TÉCNICO LISBOA

Unsupervised deep learning for ship detection in SAR images

João Gabriel Reis Moura

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. Maria Margarida Campos da Silveira

Examination Committee

Chairperson: Prof. Paulo Jorge Coelho Ramalho Oliveira
Supervisor: Prof. Maria Margarida Campos da Silveira
Member of the Committee: Prof. Pedro Miguel Berardo Duarte Pina

October 2022

To my Dad,
to my Grandmother

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgments

First of all, I would like to thank my supervisor, Professor Margarida Silveira, for all her availability and suggestions. Not only did her supervision and guidance raise the quality of this work, but also made it more enjoyable.

A thank you to Professor João Paulo Costeira and to Hemaxi for their efforts to get the dockers up and running.

To my family, my girlfriend, and my friends, thank you for your unconditional support and for being part of my life. A special thanks to Simão for making these past 5 years infinitely more pleasant and memorable.

Resumo

Da pesca ilegal ao tráfego de drogas, proteção ambiental e prevenção de ameaças, é evidente que a vigilância marítima é de extrema importância. Um dos aspectos fundamentais da vigilância marítima é o conhecimento da localização dos navios. Dado que o oceano cobre uma área tão ampla, são necessários algoritmos automáticos para monitorá-lo. Avanços recentes em aprendizagem profunda têm facilitado substancialmente o desenvolvimento de métodos de deteção de navios para imagens de radar de abertura sintética (SAR). No entanto, a maioria destas soluções são métodos supervisionados de detecção de objetos, que exigem grandes quantidades de dados anotados. Anotar estas imagens é um processo extremamente demorado. De modo a aproveitar a enorme e crescente quantidade de dados SAR, propomos dois métodos de aprendizagem profunda não supervisionada para a segmentação de navios em imagens SAR. O primeiro método é baseado num modelo de tradução de imagem para imagem, o CycleGAN, no qual exploramos as capacidades de transferência de imagens não emparelhadas para aprender o mapeamento do domínio de imagem SAR para o domínio de segmentação. A segunda abordagem, o UDSEP (U-net Detect-Select-Erase-Paste) é um método de segmentação auto-supervisionada, na qual treinamos uma rede de segmentação com dados de um novo algoritmo que gera imagens anotadas sintéticas das imagens originais SAR não anotadas. Experiências no SAR-Ship-Dataset e no SSDD revelam resultados promissores, mas ainda inferiores aos dos métodos supervisionados.

Palavras-chave: aprendizagem profunda, radar de abertura sintética, segmentação semântica de navios, não supervisionado

Abstract

From illegal fishing to drug smuggling, environmental protection, and threat prevention, it is evident that maritime surveillance is of extreme importance. One of the key aspects of maritime surveillance is having knowledge of the location of the ships. Since the ocean covers such a wide area, automatic algorithms are necessary to monitor them. Recent advances in deep learning have substantially facilitated the development of ship detection methods for synthetic aperture radar (SAR) images. However, most of the solutions are supervised object detection methods, which require large amounts of labelled data. Labelling the images is an extremely time-consuming process. To take advantage of the huge and increasing amount of SAR data, we propose two unsupervised deep learning frameworks for SAR ship segmentation. The first framework is based on an image-to-image translation model, the CycleGAN, in which we exploit the model's unpaired image style transfer capabilities to learn the mapping from the SAR image domain to a segmentation domain. The second approach, the UDSEP (U-net Detect-Select-Erase-Paste) is a self-supervised segmentation framework, in which we train a segmentation network with data from a novel algorithm that generates synthetic labelled images from the original SAR unlabelled images. Experiments on the SAR-Ship-Dataset and on SSDD reveal promising results but still inferior to those of the supervised methods.

Keywords: deep learning, synthetic aperture radar, ship semantic segmentation, unsupervised

Contents

List of Tables

List of Figures

Acronyms

2D Two dimensional

3DDM-UNet 3D Dilated multi-scale U-shape CNN

3D Three dimensional

AAM Anchor attention module

AIS Automatic identification system

AP Average precision

BCE Binary Cross Entropy

BFLOPS Billion floating-point operations

BiGAN Bidirectional generative adversarial network

CFAR Constant false alarm rate

CGAN Conditional GAN

CNN-CAM-CRF CNN - Class Activation Mapping - Conditional Random Field

CNN Convolutional neural network

CPU Central Process Unit

DC-GAN Deep convolutional GAN

DSEP Detect-Select-Erase-Paste

FSII Fine Strip-Map 2

FSI Fine Strip-Map 1

GAM Global attention module

GAN Generative Adversarial Network

GF-3 Gaofen-3

GIS Geographic Information System

GPU Graphics Processing Unit

HOG Histogram of oriented gradients

HRSID High resolution SAR images dataset

IUU Illegal, unreported and unregulated

IoU Intersection over Union

LBP Local binary pattern

MHD Modified Hausdorff distance

MR-SSD Multi-resolution SSD

MW-ACGAN Multiscale Wasserstein Auxiliary Classifier GAN

NMS Non-maximum suppression

OSGAN Optical-to-SAR GAN

QPSII Full Polarization 1

QPSI Full Polarization 1

R-CNN Region-CNN

ROI Region of Interest

ReLU Rectified linear unit

SAM Semantic attention module

SAR Synthetic aperture radar

SLS-CNN Sea-Land Segmentation-based CNN

SM S3 Strip-Map

SSDD SAR Ship Detection Dataset

SSD Single Shot MultiBox Detector

SURF Speeded up robust features

SVM Support vector machine

UDSEP U-net Detect-Select-Erase-Paste

UFS Ultrafine Strip-Map

VAE Variational autoencoder

VHF Very High Frequency

YOLO You Only Look Once

mAP Mean average precision

Chapter 1

Introduction

1.1 Motivation

Maritime surveillance has attracted a lot of attention in recent decades, particularly in vessel detection, as knowledge of vessel placements is required to attain complete maritime domain awareness [1]. Illegal exploitation of natural resources, such as illegal fishing, is one of the issues that requires a quick response from naval officials across the nations. In 2020, global estimates suggest that the illegal, unreported and unregulated (IUU) annual fishing accounts for 20% of the world catch [2]. This industry often uses bonded labour, puts food security and regional stability at risk, and threatens marine ecosystems due to overfishing in unpermitted areas [2, 3]. IUU is also linked to a host of other crimes, such as drug and arms trafficking, and wildlife smuggling. According to research conducted by the Global Financial Integrity, 15% of the illicit drug revenue is smuggled by fishing vessels around the world [4]. Illegal immigration is another illicit maritime-based activity that has been receiving much interest. In 2021 alone, over 3000 people died or went missing while attempting to cross the Mediterranean or the Atlantic to reach Europe, which corresponds to a 60% increase compared with the previous year [5]. Consequently, from threat prevention to national security, safety, and environmental protection, it is crucial to provide relevant organizations, governments, and agencies with real-time data on vessel localizations to assist decision-making processes.

To this end, there are several systems available for gathering information on the presence of ships. Typically, these systems can be classified as being cooperative or non-cooperative. In cooperative systems, the ships provide information about themselves. The Automatic Identification System (AIS) is one of the most common cooperative systems, where ships continuously provide information to relevant authorities and to other ships [6]. Despite being very successful at monitoring ships which are legally obligated to install a VHF transponder, AIS fails to detect those who are not and those that disconnect their transponder. Therefore, non-cooperative systems, which acquire information on the localization of the vessels without any collaboration, are of extreme importance. Optical and reflected infrared, hyperspectral, thermal infrared, and radar are the most common types of imaging systems that can provide data for vessel detection [7]. Each of these systems has its set of advantages and disadvantages, however,

in this work the focus will be on satellite-based radars. These systems provide global remote access, being the only viable option to monitor certain sea areas [8]. Synthetic aperture radar (SAR) is the most suitable type of radar for ship detection since its resolution is constant even when far from the observed targets, does not depend on the distance to the target, it can image wide areas at constant resolution, and works regardless of daylight and cloud cover [7, 9].

Several ship detection methods have arisen since the first SAR satellite was launched in 1978 [10]. Most traditional methods are not robust enough for SAR images with different backgrounds, especially in rough sea conditions or near shore, depend on manual tuning of model parameters and have detection speeds incompatible to suit the needs of real-time applications [11]. Recently, with the growth of artificial intelligence, various elegant deep learning solutions have obtained state-of-the-art in the SAR ship detection task. However, most of these solutions are object detection methods, which are supervised and, therefore, require large amounts of labelled data. Labelling the images is a process that requires SAR specialists and is extremely time-consuming and expensive. Unsupervised methods, which do not require the labeling of training images for feature extraction, can be a suitable alternative for ship detection, especially given the extensive and expanding amount of available SAR data.

1.2 Objectives

The goal of this thesis is to develop unsupervised deep learning methods for ship detection in SAR images. Moreover, although the proposed frameworks will only be applied to SAR ship images, they should be generic enough to be applied with relative ease to other datasets. Two distinct novel deep learning frameworks are presented, which were approached as a semantic segmentation problem. The first framework is based on an image-to-image translation model, the CycleGAN, in which we exploit the model's unpaired image style transfer capabilities to learn the mapping from the SAR image domain to the segmentation domain. The second approach, the UDSEP (U-net Detect-Select-Erase-Paste) is a self-supervised semantic segmentation framework, in which we introduce a novel algorithm to generate synthetic labelled images from the original SAR unlabelled images. Then, the generated SAR images and their segmentation masks are used to train a segmentation network, the U-net. The proposed work should contribute to filling the void in the state-of-the-art of SAR ship detection with unsupervised techniques.

1.3 Thesis Outline

This thesis is organised into 6 chapters. In Chapter 2, an overview of the state of the art in SAR ship detection is provided, including a review of both traditional approaches and more recent work utilizing deep learning. Chapter 3 presents the theoretical background for the thesis. A review of SAR imagery and its importance to ship detection is provided, followed by an explanation of the relevant classical and deep learning models for this thesis. Chapter 4 presents the implementation in depth, including a description of the datasets and the preprocessing applied to them, and the details of the two proposed

frameworks. Chapter 5 presents the experimental results, which are then analysed and discussed. Finally, in Chapter 6, the main conclusions and possible future work for this thesis are presented.

Chapter 2

State of the art

An overview of the work done in SAR ship detection will be presented in this chapter. First, a brief summary of traditional approaches is offered for historical background. Then, more recent work adopting deep learning is presented. Furthermore, a particular emphasis is placed on previous published work with the models employed in this thesis.

2.1 Classical Approaches

Long before the development of deep learning-based object detection algorithms, traditional methods were employed to perform ship detection. Usually, traditional methods follow a specific workflow, defined by some or all of three main steps: preprocessing, candidate region extraction, and discrimination [12].

The most significant preprocessing task is normally sea-land segmentation or land masking. This is usually based on GIS (Geographic Information System) or image features, and attempts to mask the land pixels to minimise interference with the next steps. Other tasks of the preprocessing step are very dependent on the posterior steps but may include filtering the speckle noise in the SAR images or normalising the pixel values [13].

The candidate region extraction step consists of an algorithm that searches the whole image for potential ship pixels. One of the first and simplest candidate region extraction algorithms simply sets a global fixed threshold and considers pixels with intensity values above the threshold as candidates. This approach was used by Lin et al. (1997) in [14] and [15]. This method is susceptible to various factors, such as the type of material of each ship, which will have a direct impact on the intensity of its pixels. Therefore, adaptive thresholds, which instead look for unusually bright pixels relative to their surroundings, are more commonly used. The CFAR is the most common adaptive threshold candidate region extraction for SAR ship detection. The CFAR method first statistically models the clutter of the SAR image and then obtains the threshold value according to a set false alarm rate. Wackerman et al. (2001) [16] provided early valuable results for CFAR in ship detection. They used a CFAR detector with a multiple-pixel target window to detect ships in Radarsat-1 SAR images. They experimented with several distributions for the clutter model, such as Gaussian, Exponential, Gamma, and K-distribution.

Throughout the years, numerous CFAR-based algorithms have been proposed for ship detection [17–20]. However, CFAR methods usually only perform well when the scenes are relatively simple. For small ships, inshore or complex offshore scenes, the methods usually underperform, with several false positives. This is directly associated with the difficulties in modelling the background.

Therefore, in an attempt to improve the detection accuracy, several frameworks include a discrimination step. This step analyses the candidate regions and chooses which pixels are ships and which are background. Typically, this is achieved by employing artificially produced features and training traditional classifiers, such as decision trees [21] or support vector machine (SVM) [22]. Several features can be used for the discrimination step, such as the length, width or aspect ratio of the detected regions, but also computer vision introduced features, such as histogram of oriented gradients (HoG) [23], speeded up robust features (SURF) [24], or local binary pattern (LBP) [25].

2.2 Deep learning approaches

In 2012, a significant breakthrough occurred in machine learning when results on regularly used computer vision benchmarks were released for the first time with human-competitiveness [26], outperforming traditional methods. Not before long, as deep learning-based object detection algorithms emerged in computer vision [27], SAR researchers began to look to this field for ideas.

The first public report using deep learning for SAR ship detection was presented by Schwegmann et al. [28] in 2016. They established a very deep High-Way CNN to accomplish SAR ship discrimination, achieving promising results. Inspired by R-CNN, proposed by Girshick et al. [29], Liu et al. (2017) [30] proposed a framework of Sea-Land Segmentation-based Convolutional Neural Network (SLS-CNN) for ship detection. Although both the mentioned reports are based on deep learning, they only apply it to the last steps of the traditional detection workflow. Thus, deep learning is only employed for the ship-background binary classification task, with several other traditional tasks still performed, such as sea-land segmentation. Later that year, the Faster R-CNN [31] was first applied to SAR ship detection by Kang et al. (2017) [32]. Considering that the model often misses small ships due to information loss from max-pooling operations, the authors proposed transferring detections with a score between 0.3 and 0.8 into a CFAR detector. Kang et al. (2017) [33] also proposed a new method to improve the Faster R-CNN with the CFAR detector, implementing a contextual region-based convolutional neural network with multilayer fusion. The network consists of a region proposal network (RPN) with high network resolution and an object detection network with contextual features. Compared to the previous work, the results revealed considerable improvements. This was also the first report to achieve full end-to-end training and testing, without the need for traditional sea-land segmentation or any post-processing. Thus, the work of Kang et al. decidedly served as a baseline and motivation for future deep learning research in SAR ship detection.

However, at that time, there were no proper open datasets dedicated to SAR ship detection based on deep learning. Hence, in December 2017, the SSDD (SAR Ship Detection Dataset) was made publicly available. This dataset provided the same data and evaluation criteria for researchers, which established

the framework for the field's quick development and endorsed a new era in SAR ship detection, marking the transition from sporadic to regular deep learning papers. As years passed, several other SAR ship datasets were made available, and the number of articles published on the topic increased considerably. According to [12], in May 2022 there were 177 published papers that used deep learning to detect ships in SAR images.

2.2.1 Object detection methods

The most common approaches for SAR ship detection are based on state-of-the-art object detection methods. Li et al. (2017) [34] first tested the SSDD dataset with an improved Faster R-CNN. They adopted the ZF-Net [35] pre-trained on ImageNet and fine-tuned the model to the SSDD dataset. Furthermore, given the difficulties for the Faster R-CNN in detecting ships of different sizes, the feature maps from the convolutional layer 3 to layer 5 were fused. This prevented the omission of important features due to the dominance of the features from the latter layers, resulting in a more robust model with improved accuracy for ships of different sizes. Also based on Faster R-CNN, Lin et al. (2019) [36] used an encoding scale vector encouraged by a squeeze and excitation mechanism to suppress redundant subfeature maps after Region of Interest (ROI) pooling. Experimental results on Sentinel-1 images revealed an increase of 9.7% on the F1-score and a 14% faster execution time when compared to the, at the time, state-of-the-art.

The YOLO (You Only Look Once), proposed by Redmon et al. [37] in 2016, is a state-of-the-art architecture for object detection. YOLO solves the object detection task as a regression problem and outputs the spatially separated bounding boxes and their corresponding class probabilities. Unlike R-CNN, this is done with a single neural network from one evaluation. The original YOLO algorithm has several variations, such as YOLOv2 [38], YOLOv3 [39], YOLOv4 [40], and YOLOv5 [41], that have been extensively tested for SAR ship detection. Motivated by the practical Graphics Processing Units (GPU) limitations of frontline marine monitoring, the majority of the following YOLO-based papers focus on lightweight backbone network improvements. Deng et al. (2019) [42] and Chang et al. (2019) [43] adopted YOLOv2 to detect ships in SAR images. Chang et al. proposed a modified architecture, the YOLOv2-reduced that has fewer layers. The YOLOv2-reduced maintained a similar AP (average precision) with 2.5 faster detection times. Inspired by the latest YOLO method at the time, YOLOv3, Zhang et al. (2019) [44] proposed a model with depthwise convolution and a pointwise convolution to replace the traditional convolution neural network. This allowed to decrease considerably the number of network parameters and the detection time. Similarly, Zhou et al. (2020) [45] proposed a lightweight convolutional neural network, the LiraNet. This model uses residual and dense connections and group convolution, including stem blocks and extractor modules. Jiang et al. (2021) [46] adopted YOLOv4 to implement YOLOv4-light, a reduced model with consequently fewer computational parameters, memory consumption, and detection time. To compensate for the accuracy loss due to the reduced model, three-channel images were used. Liu et al. (2022) [47] also proposed a lightweight ship detection network. Based on the YOLOv4-LITE model [48], which uses MobileNetv2 [49] as its backbone, the model implemented an

improved receptive field block to improve the quality of multi-scale ship detections.

The SSD (Single Shot MultiBox Detector) [50] is another state-of-the-art object detection algorithm that has been used for SAR ship detection. This approach combines the regression concept with the anchor box mechanism, which is very similar to the anchor boxes used in Faster R-CNN but is applied to multiple feature maps with different resolutions. Compared to other detection models, the SSD is usually simpler, as it eliminates proposal generation and subsequent pixel or feature resampling stages, condensing all computation into one network. In [51, 52] (2017) Wang et al. applied the original SSD to ship detection for both Sentinel-1 and Gaofen-3 images. In both works, the authors used two SSD models: SSD-300 and SSD-512, which have input sizes of 300 and 512 pixels in height and width, respectively. The models were built with a VGG16 network that was pretrained on the PASCAL VOC dataset. Moreover, Ma et al. (2018) [53] proposed a Single Shot Multi-box Detector with a multi-resolution input (MR-SSD) to classify different types of marine targets, such as boats, towers, cargo ships, etc. Compared to the original SSD, this model is able to extract more features at various resolution. More recently, Jin et al. (2021) [54] improved SSD by adding a feature fusing module to shallow feature layers and introducing an attention mechanism based on squeeze excitation modules. Compared to the traditional SSD, the proposed model was able to improve ship detection accuracy while maintaining detection speed.

All of the above discussed object detection methods are supervised. Unsupervised work, which can be a suitable alternative for ship detection has not been extensively explored. Ferreira et al. [55] proposed an unsupervised framework for SAR ship detection based on anomaly detection. They start by learning the data representations with a convolutional variational autoencoder (VAE) and then perform anomaly detection based on those representations with a clustering algorithm. Dias [56] also proposed an unsupervised anomaly detection framework for ship detection. They train a bidirectional generative adversarial network (BiGAN) with non-ship images and then use its inability to reconstruct images with ships to detect anomalies. In fact, although referred to as unsupervised, both the mentioned works rely on a supervised preselection of non-ship ocean images to train the models. Therefore, to our best knowledge, no fully unsupervised work has been developed and published for SAR ship detection.

2.2.2 Segmentation methods

In 2020, Wei et al. publicly released the High resolution SAR images dataset (HRSID) [57], which, unlike earlier published datasets, provided polygon masks of the ships. Soon after that, in September 2021, Zhang et al. [11] released an improved version of the SSDD dataset where the ships were relabeled with their polygon segmentation. Consequently, the publication of these datasets aided and encouraged the development of segmentation techniques, which were previously very challenging to develop. Instead of providing bounding boxes, these techniques aim to provide pixel-level contour information for the detected ships.

For instance, Gao et al. (2021) [58] proposed a lightweight feature extractor and an anchor-free convolutional network for SAR ship segmentation based on the Ghostnet [59]. They presented a dynamic encoder–decoder to fully disseminate feature information by transforming shared features into

task-specific features. Furthermore, utilising the geometrical shape and positional relationship between the ships, a unique loss function based on centroid distance was implemented. Moreover, Zhao et al. (2021) [60] proposed an instance segmentation framework based on a synergistic attention mechanism at the image, semantic, and target level. For feature extraction, feature fusion, and target location, the global attention module (GAM), semantic attention module (SAM), and anchor attention module (AAM) were developed, respectively.

Some authors used U-net, a deep learning semantic segmentation model introduced by Ronneberger et al. in 2015 [61], for SAR ship segmentation. For instance, Li et al. (2020) [62] proposed a 3D dilated multi-scale U-shape convolutional neural network (3DDM-UNet). They start by building a 3D image block through a multiscale stationary wavelet transform and, then, feed it to the 3D U-net. The results obtained outperformed the other compared segmentation methods, such as the original U-net. Furthermore, Mao et al. (2020) [63] proposed an efficient, low-cost SAR ship detection network consisting of two branches: a ship bounding box regression network and a score map regression network. Both branches use a simplified U-net to extract features. Afterwards, the soft non-maximum suppression (NMS) post-processing module is exploited to get the final detections. Although the ultimate goal of this strategy is to achieve ship detection, good segmentation results were obtained as an intermediate step in the score map regression network.

