8-2024

Fishing Vessel Detection in Exclusive Economic Zones From Low Earth Orbit Satellites With Power and Computational Constraints

Kyler E. Nelson
Utah State University

FISHING VESSEL DETECTION IN EXCLUSIVE ECONOMIC ZONES FROM LOW EARTH ORBIT SATELLITES WITH POWER AND COMPUTATIONAL CONSTRAINTS

by

Kyler E. Nelson

A thesis submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Science

Approved:

Mario Harper, Ph.D. Major Professor	Vladimir Kulyukin, Ph.D. Committee Member
John Edwards, Ph.D. Committee Member	D. Richard Cutler, Ph.D. Vice Provost of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2024

ABSTRACT

Fishing Vessel Detection in Exclusive Economic Zones from Low Earth Orbit Satellites with Power and Computational Constraints

by

Kyler E. Nelson, Master of Science

Utah State University, 2024

Major Professor: Mario Harper, Ph.D.
Department: Computer Science

Illegal fishing activities pose a significant threat to the sustainability of marine ecosystems and to economies and societies that are reliant on such resources. Automatic Information System (AIS) signals are vital in tracking and understanding locations of ships; many vessels engaged in illegal fishing disable this AIS signalling. Optical satellite imagery provides a potential solution that could be deployed on resource-constrained, low earth orbit satellites. Due to datasets for fishing vessel detection from optical satellite imagery are scarce, making development of an effective detection model difficult. This research explores the design of an intelligent decision-making system to respond to illegal fishing activity and proposes a new dataset augmentation technique to improve performance of the critical fishing vessel detection component of this system. The proposed decision-making system will use ship location data from a low-power optical imaging satellite and shore-based ship detection systems for identification of illegal fishing activity.

(60 pages)

PUBLIC ABSTRACT

Fishing Vessel Detection in Exclusive Economic Zones from Low Earth Orbit Satellites with Power and Computational Constraints

Kyler E. Nelson

Illegal fishing activities pose a significant threat to the sustainability of marine ecosystems and the economies and societies which rely on them. Detection of fishing vessels engaging in illegal activity is difficult, as many ships engaging in such activity actively avoid detection through radio systems used for maritime traffic monitoring. Satellite imagery provides a promising means for detecting fishing vessels, though designing an effective system is difficult due to limited availability of labeled image datasets of fishing vessels. This research proposes a system to detect illegal fishing activity through the use of a low-power ship detection satellite and proposes a decision-making process capable of analyzing ship movements and location to determine if a ship is engaging in illegal fishing. To overcome current limitations in available image datasets with labeled fishing vessels, this research proposes a new method for expanding existing image datasets to improve ship detection performance. The proposed illegal fishing detection system is further designed to be capable of weighing various social, economic, and ecological factors involved when determining a response to an incident of illegal fishing.

ACKNOWLEDGMENTS

I would like to express my deepest appreciation to my advisor, Dr. Mario Harper, for his invaluable expertise and support throughout the research process, as well for the late-night editing sessions and genuine interest in myself and my research. I am also extremely grateful to my committee members, Dr. Vladimir Kulyukin and Dr. John Edwards, for their thoughtful feedback and insight.

Many thanks are due to my collaborators from Utah State University's LACE-C3A research, Ben Lewis and Shawn Jones, who provided helpful insight in understanding the challenges inherent to space-based deployments of machine learning. I am also grateful to my friends and fellow students in the Utah State University DIRECTLab, for their helpful suggestions and close camaraderie.

Lastly, I would be remiss in not mentioning the support I received from friends and family, especially my wife, as she tended to our newly growing family during the many long hours required throughout the research process. The interest of friends and family in my research, as well as their unwavering support, kept me motivated throughout this work.

Kyler E. Nelson

CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

ACRONYMS

AIS	Automatic Identification System
BCE	Binary Cross-Entropy
CNN	Convolutional Neural Network
CPU	Central Processing Unit
EEZ	Exclusive Economic Zone
FPGA	Field Programmable Gate Array
FPS	Frames Per Second
GPU	Graphics Processing Unit
HBB	Horizontal Bounding Box
LACE-C3A	Low-power Array for CubeSat Edge Computing Architecture, Algorithms and Applications
LEO	Low Earth Orbit
MPA	Marine Protected Area
MSS	Maritime Surveillance System
NPU	Neural Processing Unit
OBB	Oriented Bounding Box
ONNX	Open Neural Network Exchange
RL	Reinforcement Learning
SAR	Synthetic Aperture Radar
SDK	Software Development Kit
SoC	System on a Chip
SODIMM	Small Outline Dual In-line Memory Module
YOLO	You Only Look Once

CHAPTER 1
INTRODUCTION

Combating illegal fishing requires addressing complex challenges across technological, economic, and social realms. This thesis outlines the development of an intelligent, autonomous system designed to efficiently collect data and formulate responses from Low-Earth Orbit (LEO) observation to combat illegal fishing activities. By breaking down the problem into data acquisition and response formulation, the thesis seeks to enhance the practicality and cost-effectiveness of monitoring systems, ensuring they operate within computational constraints to protect marine ecosystems effectively.

1.1 Motivation

Illegal fishing activity is a significant concern worldwide, impacting economies, ecosystems, food supply chains, and international relations [1]. Estimates as of 2020 predict current global economic losses due to illegal fishing to range from $26 bn to $58 bn USD annually [2], a significant increase from a 2009 estimate of $10 bn to $23.5 bn annually [3]. Ecological impacts of illegal fishing are often broad and varied, though one of the most visible and well-understood impacts is the reduced sustainability of a fish population within a fishery [1], or in the potential damage to at-risk marine populations within designated marine protected areas [4].

Illegal fishing is conducted both by vessels registered within the country controlling an EEZ, as well as by vessels registered within another country [1, 2, 5]. Offenses of illegal fishing include fishing in MPAs, foreign vessels operating within another country's EEZ without authorization, and fishing beyond specified quotas, among other offsense [1].

Many ships engaging in illegal fishing turn off or manipulate AIS and radio transponders to avoid detection [6, 7]. Remote sensing technologies using satellite imagery can provide an effective means for the detection of marine vessels that do not transmit Automatic Identification System (AIS) messages [8,9]. Deployment of satellite-based ship detection systems can aid countries in combating illegal fishing occurring within their Exclusive Economic Zone (EEZ). Additional data streams, such as surface radar, can be used for effective detection of marine vessels which disable AIS [10], though such systems may not provide sufficient data on their own due to limitations such as an insufficient range to monitor an entire EEZ, potential signal interference, or limitations on the types of information provided about detected ships (e.g. lack of ship type).

Locating fishing vessels using systems such as AIS and remote sensing satellites is only one component of identifying illegal fishing activity for the purpose of determining a response based on real-time ship movements. Analysis of ship movements and behaviors aids in gaining a stronger understanding of the extent of illegal fishing in an area of interest, as demonstrated by the methods used in [4,11]. However, analysis of the behavior of fishing vessels across a large region, where motivations for illegal fishing activity can greatly vary, is presently a difficult and time-consuming process. Further research is needed to determine a method of analyzing real-time fishing vessel location data to detect instances of illegal fishing in order to enable quick response efforts to combat illegal fishing while such an operation is still ongoing. Furthermore, once illegal fishing activity is identified, it is necessary to weigh the potential economic, social, and ecological costs of the illegal fishing activity against the costs incurred to dispatching a deterrence response.

1.2 Objectives

This thesis will outline the architecture of an intelligent Maritime Surveillance System (MSS) composed of two main components: a remote sensing element consisting of a low-power, Low Earth Orbit (LEO) satellite system tasked with detecting ships, and a ground system running a decision-making algorithm that analyzes ship location data from the satellite to assess whether a fishing vessel may be engaging in illegal activities. The architecture of this system is detailed in Chapter 2. Through the analysis of ship location data, relevant socioeconomic and ecological factors (as discussed in Section 1.1), and consideration of input data uncertainties, the decision-making algorithm can recommend whether it is worthwhile to deploy defense resources, such as coast guard patrols, for further investigation and potential intervention. This structured approach enables efficient allocation of resources for monitoring and combating illegal fishing.

In order for the MSS to perform effectively, the input data provided to the decision-making algorithm must sufficiently represent the real-world conditions and factors that the algorithm must consider. Two of the most critical inputs to this decision-making algorithm are the location of a ship and the type of ship. Additional metrics for detected ships, such as the size and heading would also be of particular interest for the response algorithm. Obtaining these valuable data points on vessels that are attempting to avoid detection through typical broadcast means, such as AIS or weather radar, is particularly difficult and would rely heavily on the ship data collected by a remote sensing satellite.

Due to the importance of fishing vessel detection performance to the MSS, this thesis will propose the use of a lightweight object detection model for ship detection from overhead optical satellite imagery. This model will be deployed onboard a low-cost, low-power remote sensing satellite platform, such as a CubeSat, which imposes very strict operational power constraints on the ship detection model. Limited dataset availability for training optical ship detection models presents a significant challenge for the achievement of high fishing vessel detection performance under such operational constraints. This thesis proposes a dataset augmentation method to improve the performance of the ship detection model within the difficult constraints of low-cost, low-power satellite platforms. The ship detection model and augmentation method proposed to accomplish the task of fishing vessel detection are discussed in Chapter 3. Ship location information from the satellite will be provided to the decision-making algorithm, which will use this location data to detect illegal fishing activity and inform response decisions.

The satellite constraints that the proposed ship detection model is designed to work within are a result of collaboration with ongoing research with Utah State University's LACE-C3A (Low-power Array for CubeSat Edge Computing Architecture, Algorithms and Applications) mission, which is researching micro-satellite hardware designs and compute architectures. A use case of interest for the LACE-C3A mission is the detection of fishing vessels within the exclusive economic zone of Brazil. The use of low-power micro-satellites for ship detection would enable the MSS to be deployed within preferred cost and performance parameters specified by the government of Brazil, which seeks to utilize the proposed MSS to combat illegal fishing activity Designing the MSS to operate within constraints of the micro-satellite platform for the LACE-C3A mission provides a more cost-effective and practical deployment solution for fishing vessel detection satellites.

This thesis will conclude with the discussion of how the fishing vessel detection model will be implemented into the proposed MSS architecture and how this model will interact with the MSS decision-making agent responsible for identifying and responding to illegal fishing activity. Avenues of future research for the ship detection model training method and the MSS will also be discussed in the conclusion of this thesis.

1.3 Document Outline

The remainder of this thesis is organized as follows: In Chapter 2, the design and architecture of an intelligent Maritime Surveillance System (MSS) for illegal fishing detection is presented. Chapter 3 presents a paper that proposes and evaluates a dataset augmentation method to enhance the fishing vessel detection performance of an optical ship detection model that is well-suited for deployment on the low-power remote sensing satellite used in the MSS. Chapter 4 presents a summary of our findings and their implications for the design and future research of intelligent maritime surveillance systems aimed at detecting and responding to illegal fishing activity.

CHAPTER 2

PROPOSED MARITIME SURVEILLANCE SYSTEM ARCHITECTURE

The proposed Maritime Surveillance System (MSS) for illegal fishing detection consists of two primary components, detailed in Fig. 2.1: a low-power ship detection satellite designed to identify fishing vessels within a region of interest, and a ground system that will receive information about detected fishing vessels to determine whether detected vessels are engaged in illegal fishing activity. The ground system will additionally weigh various factors of interest, such as the location of suspected incidents of illegal fishing, locations of response vessels, cost of response, etc. in order to determine whether a response should be dispatched to a suspected incident of illegal fishing activities. Furthermore, the ground system will incorporate information from relevant ground and shore-based sensing systems to augment data transmitted from the ship detection satellite.

