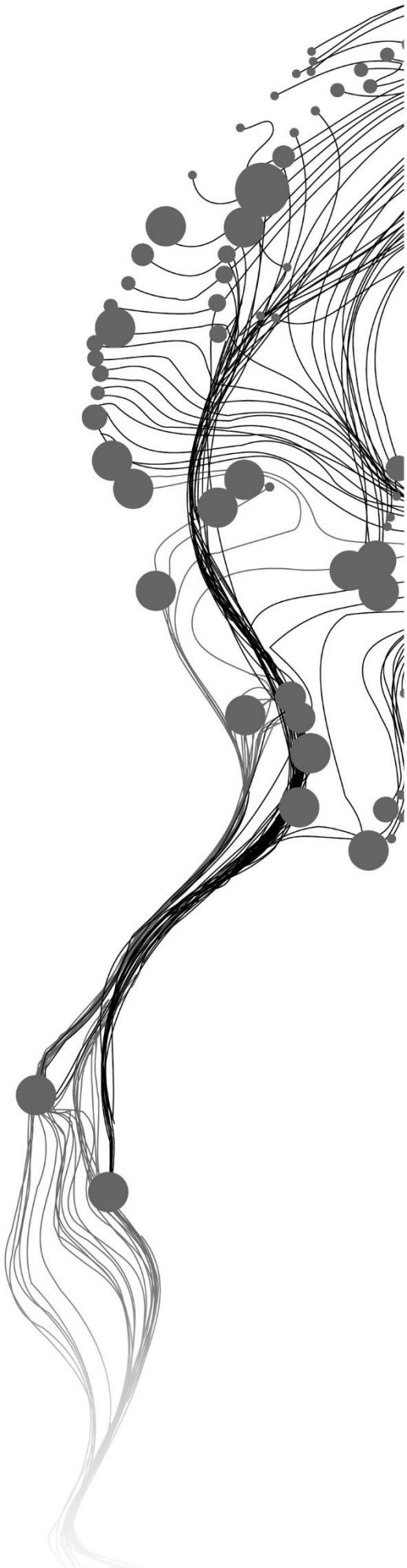
DEEP LEARNING-BASED SEMANTIC SEGMENTATION FOR DETECTING MARINE OIL SPILL

ANDI IBRAHIM
JULY, 2024

SUPERVISORS:
Dr. Ir. Suhyb Salama
Dr. Mariana Belgiu



DEEP LEARNING-BASED SEMANTIC SEGMENTATION FOR DETECTING MARINE OIL SPILL

ANDI IBRAHIM
Enschede, The Netherlands, July, 2024

Thesis submitted to the Faculty of Geo-Information Science and Earth Observation of The University of Twente in partial fulfilment of the requirements for the degree of Master of Science in Geo-Information Science and Earth Observation.
Specialization: Water Resources and Environmental Management

SUPERVISORS:
Dr. Ir. Suhyb Salama
Dr. Mariana Belgiu

THESIS ASSESSMENT BOARD:
Prof. Dr. Daphne van der Wal (Chair)
Dr. Antonio Moreno-Rodenas (External Examiner, Deltares)

ABSTRACT

The rapid expansion of global maritime industries has increased the risk of oil spills, which can cause extensive marine pollution. Due to their rapid spread, oil spills often negatively impact marine biodiversity and coastal communities. Such accidents require rapid detection to mitigate their rapidly spreading negative impact. In recent years, deep learning (DL) techniques for oil spill detection (OSD) have advanced rapidly, particularly through the use of SAR images. However, obtaining a large number of labelled images that correspond to ground truth data of oil spill events is challenging. This limitation in input data presents a main obstacle to achieving optimal semantic segmentation results in the development of DL-based.

This study investigates optimizing DL-based models by configuring relevant hyperparameters to accommodate limited datasets and adapt the oil spill appearance in SAR images for semantic segmentation in OSD. This study implements a Fully Convolutional Network-Dilated Kernel with 6 layers (FCN-DK6) model with fewer network parameters by optimizing the image preprocessing hyperparameter, specifically the patch size. Additionally, this study develops a hybrid CNN-XGBoost model based on convolutional networks from the FCN-DK6 model and a pre-existing XGBoost algorithm as an alternative solution in optimizing the FCN-DK6 model for OSD. This hybrid model has the potential to augment small datasets by generating multiple synthetic feature maps through its CNN component, which are then classified by XGBoost algorithm.

The results show that the FCN-DK6 model despite having fewer network parameters can perform semantic segmentation in OSD on a limited dataset by optimizing a patch size to 512x512. This configuration achieves an optimal IoU accuracy of 51.03% for oil spills, outperforming smaller patch size settings. Furthermore, reducing the number of classes during training the model significantly improves the FCN-DK6 mode's ability to segment oil spills by 98.80%. Meanwhile, the hybrid CNN-XGBoost model optimally performs OSD with semantic segmentation by configuring CNN filters and obtaining the optimal number of feature maps for feeding the XGBoost classifier. Combining optimal feature map settings with XGBoost hyperparameter tuning proves beneficial by achieving optimal performance with an IoU accuracy of 51.40% for OSD in a reduced class number scenario. While optimizing the FCN-DK6 model and developing the hybrid CNN-XGBoost model result in positive findings, the accuracy of OSD still needs more improvement to achieve reliable performance.

This study provides scientific advancements by demonstrating the use of optimizing hyperparameters, particularly patch size in the DL-based model, to enhance OSD using a limited dataset. Additionally, the hybrid CNN-XGBoost model addresses input data limitations in OSD by generating multiple synthetic feature maps from a single input image to expand the dataset and fine-tune hyperparameters. These contributions advance deep learning techniques for marine environmental monitoring, particularly in data resource-limited scenarios.

Keywords: oil spill detection, FCN, hybrid model, CNN-XGBoost, XGBoost, limited dataset, SAR, machine learning, convolutional layer, hyperparameters, feature extraction.

ACKNOWLEDGEMENTS

Reaching this point has been a result of relentless, step-by-step hard work, accomplished by fully embracing the process. The constant support from those around me turned heavy steps into strong strides that allowed me to run this far.

First and foremost, I would like to thank LPDP and the government of the Republic of Indonesia for believing in me as a scholarship awardee that enables me to pursue my studies abroad and supporting my educational journey.

I would like to express my sincere gratitude to my two supervisors, Dr. Ir. Suhyb Salama and Dr. Mariana Belgiu, for their support, guidance, and kindness throughout my thesis journey. Their constructive feedback and encouragement always motivated me to think critically about my work. I would like to extend my gratitude to Prof. Dr. Daphne van der Wal for chairing my thesis journey. Your support in my potential to complete my work has been invaluable.

To the friends of mine at ITC, especially WREM cohort 2022, thank you for being supportive of each other as we accomplished every quartile together. Huge thanks to the Indonesian family: Ghaly, Clava, Nasir, Rifqi, Wibi, Sry, Salsa, Amy, and Ganda for the laughter, free food, and hanging out that always made me feel supported and warm, just like at home.

I sincerely thank my parents, who have always been my pillars of strength, for their relentless prayer and unwavering support from afar. To my dearest wife, Dita Wulandari, her constant support and enduring faith in me have been my greatest source of strength throughout my journey in the Netherlands.

" Our greatest glory is not in never falling, but in rising every time we fall "

- Confucius

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF EQUATIONS

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
CNN	Convolutional Neural Network
CM	Confusion Matrix
DK	Dilated Kernel
DL	Deep Learning
ESA	European Space Agency
FCN	Fully Convolutional Neural Network
IoU	Intersection over Union
ML	Machine Learning
OSD	Oil Spill Detection
RGB	Red-Green-Blue
RS	Remote Sensing
SAR	Synthetic Aperture Radar
SDG	Stochastic Gradient Descent
SDGs	Sustainable Development Goals
SST	Sea Surface Temperature
SVM	Support Vector Machine
XGBoost	eXtreme Gradient Boosting

1. INTRODUCTION

1.1. Background

1.1.1. The Causes and Impacts of Oil Spills on Marine

The oil spill is an environmental pollution that mostly occurs at sea caused either by human activities or natural phenomena. Over 50% of oil spill events in the marine environment are caused by human activities associated with maritime industries (Kvenvolden & Cooper, 2003). The expansion of worldwide maritime industries has risen the risk of oil spills, such as offshore oil operations and petroleum tanker transport, increases the risk of oil spills in the oceans (J. Chen et al., 2019; Galieriková & Materna, 2020). Various accidents in oil operations, such as pipeline ruptures and drilling rig failures, can result in oil spills into the sea. Equally, accidents often occur in the tanker transport sectors, involving collisions and shipwrecks that cause oil to be released into the water column. Accidents involving ships or vessels are the primary cause of oil spills in the ocean, and they are responsible for about 73% of total incidents (Nagalakshmi et al., 2018). Meanwhile, natural oil slicks also occur when oil seeps out from the ocean floor due to internal earth activities, causing the escape of liquid hydrocarbons into the water column. Additionally, organic materials excreted by sea creatures can potentially create natural oil slicks. These biogenic slicks occur when lipids and other organic substances are released into the water and form a thin film on the sea surface (Gade et al., 1998; Song et al., 2020). These oil spill causes, especially those related to human activities, have been discovered for many years and remain a pressing issue today, and they still pose threats to the marine environment.

The impact of oil spills is directly associated with a decline in marine biodiversity and adversely affects coastal communities. The toxic nature of the oil substances can lead to the immediate death of marine creatures and long-term damage to their body systems that increases the risk of population decline. Mangroves and coral reef ecosystems that support various marine life are often severely affected by exposure to oil pollution. This damage can provoke potential collapses in local ecosystems. For instance, around 8 million liters of crude oil spilled in the Caribbean Sea, Panama in 1986 affecting 307 ha mangroves. A report showed that 34% of them began showing defoliations within two months following the spill, and 18% of oiled trees eventually died (Duke et al., 1997). In another situation, coastal communities that rely on fishing and tourism suffer economically as fish stocks decrease and polluted beaches discourage visitors. The contamination of coastal waters also poses health risks to residents that affect their quality of life. Ordinioha & Brisibe (2013) and Atonye (2023) revealed that oil spills cause several acute health problems such as respiratory issues, skin irritation, gastrointestinal stress, extensive epidermolysis, and esophagitis. The losses marine ecosystems and human communities face from oil spill events highlight the need for effective response strategies. Effective monitoring effort is needed to mitigate the risk of the far-reaching effects of such incidents.

1.1.2. Remote Sensing and Semantic Segmentation for Oil Spill Detection

An oil spill monitoring system is required to take control of the spreading effects in an undesired manner. The remote sensing (RS) technology of space-based satellite sensors has already proved its ability to perform large-scale monitoring of oil spills on the sea surface. There are diverse types of RS data to be used for oil spill monitoring. Active microwave sensors are widely used in oil spill monitoring since they are able to capture images both during the day and at night, and in any weather condition. Unlike optical sensors, active microwaves are not affected by cloud cover, making them a preferred choice for capturing accurate imaging of oil spill events (Sommervold et al., 2023). An active sensor that uses a radar type of waveform to map the Earth's surface is known as a Synthetic Aperture Radar (SAR). This technology identifies dark spots in

sea areas as oil slicks. Such dark features are associated with low backscattering coefficients showing smooth surfaces due to the oil presence suppressing sea surface roughness (Alpers et al., 2017). Nevertheless, the dark spots observed in SAR imagery do not solely represent oil spills. They can also be attributed to various other factors, such as low wind areas, natural films, wind fronts, wind shadows near islands, upwelling events, and algae blooms (Alpers et al., 2017; Solberg, 2012).

The various natural phenomena that closely mimic the appearance of oil spills in SAR images are called look-alikes. This similarity complicates the accurate identification of actual oil spills from other natural environmental features. Despite the advantages of SAR for oil spill detection (OSD), the presence of look-alikes makes the use of SAR sensors more challenging. Lentini et al. (2022) observed look-alikes in an oil spill event in three sites of Brazilian waters using SAR images with patch classifier methods. Although the proposed method could identify most false positive targets, such as rain cells, ships, and low wind conditions, it misclassified biofilm slicks as look-alikes due to their high similarity to oil spills. This misclassification highlights the well-known challenge in OSD in distinguishing between actual oil spills and other natural features. It shows that there is still a need for further development of algorithms or models in the future.

In recent years, the integration of deep learning (DL) with RS has become a popular method for developing OSD models. Semantic segmentation using DL applied to RS images offers advantages for OSD efforts by modelling oil spread at the pixel level, which provides precise information on the extent of oil spills on the sea surface. Practically, this method labels each pixel in an image and groups it with others of the same class to provide a complete understanding of the entire scene (Garcia-Garcia et al., 2017). Detailed pixel-wise classification is important for OSD tasks, as it allows the precise delineation to localize oil objects from their surroundings (Krestenitis et al., 2019b). Sudha & Vijendran (2021) stated that DL models for semantic segmentation tasks effectively discriminate oil spills and make this technique highly effective for identifying oil spills. Fan & Liu (2023) presented an efficient multi-task generative adversarial network for localizing oil spills on three different SAR datasets. The study showed that the proposed frameworks, which include semantic segmentation, are generally effective in distinguishing between actual oil spills and look-alikes. This situation indicates that DL is one of the important keys to improving the accuracy of applications such as automatic OSD and locating oil objects. Localizing oil spills at the pixel level helps in accurately determining the position of oil spills in the real world, which can provide early warnings and support the decision-making process.

1.2. Problem Statement

OSD with SAR is increasingly incorporating DL techniques to achieve high accuracy. DL techniques with multiple layers of depth in neural networks are recognized for learning complex patterns and extracting high-level features through end-to-end learning. These advancements allow the precise pixel-wise classification in an image with more detailed and accurate identification of oil spills. Nevertheless, successful execution of DL requires a large dataset for training the model, where the amount of dataset is crucial to enable the model to generalize complex features on target objects. Generalization in DL refers to how effectively a model can apply its training to predict unseen data. However, the large amount of data becomes a common challenge to provide when developing DL models, especially for OSD.

The problem of implementing DL in OSD efforts lies in the difficulty of collecting ground truth data in terms of both quantity and quality. The dynamic nature of the ocean, such as strong currents, influences the rapid movement of oil on the sea, making it challenging to capture accurate ground truth information or direct observation that matches the satellite sensing period (Kolokoussis & Karathanassi, 2018). Many studies rely on experts to manually label oil spills by visual interpretation in satellite images, often supplemented by adaptive thresholding techniques (Arslan et al., 2023; de Moura et al., 2022; Pelizzari & Bioucas-Dias, 2007; Singha et al., 2012; Yang et al., 2022; Zakzouk et al., 2021). This process provides labelled images needed for segmentation in OSD. However, this manual labelling is time-consuming, prone

to human error, and may not be scalable for large datasets. Thus, this situation leads to the scarcity of labelled image issues and becomes a challenge for developing effective DL models because training them with small datasets potentially leads to overfitting (Pasupa & Sunhem, 2016). Overfitting occurs when a model memorizes training data rather than generalizing, which results in poor performance on unseen data. DL models with relatively lightweight network architectures and additional fine-tuning hyperparameters can perform well even with limited samples (Feng et al., 2019; Y. Wang et al., 2018; Younis & Keedwell, 2019). Besides, machine learning (ML) seems to be capable of dealing with a small number of training datasets for classification tasks (Chawla et al., 2003; Sheykhmousa et al., 2020). ML techniques have performed well with smaller datasets by relying on feature inputs to capture the most relevant information for achieving high classification performance (Khalid et al., 2014). However, in ML, feature extraction often requires expertise and manual interpretation. This process can be time-consuming and may introduce bias, as the quality of extracted features depends on the human's judgment.

Based on the aforementioned problems, this study focuses on handling limited data for semantic segmentation in OSD. First, it will implement the Fully Connected Network (FCN) model capability with simpler network parameters with hyperparameter configurations to accommodate limited data for semantic segmentation in detecting oil spills. Second, this study proposes a hybrid model that combines convolution network elements from FCN-DK6 to automatically extract image features and the existing boosting ML algorithm to deal with limited data for semantic segmentation tasks in OSD.

1.3. Research Gap

Current approaches to semantic segmentation in OSD often rely on large datasets to achieve high accuracy and generalization. The FCN-DK6 with a simpler DL network shows promise in handling limited datasets. This makes it less prone to overfitting, which overfitting is a common problem when training on small datasets. This study initially implemented the FCN-DK6 adopting learning-related hyperparameters from Najmi et al. (2022) as the initial setting applied to OSD with SAR data. These initial settings may require adjustment when applied to different and limited datasets. Since the model's performance can potentially degrade without proper hyperparameter tuning for specific datasets. The FCN-DK6 model's performance in OSD with the initial hyperparameter setting serves as a baseline for comparing the proposed solutions in achieving optimal accuracy in OSD. Two alternative approaches are proposed to address this potential shortfall. First, a hyperparameter tuning process includes learning-related and image-preprocessing hyperparameters to optimize the FCN-DK6 model specifically for accurately detecting oil spills. Many studies focus on tuning learning-related hyperparameters such as learning rate, activation function, batch size, number of epochs, and optimizer type (Baydaroğlu & Demir, 2024; Benamrou et al., 2020; Koroniotis et al., 2022; Nair et al., 2022; Rawat et al., 2021; Stofa et al., 2020; Yaloveha et al., 2022) but only a few consider the impact of image preprocessing hyperparameters on optimizing DL models to accurately detect oil spills. The second alternative is a hybrid model approach to address the potential limitations of the initial FCN-DK6 setting in achieving optimal segmentation for OSD using limited datasets. It utilizes the convolutional network elements of FCN-DK6 as a model for automatic feature extraction, combined with XGBoost for segmentation tasks. XGBoost is highly effective for pixel-wise classification with structured data, such as tabular data where relationships between features are clearly defined, but may not capture complex patterns in images without advanced feature extraction. Convolutional Neural Network (CNN) can automatically augment images by creating multiple synthetic image features to expand the size of data. Combining these models can potentially enhance performance on small datasets in detecting oil spills by enhancing the number of relevant feature representations. Few studies have explored hybrid models that integrate CNN with XGBoost for semantic segmentation. The application of CNN-XGBoost in this research addresses challenges in OSD-specific data and cases by leveraging its capabilities to deal with limited datasets.

1.4. Research Objectives and Questions

The main objective of this research is to optimize the DL-based models for semantic segmentation tasks in OSD using a limited dataset by configuring their relevant hyperparameters. The main objective is then divided into specific objectives with the following research questions.

1.4.1. Sub-objective 1

To implement and examine FCN-DK6 model for handling limited datasets in semantic segmentation for OSD.

1.1. What is the performance of the FCN-DK6 model with initially defined learning-related hyperparameters in detecting oil spills using a limited dataset?

1.2. What are the main hyperparameters that affect model performance in detecting oil spill, in particular when using a limited dataset?

1.4.2. Sub-objective 2

To develop and evaluate a hybrid model combining two techniques: CNN element from FCN-DK6 model and existing XGBoost algorithm for semantic segmentation using limited datasets in OSD.

2.1. What are the optimal convolutional network elements from FCN-DK6 and the XGBoost hyperparameters to achieve the best performance in semantic segmentation for OSD?

2.2. What are the inefficacies of the hybrid model for OSD ?

1.5. Relations to Water Resources and Environmental Management

Water resources and environmental management focus on the sustainable use and protection of water resources both on land and in the ocean. In particular, marine management which includes water quality monitoring and marine ecosystem protection is related to controlling marine pollution. This priority is emphasized in Sustainable Development Goals 14 (Life Below Water), which focuses on conserving and sustainably using the oceans. SDGs 14 prioritizes reducing marine pollution in the first place because our seas are currently threatened by pollution from environmentally unfriendly human activities. Sources of this pollution include industrial waste, domestic waste, plastic, oil spills, etc. Therefore, it is urgently necessary to take every possible action to prevent, protect against, and reduce the spread of marine pollution.

For these reasons, large-scale detection of oil spills using satellite imagery becomes an effective strategy to mitigate the impact of such events. Therefore, this research aims to take proactive steps to prevent the severe impacts of oil spills by developing a reliable method for detecting them with satellite images and computer vision techniques, even with limited data available.

2. LITERATURE REVIEW

2.1. Remotely Sensed Data for Oil Spill Detection

Over the past few decades, RS data have been widely used to detect and monitor oil spill events. Among the various RS technologies, such as aerial photography, drones, LiDAR, and hyperspectral imaging, satellites equipped with active sensors such as Synthetic Aperture Radar (SAR) turn out to be one of the most effective tools in this regard. Consequently, its ability to continuously acquire data day and night and under all almost weather conditions makes SAR a preferred choice for many oil spill monitoring operations (Girard-Ardhuin et al., 2005; Kostianoy et al., 2006; Singha et al., 2013). In particular, the effectiveness of SAR lies in its ability to detect variations in the sea surface roughness. The sea surface appears bright in SAR images due to the capillary waves on the ocean reflecting radar energy (Fingas & Brown, 2011). On the contrary, the presence of oil spills on the sea surface dampens these capillary waves, and then they create smoother areas that appear as dark spots in SAR images (Guo et al., 2018), as illustrated in Figure 2.1. These contrast feature appearances allow the rapid identification of oil spill locations and extents through SAR images. Despite these advantages, one still major issue is the presence of natural phenomena resembling oil spills in SAR images, such as natural films, rain cells, wind front areas, low wind areas, etc, known as look-alikes (Alpers et al., 2017; Solberg, 2012) and illustrated in Figure 2.1. These look-alikes have caused false positives and misidentifications using both human interpretation techniques or even advanced algorithms.

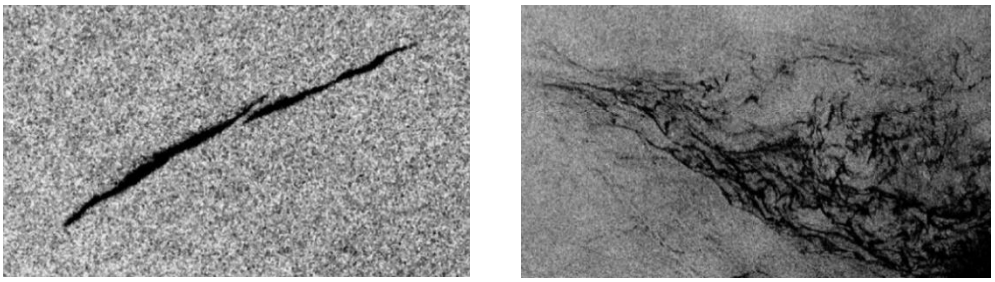


Figure 2.1 An example of oil spill (left) and look-alikes (right) in a Synthetic Aperture Radar (SAR) image (source: Krestenitis et al. (2019b)

Many previous studies on oil spills have highlighted the use of SAR imagery as an effective tool for detecting oil presence in the ocean. For instance, Topouzelis (2008), Chehresa et al. (2016), and Dutta et al. (2018) extracted oil spills from SAR images in their respective studies. Some studies even gave particular focus to the unique characteristics of SAR images as a base for developing effective methods in OSD. El-Zaart & Ghosn (2013) developed a thresholding-based algorithm derived from grey levels and texture on SAR, while Y. Li & Li (2010) utilized the identical gamma distribution on the pixels from SAR intensity to introduce a marked point process algorithm. As we have moved into the era of artificial intelligence, ML has been employed to develop automatic and robust models that integrate with SAR data for detecting oil spills. Robust refers to the models' ability to consistently perform accurately and reliably under diverse conditions. Magri et al. (2021) demonstrated a remarkable Support Vector Machine (SVM) algorithm for discriminating oil spills, especially when training samples are limited. Some studies also find that the SVM algorithm is useful for OSD using SAR images (Matkan et al., 2013; H. Zhou & Peng, 2018; Zou et al., 2016). Conceição et al. (2021) developed a new open-source methodology for detecting oil spills using a random forest (RF)-based algorithm. Tong et al. (2019) utilized RF to develop a multi-feature-based model for detecting oil spills. They argued that RF can suppress the classification error caused by the fact that in almost all SAR images, water or sea surface represents the most part. These studies highlight the growing use of SAR images and advanced detection methods for effectively identifying oil spills. This trend promises continuous development that can be conducted and synced with the advancement of the modern technology era like DL era.

2.2. Deep Learning-based for Oil Spill Detection

DL has improved the detection and monitoring of oil spills. For instance, Jiao et al. (2019) compared OSD using Unmanned Aerial Vehicles (UAVs) with DL and traditional ML. They found that Deep CNN achieved a mean average precision (mAP) accuracy of 99.30%, significantly outperforming Adaboost's 12.11%. Additionally, their DL detection system reduced the cost of OSD by 57.2% compared to traditional manual inspection process. DL uses complex computational models, several layers of processing, and hidden layers (Lecun et al., 2015). Through these techniques, the models can automatically extract and learn various features directly from the training data they are given. According to Vasconcelos et al., (2023), research into neural networks (NN) and multilayer texture analysis in OSD using CNN has increased significantly over the last 5 years. For instance, Krestenitis et al. (2019a) developed an early warning system for identifying oil spills by taking advantage of a deep CNN and applied it to SAR images. Meanwhile Das et al. (2023) demonstrated that CNN effectively extract features in images to support categorizing images into different classes in oil spill classification with high accuracy, more than 90%. The advancement of CNN led to the development of Fully Convolutional Networks (FCNs), which allow for classification at the pixel level, known as semantic segmentation. This technique is suitable for providing detailed mapping for oil spill identification.

2.3. Semantic Segmentation

Several works have adopted semantic segmentation with deep CNN and SAR images to detect oil slicks on the sea surface (Fan et al., 2023; Fan & Liu, 2023; Orfanidis et al., 2018; Sudha & Vijendran, 2021). In theory, semantic segmentation is part of the scene understanding that deals with pixel-based classification. Where the main task of pixel-wise segmentation is to provide a class label for each pixel in an entire image with a predefined class (Hao et al., 2020). This approach offers a detailed and precise mapping of multiple elements within an image and provides category information at the pixel level, as illustrated in Figure 2.2. This approach has become essential to the marine field, especially for OSD. It models oil spread at the pixel level in images to reflect real phenomena. Further, the modelling results can provide valuable information on each pixel associated with the precise location and extent of oil objects on the sea surface (Gallego et al., 2019). With the advancement of computer vision technology through the emergence of DL, a new approach offers promising performance improvements for image semantic segmentation tasks. One such approach is the Fully Convolutional Network (FCN), which adopts end-to-end learning for pixel-wise prediction.

