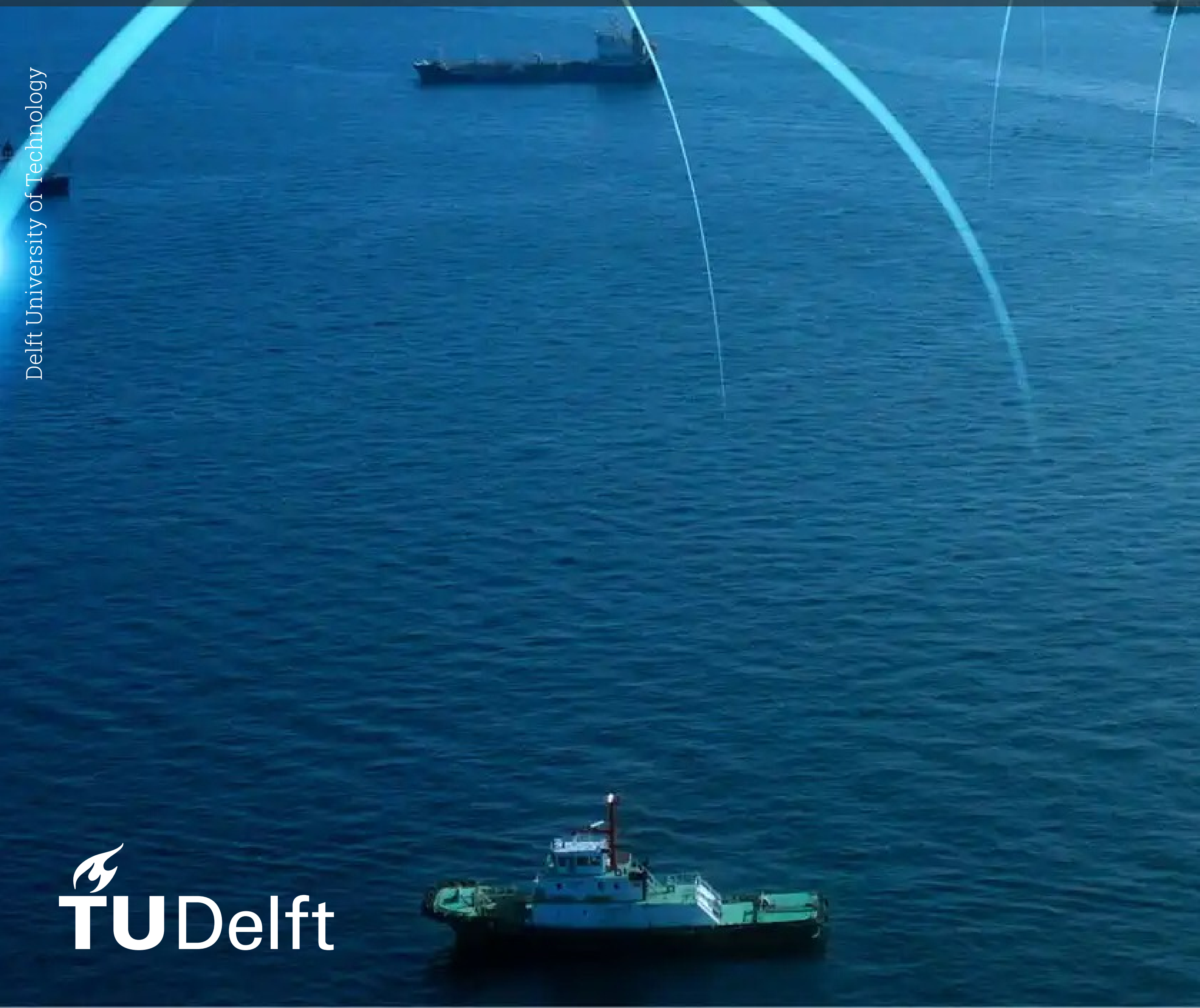
Machine Learning for IWT Ship Traffic Analysis (Object Detection vs. AIS Data)

Suryaa Vadachennimalai Selvaraj



Delft University of Technology

Machine Learning for IWT Ship Traffic Analysis (Object Detection vs. AIS Data)

by

Suryaa Vadachennimalai Selvaraj

Student Number	:	5539293
MSc Track	:	Multi Machine Engineering
Report Number	:	2023.MME.8863

Supervisor:	Peter Wenzel	
Thesis Committee:	Dr. Frederik Schulte	TU Delft Committee Chair, Faculty of 3mE
	Dr. Nadia Pourmohammadzia	TU Delft Committee Member, Faculty of CEG
Date:	October 23, 2023	

TU Delft

Preface

This Thesis is the last step in my journey as a Mechanical Engineering (Multi-Machine Engineering) master's student at TU Delft. At this point, it's only fair that I think back on my amazing journey in this beautiful country over the past two years. I was fortunate to have the opportunity to pursue my master's at TU Delft, where I believe I have grown stronger technically and also as a person.

I extend my heartfelt gratitude to my esteemed committee members. Foremost, I want to express my sincere appreciation and gratitude to my daily supervisor, Peter Wenzel. Your invaluable guidance and unwavering support throughout the entire thesis journey have been immeasurable. Your willingness to share your vast knowledge and insights during our regular meetings has played a pivotal role in shaping the trajectory of this research. Your mentorship has been a beacon of light that guided me through the complexities of this project.

I would also like to extend my thanks to Dr.Frederik Schulte for graciously accepting the role of my committee chair. Your willingness to oversee and guide this graduation project has provided me with a unique and valuable learning experience. Your expertise and perspective have undoubtedly enriched the depth of my research.

Furthermore, I cannot overlook the unwavering support I have received from my family back home. Their constant presence and moral support have served as the foundation of this accomplishment. Their importance as my pillars of strength cannot be emphasized; they have been my sanctuary in times of adversity and triumph. I want to express my heartfelt gratitude to all of my friends and well-wishers for their unwavering support over my two years in Delft.

Suryaa Vadachennimalai Selvaraj
Delft, October 2023

Summary

The Automatic Identification System (AIS) is used in the maritime domain to improve sea traffic safety by requiring vessels to broadcast real-time information such as identity, speed, location, and course. As it allows global monitoring of almost any larger vessel and has the potential to considerably improve vessel traffic services and collision risk assessment, AIS has been used in an increasing number of applications. This emphasizes why the quality of data transmitted is critical. We are also becoming more aware of the possibilities of spoofing or fabrication of AIS data, which has a direct impact on the dependability of AIS data. To find the possibility of the same in Inland Water Transport a research gap was found with the help of literature, and the following question was established

"How can video surveillance aid in identifying and reducing instances of overlapping or spoofed Automatic Identification System (AIS) data in maritime environments?"

Several sub-research questions were developed to solve the research gap. The first sub-question focuses on defining current state-of-the-art methods for determining AIS data quality, object recognition, and identifying anomalies in AIS data. The second question included the creation of an algorithm capable of identifying vessels and comparing them to AIS data. A process architecture graphic was created to explain the entire process, from the acquisition of input data through the results. To provide a complete comprehension of the code, the rationale of the method was discussed in the Pseudo code section.

The third sub-question was based on the object detection model's accuracy, which is discussed in the results and discussion chapter. With tables illustrating the actual number of vessels in the video, the number of vessels detected, and the object detection model's performance for each of the three days. A Sensitivity study also demonstrated the numerous aspects that could alter the model's performance or efficiency.

The fourth sub-question asked about the disparity between AIS data and results from the Object detection model. The tables containing the data and vessel matches from the video are provided in the results chapter, along with an overall statistics table displaying the total number of matches to the number of boats obtained. To bridge the gap between the performance of the object detection model and the actual number of vessels in the video, a manual verification process was carried out. The identified vessels were then manually matched with the AIS data, and once the vessels in the video were accounted for, a graph depicting the number of matches acquired by the model, manually, and unmatched data was shown for each of the three days, and the average was calculated. The fifth sub-question implied the model's practicability in real-world applications and discussed whether the approach was effective in identifying data mismatches along with the constraints that the proposed algorithm currently holds.

By comparing video surveillance with Automatic Identification System (AIS) data, the method was developed to detect data overlap or spoofing in the maritime environment. The effort to solve the aforementioned constraints and potential work are discussed in the section on future

work. Finally, while this thesis shows promise in terms of the model's wider implementation and spotting deviations, it still has some limitations that must be addressed. The step-by-step execution, the dataset and the source code are available in the Github Repository [49].

Contents

List of Figures

List of Tables

Nomenclature

Abbreviations

Abbreviation	Definition
AIS	Automatic Identification System
API	Application Programming Interface
AWS	Amazon Web Servers
IOU	Intersection of Union
ISOLA	Integrated System for Onboard Lineage Analysis
MMSI	Maritime Mobile Service Identities
SAR	Synthetic Aperture Radar
SSD	Single Shot Detector
UAV	Unmanned Aerial Vehicle
VTR	Vehicle Traffic Surveillance
YOLO	You Only Look Once

1 Introduction

Around 90% of global transportation occurs via maritime routes, encompassing activities such as Fishing, Sailing, and Cruising [1]. According to the standards, ships of 300 gross tonnes or more on international voyages, 500 tonnes or more on non-international voyages, and passenger vessels must be equipped with AIS devices [37]. Over 1,490,776 vessels are currently being tracked globally, according to [55]. While, in Europe, all Vessels above the length of 15m are required to provide/transmit AIS data [44]. The International Maritime Organisation (IMO), International Association of Marine Aids to Navigation and Lighthouse Authorities (IALA), International Telecommunication Union (ITU), and International Commission (IEC) collaborated in the 1990s to create the Automatic Identification System to improve navigation safety, collision risk assessment, anomaly detection, trajectory analysis, and vehicle emission analysis [24]. An Automatic Identification System device is a maritime transponder that broadcasts critical vessel information to other nearby vessels and shore-based stations, such as position, course, speed, and identity. This real-time data exchange promotes safe navigation, helps to avoid collisions, and improves maritime situational awareness. This includes both AIS class A and class B devices, with the former being required and the latter optional. Class A devices are installed on vessels required by standards to communicate AIS data, while class B devices are installed on vessels that choose to freely submit data [21]. By localizing vessels and charting maritime traffic, AIS data significantly enhances safety, efficiency, navigation, life at sea, and environmental protection. It effectively improves Vehicle Travel Surveillance (VTR), waterway management, collision avoidance, and location tracking [17].

Despite the mandatory AIS data transmission, validating the accuracy of the transmitted information remains a challenge [51]. The absence of verification mechanisms leaves room for data falsification and spoofing, where deceptive AIS data can create non-existent vessels or obscure true vessel identities [31]. Detecting this spoofing is complex due to the copious amounts of transmitted information [54]. Vessels that opt not to transmit signals are termed ”dark vessels,” with motivations varying from legitimate to illicit [54]. In December 2022, a non-profit organization dedicated to ocean governance discovered instances of position falsification by Venezuelan crude tankers, including the Russian-flagged vessel KAPITAN SCHEMILKIN, which broadcasted bogus positions [4]. Synthetic aperture radar (SAR) aids geospatial analysis, compensating for cloud-obscured satellite images [20]. Research indicates that higher channel loads correlate with increased message loss in AIS data transmission, eroding reliability [35, 24].

This study conducts a comprehensive literature review on AIS data, its applications, and quality concerns, shedding light on potential data falsification and related challenges. Furthermore, it addresses the rising interest in machine learning for maritime risk analysis and safety improvement [42]. This investigation tries to make up for the lack of research on AIS data anomalies by using live video surveillance to look into the ability of machine learning to find vessels and compare them with AIS data. We shift our focus to critical research inside this study, strengthening marine data integrity. The primary question of our investigation is:

”**How can video surveillance aid in identifying and reducing instances of overlapping or spoofed Automatic Identification System (AIS) data in maritime environments?**

The Main research question is answered in the form of the below sub-research questions:
SQ1: What are the current state-of-the-art methods and technologies for detecting and mitigating instances of overlapping or spoofed AIS data in maritime environments?
SQ2: How can an algorithm be systematically selected and designed to effectively identify and mitigate instances of overlapping or spoofed AIS data, considering factors such as accuracy, real-time processing, scalability, and integration with video surveillance?
SQ3: How accurate is the proposed object detection algorithm in identifying and localizing ships from video surveillance in various maritime conditions and environments?
SQ4: What are the discrepancies and variations between the AIS data and the ship detections obtained through the object detection algorithm, and how can these differences be quantified and analyzed?
SQ5: Can combining video surveillance of ships with existing AIS data help spot and recognize cases of overlapping or fake AIS information, and what difficulties and constraints might arise from adopting this approach?

The following chapters will go over the research process. Chapter 2 delves into existing information to identify research gaps. Chapter 3 which is titled AIS Data Validation is a critical component of our study journey, offering insight into the basic issue that needs this endeavour. Delving into the main issues and establishing the compelling reasons for conducting this research. Chapter 4 breaks down how the study puts the algorithm into practice. Following that, Chapter 5 gives the empirical results of the technique, followed by insightful deliberations on the implications and significance. Finally, Chapter 6 helps summarize the findings and emphasize the broader implications by answering the research questions and suggesting future research possibilities.

2 Literature Overview

The primary goal of this chapter is to investigate the most recent state-of-the-art methods for detecting or identifying the Quality of AIS data, advancements in the field of object detection and its application in the detection of vessels in the marine environment, and current methods for detecting spoofing in AIS data. By dissecting these essential areas, this chapter lays a robust foundation for our research, offering a holistic view of the integration of these technologies and methodologies. It not only provides valuable insights into the present state of the field but also illuminates promising avenues for further research and development in the realm of maritime data analytics and security. In the following sections, this Chapter delves into the literature on AIS data quality, object detection advancements, and AIS data spoofing.

2.1. Quality of AIS data

This section delves into the literature on AIS data quality to comprehend the evolving landscape and advancements, essential for enhancing maritime data reliability and security. This exploration aids in uncovering insights to fortify AIS data's accuracy and effectiveness in maritime operations. Within the academic literature, there exist studies[22], and [1] where scholars develop algorithms which are trained using a curated dataset. After the model is trained, it is subjected to the evaluation process using live and authentic AIS data collected in real-time. The objective is to assess the model's capability to accurately discern instances of spoofing, which involves the intentional manipulation or insertion of false information into the AIS data stream.

The papers [14], [43] address security concerns surrounding the Automatic Identification System in maritime traffic control. AIS plays a vital role in enhancing marine domain awareness but has vulnerabilities that could be exploited. The research explores Identity-Based Public Cryptography and Symmetric Cryptography as potential solutions to bolster AIS security. The paper [53] explores how AIS data can be leveraged for tasks such as traffic anomaly detection, route estimation, collision prediction, and path planning, all aimed at improving the safety of seafaring. This add more context to why the quality of AIS data is important. The article [10] reviews methods for analyzing waterway risks using non-accident critical events from AIS data. It focuses on five questions: definition, accident-theoretical basis, ranking, method use, and validation. The results suggest that more focus is needed on defining non-accident critical events and understanding the relationship between their occurrence and accident involvement for effective waterway risk analysis. The article emphasizes the importance of addressing foundational issues in risk research and safety science.

Whereas [32] suggests calculating the Haversine distance, a formula used to determine the shortest distance between two points on the surface of a sphere, for each pair of sequential position messages of a vessel. Subsequently, the time interval between these positions is evaluated, utilizing the timestamps recorded by AIS receivers. By dividing the calculated distance by the time interval, the speed of the vessel is estimated. To ensure the validity of the data, this derived speed is then checked against a feasible range, considering that a vessel's average speed is constrained, for instance, not exceeding 50 knots. If the derived speed falls within this acceptable range, the subsequent AIS message is accepted as the updated and legitimate position for that vessel.

2.2. Object detection

This research project intends to identify vessels using an object detection model, so it is important to understand the current improvements and advancements in the field. The paper [39] addresses technical hurdles in maritime image processing and machine vision from camera-generated video streams. Basic tasks like horizon detection and frame registration are tough due to dynamic backgrounds, lack of static cues and lighting effects. The paper [41] aims to run an object detection algorithm on every video frame, detecting all objects, including people, vehicles, and animals. This system is crucial for computer vision and automated driving systems. With growing computing power and deep learning popularity, high-performance algorithms are becoming more prevalent. The model allows users to detect only needed objects despite training on a larger dataset. The research [11] introduces an improved real-time object detection and recognition technique using web camera video. Using Single Shot Detector (SSD) and YOLO models, the system detects and recognizes objects in adverse environments like excess light, rotation, mirroring, and backgrounds. The convolutional neural network classifies the objects, achieving 63-90% accuracy in detection and classification.

Advances in deep learning, like R-CNN, Fast-RCNN, Faster-RCNN, YOLO, and SSD, improve accuracy. Study [2] focuses on YOLO for swift and precise object detection in images and videos, analyzing YOLOv3 and Yolo3-tiny. Paper [45] highlights YOLO's speed: base YOLO handles 45 frames/sec, Fast YOLO hits 155 frames/sec, with higher mAP. YOLO excels by reducing false positives on backgrounds and generalizing across domains, even shifting from natural images to diverse contexts.

In Inland waterways, video surveillance is crucial for ship detection, ensuring safety and automatic identification. A new algorithm for ship detection on waterways uses stationary cameras and works in variable lighting conditions. The algorithm in [19] detects all moving ships, eliminating non-ships using logic rules. The paper [28] presents a novel method for detecting and tracking ships in marine transportation systems using machine vision. The method involves two stages: the detection stage, where edges are calculated, and the tracking stage, where bounding rectangles are selected and judged as ships. A comprehensive review of academic literature [42] on maritime accident risk assessment reveals the potential of supervised machine learning and big data applications. Challenges include dataset availability, transparency, model development, and results evaluation, highlighting both novel applications and challenges. The study [12] presents a real-time object recognition and tracking system for remote video surveillance, using a statistical morphological skeleton for low computational complexity, localization accuracy, and noise robustness. The system compares an analytical approximation of the skeleton function with model objects and uses an extended Kalman filter for tracking. This paper [40] focuses on enhancing maritime traffic surveillance in inland waterways by integrating AIS and visual data. AIS provides vessel identity and position data, while cameras offer visual information but lack detailed vessel parameters. The proposed method employs anti-occlusion vessel tracking, synchronizes AIS and visual data, and utilizes multi-feature similarity measurements to robustly fuse them. The approach is evaluated using a new multi-sensor dataset and demonstrates improved maritime traffic surveillance, addressing vessel occlusion issues and enhancing safety and efficiency in inland waterway traffic.

2.3. Data spoofing with AIS data

This investigation delves into strategies to detect and mitigate spoofing, safeguarding the integrity of AIS data and bolstering maritime security. The current methods to detect spoofing are identified in the following literature. The ISOLA (Integrated System for Onboard Lineage Analysis) system as suggested in the work [48] enhances maritime vessel tracking by using advanced data analysis and machine learning to detect spoofing attacks in AIS data to establish vessel behaviour baselines, identifies anomalies, and validates data through external sources, bolstering maritime security. The authors of [16] suggest the usage of Doppler frequencies to detect spoofing, where the algorithm establishes theoretical cones based on the position of the satellite and velocity.

This study proposes a method to validate AIS data using drone-captured video imagery. By employing an enhanced algorithm for ship image extraction and position calculation, the approach cross-references AIS data with derived positions to ensure accuracy and authenticity. Experimental results in Shanghai's inland waterways show real-time UAV verification enhances maritime supervision [61].

