after first training on low-resolution photos. Predictions are smoothed by overlapping sliding windows and test-time augmentations such flips and rotations at inference. For

increased accuracy, overlapping predictions are combined using spline interpolation.

4. **Model Optimization**: Hyperparameter tuning was used extensively to optimise the OSCNet model. ReLU's quick convergence led to its selection as the activation function. The Adam optimiser outperformed Momentum in the classification of SAR images. It was discovered that the ideal starting learning rate was 5.0×10^{-5}. A batch size of 256 struck a balance between computational cost and training efficiency. The network convergence was improved by his initialisation. Strong performance in oil spill detection was ensured by these optimisations, which increased model accuracy, recall, and precision.

5. **Performance Metrics**: Class imbalance and large image sizes caused the OFCN model to perform poorly on the test set, while achieving an F1 score of 0.892 on the validation set. Additionally, a Kruskal-Wallis H-test demonstrated that the F1 scores for the various incidence angles did not differ significantly. Precision and recall were impacted by test-time threshold tweaking; higher thresholds resulted in fewer false positives. The accuracy of the categorisation model varied among the 12 categories, and texture-based classifications were particularly challenging.

2.3.3 Results

With a maximum accuracy of 95.46%, a detection rate (recall) of 85.17%, a precision of 87.30%, and an F1-score of 85.55%, OSCNet outperformed traditional machine learning classifiers, according to test results from training and evaluating 15 models. The corresponding mean values were 94.01%, 83.51%, 85.70%, and 84.59%, respectively. OSCNet's superior classification performance was further validated by the ROC curve and AUC analysis; its AUC of 0.968 was much higher than AAMLP's 0.868. Stability and generalisability were demonstrated by the minimal oscillation found in stability analysis based on loss curves from three training models. Comparative experiments using the same dataset with VGG-16 and AAMLP revealed that OSCNet significantly outperformed both networks in terms of accuracy, precision, and recall thanks to the benefits of transfer learning and data augmentation. Furthermore, three distinct SAR images were employed to test OSCNet's efficacy. Using AAMLP, OSCNet was able to enhance the frequently misclassifying impacts of internal air waves, shoals, and wind speed low regions. Despite these improvements, there were still some challenging situations, such as when biological oil films and oil spills had comparable properties and additional data or auxiliary information was required to further boost classification performance.

2.4 Large-Scale Detection and Categorization of Oil Spills from SAR Images with Deep Learning

2.4.1 Introduction

For many years, spaceborne synthetic aperture radar, or SAR, has been essential for monitoring oil spills. It helps with response preparation by detecting spills from pipelines, ships, and offshore installations. Large stored datasets that enable advanced detection algorithms are a result of the regular spills caused by generated water discharge in regions like the North Sea. Continuous monitoring regardless of weather or sunlight is made possible by the open availability of satellite data, particularly from Sentinel-1A and Sentinel-1B. This is crucial in areas that are cloudy and gloomy. Due to reduced backscatter, oil spills appear as dark patches in SAR images; nevertheless, classification is challenging due to the similar signals from natural ocean phenomena including biogenic slicks and low-wind zones. Deep learning, such as convolutional neural networks (CNNs), which automatically learn features, outperforms classical detection, which is based on a three-stage process: low-backscatter detection, feature selection, and statistical classification. However, the scarcity of labelled training data limits the uptake of CNN. With an optimised CNN, a second model for shape- and texture-based categorisation, and a high-scale visualisation pipeline, this article proposes a deep-learning system for oil spill detection and classification. This approach enhances oil spill monitoring and response planning for environmental conservation by utilising huge datasets.

2.4.2 Methodology

1. **Dataset Preparation**: Sentinel-1 SAR photos with binary masks for oil spill detection make up the data. The data is preprocessed by cutting the maximum VV values to 150 and smoothing to eliminate noise. The datasets $\mathcal{D}_1$ (classified oil spills) and $\mathcal{D}_2$ (segmentation) are created by cutting off patches of size 160 x 160 pixels. There are balanced patches with and without oil leaks in $\mathcal{D}_2$. To train the model,

the data is separated into test, validation, and training sets.

2. **Model Architecture**: Based on the reduced VGG-16, the OSCNet model distinguishes between oil spills and lookalikes in SAR imagery. It accepts 64x64 greyscale inputs, three FC layers with 128, 128, and 2 nodes, and optimally convolution layers. ReLU, Adam optimiser, 5×10^{-5} learning rate, batch size of 256, dropout of 0.1, and He initialisation are among the key characteristics. It guarantees fast convergence, high classification accuracy, and efficient feature extraction for SAR oil spills.

3. **Training Configuration**: Data augmentation and SAR dark patch data were used to train OSCNet from the start. ReLU activation, softmax cross-entropy loss, and dropout with 0.5 were the training parameters. Adam was the optimiser, and the batch size of 256 and learning rate of 5×10^{-5} ensured efficiency. He initialisation was used to initialise the weights. Network depth and FC node configurations were optimised in comparative studies to provide robust generalisation across 15 dataset variances.

4. **Model Optimization**: Large-scale hyperparameter tuning was utilised to optimise the OSCNet model. The ReLU activation function's rapid convergence led to its selection. The Adam optimiser outperformed Momentum in the classification of SAR images.5×10^{-5} was determined to be the ideal beginning learning rate. The ideal balance between computational cost and training efficiency was achieved with a batch size of 256. Network convergence was accelerated through the use of initialisation. By enhancing the model's accuracy, recall, and precision, these optimisation strategies ensured consistent performance in the detection of oil spills.

5. **Performance Metrics**: The F1-score, accuracy, recall, and precision are the performance metrics used to evaluate oil spill classification models. The model's capacity to identify actual spills is measured by recall (detection rate), whereas accuracy represents overall classification accuracy. The proportion of accurately classified spills among those that were detected is reported by Precision. The F1-score strikes a compromise between recall and precision. The capacity of the model to discriminate is also measured by the area under the curve (AUC) and receiver operating characteristic (ROC) curves; better performance is indicated by higher AUC values.

2.4.3 Results

This study presented a deep learning method for classifying and detecting large-scale oil spills from SAR imagery. In order to identify each pixel as "oil" or "non-oil," the detection was framed as an image segmentation issue. A fully convolutional neural network was trained on patches of SAR pictures and the corresponding human-annotated binary masks. High detection performance was demonstrated by experimental evaluations, yielding outcomes comparable to those of human operators. A high F1-score indicates low misclassification, whereas imitators are classified as "non-oil," and the program mostly detects mineral oil slicks from human discharge. This is the first attempt to categorise oil spills in SAR photos, as a second neural network, in addition to detection, classifies oil spills by form, texture, and contrast into 12 classes at the patch level rather than pixel-wise. In order to distinguish real spills from imitations, the categorisation results are useful in revealing information about the origin, weathering stage, and internal changes of oil slicks. Certain categories have worse classification accuracy because of their reliance on external data, noise, and human label variability because it is still difficult to distinguish texture and contrast from SAR images. Despite these drawbacks, the results provide helpful recommendations for enhancing upcoming oil spill monitoring systems. Lastly, this study showed a production pipeline that can identify and classify oil spills, gather SAR images from the ASF Sentinel-1 library, and display the results on an interactive map for further analysis of each spill. As the first comprehensive system for monitoring oil spill activities over time, this deep learning-based solution represents a major breakthrough in automated environmental monitoring and marine ecosystem preservation.

2.5 A Deep-Learning Framework for the Detection of Oil Spills from SAR Data

2.5.1 Introduction

Oil spills pose a serious threat to the environment because they harm ecosystems, birds, and marine life. Timely diagnosis is crucial because they are primarily caused by pipeline, ship, and tanker accidents. By capturing photos where oil and lookalikes appear as dark patches, Synthetic Aperture Radar (SAR) makes it easier to identify spills, while separation is still challenging. The majority of deep learning techniques rely on manually created features or have accuracy limitations. To improve detection, a two-stage deep learning approach is suggested. A 23-layer Convolutional Neural Network (CNN) is used in the first stage to classify SAR image patches with a sensitivity, specificity, and accuracy of about 99%. The next step surpasses existing methods by segmenting large oil spills with 92% accuracy, 84% precision, and an 80% Dice score using a five-stage U-Net in conjunction with an optimised Dice loss. This approach enhances autonomous oil spill detection for pollution control and environmental monitoring.

2.5.2 Methodology

1. **Dataset Preparation**: 310 pre-processed SAR images that have undergone cropping, radiometric calibration, speckle noise reduction, and dB-to-luminosity conversion make up the data, which was acquired by Sentinel-1 via ESA's Copernicus Hub. There are pictures of ships, land, sea, oil spills, and lookalikes (1250 x 650 x 3). For categorisation, one-dimensional labels and three-dimensional masks were created. Only 210 photos with oil spills or lookalikes were used because of the severe class imbalance; background pixels were labelled as "0," while oil spills were labelled as "1."

2. **Model Architecture**: There are two stages to the deep learning model that was built. Using SoftMax and cross-entropy loss optimisation, a 23-layer CNN first predicts $64 \times 64 \times 3$ SAR image patches into sections of oil spills that are significant and those that are not. The second phase then reduces generalised dice loss for class imbalance handling by segmenting using the five-stage U-Net. U-Net comprises an expansive path for precise oil spill mask production, a bottleneck bridge, and a contracting path for feature extraction.

3. **Training Configuration**: The Adam optimiser, which decayed by epochs for stability, was used to optimise the model with a learning rate of 0.001. A batch size of 32 and early halting prevented overfitting. Model generalisation was enhanced by data augmentation techniques like flipping and rotation. Whereas the U-Net segmentation process was trained for 100 epochs, the CNN classification process was trained for 50 epochs. Evaluation of the performance on various dataset partitions

was aided by cross-validation.

4. **Model Optimization**: To address class imbalance, the model was trained using generalised dice loss for segmentation and cross-entropy loss for classification. CNN parameters were optimised by backpropagation and gradient descent, while overfitting was prevented by batch normalisation and dropout. Stability was achieved by adjusting learning rates, while robustness was offered via five-fold and ten-fold cross-validation. While precision and Dice score measured segmentation accuracy, accuracy, sensitivity, specificity and AUC measured classification performance.

5. **Performance Metrics**: Accuracy, sensitivity, specificity, precision, Dice score, and Area Under the Curve (AUC) were used to measure the model's accuracy. Five and ten-fold cross-validation were used to preserve classification robustness. The agreement between the true and anticipated labels was assessed using a quadratic weighted Kappa score. Oil spill detection accuracy was measured in segmentation using Dice score and pixel-wise precision. The ACRF framework outperformed state-of-the-art algorithms with oil spill detection, achieving 92% accuracy, 84% precision, and an 80% Dice score.

2.5.3 Results

In order to reduce speckle noise, create patches, and classify SAR pictures using a 23-layer CNN, the proposed deep-learning architecture first applies a Frost filter. The pictures are separated into 64 × 64 × 3 patches, inspected for the presence of oil spills, and then categorised using a five-fold and ten-fold cross-validation method. With a 99% validation accuracy, the CNN identified high-oil spill regions for a five-stage U-Net segmentation process. Using patches with 40–60% oil spill pixels for training minimised pixel-wise class imbalance and produced a robust model with minimal overfitting. The U-Net achieved a Dice score of 80%, segmentation accuracy of 92%, and precision of 84%.The novel framework demonstrated improved precision and Dice score while reducing false positives when compared to other segmentation models, such as SegNet with a VGG-19 encoder. 64 × 64 patches were shown to offer the optimum sensitivity-precision trade-off in patch size optimisation experiments. Furthermore, lowering the Generalised Dice Loss (GDL) was similar to Jaccard loss but provided a slight performance boost when compared to recall loss.In recognising oil spills in unbalanced datasets, the model outperformed earlier approaches by Hidalgo et al. and Krestenitis et al., offering significantly higher segmentation accuracy and stability. However, there were drawbacks when segmenting patches with severe class imbalances since accuracy decreased when the patch was overloaded with background or oil spill pixels.

2.6 Comparison and Gaps Identified

2.6.1 Comparison Table

Paper Name	Technique Used	Advantages	Disadvantages
Oil spill classification based on satellite image using deep learning tech- niques	CNN with RGB and IR imagery utilising Fully Convolutional Networks (FCN)	• Use of UAVs for efficient real-time detection • 89% mIoU was attained using RMSprop and MobileNet. • Functions well in dimly lit environments	• Restricted to the use of UAVs • The quality of RGB pre-processing determines performance. • Non-oil slicks may be misinterpreted by IR pictures.

Paper Name	**Technique Used**	**Advantages**	**Disadvantages**
A Novel Deep Learning Method for Marine Oil Spill Detection from Satellite1 Synthetic Aperture Radar Imagery	Two-stage model: 23-layer CNN for classification and 5-stage U-Net for segmentation	• High accuracy of classification (99%) • 92% segmentation accuracy with enhanced dice loss • Strong to resemble misclassification	• Extreme class imbalance causes precision to decline. • Patches with a lot of background affect segmentation performance. • Little dataset (only 310 SAR pictures)
A Deep Convolutional Neural Network for Oil Spill Detection from Spaceborne SAR Images	ResNet50, RCNN with VGG16, and RCNN with InceptionV3	• The precision and F1-score are highest when using RCNN with VGG16. • Thorough evaluation of various deep learning architectures	quality of the dataset affects model performance.ResNet50 performs worse than RCNN.

Paper Name	**Technique Used**	**Advantages**	**Disadvantages**
Large-Scale Detection and Categorization of Oil Spills from SAR Images with Deep Learning	OSCNet (Modified VGG-16) for classification, Secondary model for categorization, Visualization pipeline	• The first model to classify oil spills according to contrast, texture, and shape. • Using a large dataset -time analysis through interactive mapping	categories have inferior classification accuracy.Utilising external data to classify Estimating texture and contrast from SAR photos is still difficult.
A Deep-Learning Framework for the Detection of Oil Spills from SAR Data	23-layer CNN for classification, 5-stage U-Net for segmentation	• 99% classification accuracy is attained. • For segmentation, 84% precision and 80% dice score • Surpasses cutting-edge techniques like SegNet	• Extreme class imbalance in patches reduces precision. • Class imbalance persists in spite of optimisations.

2.6.2 Gaps Identified

Defects Found in the Present State of the Art The following gaps still exist in oil spill detection despite advancements:

1. Class Imbalance Problems - In extremely unbalanced datasets, current models have trouble segmenting oil spills, which results in incorrect classification.

2. False Positives - SegNet is one of several techniques with high false positive rates, which lower precision and dependability.

3. Processing Efficiency - Real-time implementation of deep learning models is difficult due to their high processing power and big dataset requirements.

4. Generalisation Across Various SAR Datasets - Since most models are trained on particular datasets, they might not be able to generalise effectively to various satellite sources and ambient variables.

5. Patch Size Optimisation - The impact of patch size variations on segmentation performance, which affects sensitivity and precision, is not adequately explored in the literature.

2.7 Summary

This chapter provided a thorough literature evaluation of specific deep learning techniques for detecting oil spills using SAR imagery. A comparative examination of the advantages and disadvantages of various model architectures, including ResNet, R-CNN, U-Net, and SegNet, in addressing oil spill detection challenges has been provided in the reviewed literature. The study found that while CNN-based models are excellent at extracting features, segmentation models such as SegNet and U-Net are more effective at localising oil spills. Despite advancements, issues with performance in highly imbalanced datasets, computational inefficiency, and high false positive rates persist. The recently presented architecture, which combines a 23-layer CNN classification with a five-stage U-Net segmentation, provides answers to these issues by improving precision and eliminating false positives. Its resilience in attaining high accuracy and Dice scores with a balance between sensitivity and specificity was confirmed by a comparison with cutting-edge techniques. The chapter concludes by outlining research priorities, including improved localisation in challenging ocean conditions, more accurate management of extreme class imbalance, and the best training techniques to maximise generalisation across different SAR data sets. The technique in the next chapter is based on the findings.

Chapter 3

System Design

The components and their relationships are defined in this chapter, which also offers an overview at the system architecture level. It describes the algorithms used at several stages, including evaluation, model training, and data preprocessing. It shows how SAR pictures are computed from raw data to the final prediction via graphics. This chapter also covers the technologies and tools used, such as TensorFlow, OpenCV, and Python, as well as the CSIRO dataset selected for training detection models. The final section outlines the main deliverables, which include a scalable detection system, trained models, and performance comparison. The chapter concludes with a project timetable, module division, and task segregation for methodical implementation.

3.1 System Architecture

The system's architecture is displayed as follows:

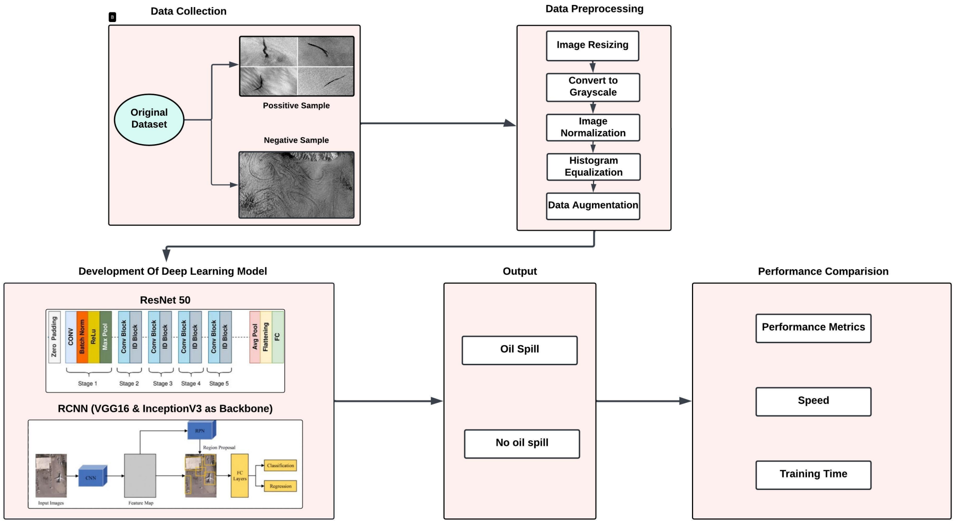


Figure 3.1: System Architecture

3.1.1 ResNet50

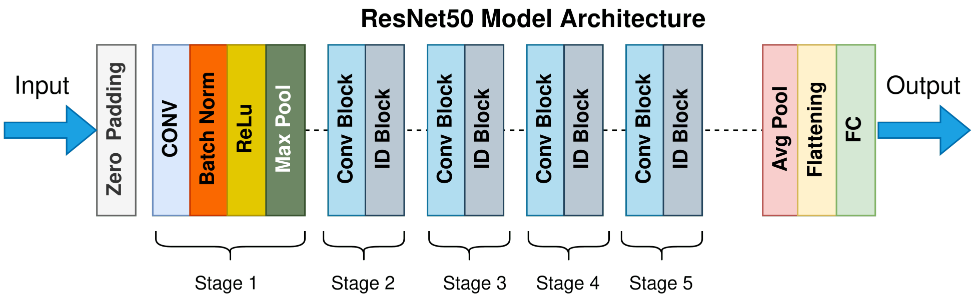


Figure 3.2: ResNet50 Architecture

To solve the vanishing gradient issue, ResNet50 is a 50-layer deep convolutional neural network (CNN) incorporating residual learning. ResNet50 is largely responsible for the development of skip (shortcut) connections, which allow gradients to move across the network without causing loss and facilitate the efficient training of deeper networks.

