IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 July 2023, accepted 5 August 2023, date of publication 9 August 2023, date of current version 18 August 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3303808

APPLIED RESEARCH

Toward Enhanced Support for Ship Sailing

MASSIMO CAFARO[1], (Senior Member, IEEE), ITALO EPICOCO[1], MARCO PULIMENO[1], AND EMANUELE SANSEBASTIANO[2]

[1]Department of Engineering for Innovation, University of Salento, 73100 Lecce, Italy
[2]Fincantieri NexTech, Follo, 19020 La Spezia, Italy

Corresponding authors: Massimo Cafaro (massimo.cafaro@unisalento.it) and Italo Epicoco (italo.epicoco@unisalento.it)

This work was supported in part by Fincantieri NexTech, and in part by Cineca.

ABSTRACT Ship sailing is a complex endeavour, requiring carefully considered proactive and reactive strategies in choosing the course of action that best suits the various events to be managed. Humans are already supported by different technologies for sailing, however these technologies are usually available in isolation. In this paper we show how to use simultaneously three different technologies by fusing their information in order to provide enhanced support for ship sailing. To the best of our knowledge no similar approach is reported in the literature from an operational point of view. In particular, we show how to fuse the video acquired from a camera with the information available from a radar/Lidar and an AIS receiver. The video frames are analyzed in order to automatically detect surrounding ships and seamarks, the Lidar is used to determine the average or minimum distance from the ship to the acquired targets and finally the AIS receiver logs are queried to determine, if available, useful information related to the surrounding ships, such as their geographic location, type of ship etc. Our experimental results are promising and encouraging. We believe that the simultaneous use of these technologies is a step towards fully autonomous ship sailing.

INDEX TERMS Ship detection, deep learning, lidar, AIS receiver, situation awareness, image processing.

I. INTRODUCTION

Ship sailing is a complex endeavour, requiring carefully considered proactive and reactive strategies in choosing the course of action that best suits the various events to be managed. Moreover, sailing is becoming increasingly complex due to the need to optimize the route. In other words, the aim of sailing is not just to reach a predefined destination, but it has to be cost-effective as well.

Sailing may be affected by several factors, including the water and weather conditions. As an example, the impact of wind on a ship is not uniform, since it depends on the draught condition of the ship. Therefore, different parts of a ship are affected differently. Ocean currents represent another important aspect that must be given full consideration. A key factor, related to secure sailing, is the so-called stopping distance. Ships differ in terms of distance actually covered in the event of a stop signal, owing at least to their size and actual load and ballast. When a stop signal is sent, a ship does not stop immediately owing to inertia, but instead continues moving along its direction, covering a certain distance before coming to a complete stop. The stopping distance is also heavily influenced by the wind and sea conditions, since wind and waves acting behind (or in front of) the ship increase (or decrease) the stopping distance.

Humans are already supported by different technologies for sailing, however these technologies are usually available in isolation. Examples include GPS (Global Positioning System), chartplotter, magnetic and gyro compass, radar, AIS (Automatic Identification system) etc. Nowadays, the data coming from those technologies are not displayed on the same device. Moreover, their human interfaces are quite cumbersome and require a lot of space. Operators are forced to continuously switch from one Human Machine Interface (HMI) to another and extrapolate global information on their own. Fusing the information coming from various systems into one HMI helps operators to quickly understand the surrounding environment and operate properly on the vessel from the very beginning even if they have a limited experience [1], [2], [3], [4], [5]. Moreover, it reduces significantly the required space in the wheelhouse. Therefore, ships can have the possibility of hosting several sensors without

The associate editor coordinating the review of this manuscript and approving it for publication was Sandra Costanzo.

decreasing the living space in the wheelhouse. Often, very helpful sensors are not installed on vessels due to the lack of space in the wheelhouse.

In this paper we show, to the best of our knowledge for the first time, how to use simultaneously three different technologies by fusing their information in order to provide enhanced support for ship sailing. In particular, we show how to fuse the video acquired from a camera with the information available from a Lidar (Laser Imaging Detection and Ranging) radar and an AIS (Automatic Identification System) receiver. Since every vessel longer than 15 meters must be equipped with AIS and GPS and since point-cloud sensors, such as radar or Lidar, are widely installed on vessels, those sensors have been selected to be used to track target location. The cameras have been selected to extrapolate the information deriving from human eye (e.g. target category) and to present the data to the user more accurately. The fused data related to a specific target are overlaid on the target itself.

The video frames are analyzed in order to automatically detect surrounding ships and seamarks whilst the Lidar is used to determine the average or minimum distance from the ship to the acquired targets and the AIS receiver logs are queried to determine, if available, useful information related to the surrounding ships, such as their geographic position, type of ship etc. Our experimental results are good and encouraging. Although more research is certainly needed, we believe that the simultaneous use of these and additional technologies is a step towards fully autonomous ship sailing.

The rest of this paper is organized as follows. Section II recalls related work. We review the most important datasets in Section III whilst Section IV recalls the state of the art algorithms for object detection and tracking. We discuss relevant requirements for our problem in Section V; this section outlines possible problems and risks to be mitigated along with possible solutions. We believe this is an additional contribution of the manuscript. Next, we introduce and discuss our software architecture in Section VI, and present our corresponding algorithmic solution in Section VII. Training and evaluation of the deep learning model used are reported respectively in Section VIII and IX. We draw our conclusions and outline possible future directions in Section X.

II. RELATED WORK

In this section we briefly recall related work. However, we remark here that, to the best of our knowledge, no previous work has dealt with simultaneously ship/seamark detection and tracking fusing the information obtained by a Lidar radar and an AIS receiver. Indeed, almost all of the previously published papers deal only with ship detection/classification. Some of them deal with a different combination of sensor families.

In [6], the authors propose a review of the operational requirements related to autonomous vessels, and then proceed to consider suitable sensors and relevant AI techniques for an operational sensor system. They discuss the integration of Global Navigation Satellite System (GNSS, a term that refers to a constellation of satellites providing signals from space that transmit positioning and timing data to GNSS receivers) receivers and Inertial Measurement Unit (IMU, an electronic device that measures and reports a body's specific force, angular rate, and sometimes the orientation of the body, using a combination of accelerometers, gyroscopes, and sometimes magnetometers)), visual sensors (monocular and stereo cameras), audio sensors (microphones), sensors for remote-sensing (RADAR and LiDAR, which is a method for determining ranges by targeting an object or a surface with a laser and measuring the time for the reflected light to return to the receiver) and Automatic Identification System (AIS, an automatic tracking system that uses transceivers on ships and is used by vessel traffic services). However, the manuscript is just a review and no actual integration of these technologies has been implemented.

The authors of [7] propose combining radar (but not Lidar) and AIS. They propose fusing the radar acquired targets with the AIS information using a Poisson multi-Bernoulli mixture filter. The manuscripts [8], [9], [10], and [11] all deal with the processing of remote sensing images, in particular Synthetic Aperture Radar (SAR) images with the information provided by an AIS receiver.

Related work in which ship detection is done using a Lidar include [12]. Among the many studies related to the use of SAR (or other types of remote sensing images) and/or infrared images, we recall here the following recent ones: [13], [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], [30], [31], [32], [33], [34], [35], [36], [37], [38], [39], [40], [41], [42], [43].

The papers more similar to our work, with regard to ship detection and tracking are the following ones. The authors of [44] use a YOLOv3 model in conjunction with AIS information, whilst [45] is based instead on YOLOv4. The paper [46] is based on YOLOv5 and DeepSort. Automated detection of small ships is the primary aim of [47], in which the authors propose the use of a mask regional Convolutional Neural Network (Mask-CNN) along with the Colliding Body's Optimization (CBO) algorithm with the weighted regularized extreme learning machine (WRELM) technique to classify detected ships. Detecting small ships is also discussed in [48], in which the authors propose the use of a Generative Adversarial Network (GAN along with a CNN). An enhanced Convolutional Neural Network (CNN) is used in [49] to obtain more reliable and robust detection results under adverse weather conditions, e.g., rain, haze, and low illumination. The authors of [50] and [51] both use a YOLOv3 model for ship detection.

Additional references can be found in surveys such as [52], [53], [54], [55], [56], [57], and [58].

To recap, there is a huge amount of related work, since this particular research fields attracted and still attracts many researchers approaching the problem using different solutions, especially targeting remote sensing images. Again, to the best of our knowledge, our work is the first one in which the ship/seamark detection and tracking is performed

by analyzing video frames from a camera using the deep learning model YOLOv6, fusing the information related to the targets acquired by a Lidar and the data provided by an AIS receiver.

III. DATASETS

A. DATASETS OF IMAGES OR VIDEOS

Here we recall the datasets related to images or videos including objects of interest, i.e., ships and/or sea markers.

The SeaSAw (Sea Situational Awareness) dataset [59] is a new dataset comprising 1.9 million images with 14.6 million objects associated with 20.4 million attributes of 12 object classes, making it the largest maritime dataset for object detection, classification and tracking. In addition, this dataset consists of 9 sources in combination with various RGB cameras mounted on different moving vessels, operating in different geographical locations globally, with variations in scenery, weather and lighting conditions. Data collection took place over 4 years, with rigorous efforts to select, annotate, manage and analyze the data in order to improve marine perception technology. Although access to this dataset was requested from the Sea Machines company that produced it, no response was provided by the company.

ABOships [60] is a dataset for the detection of marine vessels in the open sea and along the coast. The authors collected a dataset consisting of ship images taking into account several factors: background variation, weather, illumination, visible proportion, occlusion, and scale variation. Instances of vessels (including nine types of vessels), maritime signals and various floats were accurately annotated. Although [60] reports that the dataset is available on the website https://www.fairdata.fi/en/ in reality it is available on Zenodo [61].