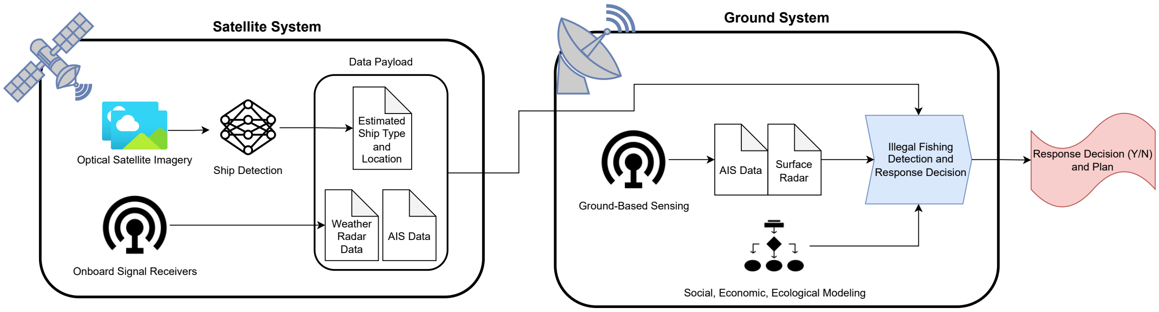

Satellite System
Data Payload
Optical Satellite Imagery
Ship Detection
Estimated Ship Type and Location
Weather Radar Data
AIS Data
Onboard Signal Receivers
Ground System
Ground-Based Sensing
AIS Data
Surface Radar
Illegal Fishing Detection and Response Decision
Social, Economic, Ecological Modeling
Response Decision (Y/N) and Plan

Fig. 2.1: Architecture of the proposed Maritime Surveillance System

One key design decision for the proposed MSS is that ship detection and classification tasks are performed onboard the ship detection satellite instead of transmitting captured images to the ground system for processing. Though this design imposes significant constraints to the compute power available to the ship detection system, strict communication and bandwidth constraints of low-power satellites, combined with the requirement of high resolution imagery for ship detection, make performing ship detection on the ground system infeasible.

2.1 Ship Detection Satellite

The satellite platform that the ship detection model targets is based on a proposed CubeSat platform design from Utah State University's ongoing LACE-C3A (Low-power Array for CubeSat Edge Computing Architecture, Algorithms and Applications) mission research. Development of the LACE-C3A mission satellite's systems and compute platform is ongoing, though two critical constraints have been identified:

1. Ship detection compute hardware must operate within a maximum power budget of 10 W.
2. The ship detection model must process images and transmit results to the ground system within a minimum of 10–15 minutes following image collection.

Performing ship detection within the 10 W power constraint is crucial for deployment onboard satellite platforms such as LACE-C3A due to the limited power generation capacity of typical CubeSat solar arrays. The 10–15 minute image processing time deadline is a result of bandwidth limitations, as well as time and power constraints due to the orbital characteristics of the LACE-C3A satellite. Additionally, the satellite must transmit all observation results to the ground system before it passes into Earth's shadow, as the satellite must power down non-essential systems, including the ship detection system, to conserve power for maintaining orbit and receiving critical flight commands due to the limited power generation capabilities of the solar array on the satellite.

The satellite will be deployed into Low Earth Orbit (LEO), which is optimal for satellite imagery tasks on such small platforms. For the target orbital elevation of the LACE-C3A mission satellite, an orbital period of approximately 90 min is expected. With the assumption of an orbit which directly passes the area to observe, the satellite is expected to have approximately 10 min–20 min per orbit for imaging the region of interest, leaving a maximum of 25 min for data processing and transmission to the ground system before passing into Earth's shadow. This estimate is based on ideal conditions as factors such as orbital decay, orbital drift, time of day, and length of day will often result in tighter time constraints, leading to the design target of a maximum image processing time of 10 min–15 min.

The LACE-C3A mission satellite platform additionally aims to deploy the ship detection model for execution on a PolarFire MPF300T FPGA if the ship detection model can operate within its constraints.

In addition to a compute platform for running a ship detection model, the satellite will be equipped with an AIS receiver and weather radar signal detection payload to aid in the detection of ships.

Satellite imagery will be provided through an optical camera system equipped on the satellite. Though ship detection using synthetic aperture radar (SAR) technology is more well-studied for identification of vessels that are attempting to avoid detection [7, 9, 12–15], power and physical satellite size constraints for the LACE-C3A mission platform prevent the inclusion of SAR equipment. Additionally, SAR equipment is more difficult to add to a satellite payload, while optical imaging payloads are often less expensive and simpler to deploy, providing a more viable payload for constrained satellites.

2.1.1 Benefit of Onboard Ship Detection over Ground System Ship Detection

Bandwidth limitations and image quality requirements for effective ship detection models prevent the use of an MSS design where images are transmitted to the ground system for ship detection, instead of performing ship detection onboard the satellite. Bandwidth limitations prevent transmission of all collected images before the satellite passes into Earth's shadow.

The satellite will communicate with the ground system over the Iridium network, which offers a maximum of 352 kbit/s of available transmit bandwidth from the satellite through the Iridium Certus 700 service [16]. With the use of lossless PNG compression, a single 12 MP image at a size of 7 MB would require approximately 20 s to transmit to the ground system, assuming full transmit bandwidth is available on the satellite. For a single 1 km-wide observation pass over an EEZ—which extends 200 nautical miles from a country's shoreline (approx. 370 km)—using an imaging resolution of $\frac{1}{3}$m/px (meters per pixel) would require transmission of 278 images to the ground system. This transmission process would require 92.6 min, exceeding the orbital period of the satellite. Furthermore, the satellite will need to image an area with an observation pass wider than 1 km, bandwidth must be shared with other operational systems on the satellite, an efficient lossless compression implementation may not be feasible for the satellite platform, and a significant amount of system memory may be required to buffer images awaiting transmission.

Reducing image quality through lossy compression or by making the spatial resolution more coarse could result in a significant detriment to the performance of the MSS as accurate optical ship detection models, such as those leveraged by the proposed MSS, require high-quality images at fine spatial resolutions.

One additional advantage to onboard ship detection can be found in the potential for enabling real-time ship detection results to be provided to the MSS, as detection result data consists of very little data, in contrast to full images. If inference speed is sufficiently fast, such that images can be processed shortly after capture, total memory requirements for the satellite platform would be significantly lower due to the elimination of the need to buffer many images in memory for transmission.

2.2 Ship Detection Model

The ship detection algorithm will be developed to operate within the constraints imposed by the target satellite platform described in Section 2.1. The purpose of the ship detection system is to identify, classify, and locate fishing vessels that are not transmitting AIS or other signals, such as weather radar, which will also be monitored by the ship detection satellite. This task of identifying and classifying ships that do not transmit these signals will be carried out by a deep-learning object detection computer vision model. The ship location and approximate size can be estimated from the object bounding box when combined with information about the geographical location and spatial resolution data associated with satellite images. Estimates of vessel heading and velocity can be obtained through wake detection and analysis [17] and examination of the change in the location of a ship over a period of time.

Data transmissions to the ground system will include information received from AIS transmissions, as well as whether a detected ship is transmitting AIS signals. Weather radar signals will also be monitored, as ships that are not transmitting AIS may be equipped with weather radar equipment.

Considering that deployment of the ship detection model to a more resource-constrained FPGA platform is one of the ideal goals of the LACE-C3A mission, the selected object detection model should be one that can be deployed to an FPGA should sufficient performance be available within these constraints.

2.2.1 Selection of Object Detection System

The YOLO family of object detection models is notable for being efficient, lightweight, and performant. These characteristics make YOLO models an excellent fit for deployment to resource-constrained edge computing platforms.

YOLO models follow a high-level architecture of a single-shot detection model coupled with a Convolutional Neural Network (CNN) backbone for feature extraction. As a single-shot model, YOLO eschews the use of separate detection and classification networks used in multi-shot models, instead using a single network. Control over the learning of object detection components, such as classification and bounding box precision, is accomplished primarily though the design of the model's loss function [18].

The YOLO architecture has continued to receive active research and development while remaining lightweight and performant, making the YOLO family of models an ideal choice for the LACE-C3A mission satellite platform's edge compute needs. The research in this thesis will focus particularly on YOLOv5 [19], released in 2020, and YOLOv8 [20], released in 2023.

Major Architectural Changes to YOLO: YOLOv2 improved on the original YOLO model architecture in many ways to address multiple shortcomings that caused the original YOLO to lag behind the then-current state-of-the-art object detection models in terms of detection performance [21]. YOLOv5 and YOLOv8 share similarities in overall network structure to each other, as both YOLOv8 and YOLOv5 leverage a modified CSPDarknet53 backbone (CSPDarknet53 was first introduced with YOLOv4 [22]). The most significant differences between YOLOv5 and YOLOv8 are found in the change to a decoupled network head and in the switch to anchor-free bounding-box predictions in YOLOv8 [23]. YOLOv8 and YOLOv5 both leverage a block based on Feature Pyramid Networks, first added in YOLOv3 [24]. Feature Pyramid Networks, first described in [25], provide the benefit of improving detection performance of objects across different sizes and scales within an image.

Comparison of YOLO Model Sizes: Table 2.1 provides a comparison of the sizes of different variants of YOLOv5 and YOLOv8, as measured by the number of network parameters in each model. Though these models are similar in many respects, several differences in model sizes are of interest. For model size variants which are medium and smaller of YOLOv5 and YOLOv8 (i.e. nano, small, and medium), YOLOv8 has a larger number of network parameters than YOLOv5; while for the model size variants larger than medium (i.e. large and x-large), YOLOv8 has fewer network parameters than YOLOv5 [19, 20].

A more in-depth study of the changes in each version of YOLO from the original YOLO model to YOLOv8 can be found in [23].

Table 2.1: Model Sizes for YOLO Models

Model	**Parameter Count (Million)**
YOLOv5n [19]	1.5
YOLOv5s [19]	7.2
YOLOv8n [20]	3.2

Further discussion and comparisons of YOLO models in this thesis will be limited to YOLOv5 and YOLOv8.

2.2.2 Ship Detection Model Compute Platform

To deploy the ship detection model within the requirements outlined in Section 2.1, two potential compute platforms have been identified as suitable candidates: the NVIDIA Jetson Orin Nano and the PolarFire MPF300T FPGA. Each of these platform options comes with a set of trade-offs, of which the trade-offs that are most relevant to the design of the ship detection system's object detection model will be discussed in this section. Most recent versions of YOLO (e.g. YOLOv5, YOLOv8, etc.) are built using PyTorch [26], which provides a powerful and efficient framework for training deep learning models using gradient-based optimization methods. As the YOLO family of models are designed for object detection based on the MS-COCO dataset [18–21, 24], YOLO models that are to be evaluated for ship detection and classification will require training on an appropriate dataset. Additionally, the ship detection compute platform will need to consume the model produced from the PyTorch training process in order to run the needed inference tasks for ship detection.

NVIDIA Jetson Orin Nano

The NVIDIA Jetson Orin Nano is a power-efficient embedded compute module based on a low-power version of NVIDIA's Orin SoC, equipped with a 6-core ARM Cortex-A78AE CPU and a 1024-core NVIDIA Ampere architecture GPU with 32 Tensor Cores (for the 8GB memory configuration) [27]. The Jetson Orin Nano module is built in a SODIMM form factor and supports two operating power profiles of 15 W and 7 W for ease of integration in various applications. The combination of the Ampere GPU and support for NVIDIA's TensorRT runtime, combined with sufficient memory and the 7 W operating power profile, the Jetson Orin Nano is a highly performant edge compute platform capable of working within the power constraints and required compute performance requirements of the ship detection satellite. Additionally, PyTorch models are straightforward to convert to TensorRT engines by providing an ONNX export of the PyTorch model to the TensorRT tool set. The primary challenge involved with leveraging a Jetson Orin Nano onboard the ship detection satellite would be in properly equipping the satellite hardware with the needed shielding and protection to prevent instability of the platform when deployed in orbit.

The Jetson Orin Nano is capable of running even the largest YOLOv8 model, YOLOv8x, with a total of 68.2 million parameters [20] at 15 Frames Per Second (FPS), based on our test results described in Table 2.2 when configured with its maximum power profile of 15 W and running model inference using the NVIDIA TensorRT inference engine. Since the LACE-C3A satellite has only a 10 W power budget for ship detection compute, we benchmarked all sizes of YOLOv8 on this hardware to better understand its performance characteristics. To benchmark YOLOv8, we exported each model to the TensorRT model format, with optimizations enabled. Benchmarking was done using the standard MS-COCO weights and on the benchmark image included with the Ultralytics package [20]. The inference speed results detailed in Tables 2.2 and 2.3 were averaged over 20 runs, after model warm-up.

Low-Power FPGAs

Low-power FPGA-based solutions provide an attractive alternative to GPU, CPU, and Tensor Processor Unit (TPU)-based compute platforms for neural networks, particularly in the case of power-constrained edge compute uses where power efficiency and good performance are both priorities [28]. The PolarFire series of FPGAs, such as the MPF300T which the LACE-C3A project aims to use, are notable for their low-power and efficient operation [29]. PolarFire FPGAs are also available in radiation-tolerant models [30], which makes the PolarFire platform an ideal choice for the LACE-C3A satellite if the ship detection model can achieve sufficient performance when deployed to this platform. This FPGA deployment would reduce overall payload power consumption, though it may reduce model inference in contrast to an implementation utilizing a larger FPGA, GPU, NPU, or CPU-based platform.