The building blocks consist of:

1. First Convolutional and Pooling Layers After a 7x7 convolution, max pooling is used to reduce spatial dimensions while maintaining important characteristics. .

2. The four residual stages that form the foundation of ResNet50 are bottleneck residual blocks (1×1, 3×3, and 1×1 convolutions). These blocks learn feature representations while maintaining computational efficiency. Global Average Pooling (GAP) Layer summarises feature maps to reduce spatial dimensions to a single vector.

3. The Fully Connected (FC) Layer creates final classification outputs by converting learning features to class probabilities.

4. Softmax Activation is used to transform raw scores into probabilities in order to detect oil spills.

3.1.2 RCNN

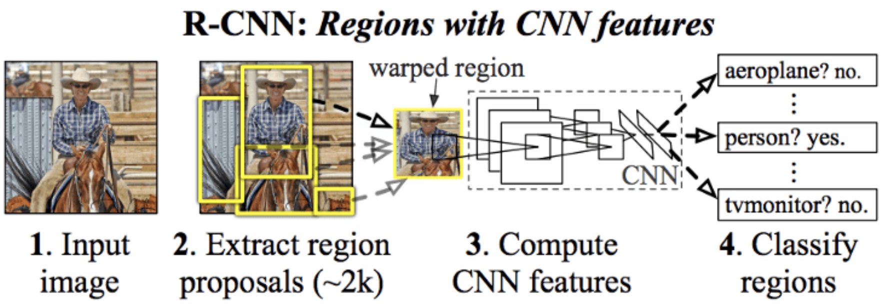


Figure 3.3: RCNN Architecture

Region-based convolutional neural networks (RCNNs) are a class of deep networks designed for object segmentation and detection. Unlike standard CNNs that categorise complete images, RCNNs are interested in recognising and locating objects in an image by suggesting areas of interest (RoIs) before classification.

The following key steps constitute the RCNN structure:

1. Region Proposal: Using Selective Search, the model produces a number of region proposals, or potential places where objects might be found.

2. Feature Extraction: Every suggested region is scaled and run through a CNN backbone that has already been trained (such as VGG16 or InceptionV3) in order to extract significant features.

3. Bounding Box Regression and Classification: A fully connected Support Vector Machine is utilised to categorise the region, and a bounding box regression model modifies the bounding box coordinates for precise localisation.

Variants Used in This Project For feature extraction, I employ two variations of RCNN using different CNN backbones:

1. Convolutional and pooling layers make up the deep yet simple architecture of RCNN with VGG16, which employs VGG16 as the feature extractor.

   Using InceptionV3, which combines multiple filter sizes in a single layer, the RCNN with InceptionV3 improves feature diversity and detection accuracy.

3.1.3 VGG16

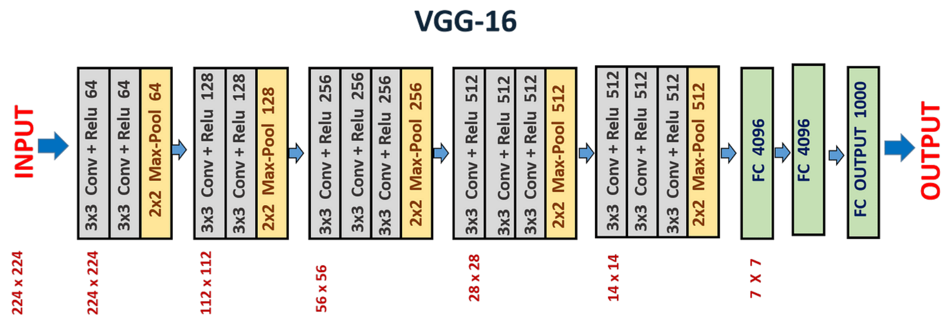


Figure 3.4: VGG16 Architecture

VGG16 (Visual Geometry Group 16-layer network) is a deep convolutional neural network (CNN) with a minimal but effective architecture. It contains 16 weighted layers with mainly 3×3 convolutional filters, thereby making it a good feature extractor for object detection and image classification tasks.

Key Components of VGG16

1. Convolutional Layers: The network contains 13 convolutional layers, each of which uses 3x3 filters with a stride of 1 to provide comprehensive feature extraction.

2. Max Pooling Layers: After a few convolutions, a max pooling layer with a stride of 2 and 2×2 dimensions preserves significant characteristics while reducing spatial dimensions.

3. The last component of the network consists of three fully connected layers that process the gathered features and generate predictions.

4. Softmax Activation: The output layer employs softmax activation for class probabilities.

3.1.4 InceptionV3

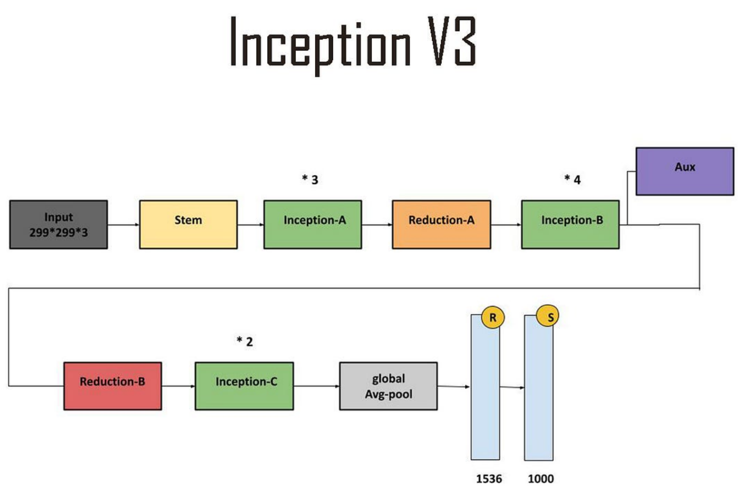


Figure 3.5: InceptionV3 Architecture

A CNN called InceptionV3 was created for effective object detection and image classification. With the use of factorized convolutions, auxiliary classifiers, and optimized grid sizes, it is more accurate compared to earlier models while lowering the cost of computation.

Key Components of InceptionV3:

1. Inception Modules: These modules simultaneously use 1x1, 3x3, and 5x5 convolutions to record spatial data at various sizes. The 1×1 convolutions reduce dimensionality prior to the application of larger filters, increasing computational efficiency.

2. Factorised Convolutions: InceptionV3 uses two smaller 3x3 convolutions rather than the larger 5x5 filters to reduce the number of parameters. This expedites training while preserving the quality of feature extraction.

3. Auxiliary Classifiers: Intermediate classifiers are positioned across the network to facilitate gradient flow and avoid vanishing gradients in the deep layers. Additionally, these classifiers provide generalisation through model regularisation.

4. Batch Normalisation: This method, which is applied after the majority of layers, normalises activations to speed up convergence and stabilise training. By lowering internal covariate shifts, this enhances the deep network's overall performance.

5. Layer of Global Average Pooling (GAP): Fully linked layers are replaced by InceptionV3's GAP layer, which calculates feature map averages before to classification. By lowering overfitting and the amount of parameters, this helps to improve the model's efficiency.

3.2 Component Design

3.2.1 Data Acquisition and Preprocessing

1. **Data Acquisition**

   For efficient object detection and image classification, a CNN known as InceptionV3 was developed. It is more accurate than previous models while reducing computation costs through the use of factorised convolutions, auxiliary classifiers, and optimised grid sizes.

2. **Preprocessing Techniques**

   To improve image quality and model performance, the following preprocessing methods were used:

   (a) Image Resizing: All images were scaled to a standard size to provide uniformity throughout the dataset and compatibility with deep learning models.

   (b) Greyscale conversion preserves important details while lowering computational complexity since SAR images contain intensity-based information.

   (c) Normalisation: Pixel values were normalised to a predetermined range (such as 0 to 1 or -1 to 1) in order to enhance convergence during model training.

   Altering pixel intensity values, histogram equalisation is used to improve contrast and increase the visibility of oil spills against the background.

(d) Data Augmentation: The dataset was artificially augmented using methods such as flipping, zooming, rotation, and brightness modifications to improve the model's generalisation and resilience.

3.2.2 Feature Extraction and Classification

1. **Feature Extraction**

Feature extraction, a crucial step in deep learning, separates oil spills from the sea surface by locating noteworthy patterns in SAR data. In this study, features were extracted using three deep learning architectures:

(a) ResNet50: This framework uses residual learning and deep hierarchical feature extraction to find intricate patterns in SAR data.

(b) RCNN with VGG16 improves the localisation of oil spills by using a deep sequential model to extract structured features from areas of an image.

(c) RCNN with InceptionV3 improves detection accuracy for different oil spill sizes and forms by applying multi-scale feature extraction through Inception modules.

2. **Classification**

Following feature extraction, a fully connected layer using a softmax classifier labels areas as either "oil spill" or "non-oil spill." Testing the models using Precision, Recall, and F1-Score ensured accurate classification. By contrasting various architectures, the best method for real-time oil leak detection and crisis management is found.

3.2.3 Sequence Diagram

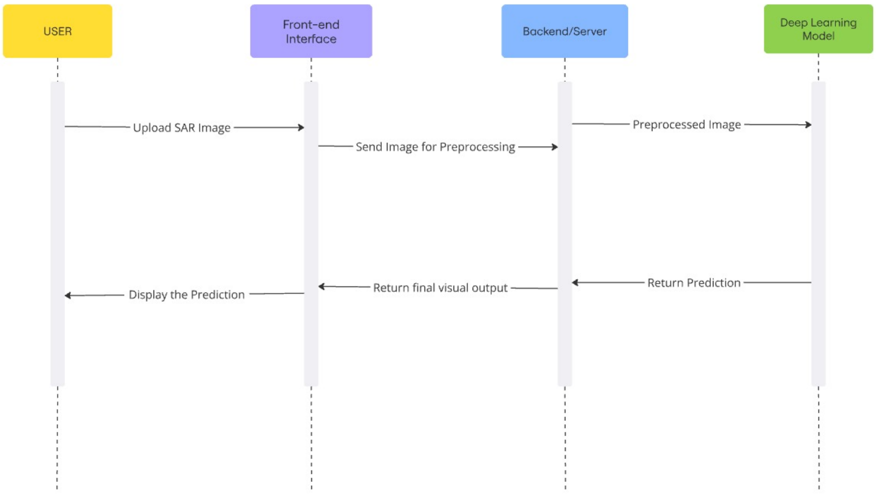


Figure 3.6: Sequence Diagram

3.2.4 Tools and Technologies

1. Software Requirements

   - Programming Language: Python 3 is a programming language used for evaluation, data preprocessing, and model construction.
   - Deep learning frameworks: TensorFlow and Keras are used to design and train deep learning models, such as ResNet50, RCNN with VGG16, and RCNN with InceptionV3.
   - Image Processing Libraries: OpenCV, for preprocessing SAR images (e.g., greyscale conversion, histogram equalisation).
   - Visualization Tools: Matplotlib is used to plot picture visualisations and model performance indicators.
   - Hardware Acceleration: Kaggle-GPU, for quicker inference and training.
   - Development Environment: For interactive model creation and testing, use Kaggle.
   - Image Analysis: Scikit-image, for tasks involving image segmentation and analysis.

2. Hardware Requirements
    - Processor: For effective computing and management of big datasets, need an AMD Ryzen 7 or Intel Core i5 processor (or higher).
    - Graphics Processing Unit (GPU): It is advised to use an NVIDIA GPU (RTX 3060 or higher) for faster deep learning model training.
    - Memory (RAM): Minimum: 8GB RAM
    - Storage: Minimum 512GB SSD
    - Additional Hardware: high-resolution screen (for results visualisation). For storing models and backing up datasets, use external storage.

3.2.5 Dataset Identified

Australian space agency CSIRO (Commonwealth Scientific and Industrial Research Organisation) supplied the data set used in this investigation. This data set consists of 5,630 Synthetic Aperture Radar (SAR) pictures that depict ocean regions free of oil spills and those affected by them.

Because the photographs include a variety of scenarios, such as different weather conditions, water textures, and spill configurations, the data set is appropriate for deep learning model training. The carefully selected data set, which includes both representative oil spills and non-spill areas, improves testing and robust model training.

3.2.6 Module Divisions

Every module of the project focusses on a different facet of the process of detecting oil spills. Among the modules that are required are:

1. Preprocessing and Data Acquisition: To improve the quality of the data, SAR images are gathered and preprocessing methods such data augmentation, image scaling, greyscale transformation, normalisation, and histogram equalisation are used.

2. Feature Extraction: ResNet50, RCNN with VGG16, and RCNN with InceptionV3 are used to extract deep features from SAR images in order to detect oil spills.

3. Region Proposal and identification: RCNN-based object identification is used to identify and locate possible oil leaks in photos.

4. Classification Module: This module uses extracted features and a fully connected layer with softmax activation to categorise detected areas as either non-oil spill or oil spill.

5. Performance Evaluation: Precision, Recall, and F1-Score are used to evaluate the model's performance in order to identify the best architecture.

6. Explainability and Visualisation: To improve model interpretability in real-time decision-making, saliency maps and heatmaps are used.

3.2.7 Project Timelines

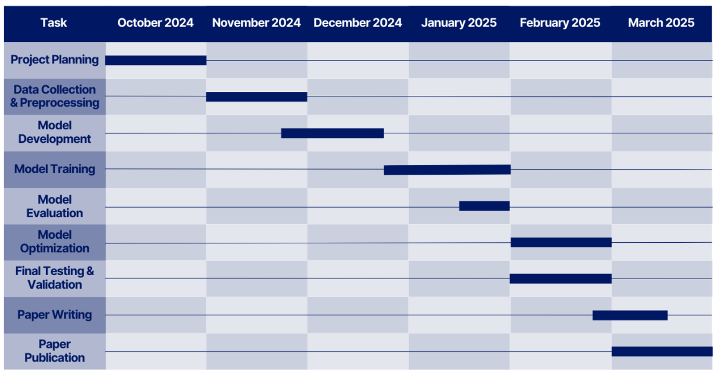


Figure 3.7: Gantt Chart

Chapter 4

System Implementation

4.1 Dataset Identified

A set of 5,630 SAR pictures from the European Space Agency (ESA), Sentinel-1, and CSIRO were used in this study. These photographs cover regions such as the Great Barrier Reef and portions of Southeast Asia, including Singapore. The dataset consists of labelled SAR images, where the negative samples are oil-free ocean surfaces that may resemble features like low wind zones and algae blooms, and the positive samples are known oil spills that show up as dark areas because of reduced radar backscatter. The dataset contains a range of situations that can be used to train deep models for oil spill detection.

4.2 Proposed Methodology

4.2.1 Dataset Collection and Selection

5,630 SAR pictures from multiple sources, including CSIRO, Sentinel-1, and ESA, make up the dataset used in this study. The photographs cover a range of maritime regions, including Southeast Asia and the Great Barrier Reef. The labelled samples in the collection show oil spills as dark marks due to reduced radar backscatter, while the negative samples show oil-free ocean surfaces with items that resemble them, such as low wind areas and algae blooms. To enable efficient model training and testing, the dataset is divided into 80% for training (4,504 photos) and 20% for testing (1,126 images).

4.2.2 Preprocessing

When training Synthetic Aperture Radar (SAR) pictures for deep learning oil spill detection, preprocessing is a crucial step. Preprocessing raises the accuracy and efficiency of the model training process, preserves consistency throughout the dataset, and enhances the quality of the images. Preprocessing techniques normalise the dataset and improve feature perceptibility because SAR images contain noise, uneven contrast levels, and different resolutions.

All SAR images are scaled to 256×256 pixels as part of the first preprocessing stage. In order for the models to accept input photos without dimension inconsistencies, this sets a standard size for all images that is compatible with deep learning architectures. Additionally, maintaining a consistent image size reduces information loss during training and makes computation simpler. Resizing ensures that all samples have a consistent spatial scale, which improves feature extraction because distinct datasets may contain images with varying resolutions.

```
# Create ImageDataGenerators with custom preprocessing function
train_data_gen = ImageDataGenerator(
    rotation_range=15,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.1,
    horizontal_flip=True,
    preprocessing_function=custom_preprocess_image  # Use the custom function here
)

test_data_gen = ImageDataGenerator(preprocessing_function=custom_preprocess_image)

# Load datasets using flow_from_directory
train_data = train_data_gen.flow_from_directory(
    directory="/kaggle/input/major-project/Train/Train",
    target_size=(img1, img2),
    color_mode='rgb',  # Use rgb since we are converting to 3 channels
    class_mode='sparse',  # Change this if you use categorical labels
    batch_size=batch_size,
    shuffle=True
)

test_data = test_data_gen.flow_from_directory(
    directory="/kaggle/input/major-project/Test/Test",
    target_size=(img1, img2),
    color_mode='rgb',  # Use rgb since we are converting to 3 channels
    class_mode='sparse',  # Change this if you use categorical labels
    batch_size=batch_size,
    shuffle=False
```

Figure 4.1: Image Resizing

The pictures are then transformed into greyscale. Since colour information is not inherent in SAR images, greyscale processing minimises computation by reducing the number of input channels from three (RGB) to one. As a result, the models can focus on pertinent texture variations rather than superfluous colour variations. For accurate classification, greyscale photos emphasise the structural and contrast features that distinguish nearby saltwater from oil spills.