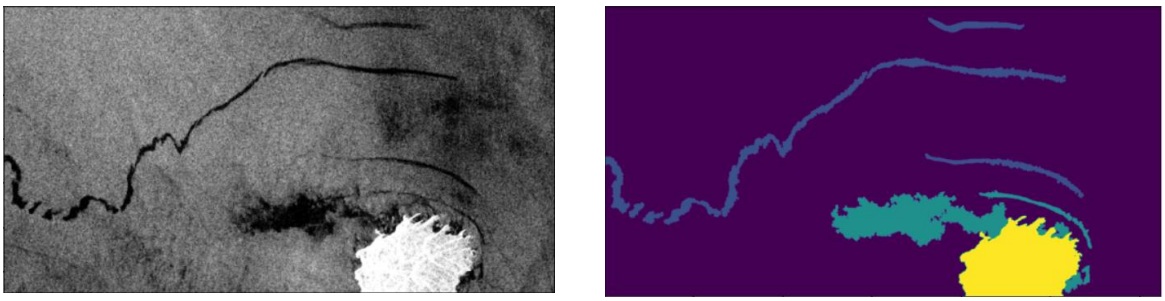


Figure 2.2 An example of semantic segmentation of oil spills (right) with default colour coding and Synthetic Aperture Radar (SAR) image (left). Purple is for sea surface, blue is for oil spills, green is for look-alikes, yellow is for land (source: author)

2.3.1. Fully Convolutional Network for Oil Spill Detection

Long et al. (2015) introduced the first DL technique for semantic segmentation using FCN. They replaced the traditional fully connected layers in CNN with convolutional layers to make dense predictions at the pixel level that maintain the same dimension as the input. By doing so, the FCN model can preserve spatial hierarchies and capture complex details across all pixels in the entire image. The FCN architecture utilized a series of convolutional and pooling layers to downsample the input image, which reduces the spatial dimension while capturing high-level features. Afterwards, it is followed by upsampling layers, which are

known as deconvolutional layers, to restore the pixel-wise prediction to the original image resolution (Long et al., 2015). This architecture allows the FCN to maintain detailed spatial information throughout the segmentation process. Figure 2.3 illustrates the sequence of the FCN architecture model in an oil spill prediction case. Since its introduction, FCN has become a foundational technique for further research and development of image semantic segmentation with DL-based approaches.

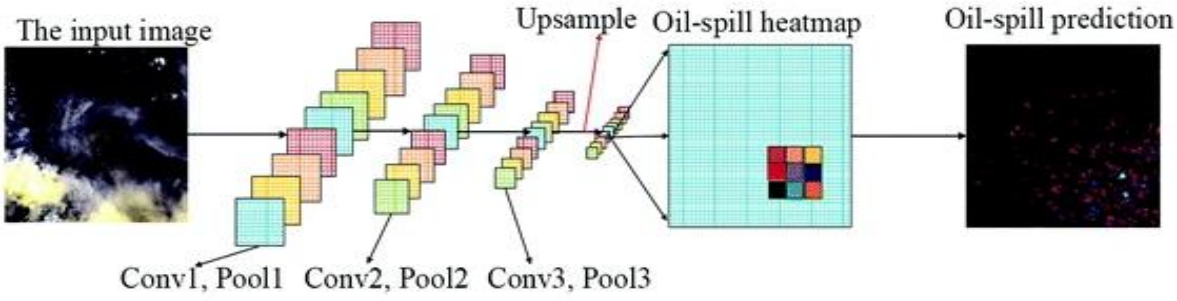


Figure 2.3 Fully Connected Network model framework for Oil Spill Detection (Source: Li, Yang, et al. (2018))

Several studies have examined and developed the FCN model-based for segmenting at the pixel level on oil spills. These studies highlight the effectiveness of FCN in accurately identifying oil spill regions. For instance, Mahmoud et al. (2023) and Song et al. (2024) demonstrated that FCN trained with SAR images can achieve high accuracy with more than 90% accuracy for isolating oil objects. Meanwhile, Li et al. (2021) examined the use of FCN trained with small data training and showed that FCN can moderately predict oil spills in SAR images. Subsequently, several researchers, including Li, Yang, et al. (2018) and Shamili et al. (2022) developed a new model based on FCN. They created a model that combines FCN with ResNet and Googlenet, with the result showing that it is feasible to use the advanced model based on FCN for detecting oil spills with substantial accuracy. Shanmukh et al. (2024) developed two stages for predicting oil spills by taking advantage of FCN with skip connections and combining it with a U-net model to precisely identify oil spill pixels within patches.

With the versatility of FCN, several studies outside of the oil pollution field have also developed novel models based on FCN. Bi et al. (2017) utilized FCN as a base to introduce a new model, stacked FCN architecture with multi-channel learning (SFCN-ML). Then, they continue to develop dual-path adversarial learning (DAL) with FCN-based architecture (Bi et al., 2018). Wang et al. (2020) proposed a CRACK-FCN model that combines multi-scale structured forests for edge detection (SFD) for crack image segmentation. Additionally, Persello & Stein (2017) presented FCN model using dilated kernel (FCN-DK) for slum mapping. It is claimed that the proposed model is able to learn features with a limited number of networks and parameters. Table 2.1 compares various DL models for semantic segmentation tasks that highlight the FCN-DK6 model has a smaller network depth and parameters than other models. This indicates that FCN-DK6 model is more lightweight and potentially more effective for handling small datasets due to its lower complexity. However, this advantage may come at the cost of a reduced ability to capture complex features compared to deeper models like Segnet and U-net.

Table 2.1 Comparison of network depth and total parameters in various Deep Learning models for semantic segmentation

Model	Depth of Networks	Total Parameters
Segnet	36 layers	19,265,989
DeepLabv3	27 layers	11,855,221
PSPnet	26 layers	15,965,189
U-net	23 layers	34,456,869
FCN-DK6	12 layers	117,445

*the numbers in the table above obtained through self-observation by the author

2.4. Convolutional Neural Network

CNN is the most utilized DL networks in computer vision for image classification and recognition. According to Lecun et al. (1998), the main idea of CNN was to use convolutional layers to handle the spatial hierarchies of features from input data. Convolutional refers to the convolution operation, which applies filters to detect patterns such as edges and textures to create feature maps that highlight the complex patterns across images (Yamashita et al., 2018; C. Yu et al., 2023). This enables CNN to automatically and effectively extract features from input images. A typical CNN model consists of three types of layers and one classifier that includes convolutional, pooling, a few fully connected layers, and a softmax (Albelwi & Mahmood, 2017), as illustrated in Figure 2.4. Convolutional layers use a series of learnable kernels of filters which aim to extract low-level features such as edges, textures, colours, etc., and output new feature maps (Albelwi & Mahmood, 2017). While, pooling layers perform to reduce the spatial dimensions of the input feature maps through a process known as downsampling, also reducing parameters to make a model more efficient (O'Shea & Nash, 2015). Fully connected layers perform to compile the output features from convolutional and pooling layers to interpret high-level features and make final decisions (Basha et al., 2020).

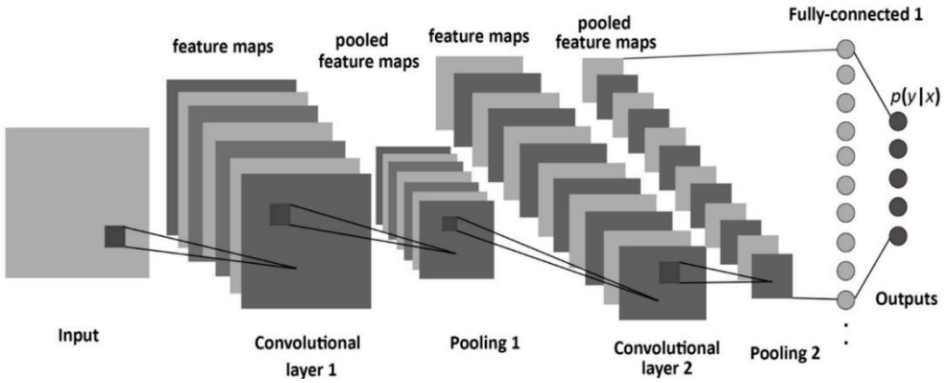


Figure 2.4. Convolutional Neural Network structure, consisting of convolutional, pooling, and fully connected layers (Source: (Albelwi & Mahmood, 2017)

2.5. eXtreme Gradient Boosting (XGBoost)

XGBoost is an ensemble learning method used to address various classification and regression problems. This classifier uses the gradient-boosting algorithm which sequentially builds new trees to predict the residuals or errors from the previous trees (G. Zhou et al., 2022). As a result, each step in the optimization process is dependent on the outcomes of the previous step. By applying the gradient descent algorithm, XGBoost minimizes loss when adding a new tree. Tree boosting is an effective ensemble learning approach that integrates several weak classifiers into a single strong classifier, it leads to improving classification performance (Ren et al., 2017). The final prediction is derived by combining the predictions of all the individual tress, as illustrated in Figure 2.5.

XGBoost controls model complexity and prevents overfitting through its objective function, which consists of two parts: the loss function and the regularization term (Tanha et al., 2020). The loss function works with gradient descent by iteratively adding weak learners, which corrects the previous error, ensuring accurate predictions (Shao et al., 2024). The regularization term penalizes complex models with L1 (lasso) providing noise resistance and robustness and L2 (ridge) offering lower computational complexity and faster processing (Zhang et al., 2022). This balance allows XGBoost to achieve both high performance and computational efficiency. The algorithm stands out for two key features: it computes faster than other gradient-boosting tools and performs exceptionally well in classification and regression modelling (Zhou et al., 2022). Additionally, XGBoost can handle non-linearities and interactions between input features.

With its robust capabilities, XGBoost has been widely used in many research fields, particularly in RS applications. Zhou et al. (2022) demonstrated that XGBoost achieved higher overall accuracy than the

spectral angle mapper method in identifying bamboo species using hyperspectral RS imagery. Zhen et al. (2024) proved that XGBoost is highly effective in detecting mangrove species using very high-resolution satellite imagery despite their similar spectral properties. Shao et al. (2024) compared the performance of Random Forest and XGBoost classifiers for mapping urban impervious surfaces using integrated optical and SAR images. They concluded that XGBoost outperforms Random Forest in terms of accuracy. These studies suggest that XGBoost offers robust performance and reliability for classification tasks, making it a useful option for various RS applications.

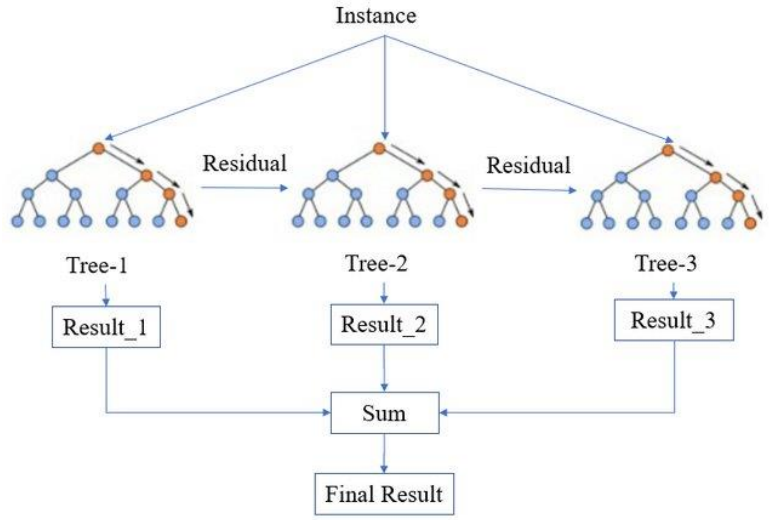


Figure 2.5 Simplified structure of XGBoost algorithm (Source: Wang et al. (2021))

2.6. Hybrid Model of Deep and Machine Learning in Semantic Segmentation

DL models, including Artificial Neural Network (ANN) and CNN perform well in image classification and segmentation. However, their limitations include a heavy reliance on large amounts of labelled training data and prone to overfitting due to their high capacity to learn complex patterns in training data. The fully connected layers in DL used for classification tasks typically account for nearly 80% of the model's total parameters, significantly increasing computational demands and memory usage (Jiang et al., 2018). This situation makes the models less efficient when deployed on devices with limited sources.

A few research studies have formulated the hybrid technique to substitute the fully connected layer in DL with ML classifiers to avoid overfitting and lower computational requirements. Nijhawan et al., 2018) worked with Landsat 8 images to map debris-covered glaciers. They leveraged the AlexNet pre-trained model with CNN-based for extracting deep features and classifying those features with random forests to accurately map debris. Lee et al. (2018) adopted the same technique by separating the feature extraction and output layers classifier (fully connected layer) of CNN, then substituting the output layer classifier with the AdaBoost algorithm. Their study proved that Boost-CNN improved CNN performance by 3%. Sudiana et al. (2023) utilized CNN with a 1-dimensional array for feature extraction from optical and SAR images to identify burned areas. The high-level features extracted from CNN were then classified with a random forest classifier, achieving an accuracy of 96% for optical images and 70% for SAR images. Kwak et al. (2021) studied early crop mapping using a hybrid classification approach that combines CNN-random forests using limited input data from UAVs. They found that the hybrid model increased accuracy by 7% compared to a standalone CNN model. These studies demonstrate the high potential of hybrid techniques in image classification and segmentation in various fields and purposes. Since such hybrid techniques have become promising for automatic feature extraction and classification of image-based semantic segmentation, this study follows a similar approach.

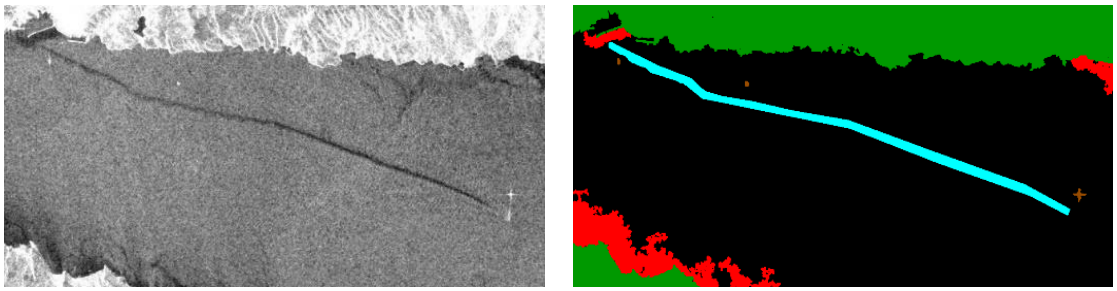
3. DATA AND METHODOLOGY

3.1. Dataset

This study uses two different datasets to train and evaluate the proposed model, including benchmark and Indonesian datasets. The benchmark dataset consists of extracted images from Sentinel-1 SAR scenes which contain oil spills and obtained from MKLab (Krestenitis et al., 2019b). This dataset was used to implement and evaluate the proposed model. Besides, the Indonesian dataset includes Sentinel-1 SAR images collected from selected specific areas. Its purpose was to test the effectiveness of a model trained on a benchmark dataset when applied to different locations with the same case.

3.1.1. Benchmarking Dataset

The benchmark dataset was derived from the Sentinel-1 SAR satellite images with C-band acquired from European Space Agency (ESA) Copernicus Open Access hub collected by MKLab. The selected polarization mode of the SAR images in the dataset was vertical-vertical (VV) with a spatial resolution of 10 metres. The SAR images had been processed through some standard pre-processing procedures, including radiometric calibration, speckle filtering, and linear transformation to convert the coefficient backscattering in decibels (dB). Then, the region containing oil spills and other valuable context was cropped from the pre-processed SAR images and rescaled to 1250x650 pixels. All SAR images in the OSD dataset provided by MKLab were formatted in 3-channel RGB images. Additionally, SAR images were accompanied by their corresponding labelled images containing 5 classes of interest, including sea surface (considered background), oil spill, look-alike, ship, and land, as illustrated in Figure 3.1. The labelled images were provided in 1D or single-channel format, which assigns an integer value from 0 to 4 corresponding to defined classes, as shown in Table 3.1. All labelled images have been verified with ground truth information and data by the European Maritime Safety Agency (EMSA) through the CleanSeaNet service. Based on this information, all SAR images and their corresponding labels were located in the European waters that cover 23 EU coastal states (EMSA, 2019). There are 1002 images provided with pixel distribution for each class, as shown in Table 3.1. This highlights the extreme diversion in pixel distribution across classes, leading to class imbalance. This study selected 500 images along with their labels, then splitting them into 400 images (80%) for training and 100 images (20%) for validation to simulate a limited data scenario. Selecting half of the images helps mitigate class imbalance by ensuring a more representative dataset for training the selected models. The image selection process involved visually inspecting images and calculating pixel distributions across classes to ensure an optimal balance across classes.



(a) Synthetic Aperture Radar image (b) Labelled image

Figure 3.1 Sample of a Synthetic Aperture Radar image (a) and corresponding labelled image (b). Black is for the sea surface, cyan represents oil spills, red for look-alikes, brown for ships, and green for land.

Table 3.1 Name of classes, respective labels, and pixel distribution of each class in MKLab dataset

Class	Colour Labels	1D Labels	Pixel distribution
Sea Surface	Black	0	88.38%
Oil Spills	Cyan	1	0.95%
Look-Alikes	Red	2	5.49%
Ship	Brown	3	0.04%
Land	Green	4	5.14%

3.1.2. Indonesian Dataset

The Indonesian dataset was processed from the raw Sentinel-1 SAR images and produce the labelled images. SAR images and their corresponding labels covered two specific locations in Indonesian waters: Karawang and Balikpapan. These locations had witnessed incidents of oil spills due to subsea pipeline ruptures. Karawang waters are located in the central-western region of Indonesia, part of West Java province. An oil spill was first identified on the sea surface on July 16th, 2019, and estimated to have spilled around 500 tonnes (Phady et al., 2019). The second site is in Balikpapan, located in the central region of Indonesia, as part of East Kalimantan province. The oil spill incidents occurred on March 31st, 2018, and resulted from the discharge of approximately 7.000 tonnes of oil (Muin et al., 2022). Figure 3.2 illustrates the study area's location in Indonesian waters, covering two specific locations. It is accompanied by SAR images showing the indicated oil spills in the areas.

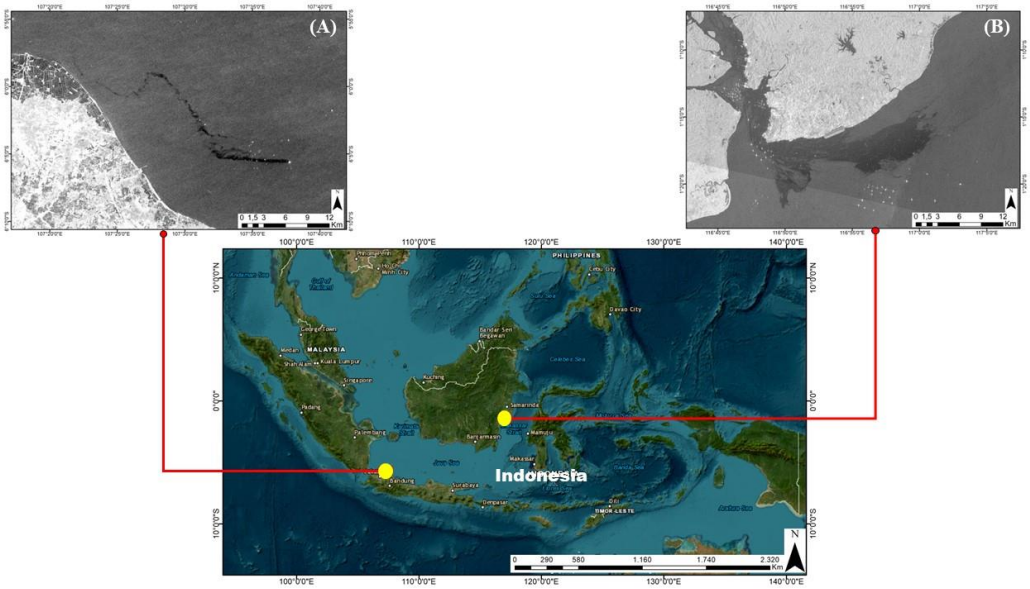


Figure 3.2 Study area map includes Synthetic Aperture Radar images indicating oil spill events across two locations in Indonesia. (A) Oil spills in Karawang; (B) Oil spills in Balikpapan

Sentinel-1 SAR images were collected from the Copernicus Open Access hub belonging to ESA (https://browser.dataspace.copernicus.eu/). To build the SAR image dataset, we only collected Sentinel-1 SAR images with VV polarization in Ground Range Detected (GRD) format. VV polarization mode was selected for its effectiveness as compared to vertical-horizontal (VH) polarization in distinguishing oil slick regions due to its sensitivity to the roughness of sea surface (Chaturvedi et al., 2019). The dataset was then complemented with information regarding geographic coordinates and timestamps of the oil spill events provided by direct field observation, study report (Abimanyu et al., 2021) and the Government Agency of Indonesia (the Ministry of Environment and Forestry; East Kalimantan Provincial Government) as ground truth information. Subsequently, the dark spots in the SAR images indicated as oil spills were confirmed by that ground truth information.

3.1.3. Pre-processing and Preparation of Indonesian Dataset

The collected Sentinel-1 SAR images for generating an Indonesian dataset were then processed in two procedures to produce SAR images and their corresponding labels to test the trained model from the benchmark dataset. In total, 11 scenes of Sentinel-1 SAR images were collected across two selected locations on dates associated with documented oil spill events. Then, all images were pre-processed using standard pre-processing procedures, including radiometric calibration, speckle filtering, and linear transformation to convert the coefficient backscattering in decibels (dB). The pre-process started by subsetting SAR scenes to focus on relevant areas. Only certain regions containing oil spills and other relevant regions to support the context of OSD are carefully chosen through image subsetting. Following this, precise orbit files were applied to the subsetted images to correct any satellite orbital position. The next steps involved removing thermal noise and followed by calibrating digital numbers to sigma to standardize backscatter values across images. This was then complemented by speckle filtering to reduce radar interference, and then the terrain correction was proceeded to adjust for topographical variations. Finally, the backscatter values were converted to decibles (dB). To ensure the consistency of output images with images in the benchmark dataset, which are formatted in 3-channel RGB format, we also save our SAR images output in RGB. This standardization guarantees that all images in this study are uniform and comparable with benchmark dataset.

After SAR images were pre-processed and formatted in backscattering coefficient with dB unit, they were used in the next step to create labelled images with the procedure as shown in Figure 3.4. The flowchart outlines the procedure for creating labelled images from Sentinel-1 SAR images. To begin with, we identified the backscattering coefficients range in the regions containing oil spills. After identifying the range of dB values, the pixel values are then classified into two binary classes by setting a threshold for dark spots. This threshold helps differentiate areas as oil spills from those that are not. It is important to note that this threshold is not a static value, it may vary for dark spots in each different scene due to the environmental conditions and background noise which affect the radar pulse reflectivity of different areas. Following this, the binary-classified raster images were converted into vector polygons for more precise spatial analysis. The binary-classified polygons were then matched with ground-truth information by considering the coordinate point source of oil spill and its presence in the SAR images to select oil spills and eliminate the noise polygon. For land and ship classes are digitized manually. Once all area in the polygon vector were assigned their corresponding labels, the polygon was converted back into a raster format and then resized to 1250x650 pixels. Finally, the outputs of this process were single-channel labelled images, and each label was assigned an integer value corresponding to predefined classes.

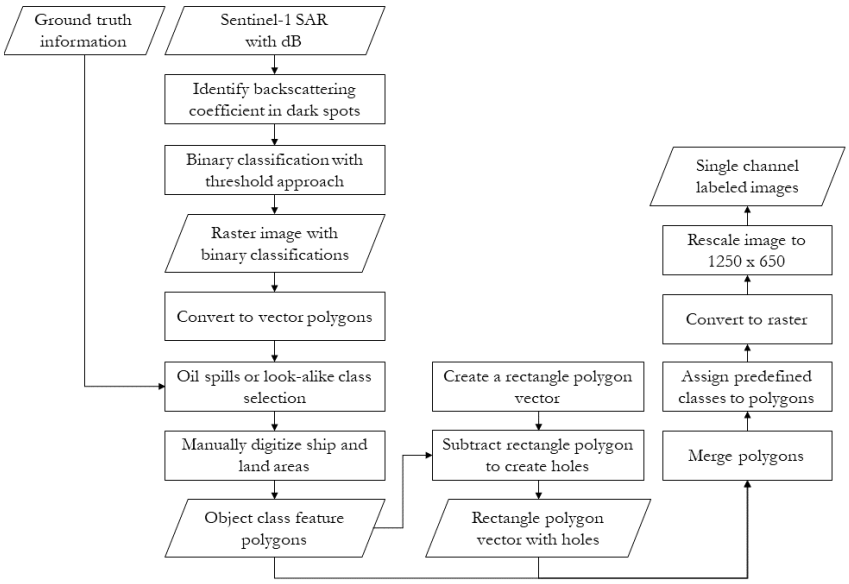


Figure 3.3 Procedures to create labelled images

3.2. Semantic Segmentation for Oil Spill Detection with FCN-DK Model

Some studies on OSD using an FCN model combined with ResNet, GoogleNet, and U-net can accurately identify oil spills down to the pixel (Li et al., 2018; Shamili et al., 2022; Shanmukh et al., 2024). These models have deep networks with many parameters, which enable them to learn intricate features and patterns from the input data. This makes them highly adaptable to new and various problems. However, such networks are prone to overfitting when trained on limited data, such as in cases of OSD.

FCN-DK6 model (Persello & Stein, 2017) with a dilated kernel in each layer is capable of maintaining the same output dimensions as the input. By preserving these dimensions throughout the network, the FCN-DK6 model can maintain high-resolution feature maps to detect the fine details in oil spills. This capability allows the model to effectively distinguish between oil spills and look-alikes. Additionally, the dilated kernel approach eliminates the need for upsampling layers, which reduces the network's complexity and number of parameters. The reduction in network parameters enables the model to handle limited data more effectively during training (Younis & Keedwell, 2019), addressing challenges faced in OSD.

