# Ship and Oil Spill Detection using Convolutional Autoencoder

Master's Thesis
VGIS1044

Mohit Singh Dhaka

Electronics and IT
Aalborg University
10th Semester Vision, Graphics and Interactive Systems
Frederik Bajers Vej 7
DK-9000 Aalborg

AALBORG UNIVERSITY
STUDENT REPORT

**10th Semester**
**Vision, Graphics and**
**Interactive Systems**
Electronics and IT
Aalborg University
Frederik Bajers Vej 7
9000 Aalborg
http://www.aau.dk

**Title:**
Ship and Oil Spill Detection using Convolutional Autoencoder

**Theme:**
Vision, Graphics and Interactive Systems

**Project Period:**
Spring Semester 2022

**Project Group:**
VGIS1044

**Participant(s):**
Mohit Singh Dhaka

**Supervisor(s):**
Mark Philip Philipsen
Anders Skaarup Johansen

**Page Numbers:** 40

**Date of Completion:**
June 1, 2022

**Abstract:**

Maritime surveillance has seen a growing interest in past few years owing to the fact that marine traffic has increased quite a lot, and securing a nation's maritime boundary is a crucial task. In this report we focus on two aspects of the maritime surveillance, namely ship detection and oil spill detection in SAR (Synthetic Aperture RADAR) satellite imagery. In this report the problem of ship and oil spill detection is formulated as that of an anomaly detection, where the ships and oil spills are the anomalies in the otherwise ocean background and use a deep Convolutional Autoencoder for the purpose of detecting these anomalies. Anomaly dataset was created from the xView3 competition dataset and SOS oil spill dataset for the training and testing of the autoencoder. The overall accuracy of the model is 96.64%, and the precision, recall and F1 score being 93.32%, 98.80% and 97.96%, respectively.

# Preface

This report is submitted as a thesis project for the Master's Programme in Vision, Graphics and Interactive Systems at Aalborg University under the supervision of Mark Philip Philipsen and Anders Skaarup Johansen. The focus of this thesis has been to develop a computer vision system that utilises several modules to detect ships and oil spills in SAR satellite images.

Citations done in this report follows APA-style, which means that the reference list is alphabetised, and citations are (Author's surname / Name of source, year).

I would also like to thank my supervisors Mark Philip Philipsen and Anders Skaarup Johansen for guidance during the thesis.

Aalborg University, June 1, 2022

# Contents

# Chapter 1

# Introduction

Due to the rise in the marine traffic, the interest in monitoring the maritime boundary is increasing for the safety and security of the resources and vessels. Maritime surveillance plays an important task to avoid activities like piracy operations, smuggling, illegal fishing [27] and is necessary for maritime traffic management, rescue operations, oil spill and toxic waste management and for military defence[6]. This is a very crucial task but is a difficult problem for all countries.

A whole array of sensors are used for the task of maritime surveillance like ground based Radars, patrol ships and aircrafts, AIS (Automatic identification systems) and remote sensing using satellites. The patrolling and terrestrial based systems have their limitations like AIS can be spoofed, many ships can't be detected by radars and patrolling the whole maritime boundary is a very expensive endeavour.

All these problems have created a need for solutions pertaining to the remote sensing using satellites and with the improvements in the sensing capabilities of satellites along with the advancements in electronics have enabled large scale datasets like those of satellite images to be trained on GPUs, and the deep learning methods have gotten better and many object detection models like Faster R-CNN and YOLO can detect images in real time. Hence, this seems to be a good avenue to explore and innovate.

In this report we will focus on creating a deep learning method that can be used for maritime surveillance. We will specifically focus on ship detection and oil spill detection in satellite imagery.

# Chapter 2

# Background

This chapter covers the background information necessary to gain a fundamental understanding of the detection of ships and oil spills using satellite imagery.

## 2.1 Remote Sensing

Remote sensing refers to process of monitoring and detecting features of an area from a satellite or an aircraft by recording reflected or emitted radiation. It has several applications like geographical surveys, ocean depth measurements, weather monitoring, wildfire monitoring, tracking of agricultural and urban growth, etc. Remote sensors provide a ton of data about different aspects of the human life on the planet Earth and this can help humanity make data informed decisions [7].

Satellites move around Earth in basically three classes of orbits, namely, low-Earth orbits, medium-Earth orbits and high-Earth orbits. Most of the Earth observation satellites are placed in the low-Earth orbit which is approximately 160 to 2000 kms above the surface of the planet. Microwaves band are higher in fre-

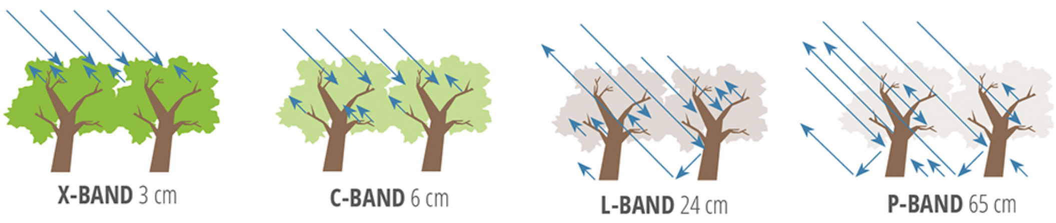


**Figure 2.1:** Different microwave bands and how they interact with vegetation [21]

quency compared to radio waves. Different sub bands of microwave have different functionalities in terms of satellite imagery. Like C-band can reveal Earth's surface by penetrating through, rain, dust, smoke and clouds. L-band is used in GPS

(Global Positioning System). X-Band is used for urban monitoring, it has difficulty penetrating through vegetation cover[19]. The Earth observation satellites can be divided into two categories based on their sensors. Active Sensors and Passive Sensors.

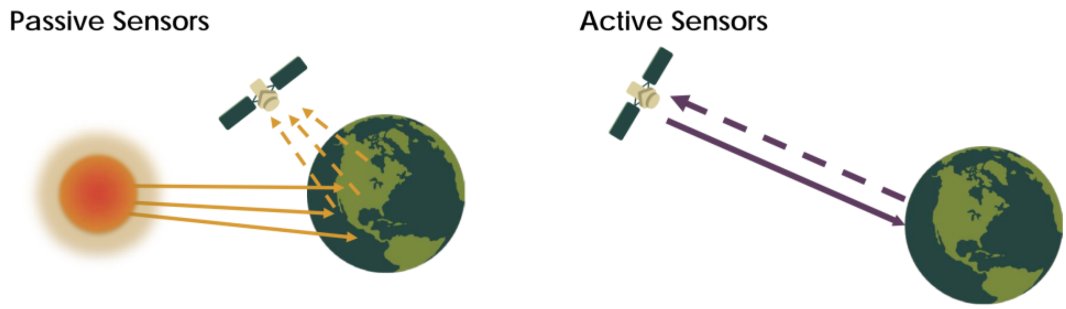


**Figure 2.2:** Diagram of active and passive sensors [19].

Active sensors aboard a satellite create their own source of energy like radio waves or microwaves. Most of the satellites having active sensors work in the microwave band of the electromagnetic spectrum. These sensors can work during both day and night, they can see through dense forest cover or clouds, they can be used in any weather condition. Example Sentinel-1 satellite [13].

Passive sensors depends on the Sun as their source of energy. They capture the reflected radiation bouncing back from the surface of the Earth. They can be used in multiple spectrum bands. Most common is the optical band which provides RGB images. But they can also work in the infrared and thermal band. Most of these sensors cannot work in rough weather and during night. Example Sentinel-2 satellite [13].

To decide which satellite data to work, there are three resolutions of the satellite that need to be kept in mind: spatial, spectral and temporal resolution. Spatial resolution refers to how much area is represented by each pixel, for example, if the spatial resolution is of 20 meters, that means each pixel covers and area of 20 square meters. Temporal resolution refers to how long does it take the satellite to be back on the same spot above the surface of Earth, for example, if the temporal resolution is of 12 days, that means it takes the satellite 12 days to provide information for a specific area. Spectral resolution refers to the fact that how many bands does the sensor aboard the satellite has to discern between the finer wavelengths. Hyperspectral images have hundreds of bands of wavelength that the sensor can record [9].

## 2.2 SAR

SAR stands for Synthetic Aperture Radar. The word synthetic means that the taking the advantage of the motion of the satellite and collecting data in short bursts, can mimic a large aperture and provide high resolution imagery despite having a smaller aperture.

SAR is a side looking satellite, meaning that unlike optical satellites which look down, SAR looks at an angle. SAR collects information after being scattered by the Earth's surface or the objects on the surface.

There are three types of scattering. Specular reflection, reflects back very less energy and so the object appears dark for example a body of water. Double bounce scattering, as the name suggests, the sensor receives the wave after it has been bounced twice, and so the object appears white, example buildings. Diffuse scattering happens when there is some specular and some double bounce scattering, the sensor neither receives high nor low backscatter. One major drawback of the SAR is that the images can get very noisy, the noise is salt and pepper speckle noise which can make detection of objects on SAR images very difficult[12].

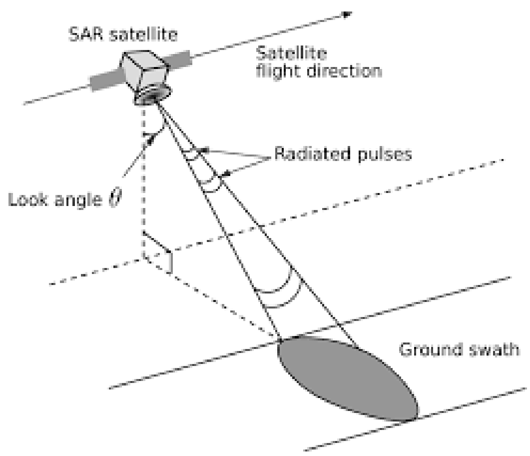


**Figure 2.3:** Working of a SAR satellite.

## 2.3 Sentinel-1

The Sentinel-1 is group of two polar orbiting, active sensor satellites which performs in the C-band of the microwave spectrum that is, it is a SAR satellite and collected information day and night and through any weather. The data collected by Sentinel-1 is freely accessible to anyone.