Another crucial preprocessing step is normalisation. Originally ranging from 0 to

255, the SAR picture pixel intensity values are normalised to a scaled range of 0 to 1. This maintains the input distributions across training cases, improving numerical stability and accelerating model convergence. Additionally, normalisation prevents certain pixels with extremely high or low intensity values from having an excessively negative effect on learning, leading to more stable weight updates throughout training.

The photos are subjected to histogram equalisation for improved feature visibility. By more equally spreading intensity values, histogram equalisation improves contrast in SAR images, which are often unevenly contrasted due to shifting illumination and sensor noise. This makes dark spots, such as oil spills, stand out more than the surrounding water. This procedure gives the model more vision, allowing it to identify minute patterns and variances that might otherwise go undetected.

Large, diverse datasets are necessary for deep learning models to generalise well, hence data augmentation techniques are used to intentionally boost the diversity of training samples. Data augmentation aids in resolving imbalances in datasets, especially when photographs of oil spills are under-represented in comparison to images of non-spills.

```
# User input for model parameters
img1 = 224
img2 = 224
batch_size = 32
epochs = 20
learning_rate = 0.0001


# Custom preprocessing function for grayscale handling, standardization, and histogram equalisation
def custom_preprocess_image(image):
    # Convert image to grayscale if it's not already
    if image.shape[-1] == 3:  # Check if the image has 3 channels (RGB)
        image = tf.image.rgb_to_grayscale(image)  # Convert to grayscale
   # Standardize the image
    image = tf.image.per_image_standardization(image)
    # Perform histogram equalization on the grayscale image
    image = tf.image.adjust_gamma(image, gamma=1.0)  # Simple adjustment as an example
    # Note: More complex histogram equalization can be implemented if needed
    # Convert grayscale (1 channel) to 3 channels by duplicating the channel
    image = tf.image.grayscale_to_rgb(image)  # Now shape will be (height, width, 3)
    return image
```

Figure 4.2: Image Preprocessing Techniques

When combined, these preprocessing techniques significantly enhance the quality and stability of datasets, enabling deep learning-based oil spill detection systems to acquire rich and consistent features. Preprocessing aids in the improved performance of oil spill detection systems by providing consistent input dimensions, improved contrast, reduced computational load, and data augmentation.

4.2.3 Model Selection

This study uses three deep learning models to detect oil spills from SAR images: ResNet-50, R-CNN with VGG16, and R-CNN with Inception V3. Accurate feature extraction, object detection, and classification are the criteria used to select the models.

Because of its capacity to use deep residual learning to solve the vanishing gradient problem, ResNet-50 was chosen. Its residual linkages facilitate the efficient flow of information across multiple layers, enabling hierarchical feature extraction that detects patterns in SAR imagery at both low and high levels. Because of this property, it can be used to distinguish between natural seafloor structures and oil slicks and to recognise complicated textures.

```
# Load ResNet50 model with transfer learning
base_model = keras.applications.ResNet50(weights='imagenet', include_top=False, input_shape=(img1, img2,

# Freeze the base model
base_model.trainable = False

# Build the model
model = Sequential([
    base_model,
    Flatten(),
    Dense(512, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(128, activation='relu'),
    BatchNormalization(),
    Dropout(0.5),
    Dense(1, activation='sigmoid')  # Use sigmoid for binary classification (0 or 1)
])

# Compile the model
optimizer = keras.optimizers.Adam(learning_rate=learning_rate)
model.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(train_data, epochs=epochs, validation_data=test_data)
```

Figure 4.3: Code of ResNet-50

Accurate region-based oil spill identification is achieved by using the R-CNN with VGG16 as the backbone. By providing detailed spatial feature extraction, VGG16 strengthens the model's ability to identify and categorise oil spills more precisely. The R-CNN method creates region proposals, which are subsequently refined and categorised, allowing the network to differentiate oil spills from other like dark structures in SAR pictures.

Besides, R-CNN with Inception V3 is investigated as an alternative backbone for its effectiveness in multi-scale feature learning. Inception V3's architectural innovations, including factorized convolutions and asymmetric kernels, maximize computational efficiency while capturing spatial information at different resolutions. This improves detec-

```python
def build_rcnn(input_shape=(224, 224, 1)):
    # Input layer with grayscale images (1 channel)
    inputs = Input(shape=input_shape)

    # Convert grayscale to RGB by repeating the channel 3 times
    x = Lambda(grayscale_to_rgb)(inputs)

    # Use VGG16 as backbone, pretrained on ImageNet (for RGB)
    backbone = VGG16(weights="imagenet", include_top=False, input_shape=(224, 224, 3))
    backbone.trainable = False  # Freeze all layers of VGG16 initially

    # Optionally unfreeze the last few layers for fine-tuning
    for layer in backbone.layers[-4:]:
        layer.trainable = True

    # Add RCNN custom layers
    x = backbone(x)
    x = GlobalAveragePooling2D()(x)  # Replace Flatten with GAP for better generalization
    x = Dense(128, activation="relu")(x)
    x = Dropout(0.4)(x)
    outputs = Dense(1, activation="sigmoid")(x)  # Binary classification

    model = Model(inputs, outputs)
    return model

# Build and Compile the Model
model = build_rcnn(input_shape=(224, 224, 1))  # Grayscale input
model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
```

Figure 4.4: Code of RCNN with VGG-16 as backbone

tion of oil spills in varying shapes and sizes, leading to higher accuracy and robustness.

```python
# Define RCNN-Like Model with InceptionV3 Backbone
def build_rcnn_inception(input_shape=(299, 299, 1)):
    inputs = Input(shape=input_shape)

    # Convert grayscale to RGB by repeating the channel 3 times
    x = Lambda(grayscale_to_rgb)(inputs)

    # Use InceptionV3 as backbone
    backbone = InceptionV3(weights="imagenet", include_top=False, input_shape=(299, 299, 3))
    backbone.trainable = False  # Freeze all layers initially

    for layer in backbone.layers[-4:]:
        layer.trainable = True  # Unfreeze last few layers for fine-tuning

    x = backbone(x)
    x = GlobalAveragePooling2D()(x)  # GAP for better generalization
    x = Dense(128, activation="relu")(x)
    x = Dropout(0.4)(x)
    outputs = Dense(1, activation="sigmoid")(x)  # Binary classification

    model = Model(inputs, outputs)
    return model

# Build and Compile the Model
model = build_rcnn_inception(input_shape=(299, 299, 1))
model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])
```

Figure 4.5: Code of RCNN with Inception-V3 as backbone

4.2.4 Model Training

To identify and categorise oil spills, preprocessed SAR image data is used to train the selected deep models, ResNet-50, R-CNN with VGG16, and R-CNN with Inception V3. The input photos are supplied into each model during training, and features are extracted using convolutional layers and classified using fully connected layers.

ResNet-50 estimates the likelihood of oil leaks using a softmax classifier and a global average pooling (GAP) layer. ResNet-50's residual learning procedure guarantees steady gradients, allowing deep networks to be used with effective training without seeing a significant drop in performance. It is perfect for distinguishing oil spills from other dark structures on the ocean surface because of its ability to extract both low-level and high-level features.

Region proposal and classification are the two phases of training for R-CNN models. By creating bounding boxes around high-probability areas, the Region Proposal Network (RPN) can identify potential oil spill areas in SAR pictures. After being smoothed, the regions are put into fully linked layers, where the network categorises them as backdrop or oil spills. Bounding box regression modifies region coordinates to improve localisation. The backbone architectures of Inception V3 and VGG16 enhance feature extraction, and Inception V3 in particular excels at multi-scale learning, making it more adaptable to variations in spill size and shape.

```
model.compile(optimizer="adam", loss="binary_crossentropy", metrics=["accuracy"])

# Train the Model
history = model.fit(
    train_gen,
    validation_data=test_gen,
    epochs=30,
    verbose=1
)
```

Figure 4.6: Code of RCNN with Inception-V3 as backbone

Adam and other optimisation techniques are used to improve model convergence and lower loss functions. To perform a comprehensive examination of the models' effectiveness in detecting oil spills, they are evaluated based on key performance metrics, including accuracy, precision, recall, and F1-score.

4.3 Summary

Dataset identification, preprocessing, model selection, and training are all part of the system's implementation. Sentinel-1, ESA, and CSIRO SAR images are collected, offering a varied collection of training data. To improve model performance and generalisation, preprocessing entails scaling the photos to 256 x 256 pixels, converting the images to greyscale, normalising, equalising the histogram, and augmenting the data.

Because of their superiority in feature extraction and classification, ResNet-50, R-CNN with VGG16, and R-CNN with Inception V3 are the models chosen. R-CNN models allow precise object detection using region proposals, while ResNet-50 is effective at extracting deep features.

During training, preprocessed images are fed into models, where fully connected layers do classification and convolutional layers extract spatial data. To effectively identify oil spills, optimisation algorithms such as Adam are utilised, and performance is measured using accuracy, precision, recall, and F1-score.

Chapter 5

Results and Discussions

5.1 Results

Oil spill detection in Synthetic Aperture Radar (SAR) data has been effectively accomplished using a variety of deep learning methods, including Region-Based Convolutional Neural Networks (RCNN) and ResNet-50. These are utilised for training and testing in order to identify and differentiate oil spills based on their ability to extract the best features and their strong pattern-detection capabilities.

Figure 5.1 shows the training accuracy and training loss of the ResNet-50 model. It is evident from Figure 5.1 that the accuracy of the training and testing data is extremely similar. This shows that the model is operating effectively and isn't experiencing overfitting or underfitting. The model has been successfully trained, obtaining essential features from SAR images that aid in precise oil spill identification, as seen by the small difference between training and testing accuracy. Furthermore, the convergence of training loss over multiple epochs suggests that the model has successfully decreased error during training. The durability and stability of the ResNet-50 model in carrying out oil spill detection tasks are demonstrated by a steady drop in loss and a steady accuracy ratio.

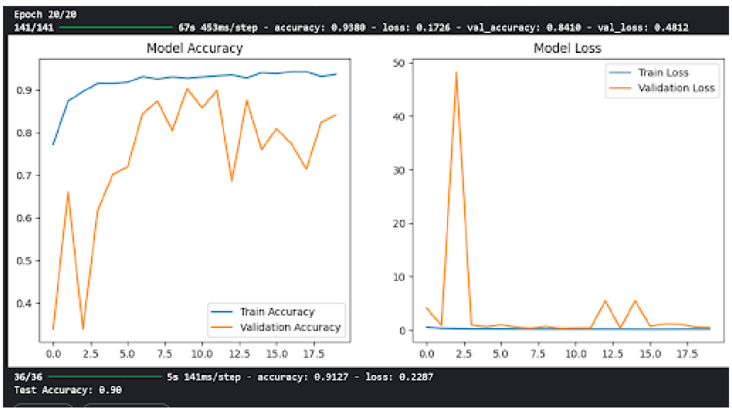


Figure 5.1: Model Accuracy and Loss of ResNet50

Figures 5.2 and 5.3 illustrate the process of prediction and categorization of SAR images to determine the presence or absence of an oil spill. These figures showcase how ResNet-50 , analyze input SAR images and classify them into two categories: ‘Oil Spill Present” or ”No Oil Spill”.

Prediction

```python
from tensorflow import keras
import tensorflow as tf
import numpy as np
model = tf.keras.models.load_model('/kaggle/input/resnet50model/resnet50_oil_spill_model.h5')

def predict_resnet50(model, image_path):
    img = tf.keras.utils.load_img(image_path, target_size=(224, 224))
    img_array = tf.keras.utils.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    pred = model.predict(img_array)
    return "Oil Spill" if pred[0] > 0.5 else "No Oil Spill"

# Example prediction
sample_image = "/kaggle/input/sampleoilspill/Samples/0_0_0_img_qBIhHv0a36ynbvIW_GIB_cls_1.jpg"
print(f"Prediction for sample image: {predict_resnet50(model, sample_image)}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 4s 4s/step
Prediction for sample image: Oil Spill
```

Figure 5.2: ResNet50 Prediction-Presence of Oil Spill

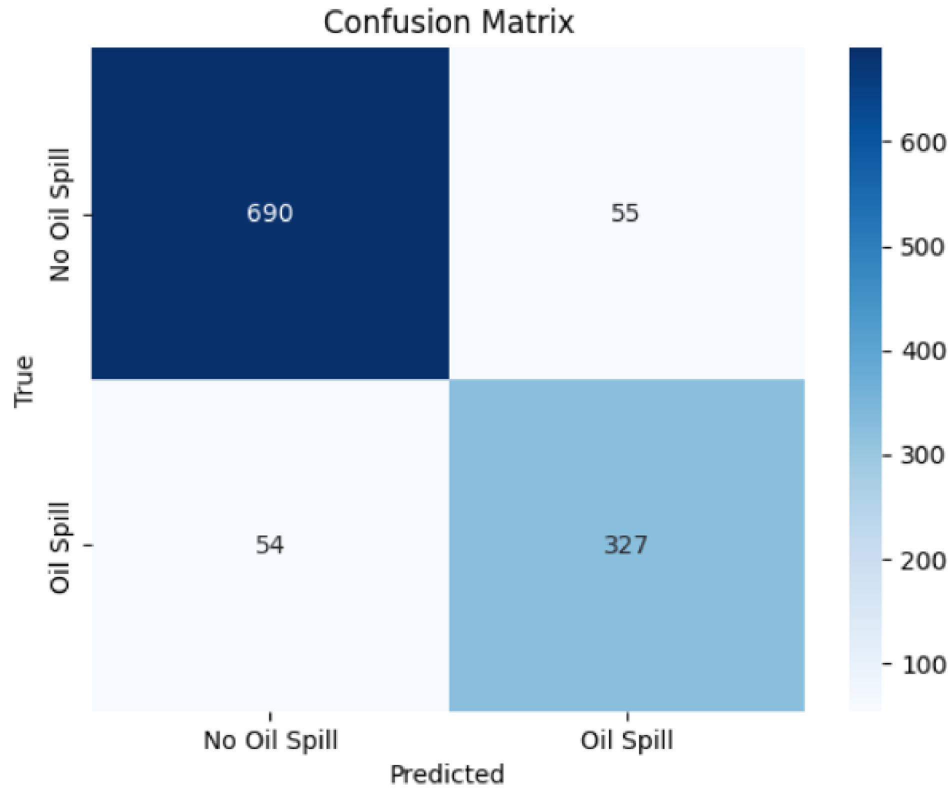
Prediction

```python
from tensorflow import keras
import tensorflow as tf
import numpy as np
model = tf.keras.models.load_model('/kaggle/input/resnet50model/resnet50_oil_spill_model.h5')

def predict_resnet50(model, image_path):
    img = tf.keras.utils.load_img(image_path, target_size=(224, 224))
    img_array = tf.keras.utils.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    pred = model.predict(img_array)
    return "Oil Spill" if pred[0] > 0.5 else "No Oil Spill"

# Example prediction
sample_image = "/kaggle/input/sampleoilspill/Samples/0_0_0_img_72gUvSnnkPTMGLF3_GBR_cls_0.jpg"
print(f"Prediction for sample image: {predict_resnet50(model, sample_image)}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 3s 3s/step
Prediction for sample image: No Oil Spill
```

Figure 5.3: ResNet50 Prediction-Absence of Oil Spill

Figure 5.4 gives the confusion matrix and classification report of ResNet-50 based on the parameters Accuracy, Precision, Recall, and F1-score. ResNet-50 yields the accuracy of 90% in detecting the oilspill.




```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.93      0.93       745
           1       0.86      0.86      0.86       381

    accuracy                           0.90      1126
   macro avg       0.89      0.89      0.89      1126
weighted avg       0.90      0.90      0.90      1126

Precision: 0.8560
Recall: 0.8583
F1 Score: 0.8571
```

Figure 5.4: ResNet50 Confusion Matrix

Figure 5.5 displays the R-CNN model's training accuracy and loss with VGG16. It is clear from the figure that testing and training data accuracy are almost similar. This demonstrates that there is neither overfitting nor underfitting in the model and that it generalizes. The model successfully learned salient elements from SAR images that facilitate oil spill recognition, as evidenced by the little difference between training and test accuracy. Additionally, the model has effectively reduced training error, as evidenced by the training loss's steady decline over multiple epochs. The R-CNN with VGG16 model's stability and resilience in oil spill recognition tasks are demonstrated by the steady decrease in loss and accuracy stability.

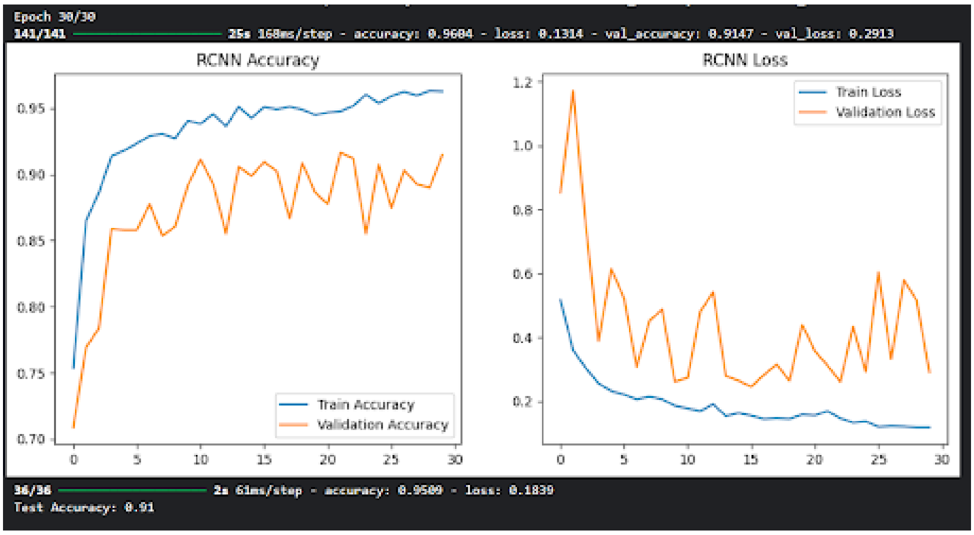


Figure 5.5: Model Accuracy and Loss of RCNN with VGG16

Figures 5.6 and 5.7 illustrate the process of prediction and categorization of SAR images to determine the presence or absence of an oil spill. These figures demonstrate how the R-CNN model with VGG16 analyzes input SAR images and classifies them into two categories: 'Oil Spill Present' or 'No Oil Spill'.