The study [20] proposes merging spaceborne Synthetic Aperture Radar and Cooperative AIS data for improved ship detection. It explores the relationship between AIS data gaps and SAR-based detection using point-to-point and point-to-line associations. By using filtered and classified AIS transmissions, SAR positions are predicted to identify and match detected targets. A practical case study demonstrates this approach's effectiveness in regions with AIS blackouts, using Sentinel-1 satellite imagery and AIS data in the central Adriatic Sea.

To combat data processing challenges, including errors, noise, and gaps, compromising data quality for maritime safety research, innovative approaches have emerged. The study [59] suggests Kinematic features filter out noise, Deep Kernel Convolution identifies anomalies, and piecewise cubic spline interpolation reconstructs missing data. These techniques collectively enhance data quality, ensuring more reliable AIS data for critical maritime applications. The paper [23] proposes a method based on data mining and clustering methods combined with an integrity assessment of AIS messages for anomaly detection, with a proposition of software architecture for data processing done both on the fly and with archived data. A data mining approach is proposed in [47] for the probabilistic characterization of maritime traffic and anomaly detection off the continental coast of Portugal. The approach groups historical traffic data based on ship types, sizes, and final destinations. It identifies waypoints along a route where significant changes in navigational behaviour are observed, using trajectory compression and clustering algorithms. This vector-based representation of ship routes facilitates route probabilistic characterization and anomaly detection. The approach uses the Douglas and Peucker algorithm to detect heading changes and clusters them using density-based spatial clustering. The method is applied to southbound maritime traffic from Cape Roca to Lisbon, Setúbal, and Sines.

In conclusion, the literature reviewed showcases the multifaceted advancements in maritime domain research, particularly in the areas of AIS data quality assessment, object detection, and countermeasures against data spoofing. The surveyed literature eloquently addresses the first research question, shedding light on the current state-of-the-art methodologies.

2.4. Comparitive Analysis of Related Studies

This section aims to compare the findings of the study to significant works in the field, emphasizing our distinct contributions. This analysis emphasizes the importance of our work. The table 2.1 compares the proposed model to approaches employed in the current literature that are thought to be relevant to this study. The study [11] employs SSD and Yolo models for detection, but the proposed model simply employs Yolo and achieves efficiency in the range specified by the study of 63-90%.

The particle filter method is used in the Research article [28] to track vessels in the frame, which means that it represents an object's position by a set of particles and continuously updates its positions based on measurements and motion models, providing an estimation of the object's location and uncertainty over time. The centroid method was chosen, which will be detailed in full along with the justification for selection in chapter 4. In the publication [31], Haversine distance is used to assess the authenticity of the AIS message, however, our approach is based on employing an object detection run on video surveillance data to detect the mismatch or abnormality in the AIS data.

The study [46] suggests the collaboration of data sources such as Radar(SAR) and AIS data to verify the transmitted signal; similarly, we intend to do so in the combination of object detection algorithm and AIS data sources, which will allow us to compare the results with the data and, of course, find data mismatches.

In a study conducted in Shanghai, China, as described in [61], aerial imagery captured by a drone was utilized to validate transmitted data, revealing the presence of dark vessels. Our proposed model shares a fundamental goal with this study — data verification through image analysis. However, our model extends this concept to real-time camera-generated video surveillance in marine environments.

Citation	Paper	Method	Proposed Model
[11]	Object Detection and Classification from a Real-Time Video Using SSD and YOLO models	Single shot Detector and Yolo model	Yolo model
[28]	Examination of automatic detection and tracking of ships on camera in marine environment	Particle filter method for tracking	Centroid Method
[31]	Countering Real-Time Stream Poisoning: An Architecture for Detecting Vessel Spoofing in Streams of AIS Data	Haversine Distance on the dataset	Object detection on Video Surveillance
[46]	Mapping Dark Shipping Zones Using Multi-Temporal SAR and AIS Data for Maritime Domain Awareness	Combination of two data sources in SAR and AIS	Video surveillance and AIS data
[61]	Verification of AIS Data by using Video Images taken by a UAV	Drone captured imagery	Camera generated video

Table 2.1: Comparison of state-of-the-art vs. the proposed model

2.5. Method Proposed in this work

This section helps elucidate the rationale behind our chosen methodology and its alignment with the current state-of-the-art in object detection, including automated vehicle detection. By carefully justifying our approach, this section aims to demonstrate the relevance, effectiveness, and potential for innovation within the context of the broader research landscape. This rationale not only underpins our research direction but also provides critical insights into how the study's methodology addresses the intricacies of the research problem.

Similarities emerge between the study [5], which focuses on the identification of cars in traffic using video surveillance from a camera mounted on the inside of the windshield, and the research conducted in the article [33], which centres around a traffic monitoring system employing computer vision techniques to detect overspeeding vehicles using multiple reference lines and the total duration of frames the vehicle took to cross the reference lines help determine the speed. While the specific objectives and motivations of these studies differ, a commonality lies in the surface-level methodology of using a YOLO model and approach to addressing the respective research questions. Both studies leverage computer vision and real-time video analysis to monitor and classify vehicles on the road, exemplifying the versatility and applicability of such methods across various traffic-related research domains.

Furthermore, the cited studies, such as the work[5] and [33], provide valuable insights into the application of computer vision and object detection techniques in real-time traffic monitoring scenarios, which bears resemblance to the core approach undertaken in this research. Although the specific research objectives may differ—car detection in traffic in their cases and vessel detection in ours—the foundational methodologies and the use of object detection models share common ground. These studies served as a source of inspiration, guiding our approach toward creating an object detection model and training it to accurately identify vessels, aligning our research with the best practices established in the broader context of automated detection of objects in dynamic environments. The lessons learned from these studies facilitated the development of a robust framework for vessel detection, reinforcing the relevance and effectiveness of our approach.

3
AIS data validation

Inland Waterway Transportation is vital for global commerce, but its complex traffic demands advanced analysis. This research centres on using object detection to identify vessels, augmenting existing methods, notably the Automatic Identification System (AIS). Object detection, a cutting-edge technology, offers opportunities to enhance safety and navigation by comparing its results with AIS data. The sections that follow will embark on a comprehensive exploration of key facets within the realm of IWT, AIS data, the discerned gap in existing literature, and the overarching goal of this thesis.

3.1. Inland Water Transport

In modern economies and transportation networks, inland water transport is essential. Its significance is multifaceted and extensive. To begin with, IWT is well-known for its exceptional environmental sustainability. It emits significantly fewer greenhouse gases than road and rail transportation, making it a more environmentally friendly choice in an era of growing environmental awareness and concerns. Second, IWT is an important artery for the transportation of bulk goods and commodities, such as raw materials and agricultural products. Its ability to efficiently transport large volumes of cargo contributes to cost savings and economic competitiveness. Furthermore, IWT promotes regional development by connecting landlocked areas to international markets, thereby promoting economic growth and decreasing regional disparities. Furthermore, it reduces traffic congestion and infrastructure wear and tear, resulting in cost savings for governments and taxpayers. In conclusion, inland water transport is more than just a mode of transportation; it is a sustainable, cost-effective, and necessary component of global trade, economic development, and environmental stewardship.

3.2. Automatic Identification System

AIS data has become a critical component of modern maritime traffic management and vessel tracking. Its significance cannot be overstated because it provides real-time information on vessel positions, speeds, and identities, which improves navigational safety, collision avoidance, and search and rescue efforts. AIS data is a valuable tool for maritime authorities, port operators, and shipping companies, and it significantly contributes to efficient traffic management in IWT. However, it is critical to recognize its limitations, particularly in terms of spoofing. Malicious actors can intentionally manipulate AIS signals to transmit false vessel information, such as location or identity, resulting in potentially hazardous situations. The vulnerability of AIS to spoofing emphasizes the need for complementary technologies and verification methods to ensure the accuracy and integrity of vessel tracking data, especially in critical IWT scenarios where safety is critical. It is critical for the continued reliability of maritime traffic management systems to develop robust mechanisms for detecting and mitigating spoofing attacks on AIS data.

3.3. Gap in Literature

The literature on AIS data is extensive in terms of its significance, future applicability, and data spoofing and misuse. This system was designed to be used in Navigation, Collision Risk Assessment, Anomaly Detection, Trajectory Analysis, Vehicle Emission Analysis, and Fisheries [24] data. A study on the quality of AIS data provided insight into the current methods for detecting spoofed or mismatched data, and relevant literature is also discussed in chapter 2. This study identified a gap in the literature by demonstrating a lack of research on the use of video surveillance data to detect and confirm the presence of overlapping AIS data, which aided in the formulation of the research question ”How can video surveillance aid in identifying and reducing instances of overlapping or spoofed Automatic Identification System (AIS) data in maritime environments?” Sub-research questions were formed to be able to answer the above research question as stated in the chapter 1.

3.4. Goal of the thesis

The project's goal is to identify vessels in a video surveillance feed by tracking them across the frame and counting them based on a point of reference set on the video frame. The time of crossing of the above vessels would then be compared to the timestamps in the extracted available AIS data and the preceding results would then be saved in an excel file. The steps to achieving the established goal will be covered in the chapter 4.

3.5. Conclusion

In conclusion, this chapter underscores the pivotal role of Inland Water Transport (IWT) and the significance of AIS data within this domain. It highlights not only the pivotal role AIS data plays but also the risks posed by its misuse, emphasizing the core problem under investigation. Additionally, this chapter establishes the existence of a notable gap in the existing literature, identified through a comprehensive review, and sets forth the overarching goal of this study.

4 Architecture and Algorithm Design

The study emphasizes the importance of accurate vessel data for maritime navigation and security, utilizing the Automatic Identification System (AIS) for real-time ship tracking. The research aims to develop an algorithm that detects and corrects issues like fake or overlapping data, considering factors like accuracy, real-time processing, scalability, and integration with video surveillance. The methodical approach includes algorithm selection, data collection, design, analysis, accuracy assessment, trade-offs, validation, and testing.

4.1. Data collection and Algorithm Design Process

The conduct of this thesis project required the collection of two sets of data: video footage of a marine corridor and corresponding AIS data for the same marine corridor and timing as the video. The initial segment of the graduation project is centred around the Rhine Corridor in Germany. The aforementioned data sources are categorized as secondary forms of data, implying that they were not gathered or generated specifically for this project, but rather are pre-existing datasets. The picture 4.1 below denotes the location of the camera, it is located in the level tower on the Rhine promenade, Vor dem Rheintor 5, 46459 Rees, Germany and on the bank of the Rhine River. The blue lines show the field view of the Rhine River that is captured through the camera. The AIS data is collected for a radius of 15 km from the position of the camera, as marked in the figure. Figure 4.3 that is displayed shows the field of view that is available from the camera placed in 4.1, and where the object detection is performed in this project.

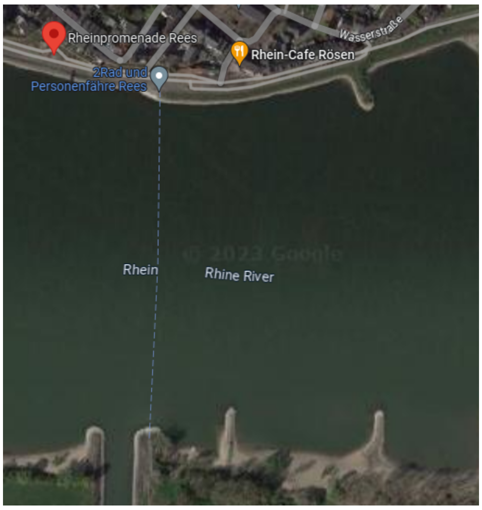


Figure 4.1: Position of the Camera [13]

The project's workflow is carefully structured as shown in 4.2 across four distinct work environments to facilitate the detection and analysis of vessels within a maritime context. Each environment plays a crucial role in handling specific tasks and processing data effectively. Firstly, Amazon Web Services (AWS) provides the foundation for the project's operations. A virtual server hosted on AWS is responsible for recording and downloading video footage. This initial step is fundamental as video data serves as a primary source for vessel detection and analysis.

Once the video is acquired, it's stored on Google Drive, forming the second environment in the process. Google Drive acts as a central repository for project data, ensuring easy access and collaboration across various stages of the workflow.

The third environment, Roboflow, plays a pivotal role in dataset creation and preparation. Vessel images are uploaded to Roboflow, and meticulous annotation is performed to identify vessels within each image. Additional post-processing steps, such as adding blur and noise, are applied to enhance the dataset's quality and diversity. Once these steps are completed, the dataset is generated and exported to the drive.

For the days when AIS data is required, it's obtained via an API. AIS data files are then copied to the hard drive, adding another layer of critical information to the project. The fourth environment, Google Colab, serves as the computational powerhouse for running code and performing data comparisons. Here, the Object Detection method is employed, leveraging the YOLOv5 model. The YOLOv5 model is trained on the dataset, resulting in the creation of weights files. The accompanying Readme file [49] provides detailed instructions on the code execution process.

Throughout the project's logic and code development, a key focus is flexibility, ensuring that the solution can be applied across diverse maritime environments rather than being limited to specific cases. The ultimate goal is to create a versatile solution that can benefit a broad user base.

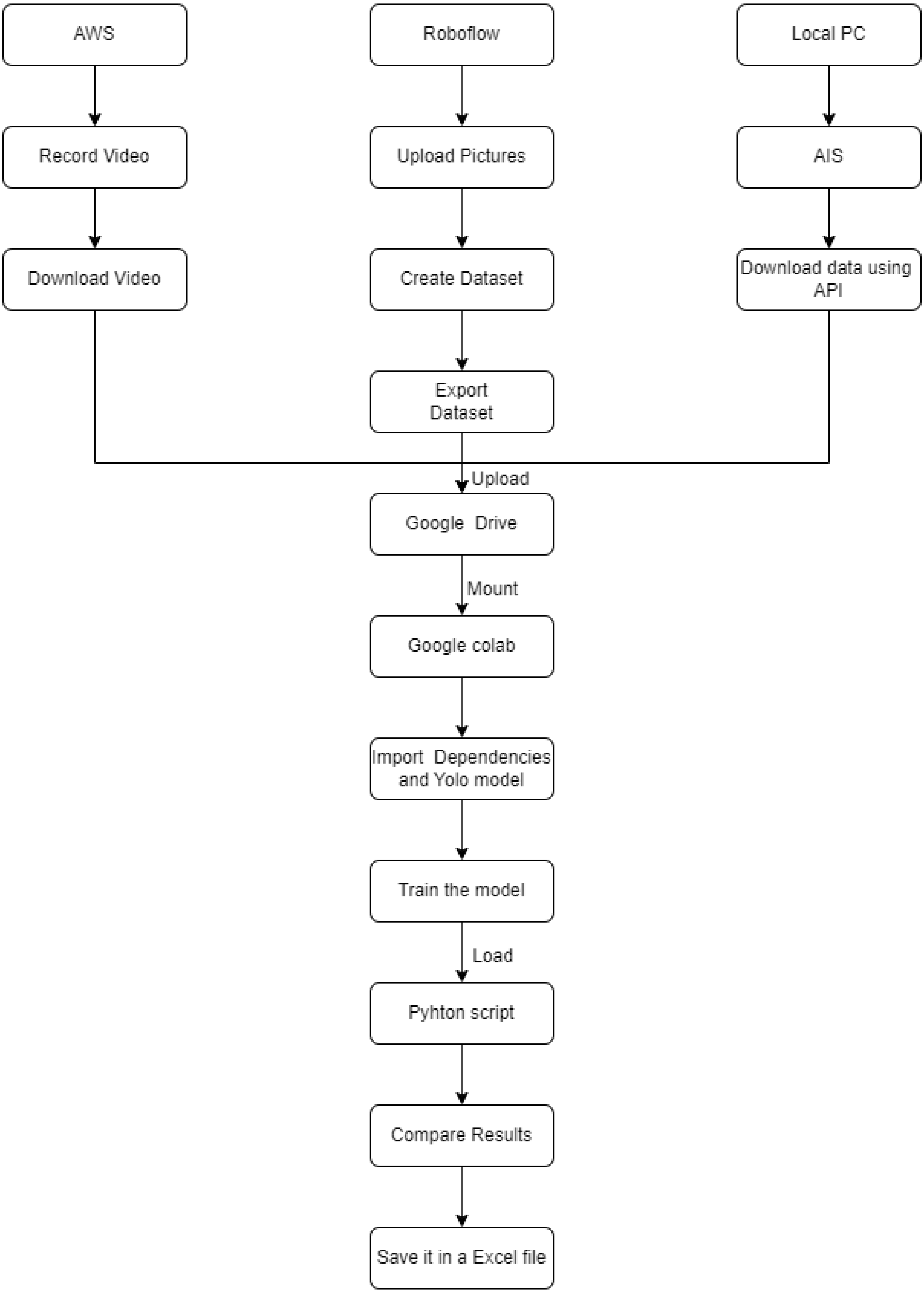


Figure 4.2: Process Architecture

Figure 4.3: Field of view

4.1.1. Tracking of the vessels

Object tracking is a pivotal component of computer vision and surveillance systems, enabling the continuous monitoring and analysis of object movement and behaviour. To detect ships in the video and count them precisely. Two distinct methods emerged for this purpose: the Centroid Method and the Intersection over Union (IOU) Method.

The centroid detection method is a straightforward yet effective approach to object detection. In this method, instead of outlining the entire object with bounding boxes, the algorithm identifies and marks the centroid of the object. The centroid represents the object's centre of mass or average position. To detect an object's centroid, the algorithm typically calculates the spatial average of the object's pixels or features. This method is especially useful for detecting objects of varying sizes, shapes, and orientations, as it provides a precise point of reference within the object.

IOU, or Intersection over Union, is a crucial metric used in object detection to evaluate the accuracy of bounding box predictions. It measures the degree of overlap between the predicted bounding box and the ground truth bounding box for an object. Specifically, IOU is calculated as the ratio of the area of intersection between the two bounding boxes to the area of their union. A higher IOU score indicates a better alignment between the predicted and true bounding boxes. IOU is often used to assess the quality of object detection models during training and evaluation, helping to determine whether a predicted bounding box sufficiently covers the actual object.

Centroid-based object tracking is favoured over Intersection over Union (IOU) primarily because it provides a more robust and continuous means of tracking objects in various scenarios. Unlike IOU, which relies on bounding box overlap and can be sensitive to object size changes and occlusions, centroid tracking focuses on the centre point of objects. This allows for smoother tracking even when objects undergo scale variations or temporary obstructions, making it a versatile and reliable choice in dynamic environments. Counters were installed to help ships count when they crossed a predefined reference point. Furthermore, the trajectory of centroid values was critical in determining the direction of each ship. The time of crossing the reference line was recorded using the video's initial timestamp, which was included in the input data. This methodical approach not only allowed for accurate ship tracking and counting but also provided information on ship directions and crossing times. The crossing time is saved and used to compare the AIS data's 'timelastupdate' to obtain the results. These results are then stored in an Excel file.

