remote sensing

MDPI

Article

ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations

Bogdan Iancu *,†, Valentin Soloviev †, Luca Zelioli † and Johan Lilius †

Faculty of Science and Engineering, Åbo Akademi University, 20500 Åbo, Finland; valentin.soloviev@abo.fi (V.S.); luca.zelioli@abo.fi (L.Z.); johan.lilius@abo.fi (J.L.)

* Correspondence: bogdan.iancu@abo.fi

† Current address: Åbo Akademi, Agora, Informationsteknologi, Vattenborgsvägen 3, 20500 Åbo, Finland.

Abstract: Availability of domain-specific datasets is an essential problem in object detection. Datasets of inshore and offshore maritime vessels are no exception, with a limited number of studies addressing maritime vessel detection on such datasets. For that reason, we collected a dataset consisting of images of maritime vessels taking into account different factors: background variation, atmospheric conditions, illumination, visible proportion, occlusion and scale variation. Vessel instances (including nine types of vessels), seamarks and miscellaneous floaters were precisely annotated: we employed a first round of labelling and we subsequently used the CSRT tracker to trace inconsistencies and relabel inadequate label instances. Moreover, we evaluated the out-of-the-box performance of four prevalent object detection algorithms (Faster R-CNN, R-FCN, SSD and EfficientDet). The algorithms were previously trained on the Microsoft COCO dataset. We compared their accuracy based on feature extractor and object size. Our experiments showed that Faster R-CNN with Inception-Resnet v2 outperforms the other algorithms, except in the large object category where EfficientDet surpasses the latter.

Keywords: maritime vessel dataset; ship detection; object detection; convolutional neural network; deep learning; autonomous marine navigation

check for updates

Citation: Iancu, B.; Soloviev, V.; Zelioli, L.; Lilius, J. ABOships—An Inshore and Offshore Maritime Vessel Detection Dataset with Precise Annotations. *Remote Sens.* **2021**, *13*, 988. https://doi.org/10.3390/rs13050988

Academic Editors: Pedro Melo-Pinto

Received: 4 February 2021
Accepted: 1 March 2021
Published: 5 March 2021

1. Introduction

Maritime vessel detection from waterborne images is a crucial aspect in various fields involving maritime traffic supervision and management, marine surveillance and navigation safety. Prevailing ship detection techniques exploit either remote sensing images or radar images, which can hinder the performance of real-time applications [1]. Satellites can provide near real-time information, but satellite image acquisition, however, can be unpredictable, since it is challenging to determine which satellite sensors can provide the relevant imagery in a narrow collection window [2]. Hence, seaborne visual imagery can tremendously help in essential tasks both in civilian and military applications, since it can be collected in real-time from surveillance videos, for instance.

Ship detection in a traditional setting depends extensively on human monitoring, which is highly expensive and unproductive. Moreover, the complexity of the maritime environment makes it difficult for humans to focus on video footage for prolonged periods of time [3]. Machine vision, however, can take the strain from human resources and provide solutions for ship detection. Traditional methods based on feature extraction and image classification, involving background subtraction and foreground detection, as well as directional gradient histograms, are highly affected by datasets exhibiting challenging environmental factors (glare, fog, clouds, high waves, rain etc.), background noise or lighting conditions.

Convolutional neural networks (CNNs) contributed massively to the image classification and object detection tasks in the past years [4–8]. They incorporate feature extractors and classifiers in multilayer architectures, whose number of layers regulate their selectiveness and feature invariance. CNNs exploit convolutional and pooling layers extracting local features, and gradually advancing object representation from simple features to complex structures, across multiple layers. CNN-based detectors can subtract compelling distinguishable features automatically unlike more traditional methods which use predefined features, manually selected. However, integrating ship features into detection proves to be challenging even in this context, given the complexity of environmental factors, object occlusion, ship size variation, occupied pixel area etc. This often leads to unsatisfactory performance of detectors on ship datasets.

To address ship detection in a range of operating scenarios, including various atmospheric conditions, background variations and illumination, we introduce a new dataset consisting of 9880 images, and annotations comprising 41, 967 carefully annotated objects.

The paper is organized as follows. Section 2 describes related work, including notable results in vessel detection and maritime datasets comprising waterborne images. Section 3 describes data acquisition, dataset diversity, dataset design and our relabelling algorithm along with basic dataset statistics based on the final annotation data. In Section 4, we discuss evaluation criteria and present experimental results; we investigate four CNN-based detectors and discuss the feature extractors and object size effect on the performance of the detectors. Section 5 provides a qualitative overview of the experimental results. In Section 6, we provide a brief analysis of our dataset specifications in comparison with other similar datasets. Conclusions are presented in Section 7.

2. Related Work

2.1. Object Detection

Object detection is one of the fundamental visual recognition problems where the requirement is to predict whether there are any objects from given categories in an image and provide their location (bounding boxes or pixel-level localization in case of instance segmentation), if any are found. Generally, this is achieved by extracting features in an image and matching them against features from trained images. Traditional approaches use sliding windows to generate proposals, then visual descriptors to generate an embedding, which are subsequently classified (such as SVM, bagging, cascade learning and AdaBoost). Traditional algorithms with best performance focus on carefully designing the descriptors for extracting the features (SIFT, Haar, SURF). However, since 2008, more and more limitations of this approach became evident [7]. We list below the most notable ones:

- Hand-annotated visual descriptors provided large number of proposals, which caused high rates of false positives.
- Visual descriptors (as mentioned above) extract low-level features, but are unsuitable for high-level features.
- Each step of a detection pipeline is optimized separately, so global optimization is difficult to attain.

In the early 2010s, deep learning approaches came to prominence and started replacing the traditional ones. Object detection networks can be roughly categorized into 2 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 the proposals, while the former considers positions within an image as potential objects and attempts to classify them immediately. The traditional approach of sliding windows for proposal generation is still used in CNNs, but other notable advances emerged, which allow for more efficient proposal generation, such as anchor-based and key-point approaches (CenterNet being one of the more notable examples of the kind) [7].

However, the key difference between traditional object detection and CNNs stems from the manner in which visual descriptors are generated. In deep learning, instead of creating visual descriptors by hand, convolutional layers perform this role. Instead of defining feature extractors by hand, basic CNNs train multiple convolutional layers to extract both high- and low-level features, which are then classified with the help of fully-connected layers. The resulting network essentially solves all the main limitations of a traditional approach, but the trade-off is that it requires a significantly larger number of training images for hyperparameter optimization [7,8].

While the requirement of a large number of training samples can prove to be a large obstacle, one of the benefits of CNN-based models is that they can be generalized into other fields with similar characteristics with the help of transfer learning. By training a model on a specific dataset, the backbone of the model can be later used to extract features in other tasks with similar features. For this reason, the aim of recent CNN-models was to be as generic as possible, since with the help of transfer learning, they can be specialized for the field of interest. The challenge, however, appears when those generic models are not suitable feature extractors for a new field and there is not enough data to train them [6]. For those specific cases, the only viable solution is creation of new datasets.