An overlay-based approach for executing the ship detection model on an FPGA is needed to avoid reprogramming the FPGA while the satellite is in orbit—a high risk

Table 2.2: YOLOv8 performance on NVIDIA Jetson Orin Nano 8GB — 15 W TensorRT

Model	Preprocess (ms)	Inference (ms)	Postprocess (ms)	Total (ms)	FPS
YOLOv8n	8.093059	7.113039	4.520750	19.726849	50.695996
YOLOv8s	8.211410	11.188936	4.587913	23.988259	41.692290
YOLOv8m	8.208847	23.460329	4.658818	36.327994	27.569564
YOLOv8l	8.152449	34.915364	4.846358	47.914171	20.871132
YOLOv8x	8.198047	56.018651	6.894732	71.111429	14.064526

Table 2.3: YOLOv8 performance on NVIDIA Jetson Orin Nano 8GB — 7 W TensorRT

Model	Preprocess (ms)	Inference (ms)	Postprocess (ms)	Total (ms)	FPS
YOLOv8n	12.055993	15.288353	6.386757	33.731103	29.650167
YOLOv8s	12.100554	25.920308	6.389236	44.410098	22.518897
YOLOv8m	12.040007	57.614601	6.835580	76.490188	13.074090
YOLOv8l	12.075269	86.382687	6.844807	105.302763	9.496504
YOLOv8x	12.089908	147.361970	6.883073	166.334951	6.012014

operation—to load a new or modified ship detection model. A reprogramming failure occurring while the satellite is in orbit would significantly disrupt mission activities and runs a substantial risk of placing the satellite compute systems in an unrecoverable state without physical access. The overlay FPGA program and associated model optimization and compilation toolchain could be implemented through the Microchip VectorBlox Software Development Kit (SDK) for PolarFire FPGAs [31] if the capabilities of this SDK and its accompanying FPGA logic core design are sufficient. Otherwise, the LACE-C3A mission will design a custom overlay FPGA program and compilation toolchain capable of consuming an ONNX exported model from PyTorch, such as the design described in [32], if more flexibility is needed.

YOLO on FPGAs: YOLO models, such as YOLOv2, have been successfully deployed on resource-constrained FPGA platforms with a trade-off of slower execution time compared to a less constrained hardware platform [33]. On more powerful FPGA platforms, more recent versions of YOLO (i.e. YOLOv3, YOLOv5, and YOLOv8) can be deployed to FPGA platforms compatible with software toolkits such as AMD Vitis [34] and Intel's FPGA AI Suite [35], both of which contain YOLOv3 in their model zoos and are compatible with model export formats supported by YOLOv5 and v8 [19, 20]. For the PolarFire MPF300T FPGA platform proposed for the LACE-C3A mission satellite, it is estimated that approximately one-third of the available logic units would be utilized for an implementation of YOLOv2 as described by Zhang, et al. [33], and many of the functional blocks used in this implementation would also be utilized in implementations for newer versions of the YOLO family of models. Microchip's VectorBlox SDK for PolarFire FPGAs [31] contains an example implementing YOLOv5, which indicates that the PolarFire FPGA selected for LACE-C3A should be capable of running more recent YOLO models. The most notable challenge of running a more recent version of YOLO on the LACE-C3A mission's FPGA platform lies in the implementation of the major architectural changes that each version of YOLO introduces while continuing to operate within the limited number of FPGA logic units and maintaining inference times within acceptable parameters.

Selection of Compute Platform for Benchmarking

As the design of the compute platform for the LACE-C3A mission is still ongoing, the research presented in this thesis will use an NVIDIA Jetson Orin Nano 8GB Developer Kit as the target compute platform for running and benchmarking the ship detection system.

Though the Jetson Orin Nano platform is capable of running larger versions of YOLO, or other more complex models, this thesis will focus on the YOLO family of models, particularly the smaller variants, to focus on the problem of fishing vessel detection within tight resource constraints. This focus has been chosen as we anticipate that the PolarFire MPF300T FPGA, which the LACE-C3A mission currently proposes to use, may not be able to run larger models within necessary inference performance requirements and other resource constraints, such as available system memory for model weight storage. Presently, we expect that only the nano or small variants of YOLOv5 or YOLOv8 would be implementable within the performance requirements of the satellite's target FPGA platform. For this reason, analyses presented in this thesis will primarily focus on these smaller variants of YOLO.

2.3 Ground System

The ground system will receive detection results and signal readings from the ship detection satellite. These readings will then be fed into an intelligent decision-making agent, which will determine the location of detected ships and the risk that unauthorized fishing vessels pose within the area of interest, which is the Exclusive Economic Zone (EEZ) of the country of Brazil.

To enable the model to balance conflicting priorities, the ground system will also be provided information about response resources, such as coast guard ships, and data points which can be used to estimate the cost of investigating a given fishing vessel. Using this information, the decision-making agent will determine the potential impact that a detected fishing vessel could have, based on location, size, and other metrics, to determine whether the potential benefit of dispatching a response crew to investigate is likely to outweigh the costs involved in such a dispatch.

In order to better handle inaccuracies and uncertainty in the object detection model used for ship detection, as well as the uncertainty inherent in quantifying the potential impact of a suspected case of illegal fishing activity, the decision-making model will use this information about data uncertainty as an input as well.

2.4 Key Research Questions

To determine the feasibility of the proposed Marine Surveillance System, there are three main research questions that must be answered:

1. Can an adequate ship detection model be trained with the currently available datasets?

2. Can a model of sufficient performance be identified or built to operate within the power limits and constraints of the ship detection satellite, as described in Section 2.1?

3. Can the decision-making agent responsible for determining a response to a suspected illegal fishing incident perform effectively when given data points which may contain a significant degree of uncertainty?

2.5 Dataset Limitations

For best results when training an object detection model from scratch, a large dataset is generally required. For example, the YOLOv5 documentation recommends at least 10,000 instances of each class [36]. Unfortunately, datasets of optical satellite imagery with fishing vessels specifically labeled, in contrast to more general categories such as warships, passenger ships, and merchant vessels, are scarce. Constructing a new dataset for the task of fine-grained ship detection is particularly time-consuming and difficult, requiring experts who can identify fine-grained ship types from satellite imagery.

Fortunately, datasets containing fine-grained labeling of ships in optical satellite imagery are publicly available, though the number of fishing vessels labeled in such datasets are often small, such as the ShipRSImageNet dataset compiled by Zhang, et al. [37]. The research in this thesis will explore methods to train an object detector to identify fishing vessels when only very limited data is available.

2.6 Accounting for Ship Detection Model Uncertainty

The decision-making agent will already be subject to measures of uncertainty in its input space, such as the potential impact of a given vessel if it is engaging in illegal fishing. This decision becomes even more uncertain when it is not always known with a high degree of confidence whether or not a given vessel is actually a fishing vessel. If a pattern or distribution can be identified within the classification errors made by the ship detection model, it may be possible to design the decision-making agent to leverage information about the types of errors that the ship detection model makes to minimize the potential impact on the agent's performance.

Reinforcement learning (RL) provides a strong candidate framework for building a capable decision-making algorithm to run on the ground system by training an agent to make illegal fishing incident response decisions. RL methods are particularly useful when working within the inherent uncertainty involved in making decisions based on uncertain input data and sometimes-conflicting priorities as the agent can learn a policy based on the various inputs about the environment provided to it. The inclusion of uncertainty information as part of the decision-making algorithm's input to the agent should allow the agent to better work within, and leverage these uncertainty metrics, to improve its performance in comparison to a simpler decision policy. The action space of the agent is relatively straightforward: the actions are to either dispatch a response to a particular detected vessel, or to take no action. In contrast, the input space is much more complex, and should consist of the ship location data provided by the detection satellite, as well as additional data points that the model should weigh when making a response decision, such as those discussed in Sec 1.1.

CHAPTER 3

TRAINING THE FISHING VESSEL DETECTOR — POSEIDON-SAT[1]

Abstract

This paper introduces POSEIDON-SAT, a novel dataset augmentation method designed to enhance the detection of fishing vessels using optical remote sensing technologies, which is critical for safeguarding important conservation and economic fishing zones from illegal fishing activities. Effective monitoring and detection are often hampered by the vessels' tactics to evade detection, such as disabling or manipulating Automatic Information System (AIS) radio transponders. Although convolutional neural networks (CNNs) are adept at detecting and classifying ships from optical imagery, the fine-grained classification required is severely restricted by the lack of high-quality, detailed datasets, especially for fishing vessels which are underrepresented in existing datasets.

POSEIDON-SAT addresses these challenges by augmenting scarce datasets with synthesized instances of fishing vessels, thereby improving the training and performance of ship detection models. This augmentation is particularly beneficial for lightweight models deployed on low-power, edge computing platforms aboard remote sensing satellites, where computational resources are limited. We compare the efficacy of POSEIDON-SAT with traditional class weighting techniques and evaluate the augmented dataset's impact on the performance of lightweight YOLO models. Our results demonstrate that POSEIDON-SAT significantly enhances model accuracy and reduces the likelihood of false positives, making it an invaluable tool in the global effort to combat illegal fishing through enhanced remote sensing capabilities.

Code for POSEIDON-SAT is publicly available.[2]

[1] This chapter was prepared as a submission to the IEEE Transactions on Intelligent Transportation Systems with the title of *POSEIDON-SAT: Data Enhancement for Optical Fishing Vessel Detection from Low-Cost Satellites* authored by Kyler Nelson and Mario Harper.

[2] Available at https://github.com/Kytech/POSEIDON-SAT

3.1 Introduction

Illegal fishing activity inflicts severe economic damages, with annual global losses estimated between \$26 and \$58 billion USD as of 2020 [2]. This practice not only undermines communities dependent on marine resources but also jeopardizes the sustainability of marine ecosystems [1, 2]. The variety of illegal fishing practices includes overfishing beyond quotas, unauthorized operations within another nation's Exclusive Economic Zone (EEZ), and violations within Marine Protected Areas (MPA) [4].

Effective monitoring of these activities often employs the Automatic Identification System (AIS), a system initially designed for collision avoidance, which has since been adapted to monitor illegal fishing activities [11]. However, AIS is not foolproof; vessels engaged in illegal activities may deactivate AIS, tamper with its signals, or operate without it, escaping detection [6, 7]. A significant percentage, 72–76%, of industrial fishing vessels remain unmonitored [7].

Satellite-based remote sensing has emerged as a potent tool against illegal fishing, particularly in detecting ”dark” vessels that do not transmit AIS signals [8]. Nevertheless, transmitting satellite imagery data remains a costly and bandwidth-intensive task. Moreover, executing ship detection tasks on edge computing devices aboard power-restricted satellites poses significant challenges.

Notably, Convolutional Neural Networks (CNNs) have shown promise in detecting and classifying dark ships using both optical and Synthetic Aperture Radar (SAR) images. Despite this, such models require extensive training datasets to perform effectively. Renowned datasets include DOTA for optical ship detection [38] and xView3-SAR for SAR imagery [13]. These datasets vary widely in the granularity of ship classifications, from basic distinctions between ship and no ship to detailed identifications of specific vessel types. Fine-grained ship detection, which includes identifying fishing vessels, typically requires specialized datasets that are not abundantly available.

The scarcity of detailed optical datasets for fishing vessels significantly hampers the performance of object detection models, especially those designed for lightweight architectures like those in the YOLO family. Given these constraints, developing effective ship detection models for use on low-power satellites such as CubeSats in Low Earth Orbit (LEO) is particularly challenging. Although comprehensive SAR-based datasets are available, the hardware required for SAR is often too cumbersome for small satellites, which makes optical imagery a more practical choice for such platforms.

Cube-satellites equipped with optical remote sensing technologies offer a promising solution to this challenge. Their ability to continuously monitor large swathes of the ocean from low Earth orbit, coupled with advancements in edge computing, allows for real-time data processing and ship detection. This satellite technology can significantly enhance the detection and monitoring of fishing vessels, making it an essential tool in the global effort to combat illegal fishing. The deployment of lightweight YOLO models on these satellites, enhanced by our POSEIDON-SAT dataset augmentation method, aims to maximize detection accuracy while operating within the stringent power and computational constraints of small-form factor satellites.