4.2. Data Analysis

The obtained AIS data is subject to a comprehensive comparative analysis and anomaly detection examination. This analytical process involves juxtaposing the outcomes of object detection with the acquired AIS data. The path delineated by the acquired AIS data serves as a crucial input, both for comparative assessment and for conducting anomaly detection analyses. This dataset is organized into a structured data frame, facilitating systematic analysis. The "timelastupdate" column, which holds temporal information, is subsequently transformed into a local time format for enhanced interpretability. To enhance data integrity, a vital step involves the removal of duplicate entries associated with the same MMSI. This de-duplication process ensures data accuracy and prevents redundant information. In scenarios where multiple entries correspond to the same MMSI, the "timelastupdate" column is adjusted to reflect the temporal range spanning the first and last entry for that specific MMSI.

Subsequently, this refined and processed AIS data is juxtaposed with the time instances when vessels cross designated reference lines captured in the video. By aligning the AIS data with the video-derived crossing times, the analysis endeavours to identify any anomalies or deviations in vessel behaviour and movement. This step serves as a pivotal juncture for anomaly detection, where irregularities in vessel trajectories, crossing times, or other relevant parameters can be scrutinized and potentially flagged for further investigation. In essence, this analytical process forms the core of the comparative and anomaly detection analysis, harnessing the synergy between object detection results and acquired AIS data to unveil noteworthy patterns, deviations, and potential anomalies within maritime scenarios.

MMSI, location, speed, course, heading, status, vessel type, dimensions, draught, destination, projected arrival time, safety messages, cargo details (for some), and static information (vessel name, call sign, registration country) could all be present in the unpacked AIS data. The entities required for comparison in this project are MMSIs, and the time since the last update is necessary. Despite the fact that the whole data file is transferred into a data frame for the procedure, only the fields listed above are used in the final results column.

4.3. Accuracy and False positive/Negatives Trade-offs

The realm of object detection introduces the critical concepts of False Negatives and False Positives, which are inherently intertwined with the accuracy of the model. A False Negative emerges when the model fails to identify an object that indeed exists in the image. In simpler terms, the model overlooks a genuine object. This circumstance poses a challenge as it indicates that the model is not encompassing all relevant objects present in the scene. False Negatives can lead to missed opportunities and a reduction in recall – a metric gauging the proportion of true objects successfully detected. Conversely, a False Positive transpires when the model erroneously predicts the presence of an object that is absent from the image. This situation results in the model erroneously flagging objects that are not actually there, consequently introducing inaccuracies in the reported information. False Positives can introduce noise, diminish precision, and impose unnecessary computational burdens.

Achieving equilibrium between false negatives and false positives is of paramount importance in developing a dependable and accurate object detection model. The optimization process involves striking a delicate balance. The reduction of false positives contributes to refining precision, while the minimization of false negatives augments recall. A pivotal metric, the F1-score, harmonizes both precision and recall, providing a comprehensive evaluation of object detection model performance. The process of striking the balance between false negatives and positives hinges on adjusting the confidence threshold. A low threshold may inadvertently result in an abundance of false positives, whereas an excessively high threshold could lead to a rise in false negatives. Therefore, finding the right equilibrium becomes pivotal. It's about identifying the optimal balance that yields the most favourable outcomes, ensuring the model's performance aligns with the complex interplay between precision, recall, and the overarching accuracy of object detection.

4.4. Validation algorithm

This section delves into the practical implementation of our ship detection and tracking methodology. As discussed in earlier chapters, the automated detection and tracking of ships in maritime videos hold immense importance for various applications, ranging from vessel traffic monitoring to maritime security. To achieve this goal, a comprehensive code implementation that employs a combination of computer vision techniques and machine learning models is presented. The code is designed to process video data, detect ships, track their movement, and determine their crossing direction with respect to a reference line. This chapter provides an in-depth breakdown of the code structure, functions, and logic employed, aiming to offer readers a clear understanding of how our proposed methodology translates into practical execution. Detailed explanations of the code blocks help highlight the critical stages of the ship-tracking approach and elucidate the role of each component in achieving accurate and efficient results. The chapter concludes by summarizing the objectives of the code implementation and setting the stage for further exploration and analysis of the obtained outcomes. The complete algorithm is attached as part of the Appendix B.

4.4.1. Functions

Algorithm 1 Functions

function calculate_centroid(box)
    $x1, y1, x2, y2 \leftarrow$ box
    centroid_x $\leftarrow (x1 + x2)/2$
    centroid_y $\leftarrow (y1 + y2)/2$
    return (centroid_x, centroid_y)
end function
function calculate_distance(centroid1, centroid2)
    $x1, y1 \leftarrow$ centroid1
    $x2, y2 \leftarrow$ centroid2
    distance $\leftarrow \sqrt{(x2 - x1)^2 + (y2 - y1)^2}$
    return distance
end function

Within the provided code, two critical functions serve as the backbone of ship tracking and counting operations. The first function, $calculate\_centroid(box)$, operates by taking a box's coordinates (x1, y1, x2, y2) as input. It then proceeds to compute the centroid's coordinates by averaging the corners of the box. Specifically, it calculates the midpoint on the x-axis and the midpoint on the y-axis, returning these values as the centroid's coordinates. The second function, $calculate\_distance(centroid1, centroid2)$, is equally instrumental. It accepts two sets of centroid coordinates, denoted as (x1, y1) and (x2, y2), and calculates the Euclidean distance between them. This distance is computed using the square root of the sum of squared differences in both x and y coordinates, providing an accurate measure of the spatial separation between two centroids. These two functions cooperatively enable the code to precisely track and count ships, offering a foundation for various spatial analytics and applications.

4.4.2. Loading model and Video Processing

This snippet presents a continuous loop designed to process video frames utilizing the YOLOv5s model for object detection. The loop is configured to operate indefinitely ("WHILE True") until the video reaches its conclusion. During each iteration, a video frame is read from the designated source (cap.read()). In cases where the frame is empty or devoid of content, the loop is terminated. The frame number is extracted and retained for reference. Subsequently, the code assesses whether the current frame number corresponds to a multiple of $skip\_frames$. If this condition is not met, the loop proceeds to the next iteration. Simultaneously, the count of processed frames is incremented. Employing the YOLOv5s model, object detection is conducted on the present frame. Detected objects that possess confidence scores surpassing $conf\_thresh$ are collated and stored within the variable "detections." This code segment lays the groundwork for the initial stages of object detection, priming the frame data for subsequent analysis or display.

Algorithm 2 Processing Video Frames

```
while True do
    ret, frame ← cap.read()
    if frame is None then
        break
    end if
    if frame_number mod skip_frames ≠ 0 then
        continue
    end if
    detections ← model(frame)
    detections ← detections.pred[0][detections.pred[0][:, 4] > conf_thresh]
end while
```

4.4.3. Ship Detection and Tracking

The Algorithm provided plays a pivotal role in a video processing system, focusing on object matching and tracking. Its core purpose is to process a sequence of object detections, assess their confidence levels, and determine whether each detection corresponds to a previously tracked object or represents a new one. This functionality is vital in scenarios where objects, such as ships, require continuous monitoring and tracking as they traverse a video stream. One of the code's key functions is the confidence-based filtering of object detections. Each detection is assigned a confidence score, indicating the algorithm's confidence in the presence of the object. Detections with confidence scores below a specified threshold, referred to as conf_thresh, are filtered out.

Algorithm 3 Detections and Tracking

```
for all detection in detections do
    confidence ← detection[4]
    if confidence > conf_thresh then
        get → x1, y1, x2, y2
        centroid_x, centroid_y ← calculate_centroid((x1, y1, x2, y2))
        matched_ship ← None
        previous_centroid ← None
        for all ship in previous_ships do
            previous_centroid ← ship[”centroid”]
            if previous_centroid is an array then
                for all centroid in previous_centroid do
                    distance ← calculate_distance(centroid, (centroid_x, centroid_y))
                    if distance < centroid_distance_threshold then
                        previous_centroid ← centroid
                        matched_ship ← ship
                        break
                    end if
                end for
            else
                distance ← calculate_distance(previous_centroid, (centroid_x, centroid_y))
                if distance < centroid_distance_threshold then
                    matched_ship ← ship
                    break
                end if
            end if
        end for
        if matched_ship is not None then
            if previous_centroid is an array then
                for all centroid in previous_centroid do
                    if calculate_distance(centroid, (centroid_x, centroid_y)) < centroid_distance_threshold then
                        matched_ship[”centroid”] ← (centroid_x, centroid_y)
                        break
                    end if
                end for
            else
                matched_ship[”centroid”] ← (centroid_x, centroid_y)
            end if
        end if
        if matched_ship is None then
            matched_ship ← {”centroid” : centroid_x, centroid_y), ”direction” : None, ”counted” : False}
            previous_ships.APPEND(matched_ship)
        end if
    end if
end for
```

This step ensures that only highly reliable detections are considered for tracking, minimizing false positives and enhancing tracking accuracy.

The code snippet computes the centroid, which represents the central point, of each detected object's bounding box. This centroid calculation provides a stable reference point for tracking purposes. Subsequently, it proceeds to match the centroids of newly detected objects with those of previously tracked objects. This matching process involves measuring the distance between centroids and assessing whether they fall within a certain threshold, known as centroid_distance_threshold. Successful matches indicate that the detected object corresponds to an already known object, facilitating its continuous tracking.

In cases where no match is established between a newly detected object and any previously tracked object, the code initializes tracking for the new object. It creates a new object entity that includes essential information such as its centroid, direction (initially set to None), and counting status (initially set to False). This new object is then added to the list of previously tracked objects, allowing for its ongoing monitoring and tracking as it appears in subsequent frames.

In summary, this code snippet serves as a crucial component of an object tracking and counting system. Its primary role is to associate detections with known objects, update their tracking information, and initialize tracking for new objects. This systematic approach ensures the accurate tracking and counting of objects, thereby enhancing the capabilities of video analysis and surveillance systems in a variety of applications.

4.4.4. Direction

The provided snippet encompasses a conditional statement geared towards assessing the existence of a $matched\_ship$ object and the unassigned status of its "direction" attribute. Should these criteria align, the code proceeds to extract the coordinates of the preceding centroid, denoted as $prev\_centroid\_x$ and $prev\_centroid\_y$. Within this context, the code undertakes an evaluation to discern whether the current centroid's coordinates suggest a left-to-right movement. This assessment hinges upon a comparison between the x and y coordinates of the present centroid and those of the previous centroid. Specifically, if both the x and y values of the current centroid surpass their corresponding values in the previous centroid, it signifies a motion from left to right. In this scenario, the "direction" attribute of the $matched\_ship$ object is assigned the value "Left to Right", effectively indicating that the ship is progressing from left to right. Conversely, if the aforementioned condition is not satisfied, the code ascribes the opposite direction to the ship, thereby encompassing the full spectrum of potential movement directions. This comprehensive approach ensures that the ship's movement direction is accurately captured based on the spatial relationship between the current and previous centroids.

Algorithm 4 Check and Set Ship Direction

if matched_ship $\neq$ None and matched_ship[direction] is None **then**
    $prev\_centroid\_x, prev\_centroid\_y \leftarrow$ previous_centroid
    `# Check if the centroid's coordinates indicate Left to Right direction`
    if centroid_x $>$ prev_centroid_x and centroid_y $>$ prev_centroid_y **then**
        matched_ship[direction] $\leftarrow$ "Left to Right"
    end if
end if

4.4.5. Counting

This code segment encapsulates a conditional statement that evaluates three distinct criteria. Firstly, it verifies whether the ship's designated direction is labelled as "Right to Left." Subsequently, it examines if the leftmost x-coordinate of the ship's bounding box (x_1) is smaller than a predetermined reference line value, denoted as ref_line. Lastly, the statement ensures that the ship's "counted" flag has not yet been marked as True.

Upon the fulfillment of all these conditions, a sequence of actions is executed. The code proceeds to increment a count that monitors the tally of ships exhibiting a right-to-left movement. Subsequently, the "counted" flag associated with the matched_ship object is updated to True. Additionally, the code generates a printed message, serving as an indication that the ship has successfully traversed the line from right to left. This code sequence succinctly manages the process of detecting and marking ships following a right-to-left trajectory as they cross the designated line of reference.

Algorithm 5 Update Count

```
1: if matched_ship[direction] = "Right to Left" and x1 < ref_line and matched_ship[counted] then
2:     right_to_left_count += 1
3:     matched_ship[counted] ← True
4: end if
```

4.4.6. Data Processing

The algorithm presented in this code snippet focuses on the crucial task of matching and counting ships within a data frame. Its primary goal is to determine if a ship, previously identified and stored in the 'previous_ships' list, corresponds to any records in the DataFrame 'df'. This process involves evaluating both temporal attributes and certain conditions to establish matches. The code employs a nested loop structure to systematically examine each ship in the 'previous_ships' list and compare it with rows in the DataFrame. The following steps outline the core matching process:

1. Checking for Prior Counting:
Within the nested loop, the code first checks whether the ship has already been counted. This verification ensures that each ship is considered for matching only once.

2. Temporal Attribute Conversion:
The code extracts and converts relevant time attributes for analysis. It transforms the "timeLastUpdate" attribute from the DataFrame row into a datetime format. Additionally, it processes the ship's "time" attribute, representing the time of crossing a designated reference line.

3. Time-Based Matching Logic:
The code distinguishes between two scenarios:

- If "timeLastUpdate" in the DataFrame is represented as a dictionary (likely denoting a time range), the code compares whether the "time_of_crossing" falls within this specified time range. When a match is found, the code marks the corresponding row as a "Matched ship" in the DataFrame, increments the "Number of matches" count, and concludes the loop for that ship.
- In cases where "timeLastUpdate" in the DataFrame is not a dictionary, the code checks if the "time_of_crossing" is within a two-minute window (plus or minus) of the "timeLastUp-

date.” Successful matches result in the same actions: marking the row as a ”Matched ship,” increasing the ”Number of matches” count, and exiting the loop.

In summary, this code segment plays a vital role in the object tracking and counting system, specifically in the context of ship monitoring. It effectively matches ships stored in ‘previous_ships‘ with relevant rows in the Data Frame, taking into account temporal attributes and predefined conditions. This systematic approach enhances the tracking and counting of ships within the dataset, facilitating comprehensive analysis and monitoring of maritime activities.

Algorithm 6 Matching Ships in DataFrame

```
for all ship in previous_ships do
    for all index, row in df.iterrows() do
        if ship[”counted”] then
            # Convert timeLastUpdate to datetime
            timeLastUpdate ← row[”timeLastUpdate”]
            # Convert time of crossing the line to datetime
            time_of_crossing ← datetime.datetime.strptime(ship[”time”], ”%Y-%m-%d %H:%M:%S”)
            if isinstance(row[”'timeLastUpdate' : ['first','last']”], dict) then
                if time_of_crossing ≥ row[”'timeLastUpdate' : ['first','last']”][”first”] and time_of_crossing ≤ row[”'timeLastUpdate' : ['first','last']”][”last”] then
                    df.at[index, ”Matched ship”] ← ”Yes”
                    df.at[index, ”Number of matches”] ← df.at[index, ”Number of matches”]+ 1
                    # PRINT "Matched ship: " + str(ship) + " with " + str(row["mmsi"])
                    break
                end if
            else
                if time_of_crossing ≥ timeLastUpdate - datetime.timedelta(minutes=2) ∧ time_of_crossing ≤ timeLastUpdate + datetime.timedelta(minutes=2) then
                    df.at[index, ”Matched ship”] ← ”Yes”
                    df.at[index, ”Number of matches”] ← df.at[index, ”Number of matches”]+ 1
                    # PRINT "Matched ship: " + str(ship) + " with " + str(row["mmsi"])
                    break
                end if
            end if
        end if
    end for
end for
```

4.5. Validation and Testing

The validation and testing phase commences with the creation of succinct code snippets meticulously designed for testing the algorithm’s functionality and results. To delve deeper into the outcome analysis, a Design Chart Exploration is undertaken. This exploration involves probing the interactions between key variables, namely the Confidence Threshold, Centroid Distance Threshold, Number of Frames Processed, and the Discrepancy between Actual and

Detected Counts. These variables exhibit interconnectedness in terms of their influence on the generated results. The Centroid Distance Threshold exhibits a correlation with the Frames Skipped variable, where an increase in the latter leads to a proportional elevation in the former. Similarly, the Confidence Threshold directly impacts the model's accuracy, influencing the propensity for false negatives and positives. A comprehensive Design Space Exploration chart is formulated, showcasing the outcomes across varying values of the aforementioned variables. This exploration extends to different snippets as well, amplifying the robustness of the analysis.

For each experimental value, accuracy is meticulously calculated, facilitating the identification of the optimal set of values that yield the most desirable algorithmic performance. This systematic approach empowers the testing of diverse value combinations, aiding in the determination of the most effective parameters for the algorithm. Furthermore, the validation process extends to the post-processing segment of the code, where results from object detection are meticulously compared. The previously generated snippets are leveraged to evaluate the efficacy of this comparison process and the subsequent interpretation of results within an Excel file.

4.6. Results

The results of the object detection phase of the code that is stored in the previous_ships dictionary are extracted to be compared with the AIS data that has been called for in the inputs and uploaded into a data frame. The previous_dictionary is used to store details like the centroid values of the ship, the direction of the ship if that has been assigned, a flag that sets itself to true if the ship has been accounted for, and the time that the ship crosses the line, which is calculated based on the starting time of the video that has been provided as one of the input arguments. The time of crossing of the ships is then compared with the 'timelastupdate' column of the AIS data in the data frame. Then an additional column is created to show 'yes' if there is a match between the entries in the data and the vessel in the video. Once a range was initialized for 'timelastupdate' it was also important to check the number of matches a particular entry had, to maximize the impact of the current results. This counter would later allow us to compare the total detections with the matches. A representation of the final table 4.1. The script also generates a plot at the end to depict the number of matches per MMSI/entry in the data.

timeLast Update	**'timeLastUpdate': ['first', 'last']**	**mmsi**	**Matched ship**	**Number of matches**
YYYY-MM-DD HH:MM	{'first': Timestamp('YYYY-MM-DD HH:MM'), 'last': Timestamp('YYYY-MM-DD HH:MM')}		Yes/No	

Table 4.1: Results table

5 Results and Discussions

This chapter of the study's research endeavours explores the potential for improved reliability of maritime data. This section is pivotal to our study, addressing Research Questions 3, 4, and 5. Question 3 (SQ3) prompts an in-depth analysis of our object detection algorithm's performance across diverse maritime conditions. This examination helps uncover both the strengths and limitations in ship identification and localization via video surveillance, showcasing its potential to enhance maritime safety and operational efficiency. Question 4 (SQ4) directs our attention to the intricacies within the data, compelling a quantitative evaluation of disparities between Automatic Identification System (AIS) data and algorithm-generated ship detections. This scrutiny deepens our grasp of real-world data variations, advancing strategies for precision improvement. Lastly, Question 5 (SQ5) guides us to explore the intersection of video surveillance and AIS data, shedding light on the challenges and opportunities of detecting falsified AIS information while integrating AIS into operational practices. This chapter bridges practical observations with theoretical foundations, yielding tangible results with wide-reaching implications for maritime reliability, precision, and integrity.