Sentinel-1 has a temporal resolution of 12 days, but since there are two satellites, the temporal resolution becomes 6 days. Since it is polar orbiting satellite, the resolution in Europe becomes 1-3 days. The spatial resolution varies from 10, 25 and 40 meters.

Sentinel-1 has four data acquisition modes:

- Wave (WV) this is the primary acquisition mode over open ocean
- Extra-wide swath (EW) this mode is used in the coastal regions for the monitoring of ships, oil spills and sea-ice.
- Interferometric Wide swath (IW) this is the primary mode of acquisition over land.
- Stripmap (SM) this mode is on request bases for exergencies.

Each of these modes have single and dual polarisation which can be selected while downloading the data. Polarisation is the direction of travel of the electromagnetic wave, it is vertical and horizontal. Sentinel-1 is designed to send and

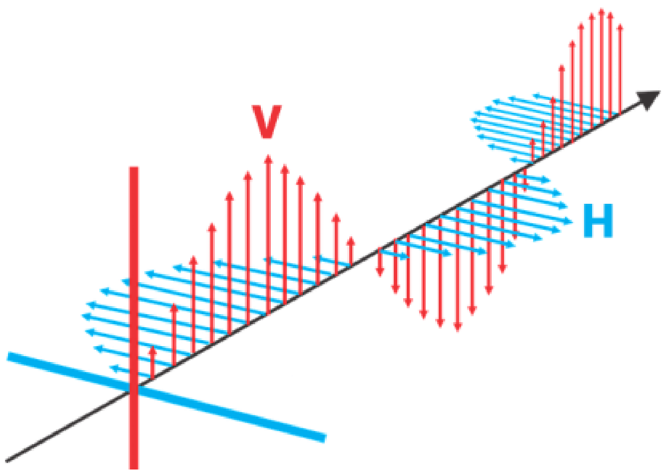


**Figure 2.4:** Polarisation in SAR imagery.

receive in both single polarisation and in dual polarisation as well. Sentinel-1 gathers very detailed information which can reveal information about structure, conditions and orientation of the surface objects. The available polarisations for Sentinel-1 are:

- VV Vertical transmission and receiving
- HH - Horizontal transmission and receiving
- HV - Horizontal transmission and Vertical receiving
- VH - Vertical transmission and Horizontal receiving
- HH+HV - Dual Polarisation
- VV+VH - Dual Polarisation

There are four different product types in which Sentinel-1 data is available:

- Level 0 this is the raw data, unfocused and compressed
- Level 1 Single Look Complex (SLC) complex images that preserve the amplitude and the phase
- Level 1 Ground Range Detected (GRD) gives data in full, high or medium resolution.
- Level 2 (OCN) provides geophysical parameters of the ocean

## 2.4 Related Work

The work presented here is not just limited to SAR images, many of the ship detection models were trained on optical satellite imagery. The papers that use unsupervised learning for the training of a ship detector.

Farr et al. [10] proposed a deep convolutional autoencoder for ship detection in optical satellite imagery captured in RBG band using Sentinel-2 satellite and converts the images to low resolution to duplicate the image resolution of nanosatellite. The purpose was to depict that such a model could be installed on nanosatellites and provide better coverage for the purposes of ship detection. They created their own dataset and got a precision of 98%.

Ferreira et al. [11] used a convolutional variational autoencoder for ship detection using the dataset created by [29] which is a SAR ship detection dataset collected using Chinese Gaofen-3 and Sentinel-1 satellite images. The paper also compared their results to a CNN of same architecture as the variational autoencoder and the accuracy achieved by variational autoencoder was 95.73% and using CNN got an accuracy of 97.6%.

Most of the papers use object detection model YOLO with a lot of innovation to improve upon the shortcomings of the YOLO for the ship detection.

Tang et al. [25] proposed a method that uses 2 modules before detecting ships in SAR satellite imagery using a YOLOv5. The fisrt module is a noise level classifying module which classifies the images as low, medium or high noise images. The next module was used to extract the potential area which had a ship. The final localisation of the ship was done using the YOLOv5 module.

To improve the ship detection capabilities in optical and SAR images of YOLOv3 and Masked-RCNN Tian et al. [26] proposed image enhancement and deep feature reuse modules. The image enhancement module was a fully convolutional network based on a Generative adversarial network to enhance the images containing ships. The dense feature reuses module concatenated the features from the previous layers to pass as the input to the next layers. Due to these modules the accuracy of the Masked-RCNN improved by 3% and for YOLO the improved was about 1%.

Chen et al. [4] proposed an attention module to improve the YOLOv3 detection method on optical imagery of ships. The attention module was used to enlarge the receptive fields to extract the salient feature maps, which helped in enhancing target regions and reducing interference due to clouds. This accelerated the detection speed to real-time.

Wang et al. [28] used optical imagery for ship detction using YOLOv3 network. They proposed to change the backbone of the YOLOv3 network to the squeeze-and-excitation structure which gave the YOLO network attention mechanism to enhance the importance of different channels hence improving the feature extraction capabilities which lead to a precision of 95.62% and recall of 95.32%.

Tang et al.[24] using optical imagery for ship detction using YOLOv3. The proposed an HSV(hue, saturation, value) module which comprised of four parts: background removal done using HSV characteristics of input images; noise removal enhanced the contract by applying thresholding; target selection was based on the fact that ship HSV was different from the background HSV, and noise deletion. This module was used to extract regions of interest in short amount of time.

Chen et al. [5] proposed YOLO-Lite for ship detection in SAR satellite imagery. They changed the backbone of the YOLO to incorporate asymmetric convolutional module and trimmed the redundant connections between the adjacent layers. Due to these changes the storage size of the model is reduced and speed is improved since the floating point operations are decreased.

Yang et al. [30] used SAR imagery to detect ships using RetinaNet with rotatable bounding boxes. The RetinaNet used here was a one-stage object detection framework to get the desired detection speed. A feature optimisation module containing task wise squeeze and excitation module was also used. They got a proportional distribution of feature maps using scale calibration. During the experimentation, the IoU threshold of 0.5 was obtained, and the average accuracy improved 13.26 percent, when compared to the other advanced RBox-based ship detection systems.

Hass et al.[14] proposed to use YOLOv3 on Sentinel-1 SAR imagery in the arctic waters to discriminate between the ships and icebergs. They created their own dataset and used different polarization as channels for a 3 channel input image and used a 53-layer deep YOLO architecture with residual connections. Iceberg detection had an accuracy of 51% and ship detection had accuracy of 70%.

Cao et al. [3] introduced a feature pyramid structure in YOLOv3 to link the deep semantic information with the shallow semantic information, and the multiscale feature mapping was incorporated to increase the detection capacity of ships in the aerial imagery.

Yue et al.[32] proposed a semi-supervised method for ship detection in SAR satellite imagery. They used a CNN to extract the class probabilities of the unlabeled data and then integrated it into the loss function of the CNN using Linear Discriminant Analysis.

Zhu et al. [37] uses SAR imagery to detect ships by reducing the speckle noise. They extracted the luminance information and made profile of the ship according to the luminance level considerable reducing the speckle noise. After thee noise removal used these luminance images to train a CNN classification model instead of using the raw SAR images.

Zhao et al .[35] created a coupled CNN for detecting small and densely grouped SAR ships. This approach is mostly made up of two sub-networks. They are the Exhaustive Ship Proposal Network (ESPN), which generates ship-like regions from many layers with multiple receptive fields, and the Accurate Ship Discrimination Network (ASDN), which eliminates warnings by referring to the context information of every proposal generated by ESPN.

Huang et al.[17] proposed a model based on upgraded regressive deep convolutional neural network. They used YOLOv2 feature extraction layer. They modified the structure of the pyramid network layer in YOLOv3, and a new feature pyramid network layer was created. They also used clustering techniques to reduce the number of anchors.

Zhang et al. [34] proposed a method called HyperLi-Net for ship detction in SAR images. It consisted of five external modules namely: Multi-Receptive-Field Module (MRF-Module) extracts all-around image information, Dilated Convolution Module (DC-Module) expands MRF-Module's receptive-field, Channel and Spatial Attention Module (CSA-Module) distinguishes important or inessential features, Feature Fusion Module (FF-Module) fuses shallow and deep features and Feature Pyramid Module (FP-Module) detects multi-scale ships. They also proposed five internal mechanisms, which were: Region-Free Model (RF-Model) avoids Region Of Interests (ROIs) generation, Small Kernel (S-Kernel) uses smaller kernels to reduce network parameters, Narrow Channel (N-Channel) uses fewer kernels to further reduce network parameters, Separable Convolution (Separa-Conv) uses Separa-Conv instead of traditional convolution and Batch Normalization Fusion (BN-Fusion) fuses BN into Separa-Conv in the detection model.

Zhao et al. [36] proposed a two stage detection model to detect ships on multi scale in SAR imagery called attention receptive pyramid network. The model used a combination of convolutional block attention module which is composed of channel and spatial attention mechanisms and receptive field block which contains many asymmetric kernel sized convolution layers to improve the connection between non-local features and refining information at different feature maps.

Oil spill detection happens better on SAR imagery, here are a few papers that use deep learning to detect oil spills.

Huang et al.[16] proposed a method to detect oil spill in SAR imagery. The method used is based on Faster R-CNN. They removed speckle noise using boxcar filtering. The region proposal network in this method consisted of fully convolutional network, instead of a image pyramid. The feature maps were generated using VGG-16 module. They got precision of 89.23% and recall of 89.14%.

Shaban et al.[23] proposed a two stage framework for detection of oil spills in SAR images.The first stage was used to classify the images into two groups based on the percentage of oil spill pixels in the image and the images having more than 1% of oil spill pixels were send for the second stage which was used for semantic segmentation using a five stage U-net. They got an accuracy of 92% and precision of 84%.