Singapore Maritime Dataset [54], [62] was created using Canon 70D cameras in Singapore waters. All videos were captured in high definition (1080 × 1920 pixels). The dataset was divided into two parts: shore-based video and shipboard video, acquired by a camera placed ashore on a fixed platform and a camera placed on board a moving vessel, respectively. The videos were acquired at different locations and routes and thus do not necessarily capture the same scene. The third part consists of near-infrared (NIR) videos, also acquired using another Canon 70D camera with the hot mirror removed and the BP800 Near-IR Mid-Opt bandpass filter. This dataset was acquired by Dilip K. Prasad and annotated by student volunteers. The dataset was acquired under various environmental conditions such as pre-dawn (40 minutes before sunrise), dawn, mid-day, afternoon, evening, after sunset (2 hours after sunset) and with haze and rain from July 2015 to May 2016.

SeaShips [63] is a dataset comprising 31,455 images with 6 classes of ships. The images were collected from fixed surveillance cameras mounted on the coast, and thus lack variation in camera position and movement. This dataset is not freely available.

Harbor Surveillance dataset [64] includes 70,513 ships in 48,966 images collected from 10 different vantage points. As in the case of SeaShips, the intended use case is limited to the surveillance of a port area and is therefore a dataset without dynamic environments, which are required to aid sailing and collision avoidance. This dataset is not freely available.

The McShips dataset [65] contains images and videos with a resolution of at least 500 × 500 pixels, collected by web crawling. The dataset includes 13 classes (7 civilian ships and 6 warships) with different lighting, views, and positions.

The Marvel dataset [66], [67] includes 2 million images of 109 ship classes collected by the Shipspotting website. The main purpose is limited to image classification because the images are typically of ships in an ideal situation (close-up view, plain background, and clear weather). The images are not representative of real scenarios with varying scale objects and harsh weather and lighting conditions and therefore will not be considered.

The GLSD dataset [68] includes 140,616 objects annotated from 100,729 images. Some of the images were collected with a video monitoring system, while the others were collected by web crawling. Although the web images are different, they are not similar to the views and challenges observed during collection on a moving vessel, and therefore this dataset will not be considered. The VAIS dataset [69] includes paired ship images in the visible and infrared, consisting of 1623 visible images from 15 categories; since it is limited in quantity, it will not be considered.

B. DATASETS RADAR/LiDAR AND AIS

The availability of specific radar/LiDAR and AIS datasets is scarce. Moreover, specifically, radar/LiDAR datasets are strictly dependent on the type and characteristics of the hardware with which the data are acquired. Therefore, although some datasets are available (e.g., ''Dataset for LiDAR-based Maritime Perception'' [70]), they have not been used in the context of this work as the AIS data provided by Fincantieri NexTech S.p.A. were found to be sufficient. Regarding AIS data, various examples are available, but in this case, the formal specification of AIS message encoding [71] is sufficient to proceed with decoding and use of related information when available.

C. DATASETS PROVIDED BY FINCANTIERI NexTech S.p.A

Fincantieri NexTech S.p.A. provided a dataset acquired from a campaign in open water sailing by acquiring both photographic images, LiDAR data, and raw data obtained from the AIS receiver installed on the vessel (no data on the vessel's GPS position). The data all have a time marker that allows the different datasets to be synchronized. The images have a resolution of 960 × 1280 pixels and are acquired from 6 different angles around the vessel, one of which is oriented in the direction of sailing. Images are acquired at an average rate of 57 images per minute; the dataset contains a combined total

of 9476 images. Each image is associated with a timestamp that allows it to be correlated with LiDAR and AIS data. The LiDAR data are in CSV (Comma Separated Values) format with the following fields:

- point Date and Time;
- point x coordinates;
- point y coordinates;
- point z coordinates;
- point intensity value.

The dataset provided contains an acquisition over a time window of 57 minutes with an average frequency of approximately 21400 points per second. The associated AIS data were acquired from the on-board AIS receiver and consist of textual Log files in raw format with AIVDM-encoded messages. The messages in the Log files are mainly positioning messages of type 1, 2 or 3 and for which no specific timestamp is associated. The timestamp is associated with the file, thus indicating the start of the acquisition. GPS data from the vessel are not available.

IV. DETECTION AND TRACKING: STATE OF THE ART

Object detection is one of the basic problems for computer vision, in which it is necessary to predict whether objects belonging to certain categories are present in an image and to provide their location (bounding box or pixel-level localization in the case of segmentation) if any are found. Typically, this is achieved by extracting features from an image and comparing them to trained images. Traditional approaches use sliding windows to generate proposals, then visual descriptors to generate an embedding, which are then classified using approaches such as SVM, bagging, cascade learning, and AdaBoost. Traditional algorithms with the best performance focus on accurate descriptor design to extract features (SIFT, Haar, SURF). However, more and more limitations of this approach have emerged since 2008 [55]:

- hand-annotated visual descriptors provided a large number of proposals, which caused a high rate of false positives;
- visual descriptors extract low-level features, but are not suitable for high-level features;
- each stage of a detection pipeline is optimized separately, so global optimization is difficult to achieve.

In the early 2010s, deep learning approaches came to the fore and began to replace traditional ones. Object detection networks can be classified into two types: one-stage detectors and two-stage detectors. The structure of the latter resembles traditional object detectors in that they generate proposal-regions and then classify proposals, whereas the former consider locations within an image as potential objects and try to classify them immediately. The traditional sliding-window approach to proposal generation is still used in CNN convolutional neural networks, but other noteworthy advances have emerged that enable more efficient proposal generation, such as anchor- and keypoint-based approaches (CenterNet is one of the most notable examples of its kind) [55].

However, the fundamental difference between traditional object detection and CNNs stems from the way visual descriptors are generated. With CNNs, instead of creating visual descriptors by hand, convolutional layers are used. Instead of defining feature extractors by hand, basic CNNs train multiple convolutional layers to extract high-level and low-level features, which are then classified with the help of fully connected layers. The resulting network essentially solves all the major limitations of a traditional approach, but the trade-off is that it requires significantly more training images for hyperparameter optimization [56]. Although the requirement for a large number of training samples may prove to be a major obstacle, one of the advantages of CNN-based models is that they can be generalized to other domains with similar characteristics through transfer learning. By training a model on a specific dataset, the backbone of the model can later be used to extract features in other domains with similar characteristics. For this reason, the goal of recent CNN models has been to be as general as possible, since with the help of transfer learning they can be specialized for the field of interest.

The challenge, however, arises when these generic models are not suitable feature extractors for a new field and there is insufficient data to train them [57]. In these specific cases, the only possible solution is the creation of new datasets. In light of the deep learning paradigm, recent research has focused on two main directions to increase the performance of neural networks dedicated to object detection. The first is based on improving the convolutional neural network itself by adjusting the underlying architecture or increasing the depth of the network. The first attempt to adjust the network architecture was made by the ZF network in 2014 [72]. Other representative examples include Google's Inception series [73], [74], [75]. Based on the idea that deeper networks should lead to higher accuracy in object detection, a number of studies have engaged in deepening the layers of the network. Representative works in this branch include VGGNet [76] and ResNet [77]. In addition, Inception ResNet [78] and ResNetXt [79] combine the advantages of these two, achieving better detection results.

With reference to the second direction, research has focused on optimizing deep learning-based object detection algorithms, including region-based and regression-based detection algorithms. Region-based algorithms begin with R-CNN [80] and later researchers proposed a number of variants such as SPP-net [81], Fast R-CNN [82], Faster R-CNN [83], R-FCN [84] and Mask R-CNN [85]. The amount of computation of these types of algorithms is high, although the detection accuracy is very high. End-to-end object detection algorithms generally include YOLO in its various versions and many variants [86], and SSD [87]. They allow location and category to be determined directly from a single neural network. As a result, multiple objects in an image can be quickly detected, although at the same time accuracy in

detecting location is sacrificed. SiamRPN++ [88] is a recent algorithm based on Siamese neural network.

Object tracking schemes can be classified as region-based tracking, active contour-based tracking, feature-based tracking and pattern-based tracking [58]. In region-based tracking, object tracking is achieved by using the variations of image regions that correspond to the moving object. Active contour-based tracking uses the object contour as the bounding contour and updates the contour dynamically in successive frames. Feature-based tracking uses the elements of an object (such as color, area, segment, and vertex) as features, which are then matched between successive frames to perform object tracking. In model-based tracking, object tracking is achieved by matching a projected model of the object to image data, where the object model is produced based on available prior information. Hu et al. [89] use a region-based approach for ship tracking as it is fast and easy to implement. The block-matching algorithm is the most commonly used method for region-based tracking due to its simplicity and efficiency for finding motion information. Hu et al. introduce an improvement of the full search algorithm based on adaptive block matchings with the removal of sea waves present in the background of the images to achieve fast and reliable ship tracking. The full search algorithm uses a modified search region obtained with a coarse to fine pattern. The complete search algorithm is able to find the best matching block among all possible search locations in the modified search region. This algorithm can greatly reduce the computational cost of obtaining the optimal motion estimation result. The full search algorithm proposed by Hu et al. outperforms the other block matching algorithms (full search (FS), three-step search (TSS), four-step search (FSS), diamond search (DS), hexagon-based search (HEXBS) and block-based gradient search (BBGDS)) in terms of matching error, execution time and number of searches [90].

Algorithms for tracking include Deep Sort, often used for vessel tracking [91]. The algorithm adds matching of apparent feature information to improve tracking performance. This extension allows the algorithm to track the target over a longer period of occlusion, effectively reducing the number of identity exchanges between overlapping objects. The algorithm uses the classical Kalman filter algorithm to predict the position of the tracked target in the current frame and update the tracker parameters. Other algorithms include STARK [92] and QDTrack [93]. Lee et al. [44] proposed detection, localization and tracking methods applied to videos taken from real vessels. The results obtained were compared with AIS data and showed that the proposed algorithm can be effectively used for environmental awareness. An approach aimed at identifying and tracking objects in the maritime environment by combining probabilistic data is reported in the study proposed by Haghbayan et al. [94].