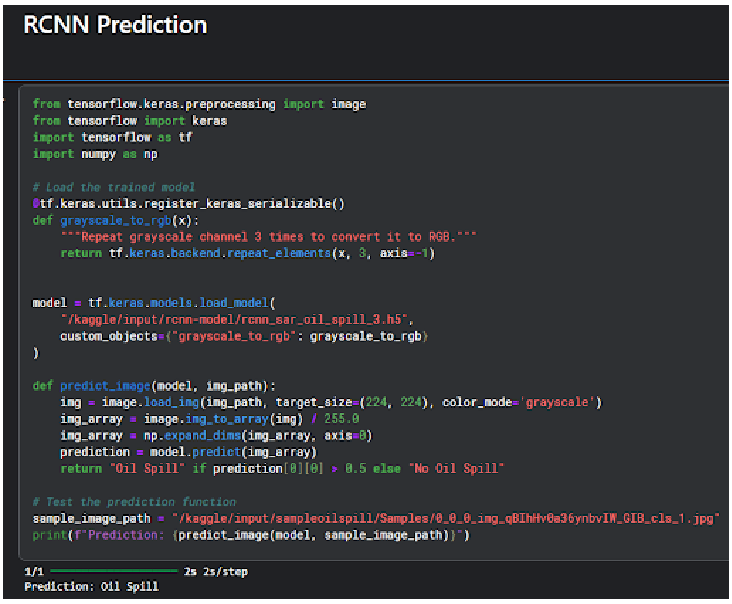


Figure 5.6: RCNN with VGG16 Prediction-Presence of Oil Spill

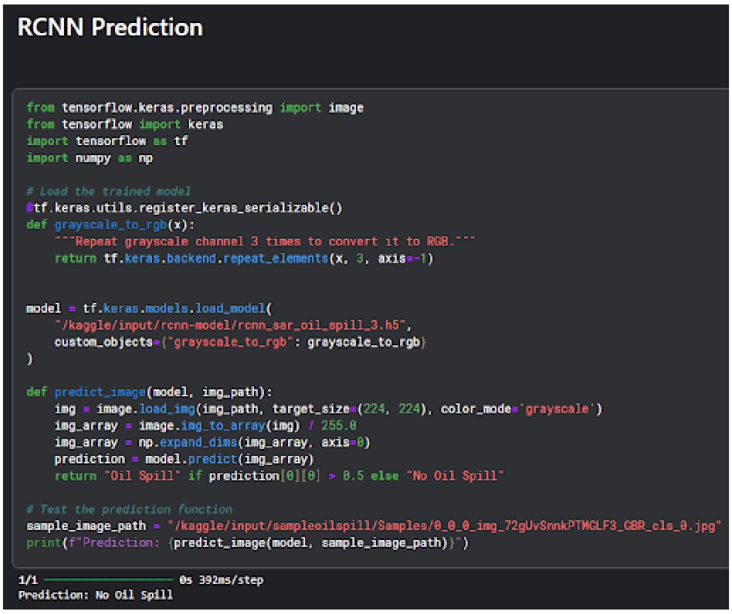
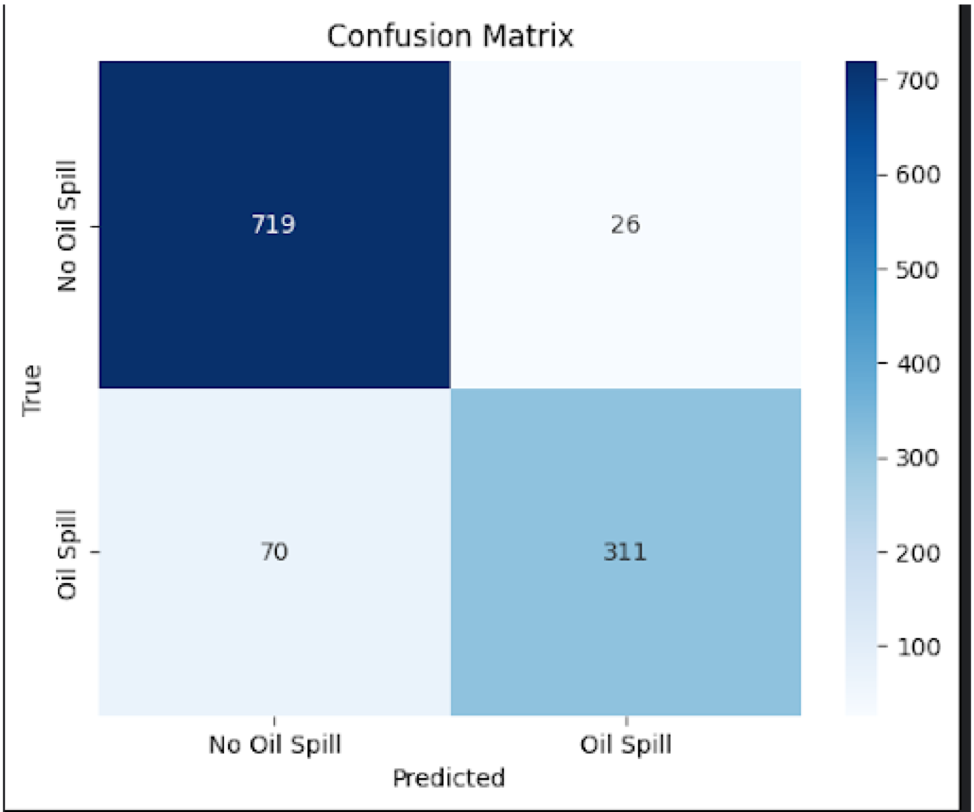


Figure 5.7: RCNN with VGG16 Prediction-Absence of Oil Spill

Figure 5.4 presents the confusion matrix and classification report of the R-CNN model with VGG16, based on the parameters Accuracy, Precision, Recall, and F1-score. The R-CNN model with VGG16 achieves an accuracy of 91% in detecting the oil spill.




```
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.97      0.94       745
           1       0.92      0.82      0.87       381

    accuracy                           0.91      1126
   macro avg       0.92      0.89      0.90      1126
weighted avg       0.92      0.91      0.91      1126

Precision: 0.9228
Recall: 0.8163
F1 Score: 0.8663
```

Figure 5.8: RCNN with VGG16 Confusion Matrix

Figure 5.9 shows the training accuracy and training loss of the R-CNN model using InceptionV3. It is evident from the chart that the accuracy of the training and testing sets is nearly identical. This indicates that there is neither overfitting nor underfitting and that the model is generalizing properly. The model has been successfully trained, capturing valuable features from SAR photos that help with precise oil spill identification, as evidenced by the small difference between training and test accuracy. Furthermore, the model has successfully decreased error during training, as evidenced by the convergence of training loss over multiple epochs. The resilience and stability of the R-CNN with InceptionV3 model in carrying out oil spill detection tasks are demonstrated by a steady accuracy rate and a consistent loss reduction.

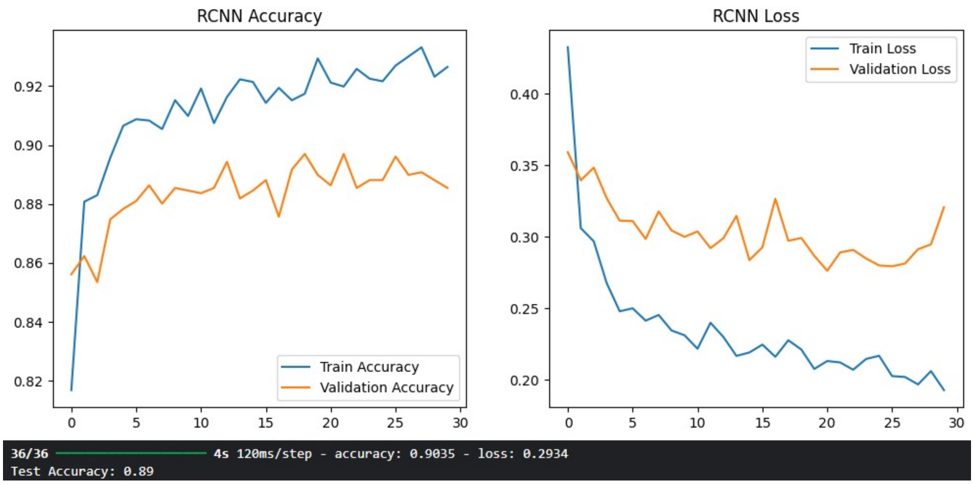


Figure 5.9: Model Accuracy and Loss of RCNN with InceptionV3

Figures 5.10 and 5.11 illustrate the process of prediction and categorization of SAR images to determine the presence or absence of an oil spill. These figures showcase how the R-CNN model with InceptionV3 analyzes input SAR images and classifies them into two categories: 'Oil Spill Present' or 'No Oil Spill'.

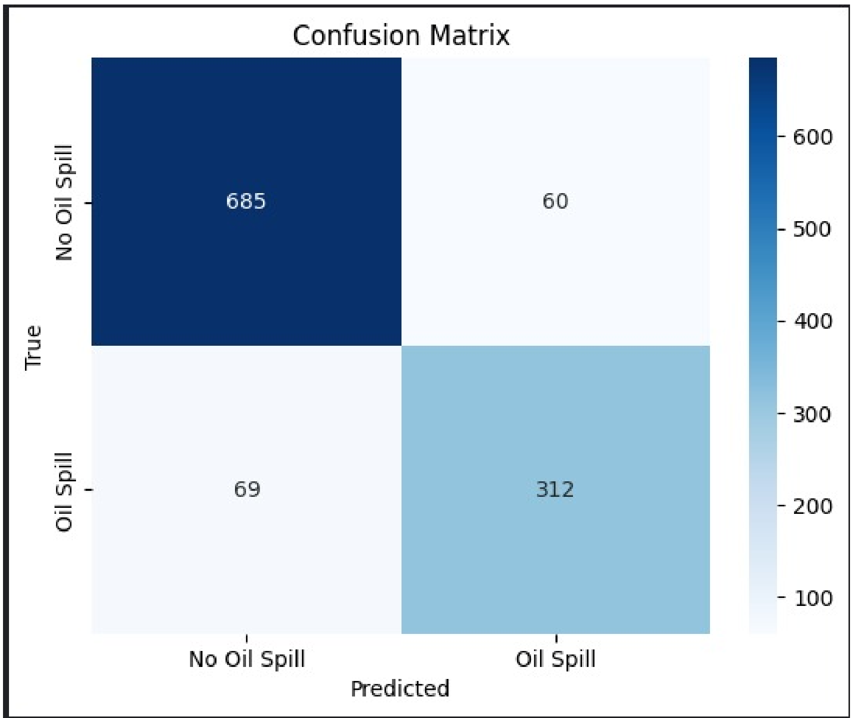
```
from tensorflow.keras.preprocessing import image
from tensorflow import keras
import tensorflow as tf
import numpy as np

# Load the trained InceptionV3 model
@tf.keras.utils.register_keras_serializable()
def grayscale_to_rgb(x):
    """Repeat grayscale channel 3 times to convert it to RGB."""
    return tf.keras.backend.repeat_elements(x, 3, axis=-1)

model = tf.keras.models.load_model(
    "/kaggle/input/rcnn-inception-v3-model/rcnn_sar_oil_spill_inception.h5",
    custom_objects={"grayscale_to_rgb": grayscale_to_rgb}
)

def predict_image(model, img_path):
    img = image.load_img(img_path, target_size=(299, 299), color_mode='grayscale')  # InceptionV3 us
    img_array = image.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    prediction = model.predict(img_array)
    return "Oil Spill" if prediction[0][0] > 0.5 else "No Oil Spill"

# Test the prediction function
sample_image_path = "/kaggle/input/sampleoilspill/Samples/0_0_0_img_qBIhHv0a36ynbvIW_GIB_cls_1.jpg"
print(f"Prediction: {predict_image(model, sample_image_path)}")

1/1 ━━━━━━━━━━━━━━━━━━━━ 6s 6s/step
Prediction: Oil Spill
```

Figure 5.10: RCNN with InceptionV3 Prediction-Presence of Oil Spill

```
from tensorflow.keras.preprocessing import image
from tensorflow import keras
import tensorflow as tf
import numpy as np

# Load the trained InceptionV3 model
@tf.keras.utils.register_keras_serializable()
def grayscale_to_rgb(x):
    """Repeat grayscale channel 3 times to convert it to RGB."""
    return tf.keras.backend.repeat_elements(x, 3, axis=-1)

model = tf.keras.models.load_model(
    "/kaggle/input/rcnn-inception-v3-model/rcnn_sar_oil_spill_inception.h5",
    custom_objects={"grayscale_to_rgb": grayscale_to_rgb}
)

def predict_image(model, img_path):
    img = image.load_img(img_path, target_size=(299, 299), color_mode='grayscale')  # InceptionV3 uses
    img_array = image.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    prediction = model.predict(img_array)
    return "Oil Spill" if prediction[0][0] > 0.5 else "No Oil Spill"

# Test the prediction function
sample_image_path = "/kaggle/input/sampleoilspill/Samples/0_0_0_img_ca9QOv0U0q6eU9Ow_PHI_cls_0.jpg"
print(f"Prediction: {predict_image(model, sample_image_path)}")

1/1 ━━━━━━━━━━━━━━━━━━━━ 9s 9s/step
Prediction: No Oil Spill
```

Figure 5.11: RCNN with InceptionV3 Prediction-Absence of Oil Spill

Figure 5.12 presents the confusion matrix and classification report of the R-CNN model with InceptionV3, based on the parameters Accuracy, Precision, Recall, and F1-score. The R-CNN model with InceptionV3 achieves an accuracy of 89% in detecting the oil spill.




```
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.92      0.91       745
           1       0.84      0.82      0.83       381

    accuracy                           0.89      1126
   macro avg       0.87      0.87      0.87      1126
weighted avg       0.88      0.89      0.89      1126

Precision: 0.8387
Recall: 0.8189
F1 Score: 0.8287
```

Figure 5.12: RCNN with InceptionV3 Confusion Matrix

By evaluating the performance of ResNet-50 and RCNN on several performance parameters, insightful conclusions about their relative advantages and disadvantages can be made. The comparative study makes it easy to identify the optimal deep learning model for precise and effective oil spill detection in SAR photos. Evaluation measures that are used to evaluate the performance of these models include accuracy, precision, recall, and F1-score. These metrics give a broad sense of how well the model detects oil spills while reducing false positives and false negatives. Table 5.1 provides a summary of the findings compared.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
ResNet50	90	85.6	85.8	85.7
RCNN with VGG16	**91**	**92.2**	81.6	**86.6**
RCNN with InceptionV3	89	83.8	81.8	82.8

Table 5.1: Performance comparison of deep learning models for oil spill detection

5.2 Discussion

- With the highest Precision (92.2%) and F1-Score (86.6%), RCNN with VGG16 was the best model for detecting oil spills. This indicates that the model detects oil spills accurately while minimizing false positives.
- ResNet50 was a reliable alternative, with a high accuracy of 90% while striking a balance between precision and recall.
- As evidenced by its lowest values, especially in Recall (81.8%), RCNN InceptionV3 missed more oil spills than the other models. Overall, though, it did well in categorization.

Based on these findings, the optimal model to identify oil spills from SAR imagery is RCNN with VGG16. The performance of all models may be enhanced, however, with further adjustments such as fine-tuning and data augmentation.

Chapter 6

Conclusions & Future Scope

Using a deep learning approach, three models—ResNet50, RCNN utilizing VGG16, and RCNN employing InceptionV3—were used to compare with one another in order to detect ocean oil spills from SAR data. Test Accuracy, Precision, Recall, and F1-Score were used to gauge the models' performance. With an F1-Score of 0.8571, precision of 0.8560, recall of 0.8583, and test accuracy of 0.90, ResNet50 achieved these results. RCNN with VGG16 outperformed the others, with the greatest test accuracy of 0.91, accuracy of 0.9228, recall of 0.8163, and F1-Score of 0.8663. An F1-Score of 0.8287, precision of 0.8387, recall of 0.8189, and test accuracy of 0.8 were obtained using RCNN with InceptionV3. According to all of these findings, RCNN-VGG16 outperforms the other two models. Preprocessing techniques such as data augmentation, histogram equalization, normalization, and grayscale greatly improved detection accuracy. This article describes how oil spills may be detected in real time using deep learning, making it a useful tool for environmental monitoring and crisis management.

In order to increase classification accuracy and operational efficiency, future oil spill detection research will employ a multi-modal fusion technique to further develop the system by merging optical, thermal infrared, and synthetic aperture radar (SAR) data. To improve feature discrimination functions, hybrid models—particularly those that employ transformer-based and attention-based architectures—will be investigated. The development of transparent, intelligible models and additional decision-making knowledge will be the main goals of the application of Explainable AI (XAI) techniques. To increase the accuracy of oil spill detection, advanced multi-sensor fusion will be used to enhance real-time data combination. Finally, the enhancement of real-time processing will be prioritized in order to increase the speed and efficiency of detecting units, allowing for effective monitoring and quick response.