Zeng et al.[33] proposed a transfer learning approach and used a deep convolutional neural network based on VGG-16 for oil spill detection in SAR satellite imagery. They got an accuracy of 94.01% and precision of 85.70%.

Yekeen et al.[31] used a Mask R-CNN model having a ResNet 101 backbone trained on COCO dataset also having a feature pyramid network for the detection of oil spill in SAR images. They got an accuracy of 98.3%.

## 2.5 Summary

In this section we went through the background satellite information to understand which satellite is better for the purpose of maritime surveillance. Since, we need day and night and all weather coverage of the maritime boundaries, the Sentinel-1 mission is the best satellite for this purpose.

# Chapter 3

# Problem Statement

The project intends to assess SAR imagery from the satellite and create a marine surveillance model which can be linked with AIS system to get real time information about the vessels in the maritime border of a country.

The focus of the model will be on the following:

- it should be able to detect ships
- it should be able to detect oil spills.

The above mentioned will help solve the biggest concerns of ghost ships and AIS spoofing that can help to keep a check on the illegal fishing, smuggling and is also a concern for the military security of a nation.

Since the oil spill detection also comes under the umbrella of maritime surveillance, so the model should be able to detect oil spills as well. But as we have seen in the literature, there is no single consolidated method that can perform these different tasks (both ship detection and oil spill detection).

In this report the main focus is to make a deep learning model which can perform the above mentioned tasks.

The hypothesis of the project is stated as follows:

"The ocean is huge and so, the surface is mostly empty, therefore, the objects on the surface can be considered anomalies on the otherwise ocean background. If an autoencoder trained to reconstruct the ocean surface is provided with images of ships and oil spills which have different features, then the autoencoder should be able to detect them as anomalies".

The objective of this report is to make a deep convolutional autoencoder and test out the above mentioned hypothesis and analyse the shortcomings of such a method.

# Chapter 4

# Implementation

This chapter covers the hardware used for the training and inference of the model. The chapter will also cover the theory of the used model, and a description of the dataset created for the training and testing of the model.

## 4.1 Hardware

The hardware used for training the model and testing the performance of the model has the specifications seen in the table below.

| | |
|---|---|
| OS | Microsoft Windows 10 Home |
| GPU | Nvidea GeForce RTX 3060 |
| CUDA Version | 11.1 |
| Nvidea Driver | 512.95 |
| CPU | Intel(R) Core(TM) i7-11800H CPU 2.30GHz |
| RAM | 16 GB DDR4 |

**Table 4.1:** The specifications for the computer utilised for the experiment.

The GPU Geforce RTX 3060 has 6GB GDDR5 VRAM, worked sufficiently well for the training on satellite images. An alternative for training on larger datasets could be using services like Kaggle or Google Colab, which provides the possibility of training models on GPUs having more VRAM.

## 4.2 The Model

The model used in this report is a deep Convolutional Autoencoder. An Autoencoder has two parts, an encoder which takes the input and maps it to a latent space which is usually of lower dimensions as compared to the input. The second part is the decoder, which takes the latent space representation and tries to approximately reconstruct the input. The training of the autoencoder is done in an unsupervised manner, that is, no labels are provided, instead the input data is considered to be label and the main target of the autoencoder is to copy the input data approximately. The decoder is symmetric to the encoder in the architecture. The learning happens by minimising a loss function which measures how dissimilar the reconstruction is from the input [15].

The baseline model used in this report consists of three convolutional layers in the encoder, having a ReLU (Rectified Linear Unit) as the activation function. The final transpose convolutional layer of the decoder had softmax activation function. The latent space size was experimentally chosen to be 800. The kernel size used were 8, 16 and 32. Maxpooling2D layers were also used. In the final autoencoder model The activation function was changed to LeakyReLU to avoid the problem of dying ReLU [20]. Also Batch Normalisation layers were added to stabilise the learning process and Dropout layers were also added to avoid overfitting. Adam optimiser with a learning rate of 0.0001 was used for both the baseline and the final model. Figure 4.1 shows the architecture of the final model used for testing. NOTE: in all the upcoming chapters the term model represents the final model.

Two loss functions were used during testing the were:
Mean Square error loss function

$$MSE = \sum_{i=1}^{D} (x_i - y_i)^2$$

SSIM loss function is 1 - SSIM, it works by measuring the similarity of the input and the reconstructed image in terms of contrast, luminance and structure.[2]. The following is the formula to calculate SSIM.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1) + (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

Masked Mean Square Error was also used, it works by not allowing the masked values to have any impact on the learning of the model. It masks the pixels with certain value from the learning process. [1]

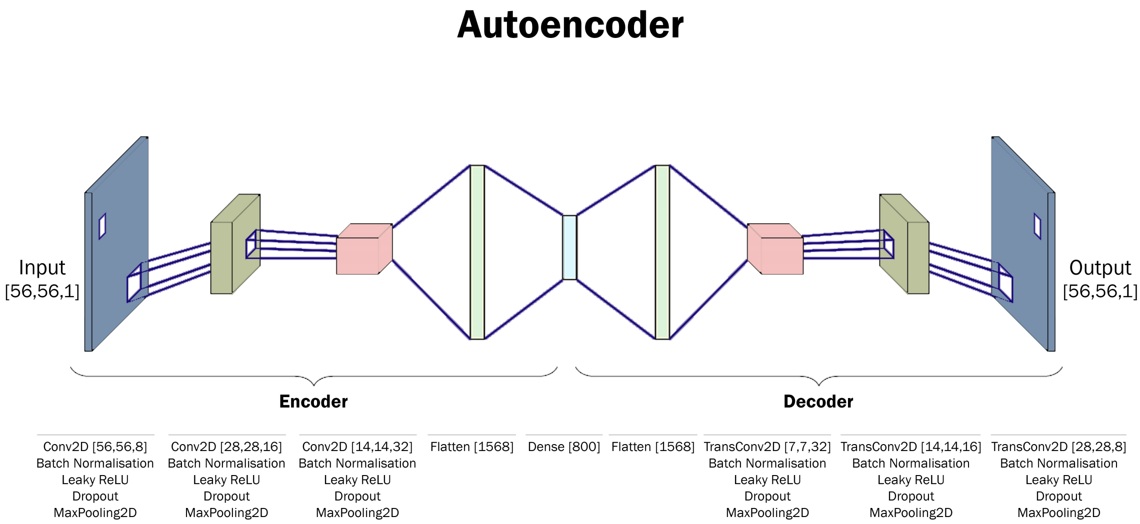


**Figure 4.1:** Architecture of the Autoencoder used as the final model

## 4.3 Dataset

In this report we have used three datasets, two for ship detection and one for oil spill. The following sub-sections go into more detail about each of the dataset.

### 4.3.1 A SAR dataset for ship detection

The dataset contains 43819 images each having ship(s) in them of size 256x256 along with labels provided by experts. The dataset was created using 108 Sentinel-1 images and 102 Chinese Gaofen-3 images. The dataset has complex backgrounds and a lot of images of shoreline [29]. A few images from the dataset are in the figures 4.2 and 4.3.

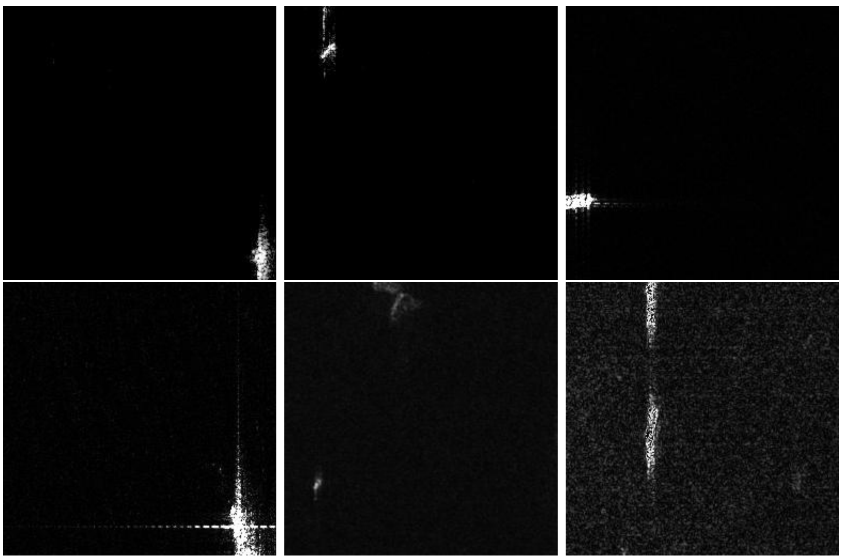

**Figure 4.2:** Example images of SAR ship detection dataset.

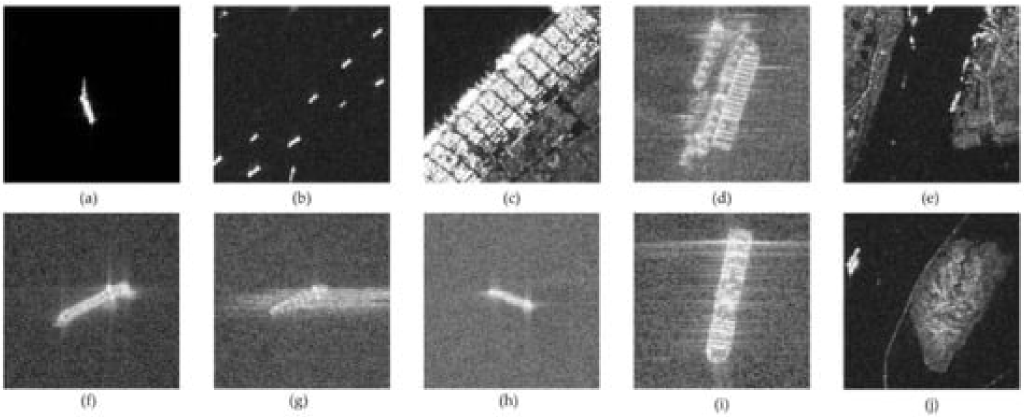

**Figure 4.3:** Example images from the paper [29].

The pre-processing done to remove the speckle noise which is very prominent in SAR images was not defined in the paper. No land/sea mask was provided with the dataset. Since there was no land/sea mask and most of the images didn't had any speckle noise, a simple process was designed to eliminate land from the images. The first step was to do thresholding using Otsu method which is a method to find adaptive threshold value. Otsu method was used because no one threshold value could be used for all the images. Then the process of "Closing" followed, which is "Dilation + Erosion" used to fill holes. After closing, the process of "Opening" was done, which is "Erosion + Dilation", this is used to remove small objects. After this process Connected Component Analysis was done with 8-connectivity. After doing this we get the number of BLOBs in the cropped image and since background is also considered as a BLOB, if the number of BLOBs was more than one, then the image was discarded. This process seemed to work well on a small trial set of 200 images.