This paper addresses the scarcity of fishing vessels in optical satellite data by comparing two prevalent methods: dataset augmentation and class weighting within the model's loss function. We apply these techniques to enhance the performance of lightweight detection models, specifically using the ShipRSImageNet dataset. In addition, we assess the classification errors of these models and discuss how integrating data on model uncertainties with the behavioral analytics of ships can forge robust mechanisms for illegal fishing detection. We will evaluate several models from the YOLO family—including YOLOv8 nano, YOLOv5 nano, and YOLOv5 small—chosen for their minimal computational demands suitable for deployment on power-limited edge compute platforms [19, 20]. Finally, we will benchmark the trained models' inference capabilities on an edge compute platform designed to operate under a stringent 10-watt power limit, demonstrating the feasibility of our methods for real-world applications on board a CubeSat.

3.2 Data Considerations

Training a YOLO network for the task of fishing vessel detection relies heavily on available datasets. This section will discuss the availability of both optical and SAR datasets, followed by a discussion of existing methods for enhancing currently available fishing vessel detection datasets.

3.2.1 Ship Classification Datasets

The DOTA dataset, compiled from optical satellite images sourced from Google Earth, stands as a prominent resource for ship detection, containing approximately 2,700 ship instances annotated with Oriented Bounding Boxes (OBBs). However, its utility for specialized tasks such as fishing vessel detection is limited by its broad categorical definition, which lumps all ships under a single label [38]. In contrast, the DIOR dataset significantly expands the available data with 64,000 ship instances, although it shares the same limitation of a singular ship category, necessitating additional label refinement for fine-grained analysis [39].

Among datasets tailored for fine-grained ship detection from optical imagery, HRSC2016, FGSD, xView, and ShipRSImageNet offer more nuanced ship categorizations within a smaller scope, each containing fewer than 20,000 ships [37,40–42]. Notably, only xView and ShipRSImageNet include categories for fishing vessels, with ShipRSImageNet providing a comprehensive refinement of ship labels through enhancements such as the introduction of OBBs and corrections to mislabeled instances [37]. Table 3.1 provides an overview of each fine-grained ship detection dataset.

SAR Datasets

Synthetic Aperture Radar (SAR) is the primary modality for ship detection in remote sensing, primarily due to its robust capability to capture high-resolution imagery independent of lighting conditions and weather, thereby offering considerable advantages over optical imaging. SAR technology excels in detecting metallic objects on the sea's surface, making it exceptionally effective for identifying ships. Due to these advantages, SAR has

Table 3.1: Comparison of Fine-Grained Ship Detection Datasets

Dataset	**# of images**	**Image Size**	**# of Ship Instances**	**# of Ship Categories**	**Fishing Vessel Category?**
HSRC2016 [40]	1,070	300×300 – 1.5k×900	2,976	25	No
FGSD [41]	2,612	930×930	5,634	43	No
xView [42]	1,413	1.5k×1.2k	5,672	9	Yes
ShipRSImageNet [37]	3,435	930×930 – 1.4k×1k	17,573	50	Yes

become the predominant method for conducting maritime surveillance and ship detection tasks, and several datasets exist for public use [13, 43–46].

Many SAR datasets often leverage Automatic Identification System (AIS) signals to augment the detection and classification of ships. This integration of AIS with SAR imagery allows for the correlation of real-time ship movement data with the detected images, significantly enhancing the accuracy of ship categorization. Techniques described in literature [14, 15, 47] detail how AIS data can be correlated with SAR images to identify and classify vessels, including those involved in illicit activities such as illegal fishing. This correlation not only improves the quality of the datasets but also simplifies the labeling process, resulting in larger, more accurate datasets with fewer errors compared to manually annotated optical imagery datasets [37].

Among the SAR datasets, the xView3-SAR dataset stands out due to its comprehensive coverage and detailed categorization of ships, including a substantial number of fishing vessels. With 47,412 fishing vessels out of more than 220,000 ship instances, xView3-SAR provides an unparalleled resource for the detection of dark fishing vessels [13]. This dataset is instrumental in advancing research on the detection of illegal fishing activities, offering detailed insights that are critical for developing robust detection algorithms. The extensive number of fishing vessels in xView3-SAR not only enhances the dataset's utility but also exemplifies the effectiveness of SAR technology combined with AIS data in maritime surveillance.

3.2.2 Dataset Augmentation for Maritime Environments

Ruiz-Ponce and colleagues developed the POSEIDON data augmentation tool, specifically designed for maritime object detection models. This method, when applied to the SeaDronesSee dataset, demonstrated a notable improvement in model performance, achieving a 4.6% increase in detection accuracy using YOLOv8 over traditional class weighting techniques [48]. The SeaDronesSee dataset, which includes various images captured by aerial drones over marine environments from multiple camera angles, shares characteristics with optical satellite imagery, making it a suitable candidate for the POSEIDON augmentation method [37, 49].

The POSEIDON method involves two primary processes: synthetic object instance generation and instance removal. These processes are designed to balance the dataset by adjusting the representation of classes within it.

Instance Generation

The instance generation component of POSEIDON involves four critical steps:

1. **Image Normalization:** This initial step ensures consistency across the dataset by rescaling all images to match the width of the smallest image while maintaining the original aspect ratio. This normalization is crucial for maintaining the integrity and comparability of features across different images.

2. **Object Instance Extraction:** Here, objects are extracted from their original images. This extraction process is necessary for subsequent steps where these objects will be recontextualized within new scenes.

3. **Image Categorization and Selection:** Images and their extracted instances are categorized based on the camera angle at the time of capture. This categorization ensures that the augmentation process respects the original perspective, thereby maintaining the realism and applicability of the augmented data.

4. **Addition of New Objects to Images:** New objects are added to different images, ensuring no overlap with existing objects. This step involves placing extracted instances into new backgrounds, effectively simulating different scenarios, and enriching the dataset's diversity. The placement also ensures that the augmented images remain realistic and useful for training robust detection models.

The augmentation process expands the dataset and enhances its variety and balance, which is crucial for training models to recognize underrepresented classes effectively.

Instance Removal

Conversely, the instance removal process targets the reduction of overrepresented classes to prevent model bias. This is achieved by replacing the pixels within the object's bounding box with black pixels and subsequently removing the associated label information. This method of undersampling helps in balancing the dataset, ensuring that no single class dominates the learning process, thus facilitating a more generalized and effective detection capability.

The integration of these two processes—instance generation and removal—provides a comprehensive approach to dataset augmentation. By enhancing the dataset's diversity and balance, POSEIDON significantly improves the performance of object detection models on complex tasks such as ship detection in varied maritime environments. This methodological innovation is pivotal in advancing the capabilities of satellite and aerial imagery analysis, particularly in the challenging conditions typical of maritime surveillance.

3.3 Proposed Augmentation Method

We choose to augment the ShipRSImageNet dataset using a heavily modified POSEIDON method [48]. Modifications were necessary as the original normalization step in POSEIDON resulted in significant detail loss across the dataset due to the use of down-scaling images to the same maximum width in the original POSEIDON normalization process. The modified POSEIDON-SAT loosely follows the same general structure of POSEIDON and is described in Fig. 3.1.

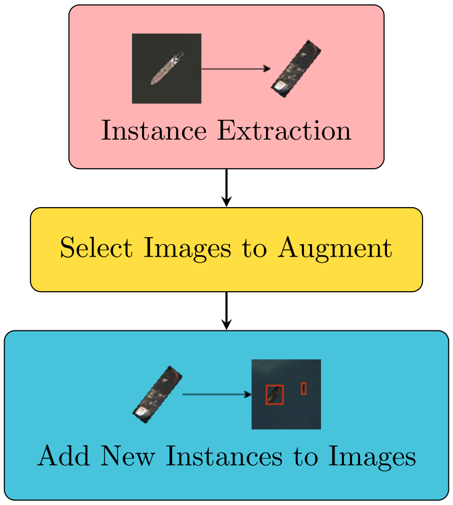


Fig. 3.1: POSEIDON-SAT Augmentation Pipeline

POSEIDON-SAT significantly modifies the processes performed in POSEIDON steps 1, 3, and 4, and makes small changes to step 2. The most significant difference is found in steps 1 and 4. We note that the normalization step from POSEIDON is folded into the augmentation step.

Similar to the original POSEIDON method, augmentation is only performed on the training set and not the validation set.

3.3.1 Instance Extraction

The process of ship instance extraction remains relatively unchanged from the POSEIDON method. The primary difference is the use of both OBBs and Horizontal Bounding Boxes (HBBs), in contrast to POSEIDON's use of only HBBs, which results in a rougher extraction of instances from the original image.

In the instance extraction step, the process involves scanning all dataset images to identify and extract ships of the class targeted for augmentation, such as fishing vessels. Each identified ship is cropped to its horizontal bounding box (HBB), and then further refined by masking out any pixels outside its oriented bounding box (OBB). These extracted ship instances are then saved as individual images, which will be used later in the image augmentation process.

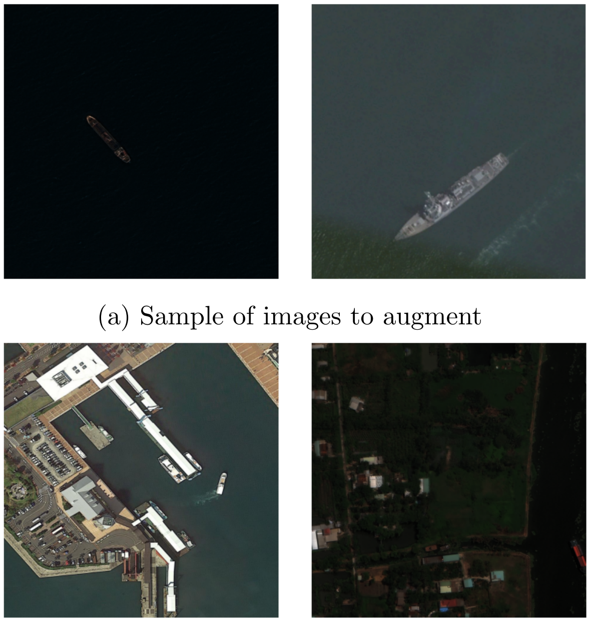
3.3.2 Image Selection

POSEIDON-SAT introduces a refined image selection criterion to identify which images from the ShipRSImageNet dataset are suitable for augmentation with synthetic ship instances. This adjustment is crucial as not all images are conducive to augmentation due to various spatial constraints. These constraints ensure that ships are not erroneously placed on land, shorelines, docks, or overlapping with other ships. The process to prevent new ships from being placed on top of existing ones involves straightforward bounding box collision checks.

However, avoiding placement on shorelines or docks presents a greater challenge since the ShipRSImageNet dataset lacks specific segmentation data for these features. This issue contrasts with the conditions of the SeaDronesSee dataset, for which the POSEIDON method was originally developed. In the SeaDronesSee dataset, shorelines are generally not present in the images, significantly reducing concerns about unrealistic placements of synthetic instances. Only images identified as ships in open water, whose image background was primarily ocean, were selected for augmentation with additional ship instances. Fig. 3.2 illustrates several examples of images that were selected or excluded for augmentation based on this criteria.

3.3.3 Image Augmentation

Once suitable images are selected from the ShipRSImageNet dataset, the augmentation process begins by generating new ships from previously extracted instances. Each selected source ship is rescaled to match the spatial resolution of the target image, expressed in meters per pixel (m/px), and then randomly flipped horizontally and vertically to enhance variety. The ships are positioned randomly within the image, ensuring they do not overlap with existing ships. This process can be repeated to add multiple ships to the same image.



(a) Sample of images to augment

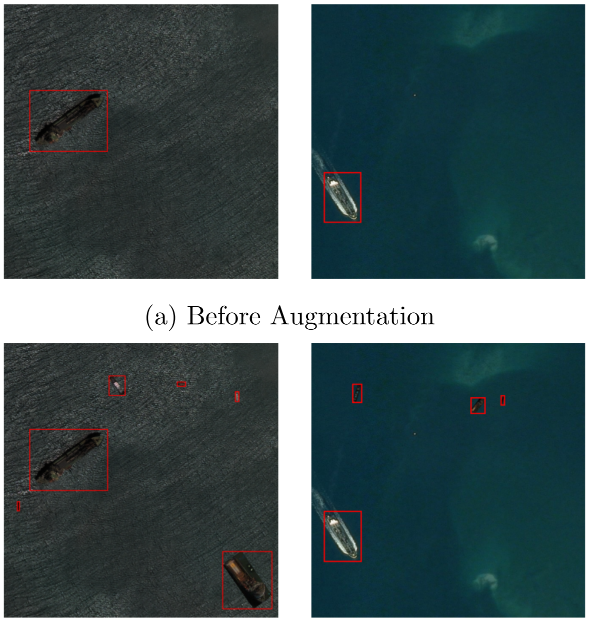
(b) Sample of images excluded from augmentation

Fig. 3.2: Images from ShipRSImageNet selected for and excluded from augmentation

The rescaling step replaces the earlier normalization method used in POSEIDON (Step 1), offering a tailored approach for satellite imagery that maintains the physical size of the ship in the image while preserving detail better than the original POSEIDON's method. If a ship instance comes from an image with the same spatial resolution as the target image, this normalization step is skipped. Most images in the ShipRSImageNet have known resolutions from metadata or can be estimated, like an image from the FGSD dataset with an estimated resolution of 1.025 m/px based on dataset averages [41]. All fishing vessels specifically are sourced from images in the xView dataset, which has a consistent resolution of 0.3 m/px [42].