3.2.1. FCN-DK Model Architecture

Persello & Stein (2017) designed FCN-DK6 model with 6 main layers. Each layer is constructed using convolution operations to extract features at different levels of abstraction. The key component of this model is the dilated convolutions, which are embedded in each convolution layer. Dilated convolutions can exponentially expand the receptive fields without adding more learnable parameters per layer and losing the original resolution (Persello & Stein, 2017; F. Yu & Koltun, 2015). With the support of spatial dilated convolution, the kernel is transformed by the equation 3.1.

$$H' \times W' = d(H-1)+1 \times d(W-1)+1 \tag{3.1}$$

Where H and W are the height and width of the original kernel and d is the dilation factor. This approach keeps the number of parameters low, which enhances computational efficiency and reduces the chance of overfitting during training with a limited dataset in OSD. By dilating the kernels, the model can capture broader contextual information, such as the spatial characteristics and patterns of oil spills, such as its distinct boundaries over a wider region of the ocean surface, while also identifying fine details of oil spills patterns within these larger areas. Figure 3.6 illustrates how the receptive field (coloured box) enlarges as the dilation factor (red dots) increases. In Figure 3.6(a), when the dilation factor $d = 1$ (no dilation) is applied, there is no enlargement of the receptive field. However, as dilation factor increases to $d = 2$ and $d = 3$, the receptive field grows exponentially as shown in Figure 3.6 (b) and 3.6(c).

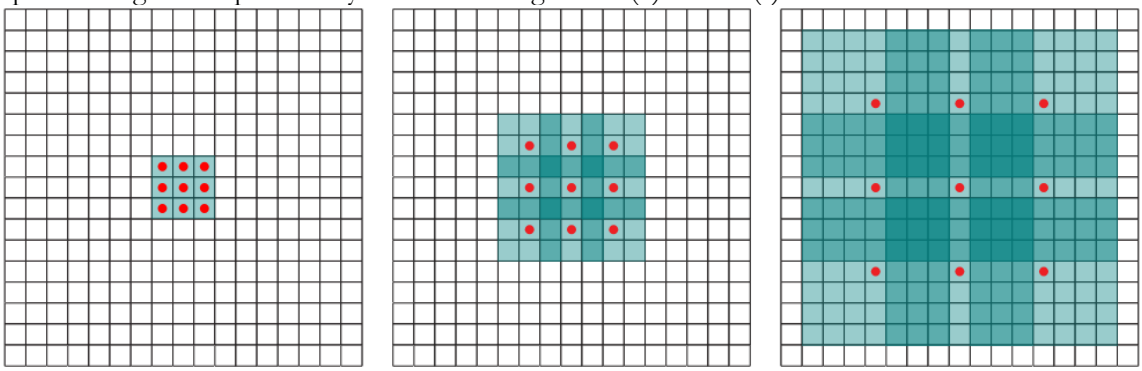


Figure 3.4 Kernel with increasing dilation factors to expand the receptive field (Source: (F. Yu & Koltun, 2015)

FCN-DK6 architecture generally has six convolutional layers with dilated kernels and one final classification layer with a 2D convolution and a softmax loss function, as illustrated in Figure 3.7. Each convolutional layer includes a zero padding 2D layer, a convolution 2D layer with a dilation rate, a batch normalization layer, a leaky ReLu layer, another zero padding 2D layer, and a max pooling 2D layer. Zero padding in this

architecture takes a role in maintaining the spatial dimensions of the input when performing convolution operations. It reduces the risk of deforming patterns in the image (Hashemi, 2019). Batch normalization layers in this network work to normalize the inputs to neural networks, which gives them a zero mean and a constant standard deviation across all layers. This helps to make the gradients more stable, leading to the network train faster, improving accuracy, and allowing for higher learning rates (Bjorck et al., 2018). Leaky ReLu plays a role as an activation function to capture the nonlinearity relationships among the outputs between layers of a neural network. This activation function allows for a small non-zero gradient when the unit is not active, which helps prevent the dying gradient. The dying gradient occurs when the neurons become inactive and stop learning during training (Banerjee et al., 2019). Following this, the max pooling layer works to reduce the spatial dimension of the input feature maps and leads to a decrease in the computational load. The model architecture is then finalized with a softmax loss function for classification tasks. This function calculates the probability that the input image belongs to each class. Then, the cross-entropy loss measures the difference between the ground truth labels and the predicted probabilities (Zhu et al., 2020). Table 3.2 presents the detailed structure and parameters used in the model.

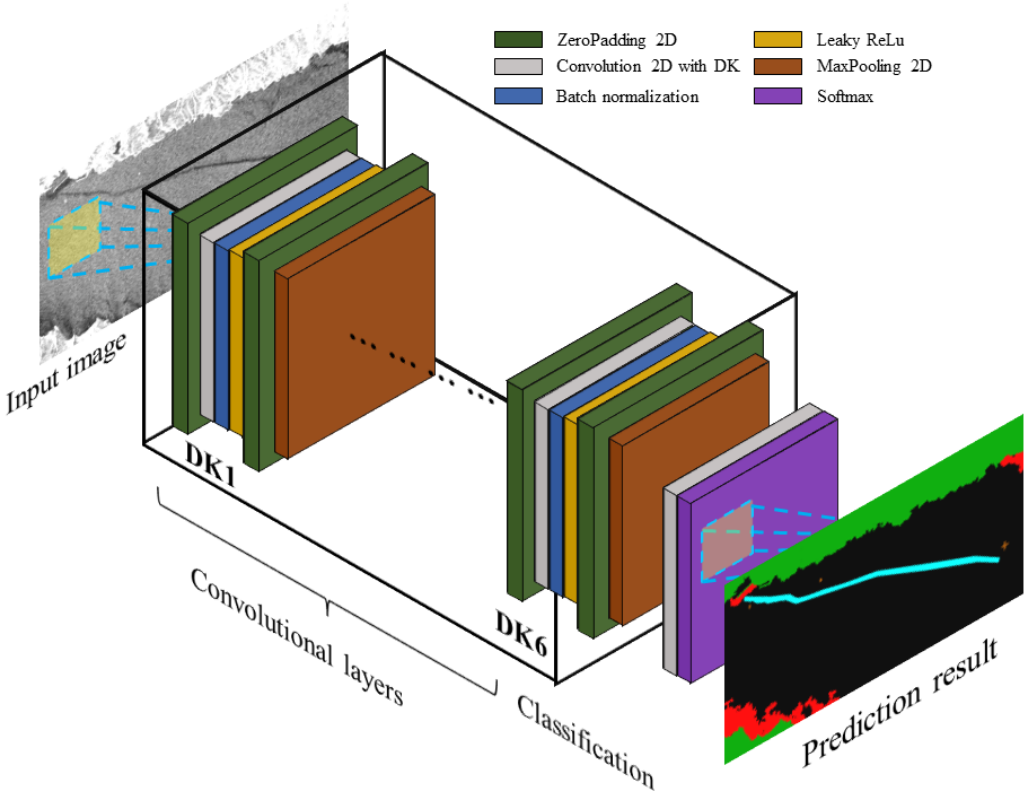


Figure 3.5 Fully Convolutional Network-Dilated Kernel 6 (FCN-DK6) model architecture (source: author)

Table 3.2 The detailed structures and parameters of the FCN-DK6 model

Layer	Modules	Parameters	Dilation
DK1	ZeroPadding 2D	kernel=2 x 2	
	Convolution 2D	kernel=5 x 5, filters=16	1
	BatchNormalizaton	-	
	Leaky ReLu	α=0.1	
	ZeroPadding 2D	kernel=2 x 2	
	MaxPooling2D	kernel=5 x 5	
DK2	ZeroPadding 2D	kernel =4 x 4	
	Convolution 2D	kernel=5 x 5, filters=32	2
	BatchNormalizaton	-	
	Leaky ReLu	α=0.1	
	ZeroPadding 2D	kernel=4 x 4	
	MaxPooling2D	kernel=9 x 9	
DK3	ZeroPadding 2D	kernel=6 x 6	
	Convolution 2D	kernel=5 x 5, filters=32	3
	BatchNormalizaton	-	
	Leaky ReLu	α=0.1	
	ZeroPadding 2D	kernel=6 x 6	
	MaxPooling2D	kernel=13 x 13	
DK4	ZeroPadding 2D	kernel=8 x 8	
	Convolution 2D	kernel=5 x 5, filters=32	4
	BatchNormalizaton	-	
	Leaky ReLu	α=0.1	
	ZeroPadding 2D	kernel=8 x 8	
	MaxPooling2D	kernel=17 x 17	
DK5	ZeroPadding 2D	kernel=10 x 10	
	Convolution 2D	kernel=5 x 5, filters=32	5
	BatchNormalizaton	-	
	Leaky ReLu	α=0.1	
	ZeroPadding 2D	kernel=10 x 10	
	MaxPooling2D	kernel=21 x 21	
DK6	ZeroPadding 2D	kernel=12 x 12	
	Convolution 2D	kernel=5 x 5, filters=32	6
	BatchNormalizaton	-	
	Leaky ReLu	α=0.1	
	ZeroPadding 2D	kernel=12 x 12	
	MaxPooling2D	kernel=25 x 25	
Classification	Convolution2D	kernel=1 x 1, filters=n_classes	
	Activation functions	Softmax	

3.2.2. FCN-DK6 Model Training

Initial FCN-DK6 model training as baseline

The initial training of the FCN-DK6 model involved configuring learning-related hyperparameters, which included the learning rate, batch size, number of epochs, and optimizer type (Yaloveha et al., 2022). The initial learning-related hyperparameters are presented in Table 3.3 referred to Najmi et al. (2022). These hyperparameters were chosen due to their proven effectiveness in similar semantic segmentation tasks. Although Najmi et al. (2022) successfully demonstrated their hyperparameter in the FCN-DK model for slum mapping, OSD presents different challenges. It requires distinguishing oil spills from various marine phenomena in large areas. These specific requirements highlight the need for adaptable hyperparameters for the specific patterns of OSD. The model training was conducted on the benchmark dataset comprising 400 training images and 100 validation images for OSD. The results of training the model with these initial hyperparameter settings were used as a baseline to assess how well the proposed improvement work on the FCN-DK6 model for OSD.

Table 3.3 Initial learning-related hyperparameter in FCN-DK6 model referred to Najmi et al. (2022)

Hyperparameter	Range Tested
Optimizer	SGD (Stochastic Gradient Descent)
Learning rate	$1x10^{-5}$
Momentum	0.9

FCN-DK6 model optimization through hyperparameter tuning

Hyperparameter optimization refers to finding the optimal set of hyperparameters that minimize the loss function on a validation dataset to achieve the best model performance (X. Yu & Lary, 2021). Hyperparameter tuning directly impacts the model's ability to learn from data efficiently and accurately. This study optimized the FCN-DK6 model by tuning two types of hyperparameters: the learning-related hyperparameter and the data/image preprocessing hyperparameter. Optimizing learning-related hyperparameters involved manually adjusting parameters by iteratively testing and fine-tuning the parameters within a specified range presented in Table 3.4. This tuning allows for the identification of hyperparameter settings that optimize the model's accuracy. The mini-batch size determines the number of images processed simultaneously, influencing both training speed and memory usage. Additionally, the optimizer is an algorithm that adjusts model weights to minimize a loss function. Typically, an optimizer works with a learning rate, which can influence how quickly the model learns. The cross-entropy loss function quantifies the difference between the actual labels and the predicted probabilities. Lastly, the number of epochs indicates how many times the model iterates over the training dataset. By carefully tuning these hyperparameters, this study aimed to optimize the performance of the FCN-DK6 model in OSD, ensuring efficient learning and accurate predictions.

Table 3.4 Tuning hyperparameter settings in FCN-DK6 model

Hyperparameters	Range Tested
Mini-batch size	12, 14, 16 images
Optimizer	Adam, SGD
Learning rate	$1x10^{-5}$, $5x10^{-5}$, $1x10^{-4}$, $5x10^{-4}$
Cross entropy loss	categorical_crossentropy
Number of epochs	60, 80, 100

The second approach involves hyperparameter testing related to data/image preprocessing. The hyperparameter set is the patch size, where the images are preprocessed using a patch-based technique by dividing each original image into smaller patches of a specified size. Processing smaller patches in DL instead of the entire image, reduces memory requirements during the training. This approach also allows the model to capture detailed local features more effectively, affecting model performance and accuracy (Hamwood et

al., 2018; Quintana et al., 2023). Additionally, patch size impacts the model's ability to learn and generalize the complex oil spill patterns identification mixed with look-alikes. Smaller patches than the original image dimension help the model focus on finer details in distinguishing oil spills from similar features (Farooq et al., 2018; Martins et al., 2021). Training the network using a patch-based approach enables the model to manage images of any size by dividing them into smaller patches (Volpi & Tuia, 2017). Determining the optimal input patch size is critical for the network to learn complex features of the target objects that enable it to distinguish from the similar objects (Martins et al., 2021). Setting the patch size too small does not capture enough relevant features of oil spills from SAR images for the DL network to accurately identify complex oil spill patterns. Conversely, setting the patch size too large may include mixed features of oil spills along with look-alikes and other similar objects that negatively affect the accuracy of oil spill predictions. Each training image in this study was split into a specified size with non-overlapping patches. The patch sizes examined in this study included 128x128, 256x256, and 512x512 pixels. Subsequently, the FCN-DK6 model was trained using each of the three image patch sizes. This experiment aimed to identify the most effective patch size configuration and understand its influence on achieving optimal model performance.

Optimizing the model through the number of class reduction

As stated earlier in this methodology section, this study implemented multiclass segmentation for five classes: sea surface, oil spills, look-alikes, ships, and land. Additionally, we conducted experiments to reduce the number of classes to four by merging sea surface and land into a single "background" class. Since our focus was on detecting oil spills in the open ocean, the land class is less relevant, where it introduces unnecessary complexity to the model. Accurate monitoring of oil spills relies on detecting the slicks on water surfaces without the interference of land signals. Thus, four class scenarios, including background, oil spills, look-alikes, and ships, can potentially improve the model accuracy by simplifying the segmentation task. Krestenitis et al. (2019a) demonstrated that reducing the number of classes from five to three in an OSD study using Deep CNN improved accuracy. This class reduction allowed the model to focus more effectively on the remaining classes. Thus, we trained the model under both 5-class and 4-class scenarios and then compare the performance of these two class settings.

3.3. Semantic Segmentation for Oil Spill Detection with Hybrid CNN-XGBoost Model

In this section, a hybrid model for semantic segmentation in OSD was proposed. This hybrid approach takes the strength of both DL and ML for specific tasks (Lee et al., 2018). DL is used for extracting high-level features, while ML is used for classification tasks. ML algorithms have been widely applied in pixel-level classification of RS images because of their ability to delineate complex and non-linear patterns accurately (Sheykhmousa et al., 2020). However, the success of classification with ML highly depends on the availability of spatial patterns or features within the image of interest. Extracting and selecting optimal features can be time-consuming and requires considerable effort before classification (Kwak et al., 2021). In OSD, obtaining ground truth data is difficult, resulting in limited training samples. Utilizing DL with limited training samples risks overfitting, while ML needs feature extraction as inputs to achieve optimal accuracy. A hybrid model offers automatic feature extraction by possibly generating many high-level features even with small training images and performs supervised classification to handle non-linear patterns and feature interactions effectively. This approach benefits OSD in semantic segmentation by achieving optimal accuracy with a small amount of training. Figure 3.8 illustrates the image classification process from input to output using different approaches. This study adopts the process illustrated by section C in Figure 3.8.

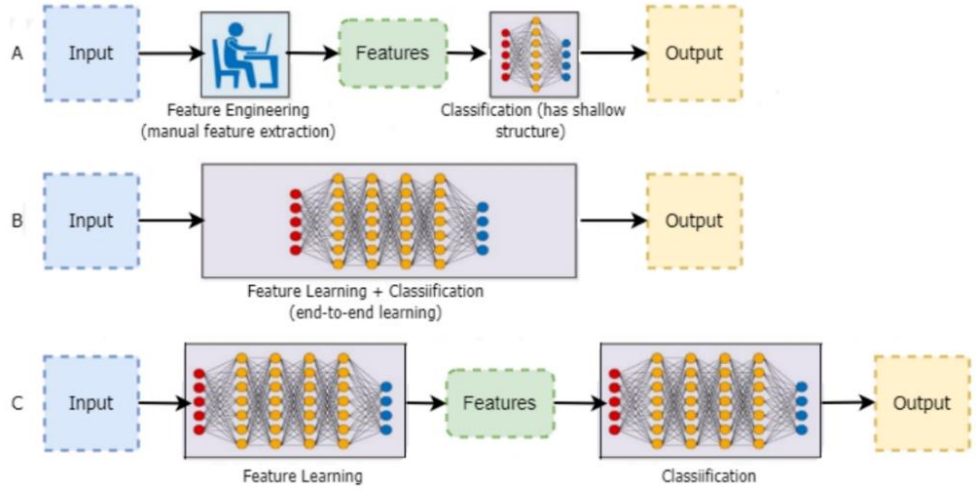


Figure 3.6 The classification process on different techniques. (A) Machine Learning technique; (B) Deep Learning technique; (C) a hybrid technique by combining Deep Learning and Machine Learning (Source: Bakasa & Viriri (2023))

3.3.1. A Hybrid CNN-XGBoost Model Architecture

The model proposed in this study integrates the convolutional networks from FCN-DK6 as a DL element, and the pre-existing XGBoost algorithm as an ML element, as shown in Figure 3.9 to tackle the limited data problem in OSD. The model is divided into two parts: feature extractor and feature classifier. First, CNN functions as a feature extractor that captures essential patterns within input images and then generates feature maps. These two parts are synced by a flattening process that transforms the CNN outputs in feature maps into a format compatible with XGBoost. This structure is designed to optimize model performance by leveraging the effective feature extraction capabilities of CNN and the robust classification of XGBoost. Figure 3.9 shows a simplified architectural overview of the proposed hybrid model.

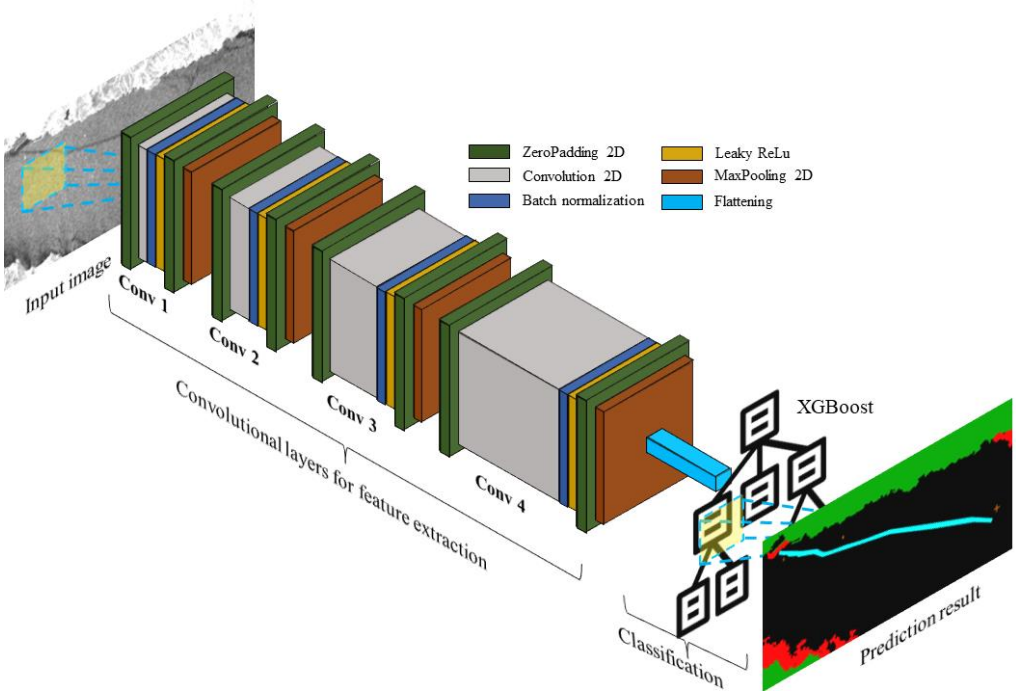


Figure 3.7 Architecture overview of the proposed hybrid model in this research (source: author)

CNN for feature extraction

In this study, only the convolutional networks from the FCN-DK6 are utilized excluding the classification layer. Instead, the role of classification layer is replaced by predefined XGBoost classifier. Convolutional networks are designed to automatically learn spatial hierarchies of features from input images using filters or kernels. Each kernel convolves across the input image to produce a feature map highlighting specific patterns within the image. This is particularly suitable for OSD cases with limited data issues, as CNN can generate multiple synthetic feature maps from one input image to expand the dataset for the training model. Figure 3.10 provides insight into how convolution layers work to produce multiple high-level feature maps from one input image of SAR image.

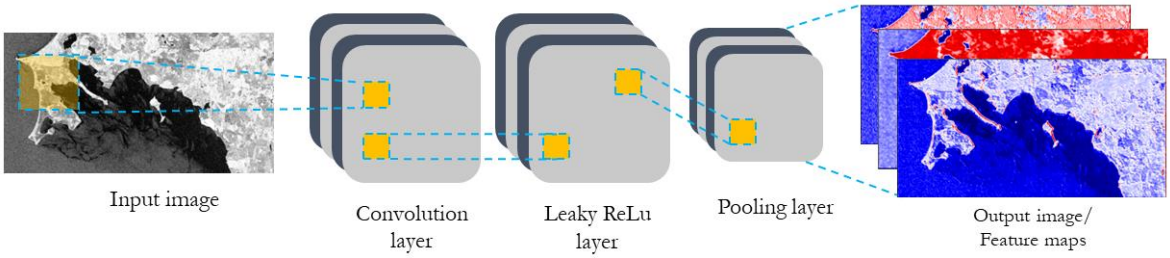


Figure 3.8 Simplified Convolutional Neural Network architecture for feature extraction (source: author)

The CNN architecture of our hybrid model consists of four convolutional layers. Each of these layers includes several sub-layers, which are designed based on the structure of FCN-DK6 as a base model. This study opted for four convolutional layers instead of six to reduce computational complexity and improve training efficiency. Each convolutional layer consists of a zero padding 2D layer, a convolution 2D layer, a batch normalization layer, a leaky ReLu layer, another zero padding 2D layer, and a max pooling 2D layer. However, dilated kernels from the FCN-DK6 were not included in this convolutional layer for the hybrid model to simplify the architecture and maintain a balance between capturing relevant features and computational efficiency. The CNN in this hybrid model utilize a 1x1 kernel window for padding across all layers. Furthermore, both convolutional and pooling operations utilize a 3x3 kernel window throughout the network. This approach is designed to keep the spatial dimension of the output feature maps the same as those of the input. Besides, the adjustment of kernel size is intended to manage the balance between spatial resolution and the receptive field. It also ensures that the model efficiently captures detailed and boundary spatial features. The number of filters in this model is designed to increase as the network goes deeper to generate more feature maps, capturing a wide variety of high-level features for final classification. This hierarchical feature learning helps improve the network's ability to classify images accurately. The structure overview and detailed parameters of the CNN layer for feature extraction can be seen in Figure 3.11 and Table 3.5. Since this study aims to identify the optimal model architecture, the structure of four convolutional layers as outlined in Table 3.5 are maintained, while the number of filters is systematically configured to meet the experimental requirements.

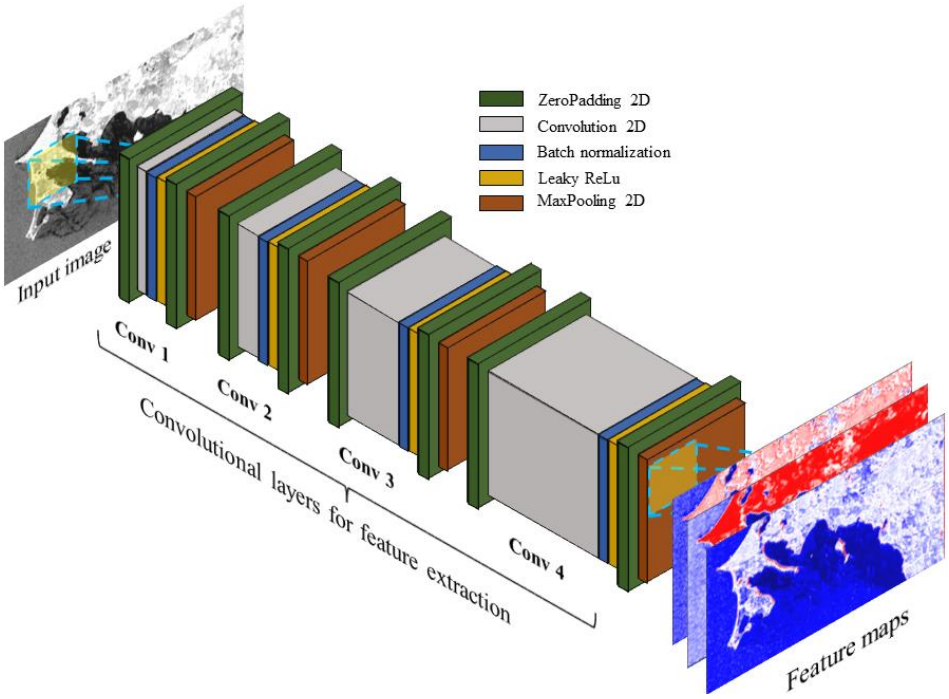


Figure 3.9 Convolutional Neural Network architecture in the hybrid model for feature extraction (source: author)

Table 3.5 The initial structures and parameters of the Convolutional Neural Network for the hybrid model

Layer	Modules	Parameters
Conv 1	ZeroPadding 2D	kernel=1 x 1
	Convolution 2D	kernel=3 x 3, filters=16
	BatchNormalizaton	-
	Leaky ReLu	α=0.1
	ZeroPadding 2D	kernel=1 x 1
	MaxPooling2D	kernel=3 x 3
Conv 2	ZeroPadding 2D	kernel =1 x 1
	Convolution 2D	kernel=3 x 3, filters=32
	BatchNormalizaton	-
	Leaky ReLu	α=0.1
	ZeroPadding 2D	kernel=1 x 1
	MaxPooling2D	kernel=3 x 3
Conv 3	ZeroPadding 2D	kernel=1 x 1
	Convolution 2D	kernel=3 x 3, filters=32
	BatchNormalizaton	-
	Leaky ReLu	α=0.1
	ZeroPadding 2D	kernel=1 x 1
	MaxPooling2D	kernel=3 x 3
Conv 4	ZeroPadding 2D	kernel=1 x 1
	Convolution 2D	kernel=3 x 3, filters=32
	BatchNormalizaton	-
	Leaky ReLu	α=0.1
	ZeroPadding 2D	kernel=1 x 1
	MaxPooling2D	kernel=3 x 3

XGBoost for Classification

The hybrid model employs the pre-existing XGBoost algorithm, which is responsible for classifying high-level features extracted by CNN. XGBoost constructs decision trees using gradient boosting, which is effective for OSD due to its ability to handle non-linear data and complex patterns in oil spill objects. This algorithm involves predicting the residuals or errors from previously trained trees. By sequentially adding these new trees, XGBoost can progressively minimize these errors and elevate the model's accuracy. Gradient descent in this algorithm, on the other hand, functions to reduce the loss caused by adding a new tree (Zhou et al., 2022). In XGBoost, the predicted output for each sample is calculated by summing the weights of the leaves for all weak classifiers (Zhen et al., 2024). The general formula for prediction at step t is as follows in Equation 3.2 (T. Chen & Guestrin, 2016):

$$f_i^{(t)} = \sum_{k=1}^{t} f_k(x_i) = f_i^{(t-1)} + f_t(x_i) \tag{3.2}$$

where $f_t(x_i)$ is the learner of step t, $f_i^{(t)}$ and $f_i^{(t-1)}$ are the prediction results of steps t and $t-1$, and x_i is the input variable. This algorithm has numerous parameters that mainly address the bias-variance trade-off (underfitting and overfitting risks), which involves balancing model accuracy and generalization capability. Two strategies can be used to prevent overfitting and also improve the model's performance in this model, by configuring some parameters. The first strategy focuses on controlling the complexity of the model by adjusting four key parameters: n_estimators, max_depth, min_child_weight, and gamma. The second strategy is to strengthen the model's ability to handle noise by increasing randomness, which helps prevent the model from learning patterns that might not apply to general situations. This involves adjusting parameters such as subsample, colasample_bytree, and eta (learning rate) (Zhou et al., 2022).