For the training set, ships in the images were cropped out with the help of the labels provided. The cropped images were all of the size 128x128. The images which did not had any ships or landmass in them totaled 52800. And images with ships totalled 48382. To make the training set 30000 images were randomly selected from the no-ship images. 25% of these images were selected for validation dataset. And to make the test set 1000 images of both ship and no-ship class each were randomly selected, giving a total of 2000 images. This dataset was used because a lot of papers used this for ship detection and so good comparative study could be done.

### 4.3.2 xView3 dataset

xView3 was a competition held in 2021 to locate vessels in SAR imagery. It is a publicly available dataset, having train data of 500 scenes and the validation data has 50 scenes collected from Sentinel-1 Satellite. Each scene is an image of approximately 30000x30000. The data comes with labels which are taken from AIS information as well as manual annotation. The labels come with a confidence level, namely low, medium and high. The data also comes with ancillary imagery which contains the land/sea mask as well. The scenes were collected from a bunch of locations namely, Adriatic sea, Gulf of Guinea, seas around Ireland and southern Norway. The evaluation of the submissions were done using on test scenes whose labels were not made public so we cannot do a comparative study for this dataset [8].

No preprocessing was done and the images were in GeoTIFF format. The land/sea mask along with the ship labels were used to crop out images of size 256x256 for the training set. A total of 30000 images were cropped out of the 50 Validation scenes and 25% of the images were used for the validation dataset. For test set, 1000 images of size 256x256 having a high confidence level in labels were used to crop out images containing ships. Both the training and test set were made again with images of size 56x56 for the reason mentioned in section 6.2. A few examples of images with and without ships are in figures 4.4 and 4.5.

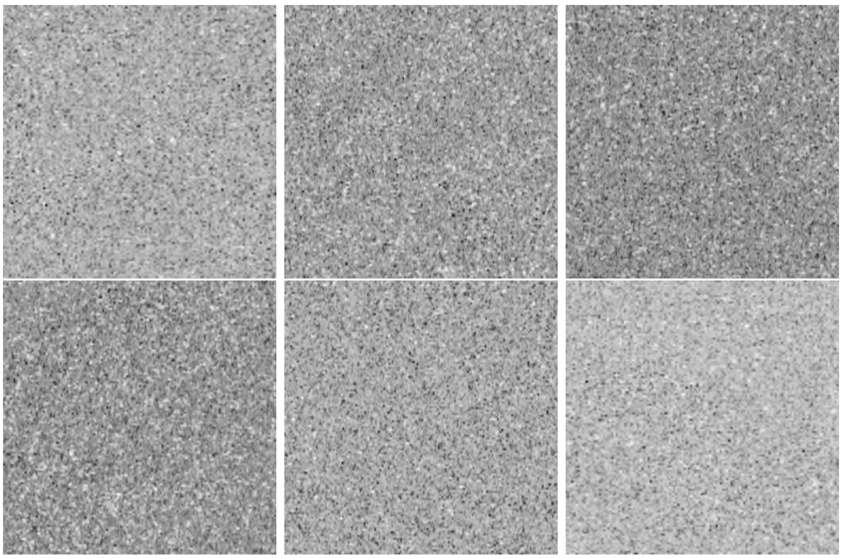

**Figure 4.4:** Example images of xView3 training data containing no ships.

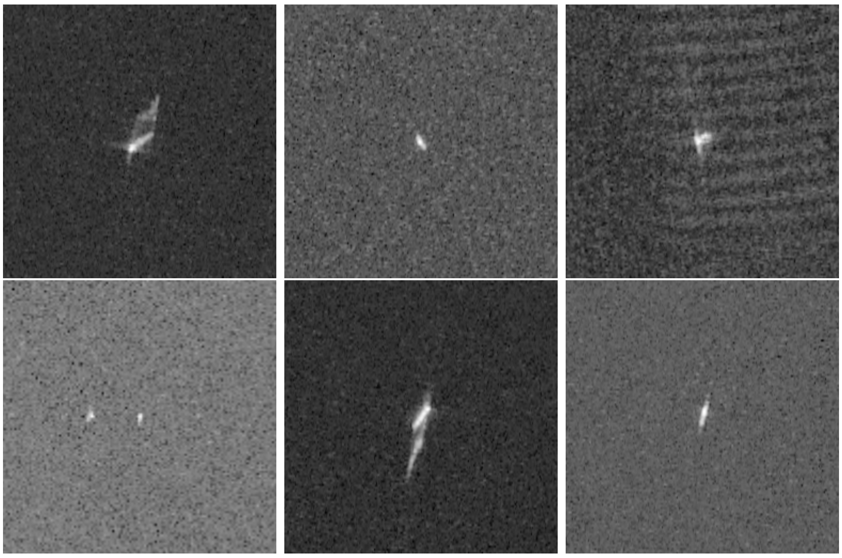

**Figure 4.5:** Example images of xView3 test data containing ships.

### 4.3.3 SOS: Deep SAR Oil Spill Dataset

This dataset contains images of oil spill from Gulf of Mexico and Persian Gulf. The images collected from Sentinel-1 Satellite and PALSAR Satellite. We only use the images collected from Sentinel-1 for our testing. Speckle filtering was done on the raw satellite images but the method used was not mentioned [38].

The images are provided with ground truth segmentation mask. The mask was used to crop out images of size 56x56 having more than 40% and less than 60% of pixels covered by oil spill for the test set. A total of 1000 images were cropped out. A few examples of oils spill images are in the figure 4.6.

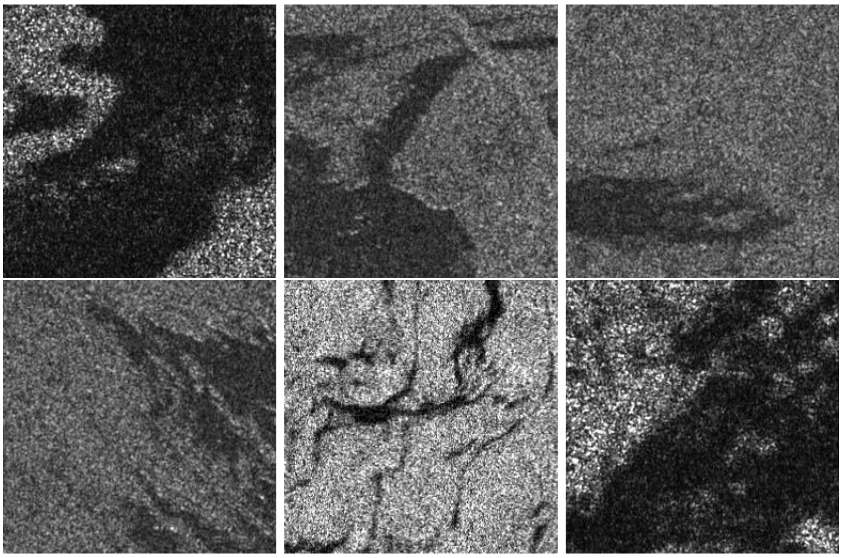

**Figure 4.6:** Example images of SOS dataset containing oil spill images.

An anomalous dataset was created for the final testing of the autoencoder which contained 2000 images from xView3 dataset having no objects in them and were labeled as normal data. 1000 images of ships from xView3 dataset and 1000 images from SOS oil spill were dataset were labeled as anomaly data. All the images were of size 56x56. The results of the model on this dataset are mentioned in section 6.3.

# Chapter 5

# Model Performance Metrics

This chapter will go through the performance metrics used to determine the performance of the model and were used on the test set of the model to get the overview of the performance.

## 5.1 Precision and Recall

The anomaly detection has two classes namely, the normal data and the anomalous data. So, the anomaly detection can be considered as a binary classification problem. The performance metrics used by most researchers in case of classification are Accuracy, Precision, Recall and F1 score [18] [22].

The formulas use the following abbreviation:

- TP stands for True Positive - The model predicts positive and the observation is also positive
- TN stands for True Negative - The model predicts negative and the observation is also negative
- FP stands for False Positive - The model predicts positive but the observation is negative
- FN stands for False Negative - The model predicts negative but the observation is positive

Accuracy is good metric when the data is balanced and since our test data is half normal data and half anomalous data, the accuracy metric is a good fit. Though in real life scenarios the data is very unbalanced and hence other metrics are used. Accuracy in simple terms is "The number of correct predictions divided by total number of predictions" [18]. This is represented by the following formula.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision measures the accuracy of the model in terms of positive predictions, that is, the objects detected by the model. It is the ratio of the true positive to all the positives predicted by the model [22]. It is calculated using the following formula

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

Recall is opposite of precision. It measures in terms of false negatives, that is, the objects that were not detected by the model. It is the ratio of the true positives to all the positives [22]. The following formula is for Recall

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

F1 score combines both the precision and recall and hence can be considered as overall accuracy. A high F1 score means low FP and FN [22]. Following is the formula for F1 score.

$$F1 = 2 * \left( \frac{Precision * Recall}{Precision + Recall} \right)$$

# Chapter 6

# Results and Discussion

This chapter will go through the results of the model on the datasets mentioned in section 4.3 and discuss the results, and the decisions taken throughout the process.