Although fully unsupervised deep learning work has not been explored for SAR ship segmentation, some authors have adopted weakly supervised methods. For instance, Gu et al. (2020) [64] proposed a three-component CNN-CAM-CRF (CNN - Class Activation Mapping - Conditional Random Field) model with two global labels, ship and nonship, that not only outputs the ship location heatmap and bounding box, but also the pixel-level segmentation. Wang et al. (2020) [65] built a weakly supervised deep hierarchical convolutional network for SAR ship segmentation that only uses noisy and missing target level annotations. First, they trained a robust ROI detection network with soft labels to account for the uncertainty of annotations with an added regularisation term to the cost function about the expected existence of ships. Regularisation is implemented to prevent the network from becoming highly unstable and oscillating throughout training, hence reducing the false alarm rate. Then, they created a statistical model, Gaussian- $\mathcal{G}_A^0$ mixture distribution where the Gaussian represents the properties of the ships and the $\mathcal{G}_A^0$ represents the sea clutter, both in the SAR ROI data. Lastly, a variational autoencoder (VAE) is trained to estimate the parameters of the mixture model, and Otsu's threshold [66] is used to segment the ships on the parameter maps.

2.2.3 Data augmentation

Some authors attempted to solve the difficulties of collecting and labelling SAR ship images using data augmentation techniques. Traditional techniques such as flipping, cropping, and affine transformation have been widely employed [67, 68]. However, these strategies have proven insufficient to fully capture the wide variety of ships and their complex backgrounds [69].

Therefore, authors have proposed to overcome this problem by exploring Generative Adversarial

Networks, mostly known as GANs, which are deep learning based generative models introduced by Ian Goodfellow et al. in 2014 [70]. For instance, motivated by the low detection accuracy for a small dataset, Zou et al. (2020) [71] used a multi-scale Wasserstein auxiliary classifier GAN [72] as a data augmentation technique to generate high-resolution SAR ship images. Moreover, Zhang et al. (2022) [73] attempted to solve the lower ship detection accuracy for the inshore scene when compared to the offshore scene. Thus, they developed a GAN to extract the scene features of SAR images and used K-means to create a scene binary cluster. Then, the inshore scene images were augmented via replication, rotation transformation, or noise addition until balance was obtained with the offshore scene. More recently, some authors attempted to solve the SAR labelling difficulties by performing unsupervised domain adaptation, which aims to transfer knowledge from a labelled source domain to a target unlabelled domain. For instance, Shi et al. (2022) [74] developed a framework for SAR ship detection through an unsupervised domain adaptation by transferring optical domain knowledge to the SAR domain. Based on the architecture of the original CycleGAN [75], an unpaired image-to-image translation model, the authors proposed a cycle-consistent GAN with skip connections on the generator that revealed valid results translating from the optical to the SAR domain. Therefore, the method developed is able to receive as input images from the optical domain, which are easier and less time-consuming to label, and generate labelled images from the SAR domain. Similarly, Kwon et al. [69] (2022) proposed training a conditional generative adversarial network (cGAN) to generate SAR ship images from electro-optical images. Once again, since the model is trained using unpaired images, a cycle-consistency loss is imposed to maintain structural information while translating the image's features. In summary, GANs have been widely explored within the ship detection task, but solely as an augmentation technique. To the best of our knowledge, GANs have not been directly applied to ship detection or segmentation.

Tables 2.1, 2.2, and 2.3 present the summary of the characteristics and main results of the deep learning methods described in this section.

Table 2.1: State-of-the-art on deep Learning SAR ship object detection methods. Q_F denotes the quality factor, P_d the detection probability, P_f the false alarm probability, and Acc the accuracy.

Author	**Year**	**Dataset**	**Main architecture**	**Main results**
Schwegmann et al. [28]	2016	22 Sentinel-1 and 3 RADARSAT-2 images	High-Way CNN	$Acc = 96.67\%$
Liu et al. [30]	2017	ALOS PALSAR and TerraSAR-X imagery	SLS-CNN	$Q_F > 80\%$ $P_d > 80\%$
Kang et al. [32]	2017	23 Sentinel-1 images	Faster R-CNN and CFAR	$P_d = 78.6\%$ $P_f = 24.2\%$
Kang et al. [33]	2017	27 Sentinel-1 images	Contextual Region-Based CNN with Multilayer Fusion	$P_d = 88.35\%$ $P_f = 13.72\%$ $F1$ = 0.873 Time = 2.180 s
Li et al. [34]	2017	SSDD	Improved Faster R-CNN	$AP = 78.8\%$ Time = 173 ms
Lin et al. [36]	2019	22 Sentinel-1 images	Squeeze and Excitation Rank Faster R-CNN	$P_d = 81.1\%$ $P_f = 13.8\%$ $F1 = 83.6\%$
Deng et al. [42]	2019	80 Sentinel-1 images	YOLOv2 with Densenet as backbone	$F1 = 75\%$, Time = 3.17 s
Chang et al. [43]	2019	SSDD and DSSDD	YOLOv2-reduced	SSDD: $Acc = 90.05\%$ Time = 25 ms, DSSDD: $Acc = 89.13\%$ Time = 27 ms
Zhang et al. [44]	2019	SSDD	Depthwise Separable CNN	$AP = 94.13\%$ Time = 9.03 ms
Zhou et al. [45]	2020	mini-RD and SSDD	Lightweight YOLO	BFLOPS: 2.980 mini-RD: $AP = 83.21\%$ SSDD: $AP = 85.46\%$
Jiang et al. [46]	2021	SSDD	Lightweight YOLO-V4 with three-channels	$AP = 90.37\%$ Time = 13.42 ms
Liu et al. [47]	2022	SSDD	YOLOv4-LITE	$AP = 95.03\%$ Time = 21.2 ms

Continued from previous page.

Author	Year	Dataset	Main architecture	Main results
Wang et al. [51]	2017	3 Gaofen-3 images	SSD300 and SSD512 with transfer learning	SSD300: $P_d = 97.87\%$ $P_f = 10.75\%$, SSD512: $P_d = 100\%$ $P_f = 6.93\%$
Wang et al. [52]	2017	3 Sentinel-1 images	SSD300 and SSD512 with transfer learning	SSD300: $F1 = 92.11\%$ SSD512: $F1 = 94.44\%$
Ma et al. [53]	2018	MTCD and MTDD	MR-SSD	$F1 = 94.57\%$ $mAP = 87.38\%$
Jin et al. [54]	2021	SSDD	SSD with feature fusing module and squeeze excitation modules	$AP = 94.41\%$ Time = 32.6 ms

Table 2.2: State-of-the-art on deep learning SAR ship segmentation methods.

Author	Year	Dataset	Main architecture	Main results
Gao et al. [58]	2021	SSDD	Lightweight Ghost Net	Offshore $AP = 64.8\%$ Inshore $AP = 46.0\%$
Zhao et al. [60]	2021	HRSID and SSDD	Synergistic Attention R-CNN	HRSID: Detection $AP = 68.7\%$ Segmentation $AP = 56.5\%$, SSDD: Detection $AP = 91.5\%$
Li et al. [62]	2020	TanDEM-X imagery	3D Dilated Multiscale U-Net	Sensitivity = 0.9004 ± 0.0295 Specificity = 0.9996 ± 0.0001 Dice = 0.9386 ± 0.0168 MHD = 0.1110 ± 0.0234
Mao et al. [63]	2020	SSDD	Simplified U-Net	$AP = 68.1\%$
Gu et al. [64]	2020	Chinese Gaofen-3 fine strip imagery	CNN-CAM-CRF	-
Wang et al. [65]	2020	6 Gaofen-3 images	YOLOv3	$F1 = 86.22\%$ $AP = 81.78\%$

Table 2.3: State-of-the-art on deep learning SAR ship data augmentation methods.

Author	**Year**	**Dataset**	**Main architecture**	**Main results**
Zou et al. [71]	2020	Gaofen-3 imagery	MW-ACGAN	Generated multi-scale and multi-class ship slices
Zhang et al. [73]	2022	SSDD	GAN	Generated inshore images to eliminate scene learning bias
Shi et al. [74]	2022	AIR-SARship-1.0 and GF2ship	OSGAN	Translated labelled images from optical domain to SAR domain
Kwon et al. [69]	2022	HRSID	cGAN	Translated labelled images from electro-optical domain to SAR domain

Chapter 3

Theoretical Background

A theoretical background is presented in this chapter. First, a brief overview of SAR and its utility for ship detection is given. Afterwards, a thorough explanation of the pertinent classical and deep learning models utilised in this thesis is provided.

3.1 Synthetic Aperture Radar Imaging

The main goal of this thesis is to develop unsupervised deep learning models to detect ships in SAR images. Therefore, it is relevant to start by introducing SAR and its relevance to ship detection.

For over 30 years, SAR has been extensively used by the scientific community for Earth remote sensing. SAR produces high-resolution images, independently of weather and daylight, that can be utilised in numerous applications, such as climate change and geoscience research, 2D and 3D mapping, change detection, and security-related monitoring [9, 76, 77]. SAR systems are based on a pulsed radar placed on a platform that moves forward and features a side-looking imaging geometry. The radar technology sends out powerful electromagnetic pulses and sequentially gathers the echoes of the backscattered signal. After acquiring the data, the radar transmits it to another antenna on earth [78]. In its simplest form, the system provides a 2D reflectivity map of the imaged area. In general, smooth surfaces appear as black areas due to single-bounce surface scattering. Rough surfaces appear brighter due to the fact that they reflect light in all directions, thereby, scattering more energy back to the antenna [79].

Applied to ship detection, the backscatter from the ocean is usually due to single-bounce surface scattering, which leads to black areas in the reflectivity map. If the water is turbulent or the incidence angle is high, this could not be true. Ships, on the other hand, typically appear as bright pixels due to double and multiple-bounce scattering. Figure 3.1 shows the scattering mechanism for calm sea conditions. Additionally, the appearance of the image may vary considerably with the radar parameters, such as frequency, resolution, incidence angle, and polarisation [13]. For instance, the shape of the same ship differs significantly at various resolutions, and ships of different shapes have varying sizes at the same resolution [7]. Furthermore, the ships have various complex surroundings due to their special imaging mechanisms that consequently interfere with their detection. These interferences can be blurred

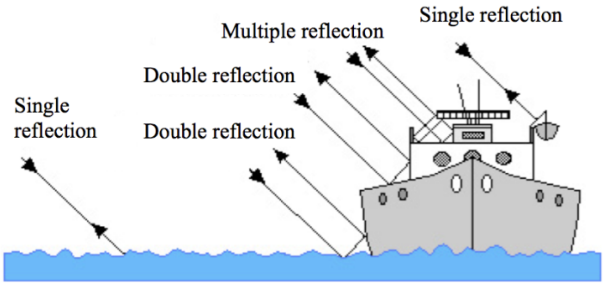


Figure 3.1: Scattering mechanisms from the sea surface and a ship in calm sea conditions (retrieved from [80]).

edges, sidelobes, ship wakes, speckle noise, inshore scenes, etc. Figure 3.2 shows some examples of the common types of interference.

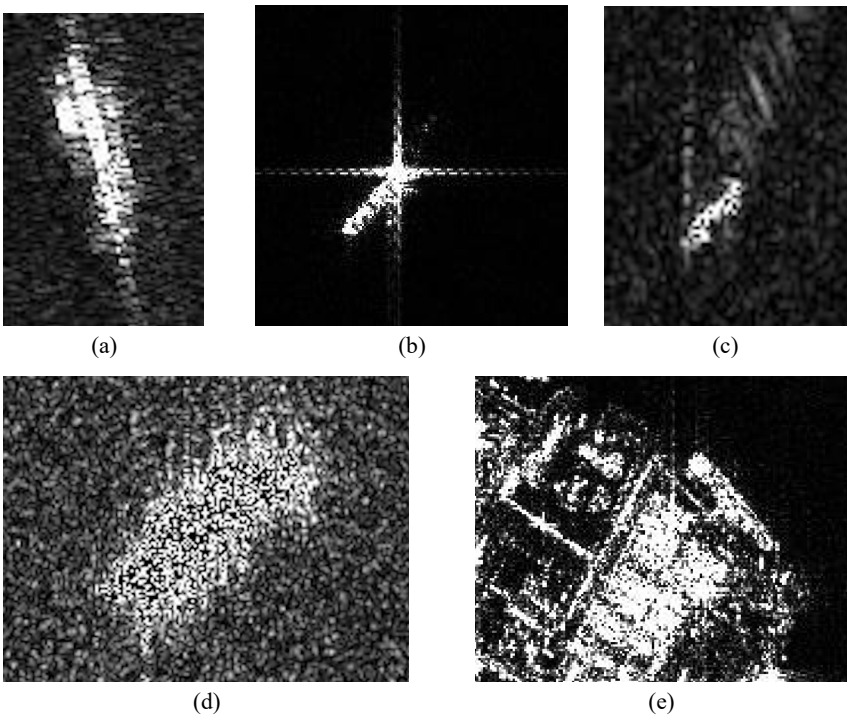


Figure 3.2: SAR ships and their background interference: blurred edges (a), sidelobes (b), ship wake (c), speckle noise (d), and inshore scenes (e).

3.2 Classical models for ship detection

In this section, the traditional methods that have been used for ship detection and are used for comparison purposes in this thesis are introduced. Therefore, the CFAR method and the spectral residual approach for saliency detection are presented.

3.2.1 CFAR

The CFAR method is an adaptive threshold-based detection algorithm that has been widely used for SAR ship detection. The fundamental concept underlying the CFAR detector is to determine the detection threshold, T, based on a false alarm probability, P_{fa}, and the clutter probability density function of the SAR image. Let I be the grey value of the pixel and $p_{\mathrm{b}}(I)$ the probability density function of the background. The probability of false alarm and threshold value are given by

$$P_{\mathrm{fa}} = \int_{T}^{\infty} p_{\mathrm{b}}(I)\mathrm{d}I. \tag{3.1}$$

For a set P_{fa} and p_{b}, we can solve (3.1) to obtain the T value. All pixels with values higher than the threshold are defined as ship, and pixels with values lower than the threshold are considered as background.

The algorithm generally consists of establishing the background distribution of the clutter based on the SAR image and then estimating the distribution parameters of the clutter pixels in the sliding window. Then, given a specified false alarm probability, the threshold is computed and compared with the test pixel to obtain the detection result. This process is repeated for each pixel of the input SAR image through a sliding window. Typically, the sliding window, depicted in Figure 3.3, is composed of three windows: the clutter window, the target window, and the protection window. The clutter and the protection window are centred in the test pixel. The clutter window is used for background clutter statistics to compute the target detection threshold, and the protection window is used to ensure that no pixel of the target is included in the background.

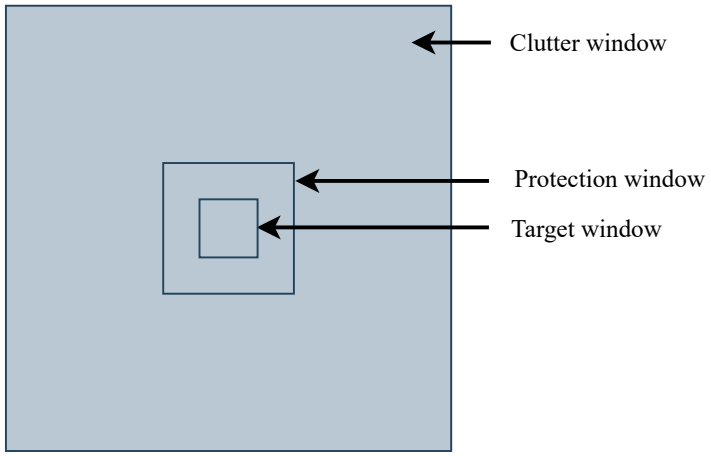


Figure 3.3: CFAR sliding window.

To deal with different sea conditions, several clutter statistical models and sliding window structures have been proposed throughout the years. Currently, the Rayleigh distribution, the Weibull distribution, the Pearson distribution, the Gamma distribution, and the G^0 distribution are the most used models [81]. However, although still some research is done on this topic, CFAR methods continue to significantly rely on sea conditions and model parameters and are not robust enough to handle multitargets and nonhomogeneous backgrounds.

3.2.2 Saliency

The human visual system functions as a filter that is able to focus greater attention on visually appealing areas or objects [82]. Furthermore, it is believed that the human saliency detection process is divided into two stages [83]. The first stage is a simple, fast, pre-attentive process in which low-level features such as edges, intensity, and orientation initially emerge. It is at this stage where the candidates to object surge. The second stage of attention is a slower and more complex process in which additional details are extracted from objects. The first stage was the inspiration for Xiaodi Hou and Liqing Zhang to introduce a technique to extract objects from their background, the spectral residual approach for saliency detection [83].

Starting from the principle of image scale invariance [84], which states that the amplitude $\mathcal{A}(f)$ of the averaged Fourier spectrum of natural images follows a $1/f$ distribution, the authors adopted a log spectrum representation of the image, $\mathcal{L}(f) = \log(\mathcal{A}(f))$. Furthermore, they believed that the information that jumps out of the smooth log spectrum curve is related to the location of the objects or areas of interest. To this end, the authors proposed to approximate the shape of $\mathcal{A}(f)$ by convoluting the input image with an average filter:

$$\mathcal{A}(f) = h_n(f) * \mathcal{L}(f), \tag{3.2}$$

where $h_n(f)$ is an $n \times n$ matrix given by

$$h_n(f) = \frac{1}{n^2} \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}. \tag{3.3}$$

Then, in order to obtain the information that stands out from the log spectrum curve, they defined $R(f)$ as the spectral residual,

$$\mathcal{R}(f) = \mathcal{L}(f) - \mathcal{A}(f). \tag{3.4}$$

The spectral residual represents the statistical singularities in the input image and can be understood as the unexpected or anomalous portion of the image.

Furthermore, with the inverse Fourier transform, it is possible to reconstruct the output image in the spatial domain, obtaining the saliency map. The complete procedure for obtaining the saliency map, $\mathcal{S}(x)$, from an input image, $\mathcal{I}(x)$, is given by

$$\mathcal{A}(f) = \Re(\mathfrak{F}[\mathcal{I}(x)]) \tag{3.5}$$

$$\mathcal{P}(f) = \Im(\mathfrak{F}[\mathcal{I}(x)]) \tag{3.6}$$

$$\mathcal{L}(f) = \log(\mathcal{A}(f)) \tag{3.7}$$

$$\mathcal{R}(f) = \mathcal{L}(f) - h_n(f) * \mathcal{L}(f) \tag{3.8}$$

$$\mathcal{S}(x) = g(x) * \mathfrak{F}^{-1}[\exp(\mathcal{R}(f) + \mathcal{P}(f))]^2, \tag{3.9}$$

where $\mathfrak{F}$ and $\mathfrak{F}^{-1}$ are the Fourier Transform and the Inverse Fourier Transform, $\mathcal{P}(f)$ the phase spectrum of the input image, and $g(x)$ a Gaussian filter used to obtain a smoother saliency map.

Afterwards, a thresholding technique, such as Otsu's threshold [66] can be applied to the saliency map in order to obtain a binary map with the highlighted salient objects.

Even though this method can be directly applied to SAR ship images, the results lack consistency, with good results for simple images and terrible results for images with complex backgrounds. Although unreliable for ship detection, this method can be very useful for other preprocessing tasks, as will be further discussed in Chapter 4.

3.3 Deep Learning models

In this section, the relevant deep learning models for this thesis will be introduced. The fundamentals of deep learning, such as deep feedforward neural networks, convolutional neural networks, backpropagation, etc., are omitted. If the reader is unfamiliar with this topic, we suggest [85] and [86]. Only the theory behind U-net, GAN, and CycleGAN will be presented.

3.3.1 U-net

Inspired by the work of Long et al. [87] using fully convolutional networks, Ronneberger et al. proposed the U-net [61]. Originally designed for biomedical segmentation, the architecture achieved state-of-the-art segmentation results, winning the Cell Tracking Challenge in 2015 by a large margin [61].

The basic architecture of the U-net, depicted in Figure 3.4, is composed of two paths: a contracting path and an expansion path. Also known as the encoder, the contracting path follows a typical convolutional network architecture, consisting of repeated convolutions followed by rectified linear unit (ReLU) activations and max-pooling, that allows for high-level feature extraction. Throughout the contracting path, the number of feature channels increases while the image size decreases. The expansion path, or decoder, consists of up-convolutions followed by convolutions and ReLU, and concatenations with features that have been captured in the encoder. Due to convolution, there is a loss of border pixels, therefore, cropping is necessary. Thus, the pixel features near the edges are removed, since they have the least amount of contextual information. The full model architecture resembles a u-shape that is able to propagate contextual information throughout the network, allowing to segment objects in an area using information from a larger overlapping area [88]. Several energy functions have been proposed to optimise the U-net. For binary classification, one of the most commonly used energy functions is the Binary Cross Entropy (BCE),

$$\mathcal{L}_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right) \log\left(1 - \hat{y}_i\right), \tag{3.10}$$

where N is the number of pixels of the training image, $y_i \in \{0, 1\}$ is the target value of pixel i, and $\hat{y_i} \in [0, 1]$ is the predicted probability for the pixel i. The capability of obtaining remarkably detailed

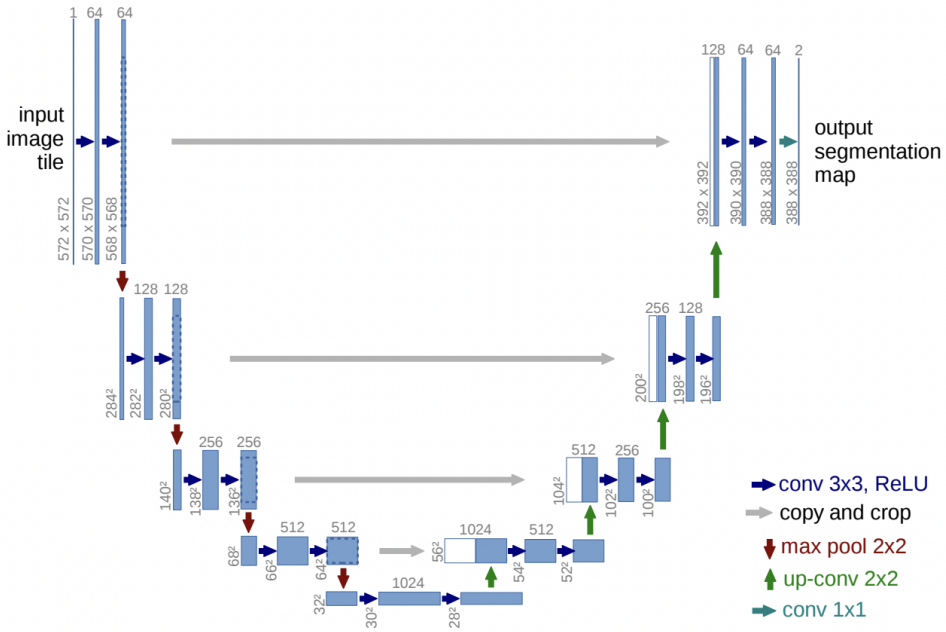


Figure 3.4: Example of a U-net architecture (retrieved from [61]). Blue boxes correspond to multi-channel feature maps and white boxes to copied feature maps. The number of channels is shown on top of the boxes. The number of feature maps is provided at the bottom of the boxes. The arrows denote the different operations.

segmentation with limited training samples quickly led to the use of the original U-net and improvements in several other fields outside medical imaging [89].