Throughout this study, data was collected and analyzed over three separate days to derive comprehensive results. Each day's analysis involved deploying the algorithm for ship detection and matching, using video surveillance and AIS data. The Videos acquired were for a period of 6 hours, for three days — May 31, August 6, and August 7, 2023, from the camera located in Rees as shown in figure 4.1. The model that is to be used for object detection was trained using a dataset created using over 100 images of the inland vessels in the Roboflow environment as mentioned in chapter 4. This is done keeping the factor of overfitting in mind, i.e., Having too many images, especially if they are similar, where the model performs well on the training data but poorly on new, unseen data. A detailed explanation of the process architecture is shown in figure 4.2. The process entailed running the algorithm on the specific day's video footage to detect ships in real-time. A Python script was developed to use the detection of the model and count based on the directions in accordance with a reference point that is set on the frame using the centroid point of the bounding box. To ensure accurate counting, the vessels had to be tracked across the frame to avoid being counted multiple times. To identify the directions, the code should have been able to save the last two centroid values of each vessel tracked so that the trend of these values could be used to determine the direction and their position on the screen concerning the reference line to increase the counters in that direction. The challenge was following the ships throughout the frames because, to make the processing shorter, not every frame could be handled, hence a parameter (explained in chapter4.4) named "centroid_distance_threshold" was developed to accommodate for this. Because this parameter takes into consideration the distance (pixels) travelled by vessels in the following frames, its value is exactly proportional to the number of frames omitted. It was difficult to find the value that would balance the false positives and negatives. Subsequently, matches were identified by comparing the algorithm's detections with corresponding entries in the AIS data. The number of detections and matches were recorded for each day. To ensure robustness and reliability, this procedure was repeated for the above three consecutive days. The collected data encompassed the number of ships detected and the instances where matches were successfully identified. These data sets were then meticulously analyzed to draw meaningful conclusions about the algorithm's performance, accuracy, and capacity to identify and mitigate instances of overlapping or fake AIS data. The aggregation of results from these three days provided a comprehensive and representative understanding of the algorithm's effectiveness across varied maritime conditions and scenarios.

5.1. Results

This section embarks on a detailed exploration of the outcomes derived from our research project's approach. Our analysis encompasses a thorough examination of the results for each of the three distinct days—May 31, August 6, and August 7. For each day, this study will meticulously present and discuss the number of ships detected, the corresponding matches identified in the AIS data, and any notable trends or patterns that emerged. By scrutinizing the results for each specific day, the results chapter aims to provide a comprehensive understanding of how our algorithm performed across different instances and timeframes. Following the presentation of individual day results, we will then engage in a broader discussion that delves into the implications of our findings in addressing our research questions. In the process critically evaluating the algorithm's accuracy, its capability to detect discrepancies between AIS data and detections, and its effectiveness in spotting instances of overlapping or fabricated AIS information. Furthermore, the following subsections will explore how these insights contribute to enhancing maritime navigation, safety, and security.

Figure 5.1: A snippet from the Detection

Manual verification played a crucial role in compensating for the inherent limitations and ensuring the reliability of our object detection model. While the model demonstrated impressive efficiency in detecting and localizing vessels, there were instances where manual verification became indispensable. The frames were processed was recorded and the vessels that the object detection model didn't detect were identified and the frame numbers of the vessels were noted. The frames using a Python script were converted to timestamps of the video and matched with the AIS data. By doing the manual verification process we are thereby account-ing for all the vessels in the video, and this helps find the discrepancies or the mismatches between the results and the AIS data. Finally, a graph is plotted between the number of matches acquired by object detection, manual verification and the unmatched data. Through this comprehensive analysis, this study aims to draw meaningful conclusions and highlight the potential real-world impact of our research.

5.1.1. Results for May 31, 2023

On May 31, 2023, the data collection process was initiated to evaluate the algorithm's performance in detecting and matching ships using video surveillance and AIS data. The algorithm was applied to the day's specific video footage, and its detections were compared with corresponding AIS data entries. This endeavour aimed to provide insights into the algorithm's effectiveness in real-time ship identification and its ability to uncover instances of overlapping or fake AIS data. The outcomes of this analysis, including the number of ships detected and the successful matches achieved, shed light on the algorithm's initial performance in diverse maritime conditions. A comprehensive assessment evaluated the algorithm's effectiveness in ship detection and matching using video surveillance and AIS data. Out of the 60 ships detected in the video, the analysis revealed that 16 of them were successfully matched with corresponding entries in the AIS data. This examination provides an initial understanding of the algorithm's performance in real-time ship identification, emphasizing its potential to identify instances of overlapping or fake AIS data in the maritime environment. Table 5.1 shows the statistics of the algorithm for the video of the first day, where it was identified manually that the total number of vessels crossing in those 6 hours was 83 and the object detection model managed to detect 60, which depicts an accuracy of 74%. The deficit shown by the model would be accounted for by manual verification.

Video Date	No.of Ships	Detected	Efficiency
31/05/2023	83	60	74%

Table 5.1: Object detection statistics (day1)

The Overall statistics after the completion of the manual verification process are presented in the table 5.2. The mentioned table shows the split between the total number of ships detected by the model which is 60. The manual verification process yielded another 29 vessels and when the obtained vessels were matches with the AIS data, they produced 34 matches.

Video Date	Detected by Model	Manually detected	Overall Matches with AIS data
31/05/2023	60	29	34

Table 5.2: Overall Statistics(31/05/2023)

The table 5.3 shows the results acquired from the video it shows the data from the AIS data. This is done by comparing the data acquired from the video and AISd data. To eliminate multiple entries of the same MMSI(s), the range of the first and final signals broadcast is taken. The matched ships are then compared and matched. This result shows a total of 16 ships that have been matched, and the highest falling to the timestamp ”2023-05-31 18:37:45” with the mmsi ”244690451” because of its huge range of timeLastUpdate': ['first', 'last'] = 'first': Timestamp('2023-05-31 18:37:45.312'), 'last': Timestamp('2023-05-31 20:15:35.713') which corresponds to about two hours of the video this directly correlates to the high number of matches in the matches done by the algorithm and also manually. The Entries without a range are provided with a threshold of 2 minutes on either side of their entry, so the time of crossing is checked in accordance and matched in the data frame. Here the matches for the entries without a range are depicted for the MMSI'(s): 244620968, 211667460. A graph 5.2 is plotted to show the number of matches per MMSI'(s) that were acquired for this particular day and particular time range.

timeLast Update	'timeLastUpdate': ['first', 'last']	mmsi	No. of matches	Manual
2023-05-31 14:42		205271390	0	
2023-05-31 15:14		244620968	4	
2023-05-31 16:15		244750327	0	
2023-05-31 16:19		211667460	2	1
2023-05-31 16:19		244790070	0	
2023-05-31 16:20		205248590	0	
2023-05-31 16:45		253242513	0	
2023-05-31 17:05	{'first': Timestamp('2023-05-31 17:05'), 'last': Timestamp('2023-05-31 17:18')}	244710657	0	1
2023-05-31 17:15		244690224	0	
2023-05-31 18:02		244077643	0	
2023-05-31 18:37	{'first': Timestamp('2023-05-31 18:37'), 'last': Timestamp('2023-05-31 20:15')}	244690451	10	15
2023-05-31 18:59		211115500	0	
2023-05-31 19:09	{'first': Timestamp('2023-05-31 19:09), 'last': Timestamp('2023-05-31 19:27')}	244670089	0	
2023-05-31 19:13		211476070	0	
2023-05-31 19:15	{'first': Timestamp('2023-05-31 19:15'), 'last': Timestamp('2023-05-31 19:46')}	211524940	0	
2023-05-31 19:32		226015720	0	
2023-05-31 19:40	{'first': Timestamp('2023-05-31 19:40'), 'last': Timestamp('2023-05-31 19:57')}	244700810	0	
2023-05-31 19:51	{'first': Timestamp('2023-05-31 19:51), 'last': Timestamp('2023-05-31 20:16')}	244660720	0	
2023-05-31 19:58'	{'first': Timestamp('2023-05-31 19:58'), 'last': Timestamp('2023-05-31 20:06')}	244850275	0	
2023-05-31 19:59	{'first': Timestamp('2023-05-31 19:59'), 'last': Timestamp('2023-05-31 20:02')}	244001635	0	

Table 5.3: Results - 31/05/2023

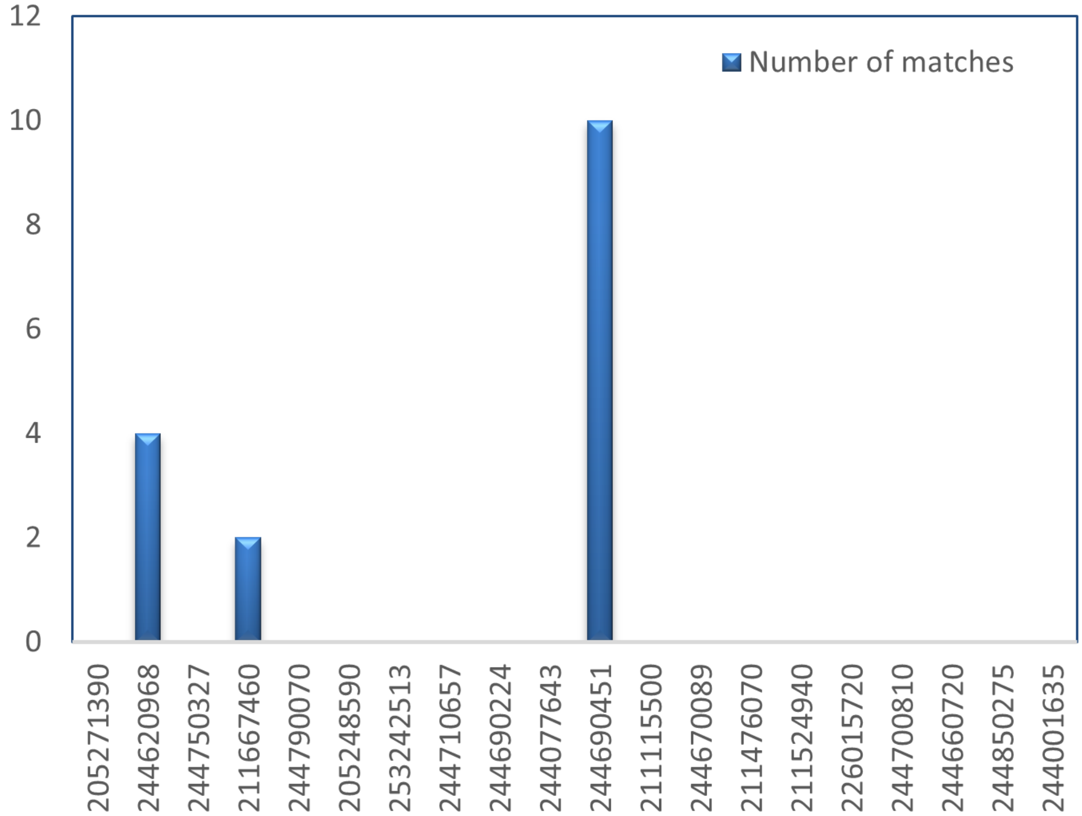


Figure 5.2: Matches per MMSI(s) - 31/05/2023

The Pie chart shown in figure 5.3 shows the split up of the total vessels identified, the same is shown in the tables 5.1,5.2. This chart shows the split between the total number of vessels that have been matched by the model, by manual verification and data that is unmatched by both processes. This helps understand the mismatch between the inland vessels travelling and the available AIS data. The unmatched data represented in the graph is both due to missing vessels and missing data.

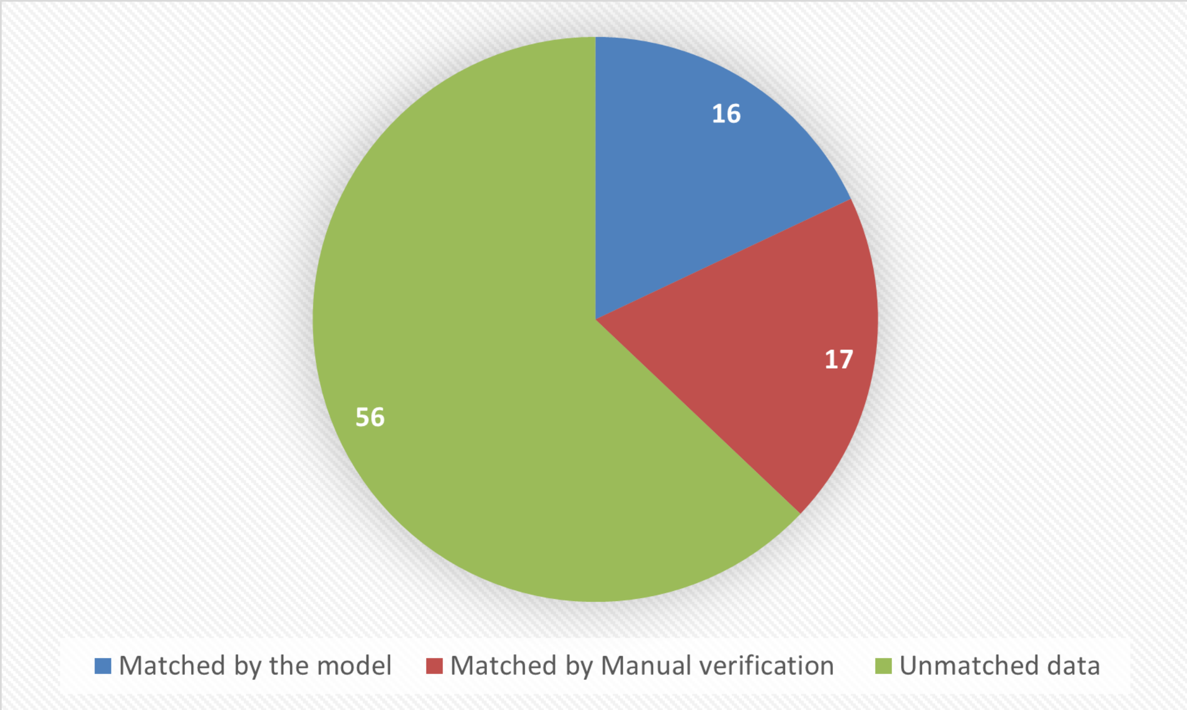


Figure 5.3: Comparison of data - 31/05/2023

5.1.2. Results for August 6, 2023

In the pursuit of robust and comprehensive insights, our investigation extends to the second day of analysis, namely August 6, 2023. This distinct day brings forth a new set of challenges and scenarios within the maritime environment, enriching our understanding of the proposed methodology's effectiveness. Delving into the results of this specific day helps gain a deeper perspective on the algorithm's performance and its ability to withstand varied conditions. By systematically scrutinizing different days, the study not only bolsters the credibility of our findings but also subjects our approach to diverse scenarios, fortifying its real-world applicability. The meticulous analysis of the second day's data stands as a testament to the adaptability and potency of our method in addressing the intricate intricacies of maritime data reliability and security. Table 5.4 shows the statistics of the algorithm for the video of the first day, where it was identified manually that the total number of vessels crossing in those 6 hours was 66 and the object detection model managed to detect 68, which depicts an accuracy of 102%. The excess in the detection model's results is because of false positives and on this particular day, there were situations where many vessels passed concurrently at the same time, which led to the generation of false positives. This could be neglected by the alteration of the parameters in the code but at the cost of the generation of false negatives. So the parameter that led to the attainment of the best possible balance between false positives and negatives was chosen.

Video Date	No.of Ships	Detected	Efficiency
06/08/2023	66	68	102%

Table 5.4: Object detection statistics (day2)

Following the meticulous manual verification process, the overarching statistics are thoughtfully outlined in Table 5.5. It's noteworthy that the total count of vessels detected stands at 66, which exceeds the actual number observed in the video. Depicts the total number of ships identified manually, here it stands at 18 vessels and the overall matches that have been attained including the manual verification which is 48, underscoring its efficacy in confirming vessel identities amid the intricacies of the maritime environment.

Video Date	Detected by Model	Manually detected	Overall Matches with AIS data
06/08/2023	68	18	48

Table 5.5: Overall Statistics(06/08/2023)

With a keen eye on the dynamics of this specific day, a comprehensive exploration of ship detection and matching results was embarked on. The statistics from this day stand as a testament to the algorithm's robustness and its ability to perform admirably across distinct maritime scenarios. The recorded count of 68 ship detections, of which 38 were successfully matched with AIS data, reinforces the efficacy of our approach in identifying and mitigating instances of overlapping or falsified data. Meticulously examining the results of this day, expands the understanding of the algorithm's performance within diverse maritime contexts, further reinforcing the credibility of our method and its potential to reshape the landscape of maritime data analytics. Similarly, the Range for the timelastupdate is attained to eliminate the duplication of MMSI(s) and the time of crossing is compared to the dataset of this particular day in 5.6. The highest number of ships have been detected and matched for the time range of 'first': Timestamp('2023-08-06 10:09:46.886'),'last': Timestamp('2023-08-06 10:40:16') with 12 ships matched and the next highest is matched with 8 detections for the range 'first': Timestamp('2023-08-06 11:45:18.339'), 'last': Timestamp('2023-08-06 12:24:27.212'). The matches for the timelastupdate of each MMSI are shown in 5.4.

timeLast Update	'timeLastUpdate': ['first', 'last']	mmsi	Number of matches	Manual
2023-08-06 10:09	{'first': Timestamp('2023-08-06 10:09'), 'last': Timestamp('2023-08-06 10:40')}	244615886	12	3
2023-08-06 10:34	{'first': Timestamp('2023-08-06 10:34'), 'last': Timestamp('2023-08-06 10:56')}	211353520	7	1
2023-08-06 10:43	{'first': Timestamp('2023-08-06 10:43'), 'last': Timestamp('2023-08-06 11:01')}	244660843	0	
2023-08-06 10:51		244630075	0	
2023-08-06 11:05	{'first': Timestamp('2023-08-06 11:05'), 'last': Timestamp('2023-08-06 11:23')}	269057800	2	
2023-08-06 11:08		211455520	0	
2023-08-06 11:36		211386900	0	
2023-08-06 11:45	{'first': Timestamp('2023-08-06 11:45'), 'last': Timestamp('2023-08-06 12:24')}	244620698	8	3
2023-08-06 11:48	{'first': Timestamp('2023-08-06 11:48'), 'last': Timestamp('2023-08-06 11:59')}	211510540	0	
2023-08-06 11:53		211667460	0	

2023-08-06 11:54		205268790	0	
2023-08-06 12:34		211328590	6	
2023-08-06 13:02		244660361	1	1
2023-08-06 14:10	{'first': Timestamp('2023-08-06 14:10'), 'last': Timestamp('2023-08-06 14:11')}	244377391	0	
2023-08-06 15:02	{'first': Timestamp('2023-08-06 15:02'), 'last': Timestamp('2023-08-06 15:03')}	244670667	0	
2023-08-06 15:19		248094000	0	1
2023-08-06 15:23	{'first': Timestamp('2023-08-06 15:23'), 'last': Timestamp('2023-08-06 17:02')}	244660495	2	1
2023-08-06 16:10	{'first': Timestamp('2023-08-06 16:10'), 'last': Timestamp('2023-08-06 16:12')}	244007308	0	

Table 5.6: Results -06/08/2023

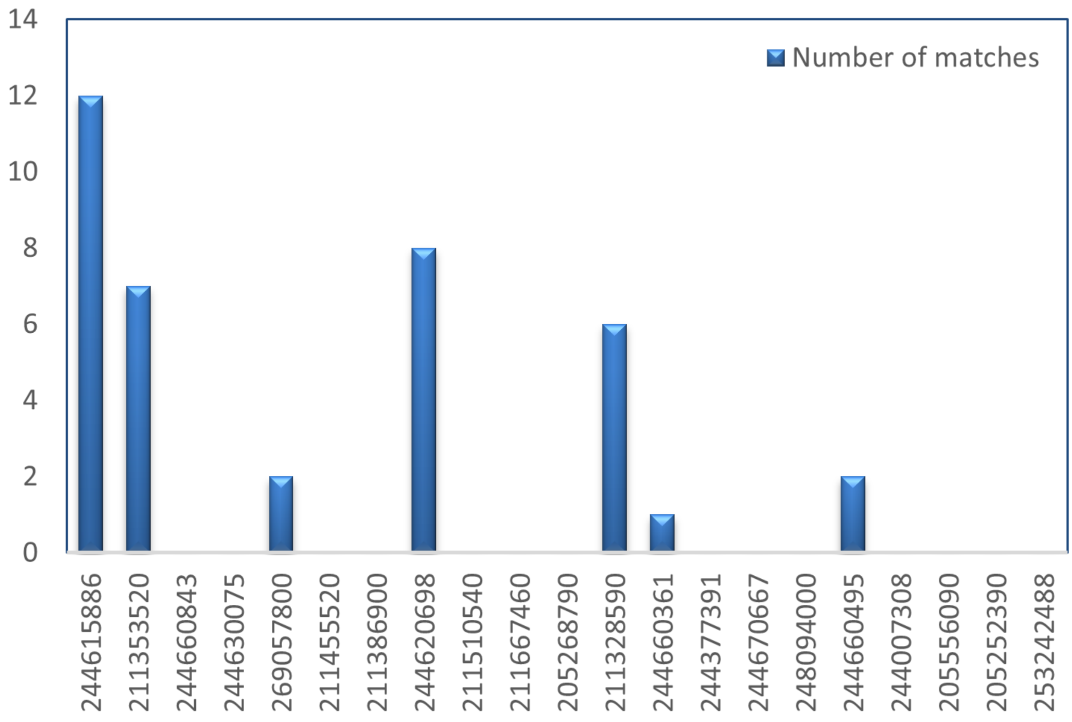


Figure 5.4: Matches per MMSI(s) - 06/08/2023

Figure 5.5 presents a visual breakdown of the total vessels identified, which amounts to 86 vessels for the specified date. This data is also presented in Tables 5.4 and 5.5 for a comprehensive reference.