## 6.1 A SAR dataset for ship detection

The first dataset that the model was tested on was for ship detection. The test results, as shown in table 6.1 proved to be unsatisfactory. However, the worst results were with the baseline model which did not contain any Batch Normalisation and Dropout layers. To improve the results further, changes were made to the architecture that included removing and adding one extra convolutional layer. With the said changes, the results obtained were always in the same range. SSIM loss function was also used but there were no improvements in the results.

| Test Scenario | Accuracy | Precision | Recall | F1 score |
|---|---|---|---|---|
| Baseline Model with MSE loss | 49.35 | 49.20 | 19.7 | 28.46 |
| Model with MSE loss | 50.17 | 50.33 | 48.76 | 47.29 |
| Model with SSIM loss | 50.01 | 51.27 | 47.43 | 49.71 |
| Model with Filtered Data | 51.66 | 50.79 | 51.93 | 49.59 |
| Filtered Data with Masked-MSE | 50.50 | 51.30 | 40.30 | 44.31 |

**Table 6.1:** The result of different testing scenarios on the SAR ship detection dataset.

One important observation was that after two epochs, both the training and validation losses would go stable as can be seen in the figure 6.1. To counter this, different values of learning rate were tried and optimizer was changed to Stochastic Gradient Descent (SGD), but this problem of losses being stable after only a few epochs didn't go away, which hinted to the problem of model getting stuck in a local minima and being unable to get out.

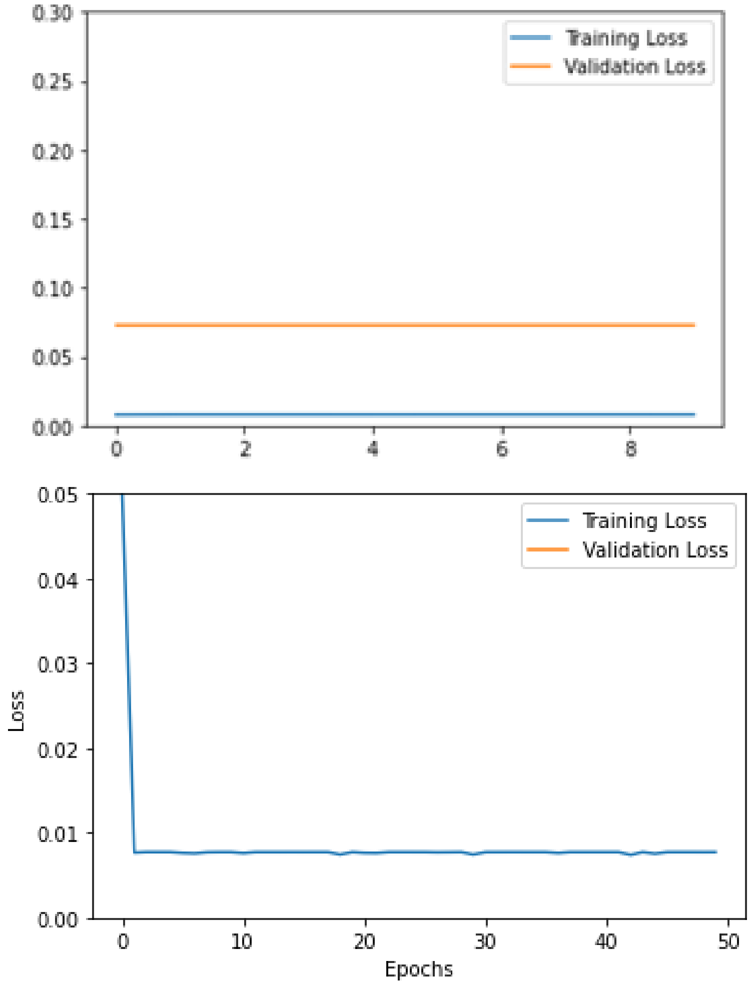


**Figure 6.1:** This figure shows the epochs on x-axis and loss on the y-axis. Top image corresponds to the baseline model with MSE loss. The bottom image corresponds to model with SSIM-loss As can be seen the value of both the losses is stable.

A thorough analysis of the dataset revealed two problems as listed below:

- Many of the images in the training set were all black that is the entire image had a value of zero for each pixel. Many more images had only small patches where the pixel value was higher than zero and rest of the image had pixel value of zero.
- Simple morphology operations to remove the land segments were not very successful and there were a lot of images where there were land and shore-

line, wherein some even had ships near the shoreline.The major concern of all were the images with all pixel values as zero.

Coupling both these things together, that is, the images with zero values and no changes in the losses, it was concluded that the model gets stuck in a local minimum as there is not much to learn from images which does not have any features in it.

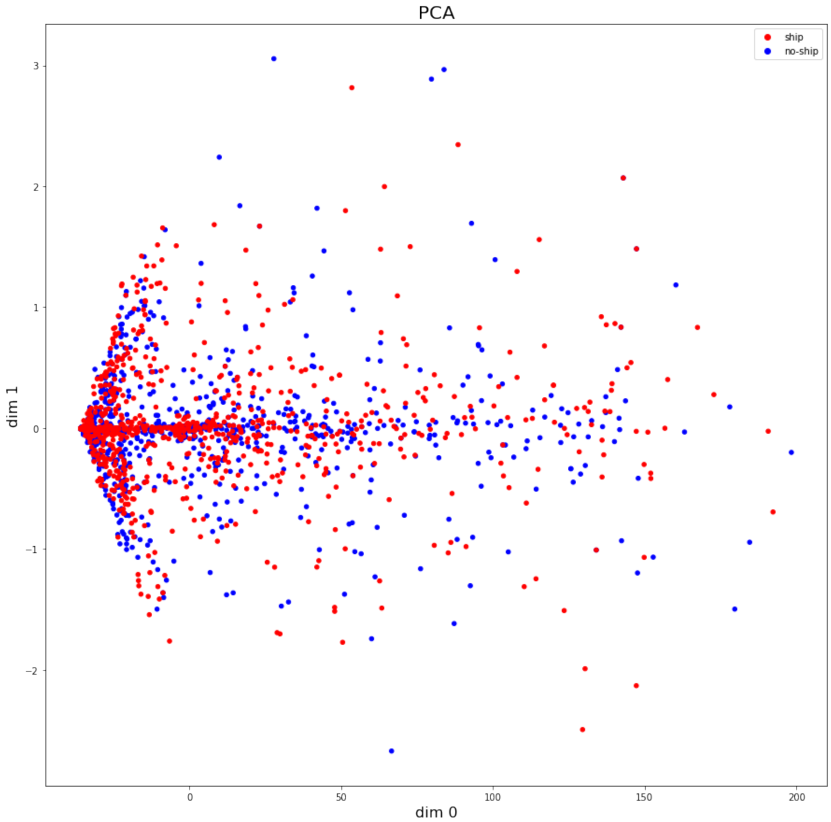


**Figure 6.2:** PCA representation of the latent space on the filtered dataset. As can be seen that there is total overlap of the data points and no separation is achieved.

All the images which had a maximum pixel value less than two were removed from the training dataset and a lot of images which had land were removed manually but not all could be removed. The results after filtering of dataset are mentioned in the table. As can be seen there were no improvements. This was also evident from the PCA representation of the latent space of the model, as seen in figure 6.2 there is no separation between the images having only background and images having ships in them.

To resolve this issue Masked-MSE loss function was used on the filtered data. The results are in the table with test scenario "Filtered Data with Masked-MSE".

The results didn't improve which was evident from the value of the losses and the PCA representation of the latent space, as can seen from the figures 6.3 6.4 that there is no clear separation between the classes and the K-means algorithms clusters them in half giving the nearly 50% accuracy. This could be due to the fact that there were not a lot of image data left after using masked mse loss function and also because the data which was left had either small unlabelled objects.

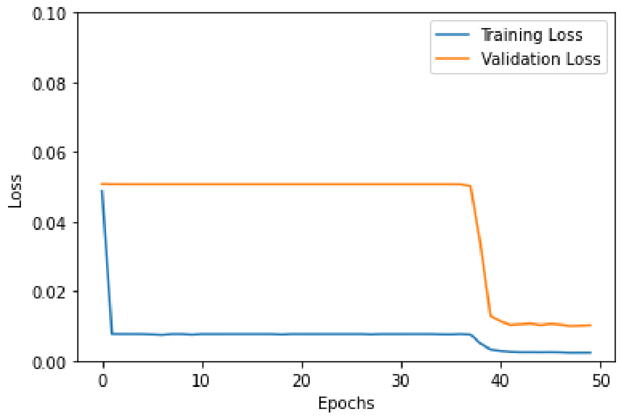


**Figure 6.3:** The figure shows the training and validation loss values using masked-mse loss function.

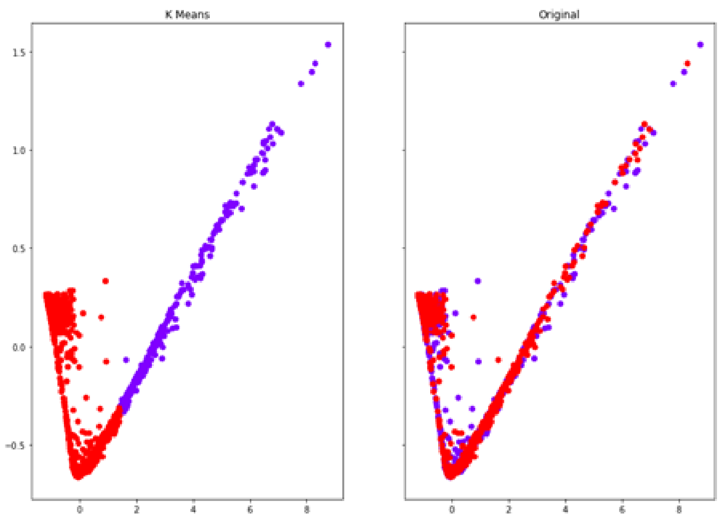


**Figure 6.4:** The left figure shows the generate labels using K-means clustering on the latent space. The right figure shows the PCA representation of the latent space with true labels. The purple dots represent images with no-ship and the red dots represent images with ships

A paper [11] used the same dataset on a variational autoencoder. The code used in the paper is publicly available and so was used on the training dataset, it removed the problem of the non-changing losses but the results were still the same, since the dataset creation was not specified in detail in paper, and the dataset used by the authors for training was also not publicly available. It was concluded that this dataset cannot be used.