3.3.2 Generative adversarial networks

Generative Adversarial Networks, a deep-learning-based generative model, was first introduced in 2014 by Ian Goodfellow et al. [70]. The authors proposed a new estimation framework for generative models based on an adversarial process. In this framework, two models are trained concurrently: a generator, G, that captures the data distribution and is able to generate new examples from the problem domain, and a discriminator, D, that calculates the probability that a new sample was drawn from the training data as opposed to sampled from G.

The generator can be represented by a differentiable function, $G(\mathbf{z}, \theta_g)$, where $\mathbf{z}$ is an input vector, sampled from $p_{\mathbf{z}}(\mathbf{z})$ and θ_g represents the network parameters. The discriminator can also be represented by a differentiable function, $D(\mathbf{x}, \theta_d)$ where $\mathbf{x}$ is its input data, that can either be from the training data or generated by G, and θ_d represents the network parameters. The discriminator function outputs a single scalar, $D(\mathbf{x})$, that expresses the probability that $\mathbf{x}$ came from the training data rather than from G. Figure 3.5 shows the architecture of the GAN model. The models are trained simultaneously, where the discriminator is trained to maximise the probability of assigning the correct label to $\mathbf{x}$, while the generator is trained to minimise $\log(1 - D(G(\boldsymbol{z})))$. Therefore, the discriminator is updated to get better at discriminating between real and synthetic samples, while the generator is updated to get better at fooling the discriminator. This competitive adversarial behaviour can be seen as a two-player minimax game,

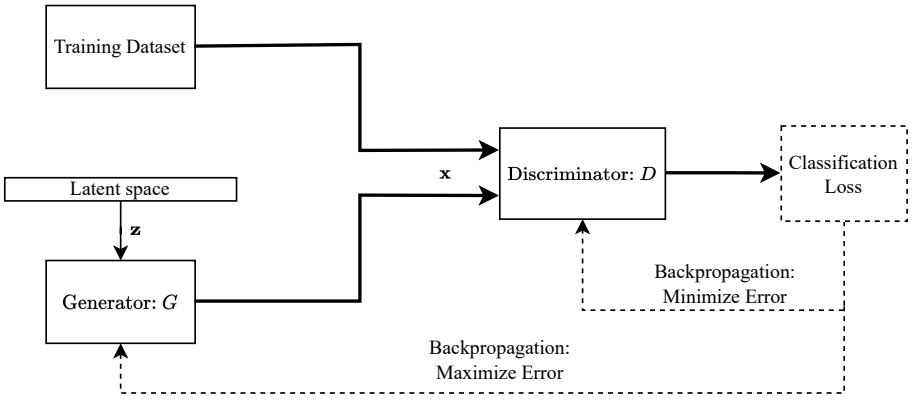


Figure 3.5: Simplified architecture of the GAN. The generator and the discriminator are trained using the discriminator's classification loss. The discriminator attempts to minimise the loss while the generator seeks to maximise it.

which is mathematically defined with the value function $V(G, D)$:

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]. \tag{3.11}$$

Furthermore, the authors realised that the second term of equation 3.11 might saturate in early training given that the discriminator is able to distinguish between real and low-quality synthetic samples with relative ease. Therefore, it was proposed as an alternative to train the generator to maximise $\log D(G(\boldsymbol{z}))$.

Although the generator and discriminator architectures were initially proposed as multilayer perceptrons, Radford et al. [90] quickly developed deep convolutional generative adversarial networks (DCGANs), which achieved promising results for unsupervised learning of images by using CNNs as the models. By addressing a conventional unsupervised problem by framing it as supervised, GANs quickly became one of the most widely used and successful generative models.

3.3.3 CycleGAN

Image-to-image translation is the task of converting an image from one domain to another. This is accomplished by learning the mapping between images of the different domains. Usually, paired image examples are used to train image-to-image translation models, such as the Pix2Pix [91], a model that achieved outstanding translation results using conditional adversarial networks. However, acquiring this type of data can be very expensive or even impossible.

Therefore, in 2017, Zhu et al. introduced CycleGAN [75], an extension of GAN that is able to learn image-to-image translation models without paired data. Although not being the first approach to attempt unpaired image-to-image translation, the method has proven to be able to generate successful outcomes for a variety of tasks backed by the fact that it does not rely on any task-specific, predefined similarity function between the domains, nor does it presuppose that they lie in the same low-dimensional embedding space [75].

The goal of the method is to learn the mapping between the domains X and Y, and vice-versa, given

the training data $x_i \in X$ with $i = 1, \ldots, N$ and $y_j \in Y$ with $j = 1, \ldots, M$[1] with distributions $x \sim p_{\text{data}}(x)$ and $y \sim p_{\text{data}}(y)$, respectively. The CycleGAN architecture is depicted in Figure 3.6. The model includes two generators $G : X \to Y$ and $F : Y \to X$ and two adversarial discriminators, D_X and D_Y, where D_X aims to distinguish between images $\{x\}$ from the X domain and translated images $\{F(y)\}$, and D_Y between images $\{y\}$ from the Y domain and $\{G(x)\}$. To be able to correctly translate domains, the authors proposed three types of terms for the objective function: adversarial loss, cycle consistency loss, and identity loss.

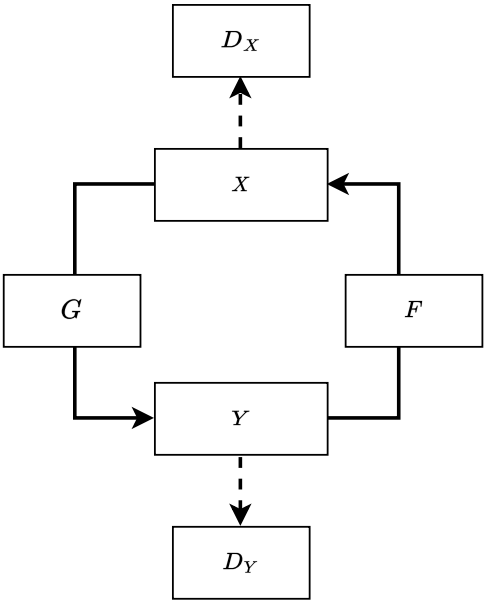


Figure 3.6: Simplified architecture of the CycleGAN. The generator G translates images from X domain to Y domain and F translates images from Y to X domain. The discriminators for the X and the Y domain are D_X and D_Y, respectively.

Adversarial Loss

The adversarial loss is responsible for approximating the distribution of the generated images to the target distribution. For the mapping function $G : X \to Y$, G attempts to generate images $G(x)$ that match the Y domain. Then, D_Y tries to distinguish the generated image from real samples, $y \in Y$. Therefore, the objective is given by

$$\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))], \tag{3.12}$$

where G attempts to minimise it and D_Y aims to maximise it, i.e., $\min_G \max_{D_Y} \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y)$. For the mapping function $F : Y \to X$, the process is similar, hence, F aims to generate images $F(y)$ matching the X domain, while D_X attempts to discriminate $F(y)$ from $x \in X$. Accordingly, the objective is given by

$$\mathcal{L}_{\text{GAN}}(F, D_X, Y, X) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D_X(x)] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log(1 - D_X(F(y))], \tag{3.13}$$

[1] Subscripts i and j will be omitted for simplicity.

where F aims to minimise it D_X tries to maximise it i.e., $\min_F \max_{D_X} \mathcal{L}_{\text{GAN}}(F, D_X, Y, X)$.

With the adversarial loss, the generators are expected to generate plausible images in the target domain, indistinguishable from the real ones. However, it does not guarantee an individual translation from input to the desired output, thus the need to introduce the cycle consistency loss.

Cycle Consistency Loss

The cycle consistency loss is able to further reduce the space of possible mapping functions by attempting to make the mapping functions cycle-consistent via an L1-norm reconstruction loss for a real image. For the X domain, the cycle ought to be able to reconstruct x, that is, $x \to G(x) \to F(G(x)) \approx x$. For the Y domain, the principle remains, thus, G and F should be updated during the training to ensure that y can be correctly reconstructed, i.e., $y \to F(y) \to G(F(y)) \approx y$. To ensure this procedure, the cycle consistency loss is defined as

$$\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\|F(G(x)) - x\|_1 \right] + \mathbb{E}_{y \sim p_{\text{data}}(y)} \left[\|G(F(y)) - y\|_1 \right], \tag{3.14}$$

where $\| \cdot \|_1$ represents the L1-norm.

Identity Loss

Inspired by Taigman et al. [92], the authors suggested the regularisation of the generators to force an identity mapping when real samples of the target domain are provided as input. Therefore, they defined the identity loss as

$$\mathcal{L}_{\text{idty}}(G, F) = \mathbb{E}_{y \sim p_{\text{data}}(y)} \left[\|G(y) - y\|_1 \right] + \mathbb{E}_{x \sim p_{\text{data}}(x)} \left[\|F(x) - x\|_1 \right]. \tag{3.15}$$

Although this loss is not fully required to successfully learn the mapping between the domains, it can improve the results depending on the translation task. The intuition behind the loss should be for the CycleGAN to only change parts of the image if required. Therefore, if something already looks like the target domain, the model should learn that it does not need to be changed.

Complete objective

The full objective is given by the weighted sum of the objectives referenced above:

$$\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F) + \alpha \mathcal{L}_{\text{idty}}(G, F), \tag{3.16}$$

where λ and α are parameters that determine the importance of each objective. Moreover, the goal is to obtain the generators that solve

$$G^*, F^* = \arg \min_{G,F} \max_{D_x, D_Y} \mathcal{L}(G, F, D_X, D_Y). \tag{3.17}$$

Training pipeline

Figure 3.7 represents the simplified training pipeline for a batch size of one[2]. The X domain is composed of SAR ship images and the Y domain of unpaired binary segmentation masks. For simplicity, only the forward propagation is shown. To train the D_Y discriminator, the input image, x, from the X domain goes through G, generating $G(x)$. The discriminator is then updated on the L2 adversarial loss upon its output for the real Y domain image, y, and the generated image, $G(x)$. To train the generators, the three losses are used. For the cycle consistency loss, the input image goes consecutively through G and F, $F(G(x))$, and the generators are updated on the L1 reconstruction loss. Moreover, for the identity loss, the input image goes directly through F, which is updated to ensure identity mapping via an L1 loss. The generator G is also updated via the L2 adversarial loss if $G(x)$ is not able to trick D_Y. The pipeline for when the input is an image y from the Y domain is similar.

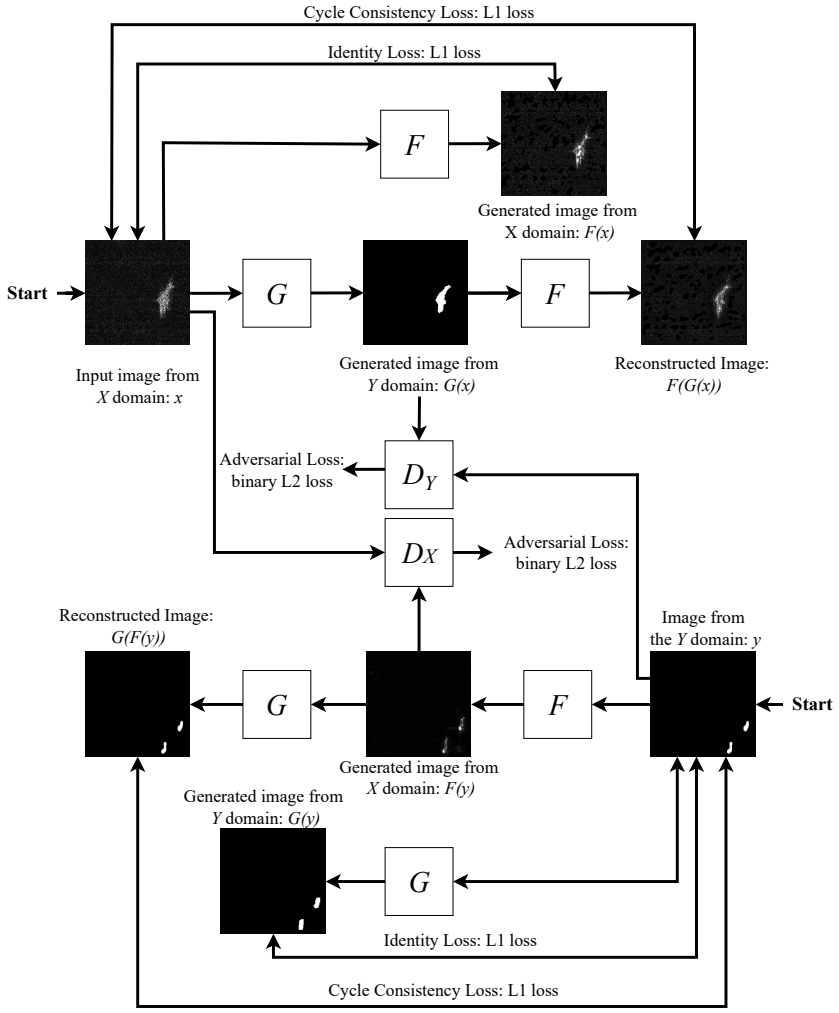


Figure 3.7: Simplified pipeline architecture for the CycleGAN training. The input image x goes through the forward cycle, $x \rightarrow G(x) \rightarrow F(G(x))$ and the generators are updated based on the L1 loss between the input and the reconstructed image. The input image x also goes directly through F which is then updated in an attempt to maintain identity mapping. As usual in GANs, the model goes through adversarial training, thus, G and D_Y are updated upon the decision of the discriminator. The principle is similar for the input image y start.

[2]The input might alternatively be a mini-batch, however for the sake of clarity, a batch of one is assumed.

Chapter 4

Proposed Approach

This chapter presents the implementation details of the proposed approaches. First, the characteristics of the two datasets used and the preprocessing performed on them are presented. Then, the framework for both approaches is described in depth.

4.1 Dataset

There are several public datasets suitable for ship detection in SAR imagery using deep learning models. Two well-known datasets from the SAR ship research community were separately used to train and evaluate the models proposed in this thesis. In this section, we will introduce them, list their characteristics, and explain the preprocessing procedure.

4.1.1 SAR-Ship-Dataset

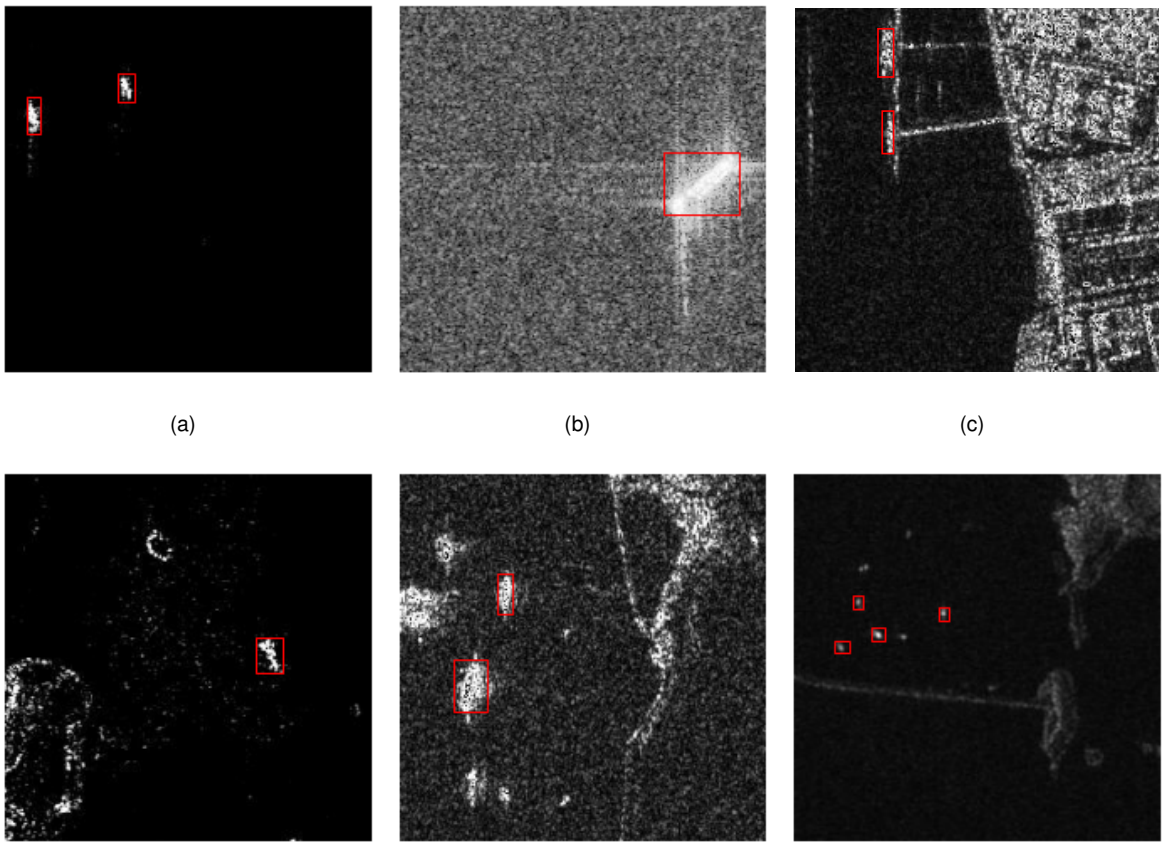
The SAR-Ship-Dataset [7], made available by Wang et. al in 2019 is one of the datasets used in this thesis. The dataset consists of 102 Chinese Gaofen-3 (GF-3) [93] images and 108 Sentinel-1 [94] images labelled by SAR experts. These images vary in imaging mode, resolution, incident angle, polarization, and background. Table 4.1 provides details on some of the characteristics mentioned above. The authors processed and cropped the satellite images to a size of 256x256, building a dataset with 39729 SAR ship chips with a total of 50885 ships labelled with the corresponding bounding box. Each chip has at least one ship.

The SAR-Ship-Dataset was chosen for a variety of reasons. First, the dataset is by far the largest one [12], which is valuable given the need for vast amounts of data to train deep learning models [95]. Then, the ships are multiscale. The smallest and the largest ship bounding boxes have an area of 24 and 25258 pixels, respectively. Moreover, the ships have distinct backgrounds, such as buildings, harbours, islands, and land. These surfaces often have double backscattering reflections, appearing similar to ships in the SAR imagery, which can lead to false positives in detection models. Land-ocean segmentation can mitigate this effect. However, this process severely limits the speed of ship detection and impedes automatic end-to-end ship detection. Sea condition is another extremely important factor

Table 4.1: Detailed information for the original SAR imagery for the SAR-Ship-Dataset (retrieved from [7]).

Sensor	Imaging Mode	Resolution Rg. × Az. (m)	Swath (km)	Incident angle (°)	Polarization	Number of Images
GF-3	UFS	3x3	30	20∼50	Single	12
GF-3	FS1	5x5	50	19∼50	Dual	10
GF-3	QPSI	8x8	30	20∼41	Full	5
GF-3	FSII	10x10	100	19∼50	Dual	15
GF-3	QPSI	25x25	40	20∼38	Full	5
Sentinel-1	SW	1.7x4.3 to 3.6x4.9	80	20∼45	Dual	49
Sentinel-1	IW	20x22	250	29∼46	Dual	10

which may difficult the ship's detection, particularly in rough sea conditions, as volume scattering may weaken the ocean-ship contrast. To offset the inevitable background diversity of practical applications, this dataset includes a large number of ships under complex backgrounds. The authors of the dataset believe that given its diversity, object detectors should perform well without the need for land-ocean segmentation. Figure 4.1 depicts some examples of images from the dataset in different conditions with the corresponding bounding boxes.



(a) (b) (c)

(d) (e) (f)

Figure 4.1: Examples of images from the SAR-Ship-Dataset with diverse backgrounds, such as calm sea conditions (a), rough sea conditions (b), harbour (c), and complex background with islands/land (d-f). The provided ship bounding boxes are represented in red.

Although the authors of the dataset refer to each of the 256x256 images as a chip, we will refer to them as images for simplicity and congruity.

Data Selection and Pre-Processing

It would usually be advisable to train deep learning models with as much representative data as there is available. However, in practise, this is often not possible. The CycleGAN, one of the models in this thesis, fits this description. Due to it being a considerably complex model, it is unfeasible to train it with the full dataset due to extremely long running times. Therefore, we were compelled to create a more concise version of the dataset. Although capturing a large diversity, a substantial part of the images from the original dataset are very similar, with most images having only one ship of relatively small size in calm sea conditions. Thus, randomly sampling the images from the original dataset would likely result in a new dataset with a predominance of these simple images. Since we want our CycleGAN model to work not only for simple images but also for complex ones, which are characterised by having ships in rough sea conditions, or near harbours or land, or of relatively large size, we must create the new dataset in such a way that it captures the original's diversity in a balanced manner. Due to the unsupervised nature of the problem, it is not possible to manually select the images of the new dataset. Nonetheless, we believe that the entropy of each image should be an adequate representation of the level of its complexity. Thus, we propose to create the new dataset with images that are equally distributed in entropy. Hence, we compute the Shannon entropy [96] for each image of the original dataset and utilise the return values to create a histogram with a fixed number of bins, which is depicted in Figure 4.2. In Figure 4.3 we show examples of images sampled from different bins, where it is possible to observe the increasing complexity of the images as we sample from bins with higher average entropy levels. Then, we sample an equal number of images from each bin, creating a new training set with a total of 7000 images. We call this dataset the concise-SAR-Ship-Dataset.