The pie chart serves as a valuable tool to illustrate the distribution of these vessels into three distinct categories: those matched by our object detection model, those verified through manual inspection, and those that remain unmatched by either process. This breakdown provides

valuable insights into the degree of concordance between vessels observed in the field and those accounted for in the available AIS data.
Of particular note is the segment representing unmatched data, a critical aspect of our analysis. This portion encompasses both missing vessels, which are present but undetected, and instances of absent AIS data, where vessels recorded visually are not corroborated in the AIS dataset. This visual representation aids in uncovering discrepancies in inland vessel traffic data, shedding light on the intricacies of maritime vessel identification and the challenges posed by data gaps. It underscores the need for further research and improvements in both detection methodologies and AIS data completeness to enhance our understanding of maritime traffic dynamics.

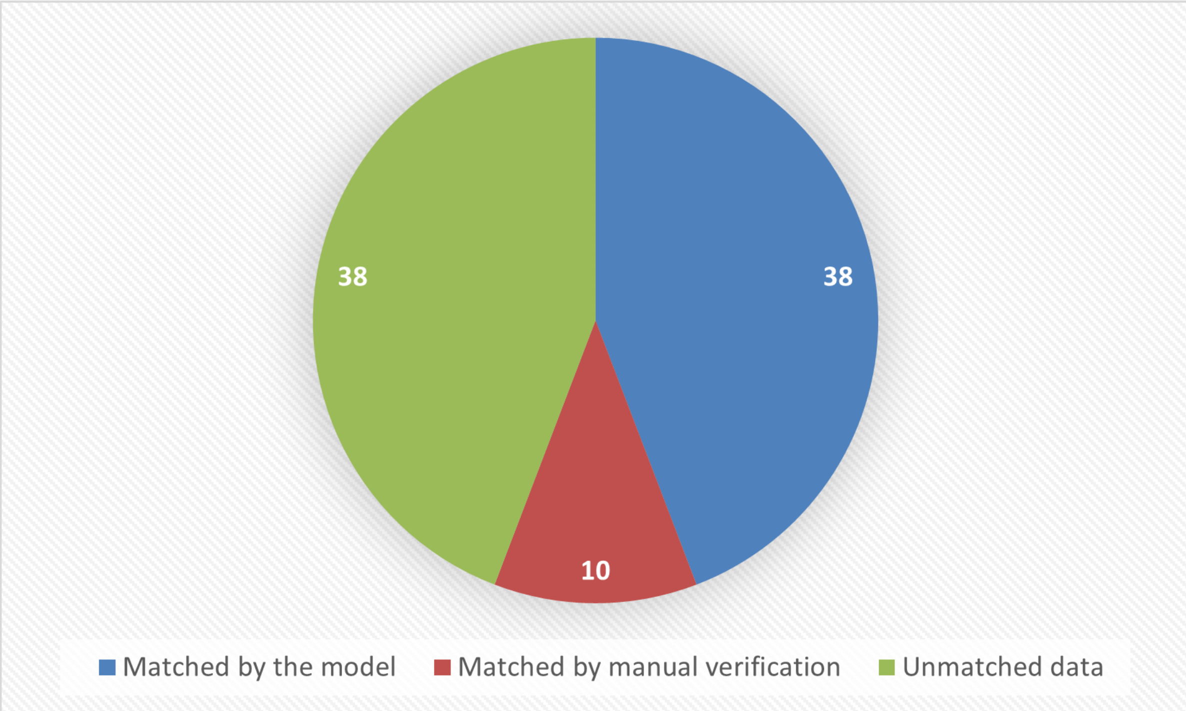


Figure 5.5: Comparison of data - 06/08/2023

5.1.3. Results for August 7, 2023

Embarking upon the third day of investigation, which unfolded on August 7, 2023, provided an invaluable opportunity to elevate the depth of our analysis. This additional day of scrutiny was not merely an arbitrary extension, but a purposeful endeavour to enhance the comprehensiveness of our study. Incorporating a third day into the examination aimed to fortify the credibility of our findings and validate the patterns observed across multiple instances. This pragmatic approach enabled us to glean insights that transcend temporal idiosyncrasies and reveal recurring themes. While the scale of this day's engagement is relatively modest—yielding 68 ship detections and 38 matches—it is emblematic of a strategic choice to triangulate our observations. This expanded endeavour serves to underscore the method's resilience and adaptability across varying conditions, thereby providing a more nuanced understanding of its efficacy. By weaving together the threads of consecutive days, our exploration extends beyond isolated incidents, unfolding a more holistic narrative that substantiates the viability of our approach. Table 5.7 shows the statistics of the algorithm for the video of the first day, where it was identified manually that the total number of vessels crossing in those 6 hours was 48 and the object detection model managed to detect 45, which depicts an accuracy of 94%. As seen in the experimentation of the previous day, this efficiency percentage of 94% isn't a perfect reflection of the capability of the model, since it contains false positives and quite a few vessels were obtained by the manual verification process.

Video Date	No.of Ships	Detected	Efficiency
07/08/2023	48	45	94%

Table 5.7: Object detection statistics (day3)

Upon the conclusion of the manual verification process, the comprehensive statistics are presented in Table 5.8. It's noteworthy that the total count of vessels detected by the model stands at 45 and 17 vessels have been manually detected. The attained vessels were then matched with the data, and 44 matches were found.

Video Date	Detected by Model	Manually detected	Overall Matches with AIS data
07/08/2023	45	17	44

Table 5.8: Overall Statistics(07/08/2023)

The most significant number of matches occurred on August 7, 2023, specifically within the time ranges 'first': Timestamp('2023-08-07 14:06:00'), 'last': Timestamp('2023-08-07 15:26:50.513000') and 'first': Timestamp('2023-08-07 13:28:26.510000'), 'last': Timestamp('2023-08-07 13:54:25').The Detailed results from processing the video are shown in the table 5.9. The results in the data of the third day of the sample set, help us understand the pattern of the results and consistency of the model. The matches per MMSI are depicted in the graph5.6.

timeLast Update	'timeLastUpdate': ['first', 'last']	mmsi	Number of matches	Manual
2023-08-07 12:15	{'first': Timestamp('2023-08-07 12:15'), 'last': Timestamp('2023-08-07 12:39')}	244690451	5	
2023-08-07 12:20	{'first': Timestamp('2023-08-07 12:20'), 'last': Timestamp('2023-08-07 13:18')}	244660297	2	
2023-08-07 12:32	244660955		0	
2023-08-07 12:32	{'first': Timestamp('2023-08-07 12:32'), 'last': Timestamp('2023-08-07 12:35')}	244650881	0	
2023-08-07 12:40	{'first': Timestamp('2023-08-07 12:40'), 'last': Timestamp('2023-08-07 12:43')}	244850318	0	
2023-08-07 12:56	{'first': Timestamp('2023-08-07 12:56'), 'last': Timestamp('2023-08-07 13:06')}	211551260	0	
2023-08-07 13:06		211552670	0	
2023-08-07 13:08	{'first': Timestamp('2023-08-07 13:08:'), 'last': Timestamp('2023-08-07 13:09')}	211883180	0	

2023-08-07 13:28	{'first': Timestamp('2023-08-07 13:28'), 'last': Timestamp('2023-08-07 13:54')}	211274030	8	
2023-08-07 13:28	{'first': Timestamp('2023-08-07 13:28'), 'last': Timestamp('2023-08-07 13:30')}	211535300	0	
2023-08-07 13:30	{'first': Timestamp('2023-08-07 13:30'), 'last': Timestamp('2023-08-07 13:54')}	211894840	0	
2023-08-07 13:54		244690528	0	
2023-08-07 14:01	211785210		0	
2023-08-07 14:04	{'first': Timestamp('2023-08-07 14:04'), 'last': Timestamp('2023-08-07 14:16')}	244130422	3	1
2023-08-07 14:06	{'first': Timestamp('2023-08-07 14:06'), 'last': Timestamp('2023-08-07 15:26')}	244700493	8	9
2023-08-07 14:10	{'first': Timestamp('2023-08-07 14:10'), 'last': Timestamp('2023-08-07 14:13')}	244150799	0	
2023-08-07 14:56	{'first': Timestamp('2023-08-07 14:56'), 'last': Timestamp('2023-08-07 15:11')}	244660499	0	
2023-08-07 15:03		211798640	0	
2023-08-07 15:46	{'first': Timestamp('2023-08-07 15:46'), 'last': Timestamp('2023-08-07 16:19')}	244670396	2	
2023-08-07 16:01		226008660	0	
2023-08-07 16:08		211845090	0	
2023-08-07 16:13	{'first': Timestamp('2023-08-07 16:13'), 'last': Timestamp('2023-08-07 16:17')}	244002399	0	
2023-08-07 16:20	{'first': Timestamp('2023-08-07 16:20'), 'last': Timestamp('2023-08-07 16:41')}	244314912	2	2
2023-08-07 16:54		211883520	0	
2023-08-07 17:15		211513810	0	
2023-08-07 17:16		244630683	0	
2023-08-07 17:17		244660684	0	
2023-08-07 17:35	{'first': Timestamp('2023-08-07 17:35'), 'last': Timestamp('2023-08-07 17:50')}	244710183	0	2
2023-08-07 17:38	{'first': Timestamp('2023-08-07 17:38'), 'last': Timestamp('2023-08-07 17:51')}	244660720	0	

Table 5.9: Results -07/08/2023

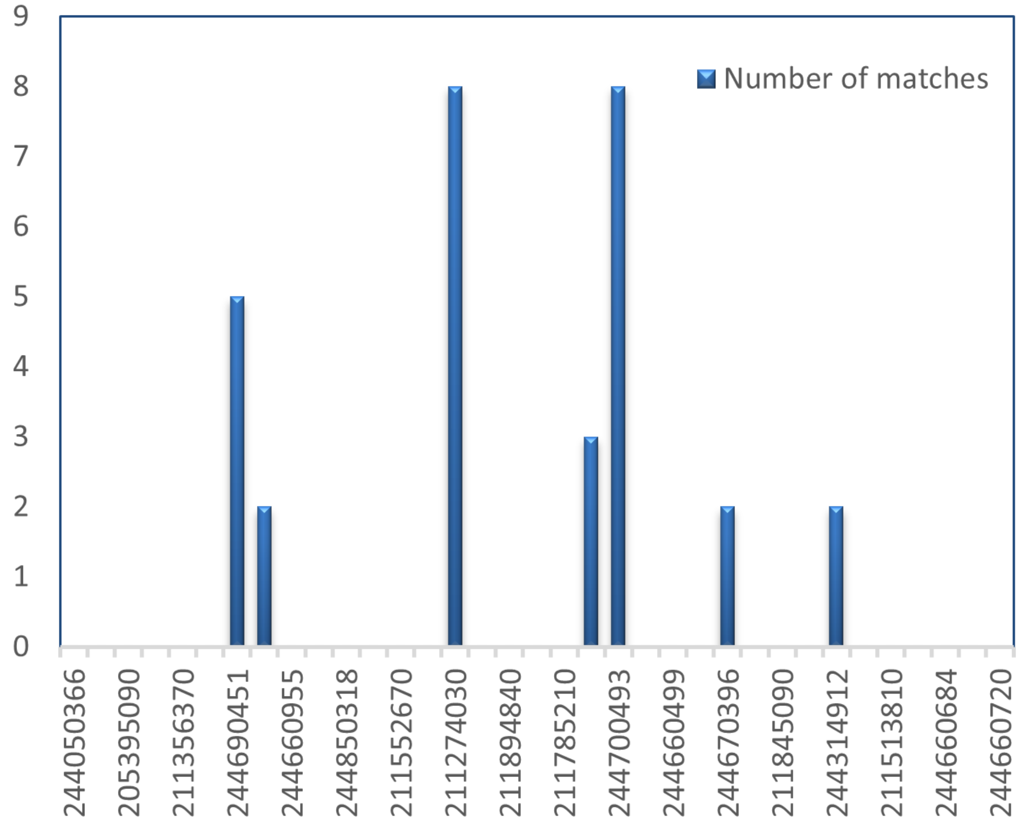


Figure 5.6: Matches per MMSI(s) - 07/08/2023

Figure 5.7 presents a detailed breakdown of the total vessels identified. This data is also comprehensively detailed in Tables 5.7 and 5.8.
The pie chart offers a visual representation that categorizes these vessels into three distinct groups: vessels matched by our object detection model, vessels verified through manual inspection, and vessels that remain unaccounted for by either process. This categorization provides crucial insights into the alignment between vessels observed in the field and those captured in the AIS dataset.
Of particular significance is the segment denoting unmatched data. This segment encompasses vessels that are either present but undetected or absent from the AIS dataset. This graphical depiction allows us to discern disparities in inland vessel traffic data, illuminating the complexities of maritime vessel identification and highlighting data gaps. It underscores the imperative for further research, aiming to refine both detection methodologies and AIS data integrity. Enhancing our comprehension of maritime traffic dynamics hinges on addressing these challenges and improving the completeness of AIS data.

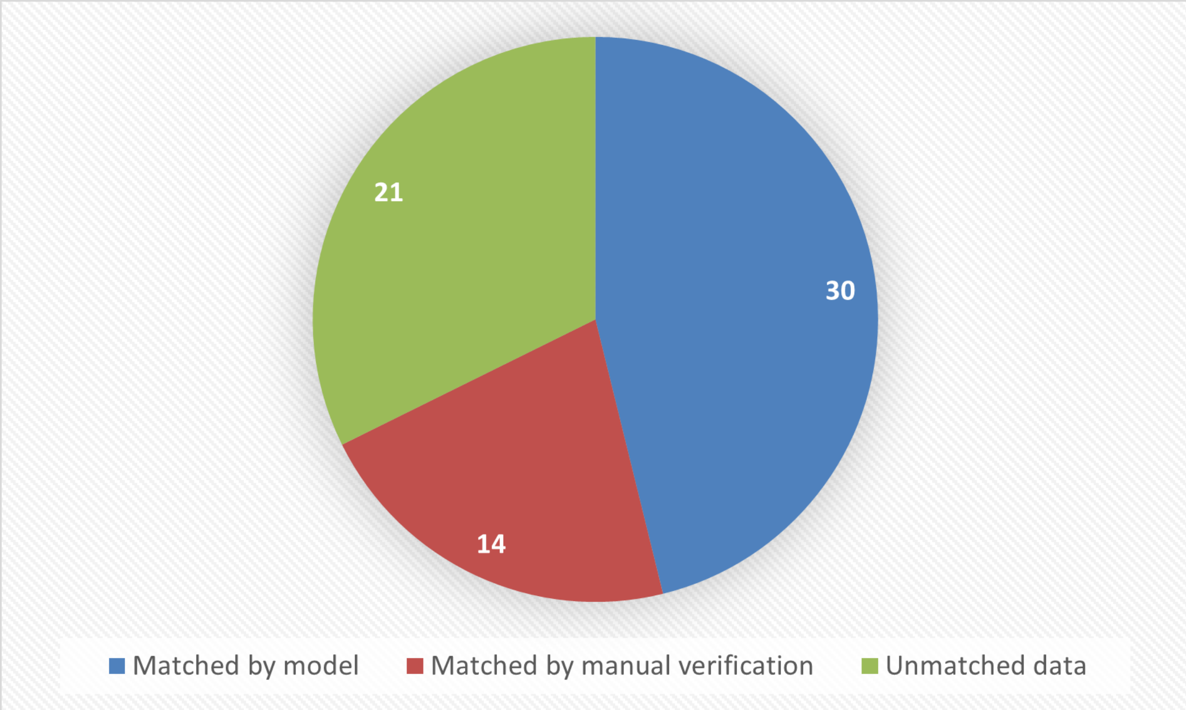


Figure 5.7: Comparison of data - 07/08/2023

5.1.4. Comparison of Results

The results obtained from the three days of analysis provide a comprehensive review of the object detection algorithm's performance and insight into the mismatch between the actual and AIS data. The accuracy of the model, i.e., the ratio of the number of ships in the video to the total number of ships detected in the model, got better in the last two days. While the model's efficiency has been emphasized, it is important to note that these results were obtained under daylight conditions, which are susceptible to lighting nuances.

Comparing the three days of data collection, it's evident that all three days consistently yielded more vessels detected by the object detection model than the actual number present in the video. This overestimation is primarily due to the presence of false positives, which can be attributed to the camera angle and challenges in tracking ships passing each other closely. It's crucial to note that the above results also shed light on the quality of the AIS data itself. The mismatches between detected vessels and AIS data entries can be attributed to various factors. These include the absence of vessels in the AIS data, and inaccuracies or delays in AIS data transmission. The sample set on day 2 includes a segment capturing heavy rain, and this meteorological condition did not undermine the model's accuracy, highlighting its robustness even in challenging weather. It's also significant to note that, according to the tables, there were different numbers of vessels crossing the said point every three days.