V. REQUIREMENTS

Our research was aimed at defining an algorithmic solution suitable for the identification and tracking of floating objects by a moving vessel based on surveys made by a video camera and a 360Â° radar/LiDAR scanner; when available, AIS data will also be used. The algorithmic solution should be able to:

- automatically detect the presence of static or moving floating objects on the water plane in the field of view of a video camera installed on board a vessel at the bow in combination with the acquisitions of a 360Â° LiDAR Scanner installed on the vessel;
- enable the identification and tracking of objects encountered during sailing (buoys, small rowing boats, small motor boats, fishing boats, yachts, merchant ships, naval vessels, etc.).

The effective identification capability of the algorithmic solution has been evaluated on the basis of the objects appearing in the available datasets and the effectiveness in identification and tracking in operational scenarios. It is worth recalling here that the solution must not necessarily work in real-time, in the sense that the first identification of the object may take a few seconds, but once the identification is confirmed, it should be able to perform tracking in near real-time regardless of its movements with regard to the video camera and LiDAR, with performance referring to a workstation of adequate characteristics and in any case such that its use on board a ship is feasible. In order to design our algorithmic solution, we took into account the constraints presented in the following subsections.

TABLE 1. Valid resolutions.

resolution and megapixels	dimensions in pixels	aspect ratio
1080p, 2 MP	1920 x 1080	16:9
3 MP	2304 x 1520	16:9
5 MP	3072 x 1728	16:9
4K UHD, 8 MP	3840 x 2160	16:9

A. REQUIREMENTS FOR VIDEO DETECTION AND IDENTIFICATION OF STATIC OR MOVING FLOATING OBJECTS ON THE WATER PLANE

Regarding video detection and identification of static or moving floating objects on the water plane, the requirements are as follows: the camera installed at the bow of the vessel must have a minimum resolution of 1080p. Table 1 shows a set of possible resolutions valid for this application.

However, it should be emphasized that resolution alone is not enough to ensure adequate quality footage. Indeed, the camera sensor is also characterized by additional parameters that determine its ability to capture high-quality images even in low light or, conversely, in situations where there is excessive sunlight. Finally, situations in which a high dynamic range (WDR) is present can be a problem. The sensor can be of two types: CCD (Charge Coupled Device) or CMOS (Complementary Metal Oxide Semiconductor). CCD is used in professional cameras, costs more but produces better quality, brighter images. CMOS is commonly used in consumer cameras, partly because CMOS-type sensors have improved their performance over time, narrowing the quality gap with CCD sensors.

Other parameters to consider are the following ones.

- Frames per second (fps): represents the number of images captured in one second. Since in addition to the ''detection'' of objects of interest, their ''tracking'' is also required, a minimum of 15 fps is necessary. A rate of 24 fps is adequate for detection and tracking; higher values, from 60 fps onward, are useless as they do not achieve better accuracy while requiring more processing time;
- type of lens: the lens should be a wide-angle type, with a focal length between 18 mm and 35 mm. This type of lens has a larger angle of view, and it widens the view, making it ideal for framing panoramas and landscapes. However, the framed objects are smaller, so the initial detection is more complicated. Taking into account that the overall analysis also involves data acquired by radar/LiDAR, it is preferable to use a wide-angle lens rather than a normal lens with a focal length between 50mm and 70mm. These lenses are characterized by a field angle roughly equivalent to that of the human eye, so objects appear larger, but the area imaged is less large;
- lens focus: this is a critically important parameter, as improper focus does not allow capturing sharp images, making detection and tracking tasks much more difficult or even impossible. From this point of view, it is necessary to set up the camera for the use of autofocus (Auto Focus, AF and continuous AF), by which the camera constantly attempts to focus on objects, even if they are moving. Although modern autofocus systems exhibit good performance, it is good to remember that, in any case, there are limitations, whereby an AF system may exhibit various problems with small and/or moving objects, i.e., one of the use cases of the application, where detection and tracking of new objects that become part of the view taken by the camera but are located far away from the vessel is necessary;
- stabilization of the lens: since the sea is not always calm, the camera will consequently be subject to wave and jerk movements. From this point of view, it is worth noting that, depending on sea conditions, with reference to the Douglas scale (or with reference to wind strength to the Beaufort scale) exceeded values 2 (2 or 3 for the Beaufort scale, respectively), it is highly likely that the detection and tracking capability of the algorithm will be strongly affected, being adversely affected by excessive deviations of the imaged view. Therefore, although lens stabilization certainly cannot completely solve the problems associated with prohibitive sea conditions (from a value of 3 on the Douglas scale and a value of 4 on the Beaufort scale), it is certainly a useful aid. A quality image requires that the lens be equipped with optical rather than digital stabilization, both because optical stabilization is inherently superior to digital stabilization and because of the processing time required by digital stabilization;
- angle of incidence of the camera with respect to the sea: since it is not possible to install the camera so that the captured view coincides the plane corresponding to the level of the calm fluid with the last acquisition line of a frame, the angle of incidence of the camera with respect to the sea must be such as to maximize the detection and tracking capability;
- the video captured by the camera should not be subject to any compression: for example, cameras allow the resolution to be artificially reduced by compressing the video in order to save storage space. For this application, no compression and/or artificial reduction of image quality must be active;
- brightness must be such that images are captured that are not affected by noise and that any objects in the camera's view can be distinguished: specific adverse weather conditions affect brightness even during the daytime, and as one approaches sunset or in periods after sunset, brightness gradually decreases to the point where images usable in the application cannot be captured.

B. REQUIREMENTS FOR FUSION OF FLOATING OBJECT DATA IDENTIFIED THROUGH VIDEO ACQUISITION PROCESSING AND LiDAR DATA

In order to correlate the data obtained by processing the video stream from the camera installed at the bow of the vessel with the data from the radar/LiDAR, the following requirement is essential: the radar/LiDAR must be installed so that it completely covers the view taken by the camera. Typically, a radar/LiDAR has a horizontal field of view of 360Â°, but a limited vertical field of view. In addition, the measurable range is typically less than 750 m.

Consequently, because the view taken by the camera using a wide-angle lens can easily exceed a range of 750 m, and because depending on the actual installation of the LiDAR and the camera, it is very easy for the LiDAR not to provide a 100% overlap with the camera view, correlation may not be possible for all objects detected using the camera or radar/LiDAR. It will also be necessary to proceed with time synchronization between data acquired from radar/LiDAR, from the camera, and AIS data (where available). In addition to time synchronization, it is necessary that the spatial reference system be consistent between LiDAR and camera. Specifically, it is necessary to calibrate both the LiDAR and the camera so that they are oriented at the same angle and the positioning offset is known.

This calibration can take place offline at the time of installation of the devices on board the vessel or online, that is, during data acquisition. In the latter scenario, it will be necessary to provide fixed markers, whose coordinates with regard to radar/LiDAR are known, to be placed on the vessel so that they are visible from both the camera and the radar/LiDAR to record the video images and radar/LiDAR data with each other using the appropriately placed markers as a reference. The radar/LiDAR data allow the detection of obstacles using a reference system relative to the vessel itself, whilst the AIS

data provide the position of any neighbouring vessels using a terrestrial reference system; in order for the AIS data to be fused as well, it is necessary to acquire the GPS coordinates of the vessel with its timestamp.

C. REQUIREMENTS FOR TRACKING DETECTED FLOATING OBJECTS, OPERATING ON VIDEO USING THE OUTPUT RESULTING FROM VIDEO AND RADAR/LiDAR DATA FUSION

Tracking of detected objects by operating on the video stream is completely independent of radar/LiDAR data processing. For tracking, the use of an object detector capable of supporting real-time tracking on a CPU or GPU of adequate power is essential. Therefore, the choice of the specific object detector must not be based solely on considerations exquisitely related to parameters such as accuracy, support for a specific programming language etc., but also on the performance that can be achieved by evaluating the time required for inference. As for the radar/LiDAR data, it will be used exclusively for objects detected by radar/LiDAR; this will be a subset of the objects detected by video stream analysis. Whenever possible, radar/LiDAR data will be used to confirm the video detection and possibly estimate the distance.

D. REQUIREMENTS FOR INTEGRATION OF FUSED VIDEO AND RADAR/LiDAR DATA WITH AVAILABLE AIS DATA

AIS data related to objects detected by analysis of the video stream and/or radar/LiDAR radar data will not necessarily be available (e.g. small boats are not equipped with an AIS radar system, or objects such as buoys etc.). To take advantage of any available AIS data requires:

- the availability of an appropriate SDK (Software Development Kit) that allows, through specific APIs (Application Programming Interfaces), to make queries related to the geographic area corresponding to the camera view;
- the geographic area being queried will be identified by latitude and longitude, for which a Global Positioning System (GPS) receiver is itself essential as a minimal requirement or, if possible, a Galileo receiver since the Galileo system provides better accuracy.

VI. SOFTWARE ARCHITECTURE

This section will provide detailed information about the architectural specification of the software. Specifically, the architecture is based on the following components:

- Management of messages provided by AIS;
- Management of the point cloud provided by the radar/LIDAR;
- Management of video frames acquired by the camera.

A. MANAGEMENT OF MESSAGES PROVIDED BY AIS

The purpose of this software component is to allow the user a query that, taking as input a log file produced by the AIS and a bounding box defined by two points in geographic coordinates (latitude, longitude) around the vessel, performs decoding of AIS messages in which the geographic coordinates provided by neighbouring vessels are present, and reports as output every vessel (among those in the AIS messages available in the log) whose geographic coordinates fall within the bounding box provided in input.

INPUT: <AIS log stream pathname> <lat1> <lon1> <lat2> <lon2>

Specifically:

- positive latitude values correspond to latitude in the N (northern) hemisphere;
- negative values of latitude correspond to latitude in the S hemisphere (south);
- positive values of longitude correspond to longitude in the E hemisphere (east);
- negative longitude values correspond to longitude in the W hemisphere (west);

OUTPUT: data structure containing <key, value> associations in which:

- the key field is an integer that uniquely identifies a vessel in the corresponding AIS radar message;
- the value field represents a geographic point via coordinates (latitude, longitude).