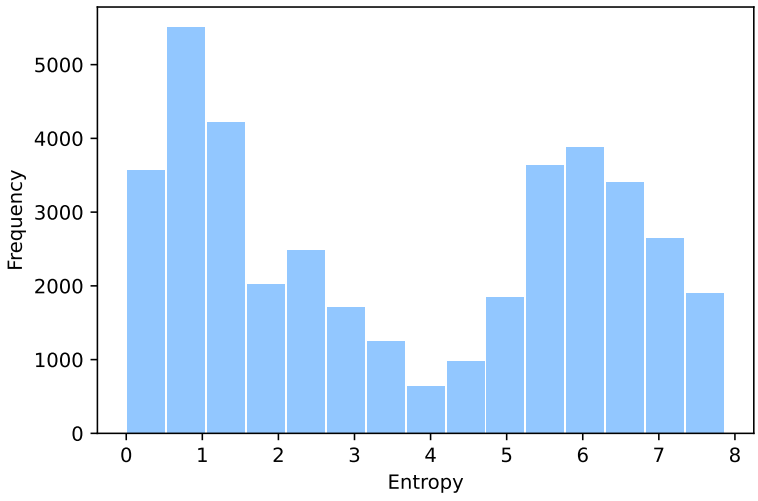
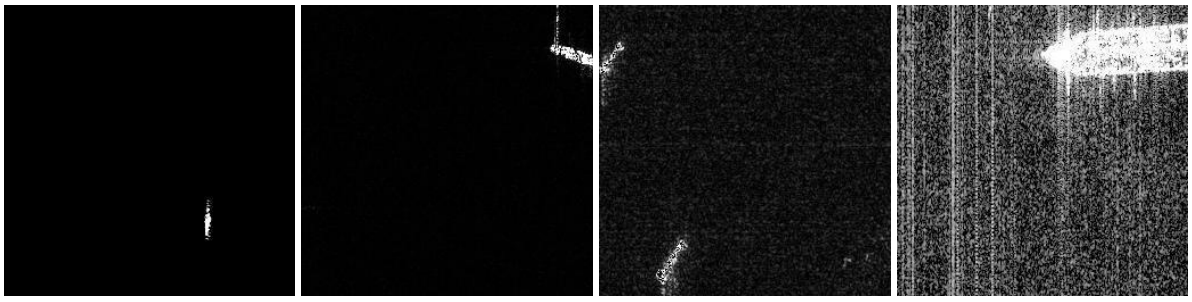


Figure 4.2: Distribution of the Shannon Entropy of the SAR-Ship-Dataset images.



(a) (b) (c) (d)

Figure 4.3: Images sampled from the first (a), fifth (b), tenth (c) and, fifteenth (d) bin of the histogram from Figure 4.2.

For the second approach, the UDSEP, as there are no computational limitations, we can use the complete dataset. This dataset and the concise-SAR-ship-dataset do not contain images from the test set, which was created by randomly selecting 1000 images from the original dataset.

Furthermore, the pixel values of the images, which range from 0 to 255, are normalized. Given the architecture of each of the proposed models, namely the output activation function, different normalisations are required for each of the approaches. For the first approach, the CycleGAN, a linear normalisation is performed, scaling the images to a [-1,1] range. For the UDSEP, the pixel values of the images that are used to train the U-net are linearly scaled to a [0,1] range.

The original dataset only provides the ship's bonding box, however, in this work we are interested in segmenting the ships. Therefore, for more accurate test results and for a fairer comparison with the supervised method, all the images from the test set and the concise-SAR-Ship-Dataset were annotated with ship segmentation through a threshold-based segmentation method supervised by us. More details on the method used to label the images are given in Appendix A. Moreover, for the purpose of analysing the method's results for the different types of images, a binary label of the level of complexity (simple or complex) is given to each test image. Typically, images that are considered simple contain offshore ships of average size in calm to moderate sea conditions. Images are deemed complex if the ships are excessively large, or in inshore conditions, or with a significant amount of spectral noise. Out of the 1000 images that make up the test set, 650 images were considered simple and 350 complex.

4.1.2 SAR Ship Detection Dataset (SSDD)

The SSDD, introduced by Li. et al [34] in 2017, is the other public dataset used in this thesis. This dataset consists of 2456 ships in 1160 images that vary considerably in size. The largest width/height of the image is 668 pixels, while the smallest is 190 pixels. Table 4.2 provides further details on the characteristics of the dataset. Although the initial version of the published article only provided the ship's bounding box, a 2019 release [11] provided additional annotations, such as the rotatable bounding box and the polygon segmentation. As with the previous dataset, we chose this dataset for a variety of reasons. For instance, it is very diverse. Small-sized ships, complex backgrounds, and dense arrangements near the harbours are some of the reasons that contribute to SSDD's diverse ship population. Then, this

Table 4.2: Detailed descriptions of SSDD imagery (adapted from [11]).

Sensors	RadarSat-2, TerraSAR-X, Sentinel-1
Polarization	HH, VV, VH, HV
Resolution	1m-15m
Places	Yantai, China; Visakhapatnam, India
Scale	1:1, 1:2, 2:1
Ship	Different sizes and materials
Sea condition	Good and bad conditions
Scenes	Inshore and offshore

dataset is by far the most utilised. According to [12], in May 2022, 67% of the published papers using deep learning for SAR ship detection utilised this dataset or improvements thereof. Moreover, contrarily to the previous dataset where the images are simply provided, the authors of this dataset also formulated some standards such as a train-test division, inshore-offshore protocol, and ship-size definition. These standards promote equitable methodological comparisons, endorsing the development of SAR ship detection. Figure 4.4 depicts some examples of images from the dataset in different scenes with the corresponding polygon segmentations.

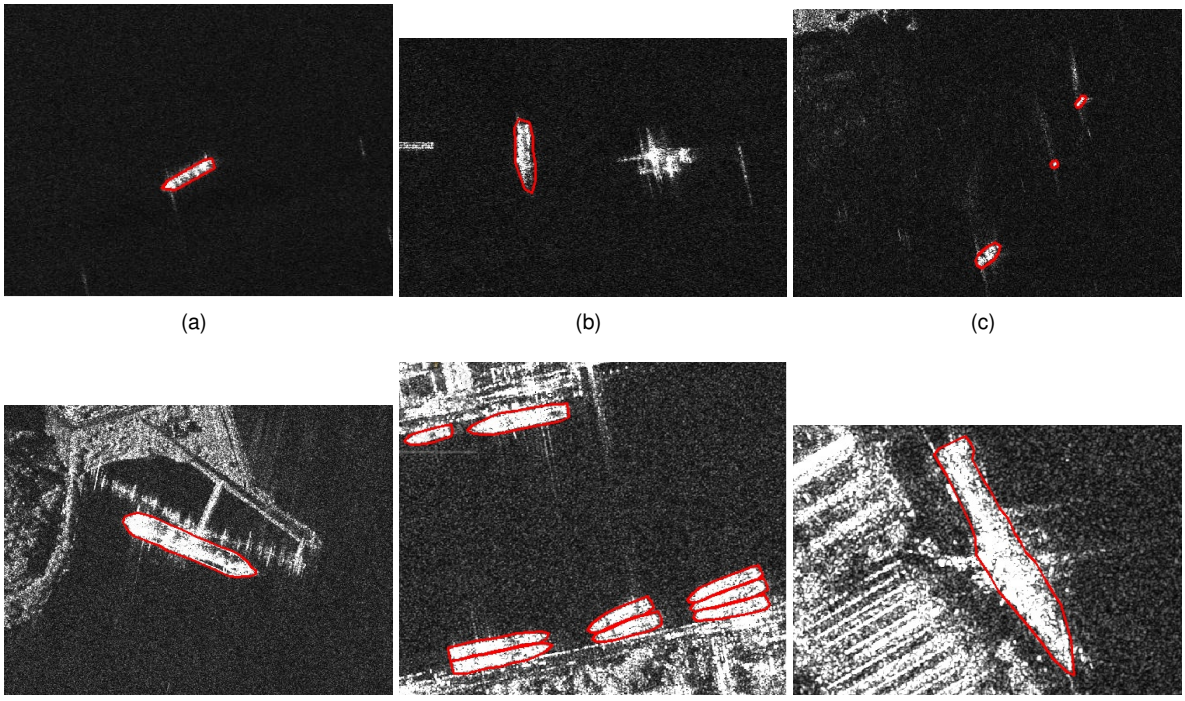


Figure 4.4: Examples of images from SSDD with different ship scenes, such as offshore (a-c), and inshore (d-f). The provided ship polygon segmentations are represented in red.

Data Selection and Pre-Processing

Given the relatively small size of the dataset, there was no need to do a preselection of images, hence the full dataset was utilised for both proposed models. Moreover, given that the image sizes vary and in order to maintain consistency throughout the models, each image was resized to 256x256 pixels. Since

the majority of resized images were shrunk, the SAR images were resized using OpenCV INTER_AREA interpolation, as suggested by [97]. The segmentation images were resized with INTER_NEAREST to keep binary values. Additionally, the dataset undergoes the same normalisation steps as the previous dataset. We use the standards provided by the authors, such as the train-test division. Thus, the train and the test set are made up of 928 and 232 images, respectively. Furthermore, there is no need to distinguish between complex and simple images given that the authors have already provided an offshore-inshore separation.

4.1.3 Comparisons between the datasets

A comparison between some characteristics of the datasets described above is employed. First of all, the SAR-Ship-Dataset is considerably larger than the SSDD. Even the concise-SAR-Ship-Dataset has more than seven times the images of the SSDD. Furthermore, the SSDD has a higher average ship per image value than the SAR-Ship-Dataset, which is 2.11 as opposed to 1.26. This is partially attributable to the fact that the SSDD contains a large number of images with various small ships, which are less common in the SAR-Ship-Dataset. This assertion is supported by Figure 4.5, which demonstrates that the relative ship area for the SSDD training set is significantly smaller.

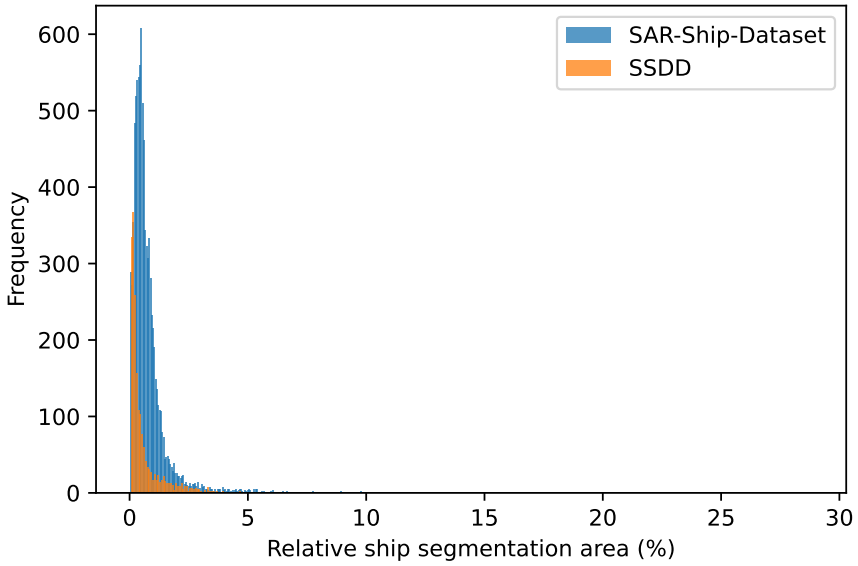


Figure 4.5: Distribution of the relative area of the ships segmentation for the training set of the SAR-Ship-Dataset and the SSDD. The relative area is given by the division between each ship's segmentation area and the total area of the image.

4.2 Methods

4.2.1 CycleGAN

The first proposed approach is based on the CycleGAN. In this framework, we aim to explore the image translation capabilities of the CycleGAN to provide semantic segmentation. To this end, we propose for the CycleGAN to learn the mapping between normal images and binary segmentation masks, and vice-versa. In the context of this thesis, and since we are only interested in the segmentation, the main goal is for the CycleGAN to learn the mapping between the SAR image domain and a binary domain, corresponding to the ships' semantic segmentation. Moreover, one independent CycleGAN model is trained for each of the datasets.

The CycleGAN is based on the original paper implementation [75]. Its simplified architecture is depicted in Figure 4.6. The model consists of two discriminators, D_L and D_{SAR} and two generators, $G_{L\_to\_SAR}$ and $G_{SAR\_to\_L}$.

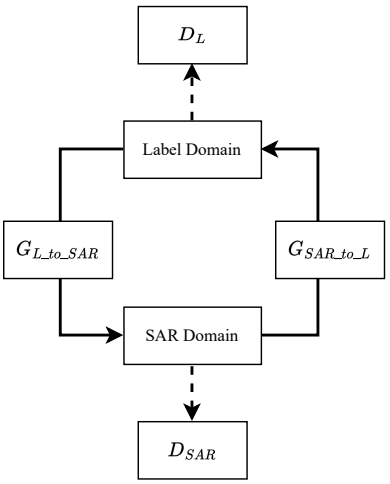


Figure 4.6: Simplified architecture of the CycleGAN. The generator $G_{L\_to\_SAR}$ translates images from the label domain to the SAR domain and $G_{SAR\_to\_L}$ translates images from the SAR domain to the label domain. The discriminators for the label domain and the SAR domain are D_L and D_{SAR}, respectively.

The discriminators are deep convolutional neural networks that receive as input an image and compute the likelihood that it came from the training data rather than being generated by the generators. Per the paper's specifications, the discriminators are PatchGAN classifiers. This model tries to classify whether each NxN patch of an input image is real or fake. The PatchGAN runs convolutionally across the image, averaging the values to compute the global likelihood of the input image. When compared to the classical ImageGAN, where NxN would equal the size of the input image, this approach allows for a cheaper model with consequently faster computing times. Following the recommendations of [75] and [91], a 70x70 PatchGAN is implemented for both discriminators. The architecture of the PatchGAN discriminator is depicted in Figure 4.7.

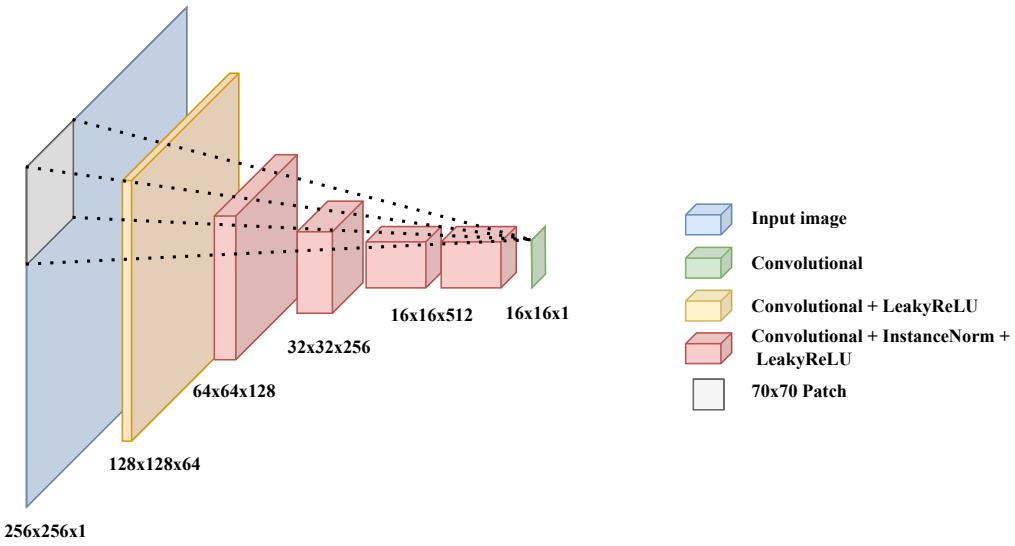


Figure 4.7: Schematic representation of the architecture of the PatchGAN discriminators. The LeakyRelu activation function has a 0.2 slope for the negative values. The numbers below the blocks represent the image size at its output.

The generators receive an image from one of the domains as input and translate it to the other domain. To accomplish this, the generators have an encoder-decoder architecture, where the input image is initially down-sampled to a latent space that goes through a series of ResNet blocks [98], followed by an upsampling to the size of the output image. The architecture of the generators is depicted in Figure 4.8.

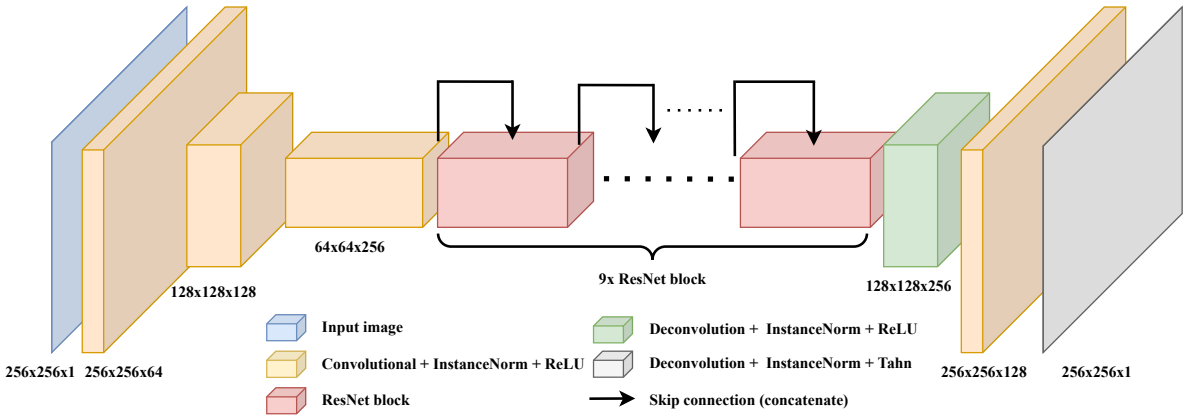


Figure 4.8: Schematic representation of the architecture of the generators. The numbers below the boxes represent the image size at its output. The ResNet blocks are made of a Convolution-InstanceNorm-ReLU block followed by a Convolution-InstanceNorm block. The convolutions of the ResNet block have a kernel size of 3x3 and a 1x1 stride. The input of each ResNet block is concatenated to its output.

Training data generation

To train the model, it is necessary to provide data from both domains. Since the data does not need to be paired, we just need to ensure that it is representative of the domain. For the SAR image domain, the concise-SAR-Ship-Dataset and the complete SSDD training set are used for each of the models.

For the binary label domain, it is required to produce two new datasets, $\mathcal{D}_{label\_SSD}$ and $\mathcal{D}_{label\_SSDD}$, with binary ship segmentation images. Figure 4.9 depicts the procedure developed to acquire this dataset for the SAR-Ship-Dataset. We apply the saliency method described in Section 3.2.2 followed by Otsu's thresholding to obtain these images. However, naively applying this method to a random SAR image would not always work, as the saliency method only yields satisfactory results for simple images, which are categorised by having ships in calm sea conditions with considerable ocean-ship contrast. Therefore, to ensure finer and more accurate segmentation, we first select the 7000 images with the lowest entropy from the first and second bin of Figure 4.2 (step A) before applying the saliency-thresholding method (step B). Additionally, the segmentation masks are post-processed using flood fill [99]. At the conclusion of the procedure, 7000 binary 256x256 ship segmentation images are obtained. A similar approach is used for the SSDD. In that case, we select the 300 images with the lowest entropy and then apply the saliency-threshold method. Additionally, data augmentation is performed on the obtained segmentation masks until they match the number of SAR images, thus avoiding the SAR domain images to be more represented. A combination of scaling, translating, rotating and elastic deformation [100] is used for the data augmentation.

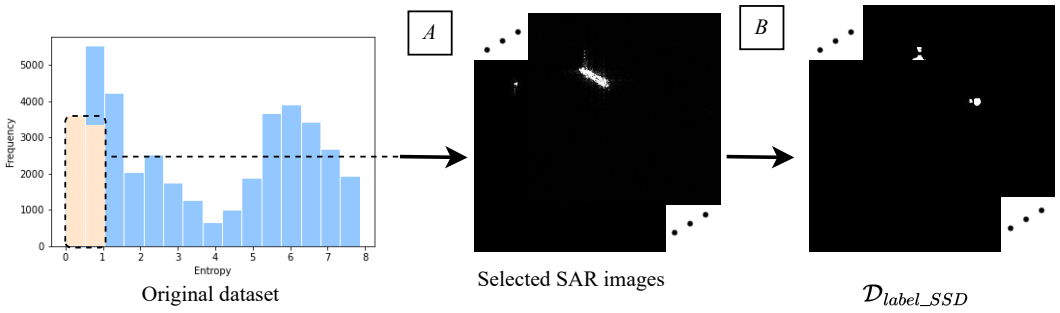


Figure 4.9: Proposed approach to obtain the ship segmentation images dataset $\mathcal{D}_{label\_SSD}$.

4.2.2 UDSEP

The second proposed framework, the UDSEP (U-net Detect-Select-Erase-Paste), is a self-supervised segmentation method. In this approach, we first generate synthetic labelled images from the original SAR unlabelled images. Then we use the synthetic images and the corresponding masks as training set for semantic segmentation with the U-net. Since the U-net is a supervised method, and in order to avoid generating the segmentation masks manually, we propose a novel algorithm to generate the new SAR images with the corresponding masks, the DSEP (Detect-Select-Erase-Paste) method. The overall framework of the UDSEP method is depicted in Figure 4.10. The framework is divided into a training phase (a) and an inference phase (b). The training starts with the generation of the synthetic labelled images where the DSEP method takes as input a SAR image x, and outputs a pair of images x_1 and m_1 where x_1 is a new SAR image and m_1 its segmentation mask. This process is repeated for each image in the training set. Then, the generated image pairs are used to train the U-net, F_ϕ with parameters ϕ, which is optimised to minimise the BCE loss between the output training masks and the target generated

masks. After training is complete, the optimised U-net is used directly to obtain segmentations of the test set. The detailed architecture of the U-net is shown in Figure 4.12. Similar to the CycleGAN approach, a separate UDSEP model is trained for each dataset.