References

1. Abba, Abubakar Salihu, et al. "Oil spill classification based on satellite image using deep learning techniques." Baghdad Science Journal 21.2 (SI) (2024): 0684-0684.

2. Huang, Xudong, Biao Zhang, William Perrie, Yingcheng Lu, and Chen Wang. "A novel deep learning method for marine oil spill detection from satellite synthetic aperture radar imagery." Marine Pollution Bulletin 179 (2022): 113666.

3. Zeng, Kan, and Yixiao Wang. "A deep convolutional neural network for oil spill detection from spaceborne SAR images." Remote Sensing 12, no. 6 (2020): 1015.

4. Bianchi, Filippo Maria, Martine M. Espeseth, and Njål Borch. "Large-scale detection and categorization of oil spills from SAR images with deep learning." Remote Sensing 12, no. 14 (2020): 2260.

5. Shaban, Mohamed, Reem Salim, Hadil Abu Khalifeh, Adel Khelifi, Ahmed Shalaby, Shady El-Mashad, Ali Mahmoud, Mohammed Ghazal, and Ayman El-Baz. "A deep-learning framework for the detection of oil spills from SAR data." Sensors 21, no. 7 (2021): 2351.

6. Magrì, Stefania, et al. "Oil Spill Identification and Monitoring from Sentinel-1 SAR satellite earth observations: A machine learning approach." Chemical Engineering Transactions 86 (2021): 379-384.

7. Matkan, A. A., M. Hajeb, and Z. Azarakhsh. "Oil spill detection from SAR image using SVM based classification." The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 40 (2013): 55-60.

8. Tong, Shengwu, et al. "Multi-feature based ocean oil spill detection for polarimetric SAR data using random forest and the self-similarity parameter." Remote Sensing 11.4 (2019): 451.

List of Publications & Funding

1. A Deep Learning Approach to Detect Ocean Oil Spills using SAR Images - Communicated to Malaysian Journal of Computer Science - SCOPUS indexed

2. 4 Lakhs funding received from MeiTy as part of India AI fellowship programme.

Appendix A: Presentation

Deep Learning Approach for Detecting Ocean Oil Spills from SAR Images

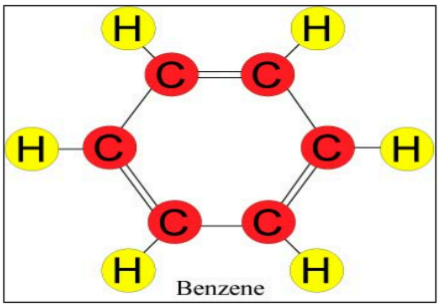
Guide
Dr. Preetha K G

Team Members
Niranjan S (U2103158)
Saurav Krishnan (U2103191)
Sradha Shajan (U2103201)
Wivin Winny (U2103216)

Contents

Introduction

An oil spill is the release of liquid petroleum hydrocarbons into the marine environment, resulting in significant contamination and environmental damage.



Introduction

- Marine oil slicks originate from
  - Pipeline Ruptures
  - Oil Rigs
  - Transport Vehicles
  - Illegal Dumping
- They can also be transported by winds and currents away from the point of origin.
- It’s therefore important to not only detect but also monitor the drift and spreading.

SAR (Synthetic Aperture Radar)

- Synthetic Aperture Radar (SAR) is a type of radar technology that uses the motion of the radar antenna usually mounted on a satellite or aircraft to create high-resolution images of the Earth's surface using microwaves.

- Unlike optical sensors, SAR can penetrate clouds, fog, and darkness, making it highly useful for monitoring the ocean and detecting oil spills, especially in challenging weather conditions or at night.

- SAR's synthetic aperture technology provides detailed and accurate images, useful for precise mapping and analysis.

Working of SAR

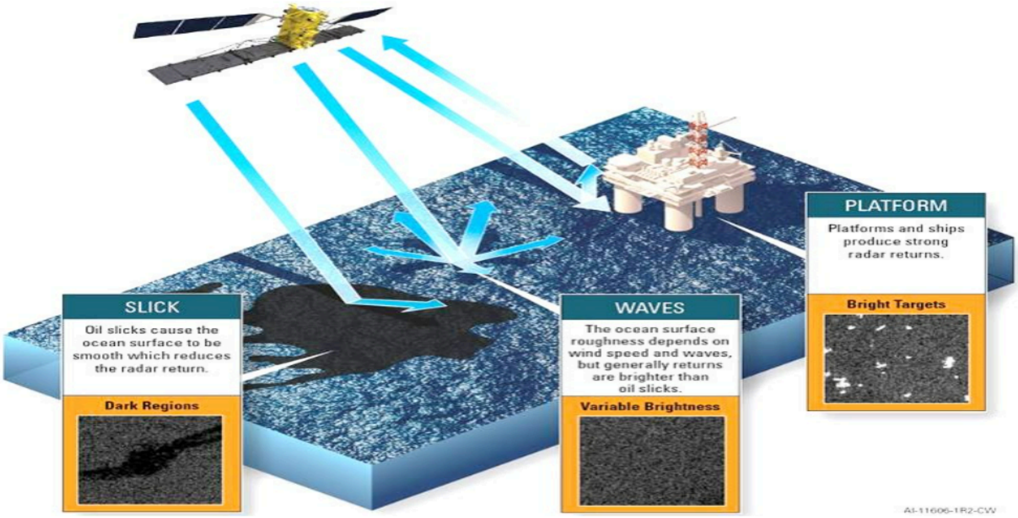
- Radar Reflection: SAR emits radar waves that reflect differently off oil slicks, which create a smoother surface compared to rougher ocean water.

- Surface Texture Detection: SAR identifies smoother oil slicks as darker areas in images due to reduced surface roughness, contrasting with brighter rough waters due to a strong backscatter signal.

- Image Contrast: SAR images show dark oil patches against bright ocean areas, allowing easy detection and mapping of the oil spill.

- All-Weather Monitoring: SAR continuously monitors the ocean, detecting oil spills in any weather or light conditions, ensuring real-time spill tracking.

Working of SAR



Problem Definition

- Oil spills threaten the environment and economy, requiring rapid detection.
- Traditional methods are slow, expensive and error-prone.
- Develop a deep learning-based system for oil spill detection from SAR images.
- Performance comparison of various deep learning technique for Oil Spill detection.

Purpose & Need

- Ensure fast and accurate oil spill detection using deep learning to minimize environmental damage .

- Overcome the limitations of traditional methods by leveraging SAR imaging for reliable detection in all weather conditions.

Applications & Beneficiaries

- Environmental Protection Agencies
  - Enables real-time monitoring of oil spills to minimize ecological damage.
  - Supports faster response and cleanup operations.
- Maritime & Coastal Authorities
  - Helps in early detection of spills to prevent further spread.
  - Assists in law enforcement by identifying sources of illegal oil discharge.

Applications & Beneficiaries

- Oil & Gas Industry
    - Ensures compliance with environmental regulations by monitoring offshore drilling sites.
    - Reduces potential financial and reputational risks from spill incidents.
- Disaster Management & Response Teams
    - Provides quick and accurate data for decision-making during oil spill crises.
    - Enhances coordination for efficient cleanup strategies.

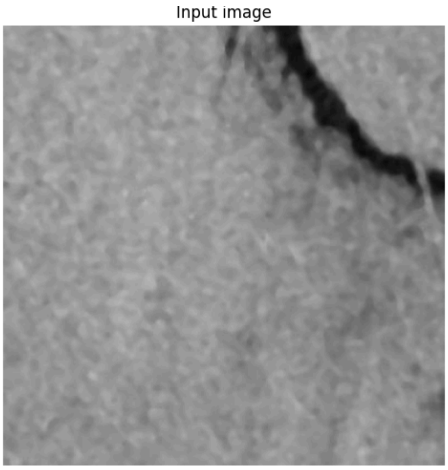
Objective

- To develop a deep learning-based system for Oil spill detection from SAR images.
- Compare ResNet-50, RCNN with VGG16, and RCNN with InceptionV3 for accuracy and efficiency.
- Enhance detection accuracy while minimizing false positives.

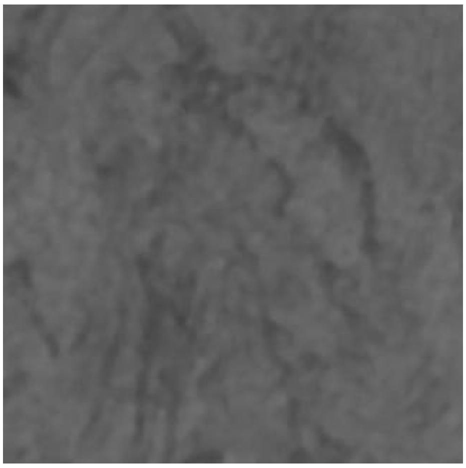
Dataset Identified

https://data.csiro.au/collection/csiro%3A57430v1

Data Size:5630 images



Oil Spilled Image



Non Oil Spilled Image

Literature Survey

Paper	Methods Used	Results	Advantages	Disadvantages
Ali Mahmoud, Mohammed Ghazal, and Ayman El-Baz. "A deep-learning framework for the detection of oil spills from SAR data." Sensors 21, no. 7 (2021)	• The dataset includes 310 images from ESA,COAH. • Preprocessing with Frost filter • Classification using CNN with 23 layers and U-Net	• CNN achieved 99% accuracy. • U-Net scored 92% accuracy on testing. • 64×64 patches gave best segmentation results.	• Noise Reduction for Better Feature Extraction • CNN and U-Net for High Accuracy. • Generalized Dice Loss for Handling Class Imbalance	• Needs a large amount of data. • Potential Overfitting Due to Model Complexity
Krestenitis, Marios, et al. "Oil spill identification from satellite images using deep neural networks." Remote Sensing 11.15 (2019)	• 1112 EMSA-confirmed SAR images. • Radiometric calibration, speckle filtering applied. • U-Net, LinkNet, PSPNet, DeepLab.	• DeepLabv3+ segments oil, look-alikes. • DeepLabv2 struggles with oil detection. • PSPNet inaccurate, many false positives. • U-Net, LinkNet detect oil, ships.	• Automates detection with high accuracy. • Scalable for large areas using satellite imagery. • Works well with SAR images.	• Needs a large amount of data. • Hard to interpret model decisions. • Computationally intensive.

Paper	Methods Used	Results	Advantages	Disadvantages
Zeng, Kan, and Yixiao Wang. "A deep convolutional neural network for oil spill detection from spaceborne SAR images." Remote Sensing 12, no. 6 (2020)	• 23,768 dark patches from 336 SAR images, manually labeled by CNOOM. • Data preprocessing and augmentation • Transfer learning using VGG-16 • Custom CNN design (OSCNet)	• Highest accuracy achieved: 95.46% by OSCNet. • Best F-measure recorded: 85.55% overall performance.	• Effective Data Augmentation. • Improved Generalization with Transfer Learning. • Optimized Architecture for Accuracy and Efficiency.	• Potential Information Loss from Fixed Input Size (64×64). • Dataset Imbalance Leading to Bias. • Computationally Intensive Model.
Bianchi, Filippo Maria, Martine M. Espeseth, and Njål Borch. "Large-scale detection and categorization of oil spills from SAR images with deep learning." Remote Sensing 12.14 (2020)	• KSAT SAR dataset, 713 images • Preprocessed and augmented data • U-Net-based OFCN model used	• Soft predictions boost categorization accuracy • CNN struggles with noisy textures • Automated large-scale SAR processing pipeline • 665 spills found in 501 products	• Scalable for large ocean analysis • Accurate in oil spill detection, categorization • Effective across diverse SAR conditions	• Needs large labeled data for categorization • High computational cost at scale • Hard to interpret model decisions

Assumptions

1. **Data Availability:**
   - It is assumed that SAR images from reliable sources like ESA and CSIRO are freely available for use.
   - Adequate annotated datasets are available or can be manually created for training and testing.
2. **SAR Image Quality:**
   - SAR images used for oil spill detection must be of sufficient resolution (e.g., Sentinel-1 images) to accurately detect features.
   - Images are expected to have undergone preprocessing for better results.

Software Requirements

Development Environment:

- Python (preferred for deep learning frameworks and image processing libraries)
- Kaggle

Operating System:

- Windows 10 (64-bit).

Software Requirements

Deep Learning Libraries:

- TensorFlow/Keras : For model implementation and training.
- OpenCV: For image processing tasks (e.g., resizing, augmentation).
- scikit-image: For image analysis and segmentation tasks.
- Matplotlib: For plotting different graphs.

Hardware Requirements

Processor (CPU):

- Intel Core (i5 or i7) or AMD Ryzen 7 (or higher) for efficient computation.

Graphics Processing Unit (GPU):

- NVIDIA GeForce RTX 2080 or better for deep learning acceleration.
- CUDA support: Ensure GPU is CUDA-enabled for TensorFlow/PyTorch.

Memory (RAM):

- Recommended: 8GB RAM or higher for handling large datasets.

Hardware Requirements

Storage:

- Solid State Drive (SSD): Minimum 128GB for fast data access and model training.

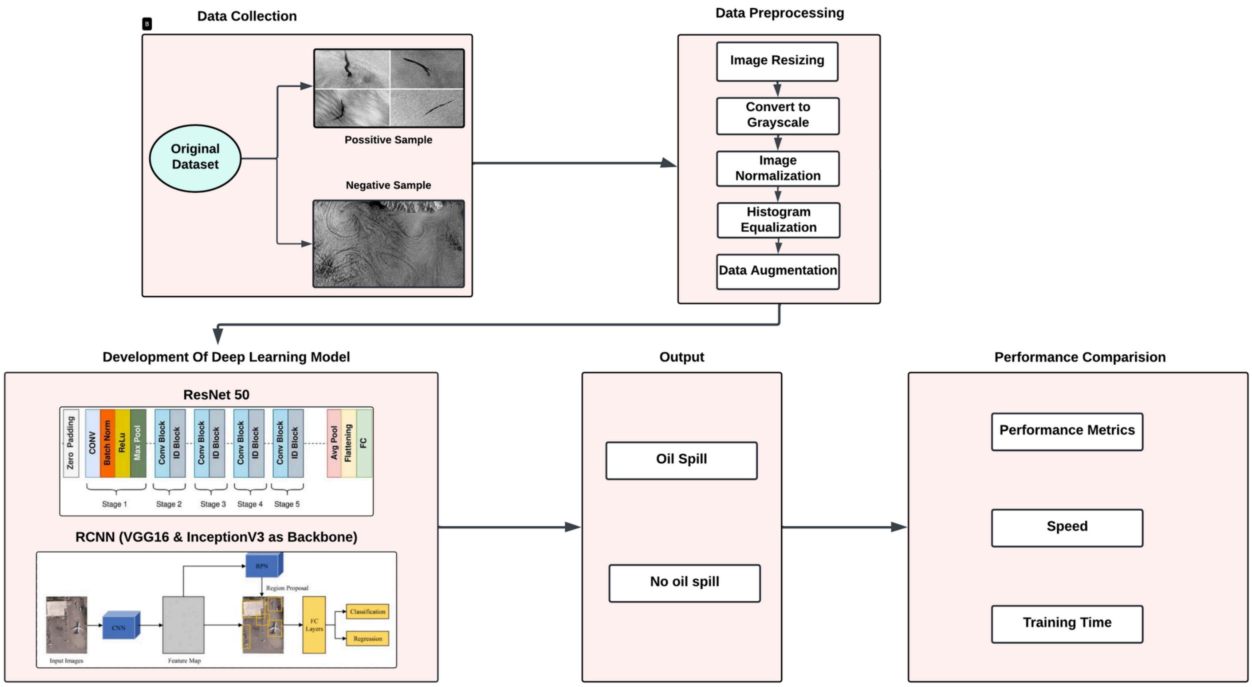
GPU Memory:

- Recommended: NVIDIA GPU (RTX 3080) for faster training and inference.

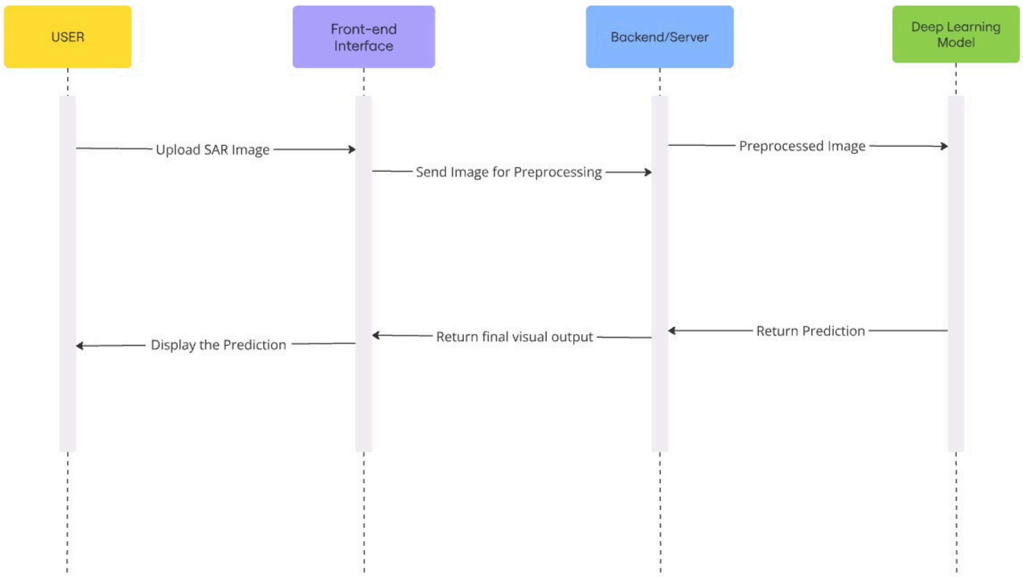
Additional Hardware:

- A stable internet connection is required for SAR dataset downloads and model updates, while backup storage ensures data redundancy and safekeeping.