3.3.2. Hybrid CNN-XGBoost Model Training

Training the hybrid model ensures that the model effectively captures detailed features from input images and utilizes them for accurate classification. The process of training hybrid models with an oil spill dataset included several steps: image preprocessing, feature extraction through convolutional layers, flattening the feature maps extracted from convolutional layers, and classification using XGBoost classifier. Additionally, optimizing the model by reducing the number of classes, as implemented in the FCN-DK6 model was also applied in this hybrid model. Below is a detailed procedure for training the hybrid model with the oil spill dataset:

Image loading and preprocessing

The preprocessing procedure involved several key steps to prepare the training data for a hybrid model. Initially, SAR and labelled images were loaded and resized from their original dimension of 1250x650 to a uniform size of 500x260 to reduce the computational burden and ensure consistency across all images in the dataset. Simultaneously, the labelled images in a single-channel format are mapped under both 5-class and 4-class scenarios, following the same reduction class experiment as in the FCN-DK6 model. These labels were then converted into one-hot encoding to transform them into binary vectors for the model. Each class index in the labelled images was set to 1 in the corresponding position of this new array, effectively encoding the class information into a format suitable for model training.

Feature extraction through convolutional layers

The feature extraction process involved the four convolutional layers adopted from FCN-DK6 model architecture. The number of filters in the convolutional layers progressively increased as the network deepens, as shown in Table 3.5. The CNN architecture with four convolutional layers was configured to independently generate feature maps with 32, 64, 128, 256, 512, and 1024 filters for the feature maps experiment. This experiment helps to determine how the filter quantity impacts performance and also identifies the optimal number of filters for optimal model performance. To examine the CNN architecture's

ability in producing various feature maps, we initially used only 100 images by selecting randomly to reduce training time. After identifying the optimal feature map configuration, the model was then trained with 400 images. Subsequently, to efficiently handle the dataset and improve training stability, images were processed in mini-batch with 8 batches. This batching technique reduces memory usage and speeds up computation to enhance the model's performance. These mini-batches were fed into the convolutional layers to extract high-level features. The output was feature maps served in four-dimensional arrays in image data, where the dimensions represent the number of images, height, width, and channels. Here, the channels correspond to the number of feature maps produced for each input image. At the end of the feature extraction process, the CNN model independently produces 32, 64, 128, 256, 512, and 1024 feature maps for each input image. These feature maps were then transformed through a flattening procedure to fit the structured data input format required by XGBoost.

Flattening the feature map

To integrate CNN with XGBoost, the output feature maps from the convolutional layers must be transformed into a tabular format compatible with XGBoost. The process begun by flattening the extracted features from a four-dimensional array into a two-dimensional array, where each row represents a feature map. The labelled images were also transformed to align with the two-dimensional SAR image data to ensure each pixel in the feature map corresponds to its label. This transformation prepared the data for subsequent ML tasks by simplifying the structure. Subsequently, the transformed features and labels were combined into a tabular format with Pandas DataFrame. Features were stored in the DataFrame column, while labels were assigned to a 'Label' column. In this study, the number of column in the tabular data corresponds to the input of feature maps from the previous process, with each column representing one feature map. Additionally, one column was included in the last column for the label. This organized dataset was then used to redefine the variable X and Y for training with XGBoost classifier. X consisted of the tabular data with all features excluding the 'Label' column, while Y contained the labels. This preparation ensured that the data was compatible with XGBoost for classification tasks.

Classification using XGBoost

The next step was to classify all data in the tabular data from the previous process using the ML algorithm classifier. This involved XGBoost from Python library package. The classification process begun with setting hyperparameters to control the model's complexity and elevate its predictive accuracy. The model is quite complex with over 20 hyperparameters to fine-tune. Some of the key parameters include n_estimator, learning rate, max_depth, min_child_weight, and gamma. The n_estimators parameter sets the number of trees in the model, where more trees potentially improve the performance but also increase computational cost. The eta or learning rate controls the step size during boosting. A lower learning rate makes the model more robust to overfitting by shrinking the contribution of each tree. The max_depth parameters determine the maximum depth of each tree that can grow, balancing model complexity and preventing overfitting. The min_child_weight parameter determines the minimum total instance weights required for a child node, with higher values making the model more conservative. Lastly, the gamma parameter introduces a regularization term in the loss function, which helps control overfitting. In this study, the hyperparameters were manually tuned and combined within a specified value range to find the optimal model configuration. The selected hyperparameters were then used to train the final hybrid model to ensure it was well-suited for the specific characteristics of the dataset and achieves optimal performance. Table 3.6 displays the hyperparameters tested within a range, including the selected values for final classification.

Table 3.6 Tuning hyperparameter settings in XGBoost for the hybrid model

Hyperparameters	Range Tested
n_estimators	100, 150, 200, 300
max_depth	6, 10, 12
min_child_weight	0.1, 0.5, 1
gamma	0, 0.1, 1
learning_rate	0.1, 0.3, 0.5

3.4. Evaluation Metrics

Accuracy assessment with defined evaluation metrics is necessary for quantifying model sensitivity to hyperparameter settings. To evaluate the performance of FCN-DK6 and the proposed hybrid model, we employed several key metrics: confusion matrix, precision, recall, F1-score, and Intersection over Union (IoU).

Confusion Matrix

The confusion matrix (CM) provides cross-tabulation of classified against reference data (Comber et al., 2012). This matrix typically uses terminology from the binary CM, where the class of interest is called the positive case and the background is called the negative case. The binary CM contains four components: true positives (TP) and true negatives (TN), which are correctly identified samples, and false positives (FP) and false negatives (FN), which are misclassified samples (Maxwell et al., 2021). Table 3.7 shows the structure of CM in binary classification. This assessment tool gives a better understanding of the distribution of classification outcomes and identifies areas where the model may be underperforming.

Table 3.7 Illustration of confusion matrix

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

Precision

Precision evaluates how accurate the positive predictions are, by comparing the count of true positives to the combined count of true positives and false positives. This indicates how well the model identifies positive instances. A high precision means a few false positives. The calculation of precision is presented in equation 3.3.

$$Precision = \frac{TP}{TP + FP} \tag{3.3}$$

Recall

Recall, also called sensitivity or true positive rate is calculated by dividing the number of true positives by the total of true positives and false negatives, as shown in equation 3.4. It indicates how well the model can find all positive instances. High recall means the model has a low rate of missing positive cases.

$$Recall = \frac{TP}{TP + FN} \tag{3.4}$$

F1-Score

The F1-score combines precision and recall into one measure by calculating their harmonic mean. This metric is particularly helpful when class distribution is uneven. A higher F1-score means a better balance between precision and recall. The formula for the F1-score is given in equation 3.5.

$$F1 - score = \frac{TP}{TP + 0.5(FP + FN)} \tag{3.5}$$

Intersection over Union (IoU)

The IoU is a widely used evaluation in semantic segmentation that measures the area of overlap between the predicted output and ground truth divided by both mask combined. In image semantic segmentation, the IoU metric quantifies the pixels common to both predicted and ground truth areas as a proportion of the total pixels for both areas. The model performance is assessed based on IoU accuracy 0 to 1 or served in percentage, which is described in equation 3.6:

$$IoU = \frac{prediction \cap ground\ truth}{prediction \cup ground\ truth} = \frac{TP}{FP + TP + FN} \tag{3.6}$$

3.5. Software and Platform

This study utilizes three main software tools: SNAP, ArcGIS, and Jupyter Notebook. SNAP version 9.0.0 is used for Sentinel-1 SAR image preprocessing, while ArcMap 10.8 is employed to further process Sentinel-1 SAR images to generate Indonesian images. Jupyter Notebook with Python programming environment is used for image preparation, model training, visualization, and performance evaluation. All processes in Jupyter Notebook are run on the Geospatial Computing Platform (GCP) from CRIB, ITC, University of Twente, equipped with an 8Vcpu NVIDIA ARMv8.2, NVIDIA GPU with 32GB memory (Girgin, 2021). All code is executed using the Python libraries TensorFlow, Keras, and scikit-learn.

4. RESULTS

4.1. Pixel Distribution Across Classes

In this study, we prepared images for training, validation, and testing to maintain a balanced distribution of classes. Specifically, 400 images were allocated for training, and 100 for validation from the benchmarking dataset (Krestenitis et al., 2019b). This 80%/20% split ensured sufficient data for training in a limited dataset simulation, while using the validation set to tune hyperparameters based on the model's accuracy assessment. The Indonesian dataset, which consists of only 24 images, was dedicated to testing the model on unseen data from different region. The dataset presented a challenge with class imbalance, as illustrated by the percentage of pixels in Figure 4.1. In this figure, the percentage of the sea surface class outnumbers other classes, which takes over 90% of the total pixel count across all selected images. The other classes, such as look-alikes, land, oil spills, and ships, have smaller regions in SAR images, accounting for only 1% to 2% of the total pixels. The dominance of the sea surface makes it challenging for the model to learn and accurately detect the minority classes, particularly the oil spill class. Consequently, the model may become biased towards the majority class, potentially reducing its sensitivity to detecting oil spills as the main target class. This analysis aims to highlight the challenges of the dataset that may influence the final model performance.

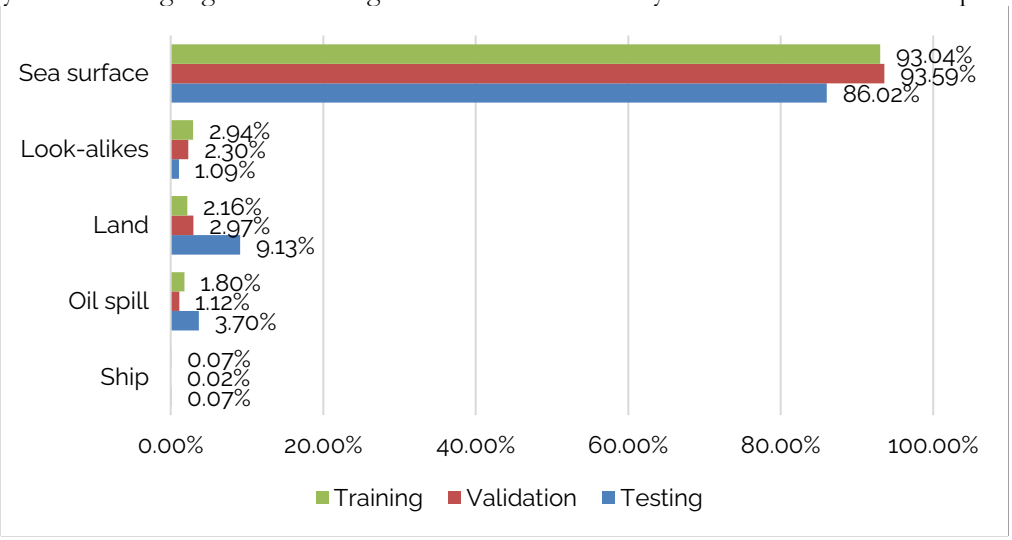


Figure 4.1 Pixels distribution across classes in selected images for training, validation and image prediction purposes

4.2. Oil Spill Detection with FCN-DK6 Model

4.2.1. Initial FCN-DK6 Model Performance

The FCN-DK6 model was trained with a predefined hyperparameter referenced from Najmi et al. (2022) as a baseline to evaluate the proposed model's performance. Table 4.1 shows that the IoU accuracy for FCN-DK6 model is suboptimal, with the main target oil spill class achieving only 23.59% and look-alike class is distinguished by the model at 24.94%. Similarly, Table 4.2 indicates that the model's precision, recall, and F1-score for the oil spill class are 39.29%, 37.11%, and 38.17%, respectively. These metrics suggest that the current hyperparameters are not optimally applied for OSD due to their limited capability in capturing complex patterns of oil spills, highlighting the need for further optimization to improve the model performance.

Table 4.1 IoU metrics accuracy of the FCN-DK6 with initial hyperparameter values

IoU Accuracy (%)					Mean IoU (%)
Sea surface	Oil spill	Look-alike	Ship	Land	
96.44	23.59	24.94	0.00	69.78	42.95

Table 4.2 Evaluation metrics of the FCN-DK6 model with initial hyperparameter values

Metrics	Metric Accuracy (%)					Mean (%)
	Sea surface	Oil spill	Look-alike	Ship	Land	
Precision	97.74	39.29	57.37	0.00	81.77	55.23
Recall	98.64	37.11	30.61	0.00	82.64	49.80
F1-score	98.19	38.17	39.92	0.00	82.20	51.70

4.2.2. FCN-DK6 Hyperparameters and Performance After Tuning

Training a DL model requires hyperparameter tuning, as hyperparameters significantly influence the model's performance. After fine-tuning to obtain the best combination within the tested range, the selected hyperparameter was implemented for further training the FCN-DK6 model. Hyperparameter tuning improves model evaluation metrics, as shown in Table 4.3 and Table 4.4. Table 4.3 compares the model's IoU accuracy between the baseline and after hyperparameter tuning, highlighting an improvement in all classes, particularly for oil spills, increasing from 23.59% to 39.41%. Similarly, Table 4.4 shows evaluation metrics, including precision, recall, and F1-score, that also improve for the oil spill class: precision increased from 39.29% to 53.99%, recall from 37.11% to 59.30%, and F1-score from 38.17% to 56.53%.

Table 4.3 Selected hyperparameter settings in FCN-DK6 model after fine-tuning

Hyperparameters	Range Tested	Selected Value
Mini-batch size	12, 14, 16 images	16 images
Optimizer	Adam and SGD (Stochastic gradient distance)	Adam
Learning rate	$1x10^{-5}$, $5x10^{-5}$, $1x10^{-4}$, $5x10^{-4}$	$5x10^{-5}$
Cross entropy loss	categorical_crossentropy	categorical_crossentropy
Number of epochs	60, 80, 100	100

Table 4.4 Comparison of FCN-DK6 IoU metrics before and after hyperparameter tuning

	IoU Accuracy (%)					Mean IoU (%)
	Sea surface	Oil spill	Look-alike	Ship	Land	
Baseline	96.44	23.59	24.94	0.00	69.78	42.95
After tuning	97.49	39.41	33.93	12.33	80.93	52.82

Table 4.5 Comparison of FCN-DK6 evaluation metrics before and after hyperparameter tuning

	Metrics	Metric Accuracy (%)					Mean (%)
		Sea surface	Oil spill	Look-alike	Ship	Land	
Baseline	Precision	97.74	39.29	57.37	0.00	81.77	55.23
	Recall	98.64	37.11	30.61	0.00	82.64	49.80
	F1-score	98.19	38.17	39.92	0.00	82.20	51.70
After tuning	Precision	98.25	53.99	60.72	39.46	94.97	69.48
	Recall	99.22	59.33	40.12	15.21	84.55	59.69
	F1-score	98.73	56.53	50.66	21.96	89.46	63.47

4.2.3. FCN-DK6 Model Performance in Patch Size and Training Size Experiments

This section presents the results of experiments with various patch sizes and training sizes in the FCN-DK6 model. The baseline FCN-DK6 model was initially trained with a 128x128 patch size, and the same patch size was used for training the model with selected hyperparameter tuning. To investigate the impact of patch size on model performance in detecting oil spills, experiments are conducted with larger patch sizes. Figure 4.2 illustrates the trends of various evaluation metrics for the main target oil spill class across different patch

size settings, all trained with 400 images. It shows that the accuracy of oil spill segmentation increases with larger patch sizes, achieving optimal accuracy with 512x512 patch size setting. As the patch size increases, the model captures more details and relevant features of oil spills, such as texture and edges. Larger patch sizes provide better context and finer details needed to distinguish oil spills from the other classes. Furthermore, this study examined the FCN-DK6 model's capability for handling limited data in OSD cases. Table 4.5 shows that increasing training size does not significantly improve the model's accuracy in OSD. The IoU accuracy for oil spills plateaus after training with 200 images, while the accuracy of the other metrics fluctuates as the training size increases. The increased noise and variability in the oil spill appearance in the SAR images of the larger training dataset could make the model unable to learn consistent patterns. Based on these results, the model achieve optimal performance in 400 images.

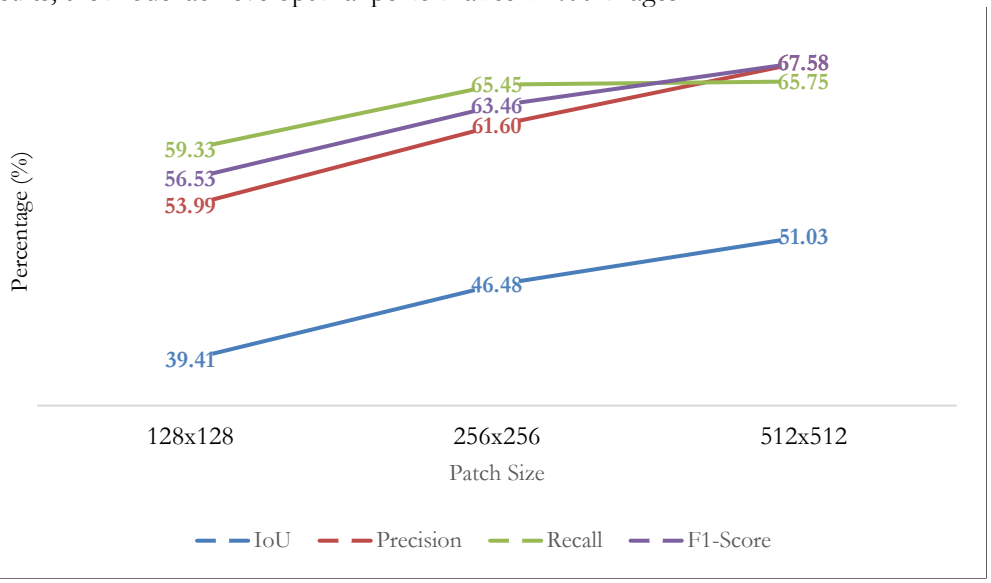


Figure 4.2 Trends in oil spill detection accuracy metrics across various patch sizes in FCN-DK6 model

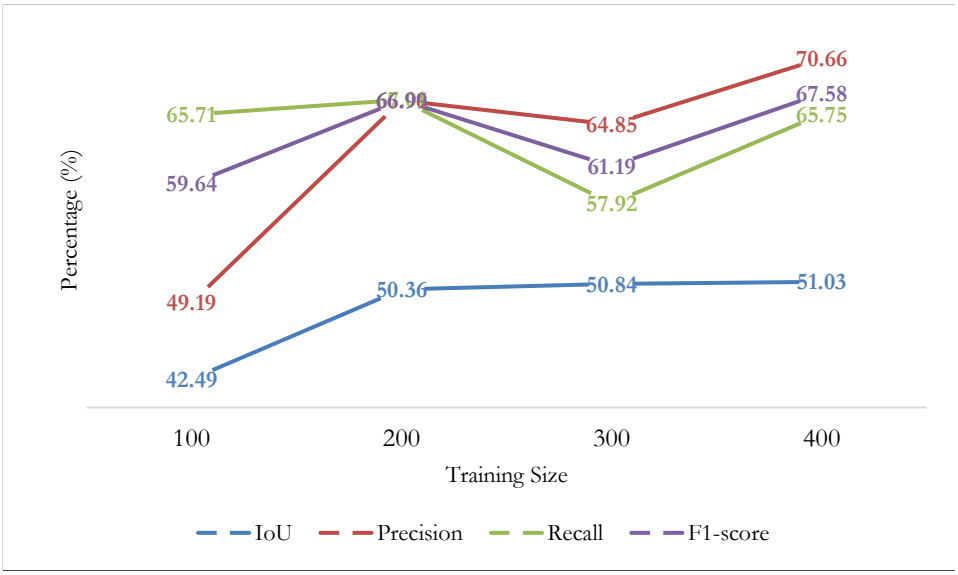


Figure 4.3 Trends in oil spill detection accuracy metrics across various training sizes in FCN-DK6 model

4.2.4. FCN-DK6 Model Performance in a 5-Class and 4-class Scenario

In this section, the model's performance is evaluated under 5-class and 4-class scenarios using quantitative and qualitative analysis. The FCN-DK6 model is trained with a 512x512 patch size, which demonstrated the best performance in the patch size experiments.

Table 4.6 presents the accuracy of IoU metrics of the FCN-DK6 model for both 5-class and 4-class scenarios. In the 5-class scenario, the model achieves a 51.03% IoU score for oil spill segmentation and detecting look-alike at 44.18%. The model faces challenges in accurately detecting due to the complex and variable patterns present in SAR images, such as the diverse texture of the ocean surface and the varying shapes and sizes of oil spills. However, when the number of classes is reduced to a 4-class scenario, the IoU accuracy for oil spills significantly increases to 98.80%. The results for oil spill accuracy show a similar pattern in both evaluation metrics and Iou metrics for both class scenarios. Table 4.7 shows that as the number of classes is reduced from 5 to 4, the accuracy in evaluation metrics for oil spills significantly increases. Precision improves from 70.66% to 99.29%, recall increases from 65.75% to 99.50%, and F1-scores rises from 67.59% to 99.40%. These improvements suggest that simplifying the classification task allows the model to focus better on distinguishing oil spills.

Table 4.6 Comparison of FCN-DK6 IoU metrics for a 5-class and 4-class scenarios

Scenario	IoU Accuracy (%)					Mean IoU (%)
	Sea surface / Background	Oil spill	Look-alike	Ship	Land	
5-class	98.72	51.03	44.18	20.93	91.57	61.29
4-class	97.76	98.80	42.26	13.81		63.16

Table 4.7 Comparison of FCN-DK6 evaluation metrics for 5-class and 4-class scenario

Scenario	Metrics	Metric Accuracy (%)					Mean (%)
		Sea surface / Background	Oil spill	Look-alike	Ship	Land	
5-class	Precision	99.34	70.66	60.76	33.58	95.66	72.00
	Recall	99.38	65.75	61.82	35.73	95.54	71.64
	F1-score	99.36	67.58	61.28	34.62	95.60	71.69
4-class	Precision	98.78	99.29	66.08	20.96	-	71.28
	Recall	98.96	99.50	53.97	28.83	-	70.32
	F1-score	98.87	99.40	59.41	24.27	-	70.49

In the previous section, the quantitative results across multiple evaluation metrics indicate that a patch size 512x512 provides better performance than the smaller patch size settings. This improvement is also reflected in the visual qualitative results, which represent similar findings. Figure 4.4 illustrates a comparison of visual outputs from the FCN-DK6 model trained with different patch size settings. In general, as the patch size increases from the baseline size of 128x128, the model is able to predict better results. The 512x512 patch size setting improves the model's ability to predict visual outputs for semantic segmentation of oil spills, as the model performs better than the smaller patch size. This setting improves the accuracy and clarity of the visual results.