2.2. General Object Detection Datasets

The two main reasons for the remarkable progress computer vision made in the past decades are the availability of large-scale datasets and powerful GPUs that made it possible for deep learning to take off considerably [9]. Deep learning made notable contributions to the field of computer vision, the tasks of image classification and object detection being in the forefront of research areas that benefited from it. International competitions such as ILSVRC, PASCAL VOC, and Microsoft COCO motivated the community tremendously, each of their contributions offering large-scale datasets that have been exploited ever since. These general object detection datasets have been extensively used for object detection with deep neural networks. They are essential for testing and training computer vision algorithms. We will discuss below some of the most prominent general-purpose object detection datasets.

Microsoft COCO [10] provides a selection of 330,000 images with a number of 2.5 million of labelled object instances, over 91 object classes. The dataset labeling used per-instance segmentation to ensure precise object localization. Two crucial aspects of the dataset are that it exhibits abundant contextual information and images contain multiple objects per image.

The ImageNet Large Scale Visual Recognition Challenge (ILSVRC) ran annually for a number of years and was established as one of the typical benchmarks for object classification and detection. The Imagenet dataset [4], the foundation of the challenge, is an image collection based on the WordNet hierarchy [11], which provides on average 1000 manually verified images for every synset (synonym set) in the hierarchy. These images are subjected to quality-control and are human-annotated. The dataset consists of over 14 million images, of which over 14 million were annotated to denote what objects are present in the image and, for over a million of them, bounding boxes are provided too.

Pattern Analysis, Statistical Modelling and Computational Learning (PASCAL) Visual Object Classes (VOC) is a prominent project in the computer vision community, which provided publicly available image datasets including ground truth annotations and standardized evaluation metrics. These datasets were exploited as part of a number of challenges on various tasks such as: classification, detection, segmentation, etc. The greater number of scientific publications regarding object detection use the PASCAL VOC challenges to benchmark their proposed algorithms. The reason is that these challenges introduced a number of evaluation methods: bootstrapping, to decide significant differences among algorithms, a normalised average precision across classes, etc. The dataset released by last PASCAL VOC challenge includes 11,530 images with 27,450 annotated objects over 20 classes. Table 1 shows a variety of object detection datasets, with their total number of images and clasess. We can notice that ImageNet is by far the largest of the ones mentioned in the table, encompassing the greater number of total images and classes.

Table 1. Different object detection datasets comprising various object classes, with their corresponding annotations.

General Object Detection Dataset			
Dataset	**Total Images**	**Total Classes**	**Annotations**
ImageNet	14,197,122	1000	1,034,908
COCO	330,000	91	2,500,000
OpenImage (V6)	9,000,000	600	16,000,000
PASCAL VOC (2012)	11,530	20	27,450

Of the general-purpose object detection datasets, in Table 1, the total number of maritime vessels included is limited, only Microsoft COCO comprising a considerable amount of vessels, 3146. All vessel counts can be found in Table 2.

Table 2. Maritime vessel instances in general object detection datasets.

Maritime Vessel Instances	
Dataset	**Vessel Count**
ImageNet	1071
COCO	3146
OpenImage	1000
PASCAL VOC	353

2.3. Maritime Vessel Detection Datasets

Maritime vessel detection from satellite imagery was employed in many studies, over the past 40 years, a review from 2018, [12], gathering a number of 119 papers regarding ship detection and classification only from optical satellites. At the same time, the studies regarding maritime vessel detection from waterborne images are still quite scarce to this day. Some studies proposed algorithms utilizing the idea of background subtraction and detection of the foreground in maritime images. This class of techniques is predominantly used in surveillance applications due to their ability to perform well with unexpected changes in illumination, frequency or background noise [13]. Other studies proposed solutions for ship detection based on the Histogram of Oriented Gradients (HOG) and sliding windows [14].

However, since the bloom of deep learning in the past 15 years, CNNs were employed in ship detection from waterborne images. Even so, datasets of seaborne images are scarce, the most notable ones we briefly discuss below.

The Singapore Maritime Dataset, introduced in [15] consists of 80 videos recorded during daytime and nighttime, and provides ground truth labels for every frame of every video, comprising bounding-boxes and object classes for the corresponding bounding-boxes. The annotations for the Singapore Maritime Dataset include 10 object classes, of which 6 ship types. This dataset is used for ship detection employing the YOLO v.2 algorithm [16].

Another recent ship dataset, SeaShips [3], consists of over 31,455 inshore and offshore images of ships, comprising 6 ship types. In [3], they employ three object detectors (Faster R-CNN [17], SSD [18] and YOLO [16]) to detect ships.

One of the most recent datasets published is MCShips [19], comprising a number of 14,709 images of ships, whose annotations cover 6 warship classes and 7 civilian ship classes. In [19], they also use the object detection algorithms above (Faster R-CNN [17], SSD [18] and YOLO [16]) to evaluate the dataset over the 13 ship classes.

We compared our ABOships dataset against other existing ship datasets. Table 3 illustrates the main differences. Our dataset has the smallest number of images (9880) amongst the four datasets, however it contains a great number of annotations (41,967) given the image total, which shows it represents well real scenarios of maritime imagery, taking into account the fact that it includes on average more than 4 annotated objects per image.

Table 3. Comparison of ABOships with other maritime datasets.

Datasets for Ship Detection			
Name	**Total Images**	**Annotations**	**Ship Types Included**
SeaShips	31,455	40,077	6
Singapore	17,450	192,980	6
MCShips	14,709	26,529	13
ABOShips	9880	41,967	9

3. Materials and Methods

3.1. Camera System

The dataset was acquired from a set of 135 videos, collected from a sightseeing watercraft, by a camera with a field of view of 65° and stored in FullHD (1920 × 720) resolution at 15 FPS in MPEG format. The route of the watercraft extended from the city of Turku to Ruissalo in South-West Finland, the videos comprising the urban area along the Aura river, the port and the Finnish Archipelago, for a duration of 13 days (26 June 2018–8 July 2018). The watercraft ran each day in a timeframe between 10.15 and 16.45. The videos were captured into 30-min long periods consisting of footage from the route that the watercraft took. While the route remained largely the same, the data contains a variety of typical maritime scenarios in a range of weather conditions.

In addition to camera video data, the platform had a LiDAR attached to it (SICK LD-MRS, FoV 110 degrees, 2 × 4 planes, up to 300 m detection, at 5 Hz). The data from the LiDAR was captured alongside the video at a rate of 5 entries of up to 800 points per 0.2 s. Given the utilized LiDAR had a detection range of up to 300 m, it was very useful for detecting other objects in the harbor environment. Due to having only 2 times 4 lasers in the height direction however, the provided data was not reliable enough for discerning the nature of the object (i.e., what object was detected). It was useful however to determine distances to the objects perceived in the videos. For the purpose of creating the dataset presented in this paper, we used the LiDAR data to filter out video segments that were captured in the harbor area (usually the ones that had too many points for a prolonged period of time).