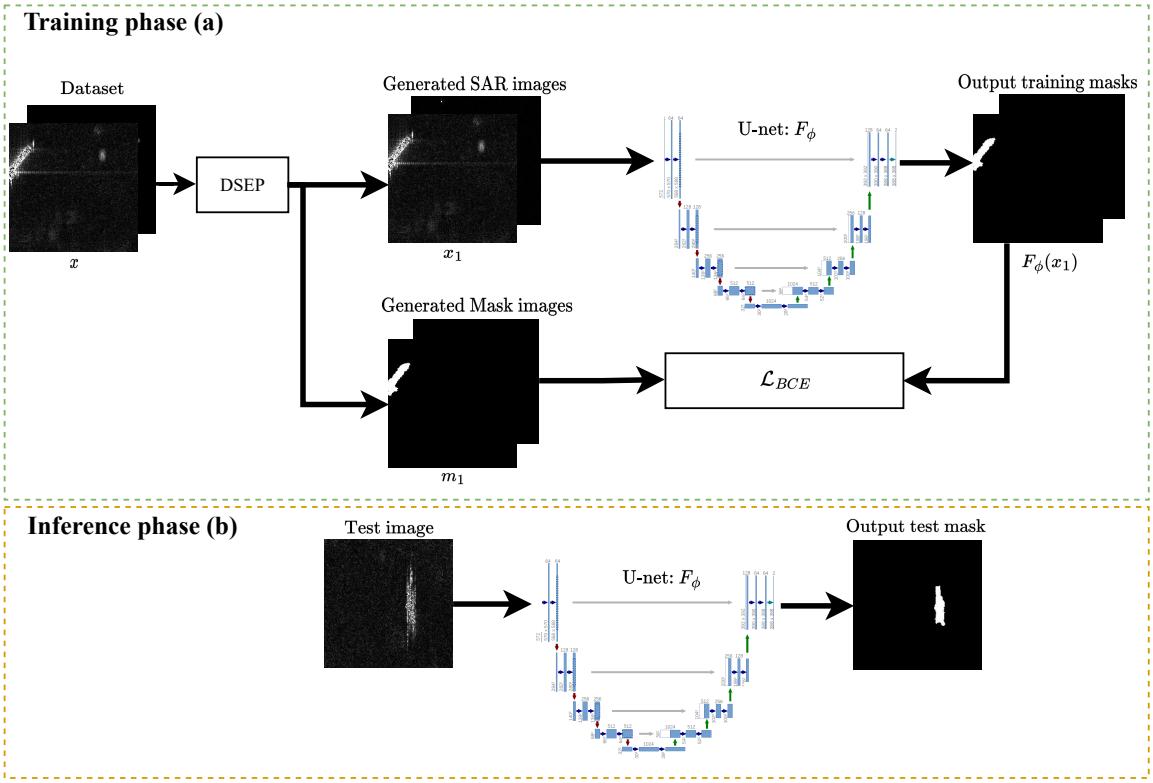


Figure 4.10: Framework for the UDSEP. (a) Training phase (b), Inference phase

DSEP method

In [101] Li et al. introduced an anomaly detection framework for high-resolution images with defects in local regions. To train the network, the authors proposed to generate images with irregularities. Therefore, they introduced a novel augmentation technique, the Cutpaste. This method consists of collecting a small rectangular image patch from a normal image and pasting it back into another normal image at a random location.

Inspired by their work, we propose the DSEP method. Named after its four main steps: Detect, Select, Erase, and Paste, the DSEP is an unsupervised augmentation process that receives as input an image with objects of the same type and transforms it to a new image, obtaining the corresponding binary segmentation mask with the location of the objects. In a concise manner, the DSEP method consists of the following steps:

- **Detect** the objects in an image.
- **Select** which of the detected objects to keep in the image and add their segmentation to the mask.
- **Erase** the objects that were chosen not to keep in the original image, by covering them with background.

- Optionally **Paste** augmented versions of the detected objects randomly in the image where the objects were erased, adding their segmentation to the mask.

Therefore, unlike Cutpaste, where the main objective was to create spatial irregularity by randomly collecting and pasting patches, we aim to, as far as possible, maintain the structure of the original image. For instance, if the detected objects are all kept and if the Paste step is not employed, the new transformed image will in fact be equal to the original input image. Thus, in practice, the method will work only as a mask generator.

Given that Detect step will be performed by an unsupervised object detector, which is likely not that robust, we introduced the Select and Erase steps. Therefore, if the initial object detector performs poorly and unduly detects a lot of objects, these steps make sure to mitigate the damage that would come from considering all those objects in the segmentation mask while training the U-net. Moreover, in an effort to increase the amount of information contained in the images without significantly altering them, we made the Paste step optional. Therefore, there is a p_a chance of pasting an augmented version of each selected object. Evidently, the quality of the images produced by the DSEP approach is highly dependent on the robustness of the initial object detector.

Training data generation

For our task, the process receives as input a SAR image and transforms it into a new image, obtaining the corresponding binary segmentation mask with the ship's location in a completely unsupervised way. Although the DSEP method can be used as an augmentation method, for this problem in particular, we are not interested in the original input image after the transformation process. Therefore, we refer to the DSEP as a transformation method as opposed to an augmentation method.

A schematic representation of the DSEP transformation method is depicted in Figure 4.11. First, the Detect step is employed. For our object detector, we use the previously trained SAR to label generator, $G_{SAR\_to\_L}$, from the CycleGAN model described in Section 4.2.1. Evidently, we use the generator model that was trained on the same dataset that we are transforming. Therefore, we apply the generator to the input image (step A). After thresholding the obtained translation, we obtain a binary map with the shapes of each detected object in the input image. Then, using the label and regionprops tools from [102], each individual object in the SAR image domain is separated (step B). At the end of the Detect step, we are left with all the objects that the generator was able to identify.

Next, in the Select step, we choose which of the detected objects to keep in the image and which ones to cover. To determine which objects to keep, we start by rejecting objects whose size deviates heavily from the mean, i.e., extremely large or small objects. Then, we keep up to N_{keep} objects. The rest of the objects are marked to be erased. Given that there is not an unsupervised robust way to give a ship's confidence level for each detected object, we randomly choose which ones to keep until the maximum of N_{keep} objects is reached (step C). Moreover, the segmentation of the chosen objects is added to the segmentation mask (step D).

After the objects to be erased are selected, we extract their segmentation from step A and, to ensure

that they are totally covered, apply the morphological operation dilation (step E). Then, we search the original image to find regions big enough for cutting that were not detected in step A, i.e., regions of background, and paste them at the locations of the objects to cover (step F). At the end of the Erase step, we are left with an image that is similar to the original but in which some of the objects have been replaced by the image's background. On first inspection, one might think that the Erase step is unnecessary and that we should keep all the objects detected in step A. However, as previously said, the initial object detector is not perfect, so we cannot guarantee that all of the detected objects are in fact ships. Therefore, considering every detected object as a ship could lead to the generation of poor quality SAR images for cases in which the SAR to label generator performed poorly with a lot of false detections. Furthermore, it is important to state that the generator will have fewer than N_{keep} detections for most images, so the Erase step will often not be employed.

Lastly, in the Paste step, there is a p_a chance of augmenting each of the kept objects and pasting them at a random location in the image that resulted from step F, and subsequently on the mask from step D. The augmentation consists of rotation and soft intensity jitter. At the end of the Paste step, we obtain a new SAR image (step H) and its corresponding segmentation mask (step I).

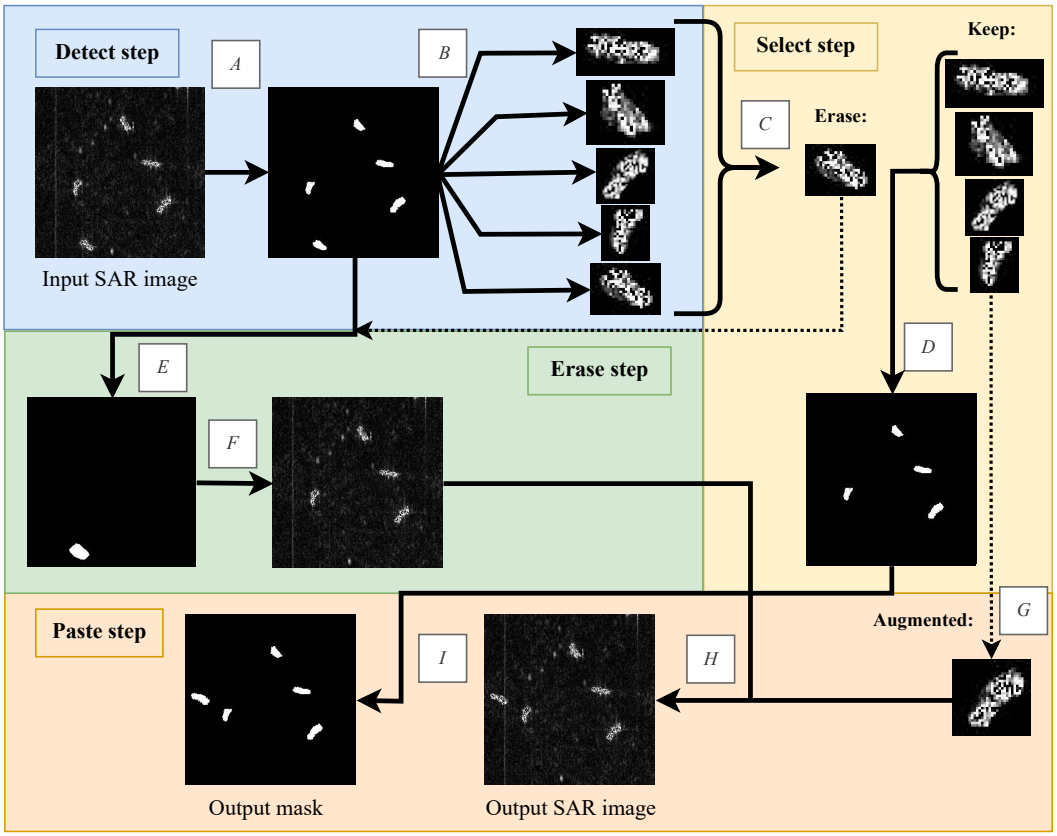


Figure 4.11: An overview of the proposed method. The DSEP method consists of four main steps: The Detect step (blue) receives as input the original SAR image and outputs the identified objects that are expected to be the existing ships; The Select step (yellow) receives all the identified objects and outputs which ones are to be kept and which ones are to be erased. It also outputs the segmentation mask of the kept objects; The Erase step (green) receives the objects to erase and the input SAR image and outputs the same image with those objects covered with background; The Paste step (orange) receives the output of the two previous steps and returns a new SAR image and the corresponding segmentation mask.

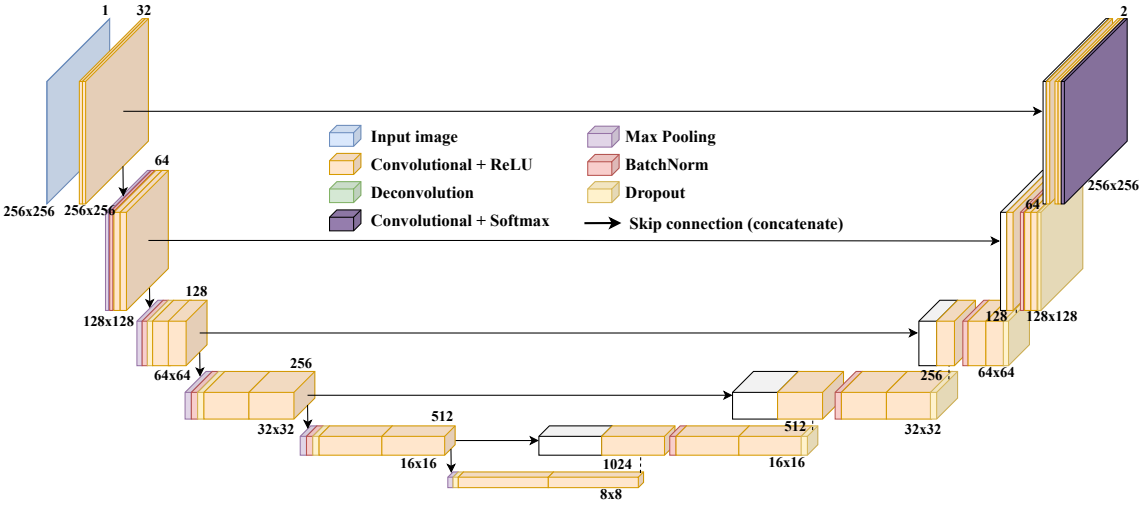


Figure 4.12: Schematic representation of the architecture of the U-net. The number of channels at the output of the boxes is shown on top of them, and the size of the feature maps at the bottom.

Chapter 5

Results

This chapter presents the evaluation metrics, training details, and results for the proposed methods as well as comparison methods. First, the data generated as a training set for the models will be analysed. Then, the segmentation and object detection results of the proposed methods will be compared to those of two conventional unsupervised methods, Saliency and CFAR, as well as a supervised U-net. In addition, a discussion and analysis of the results, an ablation study of the UDSEP method, and a computation evaluation are provided.

5.1 Evaluation Metrics

In order to evaluate the performance of the proposed methods, the results are evaluated with segmentation metrics and object detection metrics. All the deep learning models were trained several times. The best results were kept and are shown in this thesis.

For the segmentation evaluation, the IoU (Intersection over Union) and the F1-score for the ship class are computed pixel-wise. The IoU (equation 5.1) is the area of overlap between the ships' ground truth masks and their prediction masks divided by the area of union between the ships' ground truth masks and their prediction masks. The F1-score (equation 5.2) is a single evaluation metric that combines precision and recall by taking their harmonic mean. Precision (equation 5.3) refers to the proportion of correctly assigned ship pixels across all segmentation results, while recall (equation 5.4) refers to the proportion of correctly segmented ship pixels across all ground truth ship pixels. TP, FP, and FN stand for the pixel number of true positives, false positives, and false negatives, respectively.

$$\text{IoU} = \frac{\text{Area of overlap}}{\text{Area of Union}} \tag{5.1}$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.4}$$

For the detection evaluation, the F1-score is calculated for different IoU thresholds. The equations are similar to those used for the segmentation evaluation, with the exception that TP, FN, and FN are no longer defined by pixel but by objects. A ship detection is considered a true positive if the IoU value between the ground truth mask and its prediction is greater or equal to the threshold IoU value. If the IoU value is lower than the threshold, the detection is considered a false negative. A false positive occurs if there is a detection with no corresponding object in the ground truth.

5.2 Implementation Details

In this section, training and further implementation details of the models will be presented. All the deep learning algorithms were implemented with Python [103] using Keras [104] and Tensorflow [105] libraries. Moreover, the experiments were carried out with the following specifications:

- Python 3.9.10,
- Tensorflow 2.6.2,
- Cuda 11.6,
- CPU: Intel(R) Core(TM) i5-7600K CPU @ 3.80GHz,
- GPU: NVIDIA GeForce GTX 1070 - 8GB.

5.2.1 CycleGAN

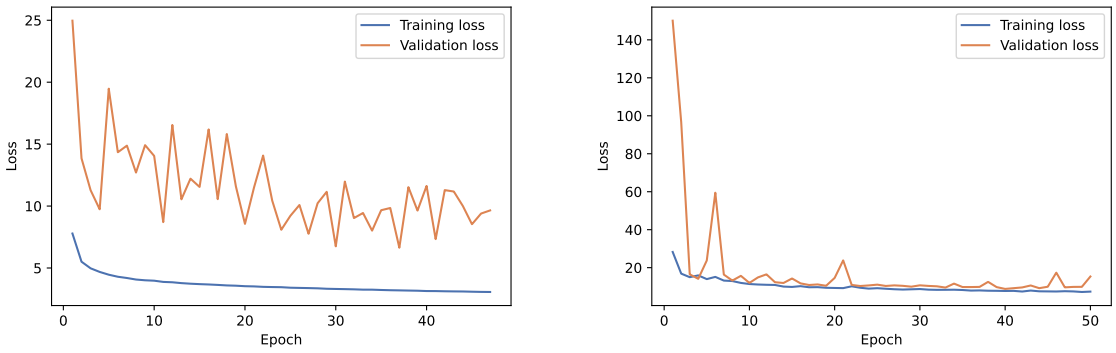
The discriminator models are optimized with an L2 loss with a factor of 0.5 in order to slow their changes relative to the generator. The discriminators are updated on both the real and generated images. However, to lessen the impact of model updates on the discriminator, a buffer of 50 generated images is kept for each of the discriminators, as suggested in [106]. At the beginning of training, the buffer is populated and then, for each iteration, there is an equal chance of using the generated image directly to train the model or of replacing it in the buffer with an existing image and using the replaced image to train the model. The use of this buffer allows the discriminator to avoid greedily attempting to beat the current generator but instead the last 50, creating a more generalized solution. The generators are updated based on the cycle consistency loss, adversarial loss, and identity loss. Following the paper recommendations [75], the cycle loss is given ten times the weight of the adversarial loss and double the weight of the identity loss.

GAN models usually do not converge, rather, an equilibrium is found between the generators and the discriminators. Moreover, analysing the evolution of the losses is not useful to determine whether training has ended, as often lower losses lead to visually worse results. Therefore, at the end of each training epoch, we save the generator models and several samples of generated images. After a few epochs of training, which is set to 15 epochs for the SAR-Ship-Dataset and 100 for the SSDD, we visually review the generated images and choose the model with the highest translation quality from the SAR domain to the label domain.

For the SAR-Ship-Dataset, the CycleGAN model is trained with the concise-SAR-Ship-Dataset and $\mathcal{D}_{label\_SSD}$. For the SSDD, the CycleGAN model is trained with the complete SSDD and $\mathcal{D}_{label\_SSSD}$. After the models are trained and selected, a 0.5 threshold is applied to the output of the $G_{SAR\_to\_Label}$ at test time to get the binary segmentation predictions. Pixels with values above 0.5 are considered as ship and below 0.5 are considered as background. Although we are only interested in the segmentation capability of the CycleGAN, the $G_{Label\_to\_SAR}$ generator is optimised to generate SAR-domain images from binary image inputs. The results of this translation operation are shown in Appendix B.

5.2.2 UDSEP

The U-net was trained from scratch, with an early stopping of 10 epochs of patience, using the Adam optimizer [107] with a learning rate of 0.001, with Xavier initialization and batch size of 5. We apply the DSEP method separately to the complete SAR-Ship-Dataset and the SSDD and split the generated image pairs with a 85%-15% ratio into the training and validation sets, respectively. The model is optimized to minimize the BCE loss function (equation 3.10). The parameter p_a is set to 0.2 to have a considerable chance of augmenting the ships, and N_{keep} is set to 4 since we have a moderate level of confidence in the CycleGAN's generator detections. Moreover, one independent U-net model is trained for each of the datasets. A plot of the train and validation loss for both datasets is depicted in Figure 5.1. After the model is trained, a threshold of 0.5 is applied to the U-Net's output in order to obtain the binary segmentation masks.



(a) (b)

Figure 5.1: Evolution of Training and validation loss for the UDSEP model. a) SAR-Ship-Dataset, b) SSDD.

5.2.3 Saliency

The framework introduced in Section 3.2.2 is applied for comparative purposes. The non-deep learning unsupervised method consists of a spectral residual approach for saliency detection followed by Otsu's thresholding. The method does not require any training, so it is directly applied to the test set. The size of the $h_n(f)$ matrix in equation 3.3 is set to 3x3 and the size of the Gaussian blur kernel from

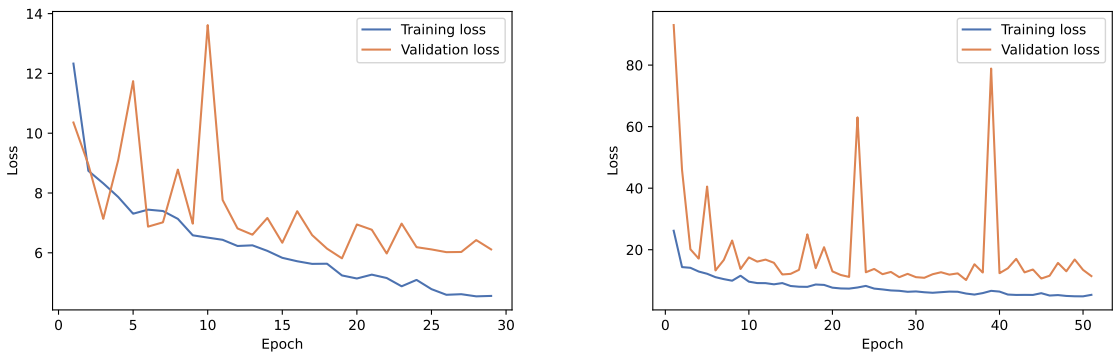
equation 3.9 is set to 7x7.

5.2.4 CFAR

A two-parameter CFAR algorithm based on Rayleigh distribution and morphological processing is also implemented for comparison. Inspired by [81], the CFAR parameters are set so that the false alarm rate is 0.04 and the target window size is 40x10. Two morphological post-processing operations are applied to the output of the CFAR detector and consist of eroding and dilating, which are then followed by small object removal and flood fill. Like the previous method, this framework does not require training so it is applied directly to the test set.