When looking at the tables 5.1, 5.4,5.7 the efficiencies of the object detection model are contrastingly different. Day 2 has the highest efficiency, but that was highly attributed to the false positives which have been explained under the results section each day. Shedding more light on the explanation of the false positives, one of the major instances that seemed to occur across the three days was the concurrent passing of ships also called occlusion. This caused the tracking system to fail since with the current field of view as seen in 4.3 when the ships pass at the same time, the ship on the other side is completely covered. This causes the creation of new vessels even though the entries for these ships have been made already, so a new entry creates a flag which is turned to be false and gets counted again or the algorithm tends to confuse the vessel between the entries and this creates a jump in the centroid values over the set threshold, which again identifies this vessel as a new vessel. The angle of the camera adds this constraint to this project.

The number of entries in the AIS data after the elimination of duplicates, where day 3 has AIS messages from 35 different MMSIs and days 2 and 1 have 21 and 20 entries, respectively. Now, with this in perspective, the matches column in the above table would be of much greater relevance, and the difference between the detected vessels and matches is evident. One factor that did affect the detections was that when a video with a higher data rate was obtained, it was known to affect the balance of the false positives and negatives by generating a lot more false positives. So for day 3, the value of the centroid distance threshold had to be determined and used for tracking.

The graphs plotted to show the split between the data matched by the algorithm, by manual process and unmatched data drawn for the three days, same can be seen in 5.5, 5.7, 5.3. The three days have 56, 38, and 32 entries that are unmatched with the data. This is due to both the factors of missing vessels for the data and also missing data, As mentioned above it could also be because of irregularly transmitted messages. As further research in this field spans, it would be possible to acquire information regarding the ship while they cross would help negate the false positives because with such information the tracking system could be made much better and the wrong creation of new vessels could be prevented.

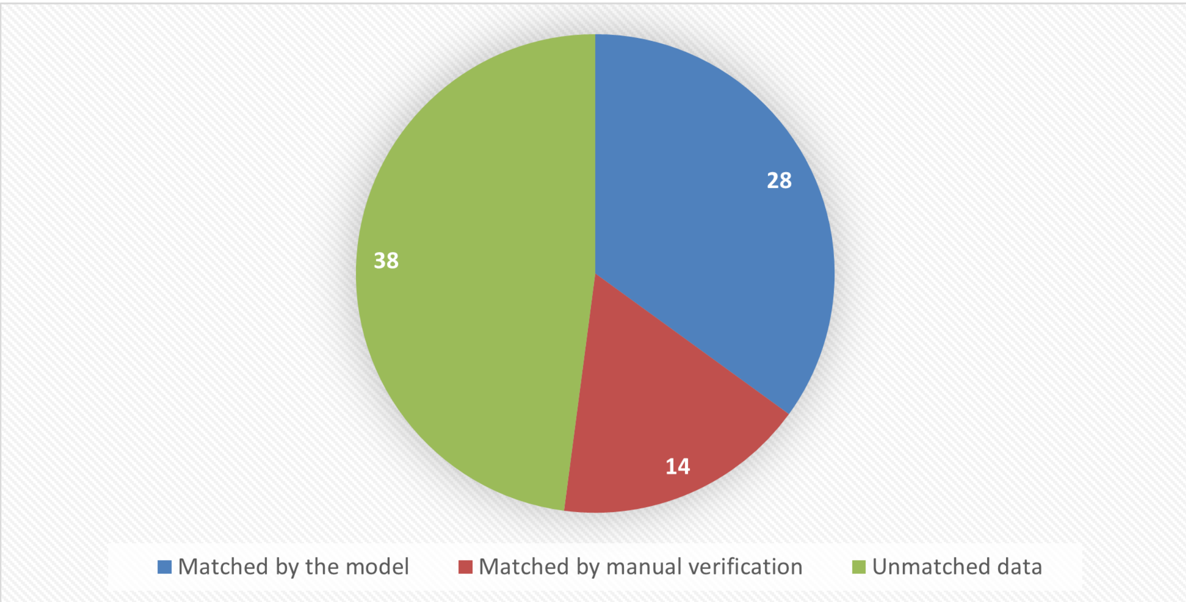


Figure 5.8: Results - Average AIS data matches

Figure 5.8 contains the average number of matches that are obtained by the algorithm, by manually identifying the vessels and the data that has been left unmatched over the three days. On average, there is a total of 28 entries that are matched by the model and a mere 14 which is matched due to the manual verification process, and 38 entries on average per day are left unmatched. The Significance and importance of AIS data are discussed in the chapters 1, 2 and the future scope of the fields that this data could be used in. It is important to examine the quality of this data, this being said the validation and testing part is complete. Based on my interpretation, the aforementioned findings suggest a deviation from the established standards regarding the transmission time limit for various classes of transponders. These standards dictate a maximum limit of three minutes. The results suggest that if the prescribed standards had been adhered to, specifically the threshold of two minutes set for this study, a significantly higher number of matches would have been observed.

5.2. Discussion

In this chapter, we reach a critical point in our research, embarking on a thorough examination of our method's results. We embark on a comprehensive journey through the implications of our research findings by unravelling the intricate interplay of data, insights, and interpretations derived from earlier sections. This chapter represents the culmination of our investigation, as we dissect underlying patterns, correlations, and insights hidden within the depths of the data. We hope to extract nuanced understandings and illuminate the narrative that emerges from the fusion of video surveillance, AIS data, and algorithmic precision through this thorough examination.

Our primary goal throughout this chapter is to reveal the practical implications of our research findings and shed light on their broader implications for maritime operations and security. Fine-tuning the algorithm's parameters helped effectively balance false negatives and false positives, which was possible due to the creation of a chat called Design Space Exploration. The Appendix contains the chart that aided in this delicate calibration by adjusting parameters such as confidence threshold, centroid distance threshold, reference line placement, and the frames to skip. This iterative process produced results that aided in the creation of the chart.

While the model's efficiency has been emphasised, it is important to note that these results were obtained under daylight conditions, which are susceptible to lighting nuances. When tested during nighttime scenarios as stated in the section 2.2, these figures may show variations. Furthermore, a significant change in ship types may necessitate additional training to ensure the model's adaptability. Surprisingly, the sample set on day 2 includes a segment capturing heavy rain, and this meteorological condition did not undermine the model's accuracy, highlighting its robustness even in challenging weather.

The results of the developed anomaly detection algorithm show gaps in the number of vessels identified in the video and matched with the AIS data. This shows promise in terms of the model's wider implementation and efficiency in detecting deviations, but it is untouched in terms of narrowing down to the specific vessel or ship that is responsible for the anomaly. The above is a constraint that we are currently dealing with because it not only provides us with the option of narrowing down the anomalies but also opens a horizon where we could check if the ships transmit data as per the standards of their class of AIS Transponders. The standards set by AIS are, [38]

- Class A AIS Transponders: Class A AIS transponders are typically used on larger vessels, and they are required to transmit AIS data every 2 to 10 seconds when the vessel is underway, depending on its speed. The faster the vessel's speed, the more frequently it transmits data.
- Class B AIS Transponders: Class B AIS transponders are often used on smaller vessels. They transmit AIS data less frequently compared to Class A transponders. Class B transponders transmit data every 30 seconds if the vessel is moving faster than 2 knots, and every 3 minutes if the vessel is stationary or moving slower than 2 knots.

- SARTs (Search and Rescue Transponders): SARTs are emergency devices used to aid in the location of a vessel or lifeboat in distress. They transmit AIS data at a rate of once every 8 seconds.
- AIS Base Stations and AtoNs (Aids to Navigation): These transmit AIS data at a fixed interval, usually every 3 minutes.

To avoid duplicates of the MMSI, the 'timelastupdate' was to be considered in ranges in the Post-processing section of the code. Because of the aforementioned constraint, this method was chosen. This also meant that the process was less accurate because it resulted in the formation of ranges that spanned more than an hour. For example, in the first day's results, MMSI: 244690451 generated the range 'first': Timestamp('2023-05-31 18:37:45.312000'), 'last': Timestamp('2023-05-31 20:15:35.713000'). This is more than 2 hours, which resulted in this entry accumulating 10 matches and also denying matches to other entries with 'timelastupdate' within this range.

5.3. Sensitivity Analysis

In our never-ending quest to understand the complexities of our research results, we take a step into the world of sensitivity analysis. This section breaks down the big effects that changes in key variables can have on our research results and final conclusions. Sensitivity analysis is an important tool that helps us figure out how strong our models are and how reliable our findings are when we have to deal with changing parameters. We carefully and methodically change these variables and pay close attention to what happens as a result. This gives us deep insights into how stable and universal our research results are.

One significant factor to consider is the time of data acquisition, as observations during the day may yield different results compared to nighttime conditions, necessitating adjustments to parameters. Additionally, the quality of the video, often indicated by a high data rate, directly affects our analysis, requiring potential adaptations in parameters like 'Centroid_distance_threshold.' Framerate, or the number of frames processed, influences our analysis by altering the distance vessels travel between frames. This, in turn, may necessitate modifications to the centroid distance threshold value.

Furthermore, the confidence threshold parameter plays a pivotal role in balancing false positives and false negatives, with sensitivity to low-light conditions and dynamic environmental factors being crucial. Surprisingly, diverse weather conditions, such as rain, appeared to have a limited impact on our results during the second day of data collection.

These sensitivity analyses underscore the importance of considering various parameters and conditions, ensuring the robustness and applicability of our research outcomes across different contexts.

6
Conclusion

Finally, the goal of the thesis was to assess the feasibility of using video surveillance to identify mismatches in AIS data. An object detection model was trained to detect vessels and compare the results with AIS data. This chapter focuses on answering research questions and drawing conclusions from the research. Additionally, the chapter concludes with recommendations for future research in this area.

6.1. Answering Research Questions

This section answers the Main research question, for which an answer would be reached by answering the following sub-research questions.

1. What are the current state-of-the-art methods and technologies for detecting and mitigating instances of overlapping or spoofed AIS data in maritime environments?

- The Literature overview conducted earlier helped us determine the literature gap ' the lack of research in the application of video surveillance in the detection of overlapping AIS data. This instigated to test of the usage of the object detection algorithm on video surveillance' This helped frame the basis for this research and set up the questions
- The Literature review in chapter 2 conducted as a part is this study helped determine the current state-of-the-art methods and technologies that are being used or suggested to determine the Quality of AIS data, in Object detection in general and in the specific case of Inland Water Transport, and in anomaly detection of AIS data.

2. How can an algorithm be systematically selected and designed to effectively identify and mitigate instances of overlapping or spoofed AIS data, considering factors such as accuracy, real-time processing, scalability, and integration with video surveillance?

- This sub-research question is answered in chapter 4, where the methodology for the project is from the Data collection, algorithm design process, data analysis, False Negative/positive Trade-offs, and Validation testing.
- The Process architecture is clearly described in 4.2
- The detailed process to set up and execute the project is mentioned in a Readme file along with the other files in the Github Repository [49].

3. How accurate is the proposed object detection algorithm in identifying and localizing ships from video surveillance in various maritime conditions and environments?

The accuracy of the proposed object detection algorithm in identifying and localizing ships from video surveillance in diverse maritime conditions and environments has been a central focus of this study. Through extensive testing and analysis, it has been revealed that the algorithm demonstrates a practical efficiency ranging from 75%, as measured by the ratio of detected ships to the total number of ships within the video footage. The actual efficiency for each of the videos is presented in the tables 5.1, 5.4, and 5.7. This accuracy has been achieved by carefully fine-tuning the algorithm's parameters, effectively addressing the trade-off between false positives and false negatives. However, it's important to acknowledge that these results were attained under daylight conditions, which can influence the accuracy due to lighting variations. Further investigations are needed to assess the algorithm's performance under varying lighting, weather conditions, and diverse ship types. Despite these considerations, the algorithm's performance signals its potential to contribute significantly to ship detection and localization tasks within maritime environments, offering a valuable tool for enhancing safety, security, and operational efficiency.

4. What are the discrepancies and variations between the AIS data and the ship detections obtained through the object detection algorithm, and how can these differences be quantified and analyzed?

- The variations between the results of Object detection and AIS data for the three days are run and the results are shown in the chapter 5.
- There proved to be clear deference in each of the three days between the matched and Detected ships as shown in the tables 5.2,5.5, and 5.8 and the split between the data are shown in the graphs 5.3, 5.5, and 5.7.
- The final results with the entries of the AIS data and the time of crossing of ships are presented as tables in 5.3, 5.6 and 5.9.

5.Can combining video surveillance of ships with existing AIS data help spot and recognize cases of overlapping or fake AIS information, and what difficulties and constraints might arise from adopting this approach?

This sub-research question is answered more in detail in the discussion(5.2) part of the results chapter. The constraints that are currently applicable to the method employed are

- The constraint that might arise is localising the cause of the anomaly, which at least currently isn't possible.
- Video footage can be affected by lighting, weather, and visibility conditions, potentially leading to false positives or false negatives.
- Improving the overall accuracy of the object detection model.

In conclusion, the application of a machine learning model to enhance the detection of anomalies and mismatches in data holds great promise. However, it is evident that for the successful real-world implementation of this algorithm, there remain numerous unexplored areas and compelling research avenues to be thoroughly investigated. These aspects will be further discussed and explored in the future research section.

6.2. Future Work

Future research in maritime object detection could extend beyond the current focus on object detection and result comparison. One of the primary objectives could be to address the challenge of pinpointing the specific vessel responsible for anomalies. This challenge arises from the complex and dynamic nature of maritime environments. Researchers may explore the development of adaptive algorithms capable of dynamically adjusting detection thresholds and parameters. These algorithms would consider factors such as changing illumination, varying weather conditions, and atmospheric effects. By tailoring algorithms to specific scenarios encountered in inland water transport, researchers aim to ensure robust performance across a wide range of conditions.

A critical aspect of future research is the continuous quest to enhance the accuracy of object detection models. The target could be to achieve an accuracy rate exceeding 99%. Achieving such precision is crucial for reliably identifying vessels and anomalies in complex maritime settings. Researchers may explore various strategies, including fine-tuning model architectures, optimizing hyperparameters, and employing larger and more diverse datasets. Rigorous testing across different environmental settings will help identify areas for parameter optimization and fine-tuning, ultimately leading to higher accuracy rates.

Additionally, researchers should explore innovative data sources, such as satellite imagery, to enhance anomaly detection and improve data reliability. Research has already been done regarding the merging of data sources as mentioned in section 2.1. This multi-source approach would enable the detection of 'dark vessels' that may attempt to evade AIS tracking. By cross-referencing satellite data with AIS broadcasts and video surveillance, researchers could significantly enhance the comprehensiveness of vessel tracking.

Beyond inland water transport, object detection models find applications in diverse domains. One promising area is offshore wind farm surveillance, where cameras placed on wind turbines can identify vessels in offshore areas. This technology enhances the safety and security of offshore operations.

Furthermore, object detection models can play a vital role in port security. Deploying these models provides real-time insights into vessel traffic, improving port security measures. By monitoring vessel movements and identifying potential anomalies or security threats, ports can bolster their security infrastructure. Future research might focus on tailoring object detection algorithms to meet the unique requirements of port environments, ensuring their effectiveness in safeguarding critical infrastructure.

Access to vessel-specific information, such as the MMSI, presents an additional opportunity to enhance object detection systems. This data can assist in localizing vessels and determining their time of crossing, adding a layer of precision to data comparison. In cases where data mismatches occur, having access to ship-specific information enables the identification of vessels that haven't adhered to established AIS data standards. Future research avenues may explore efficient methods for retrieving and integrating this vessel-specific data into object detection processes.

In conclusion, the future of research in maritime object detection holds exciting possibilities. By developing adaptive algorithms, improving model accuracy, leveraging innovative data sources, exploring new applications, and utilizing vessel-specific information, researchers can significantly advance maritime safety, security, and operational efficiency. These advancements will not only benefit inland water transport but also expand the applications of AIS data analysis in various domains, ultimately contributing to safer and more secure maritime operations.

References

[1] Nadia M. Abdallah et al. “Fixing errors in the AIS destination field”. In: *OCEANS 2019 - Marseille* (June 2019). doi: 10.1109/oceanse.2019.8867139.

[2] I Ankith. “Real Time Object Detection using YoloReal time Object detection using Yolo”. In: *International Journal for Research in Applied Science and Engineering Technology* 9.11 (Nov. 30, 2021), pp. 1504–1511. doi: 10.22214/ijraset.2021.39044. url: https://doi.org/10.22214/ijraset.2021.39044.

[3] Ernest Batty. “Data Analytics Enables Advanced AIS Applications”. In: *Lecture Notes in Computer Science* (Sept. 2017), pp. 22–35. doi: 10.1007/978-3-319-73521-4\{_}2.

[4] Bjorn Bergman. *Russian Tanker Falsifies AIS Data, Hides Likely Activity Around Malta and Cyprus*. Jan. 2023. url: https://skytruth.org/2022/12/russian-tanker-falsifies-ais-data-hides-likely-activity-around-malta-and-cyprus/.

[5] Niko Bralić and Josip Musić. “System for automatic detection and classification of cars in traffic”. In: *St open* (2022). url: https://api.semanticscholar.org/CorpusID:253328635.

[6] Han-Chieh Chao, Hsin-Te Wu, and Fan-Hsun Tseng. “AIS Meets IoT: A Network Security Mechanism of Sustainable Marine Resource Based on Edge Computing”. In: *Sustainability* 13.6 (Mar. 2021), p. 3048. doi: 10.3390/su13063048. url: https://www.mdpi.com/2071-1050/13/6/3048/pdf.

[7] Shuguang Chen, Yikun Huang, and Wei Lu. “Anomaly Detection and Restoration for AIS Raw Data”. In: *Wireless Communications and Mobile Computing* 2022 (Mar. 2022), pp. 1–11. doi: 10.1155/2022/5954483. url: https://downloads.hindawi.com/journals/wcmc/2022/5954483.pdf.

[8] Zhang Daiyong et al. “Enhance the AIS data availability by screening and interpolation”. In: *International Conference on Transportation Information and Safety* (Aug. 2017). doi: 10.1109/ictis.2017.8047888.

[9] Zhang Daiyong et al. “Enhance the AIS data availability by screening and interpolation”. In: *International Conference on Transportation Information and Safety* (Aug. 2017). doi: 10.1109/ictis.2017.8047888.

[10] Lei Du, Floris Goerlandt, and Pentti Kujala. “Review and analysis of methods for assessing maritime waterway risk based on non-accident critical events detected from AIS data”. In: *Reliability Engineering & System Safety* 200 (Aug. 2020), p. 106933. doi: 10.1016/j.ress.2020.106933. url: https://doi.org/10.1016/j.ress.2020.106933.

[11] Md. Alamin Feroz et al. *Object Detection and Classification from a Real-Time Video Using SSD and YOLO Models*. Sept. 5, 2021, pp. 37–47. doi: 10.1007/978-981-16-2543-5_4. url: https://doi.org/10.1007/978-981-16-2543-5_4.

[12] G.L. Foresti. “Object recognition and tracking for remote video surveillance”. In: *IEEE Transactions on Circuits and Systems for Video Technology* 9.7 (Jan. 1, 1999), pp. 1045–1062. doi: 10.1109/76.795058. url: https://doi.org/10.1109/76.795058.