To evaluate the models, we acquired 9880 image photos from the videos. First, we annotated all images with 11 categories: seamarks, 9 types of maritime vessels, and miscellaneous floaters. In a second round, we relabelled all the inconsistencies we found, using an algorithm based on the CSRT tracker [20].

3.2. Dataset Diversity

Maritime environments are inherently intricate, hence a range of factors have to be accounted for when desinging a dataset. Dataset design must ensure that the dataset characterizes well vessels in the environment. Of course, data augmentation methods can be considered for reproducing certain environmental conditions, however authentic conditions may be difficult to anticipate.

Background variation. Particular object detection tasks are more prone to be affected by changes in the background of the picture. For instance, facial recognition is less susceptible to background variations, because given the similar shape of most faces, it is easier to fit them into bounding boxes in a congruous manner. However, the shapes of maritime vessels are highly heterogeneous, making them more difficult to separate from the background due to a potentially vast background information in the bounding box. The accuracy of ship detection would be significantly affected if background information were classified as ship features. Figure 1 illustrates the background variation of images in our datasets, including urban landscapes and an open sea environment.

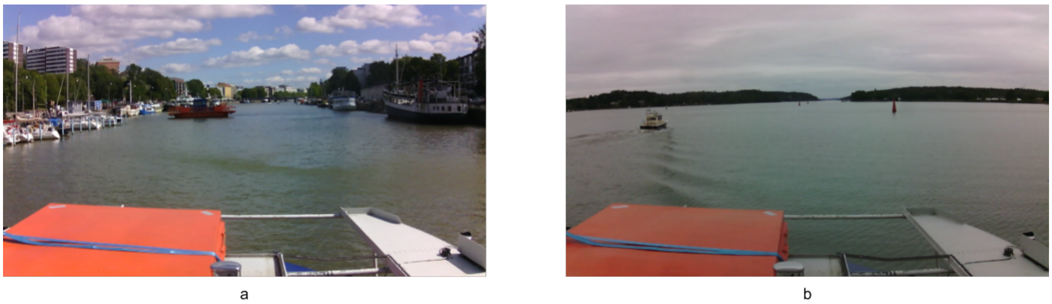

Figure 1. Example image of background variations in the ABOships dataset: (**a**) View of maritime vessels on Aura river including the urban landscape; (**b**) View of a maritime vessel in the Finnish Archipelago.

Atmospheric conditions. Atmospheric conditions were specific to Finnish summers, with very sunny periods, alternating with rainy intervals and cloudy skies. The dataset includes a variety of images of different atmospheric conditions throughout a day.

Illumination. Lighting variations can significantly impact image capture. Illumination throughout the day, in various geographical areas and with specific daylight hours in a given region can dramatically influence image detection.

Visible proportion. A great number of the images in our dataset consists of moving ships, with objects being only partially captured in the camera field of view. However, they still represent objects that were annotated since one has to detect them as well. The annotation should comprise different visible proportions of the maritime vessels.

Occlusion. Due to the fact that our dataset has been captured in an open sea environment, in the harbor area and also comprises urban landscapes, there are many occasions when maritime vessels occlude each other or occlude other objects in the environment in the harbor area or in the urban landscape. In a subset of pictures especially in the harbor area, there is significant occlusion due to a high number of maritime vessels in the proximity of each other. Two examples of occlusion are shown in Figure 2.

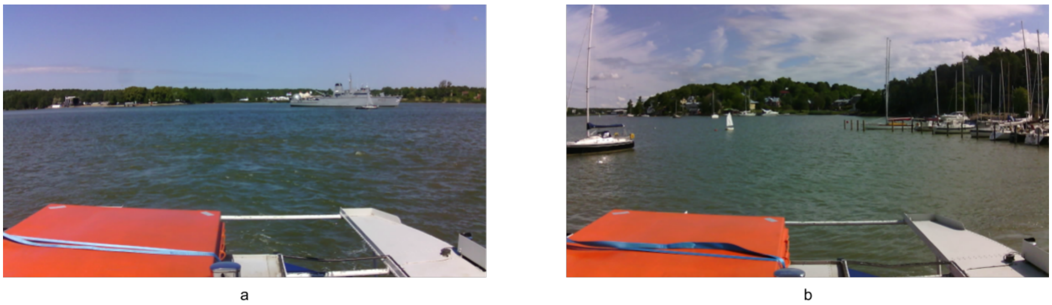

Figure 2. Example image of a occlusion: (**a**) Boat in front of a militaryship; (**b**) Several sailboats occluding each other while docked, on the right half of the image.

Scale variation. Detection of small object can prove to be quite difficult, especially in a complex environment like the sea, ships that occupy a small pixel area in the picture can be confused with other objects in the background. Maintaining a high level of detection for ships requires including several scales for ships sizes in the dataset. For more information regarding the annotation and the size of the bounding boxes, please refer to Section 3.4.

Figure 3 illustrates a sailboat from two different perspectives: a lateral and a frontal view, which shows a variation in both occupied pixel area, but also the visible proportion.

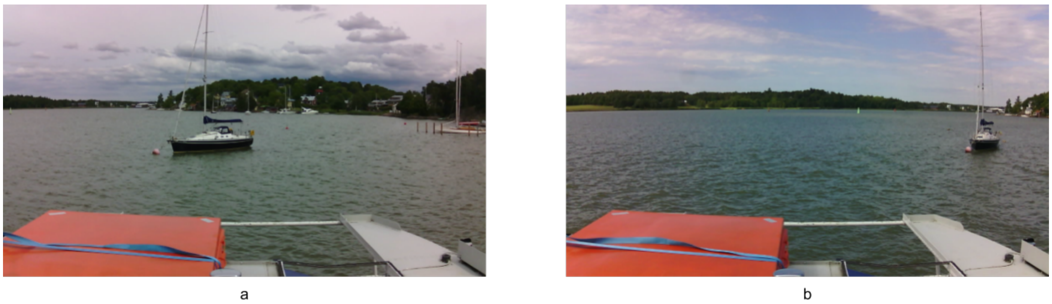

Figure 3. Example image of a sailboat, view from two perspectives: (**a**) Lateral; (**b**) Frontal.

3.3. Dataset Design

The raw data acquired from the camera on the sightseeing watercraft is captured as MPEG videos, with 720 p resolution at 15 FPS . The videos include some footage exhibiting content that is irrelevant for the scope of vessel detection (especially footage captured when the watercraft was docked, either at the start of its route on the Aura river or at the Port of Turku) or sensitive content, such as faces of people. To address the latter issue, we performed face detection on all videos and blurred all detected faces. Addressing the former issue on the other hand, required additional data from the LiDAR.