5.2.5 Supervised U-net

The supervised semantic segmentation method chosen for comparison is a U-net. The U-net has the same architecture and is trained similarly to the UDSEP method, therefore, the model is trained from scratch with an early stopping of 10 epochs of patience, using the Adam optimizer with a learning rate of 0.001, with Xavier initialization and batch size of 5. Moreover, the model is optimized to minimize the BCE loss function. For the SAR-Ship-Dataset, the model is trained with the 7000 images from the concise-SAR-Ship-Dataset and the corresponding segmentation masks obtained with the method presented in Appendix A. For the SSDD, the model is trained with the provided training set and the corresponding polygon segmentation masks. Figure 5.2 depicts the train and validation loss for both the datasets. A threshold of 0.5 is applied to the U-Net's output at test time to obtain the segmentation masks.



(a) (b)

Figure 5.2: Evolution of Training and validation loss for the supervised U-net. a) SAR-Ship-Dataset, b) SSDD.

5.3 Experimental Results and Analysis

5.3.1 Generated Data Analysis

$\mathcal{D}_{label\_SSD}$ **&** $\mathcal{D}_{label\_SSDD}$

Figure 5.3 shows several examples of the obtained binary ship segmentation images that make up $\mathcal{D}_{label\_SSD}$ and $\mathcal{D}_{label\_SSDD}$. The proposed saliency-threshold method was able to successfully generate binary segmentation masks from simple SAR images where the sea is calm and there is good contrast between ships and sea. Furthermore, the augmentations implemented for the SSDD also revealed valid results. However, it is important to note that the preselection of the low entropy images had a major impact on the results obtained. Without this preselection, the segmentation masks would have poor quality, which would later have a severe impact on the CycleGAN results.

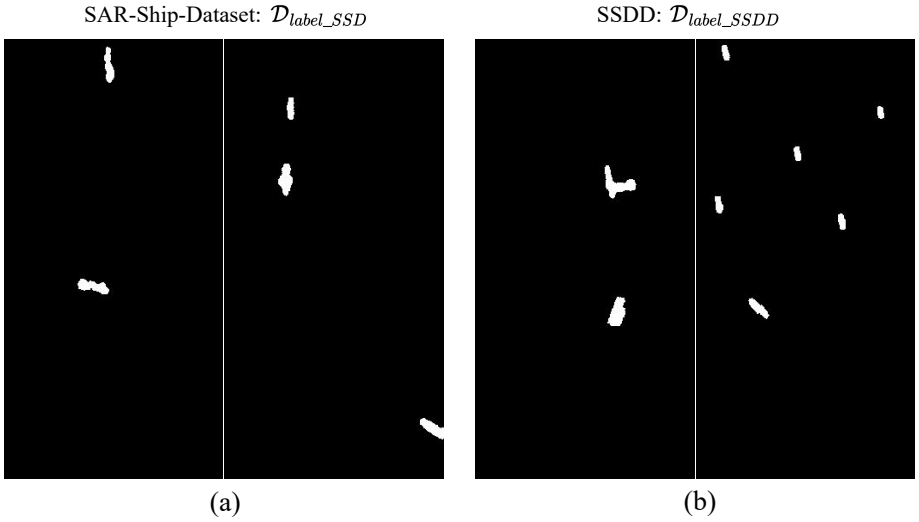


Figure 5.3: Binary ship segmentation masks obtained with the saliency-threshold method. (a) SAR-Ship-Dataset, (b) SSDD.

DSEP

Figure 5.4 shows a series of examples of the DSEP transformations. The first and fourth columns are the original SAR images, the second and fifth columns are the created SAR images, and the third and sixth columns are the corresponding segmentation masks. As can be seen from the first to the third row, the method performs extremely well for input images with low to medium complexity. In these cases, the initial object detector usually has fewer than 4 detections. Therefore, the method simply works as a mask generator, and given the success of the initial object detector, the obtained image pairs are comparable to those of the desired supervised method. Furthermore, sporadic augmentations that seem to improve the amount of information in the images can also be observed.

For inshore images, the quality of the image pairs generated considerably deteriorates. This was already expected given the limitations of the initial CycleGAN based object detector. Nonetheless, it is

on these images where we can better see the impact of the Select and Erase steps. For example, in the fourth row from Figure 5.4 (a), due to the complexity of the input image, the initial object detector identified 6 objects when the ground truth only indicates the existence of one ship. Since we defined 4 as the maximum number of objects to keep, 2 of those detected objects were covered, avoiding training the U-net with that presumably inadequate labelled data. Although we did not cover all the non-ship objects, we managed to minimize the impact of the poor CycleGAN detection. Moreover, there is always a chance to cover a ship, but we believe that unduly covering a ship from an image should have less negative impact on the network than training it with unlabelled ships.

In addition, as can be seen in the fifth row, the method yields good results even for input images with bright, noisy backgrounds. This and the general effectiveness of the DSEP method owe a great deal to the CycleGAN generator's robustness.

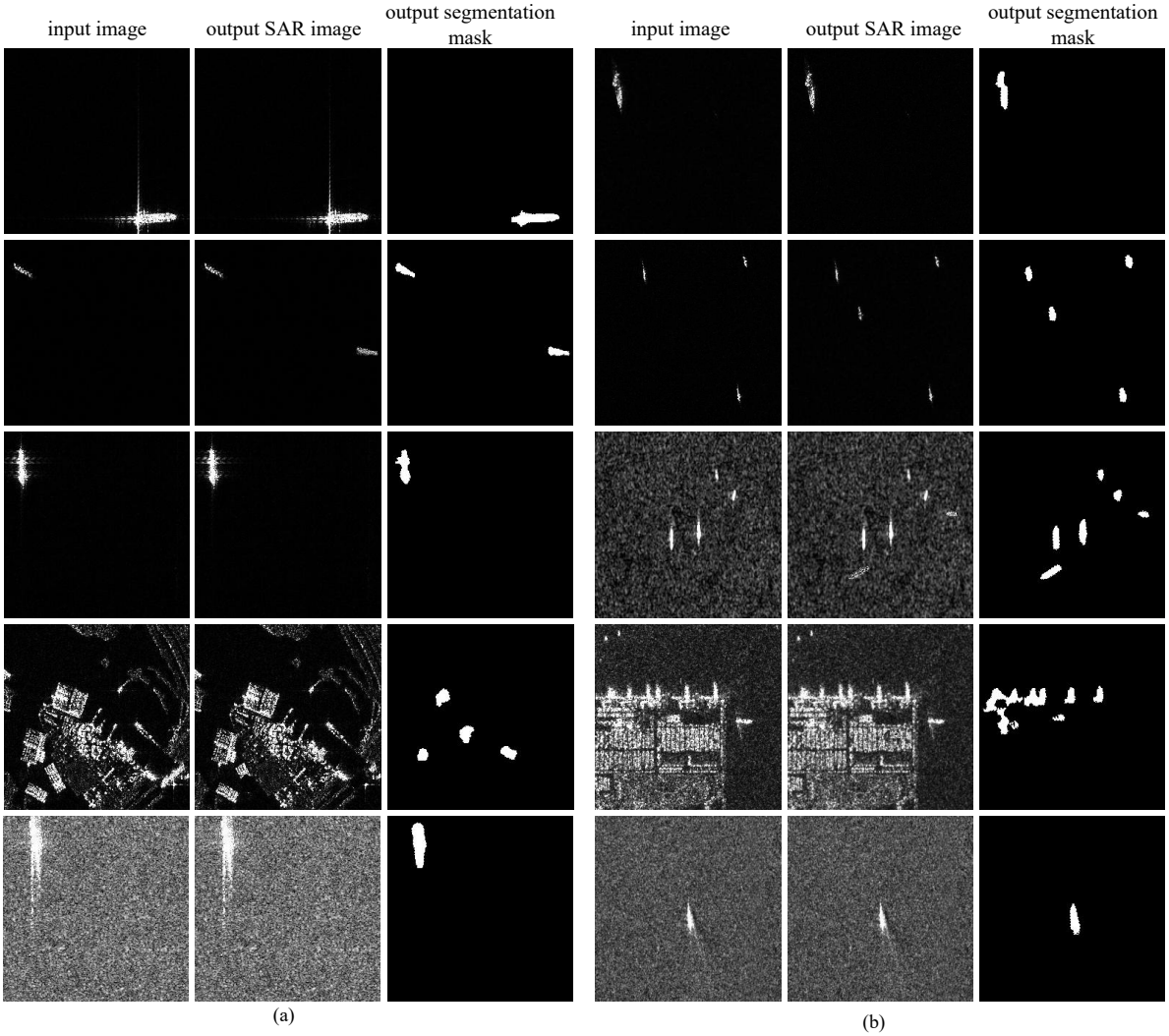


Figure 5.4: Original input SAR images and the result of the DSEP method: SAR image and its segmentation mask. (a) SAR-Ship-Dataset, (b) SSDD.

5.3.2 Results on SAR-Ship-Dataset

The test set from the SAR-Ship-Dataset, which consists of 1000 images randomly sampled from the original dataset, is used to evaluate the models.

Segmentation results

Table 5.1 presents the pixel-wise IoU and F1-score for the methods described above. To provide a more in-depth understanding of these results, several segmentation results for a selection of simple and complex images from the test set is represented in Figures 5.5 and 5.6, respectively.

Table 5.1: Segmentation results for the SAR-Ship-Dataset.

Method	IoU	F1-score
Supervised	0.773	0.841
Saliency	0.551	0.663
CFAR	0.441	0.564
CycleGAN	0.627	0.734
UDSEP	0.630	0.737

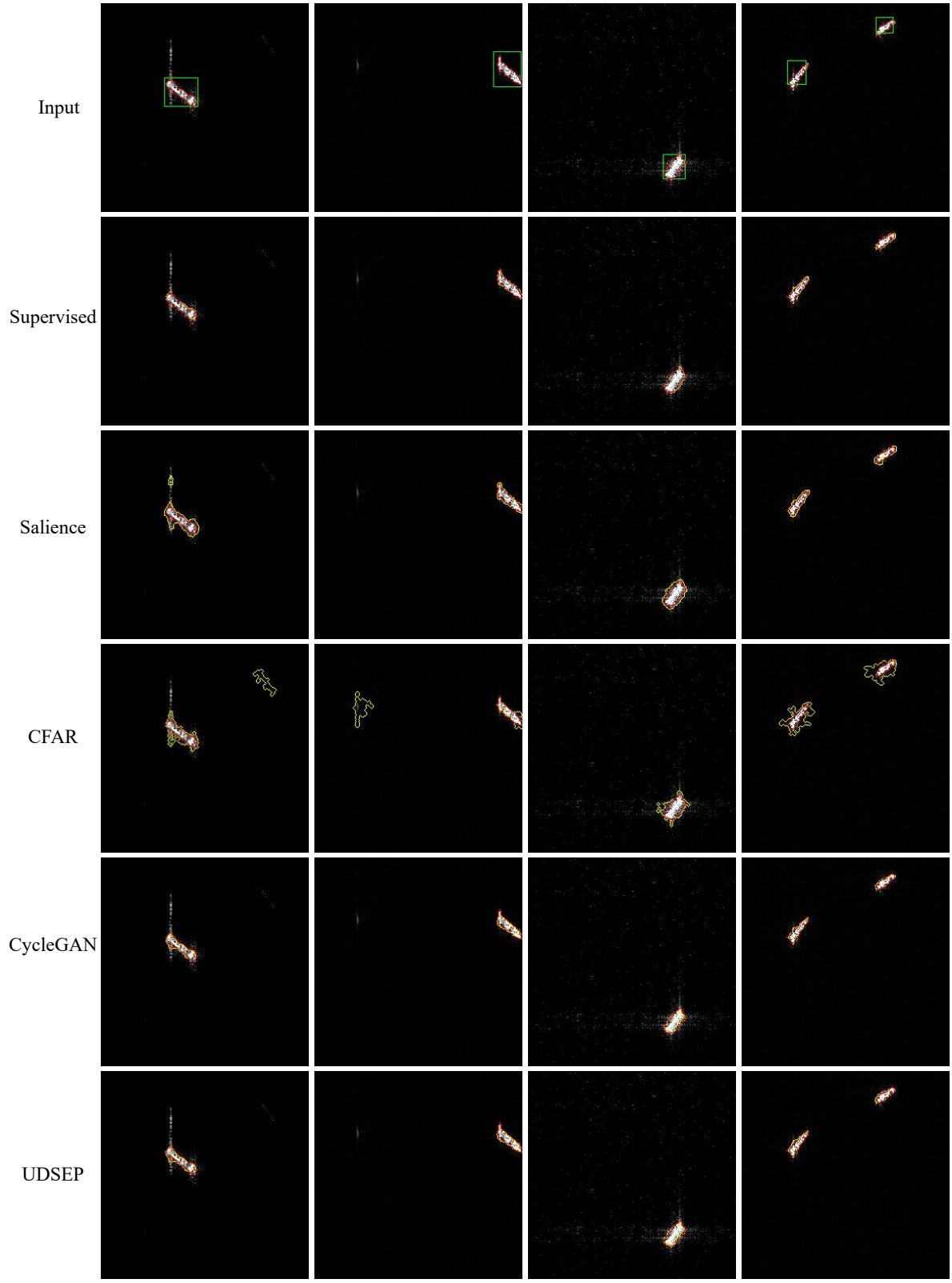


Figure 5.5: Segmentation results for simple test images. The original bounding box is represented in green in the input image, the ground truth segmentation obtained with the method explained in Appendix A is represented in red in all images, and the predictions for each model are represented in yellow.

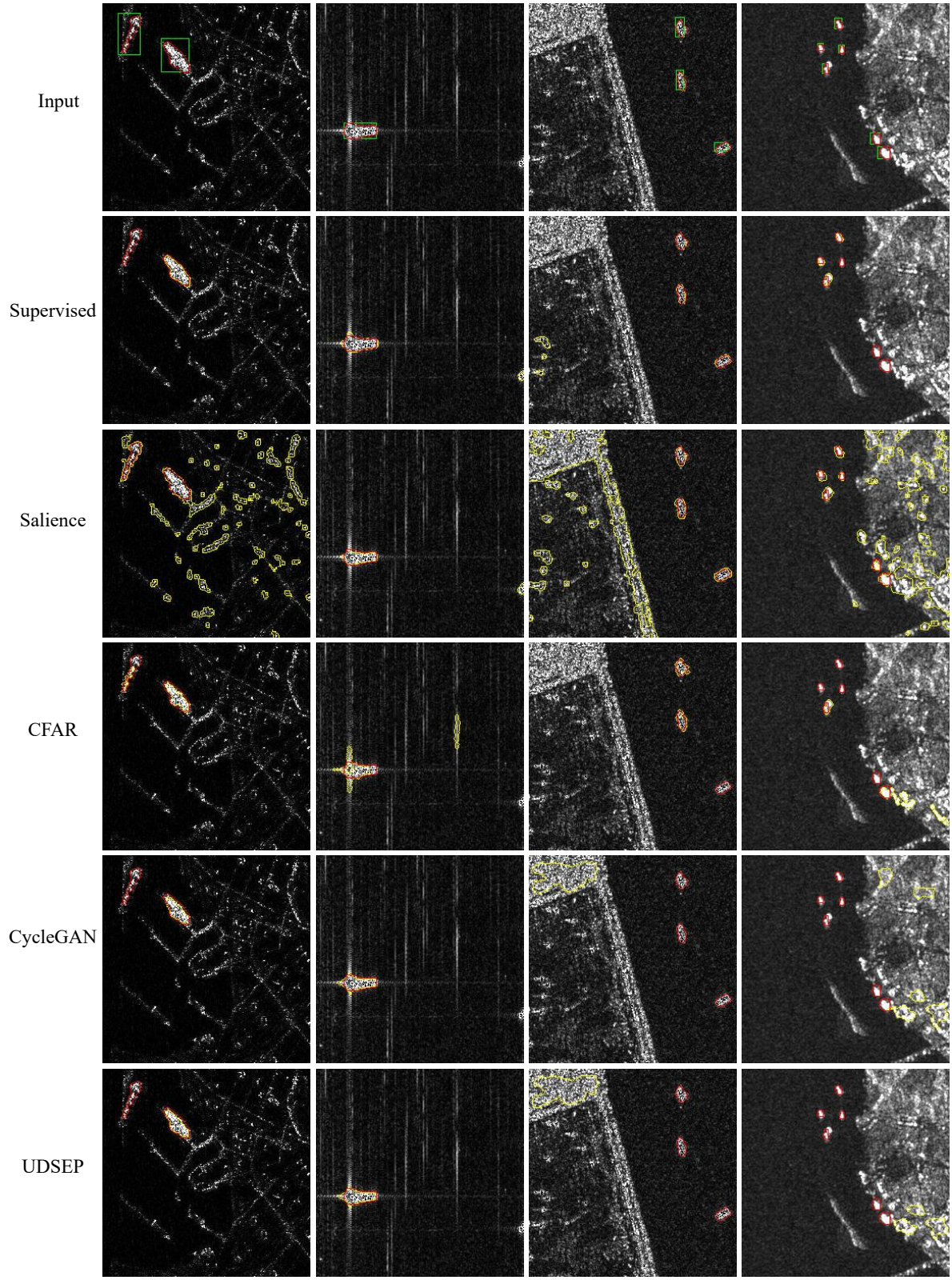


Figure 5.6: Segmentation results for complex test images. The original bounding box is represented in green in the input image, the ground truth segmentation obtained with the method explained in Appendix A is represented in red in all images, and the predictions of each model are represented in yellow.

Detection results

Figure 5.7 displays the F1-score for the methods described above across IoU thresholds ranging from 0.1 to 0.9 in steps of 0.1. To understand each technique's pluses and drawbacks in greater detail, the F1-score across the IoU thresholds computed separately for images with simple and complex background is represented in Figure 5.8.

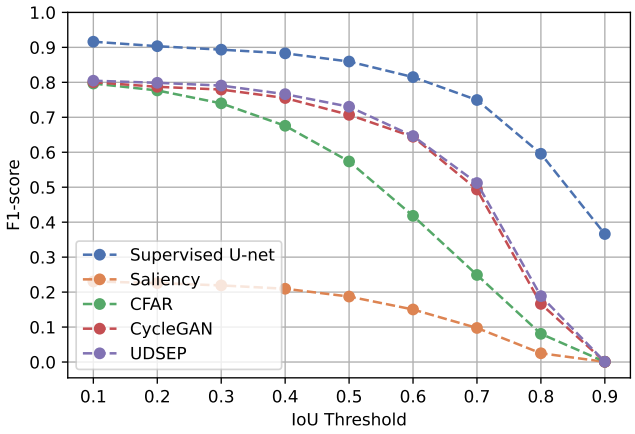
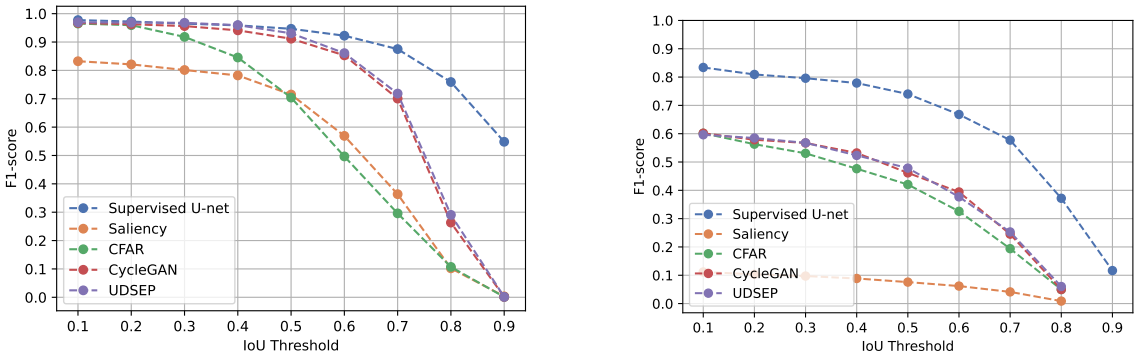


Figure 5.7: F1-score across the IoU thresholds for the different methods, for the complete SAR-Ship-Dataset test set.



(a) Simple images

(b) Complex images

Figure 5.8: F1-score across the IoU thresholds for the different methods, for different type of images from the the SAR-Ship-Dataset test set.

Analysis

First off, it should come as no surprise that the supervised method outperformed all the remaining methods, which are all unsupervised. Moreover, the good results for the supervised method validate the U-net as the feature extractor for SAR ship semantic segmentation. Furthermore, deep learning techniques performed significantly better than conventional techniques. Among the proposed methods, the UDSEP marginally outperformed the CycleGAN. For the segmentation metrics, the UDSEP achieved a 0.03 higher pixel-wise IoU and F1-score. For the object detection metrics, the evolution of the F1-score

throughout the threshold values is very similar, but still slightly in favour of the UDSEP. For the most common threshold value, 0.5, the UDSEP obtained an F1-score of 0.730 as opposed to the CycleGAN's 0.706. Given that the CycleGAN generator serves as the foundation for the UDSEP, the similarity of these results is not surprising. Nonetheless, the integration of the DSEP transformation method with the U-net improved the results of the original CycleGAN. The ablation study provided later in this section will help to better understand the impact of these additions. Moreover, the results of the proposed methods are still considerably lower than those of the supervised method, which as an F1-score at the 0.5 threshold of 0.859. It is worthwhile to comment on the saliency method, as its segmentation results are superior to those of CFAR, but its object detection results are significantly inferior. This is due to the fact that for complex or noisy images, the Saliency method behaves chaotically, considering everything as small objects, resulting in a high recall and low precision. This can be seen easily seen in Figure 5.6. Because the calculations are done by object rather than pixel, this behaviour has a greater impact on the object detection metrics.

When analysing the results separately for simple and complex images, several conclusions can be made. First, it is possible to notice that all methods perform reasonably well for simple images. Traditional methods usually have bigger segmentation masks than ground truth masks and, occasionally, have false detections. Deep learning methods detect the vast majority of ships, with few to no false and missed detections. Furthermore, it is of interest to note that up to the 0.5 threshold, the F1-score for the proposed methods is very similar to that of the supervised method. This and the visualisation of the results indicate that the results for simple images between the proposed and the supervised methods are very comparable. The deterioration of the results for higher threshold values is due to minor discrepancies between the predictions and the ground truth.