[13] *Google Maps*. url: https://www.google.com/maps/place/Rheinpromenade+Rees/@51.7554017,6.3942945,16.25z/data=!4m14!1m7!3m6!1s0x47c77c2c889357a5:0x9b08c5cbcfccaac0!2sRheinpromenade+Rees!8m2!3d51.7571185!4d6.3944047!16s%2Fg%2F11hdb7ykcw!3m5!1s0x47c77c2c889357a5:0x9b08c5cbcfccaac0!8m2!3d51.7571185!4d6.3944047!16s%2Fg%2F11hdb7ykcw!5m1!1e4?entry=ttu.

[14] Athanassios Goudossis and Sokratis K. Katsikas. “Towards a secure automatic identification system (AIS)”. In: *Journal of marine science and technology* 24.2 (June 2019), pp. 410–423. doi: 10.1007/s00773-018-0561-3.

[15] Li Guo et al. “Research on Ship Track Fusion Model based on Distributed Extended Kalman Filter”. In: *2022 IEEE 6th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC)*. 2022, pp. 845–851. doi: 10.1109/IAEAC54830.2022.9929871.

[16] Shanzeng Guo. “Space-based detection of spoofing AIS signals using Doppler frequency”. In: *Proceedings of SPIE* (May 28, 2014). doi: 10.1117/12.2050448. url: https://doi.org/10.1117/12.2050448.

[17] A Harati-Mokhtari et al. “Automatic Identification System (AIS): Data Reliability and Human Error Implications”. In: *Journal of Navigation* 60.3 (Sept. 2007), pp. 373–389. doi: 10.1017/s0373463307004298.

[18] Wei He et al. “A Novel Fitting Model for Practical AIS Abnormal Data Repair in Inland River”. In: *Elektronika Ir Elektrotechnika* 27.1 (Feb. 2021), pp. 60–70. doi: 10.5755/j02.eie.27661. url: https://eejournal.ktu.lt/index.php/elt/article/download/27661/14692.

[19] Tomasz Hyla and Natalia Wawrzyniak. *Ships detection on inland waters using video surveillance system*. Jan. 1, 2019, pp. 39–49. doi: 10.1007/978-3-030-28957-7_4. url: https://doi.org/10.1007/978-3-030-28957-7_4.

[20] “Integrating AIS and SAR to monitor fisheries: A pilot study in the Adriatic Sea”. In: *MetroSea 2020 - TC19 International Workshop on Metrology for the Sea* (2020), pp. 39–44. url: http://www.scopus.com/.

[21] Clément Iphar, Aldo Napoli, and Cyril Ray. “DATA QUALITY ASSESSMENT FOR MARITIME SITUATION AWARENESS”. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* (Aug. 2015). doi: 10.5194/isprsannals-ii-3-w5-291-2015. url: https://www.isprs-ann-photogramm-remote-sens-spatial-inf-sci.net/II-3-W5/291/2015/isprsannals-II-3-W5-291-2015.pdf.

[22] Clément Iphar, Aldo Napoli, and Cyril Ray. “Detection of false AIS messages for the improvement of maritime situational awareness”. In: *HAL (Le Centre pour la Communication Scientifique Directe)* (Oct. 2015). doi: 10.23919/oceans.2015.7401841.

[23] Clément Iphar, Aldo Napoli, and Cyril Ray. “On the Interest of Data Mining for an Integrity Assessment of AIS Messages”. In: *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*. 2016, pp. 368–373. doi: 10.1109/ICDMW.2016.0059.

[24] Clément Iphar, Cyril Ray, and Aldo Napoli. “Uses and Misuses of the Automatic Identification System”. In: *HAL (Le Centre pour la Communication Scientifique Directe)* (June 2019). doi: 10.1109/oceanse.2019.8867559.

[25] Mark James et al. “AIS data to inform small scale fisheries management and marine spatial planning”. In: *Marine Policy* 91 (May 2018), pp. 113–121. doi: 10.1016/j.marpol.2018.02.012.

[26] Krzysztof Jaskolski. "Availability of Automatic Identification System (AIS) Based on Spectral Analysis of Mean Time to Repair (MTTR) Determined from Dynamic Data Age". In: *Remote Sensing* 14.15 (Aug. 2022), p. 3692. doi: 10.3390/rs14153692. url: https://www.mdpi.com/2072-4292/14/15/3692/pdf?version=1659430134.

[27] Krzysztof Jaskolski et al. "Automatic Identification System (AIS) Dynamic Data Integrity Monitoring and Trajectory Tracking Based on the Simultaneous Localization and Mapping (SLAM) Process Model". In: *Sensors* 21.24 (Dec. 2021), p. 8430. doi: 10.3390/s21248430. url: https://www.mdpi.com/1424-8220/21/24/8430/pdf?version=1639839309.

[28] Naoya Kaido, Shigehiro Yamamoto, and Takeshi Hashimoto. "Examination of automatic detection and tracking of ships on camera image in marine environment". In: *2016 Techno-Ocean (Techno-Ocean)*. 2016, pp. 58–63. doi: 10.1109/Techno-Ocean.2016.7890748.

[29] Fotios Katsilieris, Paolo Braca, and Stefano Coraluppi. "Detection of malicious AIS position spoofing by exploiting radar information". In: *International Conference on Information Fusion* (July 2013), pp. 1196–1203. url: http://dblp.uni-trier.de/db/conf/fusion/fusion2013.html#KatsilierisBC13.

[30] Gary C. Kessler. "Protected AIS: A Demonstration of Capability Scheme to Provide Authentication and Message Integrity". In: *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation* 14.2 (June 2020), pp. 279–286. doi: 10.12716/1001.14.02.02. url: https://doi.org/10.12716/1001.14.02.02.

[31] Ioannis Kontopoulos et al. "Countering Real-Time Stream Poisoning: An Architecture for Detecting Vessel Spoofing in Streams of AIS Data". In: *Zenodo (CERN European Organization for Nuclear Research)* (Aug. 2018). doi: 10.1109/dasc/picom/datacom/cyberscitec.2018.00139.

[32] Ioannis Kontopoulos et al. "Countering Real-Time Stream Poisoning: An Architecture for Detecting Vessel Spoofing in Streams of AIS Data". In: *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech)*. 2018, pp. 981–986. doi: 10.1109/DASC/PiCom/DataCom/CyberSciTec.2018.00139.

[33] Krishna et al. "Automated traffic monitoring system using computer vision". In: *2016 International Conference on ICT in Business Industry & Government (ICTBIG)* (2016), pp. 1–5. url: https://api.semanticscholar.org/CorpusID:41746752.

[34] Richard D. Lane et al. "Maritime anomaly detection and threat assessment". In: *International Conference on Information Fusion* (July 2010). doi: 10.1109/icif.2010.5711998.

[35] Philipp Last, Martin Hering-Bertram, and Lars Linsen. "How automatic identification system (AIS) antenna setup affects AIS signal quality". In: *Ocean Engineering* 100 (2015), pp. 83–89. issn: 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2015.03.017. url: https://www.sciencedirect.com/science/article/pii/S002980181500075X.

[36] Xinglong Liu et al. "A Multi-source Information Fusion Method for Error AIS Targets Identification". In: *The Open electrical & electronic engineering journal* (Dec. 2014). doi: 10.2174/1874129001408010729.

[37] Fabio Mazzarella et al. "A novel anomaly detection approach to identify intentional AIS on-off switching". In: *Expert Systems With Applications* 78 (July 2017), pp. 110–123. doi: 10.1016/j.eswa.2017.02.011. url: https://doi.org/10.1016/j.eswa.2017.02.011.

[38] OpenAI. *Automatic Identification System (AIS) Data Transmission Frequency Guidelines*. Retrieved Month Day, Year, from [URL of this conversation]. 2023.

[39] Dilip K. Prasad et al. "Challenges in video based object detection in maritime scenario using computer vision". In: *arXiv (Cornell University)* (Aug. 3, 2016). doi: 10.48550/arxiv.1608.01079. url: http://arxiv.org/abs/1608.01079.

[40] Jingxiang Qu et al. "Improving maritime traffic surveillance in inland waterways using the robust fusion of AIS and visual data". In: *Ocean Engineering* 275 (2023), p. 114198. issn: 0029-8018. doi: https://doi.org/10.1016/j.oceaneng.2023.114198. url: https://www.sciencedirect.com/science/article/pii/S0029801823005826.

[41] P. M. Rajeshwari, P Abhishek, and P. Srikanth | T. Vinod. "Object Detection: an overview". In: *International Journal of Trend in Scientific Research and Development* Volume-3.Issue-3 (Apr. 30, 2019), pp. 1663–1665. doi: 10.31142/ijtsrd23422. url: https://doi.org/10.31142/ijtsrd23422.

[42] Andrew Rawson and Mario Brito. "A survey of the opportunities and challenges of supervised machine learning in maritime risk analysis". In: *Transport Reviews* 43.1 (Feb. 2022), pp. 108–130. doi: 10.1080/01441647.2022.2036864. url: https://doi.org/10.1080/01441647.2022.2036864.

[43] Cyril Ray et al. "DeAIS project: Detection of AIS spoofing and resulting risks". In: *HAL (Le Centre pour la Communication Scientifique Directe)* (May 2015). doi: 10.1109/oceans-genova.2015.7271729.

[44] Gerard Recasens et al. "Learning fishing information from AIS data". In: Nov. 2022, pp. 9–18. doi: 10.1145/3557921.3565543.

[45] J. Redmon et al. "You Only Look Once: Unified, Real-Time Object Detection". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, June 2016, pp. 779–788. doi: 10.1109/CVPR.2016.91. url: https://doi.ieeecomputersociety.org/10.1109/CVPR.2016.91.

[46] Maximilian Rodger and Raffaella Guida. "Mapping Dark Shipping Zones Using Multi-Temporal SAR and AIS Data for Maritime Domain Awareness". In: *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium* (July 2022). doi: 10.1109/igarss46834.2022.9883797. url: https://doi.org/10.1109/igarss46834.2022.9883797.

[47] Hao Rong, A.P. Teixeira, and C. Guedes Soares. "Data mining approach to shipping route characterization and anomaly detection based on AIS data". In: *Ocean Engineering* 198 (Feb. 1, 2020), p. 106936. doi: 10.1016/j.oceaneng.2020.106936. url: https://doi.org/10.1016/j.oceaneng.2020.106936.

[48] Loic Salmon et al. "Data fusion challenges for AIS anti-piracy measures". In: *HAL (Le Centre pour la Communication Scientifique Directe)* (Sept. 2021). doi: 10.23919/oceans44145.2021.9705709.

[49] Suryaa Vadachennimalai Selvaraj. *Machine Learning for IWT Ship Traffic Analysis: Object Detection vs. AIS Data*. GitHub Repository. 2023. url: https://github.com/SuryaaVadachennimalaiSelvaraj/Machine-Learning-for-IWT-Ship-Traffic-Analysis-Object-Detection-vs.-AIS-Data-.

[50] Jennifer Shepperson et al. "A comparison of VMS and AIS data: the effect of data coverage and vessel position recording frequency on estimates of fishing footprints". In: *Ices Journal of Marine Science* 75.3 (May 2018), pp. 988–998. doi: 10.1093/icesjms/fsx230.

[51] Milena Strozyna, Dominik Filipiak, and Krzysztof Wecel. “Data Quality Assessment – A Use Case from the Maritime Domain”. In: Nov. 2020, pp. 5–20. isbn: 978-3-030-61145-3. doi: 10.1007/978-3-030-61146-0_1.

[52] Anna Nora Tassetti, Carmen Ferrà, and Gianna Fabi. “Rating the effectiveness of fishery-regulated areas with AIS data”. In: *Ocean & Coastal Management* 175 (June 2019), pp. 90–97. doi: 10.1016/j.ocecoaman.2019.04.005.

[53] Enmei Tu et al. “Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey From Data to Methodology”. In: *IEEE Transactions on Intelligent Transportation Systems* 19.5 (May 2018), pp. 1559–1582. doi: 10.1109/tits.2017.2724551.

[54] UP42. *The Search for Dark Vessels: Fusing AIS Signals and Deep Learning-Based Ship Detection | Webinar*. Dec. 2020. url: https://www.youtube.com/watch?v=vX_rnesRrfM.

[55] Konrad Wolsing et al. “Anomaly Detection in Maritime AIS Tracks: A Review of Recent Approaches”. In: *Journal of Marine Science and Engineering* 10.1 (Jan. 2022), p. 112. doi: 10.3390/jmse10010112. url: https://www.mdpi.com/2077-1312/10/1/112/pdf?version=1642404728.

[56] Yong Wu et al. “A new multi-sensor fusion approach for integrated ship motion perception in inland waterways”. In: *Measurement* 200 (July 2022), p. 111630. doi: 10.1016/j.measurement.2022.111630.

[57] Ran Yan et al. “Development of denoising and compression algorithms for AIS-based vessel trajectories”. In: *Ocean Engineering* 252 (May 2022), p. 111207. doi: 10.1016/j.oceaneng.2022.111207.

[58] Dong Hyun Yang et al. “How big data enriches maritime research – a critical review of Automatic Identification System (AIS) data applications”. In: *Transport Reviews* 39.6 (July 2019), pp. 755–773. doi: 10.1080/01441647.2019.1649315.

[59] Jufu Zhang et al. “Incorporation of Deep Kernel Convolution into Density Clustering for Shipping AIS Data Denoising and Reconstruction”. In: *Journal of Marine Science and Engineering* 10.9 (Sept. 2022), p. 1319. doi: 10.3390/jmse10091319. url: https://www.mdpi.com/2077-1312/10/9/1319/pdf?version=1663814152.

[60] Liangbin Zhao, Guoyou Shi, and Jiaxuan Yang. “Ship Trajectories Pre-processing Based on AIS Data”. In: *Journal of Navigation* 71.5 (Sept. 2018), pp. 1210–1230. doi: 10.1017/s0373463318000188.

[61] Fan Zhou, Pan Su, and Jingjing Jiang. “Verification of AIS Data by using Video Images taken by a UAV”. In: *Journal of Navigation* 72.06 (May 8, 2019), pp. 1345–1358. doi: 10.1017/s0373463319000262. url: https://doi.org/10.1017/s0373463319000262.

A
Literature Overview

C.No	Title	Author	Method's Used	Year
1	Fixing errors in the AIS destination field[1]	Abdallah, Nadia Ben: Iphar, Clement: Arcieri, Gianfranco: Jousselme, Anne-Laure	A ”cleaning-matching”algorithm	2019
2	Data analytics enables advanced AIS applications[3]	Batty, E.	Data Analytics	2017
3	Integrating AIS and SAR to monitor fisheries: A pilot study in the Adriatic Sea [20]	Galdelli, Alessandroa: Mancini, Adrianoa: Ferrà, Carmenb: Tassetti, Anna Nora	A machine-learning approach based on the Fast Fourier Transform	2020
4	Towards a secure automatic identification system (AIS) [14]	Goudossis, Athanassios: Katsikas, Sokratis K	Identity-Based Public Cryptography and Symmetric Cryptography	2019
5	A novel anomaly detection approach to identify intentional AIS on-off switching [37]	Mazzarella, Fabio: Vespe, Michele: Alessandrini, Alfredo: Tarchi, Darioa: Aulicino, Giuseppe: Vollero, Antonio	In designing such an anomaly detector, the electromagnetic propagation conditions that characterize the channel between ship AIS transponders and BS have to be taken into consideration	2017
6	Protected ais: A demonstration of capability scheme to provide authentication and message integrity [30]	Kessler G.C.	Protected AIS (pAIS), a demonstration using public-key cryptography methods	2020
7	Automatic identification system (AIS): Data reliability and human error implications [17]	Harati-Mokhtari, Abbas: Wall, Alan: Brooks, Philip: Wang, Jin	”Swiss Cheese” Model	2007
8	Data quality assessment for maritime situation awareness [21]	Iphar C: Napoli A: Ray C	Novel methodological approach for modelling, analysing and detecting these data errors and falsifications are introduced	2015
9	Exploiting AIS Data for Intelligent Maritime Navigation: A Comprehensive Survey from Data to Methodology [53]	Tu, Enmei: Zhang,Guanghao: Rachmawati, Lily: Rajabally, Eshan: Huang, Guang-Bin	Data Mining	2018
10	How big data enriches maritime research– a critical review of Automatic Identification System (AIS) data applications [58]	Yang, Dong: Wu, Lingxiao: Wang, Shuaian: Jia, Haiyin: Li, Kevin X	AIS applications by dividing into three development stages, namely, basic application, extended application, and advanced application	2019

Table A.1: Cluster based on the Keyword ” AIS data quality”

S.No	Title	Author(s)	Methodology	Year
1	Data fusion challenges for AIS anti-piracy measures [48]	Salmon, Loica: Laso, Pedro Merino: Claramunt, Christophe: Follut, Dominique: Pelissero, Nicolas	ISOLA System to support against threats and challenges related to AIS	2021
2	Learning fishing information from AIS data [44]	Recasens, Gerard Pons: Bilalli, Besim: Abelló, Alberto: Sánchez, Santiago Blanco	By creating a training dataset where a model is learned to predict whether an area of the Ocean is productive or not	2022
3	Mapping Dark Shipping Zones using Multi-Temporal SAR and AIS Data for Maritime Domain Awareness [46]	Rodger, Maximilian: Guida, Raffaella	Kernel Density Estimation (KDE) is applied to SAR ship detections to identify 'Dark' Zones	2022

Table A.2: Cluster-based on the Keyword: AIS Data

S.No	Title	Author(s)	Methodology	Year
1	Incorporation of Deep Kernel Convolution into Density Clustering for Shipping AISn Data Denoising and Reconstruction [59]	Zhang, J: Ren, X: Li, H.:Yang, Z.	Density-Based Spatial Clustering of Application with Noise based on Deep Kernel Convolution (DBSCANDKC) and the reconstruction method	2022
2	A new multi-sensor fusion approach for integrated ship motion perception in inland waterways [56]	Wu, Y. : Chu, X.: Deng, L.: Królczyk, G.: Li, Z.	n a multi-sensor fusion perception system in s proposed in this study to monitor ship motion n in inland waterways.	2022
3	Availability of Automatic Identification System (AIS) Based on Spectral Analysis of Mean Time to Repair (MTTR) Determined from Dynamic Data Age [26]	Jaskólski, K.	Fast Fourier Transform	2022
4	Development of denoising and compression algorithms for AIS-based vessel trajectories [57]	Yan, R.: Mo, H.: Yang, D.: Wang, S.	Noise detection method based on statistical theory and sliding window. Linear interpolation is further used to rectify the glitches detected	2022
5	research on Ship Track Fusion Model-based on Distributed Extended Kalman Filter [15]	Guo, L.: Zhao, H.: Zhu, X.: Chen, D.	chip track fusion model based on distributed extended Kalman filter	2022
6	Automatic Identification System (AIS) dynamic data integrity monitoring and trajectory tracking based on the simultaneous localization and mapping (SLAM) process model [27]	Jaskólski, K.: Marchel, L.: Felski, A.: Jaskólski, M.: Specht, M.	Stochastic methods based on Markov chains.	2021
7	A novel fitting model for practical AIS abnormal data repair in the inland river [18]	He, W. : Liu, X.: Chu, X.: Fracz, P.: Li, Z.	The abnormal data were first divided into three types, i.e., erroneous data, short-time lost data, and long-time lost data. Then, a cubic spline interpolation method was employed to deal with the erroneous data and short-time lost data.	2021
8	Enhance the AIS data availability by screening and interpolation [8]	Zhang, D: Li, J:Wu, Q: Chu, X: He, W.	Piecewise cubic, Hermite interpolation and cubic spline interpolation was employed to restore the AIS data	2017
9	A multi-source information fusion method for error AIS targets identification [36]	Liu, X: He, W : Chu, X: Ma, F: Nie, Y.	Error AIS messages identification framework, which based on prior probability, expert assessment, fuzzy membership degree and Dempster-Shafer's (DS) evidence combining rule	2014