In a maritime environment, LiDAR data is relatively sparse, authors of this study observed that a high number of points detected for a prolonged duration correlates with the watercraft being docked in the harbor. By setting a point threshold to detect these (docked/harbor) cases, we were able to filter them out in their majority and extract only the images regarding mostly the maritime environment. The images were extracted at an interval of 15 s (one image every 225 frames) and still contained some images captured during docking, but most of them were facing outwards from the harbor, so the images captured in this manner still contain useful maritime data. As a result we acquired 9880 images in the maritime environment.

The acquired images were subsequently separated into workpackages in such a manner that chronologically adjacent pictures were separated into different workpackages. The workpackages were then manually labelled by different annotators. After the initial labelling was completed, we used the CSRT tracker [20] to combine labels of the same object into traces, i.e., a collection of chronologically adjacent images containing a bounding-box for that object. Due to inaccuracies in the tracking process and discrepancies in labelling, the produced traces were not always accurate. After viewing the labels in these traces, we identified the main causes for discrepancies in labelling, which were mainly caused by different interpretations of label annotations. We refined those annotations to eliminate the discrepancies and separated the data into a second collection of workpackages that were provided to annotators, who then relabelled the data, according to refined annotations. After the relabelling was completed, the images and their refined labels were compiled into a dataset of maritime images with refined annotations.

3.4. Annotation

To perform the annotation task, we first investigated the captured videos and identified the vessel types that appeared most often. Due to the fact that the videos were captured at locations with a significant number of passenger ships, there is a certain level of bias for labelers towards those types of ships. This is different from the Seaships database, for instance, which comprises a higher variety of cargo ships. For the purposes of future use in machine vision, rather than using maritime terminology as such (depicting ship scale and purpose), we selected labels that had some clearly distinct visual characteristics. A visual representation of the labels is illustrated in Figure 4. The label categories are discussed below, with more specific details for every category:

- boat—rowing boats or oval-shaped boats (from a lateral perspective), or small-sized boats, visual distinction – rowing-like boats even if they possess engine power;

- cargoship—large-scale ships used for cargo transportation, visual distinction—long ship with cargo containers or designed with container carrying capacity;
- cruiseship—large ship that transports passengers and/or cars on longer distances (assumed at least some hundreds of km);
- ferry—medium-sized ship, used to transport people and cars, a.k.a. waterbus/watertaxi, another appropriate term would be cableferry, visual distinction – it includes entrances on two opposite sides and a cabin in the middle;
- militaryship—an official ship that is either military or Coast Guard and includes a special hull with antennas. For Coast Guard fleets, usually the hulls of their ships read "Coast Guard" and the military ones are dark gray/metallic/black/brown in colour;
- miscboat—miscellaneous maritime vessel, visual distinction – generic boat that does not include any visual distinction mentioned in the other ship categories;
- miscellaneous—identified floaters (birds, other objects floating in the water) or unidentified/unidentifiable floaters;
- motorboat—primarily a speedboat, visual distinction—sleek, aerodynamic features;
- passengership—medium-sized ship, used to transport people on short distances, ex. restaurant boat, visual distinction-usually it has multiple noticeable lateral windows;
- sailboat—sails-propelled boat or a boat which exhibits sails, visual distinction—sails;
- seamark—green/red/blue/black/yellow cone-shaped metal/plastic floater or pipe emerging from the sea.

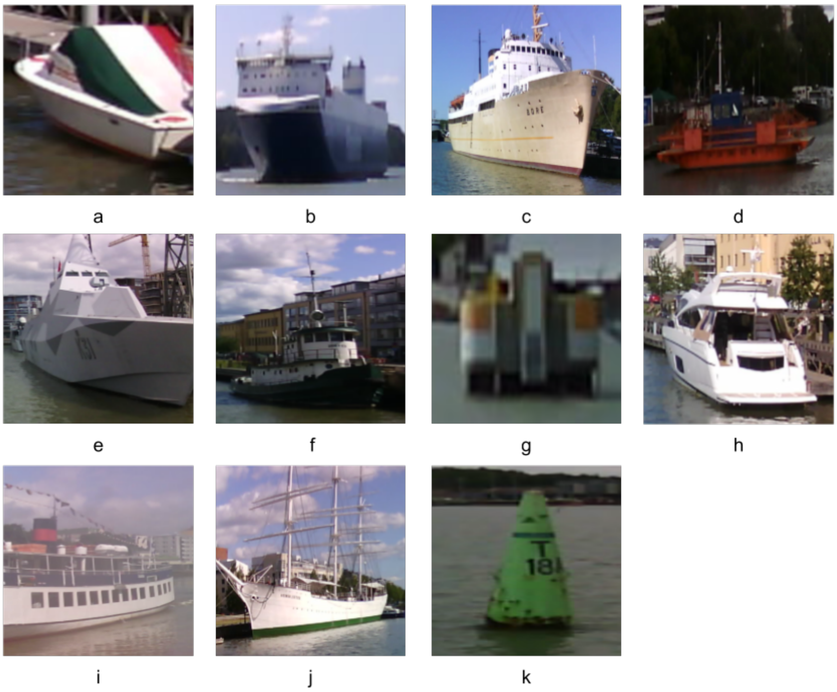

Figure 4. Example images of annotated objects in the ABOships dataset: (**a**) boat, (**b**) cargoship, (**c**) cruiseship, (**d**) ferry, (**e**) militaryship, (**f**) miscboat, (**g**) miscellaneous (floater), (**h**) motorboat, (**i**) passengership, (**j**) sailboat and (**k**) seamark.

3.5. Relabelling Algorithm

The labelling was performed by multiple annotators with different backgrounds, hence some label types were interpreted differently among them. To increase the consistency of labelling, we used the continuous nature of the raw data by tracking the labels between frames using the CSRT tracker [20]. For every labelled frame, a tracker instance was created. The aim was to track an object until the next labelled frame. At that point, the existing traces would be mapped onto the labels of the new frame, based on the IoU metric. During this mapping, it was assumed that labellers would not confuse seamarks with vessels, hence ship labels were not mapped onto seamarks or vice versa. More importantly, previous labels were not taken into consideration, so even if annotators gave the same object conflicting labels in different frames, these labels would still belong to the same trace as long as the tracker could identify them. For cases where the mapping could not be found, the trace would assign a new label, <*Unlabeled*>, to denote that even though nothing was labeled in that specific case, the tracker indicated that the object should belong to the trace.

After a certain number of frames, either the tracker would lose the object (the most common reasons for this being object occlusion, or due to the object being either too far or exiting the frame altogether) or the tracker would have none of the defined labels mapped to it enough times (which would mean it most likely drifted onto another object). In both of those cases, the tracker was stopped and the resulting trace was saved to a file for further processing as described below.

To reduce the number of errors caused by occlusion and the tracker drifting towards other objects than the current object of interest, we performed a second tracking in the backwards direction. By comparing labels identified in the traces acquired from tracking videos in both directions, one could detect situations where traces could not be mapped onto each other. Those cases signify that the tracker was either occluded or drifted to another object, so traces required to be split into smaller sequences still, until no more conflicts could be detected.