Therefore, for the correct definition of the geographic region to be used for the query, it is necessary to know the geographic coordinates (latitude, longitude) of the vessel on which the camera, AIS receiver and LIDAR radar are installed. In addition, it is also necessary to know the course, heading and bearing of the vessel. The course is the intended direction of travel. Ideally (but rarely) it coincides with the heading. On a GPS receiver, the actual direction of movement is called Course Over Ground (COG). Heading is the direction in which a ship is pointing at any given time. It is expressed as an angular distance from north, usually 0Â° north, clockwise up to 359Â°, in degrees of true, magnetic, or compass heading. It is a value that constantly changes as the boat swerves back and forth on course due to the combined effects of sea, wind, and steering error. Generally, determining the course is the job of the IMU (inertial measurement unit). However, only the best IMUs are able to do this well at low speed. Under such circumstances, a GNSS device with two antennas can be used.

A bearing is the direction from one location to another, measured in degrees of angle from an accepted reference line. When using compass bearings, the reference line is north, so ‘‘the beacon is on a bearing of 270Â°’’ means ‘‘the beacon is west of us.’’ When using relative bearings, the reference line is the centerline of the boat. Thus, at the bow is 0Â°, and a buoy to starboard (a nautical term meaning ‘‘90Â° to the right when facing forward’’) corresponds to 90Â°. GPS receivers provide a constantly updated bearing of an active waypoint. Using the vessel’s coordinates and course, heading, and bearing information, it is possible to derive the two points to be used for the query.

B. MANAGEMENT OF THE POINT CLOUD PROVIDED BY THE RADAR/LIDAR

The purpose of this component is the management and use of the point cloud provided by the radar/LIDAR, in order to infer information about the approximate distance of a point corresponding to a pixel in a frame acquired by the camera. Therefore, a pre-processing and filtering operation of the points that constitute the radar/LIDAR point cloud is necessary in order to extract the points corresponding to the acquired frame and then project them onto the related image. Once the points have been extracted, their estimated distance can be calculated using the Euclidean distance formula.

C. MANAGEMENT OF VIDEO FRAMES ACQUIRED BY THE CAMERA

The purpose of this component is the detection and tracking of boats and buoys by analyzing frames acquired by a camera installed at the bow on the boat. Specifically, this software component is responsible for extracting frames from the acquired video, and providing them as input to an artificial intelligence-based model that can classify objects in a frame as belonging to either the boat class or the seamark class. The model must also be able, after performing object detection, to proceed with tracking the detected object where present in subsequent frames of the video stream.

VII. THE DATA FUSION APPROACH

Fusion of multi-source data can be challenging due to different (i) spatio-temporal reference systems, (ii) density of information and (iii) data formats among the various sensor streams to be fused. Furthermore, particular settings, such as navigation aid systems, put additional computational burden on the fusion process in order to ensure real-time processing. In particular, the first challenge to be faced in order to compute the 2D projection of 3D point cloud to the image plane is related to deriving the correct intrinsic and extrinsic camera parameters. Intrinsic parameters, which are camera specific, include the focal length (f_x, f_y) and optical centers (c_x, c_y). These parameters are then used to create a camera matrix, which is again camera specific. Extrinsic parameters are related to the required geometric transformations, and include rotation and translation vectors needed for the projection from a 3D point cloud object coordinate space to the camera coordinate space. Moreover, another vector related to distortion coefficients is also required in order to correct radial and tangential camera distortions. To solve these issues, Fincantieri NexTech did a calibration process directly on board the ship which acquired the experimental dataset used in this work. The calibration takes advantage of a well-known target, a specific marker corresponding to key positions of the sensors. The output of the calibration has been then hand-tuned to optimise the extrinsic parameters to be used. An additional issue is related to the different time of acquisition of the Lidar and the camera sensor data, which requires extracting from the data streams matching 3D Lidar points and corresponding camera frame with regard to time. In our case, owing to different frequencies of acquisition, we had to match 3D point clouds and camera frames within one second. We remark here that this tolerance is acceptable, considering the ship sailing dynamics. Regarding AIS data, the main challenge in the fusion process was to match the video frame acquisition time to the timestamp of the AIS log files. In order to solve this issue, since AIS positioning messages of type 1, 2 and 3 do not include a full timestamp, Fincantieri NexTech provided us with AIS log files whose filename included a timestamp. The tolerance in this case was set to one minute.

Regarding the proposed algorithmic solution, this is based on the three components described in the section on software architecture.

A. AIS MESSAGES

The implementation of the geographic query is based on the C++ library libais v0.15 developed by Kurt Schwehr, available as open-source at the url https://github.com/schwehr/libais. However, since several modifications to the library itself were necessary, the provided software also includes the modified library. Specifically, the decoded AIS messages are as follows:

- messages 1, 2, 3: Position reports;
- message 4: Base station report
- message 5: Ship static and voyage related data, used by Class A shipborne and SAR aircraft AIS stations;
- message 18: Standard class B equipment position report, output periodically and autonomously instead of Messages 1, 2, or 3 by Class B shipborne mobile equipment, only.

Message decoding is based on the specification document "Recommendation ITU-R M.1371-5 (02/2014) Technical characteristics for an automatic identification system using time division multiple access in the VHF maritime mobile frequency band," available at url https://www.itu.int/rec/R-REC-M.1371-5-201402-I. The implemented functionality is accessible via a Python wrapper, developed using PyBind11, available at https://github.com/pybind/pybind11.

B. RADAR/LiDAR DATA

Since the radar/LIDAR radar data were provided in csv (comma separated values) format, it was necessary to use the following tools, available as open-source:

- LASTools: https://rapidlasso.de/product-overview/
- OpenCV: https://opencv.org

LASTools was used for data conversion to the standard LAS format, and subsequent point cloud filtering. OpenCV is a library for software development based on computer vision algorithms. This library was used for the projection of points onto the frame acquired by the camera. Again, the interface is available through Python. The point cloud acquired by radar/LIDAR is projected onto the image corresponding to the frame acquired by the camera through the point projection

algorithm available in OpenCV. Since the technical specifications of the camera (precise positioning in space relative to the LIDAR radar, focal length (f_x, f_y), and optical centers (c_x, c_y)) were not available, manual tuning was performed to determine the intrinsic camera matrix. Similarly, manual tuning was necessary to determine the extrinsic parameters (rotation matrix and translation vector).

C. VIDEO FRAMES

Among the various available model, YOLO (You Only Look Once) v6, available as open-source at url https://github.com/meituan/YOLOv6, was chosen. Although YOLO v7 was also available, YOLO v6 was officially released after YOLO v7 and was therefore the most recent of the available YOLO models. YOLO v6 provided impressive results, excelling in terms of detection accuracy and inference speed. The initial code for YOLO v6 was released in June 2022. The first document, together with the updated version of the model (v2), was released in September 2022. YOLO v6 is considered the most accurate of all object detectors. This is evident from the fact that the YOLOv6 nano model achieved a mAP of 0.363 (36.3%) on the 2017 COCO dataset (400 epochs). It also runs at over 1200 FPS on an NVIDIA Tesla T4 GPU with a batch size of 32. The following datasets were chosen for model training:

- ABOships;
- COCO 2017, which is a large-scale dataset for object detection, segmentation and captioning. Specifically, 3146 images with class ''boat'' were extracted from the dataset to be used in addition to those in the ABOships dataset. The corresponding annotations are 11189.

Since the annotations in the ABOships dataset are in YOLO format while the annotations in the COCO dataset are in a different format (COCO format, based on JSON), a conversion of the COCO annotations was necessary. Taking into account that the ABOships dataset includes 9 types of vessels (corresponding to the classes boat, cargoship, cruiseship, ferry, militaryship, miscboat, miscellaneous, motorboat, passengership, sailboat) and one buoy object (corresponding to the class seamark), we decided to consider only one generic class boat and the class seamark. However, only 7670 annotations are related to seamark class objects, whilst there are 34,297 annotations related to boat class objects. The 7670 annotations related to seamark class objects are found in 3744 images, whilst the 34,297 annotations related to boat class objects are found in 6136 images. Thus, we experienced the so called ''class imbalance'' problem, and we decided to proceed by extending the dataset through additional seamark class images generated through data augmentation. For this purpose, we used the Albumentations library, available as open-source at the url https://albumentations.ai. Therefore, the starting dataset included a total of 13,026 images (of which 840 were discarded because they lacked annotations) and 53186 annotations. After the data augmentation process, the final dataset includes 30903 images and 91298 annotations. The operations performed to create the new images include:

- Horizontal Flip;
- Rotation;
- Changing brightness and contrast values;
- Addition of Gaussian noise;
- Motion Blur;
- Defocus.

D. PUTTING ALL TOGETHER

The algorithmic solution is based on the OpenCV library for handling the video stream from the camera. The video stream is acquired one frame at a time and processed by the YOLOv6 model. The software processes the data from the radar/LIDAR by extracting from the point cloud the 3D points corresponding to the frame acquired from the camera in order to project them onto the frame and estimate the distance of the objects identified in the camera frame. In addition, the software performs a geographic query using -if available-any AIS messages in the AIS log file. The purpose of the application is to output a succession of frames in which any boat-class and/or seamark-class objects have been correctly identified. For each object, the software provides the estimated distance to the radar/LIDAR installed on board the boat and, in the case of objects corresponding to boats that have sent AIS messages within a relatively short time of their identification, also additional information such as IMO number, call sign, name, and geographic coordinates, if available.

We remark here that, to the best of our knowledge, this is the first attempt to fuse information coming from a camera, a radar/LIDAR and an AIS receiver. We experienced technical issues that have been reported, in terms of corresponding requirements, in Sections V-B, V-C and V-D. In particular, it is in general difficult to properly project the radar/LIDAR cloud made of 3D points onto the 2D frame acquired by the camera, registering the points through the correct geometrical transformations (rotations, translations etc).