The random flipping creates diverse orientations, needed for training robust models, as all images depict ships from an overhead perspective at sea. The augmentation process allows for varying the number of ships added per image, with parameters defining the minimum and maximum additions. This iterative augmentation continues until the desired number of ships is integrated into the selected images, effectively enriching the dataset with varied and realistic ship instances.

Fig. 3.3 gives an example of several images before and after augmentation.



(a) Before Augmentation

(b) After Augmentation

Fig. 3.3: Example of two images before and after augmentation with bounding boxes displayed

3.4 Class Weighting

In model training for object classification, especially when dealing with imbalanced datasets, adjusting the weight of different classes in the loss function can significantly enhance performance. This approach is often applied using cross-entropy loss, a common choice for classification tasks, which includes several methods for class weighting suitable for tasks like fishing vessel classification [50–52].

YOLOv8 and YOLOv5 are both leveraged in the classification and use cross-entropy for calculating loss through the use of Binary Cross-Entropy (BCE), also known as log loss. For object classification, a typical use of BCE loss applies a sigmoid activation to network outputs prior to computing the BCE loss on the resultant probabilities produced by the sigmoid activation function. For improved numerical stability, YOLOv5 and YOLOv8 use a modified implementation of BCE loss, provided by the PyTorch framework [53], that combines the sigmoid activation and BCE loss into a single layer. The BCE loss implementation used by these YOLO models additionally introduces a positive class weight parameter, which can be used to adjust the weight of a positive example of a particular object class in the dataset used for training. The BCE loss function ℓ used by YOLOv5 and YOLOv8 is given by:

$$\begin{gathered}\ell(x, y) = R(L), \quad L = \{\ell_1, \ldots, \ell_C\}, \\ \ell_c = -(p_c y_c \cdot \log \sigma(x_c) + (1 - y_c) \cdot \log(1 - \sigma(x_c)))\end{gathered} \tag{3.1}$$

where

C	total number of object classes in the dataset
c	class number, $1 \leq c \leq C$
ℓ_c	loss computed for class number c
x_c	activation value for class label c
y_c	ground truth value for class label c
p_c	weighting factor to apply to objects with class label c
σ	sigmoid activation function
R	a reduction operation to apply over the losses of each class c

By default, p_c is defined as 1 for all classes, effectively applying a sigmoid activation followed by the standard BCE loss. The particular reduction operation R varies between YOLOv5 and YOLOv8. For YOLOv5, the values are averaged across all classes and all instances detected in the minibatch, while YOLOv8 reduces BCE losses in a subsequent step following the BCE loss computation. Setting weighting factors p_c for each class c relative to the number of instances of each class in the dataset ensures that each class has an equal impact on the network's classification loss when training on an imbalanced dataset.

Class weighting was implemented in YOLO v5 and v8 by modifying the BCE loss function used to classify detected objects. The default BCE loss implementation used by both YOLO v8 and v5 follows the form described in (3.1) with the positive class weight p_c set to 1 for every class c, effectively using a non-weighted BCE. Though the YOLO v5 and v8 loss functions are different, the general structure of the classification component remains relatively unchanged.

To apply an equal weight to each class in the loss function in YOLO v5 and v8, the p_c values for each class c are assigned to a value proportional to the size of the largest class in the dataset as described by Algorithm 1. Effectively, positive examples of classes that are fewer in number compared to the largest class are given additional weight in the loss function.

Algorithm 1 Computation of BCE Positive Class Weights P

```
  $X \leftarrow$ list of class labels in the dataset
  $C \leftarrow$ number of classes in the dataset
Require: $\textsc{len}(X) > 0$, $C > 0$
Ensure: $P = \{p_1, \ldots, p_C\},\ \forall p_c \in P, p_c \geq 1$
  $labelCounts \leftarrow \textsc{count\_occurrences}(X)$
  $largestClassSize \leftarrow \textsc{max}(labelCounts)$
  $P \leftarrow$ a list of 1s of length $C$
  for all $p_c$ in $P$ do                                ▷ $c$ is the class number
      if $labelCounts[c] > 0$ then
          $p_c \leftarrow largestClassSize \ / \ labelCounts[c]$
      end if
  end for
```

3.5 Evaluation

To evaluate the proposed dataset augmentation technique POSEIDON-SAT, two augmented versions of the ShipRSImageNet dataset [37] were generated using different parameters and compared. Several ship categories defined in ShipRSImageNet are combined for validation due to their low relevance in fishing vessel detection. The limited availability of source fishing vessel instances guided the decision to avoid increasing the number of generated instances, due to concerns about creating insufficient variety. The POSEIDON-SAT method was evaluated alongside class weighting in the loss function. Experiments were conducted using both POSEIDON-SAT and class weighting to assess whether combining these methods could lead to further improvements in performance.

Table 3.2 describes the set of training experiments conducted to compare the POSEIDON-SAT technique with class weighting, detailing which of the augmentation methods were used, if any, and whether class weighting was used for each experiment. Each training experiment was performed on a YOLOv5 nano, YOLOv5 small, and YOLOv8 nano model. Experiments were repeated on multiple versions of YOLO to test the POSEIDON-SAT method on more than one network architecture. Performance was analyzed across several classification metrics to measure the impact of augmenting the training dataset and/or weighting classes in the loss function during training.

Table 3.2: List of Experiments

Category	**Experiment**	**Augmentation Method**	**Weighted?**
Baselines	Baseline Baseline Weighted	None None	No Yes
Augmented	Augmented MatchRes	Standard Matched Resolution	No No
Augmented +Weighted	Augmented + Weights Aug. MatchRes + Weights	Standard Matched Resolution	Yes Yes

3.5.1 Augmentation Parameters

The augmentation process of POSEIDON-SAT is controlled by several elements: the images selected for augmentation, the minimum and maximum number of ships to add, and the maximum number of ships that can be added to a single image. The image augmentation selection criteria described in Sec. 3.3.2, resulted in the selection of 171 images to augment.

The two augmented versions of the ShipRSImageNet dataset were generated using POSEIDON-SAT for evaluation. The first, ”standard” version does not limit itself to only augmenting images of 0.3 m/px resolution. The second ”matched resolution” version augments images with a spatial resolution of 0.3 m/px, which matches the spatial resolution of the extracted fishing vessel instances and avoids the need for normalization. This reduces the number of images selected for augmentation to 109. The ”matched resolution” variant is tested to determine whether the normalization step during augmentation has a notable impact on the network’s performance.

To decide how many fishing vessels to add to the dataset, we analyzed the distribution of ship categories in the ShipRSImageNet dataset, which is displayed in Figure 3.4. This analysis revealed that while the largest categories contain around 1,000 instances each, there are only 318 instances of fishing vessels. To balance the dataset, we plan to add 513 new fishing vessel instances, increasing their total to 831. The distribution of ship categories after application of augmentation is also illustrated in Figure 3.4.

During the image augmentation selection process, manual observation indicated that most selected images could comfortably accommodate up to five additional fishing ships. Thus, we set a maximum of five and a minimum of one fishing vessel per image. Initially, 513 vessels were added to the dataset. However, for the variant of the augmentation process that matched image resolutions, we reduced the total number of added ships to 327 to prevent overcrowding, as there were fewer suitable images. Despite concerns that fewer ships might decrease performance, tests showed that maintaining the original count of 513 did not significantly improve, and sometimes slightly worsened, performance.

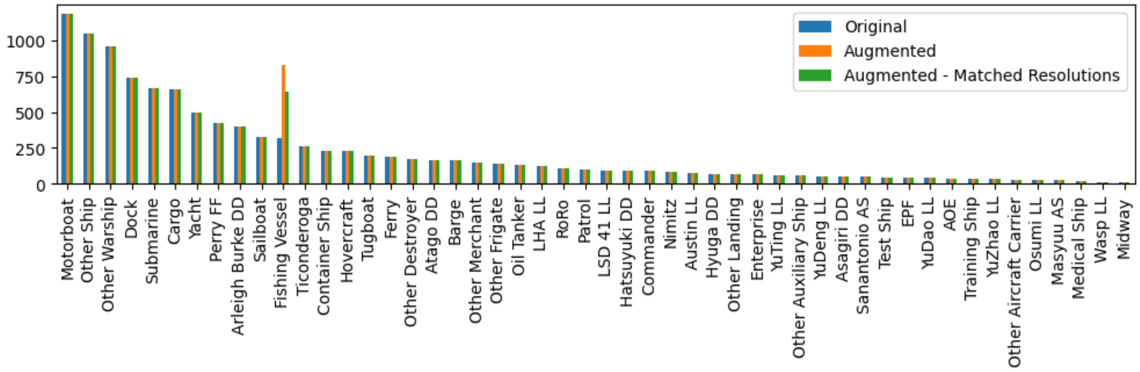


Fig. 3.4: Distributions of fine-grained ship categories in the training set of ShipRSImageNet datasets

3.5.2 Model Training and Validation

For each experiment, an NVIDIA RTX 3090 24 GB GPU was used to train the YOLOv5 nano, v5 small, and v8 nano models. A batch size of 32 was used, and models were trained for 100 epochs, starting with the standard pre-trained weights from the MS-COCO dataset for each YOLO model. For YOLOv8 nano, an additional training run of 300 epochs for each experiment was conducted. YOLOv5 models were not trained up to 300 epochs due to signs of overfitting. The model with the best performance over the span of each training job was selected for use in the evaluation of model performance on the original validation set of ShipRSImageNet.

3.5.3 Classification Performance

To evaluate the impact of augmentation and class weighting on YOLOv5 and YOLOv8, three metrics were used to evaluate the performance of fishing vessel detection: recall, precision of ship classification $Precision_S$, and the rate of false positives from the image background FPR_{bg}. As YOLO is a multi-class classifier, for the computation of classification metrics, the positive class will refer to the fishing vessel category and the negative class will refer to the remaining ship categories. Recall follows the standard definition of $\frac{TP}{TP+FN}$, while $Precision_S$ is a slightly modified form of the precision metric $\frac{TP}{TP+FP}$, with the single change that false positive predictions originating from the image background are excluded from the count of false positives FP. Effectively, $Precision_S$ measures the accuracy of correctly identifying fishing vessels as ships rather than as part of the image background. For the metrics of recall and $Precision_S$, TP, FP, and FN are defined as: the number of true positive predictions, number of false positive predictions, and number of false negative predictions, respectively. The metric FPR_{bg}, defined as BG_p/BG, where BG_p is the number of false positives from the image background and BG is the total number of false positive detections across all ship categories, specifically addresses false positives that arise from the image background.

The training results for the best model after 100 epochs for each YOLO model are detailed in Table 3.3. Only YOLOv8n was trained beyond 100 epochs due to the overfitting of YOLOv5 beyond 100 epochs. Results for the best YOLOv8 model after 300 epochs are described in Table 3.4.

YOLOv5n shows up to a 10% improvement in recall over the baseline when using augmentation, with a slight increase in precision and minimal impact on the background false positive rate. YOLOv8 shows a similar pattern, achieving an 8% increase in recall after extended training for up to 300 epochs and specifically when augmentations are matched to the resolutions of the images to prevent rescaling during normalization.