Furthermore, not only for the proposed methods but also for the comparison methods, there is a significant decline in results for the complex images. This was already expected, given the high degree of similarity between the ships and the background, which may include islands, harbors, noise, etc. This inevitably leads to more false positives. Nonetheless, there is a significantly higher gap in performance between the proposed and the supervised method for the complex images. There are some factors that can account for the lower performance of the proposed methods. First, when training the CycleGAN, even with the efforts of obtaining a concise training set with high image diversity, there is still a class imbalance in the training data. This is due to the fact that there are considerably more simple-to-medium-complex images than complex ones. For this reason, the CycleGAN will likely learn the mapping between simple images and the segmentation domain more effectively. Moreover, being an unsupervised method in which we simply fed images from the two domains, other than the shape of the provided segmentation images, there is nothing that is forcing the CycleGAN to learn to differentiate between shore and ships. For these reasons, and since the CycleGAN generator is directly related to the performance of both proposed methods, it is normal for the results to be worse for complex images. Nevertheless, the proposed methods still obtained satisfactory results for numerous complex images. Moreover, it is possible to observe that the saliency method differs from other approaches in that it produces good results for simple images but terrible for complex ones. This is the reason that

allowed the good extraction of the segmentation masks for $\mathcal{D}_{label\_SSD}$ and $\mathcal{D}_{label\_SSDD}$, but only after the preselection of low entropy images.

Ablation study

To further understand the impact of the components of the DSEP method, we conducted an ablation study on the UDSEP. Table 5.2 presents the conditions of the carried experiments. Case 0 represents the original UDSEP method. Case 5 corresponds to the UDSEP method but using the Saliency as the initial object detector instead of the CycleGAN generator. In case 1, we directly use the CycleGAN predictions to train the U-net, skipping the DSEP transformations. The same is done in case 6, but using the saliency predictions instead. In the remaining cases, some components of the DSEP transformation method are ignored. Figure 5.9 depicts the object detection results for the cases described.

Table 5.2: Conditions of the UDSEP ablation experiment.

Case	Detect	Select	Erase	Paste
Case 0	CycleGAN generator	✓	✓	✓
Case 1	CycleGAN generator	✗	✗	✗
Case 2	CycleGAN generator	✓	✓	✗
Case 3	CycleGAN generator	✗	✗	✓
Case 4	CycleGAN generator	✓	✗	✓
Case 5	Saliency	✓	✓	✓
Case 6	Saliency	✗	✗	✗

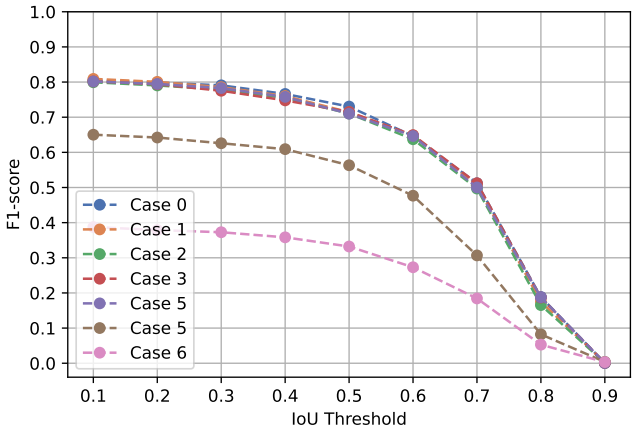


Figure 5.9: F1-score across the IoU thresholds for the different cases of Table 5.2, for the complete SAR-Ship-Dataset test set.

By analysing the outcomes of cases 0 through 4, it is extremely difficult to draw conclusions about the utility of the DSEP approach. Despite the fact that applying all the stages (case 0) results in a slightly better model, the performance of the remaining models is still extremely similar to the original. This can be explained for two reasons. First, given that the CycleGAN generator is already a highly robust object

extractor, there are typically less than 4 object detections in the majority of the images. Therefore, the Select and Erase steps, which were introduced as insurance, are unnecessary. Then, the SAR-Ship-dataset is very big, hence the augmentations provided by the Paste step are not that crucial. Therefore, although in our case the DSEP is not that relevant, we aimed at proposing a generic approach that could be used for a variety of scenarios. To validate the relevance of the DSEP, we implemented it with an initial less robust object detector, the Saliency (case 5). In this case, the DSEP method manage to increase the F1-score at the 0.5 threshold by 0.23 points when compared with the model where it was not implemented (case 6), indicating a substantial improvement.

Computation Evaluation

Table 5.3 presents model size, training and test time for all methods. As previously demonstrated, the accuracy of the traditional methods falls short, hence, further computational analysis will not be done. Considering the deep learning models, the CycleGAN model took considerably longer to train, which was expected, given its increased model and training complexity. However, since each model only needs to be trained once, training time is not the most crucial factor. In fact, the inference time is more important given the goal of processing the SAR data in real-time. The U-net-based methods have significantly faster detection speeds than the CycleGAN. This is due to the large model of the CycleGAN generators, which includes a series of ResNet blocks, inherently leading to more expensive computations. Furthermore, the detection speed of the proposed models is lower than the state-of-the-art, especially for the CycleGAN. Nonetheless, it is important to state that the UDSEP not only managed to slightly increase the detection performance of the CycleGAN but also managed to transfer its knowledge to a U-net which has 2.35 faster detection times.

However, it is important to note that this thesis's main objective was to maximise accuracy rather than obtain fast detection speeds. To reduce the time of inference, likely at the cost of accuracy, the original models could be replaced by lightweight versions, such as Lightweight U-Net [108] or a Lightweight CycleGAN implementation [109].

Table 5.3: Model size, training time and inference time per image. *The time taken to generate the images to train the models is accounted for in the training time.

Method	Model Size (MB)	Training time (hours)	Average inference time p/ image (ms)
Supervised	373.2	1.3	34.9
Saliency	-	-	1.7
CFAR	-	-	1.08×10^3
CycleGAN*	141.3	60	82.1
UDSEP*	373.2	9	34.9

5.3.3 Results on SSDD

The proposed and the comparison methods were also trained and evaluated individually on SSDD. The test set of the SSDD consists of 232 images.

Segmentation results

Table 5.4 presents the pixel-wise IoU and F1-score for the methods described above. Moreover, several segmentation results are represented for a selection of inshore and offshore scenes in Figures 5.10 and 5.11.

Table 5.4: Segmentation results for the SSDD.

Method	IoU	F1-score
Supervised	0.763	0.857
Saliency	0.466	0.585
CFAR	0.483	0.624
CycleGAN	0.554	0.676
UDSEP	0.571	0.693

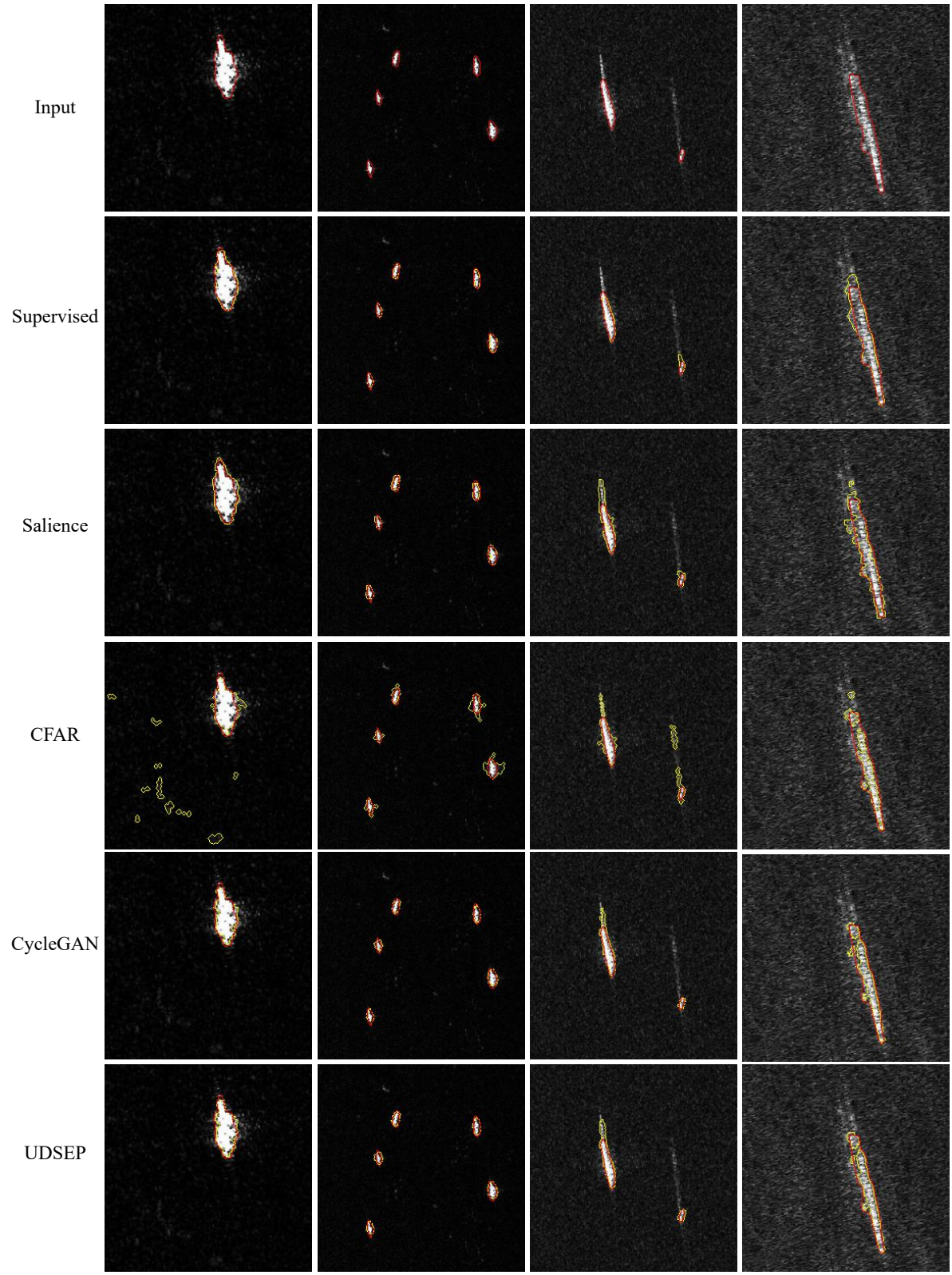


Figure 5.10: Segmentation results for offshore scene images from the SSDD test set. The ground truth segmentation is represented in red and the predictions of each model are represented in yellow.

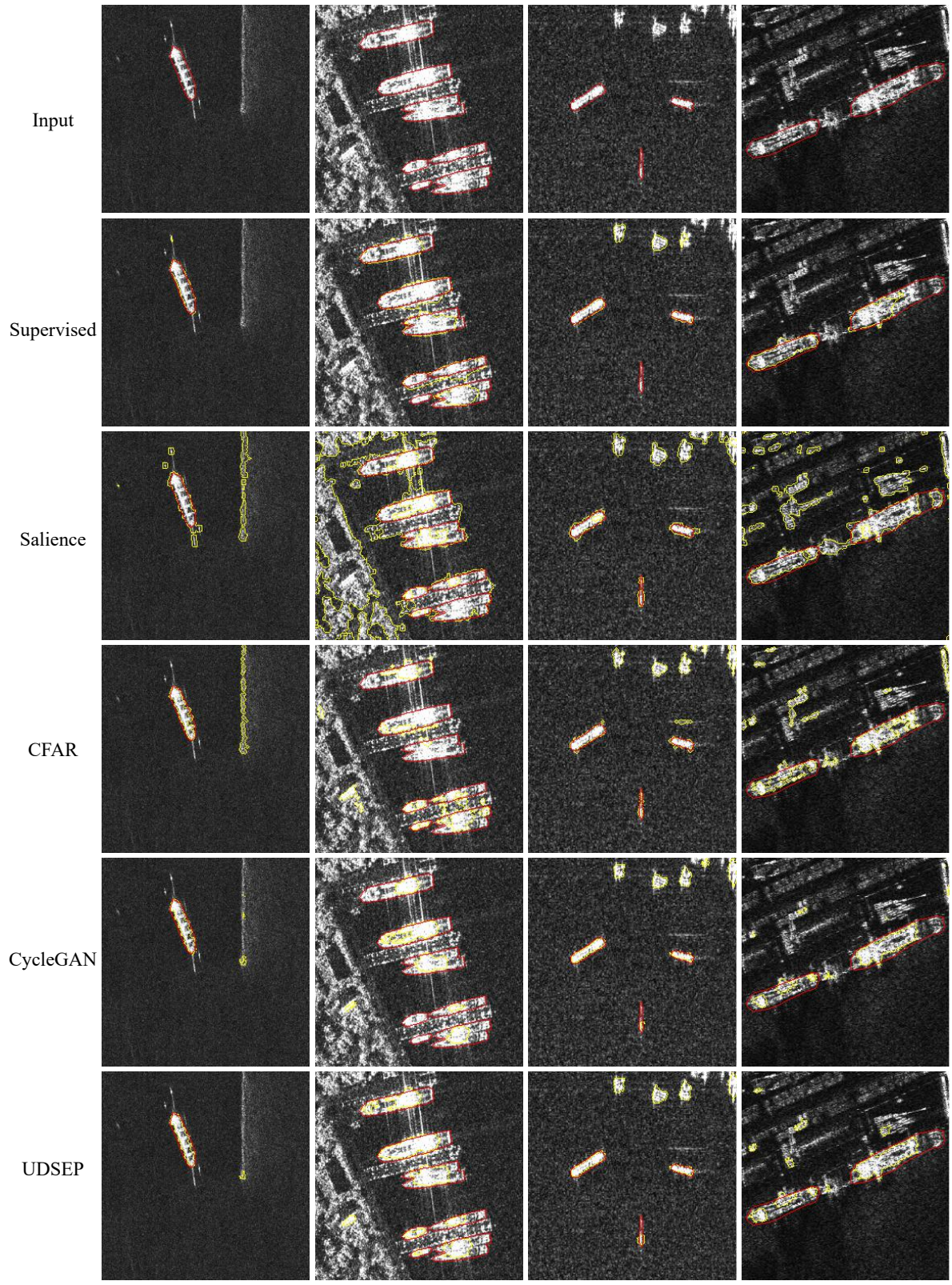


Figure 5.11: Segmentation results for inshore scene images of the SSDD test set. The ground truth segmentation is represented in red and the predictions of each model are represented in yellow.

Detection results

Figure 5.12 displays the F1-score for the different methods across the IoU thresholds for the complete test set. Figure 5.13 displays the same metric but with the F1-score computed separately for offshore and inshore scene images.

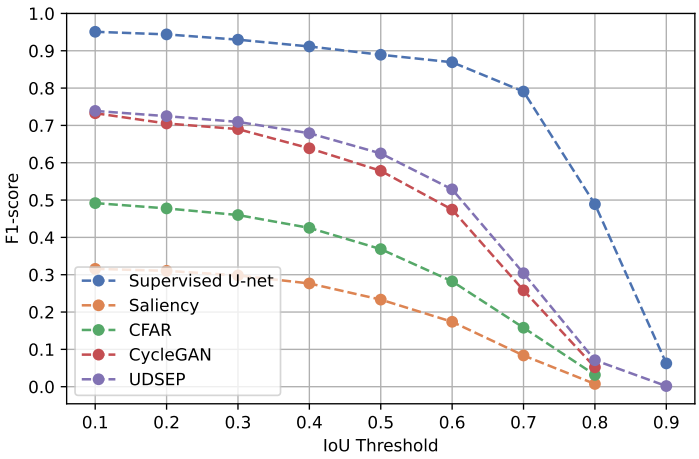
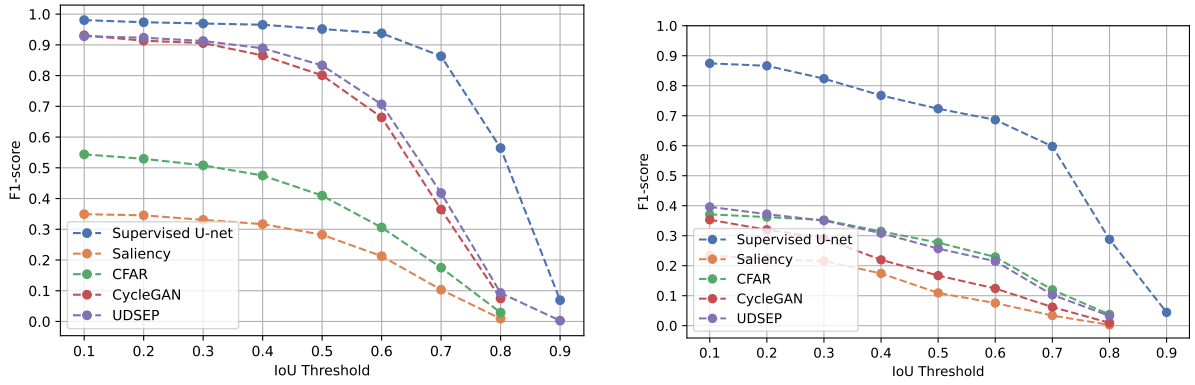


Figure 5.12: F1-score across the IoU thresholds for the different methods, for the complete SSDD test set.



(a) Offshore images

(b) Inshore images

Figure 5.13: F1-score across the IoU thresholds for the different methods, for different type of images from the the SSDD test set.

Analysis

The results are consistent with the previous data set. The supervised method clearly outperformed the remaining unsupervised methods, and the deep learning methods outperformed the traditional methods. Of the proposed methods, the UDSEP obtained better segmentation and detection results, with an F1-score at the 0.5 threshold of 0.625 compared to the CycleGAN's 0.578. In light of the fact that the CycleGAN generator has lower performance for this dataset, the improvements of the DSEP method are more noticeable. In addition, the SSDD is a considerably small dataset, thus the augmentations provided by the Paste step likely had a bigger impact than on the SAR-Ship-Dataset. Moreover, the

UDSEP managed to clearly outperform the CycleGAN model for inshore images. This is likely due to mitigations endorsed by the Select and Erase steps, which encouraged the U-net to not have as many false detections as it otherwise would have.

Overall, the results for the proposed methods on the SSDD dataset were worse than on the SAR-Ship-Dataset. Several factors support this conclusion. First, the SSDD has a significantly smaller size, which will unavoidably result in models being more vulnerable to overfitting. Then, as seen in Figure 4.5, this dataset is mainly composed of ships of small size. Since the IoU becomes more sensitive as the area of the object decreases, slight discrepancies between the ground truth and the prediction might result in low IoU values for small objects. Moreover, smaller objects are naturally harder to detect, given that their features may disappear in deeper layers. Nevertheless, the models performed reasonably well for the SSDD, validating the generalisability of the presented methodologies.

Chapter 6

Conclusions

The main goal of this thesis was to develop unsupervised deep learning techniques for ship detection in SAR images. For this purpose, two fully unsupervised frameworks were proposed for ship segmentation: the CycleGAN, an image-to-image translation model which was explored for segmentation, and the UDSEP, a U-net trained on synthetic generated data from a novel augmentation process. Although still inferior to those of the supervised method, the results obtained for the two proposed methods were extremely promising, especially given the fully unsupervised nature of the approaches. Evaluation on simple/offshore images revealed overall competitiveness with the supervised method. Evaluation on complex/inshore images proved that the proposed methods are still insufficiently robust for this type of image. However, it is important to state that ship detection in this type of image is an active challenge, even for supervised research. Given their essentially inferior robustness, the struggle for unsupervised approaches is not surprising.

Moreover, the CycleGAN approach revealed to be effective and robust for domain translation between the SAR domain and the ship segmentation domain. Consequently, the UDSEP managed to enhance the CycleGAN model in two aspects. First, there was a slight improvement in detection quality. Then, there was a severe reduction in detection time, with a decrease of over 57%.

Furthermore, the author believes that the developed work should inspire fellow researchers to develop unsupervised frameworks for SAR ship detection, which can fully exploit the amount of available raw SAR data and the increasing GPU performance.

6.1 Future Work

Future improvements could be made to attempt to improve the accuracy of the models. First, further studies could be employed to attempt to increase the robustness of the models. For instance, the CycleGAN could benefit from a distinct architecture for each of the generators, which could be more task-specific to the domain translation. Moreover, the DSEP method could benefit from improvements in the Select step. Currently, after a size-dependent preliminary removal, the selection of the objects to keep and to erase is essentially random. Several unsupervised strategies could be employed to attempt

to address this. For example, a binary cluster could be conducted by K-means to try to classify each object as a ship or non-ship. Then, objects classified as ships would be kept, and objects classified as non-ships would be marked to be erased.

Second, the low accuracy of the inshore scene should be addressed. For the CycleGAN, resolving the class imbalance between the offshore and inshore images could be a good start. The strategy introduced by [73], which used GAN and K-means to create a scene binary cluster and then augmented the inshore scene images, would be a good approach to augment these images in an unsupervised manner. The UDSEP method would indirectly benefit from this improvement.

Lastly, in an effort to make models suitable for real-time detection, the original backbone structures of the models could be replaced with lightweight versions. To this end, we propose not to change the CycleGAN model but experiment with lightweight versions of the U-net to check if it would still be possible to fully transfer the knowledge obtained with the current CycleGAN model.

Furthermore, to further test and refine the proposed methods, it would be of interest to evaluate them in other segmentation tasks using datasets other than SAR ship.

Bibliography

[1] Geospatial Intelligence Pty Ltd. Maritime surveillance in focus. https://storymaps.arcgis.com/stories/f03366c92d14470c982bf17c46e21308, 2021. Accessed: 2022-06-15.

[2] S. Widjaja, T. Long, H. Wirajuda, H. V. As, E. Bergh, A. Brett, D. Copeland, M. Fernandez, A. Gusman, S. Juwana, T. Ruchimat, S. Trent, and C. Wilcox. Illegal, Unreported and Unregulated Fishing and Associated Drivers, 2022.