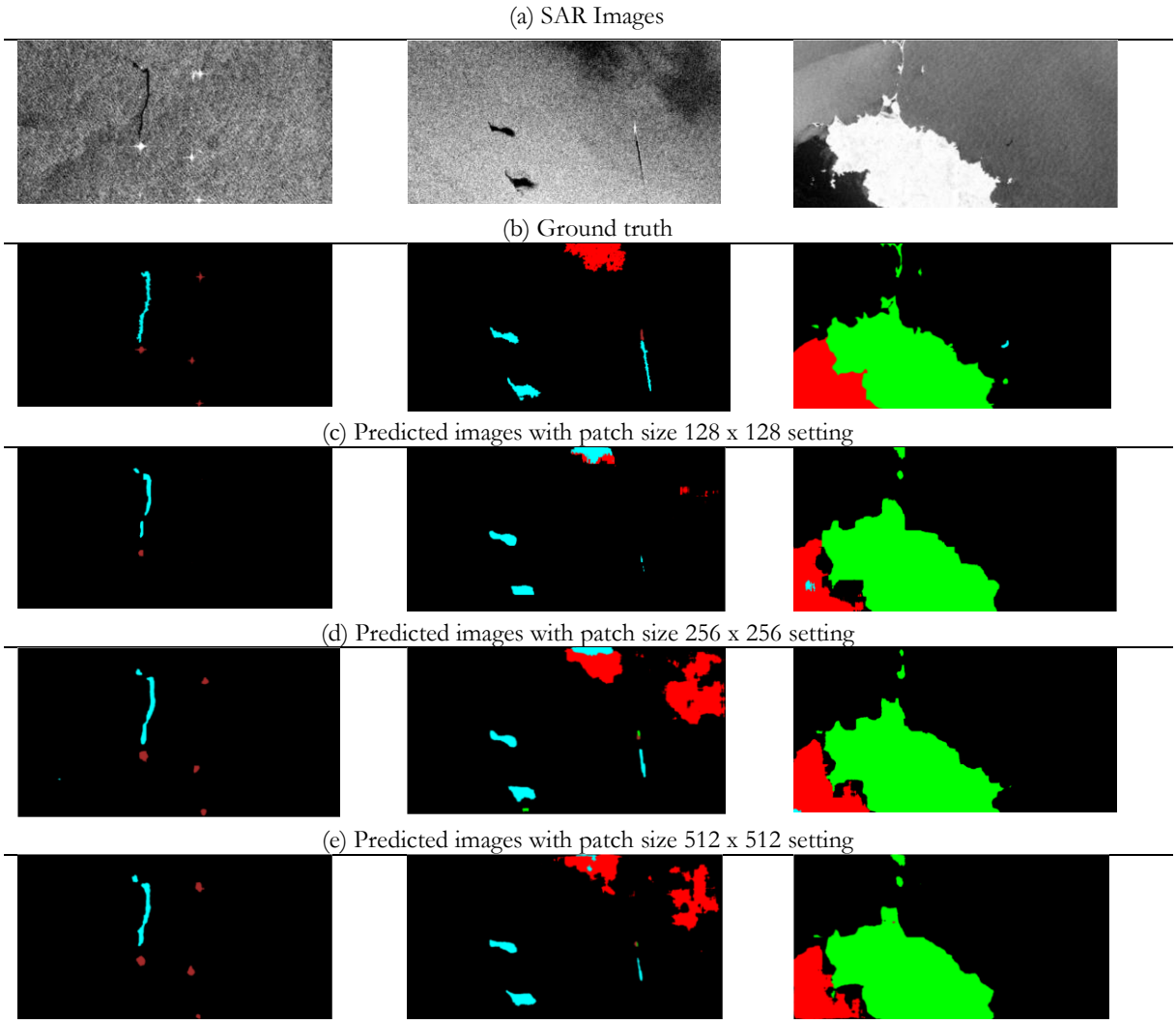


Figure 4.4 Example of qualitative results from the FCN-DK6 model with the 5-class scenario and various patch size settings. Black is for the sea surface, cyan represents oil spills, red for look-alikes, brown for ships, and green for land

To reflect the aforementioned quantitative results, as indicated in Table 4.6, the oil spill IoU accuracy improved significantly from 51.03% to 98.80% when moving from a 5-class setting to a 4-class setting. Figure 4.5 compares the quality of predicted images produced by models trained with different patch sizes and class scenarios. Firstly, as the patch size increases, the segmentation results for oil spills become more accurate and closely align with the ground truth image. As shown in Figure 4.5 (c) and (d), the 128x128 patch size does not adequately capture the full extent of the oil spill with incomplete segmentation for both 5-class and 4-class. Meanwhile, the 256x256 patch size provides more coherent detections than the smaller patch size but still outputs incomplete segmentation, as seen in Figure 4.5 (e) and (f). Figure 4.5 (g) and (h) with 512x512 patch size settings provide the most accurate predictions that capture almost the full extent of the oil spill and closely align with the ground truth. Secondly, when comparing the predicted images from the model with a 5-class and from a 4-class across patch sizes, the predicted images with a 4-class show a better refinement of detecting and delineating oil spills, as shown in Figure 4.5 (c) versus (d) and (e) versus (f). For a detailed comparison between the 5-class and 4-class models with a 512x512 patch size, Figure 4.5 (g) and (h) illustrate the results. The 4-class shows improved detection and delineation of oil spills, capturing almost the full extent of the oil spills and closely corresponding to the ground truth. Even though the model can semantically segment oil spills, it is unable to capture the look-alike class in this image sample due to the subtle appearance of the dark spots, which resemble the surrounding sea surface more closely than look-alikes.

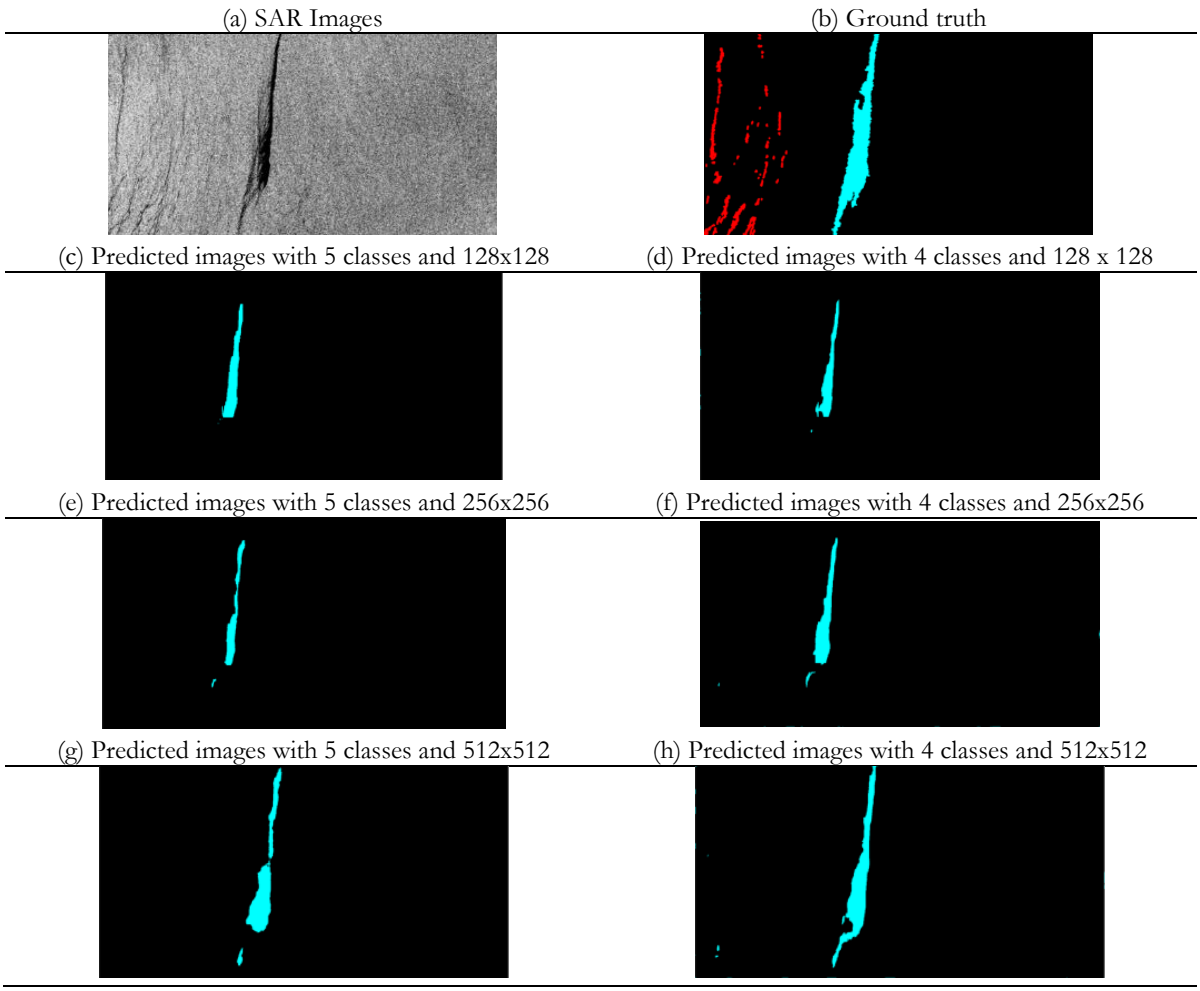


Figure 4.5 Examples of qualitative results from FCN-DK6 model comparing 5-class and 4-class scenarios with various patch size settings. Black is for the sea surface, cyan represents oil spills, red for look-alikes, brown for ships, and green for land

4.2.5. FCN-DK6 Model Performance on Indonesian Dataset

The FCN-DK6 model was trained and validated using the MKLab dataset. The trained model was utilized to predict oil spills from different datasets and regions and in order to investigate its generalization. The Indonesian dataset was used as unseen images to investigate the trained model to predict oil spills in different areas, which might have varying ocean characteristics. Since the model was trained using images that contain oil spill events in European waters. This section presents the model's performance analysis in predicting oil spills in Indonesian water using a trained model on the MKLab dataset.

Table 4.8 presents the IoU accuracy of the FCN-DK6 model for 5-class and 4-class scenarios on Indonesian dataset. We exclusively predicted the Indonesian dataset with the FCN-DK6 model trained with a patch size 512x512. As the previous quantitative and qualitative analysis indicates, this patch size yields the highest model performance compared to the other patch size settings. In the 5-class scenario, the model demonstrates low accuracy in segmenting oil spills at 26.44%, but the model barely segments look-alikes with considerably significantly low accuracy at 0.06%. When the class number is reduced to 4 classes, there is a substantial improvement in the accuracy of the oil spills from 26.44% to 80.52%. Table 4.9 shows that as the number of classes is reduced from 5 to 4, the accuracy in evaluation metrics for detecting oil spills in Indonesian water significantly increases. Precision improves from 52.30% to 91.88, recall increases from 32.74% to 86.69%, and F1-scores rises from 40.27% to 89.21%. The FCN-DK6 model that is trained with 4-

class scenarios significantly improves the detection of oil spills in Indonesian waters. However, the model is unable to segment look-alikes, which could be due to the inadequate pixel representation of look-alikes in the Indonesian dataset.

Table 4.8 Comparison of FCN-DK6 IoU metrics for 5-class and 4-class scenarios on Indonesian dataset

Scenario	IoU Accuracy (%)					Mean IoU (%)
	Sea surface / Background	Oil spill	Look-alike	Ship	Land	
5-class	95.54	26.44	0.06	10.64	47.94	36.12
4-class	94.43	80.52	0.00	7.15	-	45.53

Table 4.9 Comparison of FCN-DK6 evaluation metrics for 5-class and 4-class scenarios on Indonesian dataset

Scenario	Metrics	Metric Accuracy (%)					Mean (%)
		Sea surface/ Background	Oil spill	Look-alike	Ship	Land	
5-class	Precision	94.06	52.30	0.01	14.18	90.05	50.12
	Recall	98.66	32.74	0.00	26.16	52.37	41.99
	F1-score	96.30	40.27	0.00	18.39	66.22	44.24
4-class	Precision	96.16	91.88	0.00	17.72	-	51.44
	Recall	98.13	86.69	0.00	10.71	-	48.88
	F1-score	97.14	89.21	0.00	13.35	-	49.93

Figure 4.6 presents the predicted images of Indonesian dataset, generated using the FCN-DK6 trained model with a patch size 512x512. These predicted images were used to assess the prediction images qualitatively and to align with the quantitative analysis from previous results. As indicated in Table 4.8, there is a significant improvement in IoU scores when moving from a 5-class scenario to a 4-class scenario. Figure 4.6 visually demonstrates the comparison of predicted images under a 5-class and a 4-class scenario. The notable improvement is evident in the predicted images on the left side of Figure 4.6. In the 5-class scenario, the model misclassified oil spills as look-alikes on the predicted images as compared to its corresponding ground truth image. Meanwhile, when the model was trained with a 4-class scenario and predicted the same image, it performed better in detecting oil spills compared to the 5-class scenario, as shown in Figure 4.6. Although a small portion is still misclassified as look-alikes, the model's predictions are substantially more accurate and closely align with the ground truth image. Figure 4.5 in the centre also shows improvements in delineating oil spills in a 4-class scenario. As shown in that figure, the model with a 5-class scenario misclassifies sea surface pixels as the land class and is slightly unable to accurately delineate dark spots as oil spill areas, resulting in a false negative. In contrast, the model with a 4-class scenario shows improved predictions that focus more accurately on oil spills. The oil spill area is better delineated, although there is a small overestimation, resulting in false positives where the model incorrectly predicts oil spills in areas where there are none. Based on the predicted images of Indonesian waters using the FCN-DK6 model, the model trained with a 4-class scenario detects oil spills more effectively than the 5-class scenario. This improvement is also supported by the previous quantitative analysis on the Indonesian dataset.

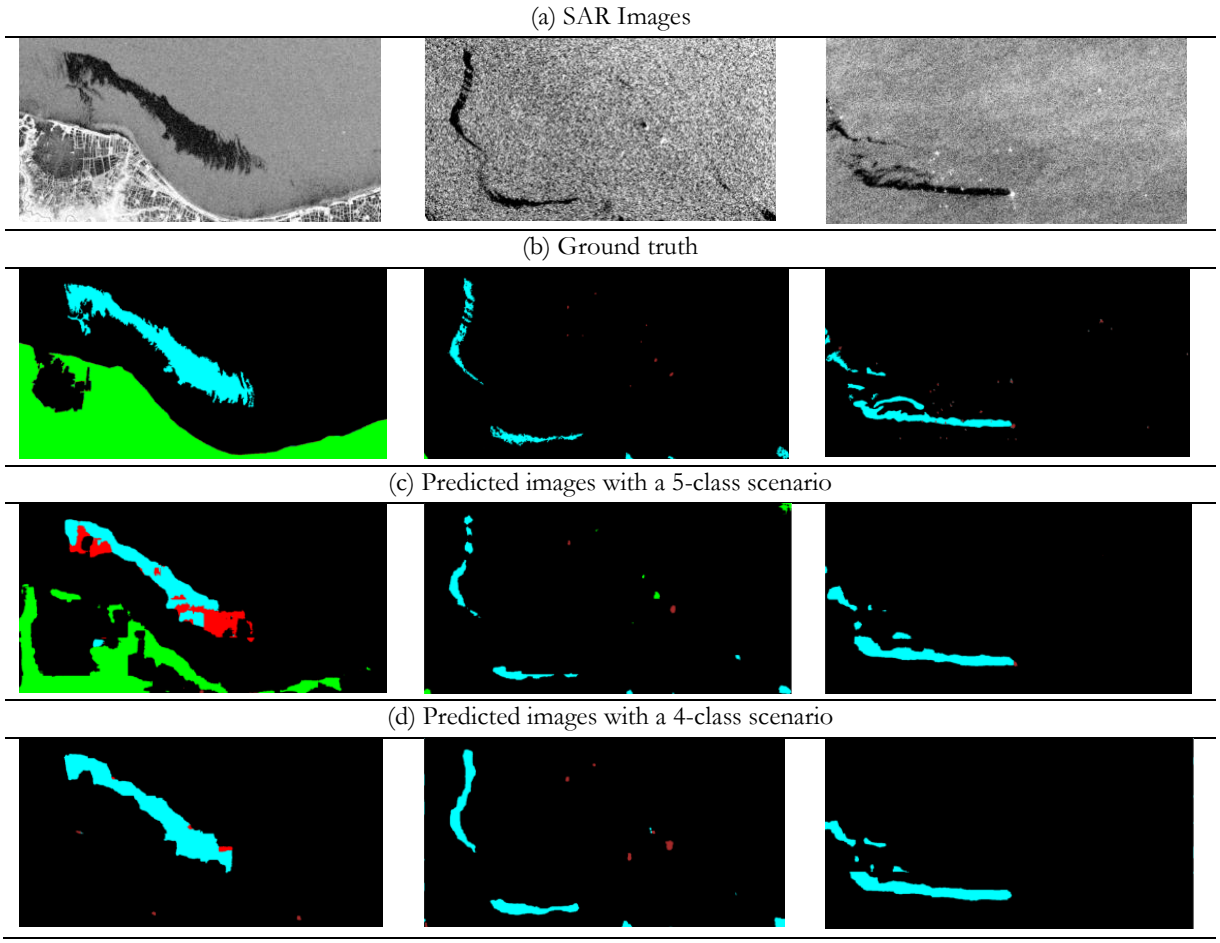


Figure 4.6 Examples of qualitative results from FCN-DK6 model comparing 5-class and 4-class scenarios on Indonesian dataset. Black is for the sea surface, cyan represents oil spills, red for look-alikes, brown for ships, and green for land

4.3. Oil Spill Detection with Hybrid CNN-XGBoost Model

4.3.1. Hybrid Model Hyperparameters and Performance Comparison to Baseline

The hyperparameter tuning in this hybrid model focuses on XGBoost. We fine-tune within a tested range to find the best combination that leads to optimal model's performance. Table 4.10 presents the range of tested hyperparameter values, along with the selected values for each that result in the optimal combination. This optimal combination of hyperparameter values is used to train the model to compare the initial hybrid CNN-XGBoost model performance with the baseline, also for the further training model. The initial hybrid model is trained using 32 feature maps, then followed by further training with an increased number of feature maps. Table 4.11 shows the IoU accuracy of the oil spills, with the initial training of the hybrid model outperforming the baseline FCN-DK6 model. The baseline achieves 23.59%, while the initial hybrid model training achieves 33.94%. Similarly, the hybrid CNN-XGBoost model shows improved accuracy of oil spills in evaluation metrics compared to the baseline as shown in Table 4.12.

Table 4.10 Selected hyperparameter settings in XGBoost after fine-tuning

Hyperparameters	Range Tested	Selected Value
n_estimators	100, 150, 200, 300	200
max_depth	6, 10, 12	12
min_child_weight	0.1, 0.5, 1	1
gamma	0, 0.1, 1	0
learning_rate	0.1, 0.3, 0.5	0.1

Table 4.11 Comparison of IoU metrics for baseline and initial training using hybrid CNN-XGBoost model

	IoU Accuracy (%)					Mean IoU (%)
	Sea surface	Oil spill	Look-alike	Ship	Land	
Baseline	96.44	23.59	24.94	0.00	69.78	42.95
Initial CNN-XGBoost	93.87	33.94	19.01	15.85	71.71	46.88

Table 4.12 Comparison of evaluation metrics for baseline and initial training hybrid CNN-XGBoost model

	Metrics	Metric Accuracy (%)					Mean (%)
		Sea surface	Oil spill	Look-alike	Ship	Land	
Baseline	Precision	97.74	39.29	57.37	0.00	81.77	55.23
	Recall	98.64	37.11	30.61	0.00	82.64	49.80
	F1-score	98.19	38.17	39.92	0.00	82.20	51.70
Initial CNN-XGBoost	Precision	97.16	57.50	24.36	68.20	95.42	68.53
	Recall	96.47	47.33	46.78	20.20	67.46	55.65
	F1-score	96.81	51.92	32.04	31.20	79.04	58.19

4.3.2. Hybrid Model Performance in Feature Map Optimization and Training Size Experiments

This study set the filter configuration in convolutional 2D layers in the CNN from FCN-DK6 model to control feature map generation for feeding XGBoost. Additional features increase the number of columns in the tabular data, providing more data input to XGBoost and potentially capturing more complexity for classification tasks. Therefore, this study examined various numbers of feature maps to determine the optimal set for feeding into XGBoost. Additionally, the impact of training size on model performance was investigated, where each increase in training size corresponds to more rows in the training data.

Figure 4.7 presents trends in oil spill detection accuracy metrics across various feature map settings. As the number of feature maps increases, the accuracy of the IoU and evaluation metrics for the oil spill class gradually improves, and optimal performance is achieved at the highest feature map setting. The IoU accuracy of oil spill improves from 33.94% in the initial training with 32 feature maps to 42.89% at 1024 feature map setting. For the evaluation metrics, precision increases from 57.50% to 64.06%, recall rises from 47.33% to 56.49%, and F1-score improves from 51.92% to 60.04%. Based on these results, the hybrid CNN-XGBoost effectively multiplies the number of training images by augmenting synthetic feature maps from a limited dataset in OSD. This approach optimizes performance in capturing oil spill areas from SAR images, as the model learns the oil spill features from more complex feature inputs, improving its ability to predict oil spills in unseen images. Additionally, Figure 4.8 shows there is no significant improvement in oil spill accuracy in all metrics as the training size increases.

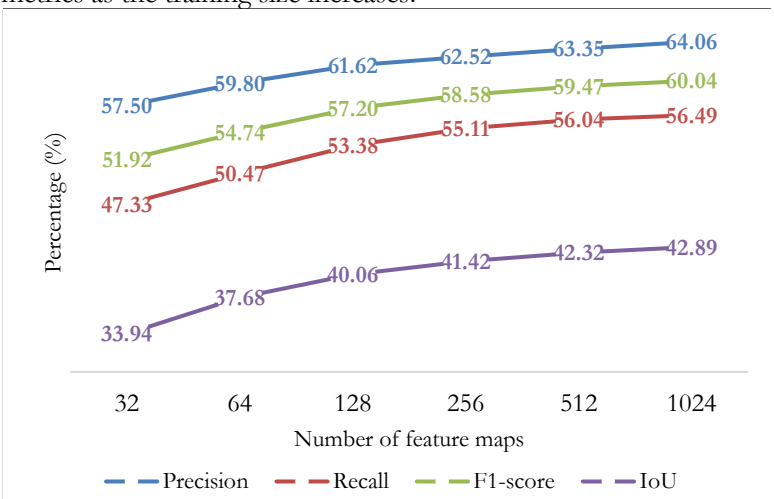


Figure 4.7 Trends in oil spill detection accuracy metrics across various numbers of feature maps setting in hybrid CNN-XGBoost model

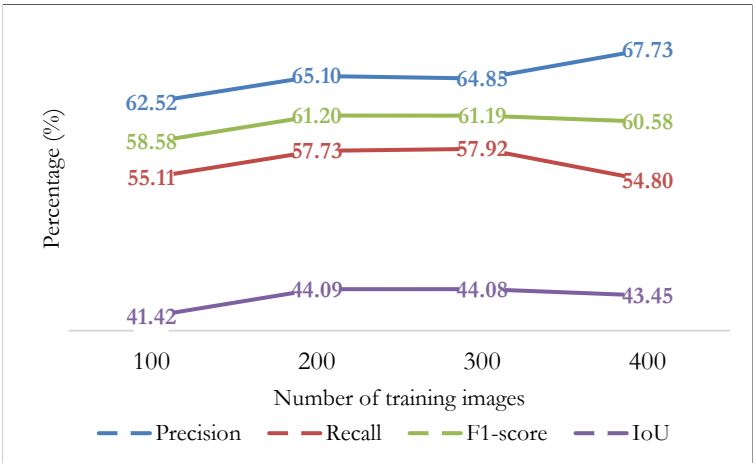


Figure 4.8 Trends in oil spill detection accuracy metrics across various training sizes in hybrid model CNN-XGBoost model

4.3.3. Hybrid Model Performance in a 5-Class and 4-Class Scenario

Similar to the FCN-DK6 model, the hybrid CNN-XGBoost model was trained under both 5-class and 4-class scenarios to investigate the assumption that reducing the number of classes would improve model accuracy. Table 4.13 presents the accuracy of IoU metrics of the hybrid CNN-XGBoost model for both 5-class and 4-class scenarios. In the 5-class scenario, the model achieves a moderate IoU accuracy of 43.45% for segmenting oil spills in SAR images, while the model only achieves 30.74% for segmenting look-alike class. The model encounters challenges in accurately detecting oil spills that might be influenced by the variability of the appearance of SAR images, such as the varying shapes of the oil spills. When the number of classes is reduced to a 4-class scenario, the IoU accuracy for oil spills slightly increases to 51.40%. Table 4.14 shows that as the number of classes is reduced from 5 to 4, the accuracy in evaluation metrics for oil spills slightly increases. Precision improves from 67.73% to 70.16%, recall increases from 54.80% to 58.23%, and F1-scores rises from 60.58% to 63.64%. These improvements suggest that simplifying the classification task allows the model to focus better on distinguishing oil spills.

Table 4.13 Comparison of hybrid CNN-XGBoost IoU metrics for a 5-class and 4-class scenarios

Scenario	IoU Accuracy(%)					Mean IoU (%)
	Sea surface	Oil spill	Look-alike	Ship	Land	
5-class	96.58	43.45	30.74	25.54	78.44	54.95
4-class	97.47	51.40	30.00	21.09	-	49.99

Table 4.14 Comparison of hybrid CNN-XGBoost evaluation metrics for 5-class and 4-class scenario

Scenario	Metrics	Metric Accuracy (%)					Mean (%)
		Sea surface / Background	Oil spill	Look-alike	Ship	Land	
5-class	Precision	97.48	67.73	56.41	75.43	93.99	78.21
	Recall	99.06	54.80	40.31	27.86	77.06	59.82
	F1-score	98.26	60.58	47.02	40.69	84.69	66.25
4-class	Precision	98.96	70.16	53.26	64.02	-	71.60
	Recall	99.13	58.23	41.00	23.93	-	55.57
	F1-score	98.74	63.64	46.33	34.83	-	60.89

Figure 4.9 complements the quantitative analysis in Table 4.13 with qualitative analysis by showing the hybrid model's image prediction results. Figure 4.9 provides a visual comparison of image segmentation produced by the hybrid model under both class scenarios. In the 5-class scenario (c) on the left side, the model shows a false negative by failing to detect the oil spill area correctly. Instead, the model misclassifies

the oil spill as look-alike. In the center part of Figure 4.9, the model detects the oil spill but does not comprehensively capture the entire area. Additionally, the image segmentation on the right side successfully identifies the land and look-alike classes closely aligning with the ground truth image. However, there is a misclassification where a look-alike is incorrectly predicted as an oil spill. As indicated in Table 4.13, the IoU scores of oil spills improve when transitioning from a 5-class setting to a 4-class setting. As seen in Figure 4.9 on the left side, the image segmentation for the 4-class scenario shows improved coverage of the oil spill area. However, it still misclassifies some areas of oil spills as look-alikes, and look-alikes are also incorrectly identified in the background. The improvement in the 4-class scenario is also evident in the center part of Figure 4.9, where the model segments the oil spill area more effectively than in the 5-class scenario. Even though the notable quantitative improvement lies in the oil spill class, the 4-class setting also positively impacts the image segmentation on the right side of the figure. The 4-class setting reduces the misclassification of look-alikes as oil spills. These image predictions reveal that the hybrid model demonstrates capabilities in performing semantic segmentation for OSD, despite some remaining misclassifications that need improvement.