Table 3.3: Classification Results for Fishing Vessels — Best Model of 100 Epochs

	YOLOv5n			**YOLOv5s**			**YOLOv8n**		
Experiment	$Recall$	$Precision_S$	FPR_{bg}	$Recall$	$Precision_S$	FPR_{bg}	$Recall$	$Precision_S$	FPR_{bg}
Baseline	0.39	0.55	0.06	0.39	**0.62**	0.06	0.45	**0.65**	0.05
Baseline Weighted	0.41	0.53	0.07	0.39	0.57	0.07	**0.53**	0.40	0.12
Augmented	**0.49**	**0.57**	0.07	**0.43**	**0.59**	0.08	**0.49**	0.58	0.06
Augmented MatchRes	**0.46**	**0.58**	0.07	0.40	0.52	0.06	0.30	**0.67**	0.04
Augmented + Weights	0.40	0.46	0.08	**0.44**	0.47	0.06	0.48	0.58	0.06
Aug. MatchRes + Weights	0.37	0.49	0.07	0.32	0.54	0.05	0.47	0.55	0.05

Table 3.4: Classification Results for Fishing Vessels — YOLOv8n Best Model of 300 Epochs

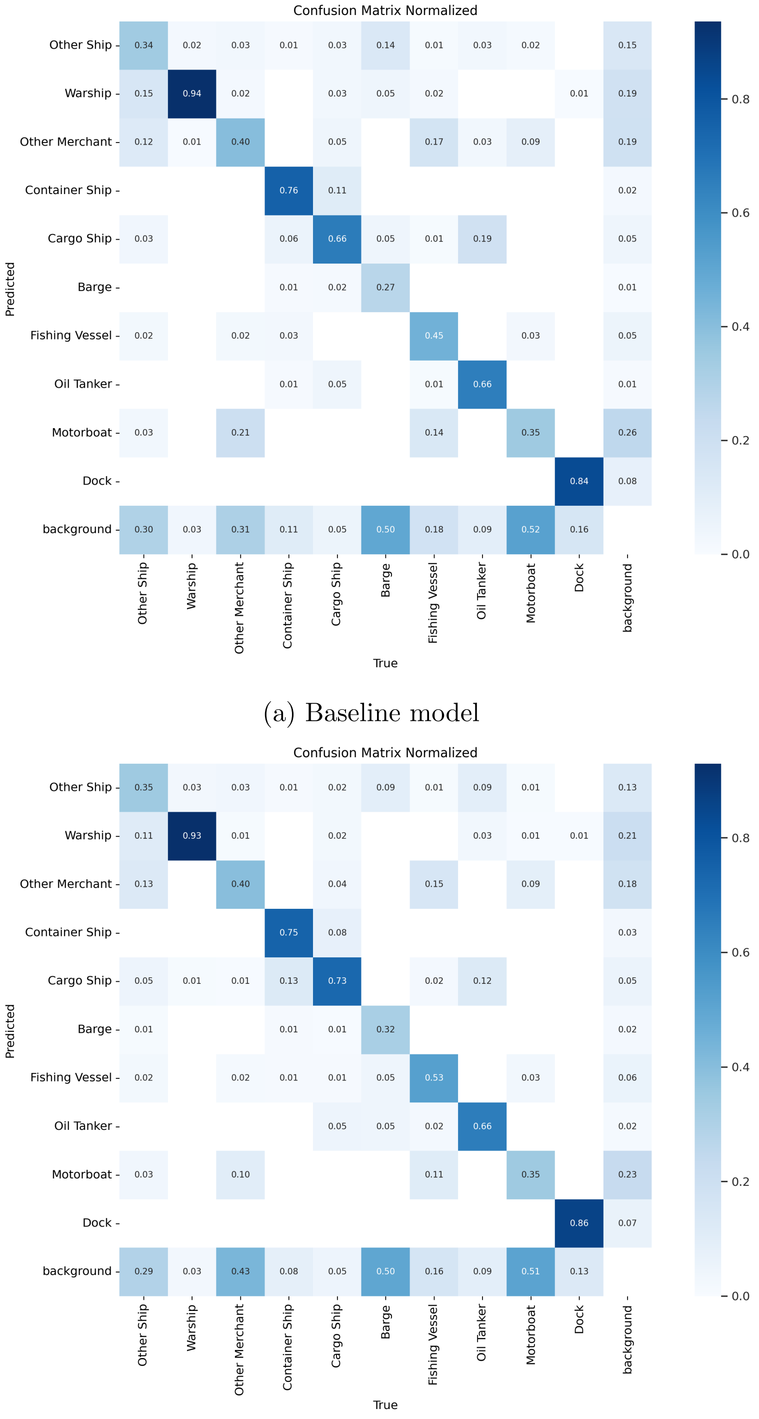
Experiment	$Recall$	$Precision_S$	FPR_{bg}
Baseline	0.45	**0.56**	0.05
Baseline Weighted	0.47	0.53	0.07
Augmented	0.36	0.45	0.06
Augmented MatchRes	**0.53**	**0.60**	0.06
Augmented + Weights	**0.49**	0.49	0.09
Aug. MatchRes + Weights	**0.49**	0.42	0.08

However, when YOLOv8n is trained for only 100 epochs, it exhibits signs of underfitting, although it still improves in recall or precision for fishing vessel detection with each augmentation method. Meanwhile, YOLOv5s sees a small gain in recall but at the expense of precision among ship classes. The lesser performance of YOLOv5s compared to YOLOv5n might be due to less optimal training parameters or insufficient training duration. Alternatively, the YOLOv5s architecture might struggle with overfitting on a relatively small dataset.

In terms of augmentation techniques, the matched resolution method generally achieves slightly higher precision than the standard method, suggesting that reducing ship normalization during augmentation enhances precision.

Combining class weighting with augmentation generally improves recall more than using class weighting alone, though its effect on precision varies. However, class weighting tends to significantly increase the false positive rate from the background (FPR_{bg}), leading to more false detections and sometimes worse performance than the baseline. This increase in FPR_{bg} is consistent across all models that use class weighting, with these classifiers producing at least twice as many background false positives across all ship categories. For instance, the baseline YOLOv8n model registered 763 false ship detections in the background, whereas the class-weighted baseline model saw 2,468 such detections. While combining augmentation and class weighting in YOLOv8n reduced the false detections to 1,895, using augmentation without class weighting was more effective, resulting in only 791 false background detections.

The top performing YOLOv8n model utilizing the matched resolution augmentation method additionally resulted in noticeable recall improvements to the cargo ship and barge ship categories, as evidenced in the confusion matrices in Fig. 3.5. When considering how other vessel types may work in tandem with dark fishing vessels (e.g. a refrigerated cargo ship with AIS operating near a fishing vessel with AIS disabled so that a catch can be offloaded and processed at sea [6]), this improved general classification performance is of further benefit for systems designed to detect illegal fishing from ship movement data.



(a) Baseline model

(b) Augmented MatchRes model

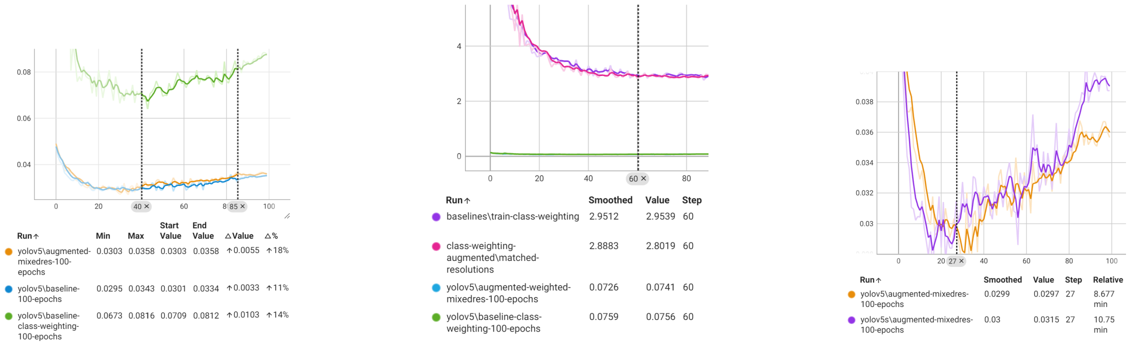
Fig. 3.5: Confusion matrices of baseline YOLOv8n model and best YOLOv8n model with augmentation — best of 300 epochs

3.5.4 Impact of Augmentation on Overfitting

The use of class weighting has been found to increase the likelihood of overfitting in models. This tendency is illustrated in Figure 3.6a, which shows the model's validation losses. A similar pattern is observed with YOLOv8, except that for experiments without class weighting, the validation loss tends to plateau over time instead of rising again.

Models trained with augmented datasets exhibit overfitting at rates comparable to those trained on baseline datasets. However, models using class weighting are more prone to overfitting. Figure 3.6b shows that combining class weighting with augmentation does not significantly mitigate overfitting, considering the specific types and amounts of augmentation used.

Figure 3.6c compares the validation loss of the best-performing YOLOv5n and YOLOv5s models. As detailed in Table 3.3, YOLOv5n outperformed YOLOv5s. One potential reason for this discrepancy in performance might be that increasing the number of parameters in a model, particularly with a small dataset, can lead most of these new parameters to contribute to overfitting rather than genuine learning.



	Run ↑	Min	Max	Start Value	End Value	△Value	△%
●	yolov5\augmented-mixedres-100-epochs	0.0303	0.0358	0.0303	0.0358	↑0.0055	↑18%
●	yolov5\baseline-100-epochs	0.0295	0.0343	0.0301	0.0334	↑0.0033	↑11%
●	yolov5\baseline-class-weighting-100-epochs	0.0673	0.0816	0.0709	0.0812	↑0.0103	↑14%

(a) YOLOv5 for baseline, augmented, and weighted experiments

	Run ↑	Smoothed	Value	Step
●	baselines\train-class-weighting	2.9512	2.9539	60
●	class-weighting-augmented\matched-resolutions	2.8883	2.8019	60
●	yolov5\augmented-weighted-mixedres-100-epochs	0.0726	0.0741	60
●	yolov5\baseline-class-weighting-100-epochs	0.0759	0.0756	60

(b) YOLOv5 and YOLOv8 for augmented-only and augmented + weights experiments

	Run ↑	Smoothed	Value	Step	Relative
●	yolov5\augmented-mixedres-100-epochs	0.0299	0.0297	27	8.677 min
●	yolov5s\augmented-mixedres-100-epochs	0.03	0.0315	27	10.75 min

(c) YOLOv5n and YOLOv5s augmented experiments

Fig. 3.6: Validation loss comparisons of trained YOLO models; YOLOv8 experiments additionally produced results very similar to a.

Unlike YOLOv5s, YOLOv8n did not show a similar extent of overfitting, likely due to its newer architecture and smaller size. However, the small variance in validation losses between YOLOv5n and YOLOv5s makes it challenging to definitively attribute the performance differences to specific causes without additional experiments.

3.5.5 Performance on Edge Compute Platforms

To rigorously assess the inference capabilities of YOLO models enhanced by POSEIDON-SAT, we conducted benchmark tests on an NVIDIA Jetson Orin Nano 8GB DevKit. This setup mirrors the computational constraints typical of edge devices aboard cube satellites and other small-form Low Earth Orbit (LEO) spacecraft, ensuring our findings are directly applicable to real-world satellite operations. We optimized the models for the NVIDIA TensorRT runtime, which is tailored to maximize inference performance on NVIDIA hardware.

The benchmarks focused on evaluating model performance under the constrained power conditions typical of small satellites. We tested the models at two distinct power profiles—15 W and 7 W—to simulate operational scenarios in space where power efficiency is paramount. The results of these benchmarks, which provide insights into both the speed and efficiency of the models under different power conditions, are comprehensively detailed in Table 3.5. Benchmarks were performed across the validation dataset with a single image batch size and an input dimension of 960×960. Additionally, the memory footprints of each model configuration, crucial for understanding their viability on resource-limited platforms, are documented in Table 3.6. Note that the Jetson Orin Nano uses a shared CPU and GPU memory architecture. This evaluation underscores our models' suitability for deployment in spaceborne systems, where minimizing power consumption and optimizing space is critical.

3.6 Conclusion

The application of the POSEIDON-SAT dataset augmentation method to YOLOv5 and YOLOv8 models trained on the ShipRSImageNet dataset has not only improved recall by 10% and 8% respectively but also increased precision by 3–4% in accurately detecting fishing vessels. This method outperforms traditional class weighting, offering less overfit-

Table 3.5: Performance of Best Models on Jetson Orin Nano 8GB DevKit

Power Profile	**15 W**				**7 W**			
Model	Pre. (ms)	Inference (ms)	Post. (ms)	FPS (960× 960)	Pre. (ms)	Inference (ms)	Post. (ms)	FPS (960× 960)
YOLOv5n Augmented	2.3	11.6	4.9	53.2	2.9	19.4	6.9	34.2
YOLOv5s Augmented	1.8	15.1	4.1	47.6	2.9	35.1	6.6	22.4
YOLOv8n Augmented + MatchRes	2.2	11.9	4.2	54.6	3.2	26.7	6.4	27.5

Table 3.6: Memory Footprint of Best Models on Jetson Orin Nano 8GB DevKit

Model	**Model Size (MiB)**	**Total Inference Memory Use (GiB)**
YOLOv5n	5.9	1.25
YOLOv5s	15.9	1.29
YOLOv8n	7.6	2.93

ting, superior classification performance, and fewer false positives from the background. Remarkably, using POSEIDON-SAT without combining it with class weighting yielded the best results.