## 6.2 xView3 Dataset

The results for the xView3 dataset are in the table 6.2. The results help prove the hypothesis that when an audoencoder is trained only on the open sea image without any landmass or other objects which in this case can be considered as the normal data, then when the trained autoencoder is provided with anomalous data, in this case, images with ships, there is clear separation in the latent space between the normal data and anomalous data.

| Test Scenario | Image Size | Accuracy | Precision | Recall | F1 score |
|---|---|---|---|---|---|
| Baseline model MSE loss | 256 | 89.40 | 89.00 | 89.9 | 89.4 |
| Baseline model SSIM loss | 256 | 89.37 | 89.13 | 89.87 | 89.67 |
| Model with MSE loss | 256 | 93.71 | 92.96 | 93.76 | 93.20 |
| Model with SSIM loss | 256 | 93.16 | 93.97 | 93.34 | 92.50 |
| Model with SSIM loss | 56 | 99.05 | 98.99 | 99.09 | 98.93 |
| **Model with MSE loss** | **56** | **99.11** | **99.18** | **99.30** | **99.07** |

**Table 6.2:** The result of different testing scenarios on the xView3 dataset. Bold numbers represents the final model selected and corresponding values

The final model which has Batch Normalisation and Dropout layers performs better than the baseline model.

When the model is trained with the images of size 256, the PCA representation of the latent space (figure 6.5) shows that is a separation but it is not very clear, as in no two clusters are visible and can be clustered using K-means algorithm (figure 6.6). After looking at the reconstructed images of the ships (figure 6.7) and background, it can be seen that the model more or less reconstructs all the images similar to each other with minor difference which is exactly what can be seen from the representation of the latent space.

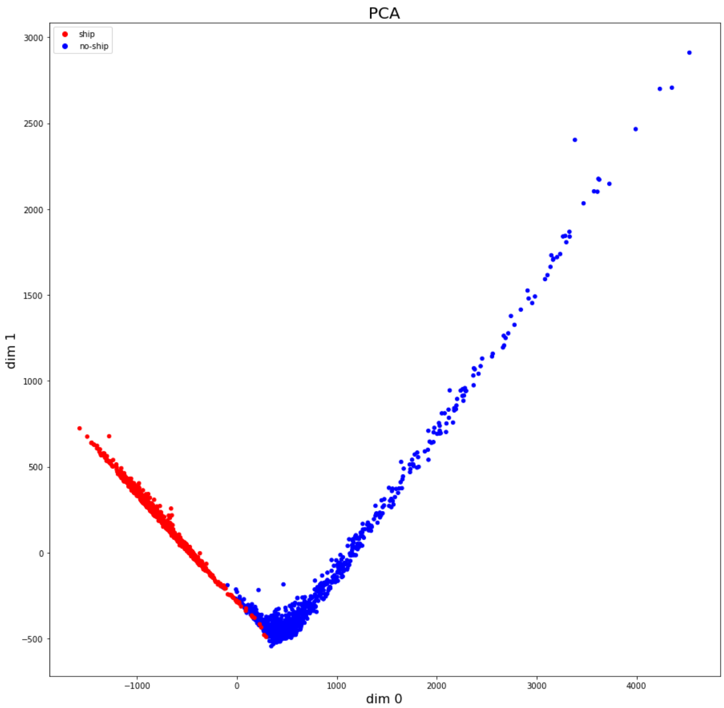


**Figure 6.5:** PCA representation of the latent space with image size 256.

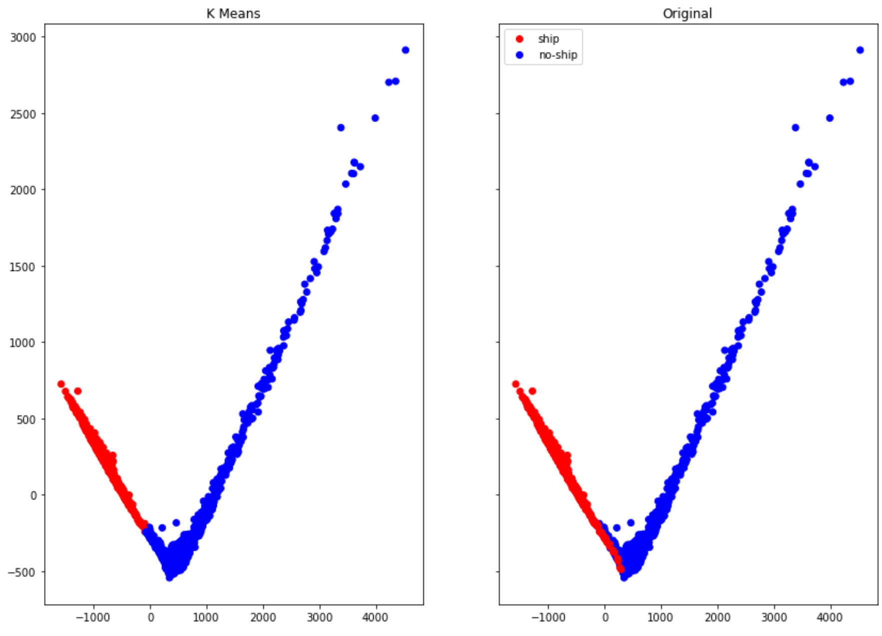


**Figure 6.6:** Right image shows the generate labels through K-means algorithm for images with size 256. Left images shows the true labels on the PCA visualisation of the latent space.

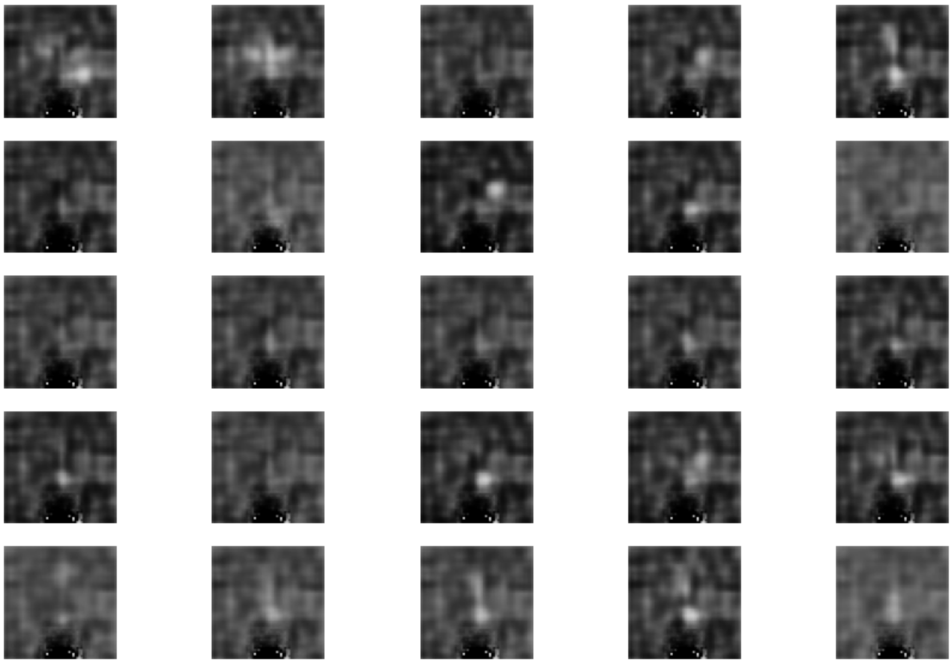

**Figure 6.7:** Reconstructed images from the autoencoder of the images having ships in them for image size 256.

After reducing the image size to 56, the ships in the images now take up majority of the pixels and hence a clear separation was achieved in the latent space which can be seen in the figure 6.8. The separation clearly can be clustered into ship and no-ship classes (figure 6.9). The ship can be clearly seen in the reconstructed images (figure 6.10) and can be clearly differentiate from the reconstructed background images (figure 6.11).