VIII. TRAINING THE YOLO MODEL

For training, we selected a YOLOV6 nano model (YOLOv6-n) because, although it is the smallest available model size, it already includes a number of weights to be learned equal to 4.3 million and is the fastest in inference (albeit with lower accuracy than the other models with a larger number of weights). The model was trained using a parallel cluster compute node equipped with 2 IBM POWER9 AC922 3.1 GHz 16 cores processors, 256 GB RAM and 4 NVIDIA Volta V100 GPUs, Nvlink 2.0, 16GB. Model training was performed using The following parameters:

- Optimization algorithm: Sthocastic Gradient Descent with momentum and cosine decay learning rate;
- Weight decay: Exponential Moving Average;
- Training dataset: 70% of the dataset, chosen pseudo-randomly;

- Validation dataset: 15% of the dataset, chosen pseudo-randomly;
- Test dataset: 15% of the dataset, chosen pseudo-randomly.

The model was trained for 800 epochs with batch size of 128, and took a total of about 26 hours to train using 4 NVIDIA Volta V100 GPUs. The metrics commonly used for performance validation are Average Precision and Average Recall. Average Precision (mAP) is a metric used to evaluate object detection patterns. Average Precision (AP) is calculated on the recall values from 0 to 1. The mAP formula is based on the following secondary metrics:

- Confusion matrix;
- Intersection over Union (IoU);
- Recall;
- Precision.

A. CONFUSION MATRIX

The confusion matrix includes four attributes:

- True Positives (TP): The model predicts a label, and the label is correct (with respect to ground truth, a term for the certain information we have: for each training, validation and test image, the ground truth includes the class of objects in it and the bounding rectangles of each object);
- True Negatives (TN): the object is not part of the ground truth, and the model does not predict a label. This attribute is not used in object detection tasks, as it is not useful for the purpose;
- False Positives (FP): The model predicts a label, which is not part of ground truth (type I error);
- False Negatives (FN): The model does not predict a label, which is part of the ground truth (type II error).

As an example of a TP, given a dog object, there is a dog prediction, whilst for a FP given cat object there is a dog prediction. Similarly, as an example of a FN given a dog object there is a non-dog prediction whilst for a TN given a cat object there is a non-dog prediction.

B. INTERSECTION OVER UNION

Intersection over Union (IoU) indicates the overlap of the coordinates of the bounding box predicted by the model (related to the identification of a given object) with the actual bounding box. A higher IoU indicates that the coordinates of the predicted bounding box are very similar to those of the actual bounding box. Specifically, IoU is the ratio of the area of the intersection to the area of the union.

C. PRECISION

Precision is defined as the total number of correctly identified objects with regard to the total number of identified objects. Therefore, this metric quantifies the number of false positives reported by an algorithm in output: a precision of 1 (or 100%) means that there are no false positives (i.e., precision is a quality metric). Specifically, precision is defined as the ratio of TP to the sum TP + FP. For object detection, an IoU threshold is normally used whereby a given detected object is classified as FP (IoU of object < IoU threshold) or TP (IoU of object > IoU threshold).

D. RECALL

Recall, on the other hand, is the total number of correctly identified objects reported versus the number of correctly identified objects provided by an exact algorithm: a recall equal to 1 (or 100%) means that there are no false negatives. Therefore, this metric is a measure of completeness (i.e., recall is a quantitative metric). Specifically, recall is defined as the ratio of TP to the sum TP + FN.

E. AVERAGE PRECISION

Average Precision (AP) is calculated as a weighted average of the accuracies obtained for each threshold; the weight is the increase in recall from the previous threshold. Mean Average Precision (mAP) is the average of the APs for each class. However, the interpretation of AP and mAP may vary in different contexts. For example, in the COCO challenge evaluation paper for object detection, AP and mAP are the same thing.

The following steps should be taken to calculate AP:

- Generate prediction scores using the model;
- Convert the prediction scores into class labels;
- Determine the confusion matrix TP, FP, TN, FN;
- Compute precision and recall metrics;
- Compute the area under the precision-recall curve;
- Measure average precision (mAP) by determining the AP for each class and then averaging the values.

The mAP takes into account the trade-off between precision and recall and considers both FPs and FNs. This property makes it valid as a metric for object detection applications. The metric for the 2017 COCO challenge is computed as follows:

- compute the AP for the IoU threshold of 0.5 for each class;
- precision is determined for each recall value (0 to 1 with a step of 0.01), then repeated for IoU thresholds of 0.55, 0.60,...,0.95;
- the average is computed over all of the 80 classes in the COCO dataset and over all of the 10 thresholds used; moreover, additional metrics are used to identify the accuracy of the model on different scales of objects (AP_{small}, AP_{medium}, and AP_{large}).

Specifically: AP_{small} is related to small objects (area < 322), AP_{medium} is related to medium-sized objects (322 < area < 962) and AP_{large} is related to large objects (area > 962). Finally, Average Recall is computed by considering images in which there is at most 1 detection (at most one object is identified by the model), at most 10 detections and at most 100 detections. For this purpose, values called $AR_{max=1}$, $AR_{max=10}$ and $AR_{max=100}$ are commonly reported. But, also

for Average Recall, values are computed with regard to the dimensions AR_{small}, AR_{medium} and AR_{large}.

The results obtained at the end of training with respect to the metrics Average Precision and Average Recall are as follows:

Average Precision
(AP)@[IoU=0.50:0.95 | area= all | maxDets=100] = 0.251
(AP)@[IoU=0.50 | area= all | maxDets=100] = 0.539
(AP)@[IoU=0.75 | area= all | maxDets=100] = 0.199
(AP)@[IoU=0.50:0.95 | area= small | maxDets=100] = 0.138
(AP)@[IoU=0.50:0.95 | area= med | maxDets=100] = 0.417
(AP)@[IoU=0.50:0.95 | area= large | maxDets=100] = 0.604
Average Recall
(AR)@[IoU=0.50:0.95 | area= all | maxDets= 1] = 0.142
(AR)@[IoU=0.50:0.95 | area= all | maxDets= 10] = 0.345
(AR)@[IoU=0.50:0.95 | area= all | maxDets=100] = 0.390
(AR)@[IoU=0.50:0.95 | area= small | maxDets=100] = 0.272
(AR)@[IoU=0.50:0.95 | area= med | maxDets=100] = 0.594
(AR)@[IoU=0.50:0.95 | area= large | maxDets=100] = 0.724

Note that the accuracy obtained, equal to AP @[IoU=0.50:0.95 | area=all | maxDets=100] = 0.251 is in line with the quality of the dataset used for training, and consistent with the use of a YOLOv6 nano model with a limited number of weights to be learned (equal to 4.3 millions). From this point of view, it is worth recalling here that there is a tradeoff between the size of the model used, the computational time required for training, and the achievable accuracy. Higher accuracy values can be easily obtained by using larger, higher quality datasets and larger model sizes.

IX. VALIDATION OF RESULTS

The dataset provided by Fincantieri NexTech S.p.A., which includes some images, radar/LIDAR data and AIS logs acquired during an experimental campaign, was used for performance evaluation. The following images (see Figures 1, 2 and 3), produced as output by the software, show the detection and tracking of objects performed by the software as the vessel moves, along with the computation of the average distance of the identified object from the vessel. In the images, red points represent points acquired via radar/LIDAR (3D point cloud), projected onto the 2D image. Note that the boat at the bottom of the images is correctly detected by YOLOv6 nano (the bounding rectangle is drawn), but because the radar/LIDAR does not cover the distance required to acquire the corresponding 3D points, the average distance cannot be computed.

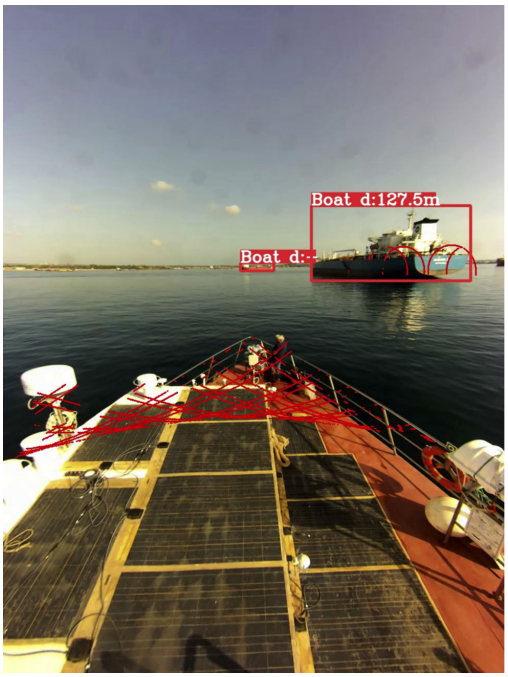


FIGURE 1. Approaching a vessel: frame 1.

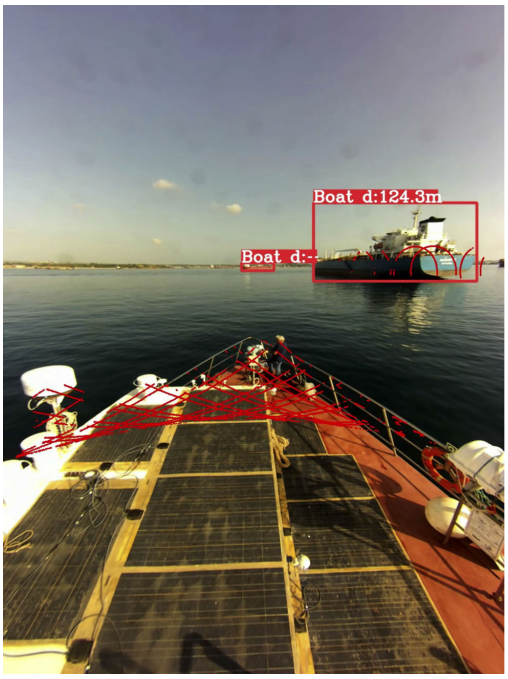


FIGURE 2. Approaching a vessel: frame 2.