The resulting traces (after the backwards tracking) were provided as batches for relabelling. Traces containing a single entry were batched together with other singular traces from the same category. This setup was done with the purpose of preventing and removing accidental labels (mislabeling), while, at the same time, providing more information about the objects being annotated. This allowed us to accurately label even the objects at a longer distance as a consequence of tracking history. Traces obtained in this manner were then provided for relabelling as a collection of labels belonging to the same trace and annotators were asked to refine the labels so that labelling would be consistent with the labelling specifications. Singular entries that did not belong to any trace were subsequently batched together with other objects of the same category. The process described above is illustrated in Figure 5, while the relabelling software application is depicted in Figure 6.

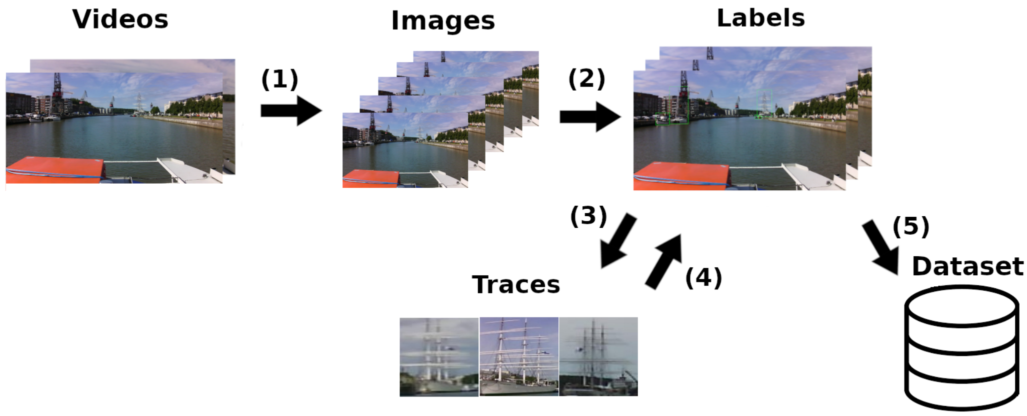

Figure 5. The video collection was separated into 48 workpackages of images (**1**), which were labelled in an initial labelling step (**2**). Using the OpenCV Tracker, the objects were tracked across frames to produce traces (**3**) and then relabelled to fix inconsistencies and fill in the labels that might have been skipped (**4**). The resulting labels were then compiled into the maritime imagery dataset (**5**).

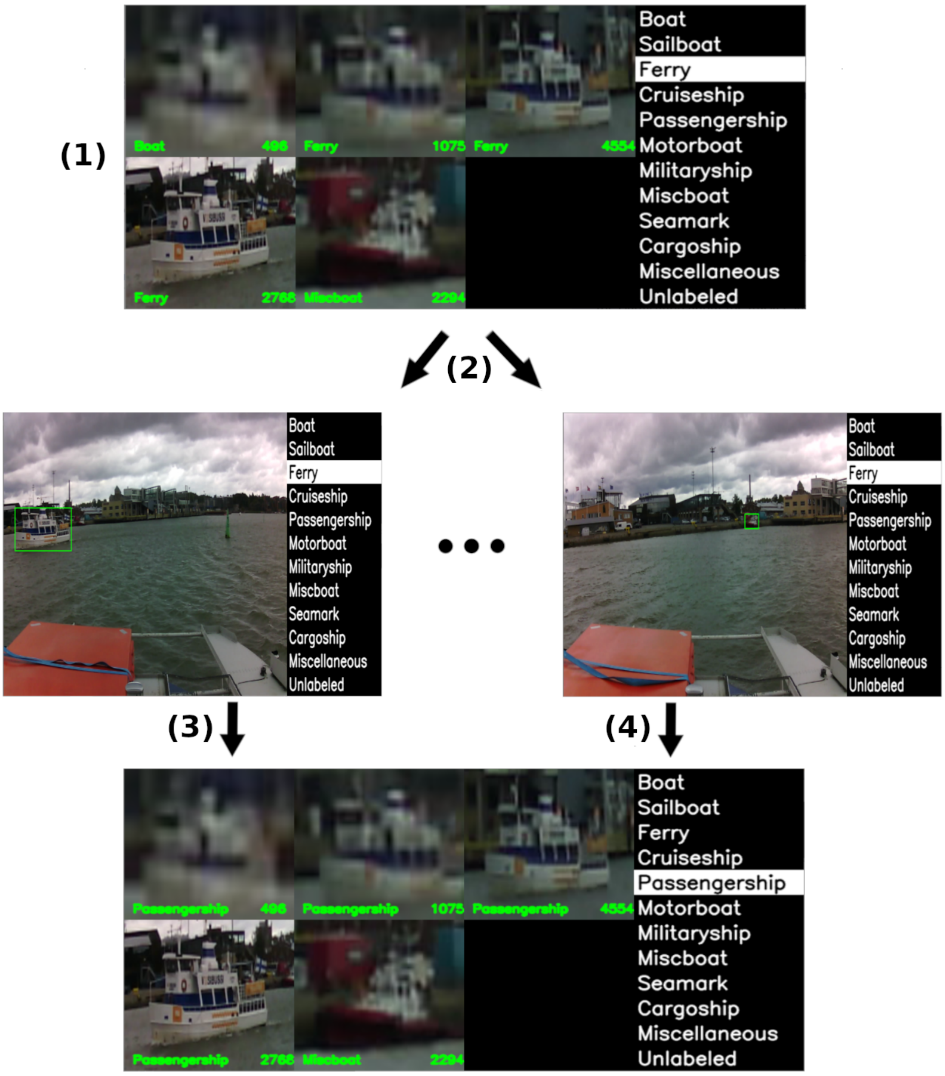

Figure 6. The relabelling process utilized our relabelling software application. Its GUI (graphical user-interface) shows the annotator traces of tracked images between annotation frames (**1**). The annotator is required to either relabel every instance by selecting the correct label from the right panel, or edit an annotation (by selecting a label that emerged distinct from others (**2**)) and change the label of each image individually and possibly fix the bounding box to fit the object more tightly (**3**). Special attention was required in certain situations when the tracker would drift onto other objects, in which case that particular entry of the trace might have had a different label from the rest (**4**). When all labels belonging to a trace were verified and steps (**1**)–(**4**) were completed (**5**), the changes were saved into a new file and the annotator was provided with the next trace.

3.6. Dataset Statistics

Table 4 shows the number of images of each category in our dataset and the number of annotations. The column Images represents the number of images that contain that particular object class and then the percentage of images that comprise that class follows. Then the column Objects represents the number of annotations for that particular class in the dataset, along with the percentage of objects annotated for that specific class out of all the annotated objects in the dataset. One can notice from Table 4 that the highest representation of labels in the images from ABOships dataset is reached by three categories: motorboats (present in 41.11% of the images), sailboats (present in 38.88% of the images), and seamarks (present in 37.89% of the images). Conversely, the lowest representation is registered for cargoships (in 1.58% of the images) and miscellaneous floaters (in 1.30% of the images).