Architecture Diagram

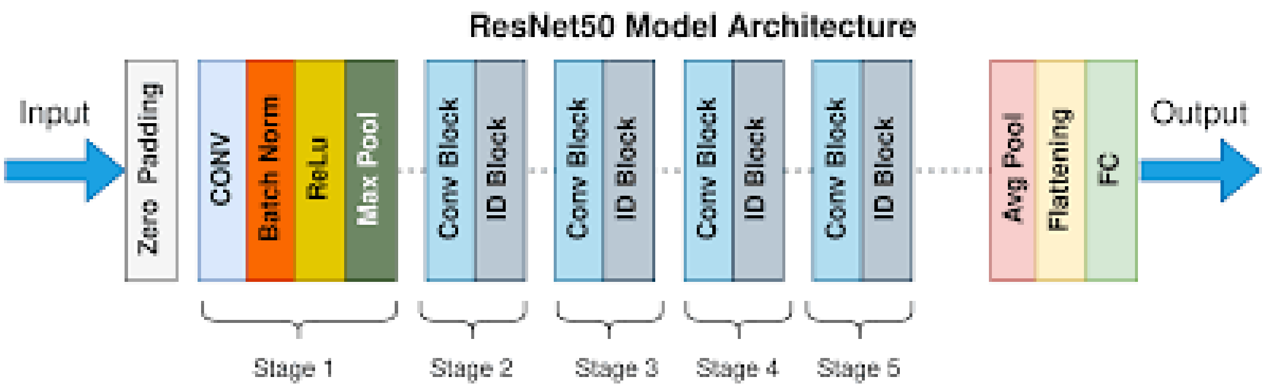


Sequence Diagram



Model Architecture

1. ResNet50



Model Architecture

1. ResNet50

- A 50-layer deep convolutional neural network designed to address vanishing gradient issues using residual learning.
- Uses skip connections to enable deeper network training without performance degradation.
- Composed of convolutional, batch normalization, ReLU and residual blocks to enhance feature extraction.
- Outputs a classification prediction of oil spill vs. non-oil spill.

Model Architecture

1. ResNet50

- Initial Convolution & Pooling Layers
- A 7×7 convolution layer followed by max pooling to extract basic features like edges.
- 4 Residual Blocks (Main Feature Extraction)
- 4 stages, each containing multiple residual blocks (1×1, 3×3, 1×1 convolutions).
- Skip connections allow gradients to flow easily, improving training efficiency.

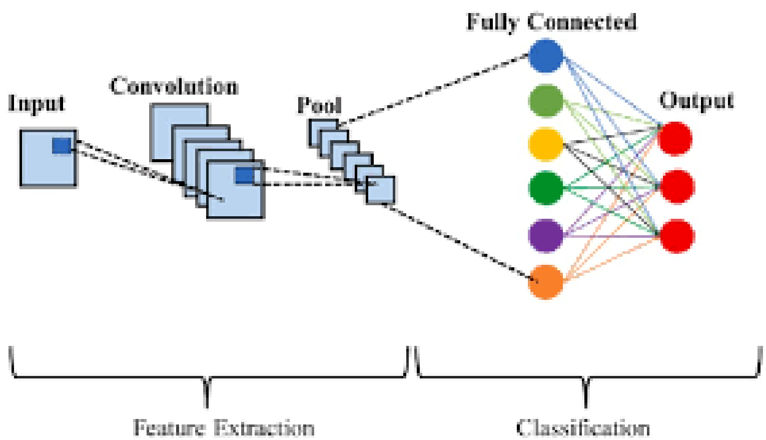
Model Architecture

1. ResNet50

- Global Average Pooling & Fully Connected Layer
- Fully connected layers classify the extracted features.
- Used for oil spill detection by extracting features from SAR images.
- Processes the entire image in a single forward pass to classify oil spills.

Model Architecture

2. RCNN (Region-based CNN)



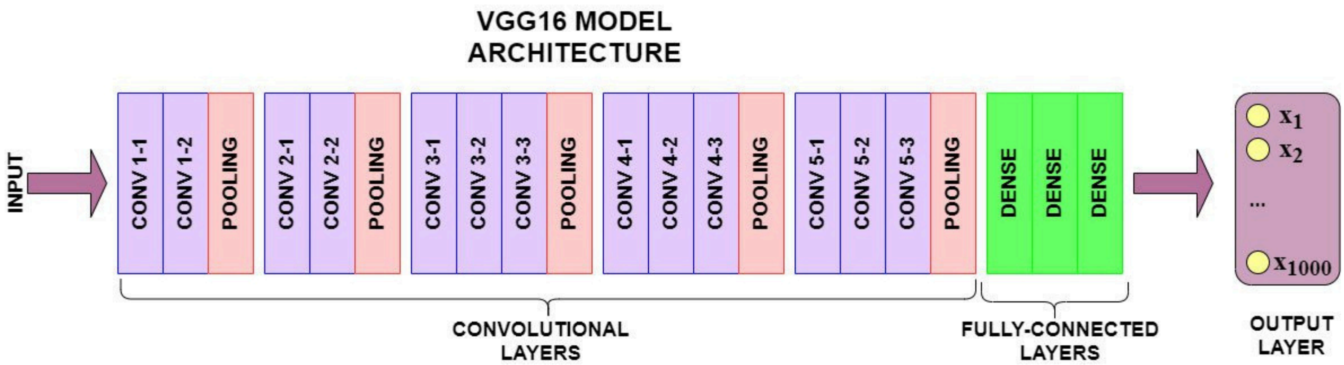
Model Architecture

2. RCNN (Region-based CNN)

- RCNN (Region-based CNN) detects oil spills in three main steps:
- Consists of three steps:
  a. Region Proposal Network (RPN) – Generates candidate regions of interest.
  b. Feature Extraction – Extracts features using a backbone CNN (VGG16 or InceptionV3).
  c. Classification & Refinement – Classifies regions and refines bounding boxes.

Model Architecture

3. VGG16 (as RCNN Backbone)



Model Architecture

3. VGG16 (as RCNN Backbone)

- Input: 224 × 224 RGB image
- Convolutional Layers: 13 convolutional layers with 3×3 filters, stride = 1
- Max-Pooling Layers: 5 max-pooling layers (2×2 kernel, stride = 2)
- Fully Connected (FC) Layers: 3 fully connected layers
- Activation Function: ReLU in all layers except output
- Output Layer: Softmax for classification

Model Architecture

3. VGG16 (as RCNN Backbone)

- A 16-layer CNN known for its simplicity and uniform architecture (3×3 conv layers).
- Acts as the feature extractor in RCNN, identifying oil spill characteristics.
- Uses max pooling layers for downsampling and fully connected layers for classification.

Model Architecture

3. VGG16 (as RCNN Backbone)

- Feature Extraction in RCNN:
- VGG16 serves as the backbone in RCNN to extract oil spill features from SAR images.
- It processes region proposals from the Region Proposal Network (RPN) to classify oil spill and non-oil spill areas.

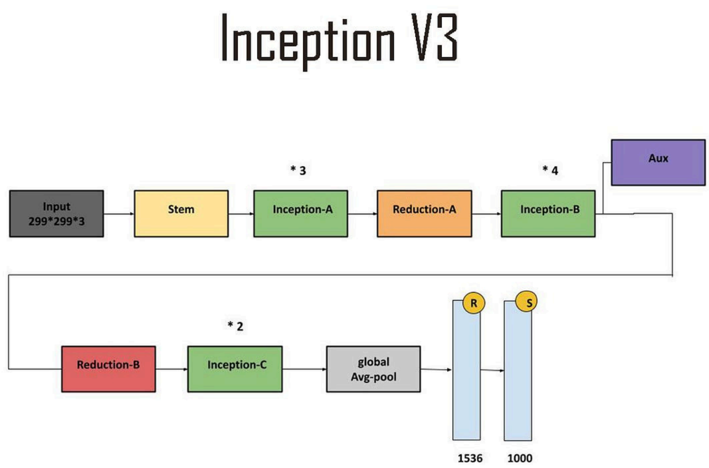
Model Architecture

3. VGG16 (as RCNN Backbone)

- Why VGG16?
- Deep yet simple architecture (16 layers) ensures efficient feature extraction.
- Smaller kernel sizes (3×3) help capture fine details in SAR images.
- Pretrained on ImageNet, making it effective for transfer learning.

Model Architecture

4. InceptionV3 (as RCNN Backbone)



Model Architecture

4. InceptionV3 (as RCNN Backbone)

- Stem (Initial Layers) – Convolutions, max pooling, and batch normalization for early feature extraction.
- Inception Modules – Parallel 1×1, 3×3, 5×5 convolutions and pooling layers to capture multi-scale features.
- Grid Reduction – Strided convolutions and pooling layers to reduce spatial dimensions efficiently.

Model Architecture

4. InceptionV3 (as RCNN Backbone)

- Auxiliary Classifiers – Intermediate layers to help with gradient propagation during training.
- Final Classification – Fully connected layers followed by Softmax for prediction.
- Uses multiple kernel sizes (1x1, 3x3, 5x5) in parallel to capture multi-scale features.
- Enhances detection accuracy by capturing both local and global image features.

Model Architecture

4. InceptionV3 (as RCNN Backbone)

- In this project, InceptionV3 is used as the backbone for RCNN, where it acts as a feature extractor for detecting oil spills in SAR images.
- The Region Proposal Network (RPN) first identifies potential oil spill regions.

Model Architecture

4. InceptionV3 (as RCNN Backbone)

- These regions are passed through InceptionV3, which extracts relevant spatial patterns from SAR images.
- The extracted features are then classified as oil spill or non-oil spill, helping in accurate detection.
- Compared to other models, RCNN with InceptionV3 captures multi-scale features, but is computationally heavier than VGG16.

Modules

1. Data Acquisition and Preprocessing

- SAR images are obtained from sources like ESA’s Sentinel-1 from CSIRO.
- Image resizing standardizes image dimensions for uniform input across models.
- Grayscale conversion converts images to a single channel, preserving essential features while reducing complexity.

Modules

- Image normalization scales pixel values to improve model convergence.
- Histogram equalization enhances image contrast by redistributing pixel intensity values, making oil spills more distinguishable.
- Data Augmentation applies transformations (rotation, flipping, etc.) to enhance model generalization.

Modules

2.Feature Extraction and Classification

- RCNN with VGG16 Backbone
  - Uses VGG16 for feature extraction, leveraging its deep architecture to capture spatial features.
  - Balances accuracy and computational efficiency while focusing on precise region classification.

Modules

- RCNN with InceptionV3 Backbone
  - InceptionV3 extracts multi-scale features using its inception modules, improving detection accuracy.
- ResNet-50(Residual Networks)
  - Extracts features and classifies regions using deep layers to learn robust representations.
  - Skip connections prevent vanishing gradients, allowing deeper learning and improved accuracy in classification.

Modules

- **Training Process**
  - **Dataset Preparation**
    - A labeled dataset of SAR images containing both oil spills and no oil spills is required.
  - **Loss Function**
    - Cross-entropy loss is used to optimize pixel-wise accuracy, ensuring precise classification .

Modules

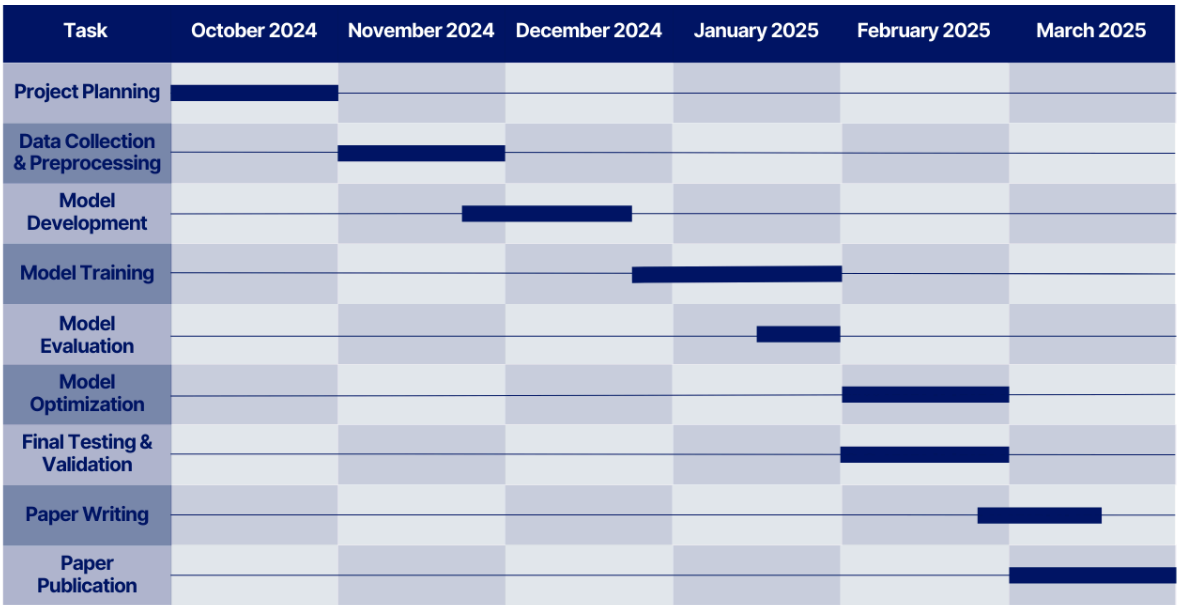
3.Comparison and Evaluation

- Prediction Comparison:
  - The outputs of ResNet-50, RCNN with VGG16, and RCNN with InceptionV3 are compared using key metrics: accuracy, precision, recall and F1-score.
  - The predicted oil spill regions are analyzed to assess how well each model performs under different conditions.

Work breakdown & responsibilities

- **Data Pre-Processing**
  Sradha Shajan
- **Model Design using ResNet-50**
  Saurav Krishnan
- **Model Design using RCNN**
  Niranjan S & Wivin Winny

Gantt Chart



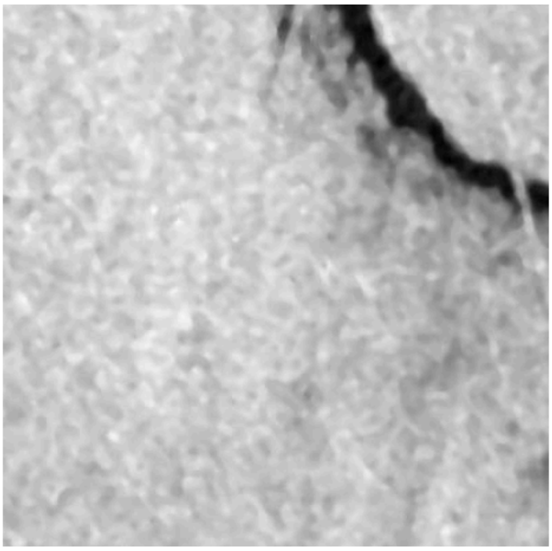
Risk & challenges

1. **Data Limitations:** High-quality annotated SAR datasets are scarce, and manual labeling is time-consuming.
2. **Model Generalization:** Environmental variations (sea state, weather) make distinguishing spills from ocean features challenging, increasing false detections.
3. **Computational & Real-Time Constraints:** Large SAR datasets demand high computational power, and real-time processing faces scalability and latency issues.