It was not possible to validate the software with regard to the use of AIS information, as the information on the geographic coordinates of the vessel, course, heading and bearing was not provided. However, a verification of the corresponding module was performed by assuming that this information was known and deriving the geographic coordinates of the two points needed to make an AIS query. The query outputs the correct information, related to the vessels in the rectangle bounded by the two points. For example, assuming that the

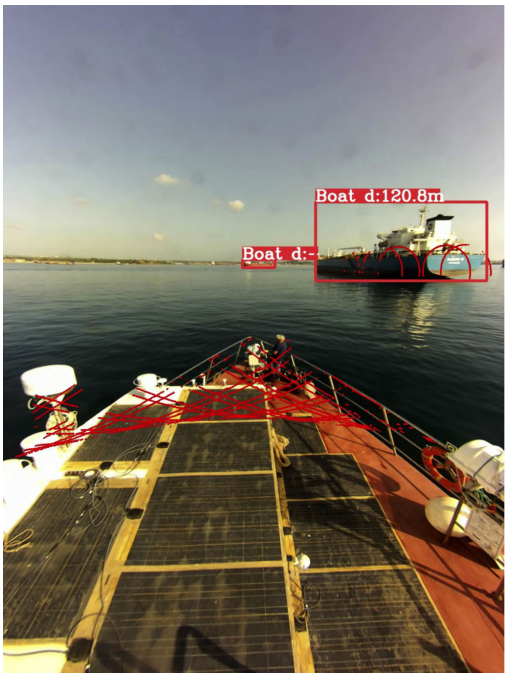


FIGURE 3. Approaching a vessel: frame 3.

vessel is located at the geographic coordinates latitude 37 N, longitude 15 E at 08:52 and 29 seconds on 06/09/2022, a query related to the AIS data with the two points delimiting the geographic area equal to (36 N, 14 E) and (38 N, 16 E), respectively, produces the following output:

Ship ID: 2470036 - Lat: 37.11668 N - Lon: 14.825 E
Ship ID: 2470074 - Lat: 37.56477 N - Lon: 15.10375 E
Ship ID: 209582000 - Lat: 37.20307 N - Lon: 15.20979 E
Ship ID: 215486000 - Lat: 37.22097 N - Lon: 15.20121 E
Ship ID: 229370000 - Lat: 37.22511 N - Lon: 15.19940 E
Ship ID: 240575000 - Lat: 37.11593 N - Lon: 15.267 E
Ship ID: 241604000 - Lat: 37.12299 N - Lon: 15.26652 E
Ship ID: 247055200 - Lat: 37.22149 N - Lon: 15.19253 E

The software, running on a MacBook Pro equipped with an 8-core, 2.3 GHz Intel Core-i9 cpu and 64 GB RAM, was evaluated by processing a dataset consisting of 1536 images, corresponding radar/LIDAR data and AIS log files. Specifically, the measured throughput for the entire pipeline consisting of reading an image from filesystem, processing the image, projecting 3D radar/LIDAR points onto the 2D image, geographic query related to AIS data, and writing the processed image to filesystem, was 6.5 FPS. When run on a compute node of the Marconi cluster, equipped with 2 IBM POWER9 AC922 3.1 GHz 16 cores processors, 256 GB RAM and 4 NVIDIA Volta V100 GPUs, Nvlink 2.0, 16GB, the throughput – using only one GPU and eliminating output writing – was 10 FPS. Since reading the images from the filesystem is computationally expensive, and in the case of the proposed application unnecessary (since the video-related frames are immediately available from the acquiring camera), we estimate a throughput between 10 FPS and 20 FPS, which is enough considering the radar/LIDAR acquisition time.

X. CONCLUSION

Ship sailing is a complex endeavour, requiring carefully considered proactive and reactive strategies in choosing the course of action that best suits the various events to be managed. Humans are already supported by different technologies as technologies for sailing, however these technologies are usually available in isolation. In this paper we have shown how to use simultaneously three different technologies by fusing their information in order to provide enhanced support for ship sailing. To the best of our knowledge no similar approach is reported in the literature from an operational point of view. In particular, we have shown how to fuse the video acquired from a camera with the information available from a radar/Lidar and an AIS receiver. The video frames are analyzed in order to detect automatically surrounding ships and seamarks whilst Lidar is used to determine the average or minimum distance from the ship to the acquired targets and finally the AIS receiver logs are queried to determine, if available, useful information related to the surrounding ships, such as their geographic position, type of ship etc.

Although more research is certainly needed, we believe that the simultaneous use of these technologies is a step towards fully autonomous ship sailing; moreover, multisensor data fusion into one HMI helps operators to quickly understand the surrounding environment and operate properly on the vessel from the very beginning even if they have a limited experience, whilst simultaneously reducing significantly the required space in the wheelhouse. Our findings are encouraging and show the effectiveness of our approach.

In order to evaluate the effectiveness of the proposed approach, future work may include the implementation of a corresponding novel dashboard for on field testing; from this perspective, it will be interesting to collect and analyze questionnaires administered to the relevant operators to infer their precious feedback regarding both the Human Machine Interface and the effectiveness of the information provided by the dashboard to support ship sailing. This is particularly relevant especially for young operators with limited experience in operating properly the vessel.

Additionally, it is certainly important the ability to detect different navigational aids such as lateral marks, cardinal marks, and other IALA (International Association of Marine Aids to sailing and Lighthouse Authorities) defined marks. Also of interest is the possible inclusion of additional technologies as well such as, for instance, Motion Reference Units. Our goal is to complement the art of ship sailing, currently based on a mix of knowledge and experience, with a novel tool which can allow taking better, informed decisions leveraging advanced technologies.

REFERENCES

[1] A. Tiano, A. Zirilli, M. Cuneo, and S. Pagnan, "Multisensor data fusion applied to marine integrated navigation systems," *Proc. Inst. Mech. Eng., M, J. Eng. Maritime Environ.*, vol. 219, no. 3, pp. 121–130, Sep. 2005.

[2] Y.-H. Liu, S.-Z. Wang, and X.-M. Du, "A multi-agent information fusion model for ship collision avoidance," in *Proc. Int. Conf. Mach. Learn. Cybern.*, Jul. 2008, pp. 6–11.

[3] R. G. Wright, "Intelligent autonomous ship navigation using multi-sensor modalities," *TransNav, Int. J. Mar. Navigat. Saf. Sea Transp.*, vol. 13, no. 3, pp. 503–510, 2019.

[4] B. Fu, J. Liu, and Q. Wang, "Multi-sensor integrated navigation system for ships based on adaptive Kalman filter," in *Proc. IEEE Int. Conf. Mechatronics Autom. (ICMA)*, Aug. 2019, pp. 186–191.

[5] J.-C. Zheng, Y. Wang, C.-C. Lin, X.-L. Zhang, J. Liu, and L.-W. Ji, "A fusion algorithm of target dynamic information for asynchronous multi-sensors," *Microsyst. Technol.*, vol. 24, no. 10, pp. 3995–4005, Oct. 2018.

[6] S. Thombre, Z. Zhao, H. Ramm-Schmidt, J. M. V. García, T. Malkamäki, S. Nikolskiy, T. Hammarberg, H. Nuortie, M. Z. H. Bhuiyan, S. Särkkä, and V. V. Lehtola, "Sensors and AI techniques for situational awareness in autonomous ships: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 64–83, Jan. 2022.

[7] *Multi-Target Tracking and Detection, Fusing RADAR and AIS Signals Using Poisson Multi-Bernoulli Mixture Tracking, in Support of Autonomous Sailing*, Zenodo, Honolulu, HI, USA, Oct. 2020. [Online]. Available: https://zenodo.org/record/4498560, doi: 10.24868/issn.2515-818X.2020.069.

[8] R. Pelich, M. Chini, R. Hostache, P. Matgen, C. Lopez-Martinez, M. Nuevo, P. Ries, and G. Eiden, "Large-scale automatic vessel monitoring based on dual-polarization Sentinel-1 and AIS data," *Remote Sens.*, vol. 11, no. 9, p. 1078, May 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/9/1078

[9] M. D. Graziano, A. Renga, and A. Moccia, "Integration of automatic identification system (AIS) data and single-channel synthetic aperture radar (SAR) images by SAR-based ship velocity estimation for maritime situational awareness," *Remote Sens.*, vol. 11, no. 19, p. 2196, Sep. 2019. [Online]. Available: https://www.mdpi.com/2072-4292/11/19/2196

[10] Z. Zhao, K. Ji, X. Xing, H. Zou, and S. Zhou, "Ship surveillance by integration of space-borne SAR and AIS—Review of current research," *J. Navigat.*, vol. 67, no. 1, pp. 177–189, Jan. 2014.

[11] F. M. Vieira, F. Vincent, J.-Y. Tourneret, D. Bonacci, M. Spigai, M. Ansart, and J. Richard, "Ship detection using SAR and AIS raw data for maritime surveillance," in *Proc. 24th Eur. Signal Process. Conf. (EUSIPCO)*, Aug. 2016, pp. 2081–2085. [Online]. Available: https://hal.science/hal-01419452

[12] R. Ma, Y. Yin, and K. Bao, "Ship detection based on LiDAR and visual information fusion," in *Proc. Conf. Lasers Electro-Opt. (CLEO)*, May 2022, pp. 1–2.

[13] Y. Zhang, M. Xing, J. Zhang, G.-C. Sun, and D. Xu, "Robust multi-ship tracker in SAR imagery by fusing feature matching and modified KCF," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[14] L. Zhang, Y. Liu, W. Zhao, X. Wang, G. Li, and Y. He, "Frequency-adaptive learning for SAR ship detection in clutter scenes," *IEEE Trans. Geosci. Remote Sens.*, early access, Feb. 28, 2023, doi: 10.1109/TGRS.2023.3249349.