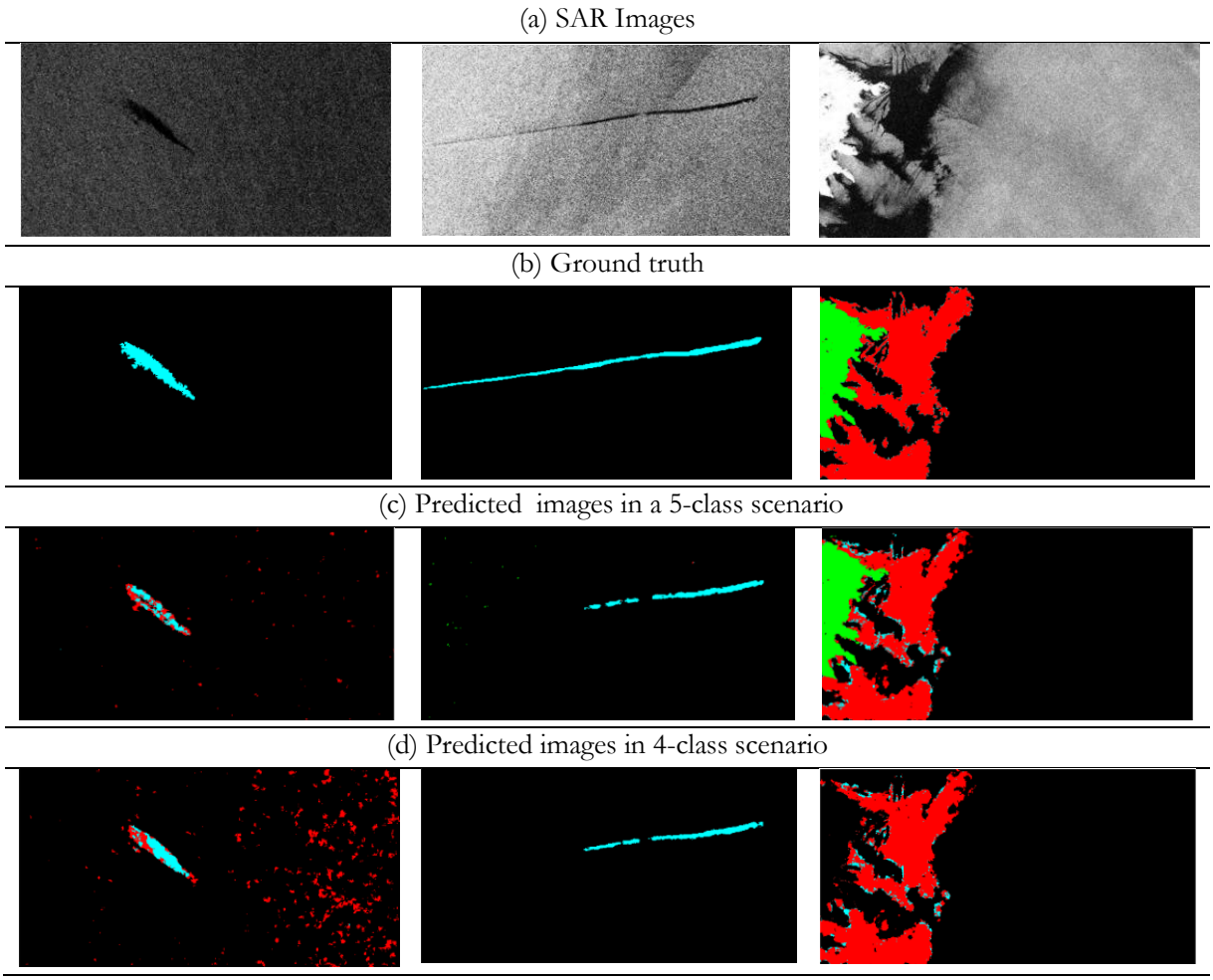


Figure 4.9 Examples of qualitative results from hybrid model comparing 5-class and 4-class scenarios. Black is for the sea surface, cyan represents oil spills, red for look-alikes, brown for ships, and green for land

4.3.4. Hybrid Model Performance on Indonesian Dataset

The IoU metrics in Table 4.15 shows the model is ineffectively identifying oil spill, look-alike, and ship classes while it performs relatively better in segmenting sea surface/background and land. This indicates that the model is ineffectively identifying vital class objects in OSD. When transitioning to the 4-class scenario, the model shows improved performance on oil spill classes from 13.30% to 20.78%. Table 4.16 demonstrates similar improvement in oil spill class across evaluation metrics from 5 to 4 classes. Precision increases from 48.30% to 55.87%, recall improves from 15.51% to 22.72%, F1-score rise from 23.48% to 32.30%. Despite improvements in oil spill accuracy across metrics, the hybrid CNN-XGBoost trained with the benchmark dataset remains ineffective in detecting oil spills in Indonesian waters. This ineffectiveness is due to varying object patterns between the training images and the images from the Indonesian dataset, influenced by different ocean characteristics captured in SAR images.

Table 4.15 Comparison of hybrid CNN-XGBoost IoU metrics for 5-class and 4-class scenarios on Indonesian dataset

Scenario	IoU Accuracy (%)					Mean IoU (%)
	Sea surface / Background	Oil spill	Look-alike	Ship	Land	
5-class	89.65	13.30	0.34	11.37	37.55	30.44
4-class	95.58	20.78	0.31	12.37	-	32.26

Table 4.16 Comparison of hybrid CNN-XGBoost evaluation metrics for 5-class and 4-class scenarios on Indonesian dataset

Scenario	Metrics	Metric Accuracy (%)					Mean (%)
		Sea surface/ Background	Oil spill	Look-alike	Ship	Land	
5-class	Precision	90.57	48.30	0.64	38.45	88.64	53.32
	Recall	98.88	15.51	0.73	13.9	33.96	32.60
	F1-score	94.54	23.48	0.68	20.42	49.11	37.65
4-class	Precision	96.81	55.87	0.56	49.80	-	50.76
	Recall	98.91	22.72	0.73	14.16	-	34.13
	F1-score	97.85	32.30	0.64	22.05	-	38.21

To validate the results of the aforementioned quantitative analysis, Figure 4.10 displays the visual comparison of image predictions generated by the hybrid model under both 4-class and 5-class scenarios on Indonesian dataset. In the left column of Figure 4.10, the model is unable to segment oil spills and land areas, often misclassifying them as background or look-alikes in the 5-class scenario. While there is no significant difference overall in 4-class, but some improvement is observed at the edges of the oil spills where segmentation is accurate. In the center and right columns of Figure 4.10, the hybrid model's image prediction in the 5-class scenario can delineate the major patterns of oil spills. However, there are issues with misclassification as look-alikes are found in areas that should be identified as oil spills. In the 4-class scenario, the segmentation of oil spills shows slight improvement. Areas that were misclassified as look-alikes in the 5-class are correctly identified as oil spills. However, these improvements are minor and only slightly improve the accuracy of the segmentation. These findings suggest that the 4-class scenario offers slight improvements in oil spill segmentation for the hybrid model on the Indonesian dataset. However, through quantitative and qualitative analysis, the hybrid CNN-XGBoost model trained with benchmark dataset demonstrates ineffectiveness in predicting oil spills in Indonesian waters.

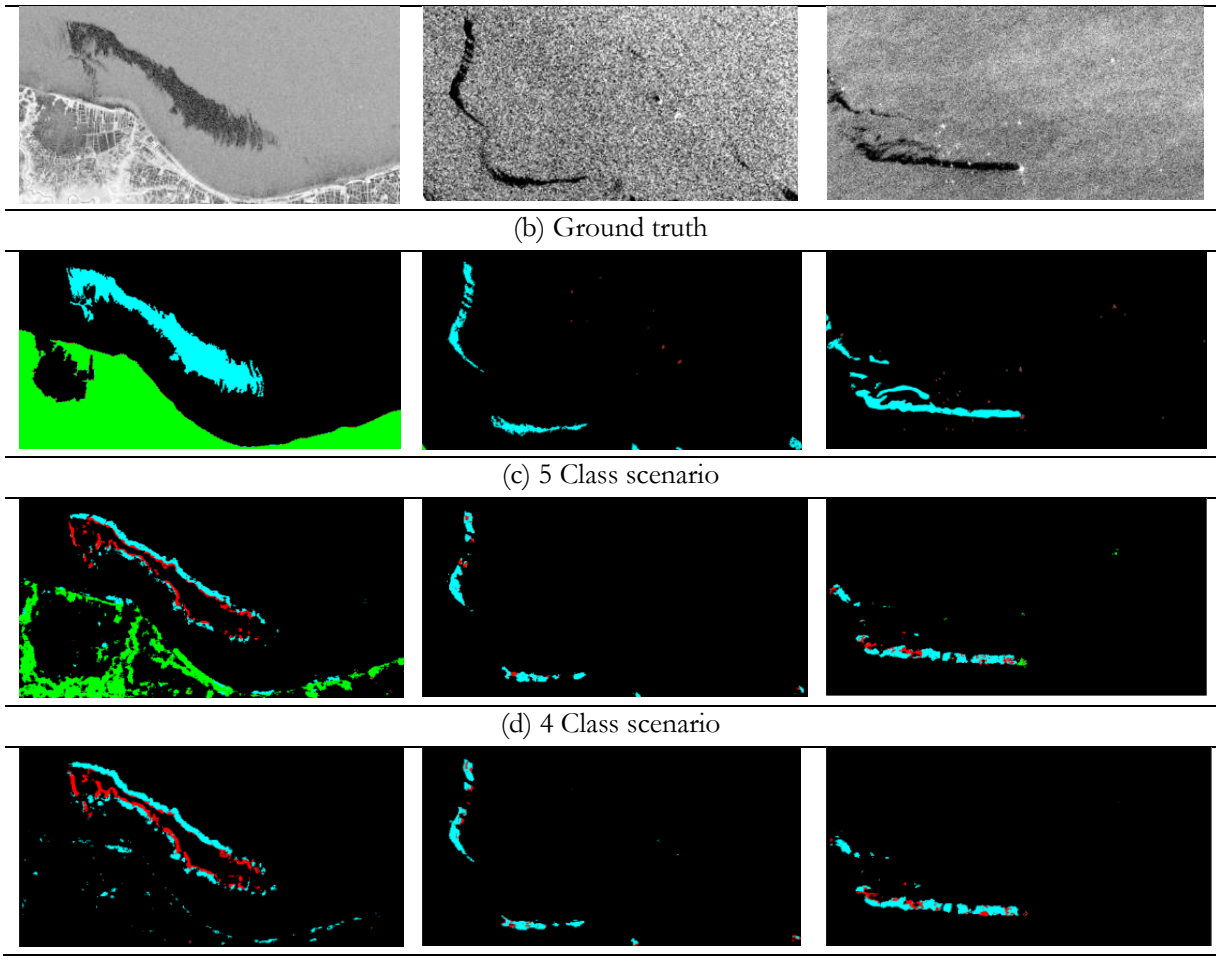


Figure 4.10 Examples of qualitative results from hybrid model comparing 5-class and 4-class scenarios on Indonesian dataset. Black is for the sea surface, cyan represents oil spills, red for look-alikes, brown for ships, and green for land

5. DISCUSSION

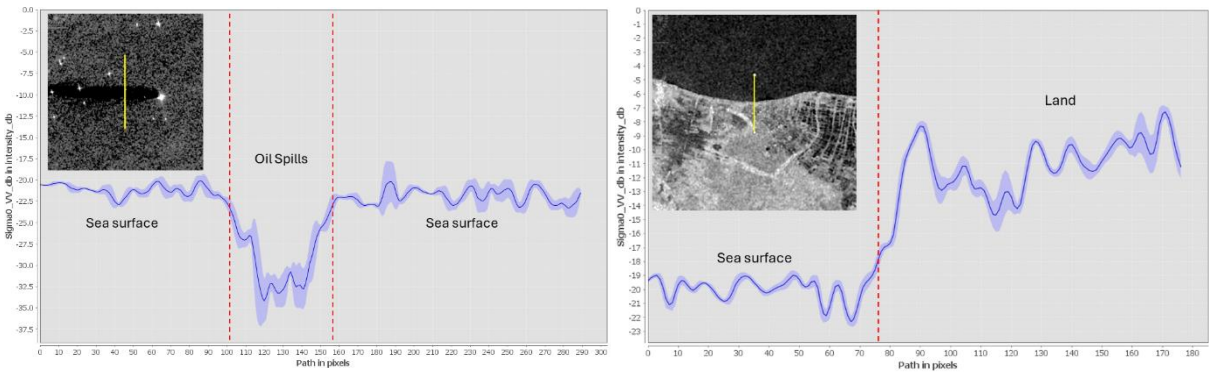
In this study, we optimized the implemented DL-based models by examining their hyperparameters to better learn oil spill features in SAR images. This approach aimed to achieve optimal accuracy in semantic segmentation for OSD despite the limitations of a small dataset. Our findings show that the FCN-DK6 model with fewer network parameters can learn oil spill patterns in SAR images from a limited dataset. This is achieved by optimizing the learning-related and image-preprocessing hyperparameter, specifically fine-tuning the patch size, which can better capture relevant oil spill features. Additionally, a reduced number of class impact significant improvement in segmenting oil spill using FCN-DK6 model. The alternative model development based on FCN-DK6 involves creating a hybrid CNN-XGBoost model. This model achieves optimal oil spill prediction performance with a limited dataset by increasing the number of filters in convolutional layers to generate multiple feature maps from input images for feeding XGBoost and also fine-tuning XGBoost hyperparameters. However, the trained FCN-DK6 and hybrid CNN-XGBoost models are ineffective at predicting oil spills in SAR images from different geographical contexts.

5.1. Oil Spill Detection with FCN-DK6 Model

Our findings indicate that training the FCN-DK6 model with a patch size of 512x512 improves performance compared to smaller patch sizes. The finding of this study in terms of patch size setting aligns with Farooq et al. (2018), who demonstrated 500x500 patch size in weed classification using RS and deep CNN achieved higher accuracy compared to smaller patch sizes of 125x125 and 250x250. Large patch sizes are more likely to capture the variations of oil spills such as shape, size, texture, and appearance within a single patch that enables the model to learn these differences effectively during training. Additionally, smaller patch sizes can split oil spill boundaries across multiple patches, making it harder for the model to capture the entire oil spill. A larger patch size can minimize this issue by capturing oil spills in one patch, providing a more complete view and improving detection accuracy. FCN-DK6 model achieves an IoU accuracy of 51.03% for segmenting oil spills with large patch size setting. In the previous study, Krestenitis et al. (2019b) investigated oil spill identification using several deep neural network models, including DeepLabv2, PSPNet, UNet, and DeepLabv3+. They trained the models with 1002 images from the same dataset and the same 5 classes used in this study, achieving IoU accuracy for oil spills of 25.27%, 40.10%, 53.79%, and 53.38% for the same sequence of the models. All the DL models used in that study are highly complex with extensive parameters to capture oil spill features from SAR images. Nevertheless, the FCN-DK6 model, with its simpler network parameters and training on only 400 images, can learn complex oil spill variability. Its prediction accuracy of oil spills outperforms the more complex DeepLabv2 and PSPNet models as detailed previously. Despite the evaluation accuracy of how the model predicts oil spills, the results indicate that the FCN-DK6 model can handle a limited dataset for OSD. Specifically, it achieves almost the same oil spill segmentation accuracy with 200 training images as it does with 400 training images. This could be due to increased noise such as speckle noise inherent in SAR imaging and variability such as different ocean surface conditions, weather effects, and varying oil spill features within the larger training dataset, which may prevent the model from learning consistent patterns. Nevertheless, the accuracy of oil spill remains moderate that only achieves 51.03%. This could be caused by class imbalance in the dataset, where the FCN-DK6 model ineffectively to learn less frequent oil spill features due to the dominance of sea surface class. This imbalance can lead the model to prioritize sea surface features over oil spill that reduce OSD accuracy.

The highest OSD accuracy in the FCN-DK6 model was achieved when the model was trained with a reduced number of classes from 5 to 4. In the 5-class scenario, the IoU accuracy for an oil spill is 51.03%, where in the 4-class scenario, IoU accuracy significantly improves to 98.80%. This quantitative improvement is reflected in the prediction image in Figure 4.5 and Figure 4.6. These results could be that merging simplifies the classification task and reduces the complexity of distinguishing between multiple classes that

may have overlapping features. By merging sea surface and land into one background class, the model can focus on learning the dark spots and characteristics of oil spills without being confused by the similar features of this class. These dark spot characteristics could include shapes and contextual patterns that are highly different from the background, making oil spills more effectively detected. Figure 5.1 illustrates the backscattering profiles of oil spills, sea surface and land in SAR images. It shows that the backscattering coefficient of oil spills is considerably lower than sea surface and land, highlighting the distinct characteristics of oil spills in images compared to the sea surface and land. This suggests that merging the sea surface and land into a single background class simplifies the FCN-DK6 model's task. The model can treat sea surface and land as one class and focus on learning distinct characteristics of oil spills such as boundary shapes, improving the accuracy of OSD. Even though this study uses RGB channels for the SAR image format to train the model, this backscattering analysis provides similar insights into how the model perceives dark areas in SAR images.



(a) (b)

Figure 5.1 Backscattering profile plots of Sentinel-1 SAR images: (a) Sea surface and oil spills; (b) Sea surface and land. Yellow line represents the transect line (source: author)

5.2. Oil Spill Detection with Hybrid CNN-XGBoost

Based on this study's experiment, configuring the filters in convolutional layers can improve the accuracy of oil spills in the hybrid CNN-XGBoost model. Increasing the number of filters in the CNN generates more high-level feature inputs, providing the XGBoost algorithm with diverse features. This can expand the dataset used to train the XGBoost algorithm. This enriched dataset allows the XGBoost to better distinguish between oil spills and other classes by capturing variations and patterns that may not be evident with fewer features. Additionally, more diverse feature inputs from a single image help the model more effectively learn the characteristics of oil spills such as their shape, texture and size. With the configured filter setting in CNN and selected optimal hyperparameters in XGBoost trained with 400 images, this proposed hybrid model achieves IoU accuracy for oil spills in 5 classes of 43.45%. The accuracy outperforms the baseline of FCN-DK6 model, which achieves 23.59% with its initial hyperparameter. The previous study conducted by Diana et al. (2021) utilized CNN to identify oil spills with the same dataset and the same number of class, reaching an IoU of 25.8% for segmenting the oil spill class. By substituting the fully connected layer in CNN for classification tasks, the hybrid CNN-XGBoost model outperforms the standard CNN model in OSD. This suggests that the hybrid model achieves comparable results using XGBoost for classification tasks. Kwak et al. (2021) stated that incorporating high-level features in ensemble learning achieves better accuracy in classification tasks compared to using only spectral information or digital values of the images. We also examine the capability of the hybrid CNN-XGBoost model in handling limited dataset by varying training size. The result in Figure 4.8 demonstrates there is no significant improvement in oil spill accuracy in all metrics as the training size increases. This suggests that the model may have reached its capacity to learn oil spill features from the initial training size. Furthermore, the additional training data could be introducing more variability from other classes, which might influence the model's ability to improve its accuracy specifically for oil spills.

Despite the potential of the hybrid CNN-XGBoost model in OSD, three inefficacies have been identified: noise in feature maps, suboptimal feature selection for XGBoost, and training inefficiencies. First, SAR images naturally contain 'salt-and-pepper' noise characterized by randomly distributed black-and-white pixels, and this noise can be carried into the feature maps generated by the CNN. This noise in the feature maps can potentially degrade the quality of the input provided to the XGBoost. Additional preprocessing steps to reduce noise could help mitigate this issue. Second, as the CNN element generates a large number of features, some of which may be redundant or irrelevant, where it can reduce the effectiveness of XGBoost in segmenting oil spills. Implementing feature selection to identify only the most relevant features can improve the performance of XGBoost in detecting oil spills. Lastly, the hybrid CNN-XGBoost model requires significant computational resources for training as filters in convolutional layer are set to generate hundreds or thousands of feature maps from to expand the dataset.

5.3. Oil Spill Detection with FCN-DK6 and Hybrid CNN-XGBoost Models on Indonesian Dataset

The FCN-DK6 and hybrid CNN-XGBoost were trained using MKLab dataset, which comprises images collected from European waters. Based on the evaluation metrics, the IoU accuracy for segmenting oil spills indicate suboptimal performance for both models. When the FCN-DK6 model predicts Indonesian SAR images containing oil spills using 4-class scenario, it achieves an IoU accuracy of 80.52% for segmenting oil spills. This is higher than 26.44% IoU accuracy that achieves in 5-class scenario in Indonesian waters. This improvement demonstrates that reducing the number of classes from 5 to 4 positively impacts the model's ability to predict unseen images from the Indonesian dataset. However, the hybrid CNN-XGBoost model shows only a small improvement in the accuracy of oil spill segmentation. Despite an improvement in accuracy, the IoU accuracy of oil spills on the Indonesian dataset predicted by the FCN-DK6 and hybrid CNN-XGBoost models trained with MKLab remains ineffective, as shown in Table 4.7 and Table 4.9. To address this issue, future work could explore the use of transfer learning to adapt the trained model to new geographical context.

The overall quantitative and qualitative analysis indicates that the FCN-DK6 and hybrid CNN-XGBoost model does not perform well in segmenting oil spills in Indonesian waters. This shortcoming may be due to the difference in ocean characteristics between European and Indonesian waters. One of the oceanographic characteristics that differ between European waters and Indonesian waters that influence the presence of oil spills in SAR images is sea surface temperature (SST). Figure 5.2 shows the mean seasonal SST in several European sea regions versus the Indonesian waters obtained from NOAA data. European seas experience seasonal variations in SST due to their temperate climate, with ranges from 2^{o}C in winter to 20^{o}C in summer depending on the regions, as shown in Figure 5.2. Colder temperatures could increase oil viscosity, causing it to spread more slowly and maintain a thicker layer of oil on the surface. This state can create an oil spill more contrast in SAR images. In contrast, Indonesian waters are characterized by consistently high SST throughout the year, ranging from 28–30^{o}C (shown in Figure 5.2) due to their tropical climate. This warm state decreases oil viscosity, causing it to spread more thinly and evenly, which can reduce the contrast between the oil spill and the surrounding water in SAR images. These conditions suggest that a model trained on SAR images from regions with significantly different oceanographic characteristics may be unable to accurately predict oil spills in different locations with different conditions. It highlights the importance of considering local variability in oceanographic characteristics when training models for OSD, as regional differences can impact model performance.

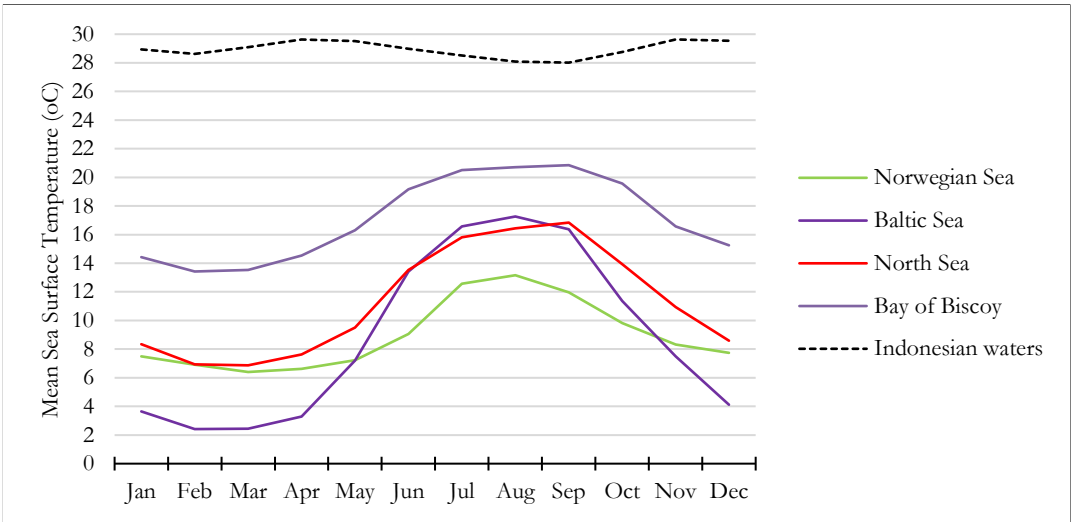


Figure 5.2 Monthly mean sea surface temperatures in European and Indonesian waters processed by the author from daily Optimum Interpolation Sea Surface Temperature (OISST) data from NOAA period Jan-Dec 2023

5.4. Limitations

While this study provides valuable insights into the optimizing DL-based model for semantic segmentation in OSD, three main limitations must be acknowledged.

1. The limitation of this study in segmenting oil spills using the optimized FCN-DK6 and the hybrid CNN-XGBoost model is that the accuracy of OSD still needs improvement. This could be due to the class imbalance in the dataset, which results in the model being trained on fewer oil spill samples compared to other classes. This class imbalance can lead to biased learning, where the model becomes better at identifying the more dominant classes and ineffective to accurately detect the less frequent oil spills. Consequently, this prevents the model from achieving high accuracy.

2. The models in this study were trained using RGB channels instead of the original of SAR images that contain the backscattering coefficient typically used in SAR data. This approach may have limited the model's ability to fully exploit the unique properties of SAR data, such as specific texture and scattering characteristics that are critical for accurately identifying and segmenting oil spills. By not utilizing the backscattering coefficient, the model may not capture the full extent of information available in SAR images, which can potentially reduce its effectiveness in OSD.

3. The dataset does not include geographical coordinates for the SAR images. This means the trained model cannot identify the exact locations of oil spill evens. Without this spatial information, the model can predict oil spills but cannot provide the necessary details for precise monitoring and response.

4. Hyperparameter tuning in both models was performed manually and stepwise without using structured automatic tuning methods. This manual approach may have missed optimal configurations that structured methods like GridSearchCV, RandomSearcCV, or HyperOpt could identify more effectively. These automated tuning methods systematically explore a wide range of hyperparameter combinations to search for the best possible model performance. The study could limit the model's performance by not implementing these advanced techniques.

6. CONCLUSION

6.1. Research Conclusion

The problem of implementing DL in OSD efforts lies in larger dataset availability since obtaining a high-quantity and quality dataset of oil spills is challenging. Factors such as the rarity of oil spill events, the need for extensive labelling, and the variability in the marine environment make it difficult to compile a large and representative dataset. This study proposes optimizing DL-based models for semantic segmentation in OSD by adjusting their hyperparameters to achieve optimal performance with a limited dataset.

This study discovered that the FCN-DK6 model with fewer network parameter is capable of performing semantic segmentation in OSD on a limited dataset by optimizing the learned-related and image preprocessing hyperparameter, specifically in patch size configurations. We find that a large patch size setting provides optimal accuracy in segmenting oil spills using a limited dataset. Large patch sizes can capture the variations of oil spills such as shape, size, texture, and appearance in SAR images within a single patch, allowing the model to learn these differences effectively during training. Additionally, the experiment by reducing the number of classes during training significantly improves the accuracy of detecting oil spills. This approach allows the model to focus on identifying the dark spots and characteristics of oil spills without being confused by the similar features of this class.