The significant improvements in recall are particularly vital for detecting fishing vessels, enhancing the model's ability to identify "dark" fishing vessels. Despite the models producing a substantial number of false positives—–likely due to the lack of background-only images—–the enhanced detection capabilities facilitated by POSEIDON-SAT are crucial for systems that rely on analyzing ship movement data. Such systems can now better differentiate between real and false-positive movements, thereby improving the overall efficacy of illegal fishing detection.

Moreover, the successful application of POSEIDON-SAT has shown to improve classification accuracy across various ship categories, which could be instrumental in identifying vessels, such as refrigerated cargo ships, potentially involved in illegal fishing operations.

Looking forward, we recommend further testing of the POSEIDON-SAT method across different CNN architectures and on larger datasets to explore whether a greater variety of source ships and backgrounds can further boost model performance. More realistic ship instance integration during augmentation and enhanced extraction processes with detailed segmentation labels are also anticipated to yield positive outcomes. Additionally, deploying POSEIDON-SAT on SAR and other types of overhead maritime aerial imagery datasets could significantly enhance dataset quality and scalability, broadening the scope of its application in fishing vessel detection.

This method's effectiveness on cube-satellites equipped with edge computing capabilities highlights its practicality for real-world satellite operations. These satellites, operating under strict power and computational constraints, benefit immensely from the lightweight nature of the augmented YOLO models, making this approach a transformative solution for continuous and efficient maritime surveillance from space.

Acknowledgment

This project is completed with collaboration from the U.S. National Aeronautics & Space Administration (NASA) funded LACE-C3A (Low-power Array for CubeSat Edge Computing Architecture, Algorithms and Applications) mission.

CHAPTER 4

CONCLUSION AND FUTURE WORK

This chapter will provide a final discussion of the impacts of the experiments discussed in Chapter 3 on the design of a Maritime Surveillance System (MSS), such as the MSS proposed in Chapter 2, and will provide concluding remarks on future research direction.

4.1 Ship Detection

Though classification performance for fishing vessel detection is still far from ideal, the POSEIDON-SAT method is able to achieve notable performance improvements over a YOLO model trained with a non-augmented dataset. Evaluation of the proposed POSEIDON-SAT augmentation method has shown that application of this method can help to overcome barriers to improving classification performance of ship detection on current and future fine-grained ship detection datasets. A detailed summary of the benefits and merits of POSEIDON-SAT were previously discussed in Section 3.6.

Of particular concern for application of a YOLO model trained with a POSEIDON-SAT augmented dataset for use in the MSS described in Chapter 2 is the inference performance on an edge compute platform with a 10 W power limit, such as the NVIDIA Jetson Orin Nano described in Section 2.2.2. Results from benchmarking inference performance of the best YOLO models on the Jetson Orin Nano 8GB DevKit, discussed in Section 3.5.5, confirms the suitability of running these models on such a compute platform based on results detailed in Table 3.5 for the 7 W power profile. Memory footprints of each model, shown in Table 3.6, are also low enough to be practically deployed to resource-constrained compute platforms.

4.1.1 Future Research Directions for Ship Detection

Finding an ultimate solution to the difficulty of constructing a lightweight ship detection model for optical satellite imagery is still ongoing, and requires further enhancements to dataset availability and model architectures. Though our POSEIDON-SAT augmentation method shows notable performance improvements for fishing vessel classification, larger and more robust base datasets are needed to understand the full potential impacts of POSEIDON-SAT on the performance of future detection models.

Several of the potential enhancements to the proposed POSEIDON-SAT augmentation method discussed in Section 3.6 have significant promise for future research. The false-positives in the ship detection model have a significant potential for improvement by adding background images to the dataset, as the ShipRSImageNet dataset this thesis applied POSEIDON-SAT to does not contain any background-only images. The best training practices for YOLO recommend including a significant number of background-only images [36], though adding such images should be relatively straightforward due to higher availability of satellite images which consist of only open ocean. Adding segmentation information about shorelines in ShipRSImageNet images would allow utilizing more images during the POSEIDON-SAT augmentation process, enabling the use of images with shorelines without potentially placing ships on land. Creating an initial model from a synthetic fine-grained ship detection dataset, such as UnityShip [54], followed by training on a real dataset augmented with POSEIDON-SAT may further enchance the ship detection model performance. Testing POSEIDON-SAT augmentation on the training dataset of an LMO-YOLO model [55], which is designed for detection of ships in low-resolution images, should improve the detection of smaller fishing vessels. Enhancing POSEIDON-SAT through the use of methods for more realistic incorporation of generated fishing vessel instances, such as the method proposed in [56], should additionally result in more robust ship detection models.

4.2 Discussion of Illegal Fishing Detection Agent Design

The decision-making system of the proposed MSS handles the task of leveraging ship location and category information to determine whether a vessel is engaging in illegal fishing activity. Effective approaches for monitoring ship movement activities to detect illegal fishing activity where high quality ship location data is available, such as data from AIS, have been proposed in [4, 11]. Similar approaches could be leveraged for ship location and category data produced by ship detection models, though the proposed ship detection approach has a higher degree of uncertainty in the produced data. However, the generally high performance of such approaches indicates that these models may be able to effectively filter out false-positives from the ship detection model, permitting the ship detection model to lean more towards prioritizing recall to potentially detect a larger percentage of all illegal fishing activity.

Consideration of the known uncertainties within the ship location and category data produced by the ship detection of the MSS is worth direct consideration and research. Leveraging known uncertainty, such as the probability of an inaccurate ship category prediction, can be used to improve the reliability of the illegal fishing detection and response decision-making agent. Techniques such as Reinforcement Learning (RL) are excellent candidates for application to environments with notable variance and uncertainty. RL has seen significant application to robotics applications, where complex behaviors and noisy inputs are common, indicating a high degree of promise for application to a illegal fishing detection and decision-making system.

If the ship detection model and the illegal fishing detection and response decision-making agent are built to enable real-time observation of an area of interest, with enough satellites to observe the area of interest over a longer time period are deployed, an RL approach could yield a significant benefit in the form of enabling real-time learning based on developing conditions. The ability to deploy ship detection on low-cost satellites makes such a multi-satellite system and real-time learning agent much more affordable and practical, especially as smaller satellites in Low Earth Orbit (LEO) may be able to make more frequent observations than a single, larger satellite.

4.3 Application to Existing and Future Datasets

The lack of large datasets for ship detection from optical satellite imagery continues to complicate research in the area of ship detection from optical satellite imagery. Dataset augmentation provides a useful technique for balancing datasets and for increasing the size of existing datasets. Though performance of models augmented with POSEIDON-SAT yield notable improvements, further enhancements are needed to improve high classification performance.

The detection models resulting from this research could be leveraged for the construction of newer datasets. Deployment of multiple low-power satellites with optical imaging capabilities and AIS receivers could continually image areas of interest with a reasonably high frequency, due to the quicker orbital period of such small satellites when deployed in LEO. Developing and running an AIS correspondence algorithm, similar to those described by [14,15,47], onboard the satellite could be used to ensure that communication bandwidth is used only to transmit images which can be automatically labeled for construction of a large dataset. The LACE-C3A satellite platform used for the proposed MSS has the benefit of these small satellites being equipped with AIS as well as optical imagery capabilities, which could allow real-time correlation of images to AIS data at the time of image capture on the satellite. Such a real-time onboard AIS correlation approach would likely use simpler algorithms than the methods referenced previously, which are focused on correlating AIS data with SAR imagery which were captured by different sources, potentially at different times. Constructing a dataset through the application of the MSS proposed in this thesis would provide a significant contribution to the field of optical fine-grained ship detection, where dataset availability is extremely limited.

The POSEIDON-SAT augmentation method additionally has potential application to SAR and other types of ship imaging datasets where ship instances are clearly distinguishable. The significant contrast between ship instances and their background in images, such as SAR, should make the POSEIDON-SAT method transfer well to other types of ship datasets. With the most significant challenge for ship detection systems from satellite imagery being centered around the lack of large datasets, future work should explore the application of augmentation techniques to larger and initially more varied datasets to aid in the creation of high-quality training datasets as future data collection work is performed.

REFERENCES

[1] A. J. Temple, D. J. Skerritt, P. E. Howarth, J. Pearce, and S. C. Mangi, "Illegal, unregulated and unreported fishing impacts: A systematic review of evidence and proposed future agenda," *Mar. Policy*, vol. 139, p. 105033, 2022, doi: 10.1016/j.marpol.2022.105033.

[2] U. R. Sumaila, D. Zeller, L. Hood, M. Palomares, Y. Li, and D. Pauly, "Illicit trade in marine fish catch and its effects on ecosystems and people worldwide," *Sci. Advances*, vol. 6, no. 9, p. eaaz3801, 2020, doi: https://doi.org/10.1126/sciadv.aaz3801.

[3] D. J. Agnew, J. Pearce, G. Pramod, T. Peatman, R. Watson, J. R. Beddington, and T. J. Pitcher, "Estimating the worldwide extent of illegal fishing," *PloS One*, vol. 4, no. 2, p. e4570, 2009, doi: 10.1371/journal.pone.0004570.

[4] D. M. Nunes, A. C. Bezerra, W. M. Barros, P. V. Araújo, I. S. Branco-Nunes, R. A. Magris, P. H. Pereira, I. C. Normande, R. S. Barboza, and A. T. Cardoso, "Evidence of illegal fishing within the largest Brazilian coastal MPA: Turning a blind eye to the obvious," *Mar. Policy*, vol. 147, p. 105324, 2023, doi: 10.1016/j.marpol.2022.105324.

[5] R. B. Cabral, J. Mayorga, M. Clemence, J. Lynham, S. Koeshendrajana, U. Muawanah, D. Nugroho, Z. Anna, Mira, A. Ghofar *et al.*, "Rapid and lasting gains from solving illegal fishing," *Nature Ecology & Evolution*, vol. 2, no. 4, pp. 650–658, 2018, doi: 10.1038/s41559-018-0499-1.

[6] H. Welch, T. Clavelle, T. D. White, M. A. Cimino, J. Van Osdel, T. Hochberg, D. Kroodsma, and E. L. Hazen, "Hot spots of unseen fishing vessels," *Sci. Advances*, vol. 8, no. 44, p. eabq2109, 2022, doi: 10.1126/sciadv.abq2109.

[7] F. S. Paolo, D. Kroodsma, J. Raynor, T. Hochberg, P. Davis, J. Cleary, L. Marsaglia, S. Orofino, C. Thomas, and P. Halpin, "Satellite mapping reveals extensive industrial activity at sea," *Nature*, vol. 625, no. 7993, pp. 85–91, 2024, doi: 10.1038/s41586-023-06825-8.

[8] J. Park, J. Lee, K. Seto, T. Hochberg, B. A. Wong, N. A. Miller, K. Takasaki, H. Kubota, Y. Oozeki, S. Doshi *et al.*, "Illuminating dark fishing fleets in North Korea," *Sci. Advances*, vol. 6, no. 30, p. eabb1197, 2020, doi: 10.1126/sciadv.abb1197.

[9] M. Fingas and C. Brown, "Review of ship detection from airborne platforms," *Can. J. Remote Sens.*, vol. 27, no. 4, pp. 379–385, 2001, doi: 10.1080/07038992.2001.10854880.

[10] A. Dzvonkovskaya, K.-W. Gurgel, H. Rohling, and T. Schlick, "Low power high frequency surface wave radar application for ship detection and tracking," in *Int. Conf. Radar*. IEEE, 2008, pp. 627–632, doi: 10.1109/RADAR.2008.4653998.

[11] D. A. Kroodsma, J. Mayorga, T. Hochberg, N. A. Miller, K. Boerder, F. Ferretti, A. Wilson, B. Bergman, T. D. White, B. A. Block *et al.*, "Tracking the

global footprint of fisheries," *Science*, vol. 359, no. 6378, pp. 904–908, 2018, doi: 10.1126/science.aao5646.