[15] W. Zhang, R. Zhang, G. Wang, W. Li, X. Liu, Y. Yang, and D. Hu, "Physics guided remote sensing image synthesis network for ship detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4700814.

[16] T. Zhang, S. Quan, W. Wang, W. Guo, Z. Zhang, and W. Yu, "Information reconstruction-based polarimetric covariance matrix for PolSAR ship detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5202815.

[17] L. Zhang, J. Cheng, J. Liu, T. Liu, D. Xiang, and Y. Su, "Unsupervised ship detection in SAR images using superpixels and CSPNet," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[18] B. Guo, R. Zhang, H. Guo, W. Yang, H. Yu, P. Zhang, and T. Zou, "Fine-grained ship detection in high-resolution satellite images with shape-aware feature learning," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1914–1926, 2023.

[19] W. Zhao, M. Syafrudin, and N. L. Fitriyani, "CRAS-YOLO: A novel multi-category vessel detection and classification model based on YOLOv5s algorithm," *IEEE Access*, vol. 11, pp. 11463–11478, 2023.

[20] Y. Zhuang, Y. Liu, T. Zhang, and H. Chen, "Contour modeling arbitrary-oriented ship detection from very high-resolution optical remote sensing imagery," *IEEE Geosci. Remote Sens. Lett.*, vol. 20, pp. 1–5, 2023.

[21] Z. Wang, R. Wang, J. Ai, H. Zou, and J. Li, "Global and local context-aware ship detector for high-resolution SAR images," *IEEE Trans. Aerosp. Electron. Syst.*, early access, Jan. 16, 2023, doi: 10.1109/TAES.2023.3237520.

[22] Y. Du, L. Du, Y. Guo, and Y. Shi, "Semisupervised SAR ship detection network via scene characteristic learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5201517.

[23] L. Bai, C. Yao, Z. Ye, D. Xue, X. Lin, and M. Hui, "Feature enhancement pyramid and shallow feature reconstruction network for SAR ship detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 1042–1056, 2023.

[24] W. Mo and J. Pei, "Nighttime infrared ship target detection based on two-channel image separation combined with saliency mapping of local grayscale dynamic range," *Infr. Phys. Technol.*, vol. 127, Dec. 2022, Art. no. 104416.

[25] X. Lou, Y. Liu, Z. Xiong, and H. Wang, "Generative knowledge transfer for ship detection in SAR images," *Comput. Electr. Eng.*, vol. 101, Jul. 2022, Art. no. 108041.

[26] H. Madjidi and T. Laroussi, "Approximate MLE based automatic bilateral censoring CFAR ship detection for complex scenes of log-normal sea clutter in SAR imagery," *Digit. Signal Process.*, vol. 136, May 2023, Art. no. 103972. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1051200423000672

[27] H. Mahgoun, N. E. Chaffa, M. Ouarzeddine, and B. Souissi, "The combination of singular values decomposition with constant false alarm algorithms to enhance ship detection in a polarimetric SAR application," *Remote Sens. Appl., Soc. Environ.*, vol. 27, Aug. 2022, Art. no. 100815. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352938522001239

[28] L. Li, G. Liu, Z. Li, Z. Ding, and T. Qin, "Infrared ship detection based on time fluctuation feature and space structure feature in sun-glint scene," *Infr. Phys. Technol.*, vol. 115, Jun. 2021, Art. no. 103693. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1350449521000657

[29] X. Wang, D. Zhu, G. Li, X.-P. Zhang, and Y. He, "Proposal-copula-based fusion of spaceborne and airborne SAR images for ship target detection," *Inf. Fusion*, vol. 77, pp. 247–260, Jan. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253521001585

[30] M. Zhang, B. Qiao, M. Xin, and B. Zhang, "Phase spectrum based automatic ship detection in synthetic aperture radar images," *J. Ocean Eng. Sci.*, vol. 6, no. 2, pp. 185–195, Jun. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S246801332030070X

[31] S. Zhao, Y. Luo, T. Zhang, W. Guo, and Z. Zhang, "A domain specific knowledge extraction transformer method for multisource satellite-borne SAR images ship detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 198, pp. 16–29, Apr. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0924271623000515

[32] B. Smith, S. Chester, and Y. Coady, "Ship detection in satellite optical imagery," in *Proc. 3rd Artif. Intell. Cloud Comput. Conf.*, Dec. 2020, pp. 11–18, doi: 10.1145/3442536.3442539.

[33] Y. Mao, X. Li, Z. Li, M. Li, and S. Chen, "Network slimming method for SAR ship detection based on knowlegde distillation," in *Proc. Int. Conf. Aviation Saf. Inf. Technol.*, Oct. 2020, pp. 177–181, doi: 10.1145/3434581.3434613.

[34] L. Han, D. Ran, W. Ye, and X. Wu, "Asymmetric convolution-based neural network for SAR ship detection from scratch," in *Proc. 9th Int. Conf. Comput. Pattern Recognit.*, Oct. 2020, pp. 90–95, doi: 10.1145/3436369.3436464.

[35] L. Han, X. Zhao, W. Ye, and D. Ran, "Asymmetric and square convolutional neural network for SAR ship detection from scratch," in *Proc. 5th Int. Conf. Biomed. Signal Image Process.*, Aug. 2020, pp. 80–85, doi: 10.1145/3417519.3417550.

[36] J. Huang, Z. Chen, H. Xu, and X. Zhang, "Fast ship detection in remote sensing images based on multi-attention mechanism," in *Proc. 5th Int. Conf. Algorithms, Comput. Syst.*, Sep. 2021, pp. 105–112, doi: 10.1145/3490700.3490718.

[37] S.-Q. Chen, R.-H. Zhan, and J. Zhang, "Robust single stage detector based on two-stage regression for SAR ship detection," in *Proc. 2nd Int. Conf. Innov. Artif. Intell.*, Mar. 2018, pp. 169–174, doi: 10.1145/3194206.3194223.

[38] F. Min and P. Liu, "Research on ship detection in the SAR image algorithm based on improved SSD," in *Proc. 4th Int. Conf. Artif. Intell. Pattern Recognit.*, Sep. 2021, pp. 205–211, doi: 10.1145/3488933.3489032.

[39] L. He, S. Yi, X. Mu, and L. Zhang, "Ship detection method based on Gabor filter and fast RCNN model in satellite images of sea," in *Proc. 3rd Int. Conf. Comput. Sci. Appl. Eng.*, Oct. 2019, pp. 1–7, doi: 10.1145/3331453.3361325.

[40] L. Zhang, Y. Liu, Q. Guo, H. Yin, Y. Li, and P. Du, "Ship detection in large-scale SAR images based on dense spatial attention and multi-level feature fusion," in *Proc. ACM Turing Award Celebration Conf. China (ACM TURC)*, Jul. 2021, pp. 77–81, doi: 10.1145/3472634.3472654.

[41] J. Koo, J. Seo, S. Jeon, J. Choe, and T. Jeon, "RBox-CNN: Rotated bounding box based CNN for ship detection in remote sensing image," in *Proc. 26th ACM SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst.*, Nov. 2018, pp. 420–423, doi: 10.1145/3274895.3274915.

[42] Y. Xu, W. Xiong, and J. Liu, "A new ship target detection algorithm based on SVM in high resolution SAR images," in *Proc. Int. Conf. Adv. Image Process.*, Aug. 2017, pp. 6–13, doi: 10.1145/3133264.3133273.

[43] Y. Mao, X. Li, Z. Li, M. Li, and S. Chen, "An anchor-free SAR ship detector with only 1.17M parameters," in *Proc. Int. Conf. Aviation Saf. Inf. Technol.*, Oct. 2020, pp. 182–186, doi: 10.1145/3434581.3434614.

[44] W.-J. Lee, M.-I. Roh, H.-W. Lee, J. Ha, Y.-M. Cho, S.-J. Lee, and N.-S. Son, "Detection and tracking for the awareness of surroundings of a ship based on deep learning," *J. Comput. Des. Eng.*, vol. 8, no. 5, pp. 1407–1430, Sep. 2021, doi: 10.1093/jcde/qwab053.

[45] B. Wang, B. Han, and L. Yang, "Accurate real-time ship target detection using YOLOv4," in *Proc. 6th Int. Conf. Transp. Inf. Saf. (ICTIS)*, Oct. 2021, pp. 222–227.

[46] Q. Jiang and H. Li, "Silicon energy bulk material cargo ship detection and tracking method combining YOLOv5 and DeepSort," *Energy Rep.*, vol. 9, pp. 151–158, Apr. 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352484723001191

[47] J. Escorcia-Gutierrez, M. Gamarra, K. Beleño, C. Soto, and R. F. Mansour, "Intelligent deep learning-enabled autonomous small ship detection and classification model," *Comput. Electr. Eng.*, vol. 100, May 2022, Art. no. 107871. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0045790622001616

[48] Z. Chen, D. Chen, Y. Zhang, X. Cheng, M. Zhang, and C. Wu, "Deep learning for autonomous ship-oriented small ship detection," *Saf. Sci.*, vol. 130, Oct. 2020, Art. no. 104812. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925753520302095

[49] R. W. Liu, W. Yuan, X. Chen, and Y. Lu, "An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system," *Ocean Eng.*, vol. 235, Sep. 2021, Art. no. 109435. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0029801821008404

[50] F. Wang, X. Yang, Y. Zhang, and J. Yuan, "Ship target detection algorithm based on improved YOLOv3," in *Proc. 3rd Int. Conf. Big Data Technol.*, Sep. 2020, pp. 162–166, doi: 10.1145/3422713.3422721.

[51] Z. Qin, L. Han, B. Shi, X. Zhang, and Y. Xu, "Improved detection and recognition of sea surface ships based on YOLOv3," in *Proc. 4th Int. Conf. Electron., Commun. Control Eng.*, Apr. 2021, pp. 40–47, doi: 10.1145/3462676.3462683.