Moreover, Figure 7 illustrates the distribution of annotated objects in our dataset based on occupied pixel area at $\log_2$-scale, for every object category, and separates every object category by size in small, medium and large objects based on the Microsoft COCO variants (small: $\log_2(\text{area}) < 10$, medium: $10 < \log_2(\text{area}) < 13.16$ and large: $\log_2(\text{area}) > 13.16$).

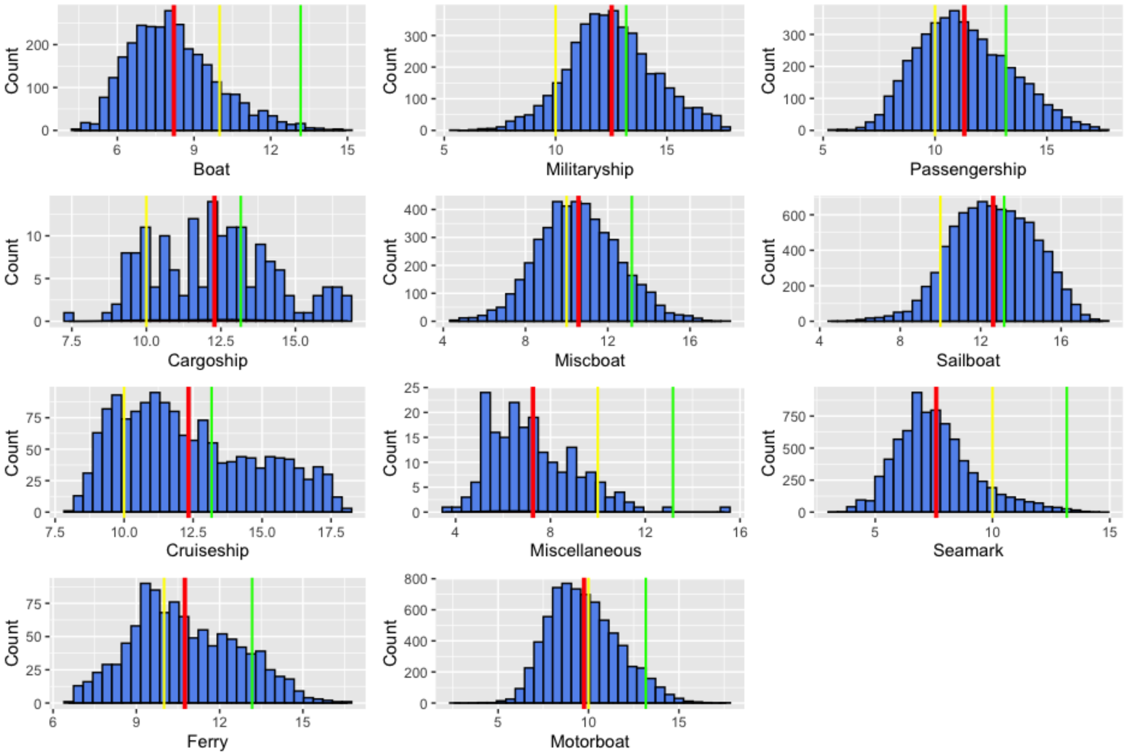

Figure 7. Histograms of occupied pixel area at log_2-scale for all annotated objects by object category, divided into three groups for each category: small, medium and large according to Microsoft COCO variants (small: $\log_2(\text{area}) < 10$, medium: $10 < \log_2(\text{area}) < 13.16$ and large: $\log_2(\text{area}) > 13.16$). The vertical colored lines represent the following values: the red line—represents the mean of the distribution, the yellow line represents the threshold for small objects and the green vertical line delineates the threshold for large objects. In each histogram, respectively, entries to the left of the yellow line represent the small objects group, entries in between the yellow and the green line show the medium-sized objects group and those to the right of the green line depict the large objects group.

Table 4. The table shows the number of images and annotations in the ABOships dataset for every object category, along with their overall percentages.

Number of Images and Annotations for Every Object Category				
Class	**Images**	**Percentage**	**Objects**	**Percentage**
Seamark	3744	37.89%	7670	18.27%
Boat	2034	20.58%	2913	6.94%
Sailboat	3842	38.88%	8147	19.41%
Motorboat	4062	41.11%	7092	16.89%
Passengership	2639	26.71%	4464	10.63%
Cargoship	157	1.58%	161	0.38%
Ferry	945	9.56%	1046	2.49%
Miscboat	2797	28.30%	4642	11.06%
Miscellaneous	129	1.30%	200	0.47%
Militaryship	2559	25.90%	4128	9.83%
Cruiseship	1347	13.63%	1504	3.58%

4. Results

4.1. Evaluation Criteria

To evaluate the performance of different object detection algorithms on specific datasets, one can employ various quantitative indicators. One of the most popular measures in object detection is the IoU (Intersection of Union), which defines the extent of overlap of two bounding boxes as the intersection between the area of the predicted bounding box B_p and the area of the ground truth bounding box B_{gt}, over their union [21]:

$$IoU = \frac{|B_p \cap B_{gt}|}{|B_p \cup B_{gt}|} \tag{1}$$

Given an overlap threshold t, one can estimate whether a predicted bounding box belongs to the background ($IoU < threshold$) or to the given classification system ($IoU > threshold$). With this measure, one can proceed to assess the average precision (AP) by calculating the precision and recall. The precision reflects the capability of a given detector to identify relevant objects and it is calculated as the proportion of detected bounding-boxes, correctly identified, over the total number of detected boxes. The recall reflects the capability of a detector to identify relevant cases and it is calculated as the proportion of correct positive predictions to all ground truth bounding boxes. Based on these two metrics one can draw a precision-recall curve, which encloses an area representing the average precision. However, in a majority of cases, this curve is highly irregular (zigzag pattern) making it challenging to estimate the area under it, i.e., the AP. To address this, one can approach it as an interpolation problem, either as an 11-point interpolation or an all-point interpolation [21].

The 11-point interpolation averages the maximum values of precision over 11 recall levels that are uniformly distributed [21], as depicted below:

$$AP_{11} = \sum_{R \in \{0,0.1,\ldots,0.9,1\}} P_i(R), \tag{2}$$

with

$$P_i(R) = \max_{R^* | R^* \geq R} P_i(R^*). \tag{3}$$

AP_{11} is calculated using the maximum precision $P_i(R)$, with a recall greater than R.

4.2. Baseline Detection

To explore the performance of CNN-based object detectors on our dataset, we focused on prevalent detectors: one-stage (SSD [18] and EfficientDet [22]) and two-stage detectors (Faster R-CNN [17] and R-FCN [23]). The detectors were previously trained on the Microsoft COCO object detection dataset, which comprises a number of 91 object categories. The training dataset contains a number of 3146 images of marine vessels. We investigated the performance of different feature extractors in the aforementioned detectors. We collect maritime vessel detection results based on SSD over different feature extractors (ResNet101, MobileNet v1, MobileNet v2). Moreover, we evaluate the performance of a new state-of-the-art detector, EfficientDet, on our dataset, which used EfficientNet D1 as feature extractor. We also evaluated two-stage detectors: Faster R-CNN and RFCN with different feature extractors. Combining all proposed detectors with the feature extractors, a total of 8 algorithms were investigated. All information regarding the specific configuration of these detectors can be found in [24].