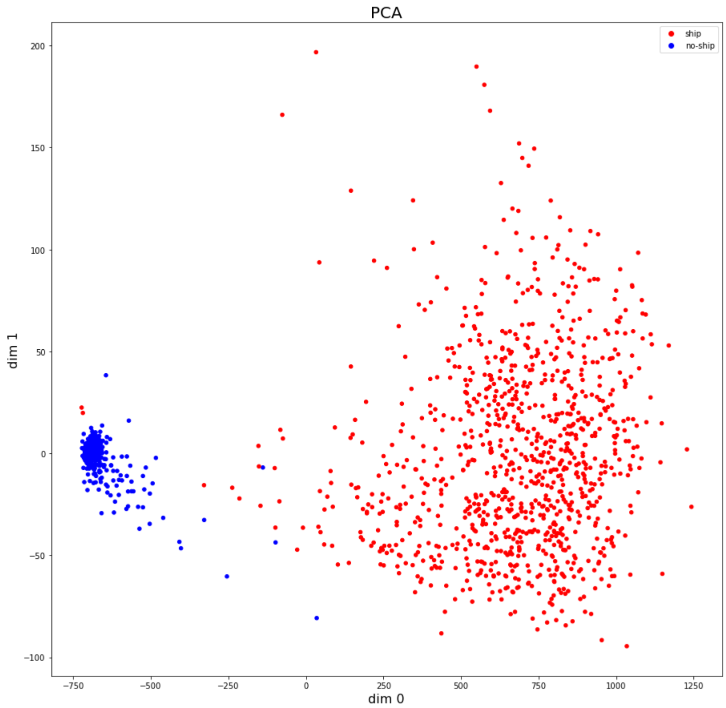


**Figure 6.8:** PCA representation of the latent space with image size 56.

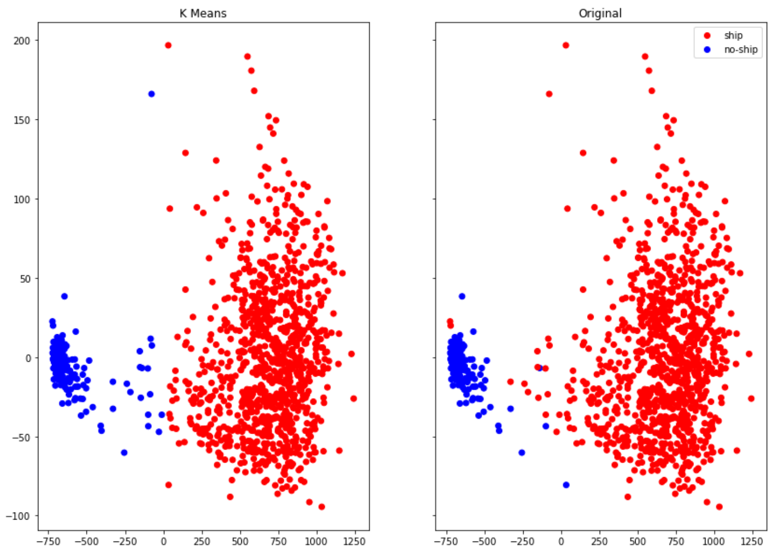


**Figure 6.9:** Right image shows the generate labels through K-means algorithm for images with size 56. Left images shows the true labels on the PCA visualisation of the latent space.

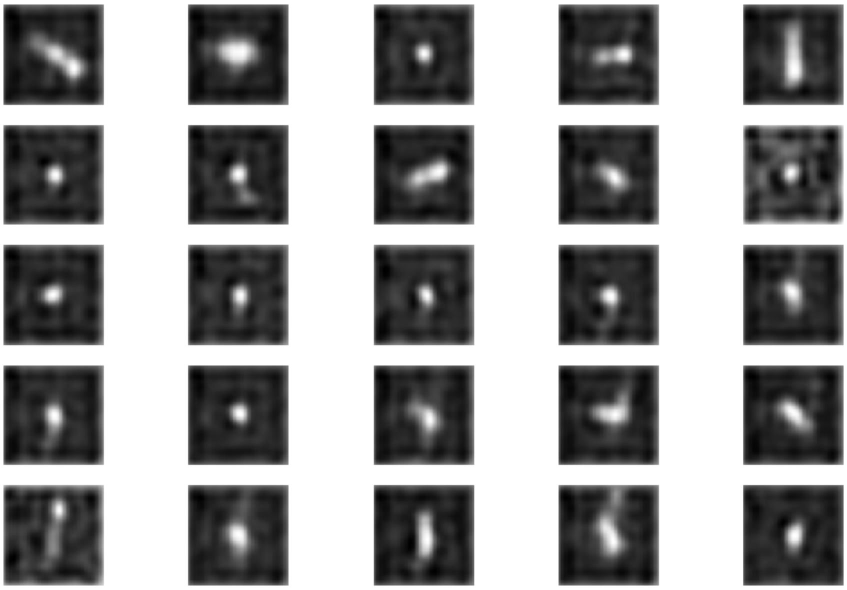

**Figure 6.10:** Reconstructed images from the autoencoder of the images having ships in them for image size 56.

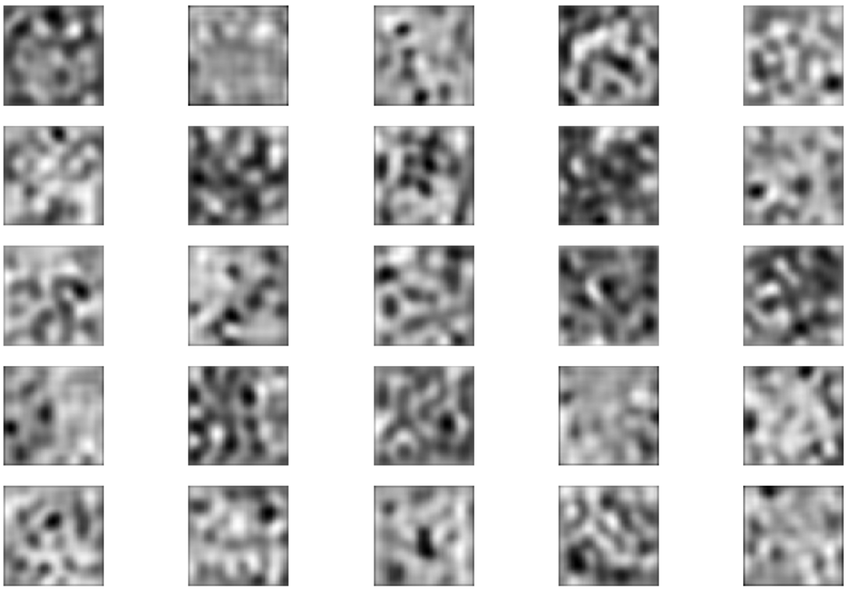

**Figure 6.11:** Reconstructed images from the autoencoder of the images having only background in them for image size 56.

## 6.3 SOS Dataset and Final Results

Here we tabulate the results of testing images containing oil spill on the final model trained for the ship detection and we also merge the ship detection and oil spill data to make a maritime anomaly detection dataset. The results are in table 6.3

| Test Scenario | Accuracy | Precision | Recall | F1 score |
|---|---|---|---|---|
| Oil Spill | 96.10 | 99.78 | 92.40 | 95.95 |
| **Anomaly Detection** | **96.64** | **93.32** | **98.80** | **97.96** |

**Table 6.3:** The result of different testing scenarios on the SOS dataset and the final result when oil spill data and ship detection data is merged to make maritime anomaly detection dataset.

As can be seen from the results the model performs quite well to find the oil spill images and the separation in PCA visualisation of the latent space is good enough for K-means algorithm to cluster them into two categories as can be seen in the figure 6.12. A comparison of oil spill detection is presented in table 6.4. As can be seen from the comparison that our autoencoder based oil spill detection works better on the SOS oil spill dataset.

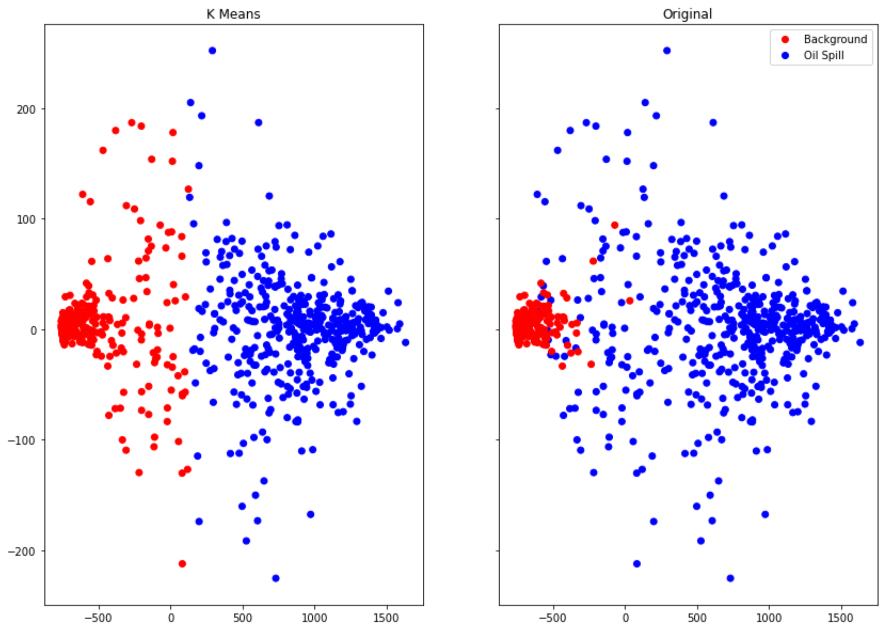


**Figure 6.12:** Right image shows the generate labels through K-means algorithm. Left images shows the true labels on the PCA visualisation of the latent space.

| Method | F1 score | Recall | Precision |
|---|---|---|---|
| U-Net | 86.10 | 81.22 | 85.61 |
| D-LinkNet | 87.08 | 85.22 | 85.22 |
| Deeplabv3 | 87.70 | 84.76 | 88.08 |
| CBD-Net | 87.87 | 87.32 | 91.20 |
| **Ours** | **95.95** | **92.40** | **99.78** |

**Table 6.4:** Comparison with other methods for oil spill detection

The last row of the table 6.3 represents the values of the final testing of the model on the anomaly detection dataset, where the anomalies are ships and oil spill. The model performs quite well on the dataset giving a clear separation in the latent space 6.13.

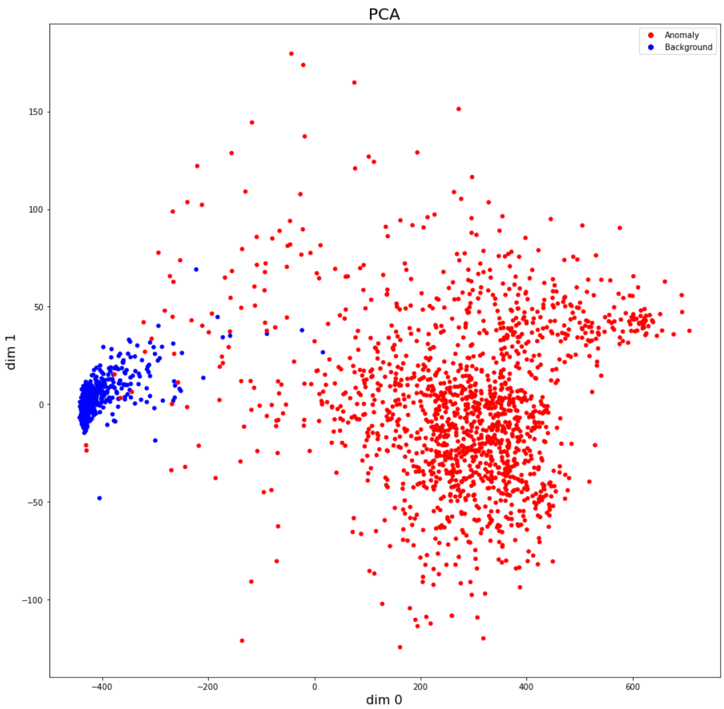


**Figure 6.13:** PCA visualisation of the latent space on maritime anomaly detection dataset.

The future work that can be done is to first make a model to classify the anomalies, since the autoencoder is too sensitive, it would classify even small landmasses as anomalies, and hence for a surveillance system it will be better if the anomalous patch goes through a classification model before warning. Also, the testing of the autoencoder should be done on real-time dataset and if that works perfectly then the model should be linked with the AIS to get information about ships and then the whole system could easily detect ghost ships and AIS spoofing.