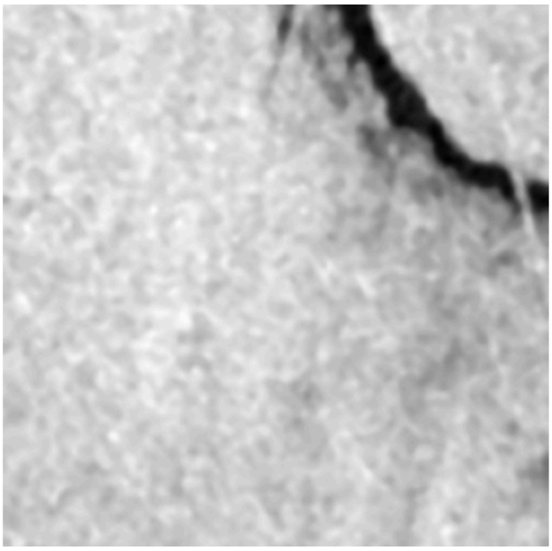
Results

Preprocessing

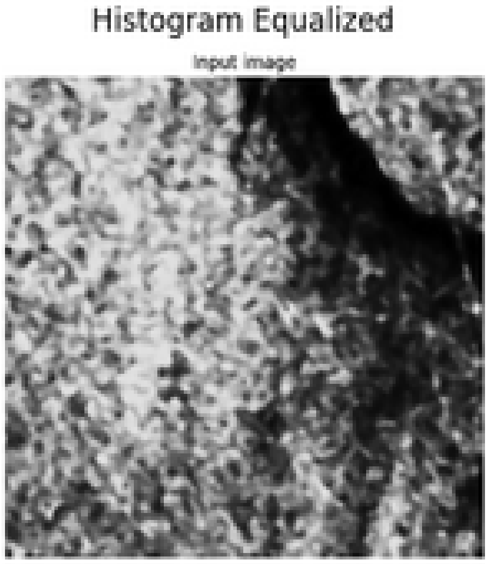
Gray Scaled Image



Normalization

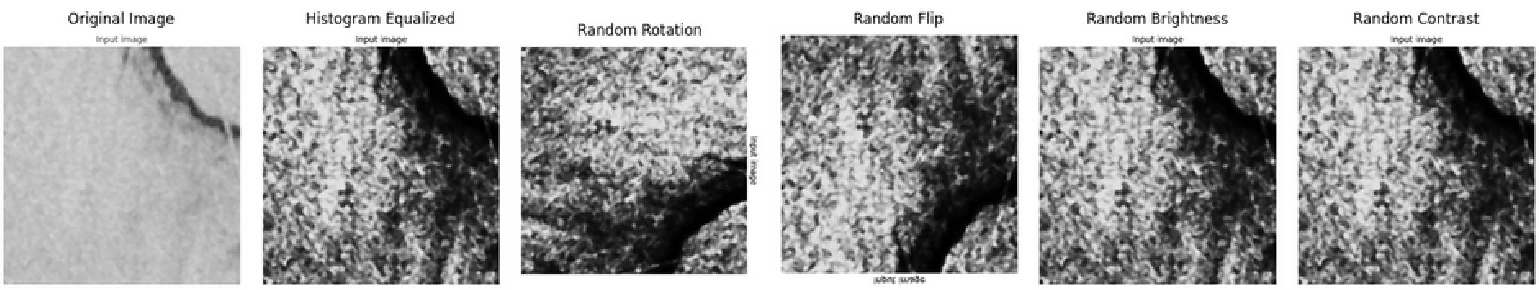


Preprocessing



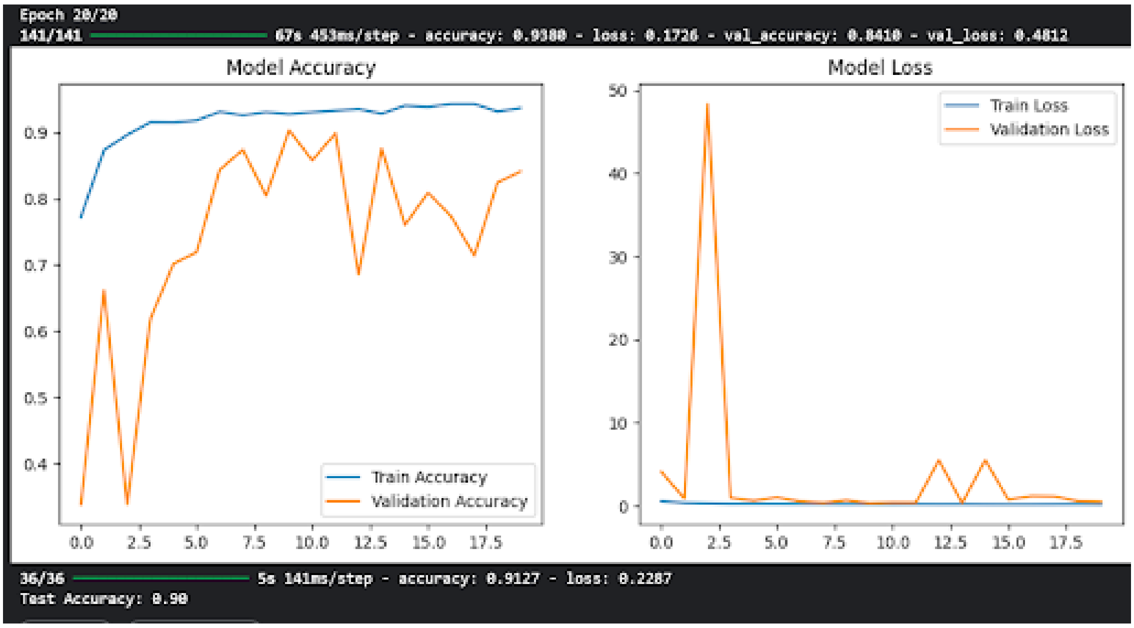
Histogram Equalization

Preprocessing



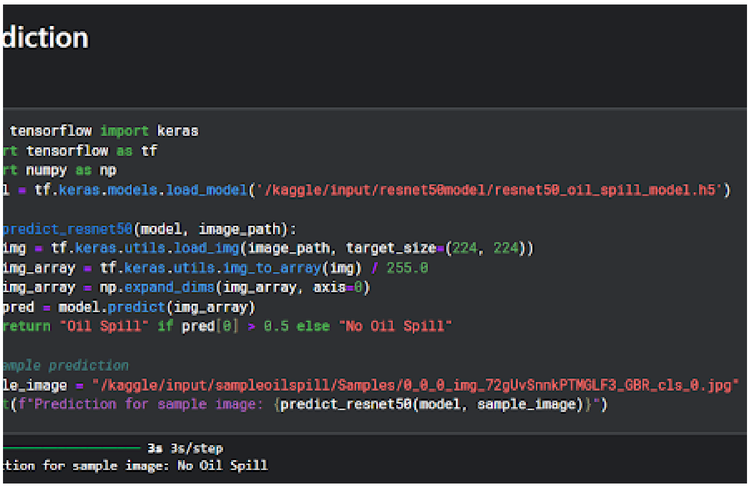
Data Augmentation

ResNet 50-Model Accuracy and Loss



ResNet 50-Prediction

Presence of Oilspill

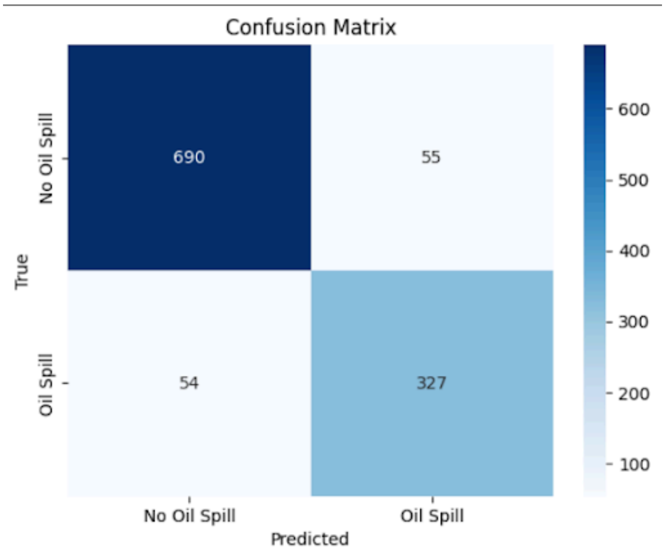


Absence of Oilspill

Evaluation Metrics

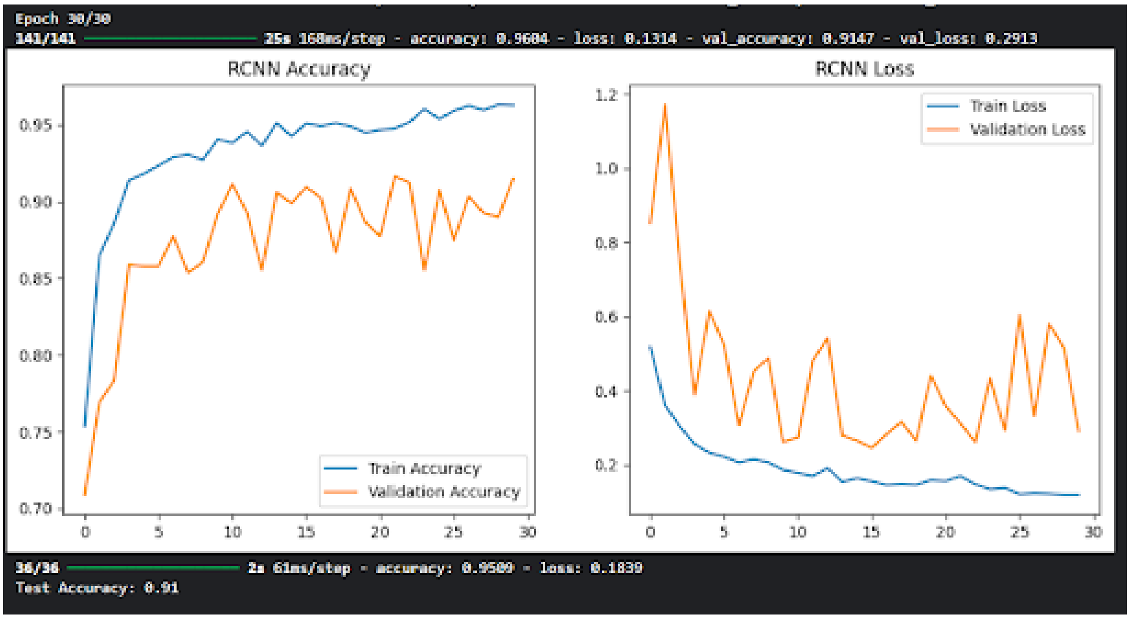
1. **Precision**
   - Measures how many predicted oil spills are actually correct.
   - Formula: Precision = TP / (TP + FP) (where TP = True Positives, FP = False Positives).
2. **Recall**
   - Measures how many actual oil spills are correctly detected.
   - Formula: Recall = TP / (TP + FN) (where FN = False Negatives).
3. **F1-Score**
   - Balances precision and recall for a more reliable performance measure.
   - Formula: F1-Score = 2 × (Precision × Recall) / (Precision + Recall).

ResNet-50: Confusion Matrix and Classification Report

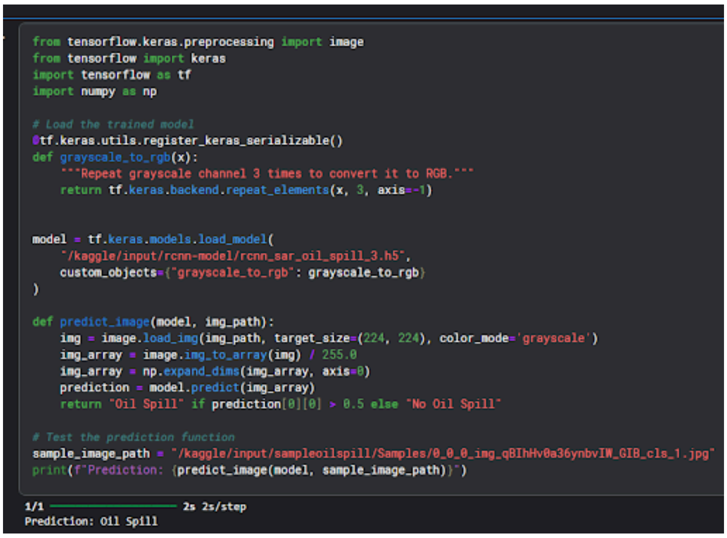



```
Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.93      0.93       745
           1       0.86      0.86      0.86       381

    accuracy                           0.90      1126
   macro avg       0.89      0.89      0.89      1126
weighted avg       0.90      0.90      0.90      1126

Precision: 0.8560
Recall: 0.8583
F1 Score: 0.8571
```

RCNN with VGG16-Model Accuracy and Loss

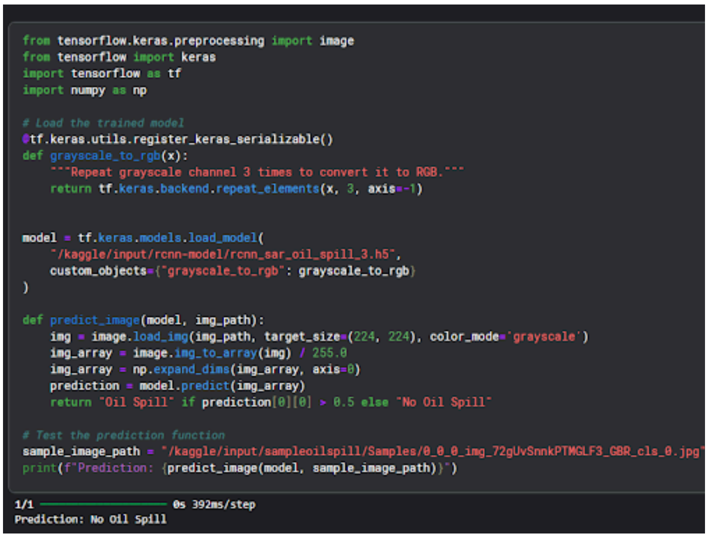


RCNN with VGG16-Prediction



```
from tensorflow.keras.preprocessing import image
from tensorflow import keras
import tensorflow as tf
import numpy as np

# Load the trained model
@tf.keras.utils.register_keras_serializable()
def grayscale_to_rgb(x):
    """Repeat grayscale channel 3 times to convert it to RGB."""
    return tf.keras.backend.repeat_elements(x, 3, axis=-1)


model = tf.keras.models.load_model(
    "/kaggle/input/rcnn-model/rcnn_sar_oil_spill_3.h5",
    custom_objects={"grayscale_to_rgb": grayscale_to_rgb}
)

def predict_image(model, img_path):
    img = image.load_img(img_path, target_size=(224, 224), color_mode='grayscale')
    img_array = image.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    prediction = model.predict(img_array)
    return "Oil Spill" if prediction[0][0] > 0.5 else "No Oil Spill"

# Test the prediction function
sample_image_path = "/kaggle/input/sampleoilspill/Samples/0_0_0_img_qBIhHv0a36ynbvIW_GIB_cls_1.jpg"
print(f"Prediction: {predict_image(model, sample_image_path)}")

1/1 2s 2s/step
Prediction: Oil Spill
```

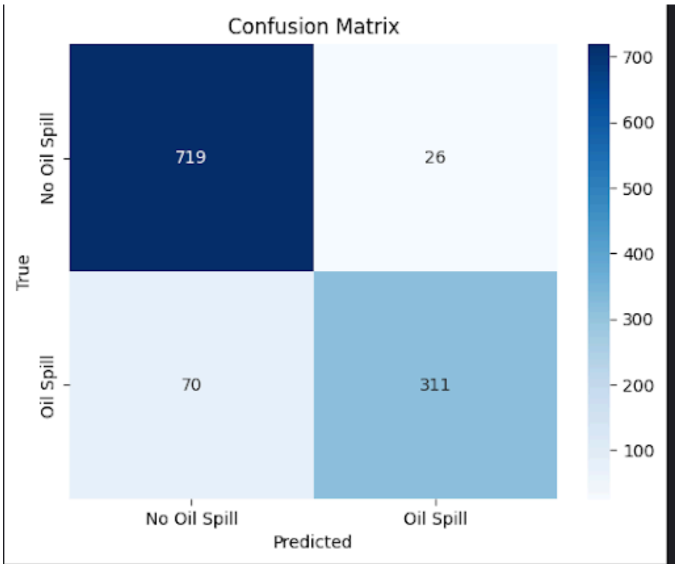
Presence of Oilspill



```
from tensorflow.keras.preprocessing import image
from tensorflow import keras
import tensorflow as tf
import numpy as np

# Load the trained model
@tf.keras.utils.register_keras_serializable()
def grayscale_to_rgb(x):
    """Repeat grayscale channel 3 times to convert it to RGB."""
    return tf.keras.backend.repeat_elements(x, 3, axis=-1)


model = tf.keras.models.load_model(
    "/kaggle/input/rcnn-model/rcnn_sar_oil_spill_3.h5",
    custom_objects={"grayscale_to_rgb": grayscale_to_rgb}
)

def predict_image(model, img_path):
    img = image.load_img(img_path, target_size=(224, 224), color_mode='grayscale')
    img_array = image.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    prediction = model.predict(img_array)
    return "Oil Spill" if prediction[0][0] > 0.5 else "No Oil Spill"

# Test the prediction function
sample_image_path = "/kaggle/input/sampleoilspill/Samples/0_0_0_img_72gUvSnnkPTMGLF3_GBR_cls_0.jpg"
print(f"Prediction: {predict_image(model, sample_image_path)}")

1/1 0s 392ms/step
Prediction: No Oil Spill
```

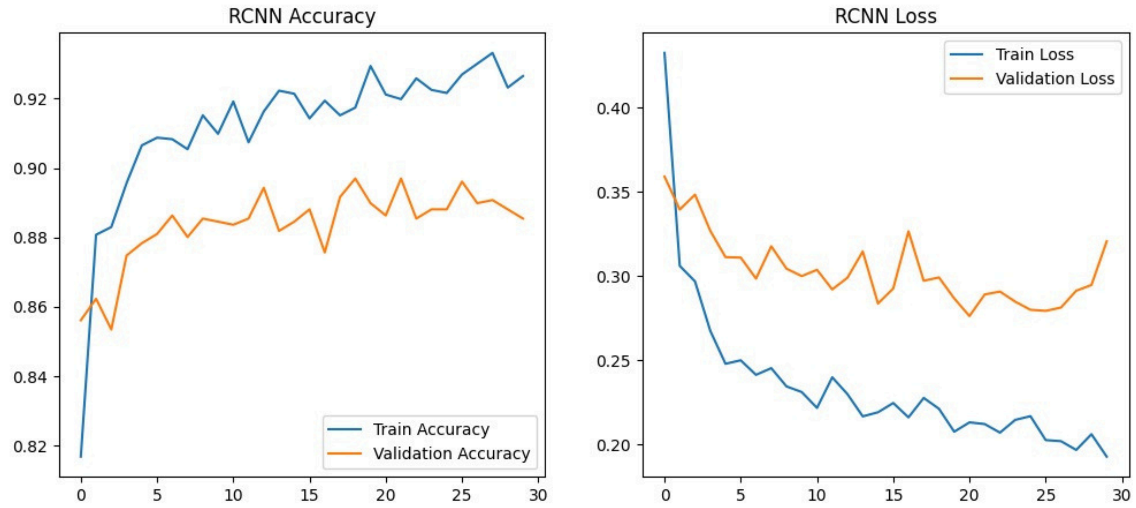
Absence of Oilspill

RCNN with VGG16: Confusion Matrix and Classification Report




```
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.97      0.94       745
           1       0.92      0.82      0.87       381

    accuracy                           0.91      1126
   macro avg       0.92      0.89      0.90      1126
weighted avg       0.92      0.91      0.91      1126

Precision: 0.9228
Recall: 0.8163
F1 Score: 0.8663
```


RCNN with InceptionV3-Model Accuracy and Loss




```
36/36 ━━━━━━━━━━━━━━━━━━━━ 4s 120ms/step - accuracy: 0.9035 - loss: 0.2934
Test Accuracy: 0.89
```

RCNN with InceptionV3-Prediction

```python
from tensorflow.keras.preprocessing import image
from tensorflow import keras
import tensorflow as tf
import numpy as np

# Load the trained InceptionV3 model
@tf.keras.utils.register_keras_serializable()
def grayscale_to_rgb(x):
    """Repeat grayscale channel 3 times to convert it to RGB."""
    return tf.keras.backend.repeat_elements(x, 3, axis=-1)

model = tf.keras.models.load_model(
    "/kaggle/input/rcnn-inception-v3-model/rcnn_sar_oil_spill_inception.h5",
    custom_objects={"grayscale_to_rgb": grayscale_to_rgb}
)

def predict_image(model, img_path):
    img = image.load_img(img_path, target_size=(299, 299), color_mode='grayscale')  # InceptionV
    img_array = image.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    prediction = model.predict(img_array)
    return "Oil Spill" if prediction[0][0] > 0.5 else "No Oil Spill"

# Test the prediction function
sample_image_path = "/kaggle/input/sampleoilspill/Samples/0_0_0_img_qBIhHv0a36ynbvIW_GIB_cls_1.j
print(f"Prediction: {predict_image(model, sample_image_path)}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 6s 6s/step
Prediction: Oil Spill
```