This study developed a hybrid CNN-XGBoost model based on convolutional networks from the FCN-DK6 model and a pre-existing XGBoost algorithm as an alternative solution in optimizing the FCN-DK6 model. This model achieves optimal oil spill prediction performance with a limited dataset by increasing the filters in convolutional layers to generate multiple feature maps from input images and by fine-tuning XGBoost hyperparameters. This enriched dataset improves XGBoost's ability to distinguish oil spills from other classes by capturing subtle variations and patterns. Furthermore, the diverse features help the model effectively learn the distinctive shape, texture, and size of oil spills in SAR images.

Although optimizing the FCN-DK6 mode and developing the hybrid CNN-XGBoost model demonstrate positive findings, this study still has some limitations. The trained FCN-DK6 and hybrid CNN-XGBoost models are ineffective in accurately predicting oil spills in SAR images from different geographical regions. Additionally, the accuracy of OSD remains moderate and requires further improvement to achieve reliable performance.

6.2. Recommendations for future works

Based on the content of this study, there are three recommendations for future research:

1. Training dataset development
   The OSD dataset provided by MKLab (Krestenitis et al., 2019b) is the only publicly available dataset suitable for training DL model for semantic segmentation purposes. However, this dataset only offers SAR images in RGB channels, lacking the backscattering coefficient typically included in SAR images. Further standard SAR images with backscattering coefficients are needed to meet the criteria of SAR images, providing specific texture and scattering characteristics that are potentially effective for semantic segmentation in OSD. Providing the real SAR dataset for OSD is expected to provide a trained model that can be used in real operations for monitoring oil spill events using SAR images. The development of this SAR dataset should include geographic coordinates to enable the trained model to pinpoint the exact location of oil spill events for quick response in the fields.
2. Transfer learning
   To strengthen the effectiveness of OSD using deep learning, we recommend the use of transfer learning. Transfer learning allows the pre-trained model to be fine-tuned with specific SAR image

dataset relevant to oil spills. This approach not only improves model accuracy but also enables the DL network to learn and adapt to local characteristics and unique environments in specific geographical areas. This can provide more effective model to detect oil spill events in different local contexts.

7. ETHICAL CONSIDERATIONS

This study utilized a dataset from MKLab group that is part of Center Research and Technology Hellas, Greece. This dataset is licensed for non-commercial use and is exclusively available for academic research. Official permission from the provider was secured to use the dataset for this MSc research as for academic purposes. The study intentionally utilized the SAR images and their labelled images from this dataset, as they meet the necessary criteria for developing a DL-based model in OSD. All research activities that are related to this dataset strictly followed ethical standards and complied with provider guidelines.

LIST OF REFERENCES

Abimanyu, A., Pranowo, W. S., Faizal, I., Afandi, N. K. A., & Purba, N. P. (2021). Reconstruction of Oil Spill Trajectory in the Java Sea, Indonesia Using SAR imagery. *Geography, Environment, Sustainability*, *14*(1), 177–184. https://doi.org/10.24057/2071-9388-2020-21

Albelwi, S., & Mahmood, A. (2017). A Framework for Designing the Architectures of Deep Convolutional Neural Networks. *Entropy*, *19*(6). https://doi.org/10.3390/e19060242

Alpers, W., Holt, B., & Zeng, K. (2017). Oil spill detection by imaging radars: Challenges and pitfalls. *Remote Sensing of Environment*, *201*, 133–147. https://doi.org/10.1016/j.rse.2017.09.002

Arslan, N., Majidi Nezhad, M., Heydari, A., Astiaso Garcia, D., & Sylaios, G. (2023). A Principal Component Analysis Methodology of Oil Spill Detection and Monitoring Using Satellite Remote Sensing Sensors. *Remote Sensing*, *15*(5). https://doi.org/10.3390/rs15051460

Atonye, J. J. (2023). Acute Health Effects of Crude Oil Spill in a Rural Community in Bonny, Rivers State, Nigeria. *Central Asian Journal of Medical and Natural Science*, *4*(2). https://doi.org/https://doi.org/10.17605/cajmns.v4i2.1406

Bakasa, W., & Viriri, S. (2023). VGG16 Feature Extractor with Extreme Gradient Boost Classifier for Pancreas Cancer Prediction. *Journal of Imaging*, *9*(7). https://doi.org/10.3390/jimaging9070138

Banerjee, C., Mukherjee, T., & Pasiliao, E. (2019). An Empirical Study on Generalizations of the ReLu Activation Function. *2019 ACM Southeast Conference*. https://doi.org/https://doi.org/10.1145/3299815.3314450

Basha, S. H. S., Dubey, S. R., Pulabaigari, V., & Mukherjee, S. (2020). Impact of Fully Connected Layers on Performance of Convolutional Neural Networks for Image classification. *Neurocomputing*, *378*, 112–119. https://doi.org/10.1016/j.neucom.2019.10.008

Baydaroğlu, Ö., & Demir, I. (2024). Temporal and Spatial Satellite Data Augmentation for Deep Learning-Based Rainfall Nowcasting. *Journal of Hydroinformatics*, *26*(3), 589–607. https://doi.org/10.2166/hydro.2024.235

Benamrou, B., Ouardouz, M., Allaouzi, I., & Ahmed, M. Ben. (2020). A Proposed Model to Forecast Hourly Global Solar Irradiation Based on Satellite Derived Data, Deep Learning and Machine Learning Approaches. *Journal of Ecological Engineering*, *21*(4), 26–38. https://doi.org/10.12911/22998993/119795

Bi, L., Feng, D., & Kim, J. (2018). Dual-Path Adversarial Learning for Fully Convolutional Network (FCN)-Based Medical Image Segmentation. *Visual Computer*, *34*(6–8), 1043–1052. https://doi.org/10.1007/s00371-018-1519-5

Bi, L., Kim, J., Kumar, A., Fulham, M., & Feng, D. (2017). Stacked Fully Convolutional Networks with Multi-channel Learning: Application to Medical Image Segmentation. *Visual Computer*, *33*(6–8), 1061–1071. https://doi.org/10.1007/s00371-017-1379-4

Bjorck, J., Gomes, C., Selman, B., & Weinberger, K. Q. (2018). Understanding Batch Normalization. *NeurIPS Proceedings*. https://proceedings.neurips.cc/paper_files/paper/2018/file/36072923bfc3cf47745d704feb489480-Paper.pdf

Chaturvedi, S. K., Banerjee, S., & Lele, S. (2019). An assessment of oil spill detection using Sentinel 1 SAR-C images. *Journal of Ocean Engineering and Science*, *5*(2), 116–135. https://doi.org/https://doi.org/10.1016/j.joes.2019.09.004

Chawla, N. V, Lazarevic, A., Hall, L. O., & Bowyer, K. W. (2003). SMOTEBoost: Improving Prediction of the Minority Class in Boosting. *7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, 107–119.

Chehresa, S., Amirkhani, A., Rezairad, G. A., & Mosavi, M. R. (2016). Optimum Features Selection for oil Spill Detection in SAR Image. *Journal of the Indian Society of Remote Sensing*, *44*(5), 775–787. https://doi.org/10.1007/s12524-016-0553-x

Chen, J., Zhang, W., Wan, Z., Li, S., Huang, T., & Fei, Y. (2019). Oil spills from global tankers: Status review and future governance. In *Journal of Cleaner Production* (Vol. 227, pp. 20–32). Elsevier Ltd. https://doi.org/10.1016/j.jclepro.2019.04.020

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, *13-17-August-2016*, 785–794. https://doi.org/10.1145/2939672.2939785

Comber, A., Fisher, P., Brunsdon, C., & Khmag, A. (2012). Spatial Analysis of Remote Sensing Image Classification Accuracy. *Remote Sensing of Environment*, *127*, 237–246. https://doi.org/10.1016/j.rse.2012.09.005

Conceição, M. R. A., Mendonça, L. F. F., Lentini, C. A. D., Lima, A. T. C., Lopes, J. M., Vasconcelos, R. N., Gouveia, M. B., & Porsani, M. J. (2021). SAR Oil Spill Detection System through Random Forest Classifiers. *Remote Sensing*, *13*(11). https://doi.org/10.3390/rs13112044

Das, K., Janardhan, P., & Narayana, H. (2023). Application of CNN based image classification technique for oil spill detection. *Indian Journal of Geo-Marine Sciences*, *52*(1), 5–14. https://doi.org/10.56042/ijms.v52i01.5438

de Moura, N. V. A., de Carvalho, O. L. F., Gomes, R. A. T., Guimarães, R. F., & de Carvalho Júnior, O. A. (2022). Deep-Water Oil-Spill Monitoring and Recurrence Analysis in the Brazilian Territory using Sentinel-1 Time Series and Deep Learning. *International Journal of Applied Earth Observation and Geoinformation*, *107*. https://doi.org/10.1016/j.jag.2022.102695

Diana, L., Xu, J., & Fanucci, L. (2021). Oil Spill Identification from SAR Images for Low Power Embedded Systems using CNN. *Remote Sensing*, *13*(18). https://doi.org/10.3390/rs13183606

Duke, N. C., Pinzón M., Z. S., & Prada T, M. C. (1997). Large-scale Damage to Mangrove Forests Following Two Large Oil Spills in Panama. *Biotropica*, *29*(1), 2–14. https://doi.org/10.1111/j.1744-7429.1997.tb00001.x

Dutta, S., Joseph, M., Kumari, E. V. S. S., & Prasad, A. V. V. (2018). Automated Approach for Extraction of Oil Spill from SAR Imagery. *Journal of the Indian Society of Remote Sensing*, *46*(4), 633–639. https://doi.org/10.1007/s12524-017-0728-0

El-Zaart, A., & Ghosn, A. A. (2013, April 27). SAR Images Thresholding for Oil Spill Detection. *Saudi International Electronics, Communications and Photonics Conference*. https://doi.org/10.1109/SIECPC.2013.6550755

EMSA. (2019). Cleanseanet Service: Detecting Marine Pollution from Space. In *European Maritime Safety Agency*. https://www.emsa.europa.eu/csn-menu/items.html?cid=122&id=3771

Fan, J., & Liu, C. (2023). Multitask GANs for Oil Spill Classification and Semantic Segmentation Based on SAR Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 2532–2546. https://doi.org/10.1109/JSTARS.2023.3249680

Fan, J., Zhang, S., Wang, X., & Xing, J. (2023). Multifeature Semantic Complementation Network for Marine Oil Spill Localization and Segmentation Based on SAR Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *16*, 3771–3783. https://doi.org/10.1109/JSTARS.2023.3264007

Farooq, A., Hu, J., & Jia, X. (2018). Weed Classification in Hyperspectral Remote Sensing Images Via Deep Convolutional Neural Network. *2018 IEEE International Geoscience and Remote Sensing Symposium*. https://doi.org/10.1109/IGARSS.2018.8518541

Feng, S., Zhou, H., & Dong, H. (2019). Using Deep Neural Network with Small Dataset to Predict Material Defects. *Materials and Design*, *162*, 300–310. https://doi.org/10.1016/j.matdes.2018.11.060

Fingas, M., & Brown, C. E. (2011). Oil Spill Remote Sensing: A Review. In *Oil Spill Science and Technology* (pp. 111–169). Elsevier Inc. https://doi.org/10.1016/B978-1-85617-943-0.10006-1

Gade, M., Alpers, W., Hühnerfuss, H., Masuko, H., & Kobayashi, T. (1998). Imaging of Biogenic and Anthropogenic Ocean Surface Films by the Multifrequency/Multipolarization SIR-C/X-SAR. *Journal of Geophysical Research: Oceans*, *103*(C9), 18851–18866. https://doi.org/10.1029/97jc01915

Galieriková, A., & Materna, M. (2020). World Seaborne Trade with Oil: One of Main Cause for Oil Spills? *Transportation Research Procedia*, *44*, 297–304. https://doi.org/10.1016/j.trpro.2020.02.039

Gallego, A. J., Gil, P., Pertusa, A., & Fisher, R. B. (2019). Semantic Segmentation of SLAR Imagery with Convolutional LSTM Selectional Autoencoders. *Remote Sensing*, *11*(12). https://doi.org/10.3390/rs11121402

Garcia-Garcia, A., Orts-Escolano, S., Oprea, S., Villena-Martinez, V., & Garcia-Rodriguez, J. (2017). A Review on Deep Learning Techniques Applied to Semantic Segmentation. *ArXiv*. https://doi.org/https://doi.org/10.48550/arXiv.1704.06857

Girard-Ardhuin, F., Mercier, G., Collard, F., & Garello, R. (2005). Operational Oil-Slick Characterization by SAR Imagery and Synergistic Data. *IEEE Journal of Oceanic Engineering*, *30*(3), 487–495. https://doi.org/10.1109/JOE.2005.857526

Girgin, S. (2021). *Using FOSS to Develop and Operate A Geospatial Computing Platform*. Zenodo. https://doi.org/https://doi.org/10.5281/zenodo.6025282

Guo, H., Wei, G., & An, J. (2018). Dark Spot Detection in SAR images of Oil Spill using Segnet. *Applied Sciences (Switzerland)*, *8*(12). https://doi.org/10.3390/app8122670

Hamwood, J., Alonso-Caneiro, D., Read, S. A., Vincent, S. J., & Collins, M. J. (2018). Effect of Patch Size and Network Architecture on a Convolutional Neural Network Approach for Automatic Segmentation of OCT Retinal Layers. *Biomedical Optics Express*, *9*(7), 3049. https://doi.org/10.1364/boe.9.003049

Hao, S., Zhou, Y., & Guo, Y. (2020). A Brief Survey on Semantic Segmentation with Deep Learning. *Neurocomputing*, *406*, 302–321. https://doi.org/10.1016/j.neucom.2019.11.118

Hashemi, M. (2019). Enlarging Smaller Images Before Inputting Into Convolutional Neural Network: Zero-Padding vs Interpolation. *Journal of Big Data*, *6*(1). https://doi.org/10.1186/s40537-019-0263-7

Jiang, S., Zhao, H., Wu, W., & Tan, Q. (2018). A Novel Framework for Remote Sensing Image Scene Classification. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *42*(3), 657–663. https://doi.org/10.5194/isprs-archives-XLII-3-657-2018

Jiao, Z., Jia, G., & Cai, Y. (2019). A New Approach to Oil Spill Detection that Combines Deep Learning with Unmanned Aerial Vehicles. *Computers & Industrial Engineering*, *135*, 1300–1311. https://doi.org/https://doi.org/10.1016/j.cie.2018.11.008

Kementerian Lingkungan Hidup dan Kehutanan Republik Indonesia. (2018). *Laporan Tim Penanganan Kejadian Tumpahan Minyak (Oil Spill) Di Perairan Teluk Balikpapan Kota Balikpapan Dan Kabupaten Penajam Pasir Utara, Provinsi Kalimantan Timur*.

Khalid, S., Khalil, T., & Nasreen, S. (2014). A Survey of Feature Selection and Feature Extraction Techniques in Machine Learning. *Proceedings of 2014 Science and Information Conference, SAI 2014*, 372–378. https://doi.org/10.1109/SAI.2014.6918213

Kolokoussis, P., & Karathanassi, V. (2018). Oil Spill Detection and Mapping Using Sentinel 2 Imagery. *Journal of Marine Science and Engineering*, *6*(1). https://doi.org/10.3390/jmse6010004

Koroniotis, N., Moustafa, N., & Slay, J. (2022). A new Intelligent Satellite Deep Learning Network Forensic Framework for Smart Satellite Networks. *Computers and Electrical Engineering*, *99*. https://doi.org/10.1016/j.compeleceng.2022.107745

Kostianoy, A., Litovchenko, K., Lavrova, O., Mityagina, M., Bocharova, T., Lebedev, S., Stanichny, S., Soloviev, D., Sirota, A., Pichuzhkina, O., & Shirshov, P. P. (2006). Operational Satellite Monitoring of Oil Spill Pollution in the Southeastern Baltic Sea: 18 Months Experience. *Environmental Research, Engineering and Management*, *38*(4), 70–77.

Krestenitis, M., Orfanidis, G., Ioannidis, K., Avgerinakis, K., Vrochidis, S., & Kompatsiaris, I. (2019a). Early Identification of Oil Spills in Satellite Images Using Deep CNNs. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *11295 LNCS*, 424–435. https://doi.org/10.1007/978-3-030-05710-7_35

Krestenitis, M., Orfanidis, G., Ioannidis, K., Avgerinakis, K., Vrochidis, S., & Kompatsiaris, I. (2019b). Oil spill identification from satellite images using deep neural networks. *Remote Sensing*, *11*(15). https://doi.org/10.3390/rs11151762

Krestenitis, M., Orfanidis, G., Ioannidis, K., Avgerinakis, K., Vrochidis, S., & Kompatsiaris, I. (2019c). Oil Spill Identification from Satellite Images using Deep Neural Networks. *Remote Sensing*, *11*(15). https://doi.org/10.3390/rs11151762

Kvenvolden, K. A., & Cooper, C. K. (2003). Natural Seepage of Crude Oil Into the Marine Environment. *Geo-Marine Letters*, *23*(3–4), 140–146. https://doi.org/10.1007/s00367-003-0135-0

Kwak, G. H., Park, C. W., Lee, K. Do, Na, S. Il, Ahn, H. Y., & Park, N. W. (2021). Potential of Hybrid CNN-RF Model for Early Crop Mapping with Limited Input Data. *Remote Sensing*, *13*(9). https://doi.org/10.3390/rs13091629

Lecun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. In *Nature* (Vol. 521, Issue 7553, pp. 436–444). Nature Publishing Group. https://doi.org/10.1038/nature14539

Lecun, Y., Bottou, E., Bengio, Y., & Haffner, P. (1998). Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE*, 2278–2324. https://doi.org/https://doi.org/10.1109/5.726791

Lee, S. J., Chen, T., Yu, L., & Lai, C. H. (2018). Image Classification Based on the Boost Convolutional Neural Network. *IEEE Access*, *6*, 12755–12768. https://doi.org/10.1109/ACCESS.2018.2796722

Lentini, C. A. D., DE MENDONÇA, L. F. F., Conceição, M. R. A., Lima, A. T. C., DE VASCONCELOS, R. N., & Porsani, M. J. (2022). Comparison Between Oil Spill Images and Look-Alikes: An Evaluation of SAR-Derived Observations of the 2019 Oil Spill Incident Along Brazilian Waters. *Anais Da Academia Brasileira de Ciencias*, *94*. https://doi.org/10.1590/0001-3765202220211207

Li, Y., & Li, J. (2010). Oil Spill Detection from SAR Intensity Imagery Using a Marked Point Process. *Remote Sensing of Environment*, *114*(7), 1590–1601. https://doi.org/https://doi.org/10.1016/j.rse.2010.02.013

Li, Y., Lyu, X., Frery, A. C., & Ren, P. (2021). Oil Spill Detection with Multiscale Conditional Adversarial Networks with Small-data Training. *Remote Sensing*, *13*(12). https://doi.org/10.3390/rs13122378

Li, Y., Yang, X., Ye, Y., Cui, L., Jia, B., Jiang, Z., & Wang, S. (2018). Detection of Oil Spill Through Fully Convolutional Network. *Communications in Computer and Information Science*, *848*, 353–362. https://doi.org/10.1007/978-981-13-0893-2_38

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 431–440. https://doi.org/https://doi.org/10.1109/CVPR.2015.7298965

Magri, S., Vairo, T., Reverberi, A. P., & Fabiano, B. (2021). Oil Spill Identification and Monitoring from Sentinel-1 SAR satellite earth observations: A machine learning approach. In *Chemical Engineering Transactions* (Vol. 86, pp. 379–384). Italian Association of Chemical Engineering - AIDIC. https://doi.org/10.3303/CET2186064

Mahmoud, A. S., Mohamed, S. A., El-Khoriby, R. A., AbdelSalam, H. M., & El-Khodary, I. A. (2023). Oil Spill Identification based on Dual Attention UNet Model Using Synthetic Aperture Radar Images.

Journal of the Indian Society of Remote Sensing, *51*(1), 121–133. https://doi.org/10.1007/s12524-022-01624-6

Martins, J., Nogueira, K., Zamboni, P., de Oliveira, P. T. S., Gonçalves, W. N., dos Santos, J. A., & Junior, J. M. (2021). Segmentation of Tree Canopies in Urban Environments Using Dilated Convolutional Neural Network. *International Geoscience and Remote Sensing Symposium (IGARSS)*, *2021-July*, 6932–6935. https://doi.org/10.1109/IGARSS47720.2021.9553218

Matkan, A. A., Hajeb, M., & Azarakhsh, Z. (2013, October 8). Oil Spill Detection from SAR Image Using SVM Based Classification. *International Archives of the Photogrammetry, Remote Sensing, and Spatial Information Sciences*. https://doi.org/https://doi.org/10.5194/isprsarchives-XL-1-W3-55-2013

Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review. *Remote Sensing*, *13*(13). https://doi.org/10.3390/rs13132450

Muin, M., Muslim, A. B., & Puspitasari, T. A. (2022). The Effect of Non-Linear Wave on Oil Spill Dispersion. *IOP Conference Series: Earth and Environmental Science*, *1065*(1). https://doi.org/10.1088/1755-1315/1065/1/012006

Nagalakshmi, R., Rameshwaran, P. M., & Santhosh, R. (2018). Preliminary Water Sample Analysis in Ennore Oil Spill Area. *Rasayan Journal of Chemistry*, *11*(1), 181–186. https://doi.org/10.7324/RJC.2018.1111774

Nair, A., Srujan, K. S. S. S., Kulkarni, S. R., Alwadhi, K., Jain, N., Kodamana, H., Sandeep, S., & John, V. O. (2022). A Deep Learning Framework for the Detection of Tropical Cyclones from Satellite Images. *IEEE Geoscience and Remote Sensing Letters*, *19*. https://doi.org/10.1109/LGRS.2021.3131638

Najmi, A., Gevaert, C. M., Kohli, D., Kuffer, M., & Pratomo, J. (2022). Integrating Remote Sensing and Street View Imagery for Mapping Slums. *ISPRS International Journal of Geo-Information*, *11*(12). https://doi.org/10.3390/ijgi11120631

Nijhawan, R., Das, J., & Balasubramanian, R. (2018). A Hybrid CNN + Random Forest Approach to Delineate Debris Covered Glaciers Using Deep Features. *Journal of the Indian Society of Remote Sensing*, *46*(6), 981–989. https://doi.org/10.1007/s12524-018-0750-x

Ordinioha, B., & Brisibe, S. (2013). The Human Health Implications of Crude Oil Spills in the Niger Delta, Nigeria: An Interpretation of Published Studies. *Nigerian Medical Journal*, *54*(1), 10. https://doi.org/10.4103/0300-1652.108887

Orfanidis, G., Ioannidis, K., Avgerinakis, K., Vrochidis, S., & Kompatsiaris, I. (2018, October 10). A Deep Neural Network for Oil Spill Semantic Segmentation in SAR Images. *IEEE International Conference on Image Processing*.

O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *ArXiv*. https://doi.org/https://doi.org/10.48550/arXiv.1511.08458

Pasupa, K., & Sunhem, W. (2016, February 28). A Comparison Between Shallow and Deep Architecture Classifiers on Small Dataset. *8th International Conference on Information Technology and Electrical Engineering*. https://doi.org/10.1109/ICITEED.2016.7863293

Pelizzari, S., & Bioucas-Dias, J. (2007, July 28). Oil Spill Segmentation of SAR Images Via Graph Cuts. *IEEE International Geoscience and Remote Sensing Symposium*. https://doi.org/10.1109/IGARSS.2007.4423048

Persello, C., & Stein, A. (2017). Deep Fully Convolutional Networks for the Detection of Informal Settlements in VHR Images. *IEEE Geoscience and Remote Sensing Letters*, *14*(12), 2325–2329. https://doi.org/10.1109/LGRS.2017.2763738

Phady, A., Ramadani Rahim, F., Melati Suci, I., Muhammad Alfian Arafat, dan A., & Taufiqur Rachman. (2019, October 1). Kajian Teknologi Penanganan Kebocoran Pipa pada Bangunan Lepas Pantai di Laut Utara Karawang. *Seminar Sains Dan Teknologi Kelautan*.