[12] U. Kanjir, H. Greidanus, and K. Oštir, "Vessel detection and classification from spaceborne optical images: A literature survey," *Remote Sens. Environ.*, vol. 207, pp. 1–26, 2018, doi: 10.1016/j.rse.2017.12.033.

[13] F. Paolo, T.-t. T. Lin, R. Gupta, B. Goodman, N. Patel, D. Kuster, D. Kroodsma, and J. Dunnmon, "xView3-SAR: Detecting dark fishing activity using synthetic aperture radar imagery," in *Advances Neural Inf. Process. Syst.*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 37 604–37 616. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/f4d4a021f9051a6c18183b059117e8b5-Paper-Datasets_and_Benchmarks.pdf

[14] R. Pelich, M. Chini, R. Hostache, P. Matgen, C. Lopez-Martinez, M. Nuevo, P. Ries, and G. Eiden, "Large-scale automatic vessel monitoring based on dual-polarization Sentinel-1 and AIS data," *Remote Sens.*, vol. 11, no. 9, p. 1078, 2019, doi: 10.3390/rs11091078.

[15] G. Margarit, J. A. Barba Milanés, and A. Tabasco, "Operational ship monitoring system based on synthetic aperture radar processing," *Remote Sens.*, vol. 1, no. 3, pp. 375–392, 2009, doi: 10.3390/rs1030375.

[16] Iridium Satellite LLC, McLean, VA, USA. *Fact Sheet: Iridium Certus 700.* (2023). Ver. 2.0. Accessed: May 17, 2024. [Online]. Available: https://www.iridium.com/resources/resource-download/?resource-id=303179

[17] H. Bouma, R. J. Dekker, R. M. Schoemaker, and A. A. Mohamoud, "Segmentation and wake removal of seafaring vessels in optical satellite images," in *Electro-Optical Remote Sens., Photonic Technologies, Appl. VII; Mil. Appl. Hyperspectral Imag. High Spatial Resolution Sens.*, vol. 8897. SPIE, 2013, pp. 93–103, doi: 10.1117/12.2029791.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.

[19] G. Jocher, "Ultralytics YOLOv5," Jun. 2020, Ver. 7.0. Accessed: Apr. 9, 2024. doi: 10.5281/zenodo.3908559. [Online]. Available: https://github.com/ultralytics/yolov5

[20] G. Jocher, A. Chaurasia, and J. Qiu, "Ultralytics YOLOv8," Jan. 2023, Ver. 8.0.0. Accessed: Feb. 26, 2024. [Online]. Available: https://github.com/ultralytics/ultralytics

[21] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 7263–7271, doi: 10.1109/CVPR.2017.690.

[22] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[23] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A comprehensive review of YOLO architectures in computer vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680–1716, 2023, doi: 10.3390/make5040083.

[24] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.

[25] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 2117–2125, doi: 10.1109/CVPR.2017.106.

[26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances Neural Inf. Process. Syst.*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, Eds., vol. 32. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[27] NVIDIA Corporation, Santa Clara, CA, USA. *NVIDIA Jetson Orin Nano Series Modules.* (2023). No. DS-11105-001_v1.2, ver. 1.2. Accessed: Mar. 22, 2024. [Online]. Available: https://developer.nvidia.com/downloads/assets/embedded/secure/jetson/orin_nano/docs/jetson_orin_nano_ds

[28] P. Plagwitz, F. Hannig, M. Ströbel, C. Strohmeyer, and J. Teich, "A safari through FPGA-based neural network compilation and design automation flows," in *2021 IEEE 29th Annu. Int. Symp. Field-Programmable Custom Comput. Machines (FCCM)*. IEEE, 2021, pp. 10–19, doi: 10.1109/FCCM51124.2021.00010.

[29] Microchip Technology Inc., Chandler, AZ, USA. *PolarFire FPGA Product Overview.* (2023). No. DS60001657M. Accessed: May 20, 2024. [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/ProductBrief/PolarFire-FPGA-Product-Overview-60001657.pdf

[30] ——. *Radiation-Tolerant PolarFire FPGA Product Overview.* (2024). No. DS00004232B. Accessed: May 20, 2024. [Online]. Available: https://ww1.microchip.com/downloads/aemDocuments/documents/FPGA/ProductDocuments/ProductBrief/Radiation-Tolerant-PolarFire-FPGA-Product-Overview.pdf

[31] "VectorBlox Accelerator Software Development Kit for PolarFire FPGAs." Microchip. Accessed: May 20, 2024. [Online]. Available: https://www.microchip.com/en-us/products/fpgas-and-plds/fpga-and-soc-design-tools/vectorblox

[32] Y. Yu, T. Zhao, K. Wang, and L. He, "Light-OPU: An FPGA-based overlay processor for lightweight convolutional neural networks," in *Proc. 2020 ACM/SIGDA Int. Symp. Field-Programmable Gate Arrays*, 2020, pp. 122–132, doi: 10.1145/3373087.3375311.

[33] Z. Zhang, M. P. Mahmud, and A. Z. Kouzani, “Resource-constrained FPGA implementation of YOLOv2,” *Neural Comput. and Appl.*, vol. 34, no. 19, pp. 16 989–17 006, 2022, doi: 10.1007/s00521-022-07351-w.

[34] “Vitis AI.” AMD. Accessed: Apr. 26, 2024. [Online]. Available: https://www.xilinx.com/products/design-tools/vitis/vitis-ai.html

[35] “Intel FPGA AI Suite.” Intel. Accessed: Apr. 26, 2024. [Online]. Available: https://www.intel.com/content/www/us/en/software/programmable/fpga-ai-suite/overview.html

[36] G. Jocher. *Tips for Best Training Results - YOLOv5*. (2024). Ultralytics. Accessed: Mar. 27, 2024. [Online]. Available: https://docs.ultralytics.com/yolov5/tutorials/tips_for_best_training_results/

[37] Z. Zhang, L. Zhang, Y. Wang, P. Feng, and R. He, “ShipRSImageNet: A large-scale fine-grained dataset for ship detection in high-resolution optical remote sensing images,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 8458–8472, 2021, doi: 10.1109/JSTARS.2021.3104230.

[38] G.-S. Xia, X. Bai, J. Ding, Z. Zhu, S. Belongie, J. Luo, M. Datcu, M. Pelillo, and L. Zhang, “DOTA: A large-scale dataset for object detection in aerial images,” in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 3974–3983, doi: 10.1109/CVPR.2018.00418.

[39] K. Li, G. Wan, G. Cheng, L. Meng, and J. Han, “Object detection in optical remote sensing images: A survey and a new benchmark,” *ISPRS J. Photogrammetry and Remote Sens.*, vol. 159, pp. 296–307, 2020, doi: 10.1016/j.isprsjprs.2019.11.023.

[40] Z. Liu, L. Yuan, L. Weng, and Y. Yang, “A high resolution optical satellite image dataset for ship recognition and some new baselines,” in *Proc. 6th Int. Conf. Pattern Recognit. Appl. and Methods - ICPRAM*, vol. 1, INSTICC. SciTePress, 2017, pp. 324–331, doi: 10.5220/0006120603240331.

[41] K. Chen, M. Wu, J. Liu, and C. Zhang, “FGSD: A dataset for fine-grained ship detection in high resolution satellite images,” 2020, *arXiv:2003.06832*.

[42] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord, “xView: Objects in context in overhead imagery,” 2018, *arXiv:1802.07856*.

[43] L. Huang, B. Liu, B. Li, W. Guo, W. Yu, Z. Zhang, and W. Yu, “OpenSARShip: A dataset dedicated to Sentinel-1 ship interpretation,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 1, pp. 195–208, 2017, doi: 10.1109/JSTARS.2017.2755672.

[44] B. Li, B. Liu, L. Huang, W. Guo, Z. Zhang, and W. Yu, “OpenSARShip 2.0: A large-volume dataset for deeper interpretation of ship targets in Sentinel-1 imagery,” in *2017 SAR Big Data Era: Models, Methods and Appl. (BIGSARDATA)*. IEEE, 2017, pp. 1–5, doi: 10.1109/BIGSARDATA.2017.8124929.

[45] X. Hou, W. Ao, Q. Song, J. Lai, H. Wang, and F. Xu, "FUSAR-Ship: building a high-resolution SAR-AIS matchup dataset of Gaofen-3 for ship detection and recognition," *Sci. China Inf. Sciences*, vol. 63, pp. 1–19, 2020, doi: 10.1007/s11432-019-2772-5.

[46] S. Lei, D. Lu, X. Qiu, and C. Ding, "SRSDD-v1.0: A high-resolution SAR rotation ship detection dataset," *Remote Sens.*, vol. 13, no. 24, p. 5104, 2021, doi: 10.3390/rs13245104.

[47] A. Milios, K. Bereta, K. Chatzikokolakis, D. Zissis, and S. Matwin, "Automatic fusion of satellite imagery and AIS data for vessel detection," in *2019 22th Int. Conf. on Inf. Fusion (FUSION)*. IEEE, 2019, pp. 1–5, doi: 10.23919/FUSION43075.2019.9011339.

[48] P. Ruiz-Ponce, D. Ortiz-Perez, J. Garcia-Rodriguez, and B. Kiefer, "POSEIDON: A data augmentation tool for small object detection datasets in maritime environments," *Sensors*, vol. 23, no. 7, p. 3691, 2023, doi: 10.3390/s23073691.

[49] R. Ribeiro, G. Cruz, J. Matos, and A. Bernardino, "A data set for airborne maritime surveillance environments," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 9, pp. 2720–2732, 2017, doi: 10.1109/TCSVT.2017.2775524.

[50] Y. Cui, M. Jia, T.-Y. Lin, Y. Song, and S. Belongie, "Class-balanced loss based on effective number of samples," in *Proc. IEEE/CVF Conf. Comput. Vision and Pattern Recognit.*, 2019, pp. 9268–9277, doi: 10.1109/CVPR.2019.00949.

[51] Y. Zhang, R. Togneri, and D. Huang, "A unified loss function to tackle inter-class and intra-class data imbalance in sound event detection," in *ICASSP 2024-2024 IEEE Int. Conf. on Acoust., Speech and Signal Process.* IEEE, 2024, pp. 996–1000, doi: 10.1109/ICASSP48485.2024.10447675.

[52] Y. Ho and S. Wookey, "The real-world-weight cross-entropy loss function: Modeling the costs of mislabeling," *IEEE Access*, vol. 8, pp. 4806–4813, 2020, doi: 10.1109/ACCESS.2019.2962617.

[53] J. Ansel, E. Yang, H. He, N. Gimelshein, A. Jain, M. Voznesensky, B. Bao, P. Bell, D. Berard, E. Burovski, G. Chauhan, A. Chourdia, W. Constable, A. Desmaison, Z. DeVito, E. Ellison, W. Feng, J. Gong, M. Gschwind, B. Hirsh, S. Huang, K. Kalambarkar, L. Kirsch, M. Lazos, M. Lezcano, Y. Liang, J. Liang, Y. Lu, C. Luk, B. Maher, Y. Pan, C. Puhrsch, M. Reso, M. Saroufim, M. Y. Siraichi, H. Suk, M. Suo, P. Tillet, E. Wang, X. Wang, W. Wen, S. Zhang, X. Zhao, K. Zhou, R. Zou, A. Mathews, G. Chanan, P. Wu, and S. Chintala, "PyTorch 2: Faster machine learning through dynamic python bytecode transformation and graph compilation," in *Proc. 29th ACM Int. Conf. on Architectural Support for Programming Languages and Operating Systems (ASPLOS '24)*, vol. 2. Association for Computing Machinery, Apr. 2024, p. 929–947, doi: 10.1145/3620665.3640366.

[54] B. He, X. Li, B. Huang, E. Gu, W. Guo, and L. Wu, "UnityShip: A large-scale synthetic dataset for ship recognition in aerial images," *Remote Sens.*, vol. 13, no. 24, p. 4999, 2021, doi: 10.3390/rs13244999.

[55] Q. Xu, Y. Li, and Z. Shi, “LMO-YOLO: A ship detection model for low-resolution optical satellite imagery,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4117–4131, 2022, doi: 10.1109/JSTARS.2022.3176141.

[56] W. Zhang, R. Zhang, G. Wang, W. Li, X. Liu, Y. Yang, and D. Hu, “Physics guided remote sensing image synthesis network for ship detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–14, 2023, doi: 10.1109/TGRS.2023.3248106.