Table A.3: Cluster-based on the Keyword "Experiment of AIS data"

S.No	Title	Author(s)	Method	Year
1	Anomaly Detection in Maritime AIS Tracks: A Review of Recent Approaches [55]	Wolsing, Konrad: Roepert, Linus: Bauer, Jan: Wehrle, Klaus	Surveying research papers	2022
2	Uses and Misuses of the Automatic Identification System [24]	Iphar, Clement: Ray, Cyril: Napoli, Aldo	Methodology to assess the integrity of AIS messages	2019
3	Enhance the AIS data availability by screening and interpolation [9]	Zhang, Daiyong: Li, Jia: Wu, Qing: Liu, Xinglong: Chu, Xiumin: He, Wei	Cubic Spline Interpolation	2017
4	Countering real-Time stream poisoning: An architecture for detecting vessel spoofing in streams of AIS Data [31]	Kontopoulos, Ioannis: Spiliopoulos, Giannis: Zissis, Dimitrios: Chatzikokolakis, Konstantinos: Artikis, Alexander	Calculating Haversine Distance	2018

Table A.4: Cluster-based on the Keyword: AIS

S.No	Title	Author(s)	Method(s)	Year
1	Maritime anomaly detection and threat assessment [34]	Lane, Richard O: Nevell, David A: Hayward, Steven D: Beaney, Thomas W.	Bayesian Network and by using detector output	2010
2	Detection of false AIS messages for the improvement of maritime situational awareness [22]	Iphar, Clement: Ray, Cyril: Napoli, Aldo	Message Based approach	2015
3	Detection of malicious AIS position spoofing by exploiting radar information [29]	Katsilieris, Fotios: Braca, Paolo: Coraluppi, Stefano	Hypothesis testing framework	2013
4	DeAIS project: Detection of AIS spoofing and resulting risks [43]	Ray, Cyril: Gallen, Romai: Iphar, Clement: Napoli, Aldo: Bouju, Alain	Done by modelling risks with ontologies, analysed and detecting malicious events	2015

Table A.5: Cluster-based on the Keyword: Automatic Identification system

S.No	Title	Author(s)	Data Sources	Year
1	Data Quality Assessment – A Use Case from the Maritime Domain [51]	Stróżyna, Milena: Filipiak, Dominik: Węcel, Krzysztof	AIS data was received in January-December 2015 from Orbcomm satellites (for the whole globe). AIS data set covers weeks 33–35 of 2018 from the globe. The analysed dataset contained 65,896,367 messages. It was used to analyse: navigational status, speed over ground, course over ground, true heading, IMO number, call sign, and name	2020
2	Ship Trajectories Pre-processing Based on AIS Data [60]	Zhao, Liangbin: Shi, Guoyou: Yang, Jiaxuan	Data was collected from the AIS data centre of the Chinese Maritime Authority in Ningbo. The data is from January to February 2015	2018
3	Anomaly Detection and Restoration for AIS Raw Data [7]	Chen, S: Huang, Y: Lu, W.	Original AIS data were downloaded through the VTExplorer website for experiments. We select the part of data about Xiamen Port and surrounding waters. The time range is from December 21, 2018, to January 3, 2019; the spatial range is 117.7737° E and 24.08784° N to 118.63037° E and 24.691° N, including 12158622 pieces of position data and 387745 pieces of static data.	2022
4	AIS meets IoT: A network security mechanism of sustainable marine resource based on edge computing [6]	Chao, Han-Chieh: Wu, Hsin-Te: Tseng, Fan-Hsun	a stationary IoT-enabled (Internet of Things) vessel tracking system of a sustainable marine environment is proposed	2021
5	Rating the effectiveness of fishery-regulated areas with AIS data [52]	Tassetti A.N: Ferrà C: Fabi G.	AIS data from 2014 were purchased from a private provider with resolution high 5-min poll frequency	2019
6	AIS data to inform small scale fisheries management and marine spatial planning [25]	James, Mark: Mendo, Tania: Jones, Esther L: Orr, Kyla: McKnight, Ali: Thompson, John	Three months of AIS data derived from 274 Scottish small scale fishing vessels were used	2018
7	A comparison of VMS and AIS data: The effect of data coverage and vessel position recording frequency on estimates of fishing footprints [50]	Shepperson, Jennifer: Hintzen, Niels T: Szostek, Claire L: Bell, Ewen: Murray, Lee G: Kaiser, Michel J	Data source isn't specified due to the confidentiality of VMS and AIS data. The above data were collected from the fisherman so the fishing activity wouldn't be displayed.	2018

Table A.6: Cluster on Data sources

B
Code

```
import cv2
import numpy as np
import torch
import json
import pandas as pd
import sys
import datetime

# Get JSON file in argument of the script
json_file = sys.argv[1]
video_file = sys.argv[2]
start_time = sys.argv[3]

# Initialize a dictionary that saves previous ship's centroids, Direction and counted statu
previous_ships = []

def calculate_centroid(box):
    x1, y1, x2, y2 = box
    centroid_x = (x1 + x2) / 2
    centroid_y = (y1 + y2) / 2
    return (centroid_x, centroid_y)

def calculate_distance(centroid1, centroid2):
    x1, y1 = centroid1
    x2, y2 = centroid2
    return np.sqrt((x2 - x1)**2 + (y2 - y1)**2)


# Load YOLO and import my weights
model=torch.hub.load('ultralytics/yolov5','custom',path="C:/Users/surya/yolov5/best.pt")

# Load video
cap = cv2.VideoCapture(video_file)

# Set minimum confidence threshold for detections
```

```
conf_thresh = 0.75

# Reference line for counting for right to left direction
ref_line = 1000

# Define centroid distance threshold for ship matching
centroid_distance_threshold = 25

#initialize a parameter for the number of frames to skip
skip_frames = 20

#Initialize counts
left_to_right_count = 0
right_to_left_count = 0

#Initialize frame count
count_frames = 0
#print total number of frames in the video
print("Total number of frames: " + str(cap.get(cv2.CAP_PROP_FRAME_COUNT)))

FPS = cap.get(cv2.CAP_PROP_FPS)

while True:
    # Read frame from video
    ret, frame = cap.read()

    # Stop if end of video or if its a blank frame
    if frame is None:
        break

    # Count the frame number
    frame_number = int(cap.get(cv2.CAP_PROP_POS_FRAMES))

    # process every other frame
    if frame_number % skip_frames != 0:
        continue

    # Count the frames that are processed
    count_frames = count_frames + 1

    # Pass frame through YOLOv5s model
    detections = model(frame)

    # get detections with confidence higher than conf_thresh
    detections = detections.pred[0][detections.pred[0][:, 4] > conf_thresh]

    # loop through detections
    for detection in detections:
        # get confidence score and class index
```

```
confidence = detection[4]
class_index = int(detection[5])

# chech if the confidence is higher than the threshold
if confidence > conf_thresh:
    # get bounding box coordinates
    x1, y1, x2, y2 = detection[:4].detach().numpy().astype(np.int32)

    #centroid of the bounding box
    centroid_x,centroid_y = calculate_centroid((x1, y1, x2, y2))

    # Check if the current centroid is near any previous centroid only
    then consider it as the same ship
    matched_ship = None
    previous_centroid = None
    for ship in previous_ships:
        previous_centroid = ship["centroid"]
        if isinstance(previous_centroid, np.ndarray):
            for centroid in previous_centroid:
                distance = calculate_distance(centroid,(centroid_x, centroid_y))
                if distance < centroid_distance_threshold:
                    previous_centroid = centroid
                    matched_ship = ship
                    break
        else:
            distance=calculate_distance(previous_centroid,(centroid_x,centroid_y))
            if distance < centroid_distance_threshold:
                matched_ship = ship
                break

    # Also check if the direction is not assigned and save the direction based
    on the previous centroid centre point, even if the difference is small
    if matched_ship is not None and matched_ship["direction"] is None:
        prev_centroid_x, prev_centroid_y = previous_centroid
        # print("Previous centroid: " + str(previous_centroid) + "\n")
        if centroid_x > prev_centroid_x and centroid_y > prev_centroid_y:
            matched_ship["direction"] = "Left to Right"
            print("Direction assigned: Left to Right for "+str(matched_ship))
            #print co-ordinates of the ship
            # print("Ship co-ordinates: " + str((x1, y1, x2, y2)) + "\n")

        elif centroid_x < prev_centroid_x and centroid_y < prev_centroid_y:
            matched_ship["direction"] = "Right to Left"
            print("Direction assigned: Right to Left for " + str(matched_ship))
            # print("Ship co-ordinates: " + str((x1, y1, x2, y2)) + "\n")

    # Update the centroid of the matched ship if it's relatively close to
    the previous centroid
    if matched_ship is not None:
```

```
        if isinstance(previous_centroid, np.ndarray):
            for centroid in previous_centroid:
                if calculate_distance(centroid, (centroid_x, centroid_y))
                                            < centroid_distance_threshold:
                    matched_ship["centroid"] = (centroid_x, centroid_y)
                    break
        else:
            matched_ship["centroid"] = (centroid_x, centroid_y)

    # If no match found, create a new ship
    if matched_ship is None:
        matched_ship = {"centroid": (centroid_x, centroid_y), "direction":
                                                None, "counted": False}
        # Print values of the new ship
        print("New ship found: " + str(matched_ship) + "+++++++++++++\n")
        previous_ships.append(matched_ship)

  #Check if the box border is crossing the line and
  cross check with the direction of the ship
    if matched_ship["direction"] == "Right to Left" and x1 < ref_line
                                    and not matched_ship["counted"]:
        # print values of the matched ship
        right_to_left_count += 1
        # print("Coordinates of the matched ship: " + str((x1, y1, x2, y2)))
        matched_ship["counted"] = True
        print("Ship crossed the line from Right to Left: " + str(matched_ship))
        # Also based on FPS and current frame number, note the time of
                                                crossing the line
        match_time = (frame_number / cap.get(cv2.CAP_PROP_FPS))
        matched_ship["counted_frame"] = frame_number
        # print("Time of crossing the line: " + str(match_time) + " seconds\n")
        matched_ship["time"] = match_time

        print("Ship crossed the line from Left to Right: " + str(matched_ship))

    elif matched_ship["direction"] == "Left to Right" and x2 > ref_line
                                    and not matched_ship["counted"]:
        # print values of the matched ship
        left_to_right_count += 1
        #print coordinates of the matched ship in the frame
        # print("Coordinates of the matched ship: " + str((x1, y1, x2, y2)))
        matched_ship["counted"] = True
        # Also based on FPS and current frame number, note the time of
                                                crossing the line
        match_time = (frame_number / cap.get(cv2.CAP_PROP_FPS))
        matched_ship["counted_frame"] = frame_number
        # print("Time of crossing the line: " + str(match_time) + " seconds\n")
        matched_ship["time"] = match_time
        print("Ship crossed the line from Left to Right: " + str(matched_ship)\)
```

```
            # Draw bounding box to fit the detected object
            cv2.rectangle(frame, (x1, y1), (x2, y2), (0, 255, 0), 2)

            # Draw centroids for ships
            cv2.circle(frame, (int(centroid_x), int(centroid_y)), 5, (0, 255, 0), -1)

    # Draw reference line for counting
    cv2.line(frame, (ref_line, 0), (ref_line, 720), (0, 0, 255), 2)

    # Display counts on frame
    cv2.putText(frame, "Left to Right: " + str(left_to_right_count), (10, 50)
                                    , cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 255), 2)
    cv2.putText(frame, "Right to Left: " + str(right_to_left_count), (10, 100)
                                    , cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 255), 2)

    # Display frame
    cv2.imshow("YOLOv5", frame)

    # Press Q on keyboard to exit
    if cv2.waitKey(1) & 0xFF == ord("q"):
        break

# When everything done, release the video capture object
cap.release()

# Closes all the frames
cv2.destroyAllWindows()

# Print counts
print("Left to Right: " + str(left_to_right_count))
print("Right to Left: " + str(right_to_left_count))
print("Total number of frames: " + str(count_frames) + "\n\n")

# Print total number of ships which is the sum of left to right and right to left
print("Total number of ships: " + str(left_to_right_count + right_to_left_count))


# based on the start time of the video, calculate the time of
                                    crossing the line for each ship
start_time = datetime.datetime.strptime(start_time, "%Y-%m-%d %H:%M:%S")
for ship in previous_ships:
    if ship["counted"]:
        # Find the time of crossing the line and save as string in the ship
                                        dictionary in format YYYY-MM-DD HH:MM:SS
        time_of_crossing = start_time + datetime.timedelta(seconds=ship["time"])
        ship["time"] = time_of_crossing.strftime("%Y-%m-%d %H:%M:%S")
        print(ship)
```

```
# Load JSON file
with open(json_file) as f:
    json_data = json.load(f)

# Convert JSON to DataFrame
df = pd.DataFrame(json_data)

# Convert timestamp to datetime
df['timeLastUpdate'] = pd.to_datetime(df['timeLastUpdate'], unit='ms')

# Sort DataFrame by timestamp in ascending order
df = df.sort_values('timeLastUpdate')

# add a column to the DataFrame to save the matched ship(yes/no)
df["Matched ship"] = ""

# create a column to the dataframe to save the number of matches per range or timestamp
df["Number of matches"] = ""
# assign 0 to all the entries in the column
df["Number of matches"] = 0
# make the varable type as int
df["Number of matches"] = df["Number of matches"].astype(int)


# add a column to save the timelasupdate as a range of values
df["'timeLastUpdate': ['first', 'last']"] = ""

# if multiple entries with the same mmsi, keep one entry with the timestamp value being
        the range of the first and last timestamp, enter this value in the new column
# After a range is obtained for a ship, delete all other entries with the same mmsi
for index, row in df.iterrows():
    if df[df["mmsi"] == row["mmsi"]].shape[0] > 1:
        df.at[index, "'timeLastUpdate': ['first', 'last']"] = {"first": df[df["mmsi"]
        == row["mmsi"]]["timeLastUpdate"].iloc[0], "last": df[df["mmsi"] ==
                                row["mmsi"]]["timeLastUpdate"].iloc[-1]}
        df.drop(df[df["mmsi"] == row["mmsi"]].index[1:], inplace=True)



# Try to mach the previous_ships list by comparing the time of crossing the line
with the timeLastUpdate
for ship in previous_ships:
    for index, row in df.iterrows():
        if ship["counted"]:
            # convert timeLastUpdate to datetime
            timeLastUpdate = row["timeLastUpdate"]
            # Convert time of crossing the line to datetime
            time_of_crossing = datetime.datetime.strptime(ship["time"],
```

```
                                                   "%Y-%m-%d %H:%M:%S")
            # print("Time of crossing the line: " + str(time_of_crossing) +
                              " Time Last Update  " + str(timeLastUpdate) +   "\n")
            # if the time last update is a range of values, check if the time of
                                                    crossing the line is in that range
            if isinstance(row["'timeLastUpdate': ['first', 'last']"], dict):
                    # if time of crossing the line is in the range of timeLastUpdate,
                    then match the ship
                if time_of_crossing >= row["'timeLastUpdate': ['first', 'last']"]
                ["first"] and time_of_crossing <= row["'timeLastUpdate':
                                                    ['first', 'last']"]["last"]:
                    df.at[index, "Matched ship"] = "Yes"
                    # increase the number of matches for that range
                    df.at[index, "Number of matches"] = df.at[index, "Number of
                                                                        matches"] + 1
                    print("Matched ship: " + str(ship) + " with " + str(row["mmsi"]))
                    break
            # if the time last update is not a range of values, check if the time
                        of crossing the line is within 2 minutes of the timeLastUpdate
            else:
                if time_of_crossing >= timeLastUpdate - datetime.timedelta(minutes=2)
                and time_of_crossing <= timeLastUpdate + datetime.timedelta(minutes=2):
                    df.at[index, "Matched ship"] = "Yes"
                    # increase the number of matches for that range
                    df.at[index, "Number of matches"] = df.at[index,
                                                    "Number of matches"] + 1
                    print("Matched ship: " + str(ship) + " with " + str(row["mmsi"]))
                    break

# Also print the sum of the number of matches column in the end of the column
print("Total number of matches: " + str(df["Number of matches"].sum()) + "\n\n")

# Save only timeLastUpdate, mmsi column and if the ship is matched to that entry
into excel file from the DataFrame and print it to excel and save it in the same folder as
df.to_excel("AIS_0608_data.xlsx", columns=["timeLastUpdate","'timeLastUpdate':
['first', 'last']", "mmsi", "Matched ship", "Number of matches"], index=False)

import matplotlib.pyplot as plt
# plot a bargraph of the number of matches per timelasupdate
df.plot.bar(x="mmsi", y="Number of matches", rot=90)
#decrease the size of the label on x axis
plt.xticks(fontsize=5)
#make the label of y axis whole numbers instead of decimals
plt.yticks(np.arange(0, 20, 1))
plt.show()
#save the plot as a png file
plt.savefig("Number of matches per mmsi.png")
```

C

Design Space Exploration

Video	length (mins)	FPS	Confidence threshold	Centroid threshold	Frames skipped	Ref. line	Actual count		Algoritm count	
							L - R	R-L	L - R	R-L
Snippet 1	00:58	30	60	67	15	500	2	1	2	1
Snippet 2	15:00	30	75	50	15	500	4	2	4	1
Snippet 2	15:00	30	75	35	15	500	4	2	4	1
Snippet 2	15:00	30	75	30	15	500	4	2	4	1
Snippet 2	15:00	30	70	25	15	500	4	2	8	3
Snippet 2	15:00	30	72	26	15	500	4	2	8	1
Snippet 2	15:00	30	75	67	15	500	4	2	6	1
Snippet 2	15:00	30	75	15	15	500	4	2	6	1
Snippet 2	15:00	30	70	25	15	500	4	2	8	3
Snippet 2	15:00	30	78	25	15	500	4	2	4	0
Snippet 2	15:00	30	78	22	15	500	4	2	5	0
Snippet 2	15:00	30	78	20	15	500	4	2	5	1
Snippet 2	15:00	30	76	21	15	500	4	2	5	1
Snippet 3	15:00	30	76	25	15	850	1	3	4	2
snippet 3	15:00	30	75	25	15	850	1	3	3	4
snippet 3	15:00	30	75	30	15	850	1	3	3	3
snippet 3	15:00	30	75	45	15	850	1	3	3	2
snippet 3	15:00	30	60	30	15	850	1	3	3	2
Snippet 3	15:00	30	80	30	15	850	1	3	3	3
Final	6 hour	30	78	25	15	850	49	35	34	22
Final	6 hour	30	75	25	15	850	49	35	33	21

Table C.1: Design Space Exploration