We estimated the performance of these algorithms in detecting maritime vessels, so we excluded seamark and miscellaneous labels from our experiments and focused on detecting vessels. Moreover, we chose images with an occupied pixel area larger than 16^2 pixels. Based on these experiments, we attained Table 5.

Our experiments indicated that the object size impacts the detection accuracy. To corroborate this observation, we divided all vessel labels (with an occupied pixel area larger than 16^2 pixels) in our datasets into three categories, based on Microsoft COCO challenge's variants: small (16^2 < area < 32^2), medium (32^2 < area < 96^2) and large (area > 96^2). Out of the annotated vessels with an occupied pixel area larger than 16^2 pixels in our dataset, 30.25% of the annotated vessels are small, 49.37% are medium and 20.37% are large.

Analyzing the results from our experiments, we observe that detection accuracy decreases with object size. The AP for best-performing detector on the ABOships dataset (Faster R-CNN with Inception ResNet v2 as feature extractor) with a registered AP of 35.18% more than doubles in size from small ($AP_S = 23.16\%$) to large objects ($AP_L = 46.84\%$). The second best detector on the whole dataset (EfficientDet with EfficientNet as feature extractor) however had the best performance on the large-objects category, with an $AP_L = 55.48\%$. In general, detecting small objects turns out to be more difficult than larger objects given that there is less information associated with a smaller occupied pixel area. For medium-sized objects, the best performance is attained by SSD with ResNet101 as feature extractor ($AP_M = 31.18\%$). For small objects, the best-performing detector, Faster R-CNN with Inception ResNet v2, outperforms the other detectors with a registered $AP_S = 23.16\%$. Among the SSD configurations, best performing, in general, was the one having ResNet101 as feature extractor.

Table 5. Average Precision (AP) (in %) of the proposed CNN-based detectors on ABOships dataset, with different feature extractors and object sizes, for all objects with an occupied pixel area > 16^2 pixels.

Detection Performance of Different Detectors on the ABOships Dataset					
Method	**Feature Extractor**	AP_S	AP_M	AP_L	AP
Faster RCNN	Inception ResNet V2	23.16	30.86	46.84	**35.18**
	ResNet50 V1	9.76	20.94	41.65	26.49
	ResNet101	18.42	25.07	38.17	30.26
SSD	ResNet101 V1 FPN	21.39	31.18	42.07	30.03
	MobileNet V1 FPN	12.34	27.61	37.83	28.59
	MobileNet V2	3.01	17.05	27.37	17.48
EfficientDet	EfficientNet D1	10.94	29.68	**55.48**	33.83
RFCN	ResNet101	18.05	26.20	41.61	32.46

5. Qualitative Results

Figure 8 illustrates an example of detection results for the proposed methods, selecting for each the combination of feature extractor that scored the highest AP in each category. We can notice in Figure 8 that Faster R-CNN with a Inception-ResNet-v2 feature extractor (a) and R-FCN with a ResNet101 feature extractor (c) provide detected regions registering high scores ranging from 0.91 to 0.99. The other two detectors in Figure 8, EfficientDet with EfficientNet as feature extractor (b) and SSD with ResNet101 as feature extractor (d), register satisfying results registering with scores ranging from 0.55 to 0.67.

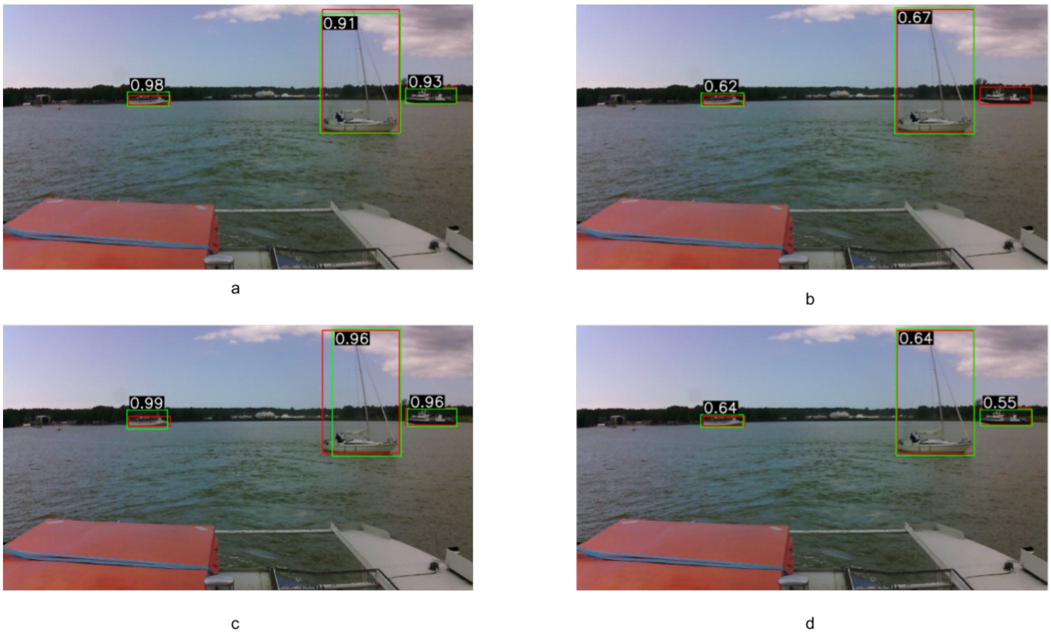

Figure 8. Qualitative detection results for the ABOships dataset on (**a**) Faster R-CNN and Inception-ResNet-v2 as feature extractor, (**b**) EfficientDet with EfficientNet as feature extractor, (**c**) R-FCN with ResNet101 as feature extractor, and (**d**) SSD with ResNet101 as feature extractor. The ground truth bounding-boxes are shown as red rectangles. Predicted boxes by these methods are depicted as green bounding boxes. Each output box is associated with a class label and a score with a value in the interval [0, 1].

6. Discussion

Maritime vessel detection of inshore and offshore images is a topical issue in many areas, such as maritime surveillance and safety, marine and coastal area management, etc. Many of these fields require intricate management of disparate activities, which in practice often necessitate real-time monitoring. This implies, among other aspects, real-time detection of inshore and offshore ships. However, in their majority, ship detection studies and methodology are mostly concerned with either satellite or radar imagery, which can prove to be unreliable in a real-time setting. For this very reason, algorithms, and specifically CNNs, employed on waterborne imagery are especially beneficial either on their own, or in fusion architectures.