# Chapter 7

# Conclusion

The project problem statement highlighted the following:

- intention of creating a maritime surveillance system which can detect ships and ghost ships, and
- to link it with the AIS system to detect detect any illegal fishing or AIS spoofing

In addition to the above stated problems, the system should also be able detect any oil spills, icebergs or any other debris using Satellite Imagery by applying deep learning algorithm.

Based on the results obtained, it can be stated with confidence that a Convolutional Autoencoder can detect ships and oil spills in SAR Satellite Imagery.

The use of an autoencoder was based on the formulation of the above problem as an anomaly detection task, where the ocean without any object in it is the normal data and any object in the ocean is the anomalous data. This hypothesis that if an autoencoder trained to only reconstruct images with ocean gets images with objects in it, then that will result in increased reconstruction loss which can be directly translated to two separable clusters in the latent space of the autoencoder and then using clustering algorithm like K-means, the data points can be clustered into two groups representing the normal data and the anomalous data. This is also proved based on the results, the PCA representation of the latent space and the reconstructed images.

The Convolutional Autoencoder is very sensitive to the data used for training and testing. As seen in the section 6.1 if the training set has any land area of other objects in it, it will affect the training of the autoencoder, and a similar thing can be said about the testing as well. In section 6.2 the testing was done only on images of ships and if any images with landmass or shoreline is provided, then it will be considered as an anomaly. Hence, the trained autoencoder can only be used to detect objects away from the shoreline. This comes to point about the robustness of the model, object detector like YOLO that can be used to detect ships on the shoreline as well, but so far in literature, no object detector was found to detect both ships and oil spills.

The autoencoder proposed in this report cannot be used on images where the size of the object is too small as compared to the image, since the underlying theory is that the anomalous data should produce more reconstruction error than normal data would, so if the size of the object is too small then the reconstruction loss from the small object would not be enough to register a recognisable change. This was also the case in section 6.2.

Overall, the model performed well, giving remarkable accuracy of 96.64% on the anomaly detection dataset, the metrics also show good performance with F1 score being 97.96% and precision and recall having the values as 93.32% and 98.80%, respectively.

# Bibliography


[1] baldassarreFe. "Sparse Means Square Error". In: (2017). URL: https://github.com/keras-team/keras/issues/7065#issuecomment-318002081.

[2] Paul Bergmann et al. "Improving unsupervised defect segmentation by applying structural similarity to autoencoders". In: *arXiv preprint arXiv:1807.02011* (2018).

[3] Changqing Cao et al. "Research on airplane and ship detection of aerial remote sensing images based on convolutional neural network". In: *Sensors* 20.17 (2020), p. 4696.

[4] Liqiong Chen, Wenxuan Shi, and Dexiang Deng. "Improved YOLOv3 based on attention mechanism for fast and accurate ship detection in optical remote sensing images". In: *Remote Sensing* 13.4 (2021), p. 660.

[5] Shiqi Chen et al. "Learning slimming SAR ship object detector through network pruning and knowledge distillation". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2020), pp. 1267–1282.

[6] Maritime Denmark. "Fake Russian warship off Skagen". In: (2021). URL: https://maritimedanmark.dk/?Id=47164.

[7] DIU. "What is remote sensing and what is it used for?" In: (). URL: https://www.usgs.gov/faqs/what-remote-sensing-and-what-it-used#:~:text=Remote%20sensing%20is%20the%20process,sense%22%20things%20about%20the%20Earth..

[8] DIU. "xView3 Competition to detect vessels in SAR images." In: (2021). URL: https://iuu.xview.us/.

[9] EOS. "Satellite Resolution". In: (2020). URL: https://eos.com/blog/satellite-data-what-spatial-resolution-is-enough-for-you/.

[10] Alexander J Farr et al. "Self-Supervised Vessel Detection from Low Resolution Satellite Imagery". In: *AIAA SCITECH 2022 Forum*. 2022, p. 2110.

[11] Nuno Ferreira and Margarida Silveira. "Ship Detection in SAR Images Using Convolutional Variational Autoencoders". In: *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*. IEEE. 2020, pp. 2503–2506.

[12] GISGeography. "Learn Synthetic Aperture Radar (SAR) by Example". In: (2020). URL: `https://gisgeography.com/synthetic-aperture-radar-examples/`.

[13] GISGeography. "Passive vs Active in Remote sensing". In: (2020). URL: `https://gisgeography.com/passive-active-sensors-remote-sensing/`.

[14] Frederik Seerup Hass and Jamal Jokar Arsanjani. "Deep learning for detecting and classifying ocean objects: application of YoloV3 for iceberg–ship discrimination". In: *ISPRS International Journal of Geo-Information* 9.12 (2020), p. 758.

[15] Jeff Heaton. *Ian goodfellow, yoshua bengio, and aaron courville: Deep learning*. 2018.

[16] Xudong Huang et al. "A novel deep learning method for marine oil spill detection from satellite synthetic aperture radar imagery". In: *Marine Pollution Bulletin* 179 (2022), p. 113666.

[17] Zhijian Huang et al. "An intelligent ship image/video detection and classification method with improved regressive deep convolutional neural network". In: *Complexity* 2020 (2020).

[18] Clare Liu. "More Performance Evaluation Metrics for Classification Problems You Should Know". In: (2020). URL: `https://https://www.kdnuggets.com/2020/04/performance-evaluation-metrics-classification.html`.

[19] NASA. "Remote sensing". In: (2020). URL: `https://www.earthdata.nasa.gov/learn/backgrounders/remote-sensing`.

[20] Amir Hosein Oveis et al. "A Survey on the Applications of Convolutional Neural Networks for Synthetic Aperture Radar: Recent Advances". In: *IEEE Aerospace and Electronic Systems Magazine* (2021).

[21] Mark Philip Philipsen. "SAR Data". In: (2022). URL: `https://github.com/markpp/satellite_tools/blob/main/docs/SAR_data.md`.

[22] Mark Sanderson. "Christopher D. Manning, Prabhakar Raghavan, Hinrich Schütze, Introduction to Information Retrieval, Cambridge University Press. 2008. ISBN-13 978-0-521-86571-5, xxi+ 482 pages." In: *Natural Language Engineering* 16.1 (2010), pp. 100–103.

[23] Mohamed Shaban et al. "A deep-learning framework for the detection of oil spills from SAR data". In: *Sensors* 21.7 (2021), p. 2351.

[24] Gang Tang et al. "H-YOLO: A single-shot ship detection approach based on region of interest preselected network". In: *Remote Sensing* 12.24 (2020), p. 4192.

[25] Gang Tang et al. "N-Yolo: A SAR ship detection using noise-classifying and complete-target extraction". In: *Remote Sensing* 13.5 (2021), p. 871.

[26] Ling Tian et al. "Image enhancement driven by object characteristics and dense feature reuse network for ship target detection in remote sensing imagery". In: *Remote Sensing* 13.7 (2021), p. 1327.

[27] VP. "Illegal fishing in Denmark". In: (2021). URL: `https://nyheder.tv2.dk/samfund/2021-09-05-udenlandske-spoegelsesskibe-stoevsuger-danske-farvande-for-fisk-de-taeppebomber`.

[28] Qin Wang et al. "Ship detection based on fused features and rebuilt YOLOv3 networks in optical remote-sensing images". In: *International Journal of Remote Sensing* 42.2 (2021), pp. 520–536.

[29] Yuanyuan Wang et al. "A SAR dataset of ship detection for deep learning under complex backgrounds". In: *remote sensing* 11.7 (2019), p. 765.

[30] Rong Yang et al. "A novel CNN-based detector for ship detection based on rotatable bounding box in SAR images". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 14 (2021), pp. 1938–1958.

[31] Shamsudeen Temitope Yekeen, Abdul-Lateef Balogun, and Khamaruzaman B Wan Yusof. "A novel deep learning instance segmentation model for automated marine oil spill detection". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 167 (2020), pp. 190–200.

[32] Zhenyu Yue et al. "A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition". In: *Cognitive Computation* 13.4 (2021), pp. 795–806.

[33] Kan Zeng and Yixiao Wang. "A deep convolutional neural network for oil spill detection from spaceborne SAR images". In: *Remote Sensing* 12.6 (2020), p. 1015.

[34] Tianwen Zhang et al. "HyperLi-Net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 167 (2020), pp. 123–153.

[35] Juanping Zhao et al. "A coupled convolutional neural network for small and densely clustered ship detection in SAR images". In: *Science China Information Sciences* 62.4 (2019), pp. 1–16.

[36] Yan Zhao et al. “Attention receptive pyramid network for ship detection in SAR images”. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 13 (2020), pp. 2738–2756.

[37] Hongliang Zhu, Weiye Wang, and Rocky Leung. “SAR target classification based on radar image luminance analysis by deep learning”. In: *IEEE Sensors Letters* 4.3 (2020), pp. 1–4.

[38] Qiqi Zhu et al. “Oil Spill Contextual and Boundary-Supervised Detection Network Based on Marine SAR Images”. In: *IEEE Transactions on Geoscience and Remote Sensing* (2021).