[52] J. Li, J. Chen, P. Cheng, Z. Yu, L. Yu, and C. Chi, "A survey on deep-learning-based real-time SAR ship detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 16, pp. 3218–3247, 2023.

[53] B. Li, X. Xie, X. Wei, and W. Tang, "Ship detection and classification from optical remote sensing images: A survey," *Chin. J. Aeronaut.*, vol. 34, no. 3, pp. 145–163, Mar. 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1000936120304544

[54] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 8, pp. 1993–2016, Aug. 2017.

[55] X. Wu, D. Sahoo, and S. C. H. Hoi, "Recent advances in deep learning for object detection," *Neurocomputing*, vol. 396, pp. 39–64, Jul. 2020. https://www.sciencedirect.com/science/article/pii/S0925231220301430

[56] L. Liu, W. Ouyang, X. Wang, P. Fieguth, J. Chen, X. Liu, and M. Pietikäinen, "Deep learning for generic object detection: A survey," *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 261–318, Feb. 2020.

[57] J. Zhang, W. Li, P. Ogunbona, and D. Xu, "Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective," *ACM Comput. Surv.*, vol. 52, no. 1, pp. 1–38, Feb. 2019, doi: 10.1145/3291124.

[58] W. Hu, T. Tan, L. Wang, and S. Maybank, "A survey on visual surveillance of object motion and behaviors," *IEEE Trans. Syst., Man Cybern. C, Appl. Rev.*, vol. 34, no. 3, pp. 334–352, Aug. 2004, doi: 10.1109/TSMCC.2004.829274.

[59] P. Kaur, A. Aziz, D. Jain, H. Patel, J. Hirokawa, L. Townsend, C. Reimers, and F. Hua, "Sea situational awareness (SeaSAw) dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 2578–2586.

[60] B. Iancu, V. Soloviev, L. Zelioli, and J. Lilius, "ABOships—An inshore and offshore maritime vessel detection dataset with precise annotations," *Remote Sens.*, vol. 13, no. 5, p. 988, Mar. 2021. [Online]. Available: https://www.mdpi.com/2072-4292/13/5/988

[61] B. Iancu, V. Soloviev, L. Zelioli, and J. Lilius, "Aboships," Faculty Sci. Eng., Åbo Akademi Univ., Åbo, Finland, Tech. Rep. Version v1.0, May 2021, doi: 10.5281/zenodo.4736931.

[62] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek. (2017). *Singapore Maritime Dataset*. [Online]. Available: https://sites.google.com/site/dilipprasad/home/singapore-maritime-dataset?authuser=0

[63] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "SeaShips: A large-scale precisely annotated dataset for ship detection," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2593–2604, Oct. 2018.

[64] M. H. Zwemer, R. G. J. Wijnhoven, and P. H. N. de With, "Ship detection in harbour surveillance based on large-scale data and CNNs," in *Proc. 13th Int. Joint Conf. Comput. Vis., Imag. Comput. Graph. Theory Appl.*, 2018, pp. 153–160.

[65] Y. Zheng and S. Zhang, "Mcships: A large-scale ship dataset for detection and fine-grained categorization in the wild," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2020, pp. 1–6.

[66] E. Gundogdu, B. Solmaz, V. Yücesoy, and A. Koç, "Marvel: A large-scale image dataset for maritime vessels," in *Computer Vision ACCV 2016*, S.-H. Lai, V. Lepetit, K. Nishino, and Y. Sato, Eds. Cham, Switzerland: Springer, 2017, pp. 165–180.

[67] E. Gundogdu, B. Solmaz, V. Yücesoy, and A. Koç. (2017). *Marvel Dataset*. [Online]. Available:https://github.com/avaapm/marveldataset2016

[68] Z. Shao, J. Wang, L. Deng, X. Huang, T. Lu, F. Luo, R. Zhang, X. Lv, C. Dang, Q. Ding, and Z. Wang, "GLSD: The global large-scale ship database and baseline evaluations," 2021, *arXiv:2106.02773*.

[69] M. M. Zhang, J. Choi, K. Daniilidis, M. T. Wolf, and C. Kanan, "VAIS: A dataset for recognizing maritime imagery in the visible and infrared spectrums," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2015, pp. 10–16.

[70] J. Lin, P. Diekmann, C.-E. Framing, R. Zweigel, and D. Abel, "Maritime environment perception based on deep learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15487–15497, Sep. 2022.

[71] *Technical Characteristics for an Automatic Identification System Using Time Division Multiple Access in the VHF Maritime Mobile Frequency Band*, document Recommendation M.1371-5, I. I. T. Union, Feb. 2023. [Online]. Available: https://www.itu.int/rec/R-REC-M.1371-5-201402-I/en

[72] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Computer Vision ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 818–833.

[73] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[74] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, vol. 37, Jul. 2015, pp. 448–456.

[75] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[76] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[77] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[78] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-Resnet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 4278–4284.

[79] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5987–5995.

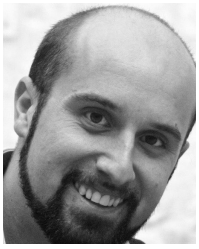
[80] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Region-based convolutional networks for accurate object detection and segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 1, pp. 142–158, Jan. 2016.

[81] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Computer Vision ECCV 2014*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 346–361.

[82] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[83] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. 28th Int. Conf. Neural Inf. Process. Syst. (NIPS)*. Cambridge, MA, USA: MIT Press, vol. 1, 2015, pp. 91–99.
[84] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.* Red Hook, NY, USA: Curran Associates, 2016, pp. 379–387.
[85] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.
[86] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788, doi: 10.1109/CVPR.2016.91.
[87] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Computer Vision ECCV*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham, Switzerland: Springer, 2016, pp. 21–37.
[88] B. Li, W. Wu, Q. Wang, F. Zhang, J. Xing, and J. Yan, "SiamRPN++: Evolution of Siamese visual tracking with very deep networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4277–4286.
[89] W.-C. Hu, C.-Y. Yang, and D.-Y. Huang, "Robust real-time ship detection and tracking for visual surveillance of cage aquaculture," *J. Vis. Commun. Image Represent.*, vol. 22, no. 6, pp. 543–556, Aug. 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1047320311000514
[90] I.-C. Hu and C.-Y. Yang, "An improved full search algorithm with adaptive template block for fast and accurate object tracking," *Int. J. Innov. Comput., Inf. Control (IJICIC)*, vol. 6, no. 11, pp. 5115–5130, 2010.
[91] Y. Jie, L. Leonidas, F. Mumtaz, and M. Ali, "Ship detection and tracking in inland waterways using improved YOLOv3 and deep sort," *Symmetry*, vol. 13, no. 2, p. 308, 2021. [Online]. Available: https://www.mdpi.com/2073-8994/13/2/308
[92] B. Yan, H. Peng, J. Fu, D. Wang, and H. Lu, "Learning spatio-temporal transformer for visual tracking," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10428–10437.
[93] J. Pang, L. Qiu, X. Li, H. Chen, Q. Li, T. Darrell, and F. Yu, "Quasi-dense similarity learning for multiple object tracking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 164–173.
[94] M.-H. Haghbayan, F. Farahnakian, J. Poikonen, M. Laurinen, P. Nevalainen, J. Plosila, and J. Heikkonen, "An efficient multi-sensor fusion approach for object detection in maritime environments," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2163–2170.

MASSIMO CAFARO (Senior Member, IEEE) received the Laurea (M.Sc.) degree in computer science from the University of Salerno and the Ph.D. degree in computer science from the University of Bari. He is an Associate Professor with the Department of Engineering for Innovation, University of Salento. His research covers parallel and distributed computing, cloud and grid computing, data mining, machine learning, deep learning, big data, security, and cryptography. He is the Director of the Master in Applied Data Science and the Head of the HPC Laboratory, University of Salento. He is the author of more than 110 refereed articles and holds a patent on distributed database technologies. He focuses his research on high-performance and distributed computing on both theoretical and practical aspects, in particular the design and analysis of sequential, parallel, and distributed algorithms. He is a Senior Member of the IEEE Computer Society and the ACM and the Vice Chair of the Regional Centers and Coordinator of the Technical Area on Data Intensive Computing for the IEEE Technical Committee on Scalable Computing. He serves as an Associate Editor for IEEE Access and *Future Internet* (MDPI), and as a moderator for the IEEE TechRxiv preprint repository ("Computing and Processing" category).



ITALO EPICOCO received the Ph.D. degree in "innovative materials and technologies" from ISUFI, University of Lecce, Italy, in 2003. He is an Assistant Professor with the University of Salento, Lecce, and the Director of the Advanced Scientific Computing (ASC) Division at the Euro-Mediterranean Center on Climate Change Foundation (CMCC). He is the Director of the Master in Scientific Programming and the Co-Leader of the HPC Laboratory at the University of Salento. His main skills concern computer engineering and computer science. His research interests include the design of data mining algorithms on high-end computing architectures, high performance, and distributed computing. He is currently working on the optimization of numerical kernels for solving PDEs. The current application field is the Earth system models with a particular focus on ocean models. He has published more than 80 articles in refereed books, journals, and conference proceedings on parallel and grid computing. His research activity also includes the benchmarking and evaluation of new emerging computational technologies based on GP-GPU, hybrid architectures, and FPGA-based computing.

MARCO PULIMENO received the Ph.D. degree in mathematics and computer science from the University of Salento, Italy. He is an Assistant Professor with the University of Salento. He published on the topic of frequent items and quantiles in several refereed journals and conference proceedings. His research interests include high-performance computing, distributed computing, and, in particular, parallel data mining.

EMANUELE SANSEBASTIANO is the Senior Engineering Project Manager of Fincantieri NexTech.

Open Access funding provided by 'Università del Salento' within the CRUI CARE Agreement