[3] World Wild Life. Illegal Fishing. https://www.worldwildlife.org/threats/illegal-fishing, 2021. Accessed: 2022-06-15.

[4] D. Kar and J. Spanjers. Transnational Crime and the Developing World. Global Financial Integrity, 2017.

[5] UNHCR. Protection, saving lives, solutions for refugees in dangerous journeys: Routes towards the Western & Central Mediterranean Sea, 2022.

[6] IMO 2000. SOLAS Convention 1974 Chapter V regulation 19, 2000.

[7] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei. A SAR Dataset of Ship Detection for Deep Learning under Complex Backgrounds. *Remote Sensing*, 11(7), 2019.

[8] U. Kanjir, H. Greidanus, and K. Oštir. Vessel detection and classification from spaceborne optical images: A literature survey. *Remote Sensing of Environment*, 207, 2018.

[9] C. Oliver and S. Quegan. *Understanding Synthetic Aperture Radar Images*. EngineeringPro collection. SciTech Publ., 2004.

[10] G. H. Born, J. A. Dunne, and D. B. Lame. Seasat Mission Overview. *Science*, 204(4400):1405–1406, 1979.

[11] T. Zhang, X. Zhang, J. Li, X. Xu, B. Wang, X. Zhan, Y. Xu, X. Ke, T. Zeng, H. Su, I. Ahmad, D. Pan, C. Liu, Y. Zhou, J. Shi, and S. Wei. SAR Ship Detection Dataset (SSDD): Official Release and Comprehensive Data Analysis. *Remote Sensing*, 13(18), 2021.

[12] J. Li, C. Xu, H. Su, L. Gao, and T. Wang. Deep Learning for SAR Ship Detection: Past, Present and Future. *Remote Sensing*, 14, 2022.

[13] D. Crisp. The State-of-the-art in Ship detection in Synthetic Aperture Radar imagery. *organic letters*, 2004.

[14] I.-I. Lin and V. Khoo. Computer-based algorithm for ship detection from ERS SAR imagery. In *Proc 3rd ERS Symp Space Service Environ*, volume 414, page 1411, 1997.

[15] I.-I. Lin, L. K. Kwoh, Y.-C. Lin, and V. Khoo. Ship and ship wake detection in the ERS SAR imagery using computer-based algorithm. In *IGARSS'97. 1997 IEEE International Geoscience and Remote Sensing Symposium Proceedings. Remote Sensing - A Scientific Vision for Sustainable Development*, volume 1, pages 151–153 vol.1, 1997.

[16] C. Wackerman, K. Friedman, W. Pichel, P. Clemente-Colón, and X. Li. Automatic Detection of Ships in RADARSAT-1 SAR Imagery. *Canadian Journal of Remote Sensing*, 27(5):568–577, 2001.

[17] Y. Ji, J. Zhang, J. Meng, and X. Zhang. A new CFAR ship target detection method in SAR imagery. *Acta Oceanologica Sinica*, 29(1):12–16, 2010.

[18] J. Ai, X. Qi, W. Yu, Y. Deng, F. Liu, and L. Shi. A New CFAR Ship Detection Algorithm Based on 2-D Joint Log-Normal Distribution in SAR Images. *IEEE Geoscience and Remote Sensing Letters*, 7(4):806–810, 2010.

[19] W. An, C. Xie, and X. Yuan. An Improved Iterative Censoring Scheme for CFAR Ship Detection With SAR Imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 52(8):4585–4595, 2014.

[20] O. Pappas, A. Achim, and D. Bull. Superpixel-Level CFAR Detectors for Ship Detection in SAR Imagery. *IEEE Geoscience and Remote Sensing Letters*, 15(9):1397–1401, 2018.

[21] X. Wu, V. Kumar, J. R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, S. Y. Philip, et al. Top 10 algorithms in data mining. *Knowledge and information systems*, 14(1): 1–37, 2008.

[22] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[23] H. Eum, J. Bae, C. Yoon, and E. Kim. Ship detection using edge-based segmentation and histogram of oriented gradient with ship size ratio. *The International Journal of Fuzzy Logic and Intelligent Systems*, 15:251–259, 2015.

[24] K. Sambodo. Semi-automatic ship detection using pi-sar-l2 data. *34th Asian Conference on Remote Sensing 2013, ACRS 2013*, 2:1530–1536, 2013.

[25] Y. Xia, S. Wan, P. Jin, and L. Yue. A novel sea-land segmentation algorithm based on local binary patterns for ship detection. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 7:237–246, 2014.

[26] D. Ciregan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3642–3649, 2012.

[27] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[28] C. P. Schwegmann, W. Kleynhans, B. P. Salmon, L. W. Mdakane, and R. G. V. Meyer. Very deep learning for ship discrimination in Synthetic Aperture Radar imagery. *2016 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*, pages 104–107, 2016.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[30] Y. Liu, M.-h. Zhang, P. Xu, and Z.-w. Guo. SAR ship detection using sea-land segmentation-based convolutional neural network. In *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, pages 1–4, 2017.

[31] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[32] M. Kang, X. Leng, Z. Lin, and K. Ji. A modified faster R-CNN based on CFAR algorithm for SAR ship detection. In *2017 International Workshop on Remote Sensing with Intelligent Processing (RSIP)*, pages 1–4, 2017.

[33] M. Kang, K. Ji, X. Leng, and Z. Lin. Contextual Region-Based Convolutional Neural Network with Multilayer Fusion for SAR Ship Detection. *Remote Sensing*, 9(8), 2017.

[34] J. Li, C. Qu, and J. Shao. Ship detection in SAR images based on an improved faster R-CNN. In *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA)*, pages 1–6, 2017.

[35] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013.

[36] Z. Lin, K. Ji, X. Leng, and G. Kuang. Squeeze and Excitation Rank Faster R-CNN for Ship Detection in SAR Images. *IEEE Geoscience and Remote Sensing Letters*, 16(5):751–755, 2019.

[37] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[38] J. Redmon and A. Farhadi. YOLO9000: Better, Faster, Stronger. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6517–6525, 2017.

[39] J. Redmon and A. Farhadi. YOLOv3: An Incremental Improvement. *ArXiv*, abs/1804.02767, 2018.

[40] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. *ArXiv*, abs/2004.10934, 2020.

[41] G. Jocher, L. Changyu, A. Hogan, L. Yu, changyu98, P. Rai, and T. Sullivan. ultralytics/yolov5: Initial Release. *Zenodo*, 2020.

[42] Z. Deng, H. Sun, S. Zhou, and J. Zhao. Learning Deep Ship Detector in SAR Images From Scratch. *IEEE Transactions on Geoscience and Remote Sensing*, 57(6):4021–4039, 2019.

[43] Y.-L. Chang, A. Anagaw, L. Chang, Y. C. Wang, C.-Y. Hsiao, and W.-H. Lee. Ship Detection Based on YOLOv2 for SAR Imagery. *Remote Sensing*, 11(7), 2019.

[44] T. Zhang, X. Zhang, J. Shi, and S. Wei. Depthwise Separable Convolution Neural Network for High-Speed SAR Ship Detection. *Remote Sensing*, 11(21), 2019.

[45] Z. Long, W. Suyuan, C. Zhongma, F. Jiaqi, Y. Xiaoting, and D. Wei. Lira-YOLO: A lightweight model for ship detection in radar images. *Journal of Systems Engineering and Electronics*, 2020.

[46] J. Jiang, X. Fu, R. Qin, X. Wang, and Z. Ma. High-Speed Lightweight Ship Detection Algorithm Based on YOLO-V4 for Three-Channels RGB SAR Image. *Remote Sensing*, 13(10), 2021.

[47] S. Liu, W. Kong, X. Chen, M. Xu, M. Yasir, L. Zhao, and J. Li. Multi-Scale Ship Detection Algorithm Based on a Lightweight Neural Network for Spaceborne SAR Images. *Remote Sensing*, 14(5), 2022.

[48] J. Pedoeem and R. Huang. YOLO-LITE: A Real-Time Object Detection Algorithm Optimized for Non-GPU Computers. *2018 IEEE International Conference on Big Data (Big Data)*, pages 2503–2510, 2018.

[49] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.

[50] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[51] Y. Wang, C. Wang, H. Zhang, C. Zhang, and Q. Fu. Combing Single Shot Multibox Detector with transfer learning for ship detection using Chinese Gaofen-3 images. In *2017 Progress in Electromagnetics Research Symposium - Fall (PIERS - FALL)*, pages 712–716, 2017.

[52] Y. Wang, C. Wang, and H. Zhang. Combining a single shot multibox detector with transfer learning for ship detection using Sentinel-1 SAR images. *Remote Sensing Letters*, 9:780–788, 2018.

[53] M. Ma, J. Chen, W. Liu, and W. Yang. Ship Classification and Detection Based on CNN Using GF-3 SAR Images. *Remote Sensing*, 10(12), 2018.

[54] L. Jin and G. Liu. An Approach on Image Processing of Deep Learning Based on Improved SSD. *Symmetry*, 13(3), 2021.

[55] N. Ferreira and M. Silveira. Ship Detection in SAR Images Using Convolutional Variational Autoencoders. *IEEE International Geoscience and Remote Sensing Symposium*, page 503–2506, 2020.

[56] P. Dias. Unsupervised Ship Detection in SAR Images using Generative Adversarial Networks. Master's thesis, Instituto Superior Técnico, University of Lisbon, 2020.

[57] S. Wei, X. Zeng, Q. Qu, M. Wang, H. Su, and J. Shi. HRSID: A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation. *IEEE Access*, 8:120234–120254, 2020.

[58] F. Gao, Y. Huo, J. Wang, A. Hussain, and H. Zhou. Anchor-Free SAR Ship Instance Segmentation With Centroid-Distance Based Loss. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14:11352–11371, 2021.

[59] K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu, and C. Xu. Ghostnet: More features from cheap operations. *CoRR*, abs/1911.11907, 2019.

[60] D. Zhao, C. Zhu, J. Qi, X. Qi, Z. Su, and Z. Shi. Synergistic Attention for Ship Instance Segmentation in SAR Images. *Remote Sensing*, 13(21), 2021.

[61] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, 2015. Springer International Publishing.

[62] J. Li, C. Guo, S. Gou, Y. Chen, M. Wang, and J.-W. Chen. Ship Segmentation on High-Resolution SAR Image by a 3D Dilated Multiscale U-Net. In *IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium*, pages 2575–2578, 2020.

[63] Y. Mao, Y. Yang, Z. Ma, M. Li, H. Su, and J. Zhang. Efficient Low-Cost Ship Detection for SAR Imagery Based on Simplified U-Net. *IEEE Access*, 8:69742–69753, 2020.

[64] F. Gu, H. Zhang, C. Wang, and B. Zhang. Weakly supervised ship detection from SAR images based on a three-component CNN-CAM-CRF model. *Journal of Applied Remote Sensing*, 14(2): 026506, 2020.

[65] J. Wang, Z. Wen, Y. Lu, X. Wang, and Q. Pan. Weakly Supervised SAR Ship Segmentation Based on Variational Gaussian G(A)(0) Mixture Model A Learning. In *2020 Chinese Automation Congress (CAC)*, pages 6072–6077, 2020.

[66] N. Otsu. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[67] Z. Cui, Q. Li, Z. Cao, and N. Liu. Dense Attention Pyramid Networks for Multi-Scale Ship Detection in SAR Images. *IEEE Transactions on Geoscience and Remote Sensing*, 57(11):8983–8997, 2019.

[68] C. Chen, C. He, C. Hu, H. Pei, and L. Jiao. A Deep Neural Network Based on an Attention Mechanism for SAR Ship Detection in Multiscale and Complex Scenarios. *IEEE Access*, 7:104848–104863, 2019.

[69] KwonHyeongjun, JeongSomi, KimSungTai, LeeJaeseok, and SohnKwanghoon. Deep-learning based SAR Ship Detection with Generative Data Augmentation. *Journal of Korea Multimedia Societ*, 25(1):1–9, 2022.

[70] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[71] Zou, Lichuan and Zhang, Hong and Wang, Chao and Wu, Fan and Gu, Feng. MW-ACGAN: Generating Multiscale High-Resolution SAR Images for Ship Detection. *Sensors*, 20(22), 2020.

[72] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35(1):53–65, 2018.

[73] T. Zhang, X. Zhang, J. Shi, S. Wei, J. Wang, J. Li, H. Su, and Y. Zhou. Balance Scene Learning Mechanism for Offshore and Inshore Ship Detection in SAR Images. *IEEE Geoscience and Remote Sensing Letters*, 19:1–5, 2022.

[74] Y. Shi, L. Du, Y. Guo, and Y. Du. Unsupervised Domain Adaptation Based on Progressive Transfer for Ship Detection: From Optical to SAR Images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–17, 2022.

[75] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, 2017.

[76] J. C. Curlander and R. N. McDonough. *Synthetic Aperture Radar : Systems and Signal Processing*. New York: Wiley, 1991.

[77] R. Ryerson, F. Henderson, S. Morain, A. Lewis, A. Budge, A. S. for Photogrammetry, R. Sensing, A. Rencz, and S. Ustin. *Manual of Remote Sensing: Principles and applications of imaging radar*. Number vol. 1-6 in Manual of Remote Sensing. J. Wiley, 1998.

[78] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou. A tutorial on synthetic aperture radar. *IEEE Geoscience and Remote Sensing Magazine*, 1(1):6–43, 2013.

[79] Alaska Satellite Facility. What is SAR? `https://asf.alaska.edu/information/sar-information/what-is-sar/`, 2021. Accessed: 2022-06-29.

[80] K. Ouchi. Current Status on Vessel Detection and Classification by Synthetic Aperture Radar for Maritime Security and Safety. In *The 38th Symposium on Remote Sensing for Environmental Sciences*, pages 5–12, 2016.

[81] R. Wu. Two-Parameter CFAR Ship Detection Algorithm Based on Rayleigh Distribution in SAR Images. *Preprints*, 2021.

[82] R. Cong, J. Lei, H. Fu, M.-M. Cheng, W. Lin, and Q. Huang. Review of Visual Saliency Detection With Comprehensive Information. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(10):2941–2959, 2019.

[83] X. Hou and L. Zhang. Saliency Detection: A Spectral Residual Approach. *IEEE Conference in Computer Vision and Pattern Recognition*, 2007.

[84] D. L. Ruderman. The statistics of natural images. *Network: Computation In Neural Systems*, 5: 517–548, 1994.

[85] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. `http://www.deeplearningbook.org`.

[86] F. Chollet. *Deep Learning with Python*. Manning, 2017.

[87] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *CoRR*, abs/1411.4038, 2014.

[88] N. Siddique, S. Paheding, C. P. Elkin, and V. Devabhaktuni. U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. *IEEE Access*, 9:82031–82057, 2021.

[89] A. Tewari. U-Net architecture. `https://iq.opengenus.org/u-net/`, 2021. Accessed: 2022-07-15.

[90] A. Radford, L. Metz, and S. Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In *arXiv preprint arXiv:1511.06434*, 2015.

[91] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *CoRR*, abs/1611.07004, 2016.

[92] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised Cross-Domain Image Generation. *CoRR*, abs/1611.02200, 2016.

[93] Z. Qingjun. System Design and Key Technologies of the GF-3 Satellite. *Acta Geodaetica et Cartographica Sinica*, 46(3):269, 2017.

[94] E. S. A. (ESA). Sentinel-1 User Handbook. `https://sentinels.copernicus.eu/documents/247904/685163/Sentinel-1_User_Handbook`, 2021. Accessed: 2022-07-17.

[95] M. Z. Alom, T. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. Nasrin, M. Hasan, B. Essen, A. Awwal, and V. Asari. A State-of-the-Art Survey on Deep Learning Theory and Architectures. *Electronics*, 8:292, 2019.

[96] C. E. Shannon. A Mathematical Theory of Communication. *The Bell System Technical Journal*, 27:379–423, 1948.

[97] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[98] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[99] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[100] E. Castro, J. S. Cardoso, and J. C. Pereira. Elastic deformations for data augmentation in breast cancer mass detection. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 230–234. IEEE, 2018.

[101] C. Li, K. Sohn, J. Yoon, and T. Pfister. CutPaste: Self-Supervised Learning for Anomaly Detection and Localization. *CoRR*, abs/2104.04015, 2021.

[102] S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. Yu. scikit-image: image processing in Python. *PeerJ*, 2:e453, 2014.

[103] G. Van Rossum and F. L. Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[104] F. Chollet et al. Keras, 2015. URL `https://github.com/fchollet/keras`.

[105] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[106] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from Simulated and Unsupervised Images through Adversarial Training. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2242–2251, 2017.

[107] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. In Y. Bengio and Y. LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[108] M. Tardy and D. Mateus. Lightweight u-net for high-resolution breast imaging. *CoRR*, abs/2011.13698, 2020.

[109] E. Sari. GitHub - eyyub/tensorflow-cyclegan: Lightweight CycleGAN tensorflow implementation. `https://github.com/Eyyub/tensorflow-cyclegan`, 2021. Accessed: 2022-08-22.

Appendix A

Annotation

Figure A.1 depicts a flowchart of the framework adopted to annotate the SAR ship images from the SAR-Ship-Dataset with the ship's segmentation. First, the input SAR image goes through Gaussian blur followed by a fixed threshold method. Pixels inside the original bounding box with values above the threshold are set as ship, and pixels with values below the threshold are set as background. Then, the SAR image with the segmentation drawn is visually analysed. If the result seems satisfactory, the image with the corresponding segmentation is saved. If the result is not satisfactory, the size of the Gaussian blur filter kernel, the threshold value, or the size of the original bounding box can be changed by the reviewer. The size of the bounding box occasionally needs to be changed due to poor annotation by the authors of the dataset. This process is done iteratively for each image of the training set (for the supervised method) and for the test set (for testing the models) until an accurate segmentation is obtained.

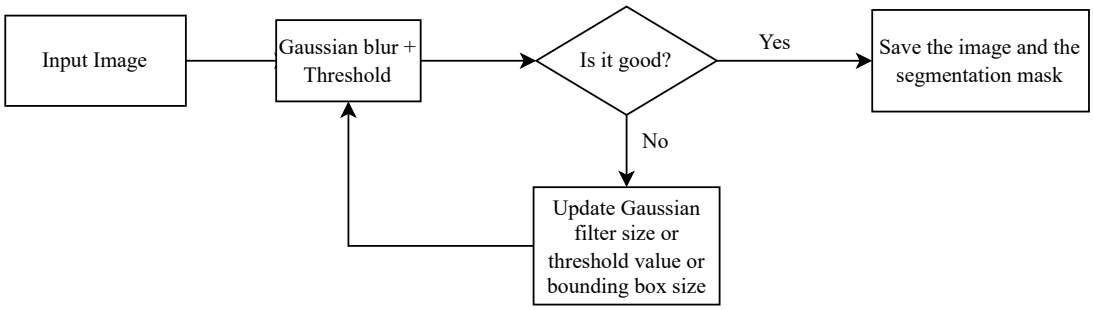


Figure A.1: Flowchart of the method used to annotate the SAR ship images with the ship's segmentation. The initial size of the gaussian blur kernel is 11x11 and the initial threshold value is 70 (with 0-255 pixel intensity).

Appendix B

SAR images generated by the CycleGAN

Figure B.1 depicts some examples of generated images from the SAR domain when using the binary masks as input.

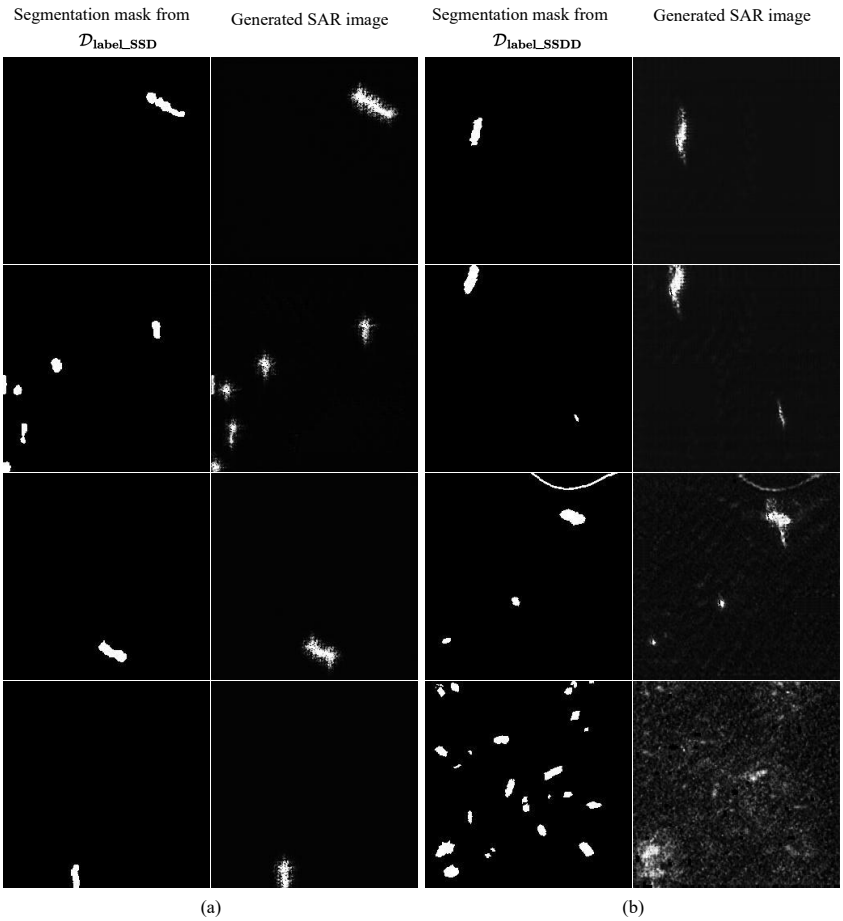


Figure B.1: SAR images generated from the CycleGAN generator $G_{L\_to\_SAR}$. The $\mathcal{D}_{label\_SSD}$ and D_{label_SSDD} are used as input. (a) SAR-Ship-Dataset, (b) SSDD