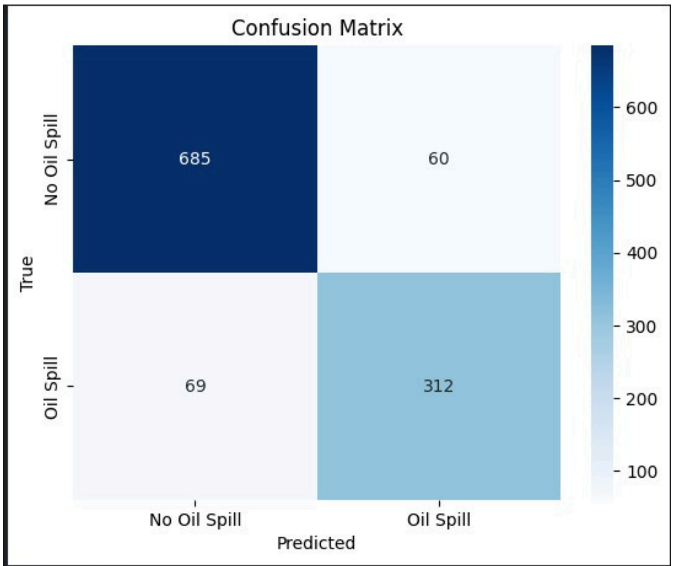
Presence of Oilspill

```python
from tensorflow.keras.preprocessing import image
from tensorflow import keras
import tensorflow as tf
import numpy as np

# Load the trained InceptionV3 model
@tf.keras.utils.register_keras_serializable()
def grayscale_to_rgb(x):
    """Repeat grayscale channel 3 times to convert it to RGB."""
    return tf.keras.backend.repeat_elements(x, 3, axis=-1)

model = tf.keras.models.load_model(
    "/kaggle/input/rcnn-inception-v3-model/rcnn_sar_oil_spill_inception.h5",
    custom_objects={"grayscale_to_rgb": grayscale_to_rgb}
)

def predict_image(model, img_path):
    img = image.load_img(img_path, target_size=(299, 299), color_mode='grayscale')  # InceptionV3 uses
    img_array = image.img_to_array(img) / 255.0
    img_array = np.expand_dims(img_array, axis=0)
    prediction = model.predict(img_array)
    return "Oil Spill" if prediction[0][0] > 0.5 else "No Oil Spill"

# Test the prediction function
sample_image_path = "/kaggle/input/sampleoilspill/Samples/0_0_0_img_ca9QOv0U0q6eU90w_PHI_cls_0.jpg"
print(f"Prediction: {predict_image(model, sample_image_path)}")
```

```
1/1 ━━━━━━━━━━━━━━━━━━━━ 9s 9s/step
Prediction: No Oil Spill
```

Absence of Oilspill

RCNN with InceptionV3: Confusion Matrix and Classification Report




```
Classification Report:
              precision    recall  f1-score   support

           0       0.91      0.92      0.91       745
           1       0.84      0.82      0.83       381

    accuracy                           0.89      1126
   macro avg       0.87      0.87      0.87      1126
weighted avg       0.88      0.89      0.89      1126

Precision: 0.8387
Recall: 0.8189
F1 Score: 0.8287
```

Comparison of 3 Deep learning Models

Performance Metrices	**ResNet-50**	**R-CNN with VGG-16 Backbone**	**R-CNN with Inception V3 Backbone**
Training Accuracy	0.9380	0.9604	0.9292
Testing Accuracy	0.90	0.91	0.89
Loss	0.1726	0.1314	0.1957
Precision	0.8560	0.9228	0.8387
Recall	0.8583	0.8163	0.8189
F1 Score	0.8571	0.8663	0.8287

Paper Publication and Funding

- **A Deep Learning Approach to Detect Ocean Oil Spills using SAR Images** - Communicated to Malaysian Journal of Computer Science - SCOPUS indexed
- 4 Lakhs funding received from MeiTy as part of India AI fellowship programme.

Conclusion

- In this work, a deep learning system for automatically identifying oil spills from Synthetic Aperture Radar (SAR) data is presented.
- Advanced SAR picture preprocessing methods, such as data augmentation, data normalization and histogram equalization, are used to improve detection accuracy.
- R-CNN with VGG-16 , RCNN with Inception-V3 and ResNet-50 are the three deep learning models that are compared and their performance is assessed using precision, recall, and F1-score.
- The testing accuracy achieved by models are—91% for R-CNN with VGG-16 backbone, 90% for ResNet-50 and 89% for R-CNN with Inception V3 backbone.
- Compared to ResNet-50 and R-CNN with Inception V3 backbone, it is evident that R-CNN with VGG-16 backbone is more reliable with greater accuracy to detect the oil spill

Future Scope

- Multi-Modal Fusion Approach: Integrating SAR, optical, and thermal infrared data to enhance classification accuracy and efficiency.
- Hybrid Models: Exploring transformer-based and attention-driven architectures for improved feature discrimination.
- Explainable AI (XAI): Developing interpretable models to provide transparency and insights into decision-making processes.
- Enhanced Multi-Sensor Integration: Strengthening real-time data fusion for more reliable oil spill detection.
- Real-Time Processing: Improving the speed and efficiency of detection systems for timely response and monitoring.

References

- Shaban, Mohamed, Reem Salim, Hadil Abu Khalifeh, Adel Khelifi, Ahmed Shalaby, Shady El-Mashad, Ali Mahmoud, Mohammed Ghazal, and Ayman El-Baz. "A deep-learning framework for the detection of oil spills from SAR data." Sensors 21, no. 7 (2021): 2351.

- Krestenitis, M., Orfanidis, G., Ioannidis, K., Avgerinakis, K., Vrochidis, S. and Kompatsiaris, I., 2019. Oil spill identification from satellite images using deep neural networks. Remote Sensing, 11(15), p.1762.

References

- Zeng, Kan, and Yixiao Wang. "A deep convolutional neural network for oil spill detection from spaceborne SAR images." Remote Sensing 12, no. 6 (2020): 1015.

- Matkan, A. A., M. Hajeb, and Z. Azarakhsh. "Oil spill detection from SAR image using SVM based classification." The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 40 (2013): 55-60.

References

- Bianchi, F.M., Espeseth, M.M. and Borch, N., 2020. Large-scale detection and categorization of oil spills from SAR images with deep learning. Remote Sensing, 12(14), p.2260.

- Tong, Shengwu, et al. "Multi-feature based ocean oil spill detection for polarimetric SAR data using random forest and the self-similarity parameter." Remote Sensing 11.4 (2019): 451.

Thank You

Appendix B: Vision, Mission, Programme Outcomes and Course Outcomes

Vision, Mission, Programme Outcomes and Course Outcomes

Institute Vision

To evolve into a premier technological institution, moulding eminent professionals with creative minds, innovative ideas and sound practical skill, and to shape a future where technology works for the enrichment of mankind.

Institute Mission

To impart state-of-the-art knowledge to individuals in various technological disciplines and to inculcate in them a high degree of social consciousness and human values, thereby enabling them to face the challenges of life with courage and conviction.

Department Vision

Department Mission

Programme Outcomes (PO)

Engineering Graduates will be able to:

1. Engineering Knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

2. Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

3. Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

4. Conduct investigations of complex problems: Use research-based knowledge including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

5. Modern Tool Usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

6. The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal, and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

7. Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

8. Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

9. Individual and Team work: Function effectively as an individual, and as a member or leader in teams, and in multidisciplinary settings.

10. Communication: Communicate effectively with the engineering community and with society at large. Be able to comprehend and write effective reports documentation. Make effective presentations, and give and receive clear instructions.

11. Project management and finance: Demonstrate knowledge and understanding of engineering and management principles and apply these to one's own work, as a member and leader in a team. Manage projects in multidisciplinary environments.

12. Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

Programme Specific Outcomes (PSO)

Course Outcomes (CO)

Appendix C: CO-PO-PSO Mapping

COURSE OUTCOMES:

After completion of the course, the student will be able to:

SL.NO	DESCRIPTION	Bloom's Taxonomy Level
CO1	Model and solve real-world problems by applying knowledge across domains (Cognitive knowledge level:Apply).	Level3: Apply
CO2	Develop products, processes, or technologies for sustainable and socially relevant applications. (Cognitive knowledge level:Apply).	Level 3: Apply
CO3	Function effectively as an individual and as a leader in diverse teams and comprehend and execute designated tasks. (Cognitive knowledge level:Apply).	Level 3: Apply
CO4	Plan and execute tasks utilizing available resources within timelines, following ethical and professional norms (Cognitive knowledge level:Apply).	Level 3: Apply
CO5	Identify technology/research gaps and propose innovative/creative solutions (Cognitive knowledge level:Analyze).	Level 4: Analyze
CO6	Organize and communicate technical and scientific findings effectively in written and oral forms (Cognitive knowledge level:Apply).	Level 3: Apply

CO-PO AND CO-PSO MAPPING

CO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	2	2	2	1	2	2	2	1	1	1	1	2	3	3	1
CO2	2	2	2		1	3	3	1	1		1	1	3	3	1
CO3									3	2	2	1	3	3	2
CO4					2			3	2	2	3	2	3	3	3
CO5	2	3	3	1	2							1	3	3	2
CO6					2			2	2	3	1	1	3	3	3

3/2/1: high/medium/low

JUSTIFICATIONS FOR CO-PO MAPPING

Mapping	Level	Justification
101003/CS822U.1-PO1	M	Knowledge in the area of technology for project development using various tools results in better modeling.
101003/CS822U.1-PO2	M	Knowledge acquired in the selected area of project development can be used to identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions.
101003/CS822U.1-PO3	M	Can use the acquired knowledge in designing solutions to complex problems.
101003/CS822U.1-PO4	M	Can use the acquired knowledge in designing solutions to complex problems.
101003/CS822U.1-PO5	H	Students are able to interpret, improve, and redefine technical aspects for design of experiments, analysis, and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS822U.1-PO6	M	Students are able to interpret, improve, and redefine technical aspects by applying contextual knowledge to assess societal, health, and consequential responsibilities relevant to professional engineering practices.
101003/CS822U.1-PO7	M	Project development based on societal and environmental context solution identification is the need for sustainable development.
101003/CS822U.1-PO8	L	Project development should be based on professional ethics and responsibilities.
101003/CS822U.1-PO9	L	Project development using a systematic approach based on well-defined principles will result in teamwork.
101003/CS822U.1-PO10	M	Project brings technological changes in society.
101003/CS822U.1-PO11	H	Acquiring knowledge for project development gathers skills in design, analysis, development, and implementation of algorithms.

101003/CS822U.1-PO12	H	Knowledge for project development contributes engineering skills in computing and information gatherings.
101003/CS822U.2-PO1	H	Knowledge acquired for project development will also include systematic planning, developing, testing, and implementation in computer science solutions in various domains.
101003/CS822U.2-PO2	H	Project design and development using a systematic approach brings knowledge in mathematics and engineering fundamentals.
101003/CS822U.2-PO3	H	Identifying, formulating, and analyzing the project results in a systematic approach.
101003/CS822U.2-PO5	H	Systematic approach is the tip for solving complex problems in various domains.
101003/CS822U.2-PO6	H	Systematic approach in the technical and design aspects provides valid conclusions.
101003/CS822U.2-PO7	H	Systematic approach in the technical and design aspects demonstrates the knowledge of sustainable development.
101003/CS822U.2-PO8	M	Identification and justification of technical aspects of project development demonstrates the need for sustainable development.
101003/CS822U.2-PO9	H	Apply professional ethics and responsibilities in engineering practice of development.
101003/CS822U.2-PO11	H	Systematic approach also includes effective reporting and documentation, which gives clear instructions.
101003/CS822U.2-PO12	M	Project development using a systematic approach based on well-defined principles will result in better teamwork.
101003/CS822U.3-PO9	H	Project development as a team brings the ability to engage in independent and lifelong learning.

101003/CS822U.3-PO10	H	Identification, formulation, and justification in technical aspects will be based on acquiring skills in design and development of algorithms.
101003/CS822U.3-PO11	H	Identification, formulation, and justification in technical aspects provides the betterment of life in various domains.
101003/CS822U.3-PO12	H	Students are able to interpret, improve, and redefine technical aspects with mathematics, science, and engineering fundamentals for the solutions of complex problems.
101003/CS822U.4-PO5	H	Students are able to interpret, improve, and redefine technical aspects with identification, formulation, and analysis of complex problems.
101003/CS822U.4-PO8	H	Students are able to interpret, improve, and redefine technical aspects to meet the specified needs with appropriate consideration for public health and safety, and the cultural, societal, and environmental considerations.
101003/CS822U.4-PO9	H	Students are able to interpret, improve, and redefine technical aspects for design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
101003/CS822U.4-PO10	H	Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools for better products.
101003/CS822U.4-PO11	M	Students are able to interpret, improve, and redefine technical aspects by applying contextual knowledge to assess societal, health, and consequential responsibilities relevant to professional engineering practices.
101003/CS822U.4-PO12	H	Students are able to interpret, improve, and redefine technical aspects for demonstrating the knowledge of, and need for sustainable development.

101003/CS822U.5-PO1	H	Students are able to interpret, improve, and redefine technical aspects, apply ethical principles, and commit to professional ethics and responsibilities and norms of the engineering practice.
101003/CS822U.5-PO2	M	Students are able to interpret, improve, and redefine technical aspects, communicate effectively on complex engineering activities with the engineering community and society at large, such as being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
101003/CS822U.5-PO3	H	Students are able to interpret, improve, and redefine technical aspects to demonstrate knowledge and understanding of the engineering and management principle in multidisciplinary environments.
101003/CS822U.5-PO4	H	Students are able to interpret, improve, and redefine technical aspects, recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.
101003/CS822U.5-PO5	M	Students are able to interpret, improve, and redefine technical aspects in acquiring skills to design, analyze, and develop algorithms and implement those using high-level programming languages.
101003/CS822U.5-PO12	M	Students are able to interpret, improve, and redefine technical aspects and contribute their engineering skills in computing and information engineering domains like network design and administration, database design, and knowledge engineering.
101003/CS822U.6-PO5	M	Students are able to interpret, improve, and redefine technical aspects and develop strong skills in systematic planning, developing, testing, implementing, and providing IT solutions for different domains, which helps in the betterment of life.

101003/CS822U.6-PO8	H	Students will be able to associate with a team as an effective team player for the development of technical projects by applying the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
101003/CS822U.6-PO9	H	Students will be able to associate with a team as an effective team player to identify, formulate, review research literature, and analyze complex engineering problems.
101003/CS822U.6-PO10	M	Students will be able to associate with a team as an effective team player for designing solutions to complex engineering problems and design system components.
101003/CS822U.6-PO11	M	Students will be able to associate with a team as an effective team player, use research-based knowledge and research methods including design of experiments, analysis, and interpretation of data.
101003/CS822U.6-PO12	H	Students will be able to associate with a team as an effective team player, applying ethical principles and committing to professional ethics and responsibilities and norms of the engineering practice.
101003/CS822U.1-PSO1	H	Students are able to develop Computer Science Specific Skills by modeling and solving problems.
101003/CS822U.1-PSO2	H	Students develop Programming and Software Development Skills by designing algorithms and applying software development practices to model and solve real-world problems across multiple domains, ensuring industry-relevant solutions.
101003/CS822U.1-PSO3	L	Students develop Professional Skills by applying computer science fundamentals to model and solve real-world problems, fostering innovation and research to create impactful solutions that address societal needs.

101003/CS822U.2-PSO1	H	Students are able to develop Computer Science Specific Skills by designing solutions for complex engineering problems that contribute to sustainable and socially relevant applications.
101003/CS822U.2-PSO2	H	Developing products, processes or technologies for sustainable and socially relevant applications can promote Programming and Software Development Skills.
101003/CS822U.2-PSO3	L	Developing products, processes, or technologies for sustainable and socially relevant applications enhances Professional Skills by promoting research, innovation, and entrepreneurship to address societal needs.
101003/CS822U.3-PSO1	H	Students develop Computer Science Specific Skills by effectively collaborating in diverse teams to design solutions for complex engineering problems, applying core computational principles to address national grand challenges and socially relevant applications.
101003/CS822U.3-PSO2	H	Teamwork enhances Programming Skills by ensuring efficient algorithm design and software development best practices.
101003/CS822U.3-PSO3	M	Working in a team can result in the effective development of Professional Skills.
101003/CS822U.4-PSO1	H	Students develop Computer Science Specific Skills by efficiently planning and executing tasks within given constraints, applying core computational principles to solve complex engineering problems while adhering to ethical and professional standards.
101003/CS822U.4-PSO2	H	Planning and executing tasks efficiently enhances Programming and Software Development Skills by ensuring structured development, timely delivery, and adherence to industry standards.
101003/CS822U.4-PSO3	H	Planning and scheduling can result in the effective development of Professional Skills.

101003/CS822U.5-PSO1	H	Students are able to develop Computer Science Specific Skills by creating innovative solutions to problems.
101003/CS822U.5-PSO2	H	Identifying gaps enhances Programming Skills by fostering innovation in algorithms and software development.
101003/CS822U.5-PSO3	M	Identifying gaps enhances Professional Skills by fostering research, innovation, and entrepreneurial solutions.
101003/CS822U.6-PSO1	H	Effective communication enhances Computer Science Skills by conveying complex technical solutions clearly and accurately.
101003/CS822U.6-PSO2	H	Organizing and communicating technical and scientific findings can help in the effective development of Professional Skills..
101003/CS822U.6-PSO3	H	Effective communication enhances Programming Skills by clearly presenting algorithms and software solutions.