Quintana, G. I., Li, Z., Vancamberg, L., Mougeot, M., Desolneux, A., & Muller, S. (2023). Exploiting Patch Sizes and Resolutions for Multi-Scale Deep Learning in Mammogram Image Classification. *Bioengineering*, *10*(5). https://doi.org/10.3390/bioengineering10050534

Rawat, A., Kumar, A., Upadhyay, P., & Kumar, S. (2021). Deep Learning-Based Models for Temporal Satellite Data Processing: Classification of Paddy Transplanted Fields. *Ecological Informatics*, *61*. https://doi.org/10.1016/j.ecoinf.2021.101214

Ren, X., Guo, H., Li, S., Wang, S., & Li, J. (2017). A Novel Image Classification Method with CNN-XGBoost Model. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *10431 LNCS*, 378–390. https://doi.org/10.1007/978-3-319-64185-0_28

Shamili, F. S., Khartika, N., Gayatri, C., & Geetha, K. (2022). Oil Spill Detection using CNN Algorithm. *International Journal of Research Publication and Reviews m*, *3*(5), 439–443. www.ijrpr.com

Shanmukh, M. P., Priya, S. B., & Madeswaran, T. (2024). Improving Oil Spill Detection in Marine Environments Through Deep Learning Approaches. *4th International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies, ICAECT 2024*. https://doi.org/10.1109/ICAECT60202.2024.10468888

Shao, Z., Ahmad, M. N., & Javed, A. (2024a). Comparison of Random Forest and XGBoost Classifiers Using Integrated Optical and SAR Features for Mapping Urban Impervious Surface. *Remote Sensing*, *16*(4). https://doi.org/10.3390/rs16040665

Shao, Z., Ahmad, M. N., & Javed, A. (2024b). Comparison of Random Forest and XGBoost Classifiers Using Integrated Optical and SAR Features for Mapping Urban Impervious Surface. *Remote Sensing*, *16*(4). https://doi.org/10.3390/rs16040665

Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support Vector Machine Versus Random Forest for Remote Sensing Image Classification: A Meta-Analysis and Systematic Review. In *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* (Vol. 13, pp. 6308–6325). Institute of Electrical and Electronics Engineers Inc. https://doi.org/10.1109/JSTARS.2020.3026724

Singha, S., Bellerby, T. J., & Trieschmann, O. (2012, August 27). Detection and Classification of Oil Spill and Look-Alike Spots from SAR Imagery using an Artificial Neural Network. *IEEE International Geoscience & Remote Sensing Symposium*. https://doi.org/10.1109/IGARSS.2012.6352042

Singha, S., Vespe, M., & Trieschmann, O. (2013). Automatic Synthetic Aperture Radar based Oil Spill Detection and Performance Estimation via a Semi-Automatic Operational Service Benchmark. *Marine Pollution Bulletin*, *73*(1), 199–209. https://doi.org/10.1016/j.marpolbul.2013.05.022

Solberg, A. H. S. (2012). Remote sensing of ocean oil-spill pollution. *Proceedings of the IEEE*, *100*(10), 2931–2945. https://doi.org/10.1109/JPROC.2012.2196250

Sommervold, O., Gazzea, M., & Arghandeh, R. (2023). A Survey on SAR and Optical Satellite Image Registration. *Remote Sensing*, *15*(3). https://doi.org/10.3390/rs15030850

Song, D., Huang, Q., Gao, H., Wang, B., Zhang, J., & Chen, W. (2024). Adaptive Oil Spill Detection Network for Scene-based PolSAR Data using Dynamic Convolution and Boundary Constraints. *International Journal of Applied Earth Observation and Geoinformation*, *130*. https://doi.org/10.1016/j.jag.2024.103914

Song, D., Zhen, Z., Wang, B., Li, X., Gao, L., Wang, N., Xie, T., & Zhang, T. (2020). A Novel Marine Oil Spillage Identification Scheme Based on Convolution Neural Network Feature Extraction from Fully Polarimetric SAR Imagery. *IEEE Access*, *8*, 59801–59820. https://doi.org/10.1109/ACCESS.2020.2979219

Stofa, M. M., Zulkifley, M. A., & Muhammad Zaki, S. Z. (2020). A Deep Learning Approach to Ship Detection using Satellite Imagery. *IOP Conference Series: Earth and Environmental Science*, *540*(1). https://doi.org/10.1088/1755-1315/540/1/012049

Sudha, V., & Vijendran, A. S. (2021). Oil Spill Discrimination of SAR Satellite Images Using Deep Learning Based Semantic Segmentation. *International Conference on Computing Science, Communication and Security*, 127–139. https://doi.org/10.1007/978-3-030-76776-1_9

Sudiana, D., Lestari, A. I., Riyanto, I., Rizkinia, M., Arief, R., Prabuwono, A. S., & Sri Sumantyo, J. T. (2023). A Hybrid Convolutional Neural Network and Random Forest for Burned Area Identification with Optical and Synthetic Aperture Radar (SAR) Data. *Remote Sensing*, *15*(3). https://doi.org/10.3390/rs15030728

Tanha, J., Abdi, Y., Samadi, N., Razzaghi, N., & Asadpour, M. (2020). Boosting Methods for Multi-Class Imbalanced Data Classification: An Experimental Review. *Journal of Big Data*, *7*(1). https://doi.org/10.1186/s40537-020-00349-y

Tong, S., Liu, X., Chen, Q., Zhang, Z., & Xie, G. (2019). Multi-Feature based Ocean Oil Spill Detection for Polarimetric SAR Data using Random Forest and the Self-Similarity Parameter. *Remote Sensing*, *11*(4). https://doi.org/10.3390/rs11040451

Topouzelis, K. N. (2008). Oil Spill Detection by SAR Images: Dark Formation Detection, Feature Extraction and Classification Algorithms. *Sensors*, *8*(10), 6642–6659. https://doi.org/10.3390/s8106642

Vasconcelos, R. N., Lima, A. T. C., Lentini, C. A. D., Miranda, J. G. V., de Mendonça, L. F. F., Lopes, J. M., Santana, M. M. M., Cambuí, E. C. B., Souza, D. T. M., Costa, D. P., Duverger, S. G., & Franca-Rocha, W. S. (2023). Deep Learning-Based Approaches for Oil Spill Detection: A Bibliometric Review of Research Trends and Challenges. In *Journal of Marine Science and Engineering* (Vol. 11, Issue 7). Multidisciplinary Digital Publishing Institute (MDPI). https://doi.org/10.3390/jmse11071406

Volpi, M., & Tuia, D. (2017). Dense Semantic Labeling of Subdecimeter Resolution Images with Convolutional Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing*, *55*(2), 881–893. https://doi.org/10.1109/TGRS.2016.2616585

Wang, S., Wu, X., Zhang, Y., Liu, X., & Zhao, L. (2020). A Neural Network Ensemble Method for Effective Crack Segmentation using Fully Convolutional Networks and Multi-scale Structured Forests. *Machine Vision and Applications*, *31*(7–8). https://doi.org/10.1007/s00138-020-01114-0

Wang, W., Chakraborty, G., & Chakraborty, B. (2021). Predicting the Risk of Chronic Kidney Disease (CKD) using Machine Learning Algorithm. *Applied Sciences (Switzerland)*, *11*(1), 1–17. https://doi.org/10.3390/app11010202

Wang, Y., Wang, C., & Zhang, H. (2018). Ship Classification in High-Resolution SAR Images Using Deep Learning of Small Datasets. *Sensors (Switzerland)*, *18*(9). https://doi.org/10.3390/s18092929

Yaloveha, V., Podorozhniak, A., & Kuchuk, H. (2022). Convolutional Neural Network Hyperparameter Optimization Applied to Land Cover Classification. *Radioelectronic and Computer Systems*, *2022*(1), 115–128. https://doi.org/10.32620/reks.2022.1.09

Yamashita, R., Nishio, M., Do, R. K. G., & Togashi, K. (2018). Convolutional Neural Networks: An Overview and Application in Radiology. *Insights into Imaging*, *9*(4), 611–629. https://doi.org/10.1007/s13244-018-0639-9

Yang, Y. J., Singha, S., & Mayerle, R. (2022). A Deep Learning Based Oil Spill Detector using Sentinel-1 SAR Imagery. *International Journal of Remote Sensing*, *43*(11), 4287–4314. https://doi.org/10.1080/01431161.2022.2109445

Younis, M. C., & Keedwell, E. (2019). Semantic Segmentation on Small Datasets of Satellite Images using Convolutional Neural Networks. *Journal of Applied Remote Sensing*, *13*(04), 1. https://doi.org/10.1117/1.jrs.13.046510

Yu, C., Bi, X., & Fan, Y. (2023). Deep Learning for Fluid Velocity Field Estimation: A Review. *Ocean Engineering*, *271*. https://doi.org/10.1016/j.oceaneng.2023.113693

Yu, F., & Koltun, V. (2015). Multi-Scale Context Aggregation by Dilated Convolutions. *ArXiv*. https://doi.org/https://doi.org/10.48550/arXiv.1511.07122

Yu, X., & Lary, D. J. (2021). Cloud Detection Using An Ensemble of Pixel-Based Machine Learning Models Incorporating Unsupervised Classification. *Remote Sensing*, *13*(16). https://doi.org/10.3390/rs13163289

Zakzouk, M., El-Magd, I. A., Ali, E. M., Abdulaziz, M. A., Rehman, A., Rehman, A., & Saba, T. (2021). Novel Oil Spill Indices for Sentinel-2 Imagery: A Case Study of Natural Seepage in Qaruh Island, Kuwait. *MethodsX*, *8*. https://doi.org/10.1016/j.mex.2021.101327

Zhang, P., Jia, Y., & Shang, Y. (2022). Research and Application of XGBoost in Imbalanced Data. *International Journal of Distributed Sensor Networks*, *18*(6). https://doi.org/10.1177/15501329221106935

Zhen, J., Mao, D., Shen, Z., Zhao, D., Xu, Y., Wang, J., Jia, M., Wang, Z., & Ren, C. (2024a). Performance of XGBoost Ensemble Learning Algorithm for Mangrove Species Classification with Multi-Source Spaceborne Remote Sensing Data. *Journal of Remote Sensing*. https://doi.org/10.34133/remotesensing.0146

Zhen, J., Mao, D., Shen, Z., Zhao, D., Xu, Y., Wang, J., Jia, M., Wang, Z., & Ren, C. (2024b). Performance of XGBoost Ensemble Learning Algorithm for Mangrove Species Classification with Multi-Source Spaceborne Remote Sensing Data. *Journal of Remote Sensing*. https://doi.org/10.34133/remotesensing.0146

Zhou, G., Ni, Z., Zhao, Y., & Luan, J. (2022). Identification of Bamboo Species Based on Extreme Gradient Boosting (XGBoost) Using Zhuhai-1 Orbita Hyperspectral Remote Sensing Imagery. *Sensors*, *22*(14). https://doi.org/10.3390/s22145434

Zhou, H., & Peng, C. (2018). Oil Spills Identification in SAR Image Using mRMR and SVM Model. *Proceedings - 2018 5th International Conference on Information Science and Control Engineering, ICISCE 2018*, 355–359. https://doi.org/10.1109/ICISCE.2018.00081

Zhu, Q., He, Z., Zhang, T., & Cui, W. (2020). Improving Classification Performance of Softmax Loss Function Based on Scalable Batch-Normalization. *Applied Sciences (Switzerland)*, *10*(8). https://doi.org/10.3390/APP10082950

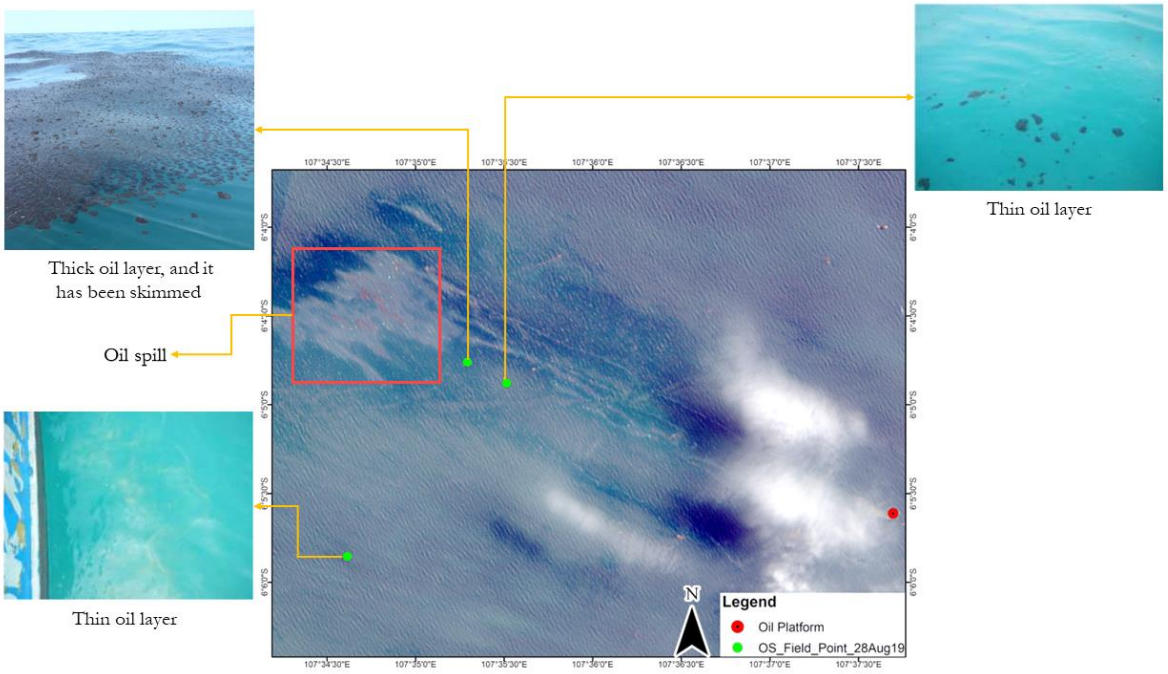
Zou, Y., Shi, L., Zhang, S., Liang, C., & Zeng, T. (2016). Oil Spill Detection by a Support Vector Machine based on Polarization Decomposition Characteristics. *Acta Oceanologica Sinica*, *35*(9), 86–90. https://doi.org/10.1007/s13131-016-0935-5

APPENDIX

Appendix 1: Ground truth information for Indonesian dataset

Karawang

Date	Sentinel-1 satellite data
July, 18th 2019	S1A_IW_GRDH_1SDV_20190718T111514_20190718T111539_028170_032E9E_8AD1
July, 30th 2019	S1A_IW_GRDH_1SDV_20190730T111514_20190730T111539_028345_0333FB_3615
August, 11th 2019	S1A_IW_GRDH_1SDV_20190811T111515_20190811T111540_028520_03397F_CEB8
August, 19th 2019	S1A_IW_GRDH_1SDV_20190819T223326_20190819T223351_028644_033DBE_7ABF
August, 23rd 2019	S1A_IW_GRDH_1SDV_20190823T111516_20190823T111541_028695_033F86_ED5E
August, 31st 2019	S1A_IW_GRDH_1SDV_20190831T223327_20190831T223352_028819_0343DC_3E1A
September 4th 2019	S1A_IW_GRDH_1SDV_20190904T111516_20190904T111541_028870_0345A8_DF0F
September 12th 2019	S1A_IW_GRDH_1SDV_20190912T223327_20190912T223352_028994_0349EF_FC80



Field observations overlaid with Sentinel-2 image in the same timestep (28th August 2019, 10:28 local time)

Historical reports from authority in the field (Abimanyu et al., 2021)

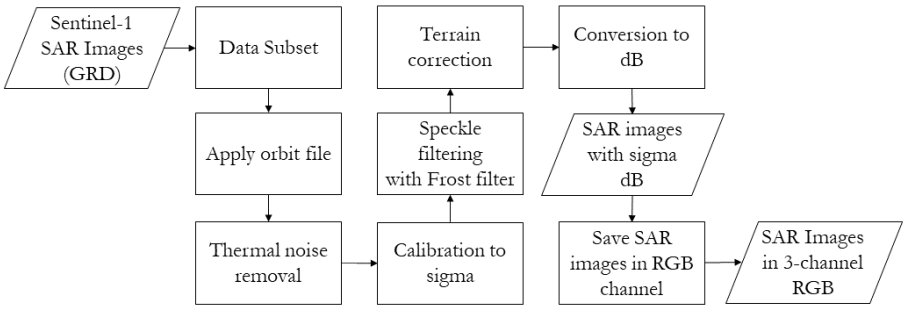
Date	Report from Authority
July, 16th 2019	Oil sheen visible at the sea surface
July, 18th 2019	Oil sheen starts to spread towards the west of YYA-1 Rig
July, 30th 2019	8 villages in Karawang and Bekasi are affected
August, 11th 2019	The oil boom and emergency posts installation along the coastline of six villages, cleaning of the oil spill in the affected villages
August, 23rd 2019	6,825 meters of shoreline oil boom along Karawang towards Seribu Islands
September, 4th 2019	7,995 meters of shoreline oil boom along Karawang, Bekasi, and Seribu Islands, 9 health services posts in Karawang Regency
September, 16th 2019	9,950 metres of shoreline oil boom along Karawang, Bekasi, and Seribu Islands, cleaning of the oil spill in the affected villages

Balikpapan

Date	Balikpapan
April, 1st 2018	S1A_IW_GRDH_1SDV_20180401T215050_20180401T215115_021279_0249B3_2C21
	S1A_IW_GRDH_1SDV_20180401T215115_20180401T215140_021279_0249B3_7FED

Ground truth information	Source
- The oil spill incident in Balikpapan Bay, occurred on Saturday, March 31st 2018 at 02:00 local time. - The oil spill occurred around 3 km from the Pig launcher station at coordinates 116°47'22.96" E; 1°14'44.75" S. - The estimated oil spill is approximately 44, 868 barrels	East Kalimantan Provincial Government (https://www.kaltimprov.go.id/berita/pernyataan-resmi-pemprov-kaltim-tentang-tumpahan-minyak-di-teluk-Balikpapan)
- The area affected by the oil spill is estimated to be approximately 7,000 hectares, with the length of the impacted coastline extending about 60 kilometers, encompassing parts of Balikpapan City and North Penajam Pasir Regency	Ministry of Environment and Forestry of the Republic of Indonesia (Kementerian Lingkungan Hidup dan Kehutanan Republik Indonesia, 2018)

Appendix 2: Flowchart for Sentinel-1 SAR image pre-processing

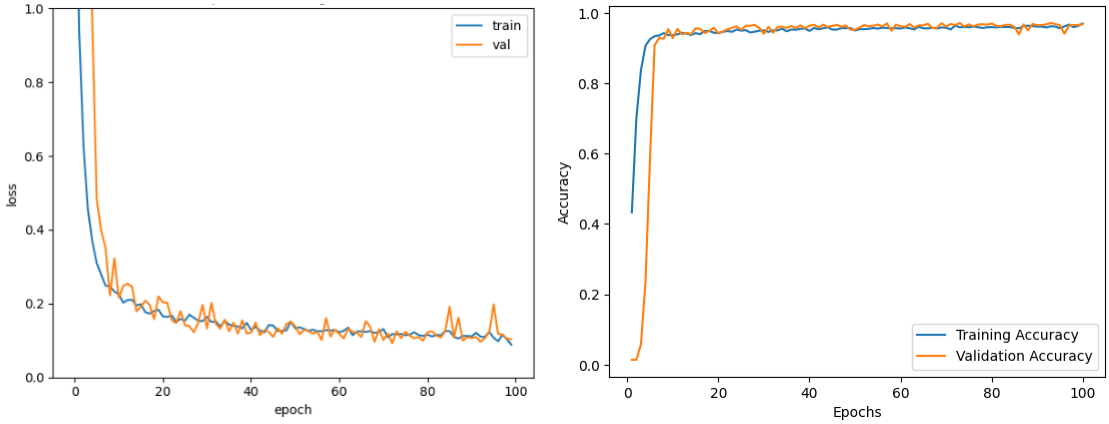


Appendix 3: FCN-DK6 Architecture

Layer (type)	Output Shape	Param #
zero_padding2d (ZeroPadding2D)	(None, 516, 516, 3)	0
conv2d (Conv2D)	(None, 512, 512, 16)	1216
batch_normalization (BatchNormalization)	(None, 512, 512, 16)	64
leaky_re_lu (LeakyReLU)	(None, 512, 512, 16)	0
zero_padding2d_1 (ZeroPadding2D)	(None, 516, 516, 16)	0
max_pooling2d (MaxPooling2D)	(None, 512, 512, 16)	0
zero_padding2d_2 (ZeroPadding2D)	(None, 520, 520, 16)	0
conv2d_1 (Conv2D)	(None, 512, 512, 32)	12832
batch_normalization_1 (BatchNormalization)	(None, 512, 512, 32)	128
leaky_re_lu_1 (LeakyReLU)	(None, 512, 512, 32)	0
zero_padding2d_3 (ZeroPadding2D)	(None, 520, 520, 32)	0
max_pooling2d_1 (MaxPooling2D)	(None, 512, 512, 32)	0
max_pooling2d_3 (MaxPooling2D)	(None, 512, 512, 32)	0
zero_padding2d_8 (ZeroPadding2D)	(None, 532, 532, 32)	0
conv2d_4 (Conv2D)	(None, 512, 512, 32)	25632
batch_normalization_4 (BatchNormalization)	(None, 512, 512, 32)	128
leaky_re_lu_4 (LeakyReLU)	(None, 512, 512, 32)	0
zero_padding2d_9 (ZeroPadding2D)	(None, 532, 532, 32)	0
max_pooling2d_4 (MaxPooling2D)	(None, 512, 512, 32)	0
zero_padding2d_10 (ZeroPadding2D)	(None, 536, 536, 32)	0
conv2d_5 (Conv2D)	(None, 512, 512, 32)	25632
batch_normalization_5 (BatchNormalization)	(None, 512, 512, 32)	128
leaky_re_lu_5 (LeakyReLU)	(None, 512, 512, 32)	0
zero_padding2d_11 (ZeroPadding2D)	(None, 536, 536, 32)	0
max_pooling2d_5 (MaxPooling2D)	(None, 512, 512, 32)	0
conv2d_6 (Conv2D)	(None, 512, 512, 5)	165
activation (Activation)	(None, 512, 512, 5)	0

Total params: 117,445
Trainable params: 117,093
Non-trainable params: 352

Appendix 4: Loss curve and accuracy curve of FCN-DK6 during training

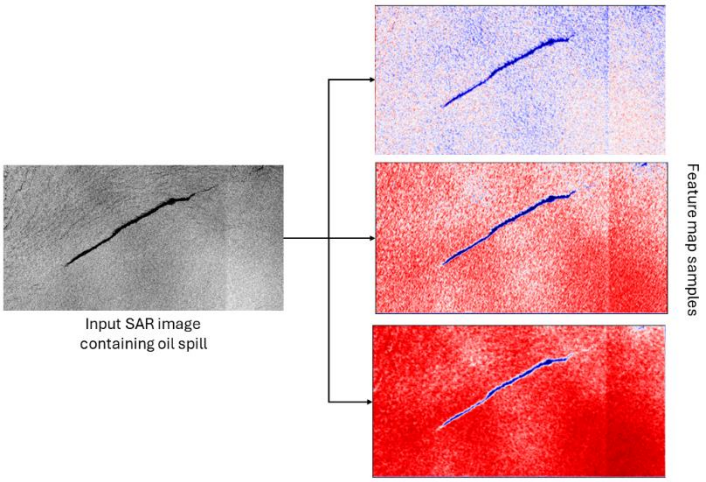


Appendix 5: CNN architecture in the hybrid CNN-XGBoost model

Layer (type)	Output Shape	Param #
zero_padding2d_8 (ZeroPadding2D)	(None, 262, 502, 3)	0
conv2d_4 (Conv2D)	(None, 260, 500, 32)	896
batch_normalization_4 (BatchNormalization)	(None, 260, 500, 32)	128
leaky_re_lu_4 (LeakyReLU)	(None, 260, 500, 32)	0
zero_padding2d_9 (ZeroPadding2D)	(None, 262, 502, 32)	0
max_pooling2d_4 (MaxPooling2D)	(None, 260, 500, 32)	0
zero_padding2d_10 (ZeroPadding2D)	(None, 262, 502, 32)	0
conv2d_5 (Conv2D)	(None, 260, 500, 64)	18496
batch_normalization_5 (BatchNormalization)	(None, 260, 500, 64)	256
leaky_re_lu_5 (LeakyReLU)	(None, 260, 500, 64)	0
zero_padding2d_11 (ZeroPadding2D)	(None, 262, 502, 64)	0
max_pooling2d_5 (MaxPooling2D)	(None, 260, 500, 64)	0
zero_padding2d_12 (ZeroPadding2D)	(None, 262, 502, 64)	0
conv2d_6 (Conv2D)	(None, 260, 500, 128)	73856
batch_normalization_6 (BatchNormalization)	(None, 260, 500, 128)	512
leaky_re_lu_6 (LeakyReLU)	(None, 260, 500, 128)	0
zero_padding2d_13 (ZeroPadding2D)	(None, 262, 502, 128)	0
max_pooling2d_6 (MaxPooling2D)	(None, 260, 500, 128)	0
zero_padding2d_14 (ZeroPadding2D)	(None, 262, 502, 128)	0
conv2d_7 (Conv2D)	(None, 260, 500, 256)	295168
batch_normalization_7 (BatchNormalization)	(None, 260, 500, 256)	1024
leaky_re_lu_7 (LeakyReLU)	(None, 260, 500, 256)	0
zero_padding2d_15 (ZeroPadding2D)	(None, 262, 502, 256)	0
max_pooling2d_7 (MaxPooling2D)	(None, 260, 500, 256)	0

Total params: 390,336
Trainable params: 389,376
Non-trainable params: 960

Appendix 6: Sample of feature maps generated by CNN



Appendix 7: Transformation of feature maps into structured tabular data for XGBoost input

	0	1	2	3	4	5	6	7	8	9	...	247	248	249	250	251	252	253	254	255	Label
0	0.0	19.487734	22.150867	22.064484	0.0	11.765628	20.923546	70.397659	0.000000	1.232345	...	0.0	27.865894	16.276020	0.0	28.479055	96.790146	32.903625	0.000000	0.0	0
1	0.0	19.487734	22.150867	22.064484	0.0	15.740504	20.923546	70.397659	0.000000	1.232345	...	0.0	27.865894	18.305979	0.0	28.479055	97.010490	32.903625	0.000000	0.0	0
2	0.0	3.609415	15.394082	22.163540	0.0	18.188093	17.008894	70.397659	0.000000	0.000000	...	0.0	27.865894	18.305979	0.0	28.479055	97.010490	29.118706	0.000000	0.0	0
3	0.0	3.609415	17.451153	24.749775	0.0	18.757120	14.975723	67.872505	0.000000	0.000000	...	0.0	23.724115	18.305979	0.0	26.412462	97.010490	29.118706	0.000000	0.0	0
4	0.0	3.609415	18.512156	26.740328	0.0	18.757120	6.263033	59.051132	0.000000	0.000000	...	0.0	23.724115	17.605415	0.0	26.081329	92.141754	26.022936	0.000000	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
25999995	0.0	11.122479	3.013164	10.884351	0.0	12.146110	6.851838	16.515093	1.051060	0.000000	...	0.0	0.000000	19.010429	0.0	0.545989	36.170189	7.360895	0.000000	0.0	0
25999996	0.0	10.888323	4.287965	10.955470	0.0	11.954576	6.484528	15.860556	2.508451	0.000000	...	0.0	0.000000	18.385916	0.0	0.545989	36.196274	6.714584	0.000000	0.0	0
25999997	0.0	10.888323	4.287965	11.863730	0.0	12.044317	5.935301	13.357375	2.508451	0.000000	...	0.0	0.000000	19.565844	0.0	2.398489	36.196274	6.714584	0.000000	0.0	0
25999998	0.0	15.008539	4.287965	13.941563	0.0	12.044317	7.172094	16.322630	2.508451	0.000000	...	0.0	8.225703	19.565844	0.0	8.244925	36.196274	6.714584	3.992197	0.0	0
25999999	0.0	15.008539	2.675263	13.941563	0.0	12.044317	7.172094	16.322630	2.504194	0.000000	...	0.0	8.225703	19.565844	0.0	8.244925	35.772961	3.435331	3.992197	0.0	0

In this tabular data, rows represent the number of pixels from input images, while columns represent the number of feature maps