Traditional ship detection methods using either background separation or histograms of oriented gradients provide satisfactory results under favorable sea conditions. However, the complexity of the marine environment, including challenging environmental factors (glare, fog, clouds, high waves, rain etc.), renders the extraction of low-level features unreliable. Recent studies involving CNNs address this issue, but deep learning requires domain-specific datasets to produce satisfactory performance. However, public datasets specifically designed for maritime vessel detection are scarce to this day [1]. We discuss this in more detail in Section 2.

Performing exploratory analysis on our dataset, in comparison with other recent maritime object detection datasets (Singapore Maritime Dataset [15], SeaShips [3], MCShips [19]), there are a few aspects that emerge that we discuss as follows. Comparing our dataset to the Singapore Maritime Dataset, one can notice (from Table 3) that ABOships registers a higher number of ship types (9 vs. 6). However, considering the number of annotations per image, the Singapore dataset registers almost 3 times more annotations on average per image (11.05 vs. 4.2). The SeaShips dataset consists of 31,455 images, more than 3 times the image total of our dataset, but ABOships provides more annotations than the former, with a greater average number of annotations per image (4.2 vs. 1.2). SeaShips consists mostly of images with one annotation per image. MCShips provides a number of 13 ship categories (vs. 9 ship categories in ABOships), but only offers just over 26K annotations, with an average of 1.8 annotations per image, see Table 3. We note that our dataset annotations comprise also seamarks and miscellaneous floaters in addition to the 9 ship types.

We tested our relabelling software application on the Singapore Maritime Dataset, as suggested by our reviewers, and the tracker was able to consistently map object labels from one frame to another correctly (without drifting from the object of interest to other objects), which did not always occur when we performed the tracking on the ABOships dataset. There are a few aspects that can influence the tracker's performance and those most probably affected its performance on the ABOships dataset. First, the videos included in the Singapore Maritime dataset have a higher frame rate (30 FPS), double than those in our dataset (15 FPS). Moreover, the videos from the Onshore dataset (one part of the Singapore Maritime Dataset) have higher resolution. Videos in the Onshore dataset do not have a high density of annotations per video. Furthermore, the environment present in the images of our dataset is far more complex, including urban landscapes and complicated background, especially in the port area.

7. Conclusions

This paper provides a solution for addressing the annotation inconsistencies appeared as a consequence of manual labeling of images, using the CSRT tracker [20]. We build traces of the images in the videos they originated from and use the CSRT tracker to traverse these videos in both directions and identify the possible inconsistencies. After this step, we employed a second round of labeling and obtained a set of 41,967 carefully annotated objects, of which 9 types of maritime vessels (boat, miscboat, cargoship, passengership, militaryship, motorboat, ferry, cruiseship, sailboat), miscellaneous floaters and seamarks.

We ensured the dataset consists of images taking into account the following factors: background variation, atmospheric conditions, illumination, visible proportion, occlusion and scale variation. We performed a comparison of the out-of-the-box performances of four state-of-the-art CNN-based detectors (Faster R-CNN [17], R-FCN [23], SSD [18] and EfficientDet [22]). These detectors were previously trained on the Microsoft COCO dataset. We assess the performance of these detectors based on feature extractor and object size. Our experiments show that Faster R-CNN with Inception-Resnet v2 outperforms the other algorithms for objects with an occupied pixel area $> 16^2$ pixels, except in the large object category where EfficientDet registers the best performance with an $AP = 55.48\%$.

For future research, we plan to investigate different types of errors in the manual labelling, for cases where the labels still have inconsistencies, such as: fine-grained recognition (which renders it more difficult for human even to detect objects even when they are in plain view [25], class unawareness (some annotators become unaware of certain classes as ground truth options) and insufficient training data (not enough training data for the annotators).

Moreover, we plan to investigate in more detail the detection of small and very small objects, including those with an occupied pixel area $< 16^2$ pixels. Furthermore, distinguishing between different vessel types in our datasets will be an essential focus as the next steps in our experiments. In order to do this, we plan to exploit transfer learning both in the form of heterogeneous transfer learning, but also homogeneous domain adaptation.

To further our research, we will employ maritime vessel tracking detectors on the original videos captured in the Finnish Archipelago and examine the impact on autonomous navigation and navigational safety.

Author Contributions: V.S. and J.L. planned video capture and collection in the Finnish Archipelago. B.I. and V.S. planned the annotation process and wrote the annotation requirements. L.Z. and V.S. supervised and participated in the annotation process. V.S. implemented the relabelling algorithm. B.I. planned the experiments on the relabelled dataset and supervised their implementation. L.Z. wrote the software for the evaluation of the algorithms on the datasets. V.S. wrote the software for

AP calculations. All authors contributed to the interpretation of results. B.I. wrote the following sections and subsections: Introduction, Conclusion, Experimental Results, Dataset Statistics, Dataset Diversity, Annotation. L.Z. and V.S. wrote the Related Work section. Valentin Soloviev wrote the following subsections in Materials and Methods: Dataset Design, Relabelling Algorithm. The annotation subsection was written by V.S. and B.I. B.I. planned the manuscript writing, and revised the final writing of each section. B.I. and J.L. supervised the evaluation of the algorithms, AP calculation. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to being in the process of publishing, it is planned to be published at: https://www.fairdata.fi/en/ (accessed on 4 February 2021). For reviewers we can provide a separate package with data and any necessary code in the meantime.

Acknowledgments: The annotation of the ABOships dataset was completed with the help of the following persons: Sabina Bäck, Imran Shahid, Joel Sjöberg and Alina Torbunova.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Shao, Z.; Wang, L.; Wang, Z.; Du, W.; Wu, W. Saliency-aware convolution neural network for ship detection in surveillance video. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *30*, 781–794. [CrossRef]
2. Liu, S. Remote Sensing Satellite Image Acquisition Planning: Framework, Methods and Application. Ph.D. Thesis, University of South Carolina, Columbia, SC, USA, 2014.
3. Shao, Z.; Wu, W.; Wang, Z.; Du, W.; Li, C. Seaships: A large-scale precisely annotated dataset for ship detection. *IEEE Trans. Multimed.* **2018**, *20*, 2593–2604. [CrossRef]
4. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
5. Druzhkov, P.; Kustikova, V. A survey of deep learning methods and software tools for image classification and object detection. *Pattern Recognit. Image Anal.* **2016**, *26*, 9–15. [CrossRef]
6. Zhang, J.; Li, W.; Ogunbona, P.; Xu, D. Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective. *ACM Comput. Surv. (CSUR)* **2019**, *52*, 1–38. [CrossRef]
7. Wu, X.; Sahoo, D.; Hoi, S.C. Recent advances in deep learning for object detection. *Neurocomputing* **2020**, *396*, 39–64. [CrossRef]
8. Liu, L.; Ouyang, W.; Wang, X.; Fieguth, P.; Chen, J.; Liu, X.; Pietikäinen, M. Deep learning for generic object detection: A survey. *Int. J. Comput. Vis.* **2020**, *128*, 261–318. [CrossRef]
9. Pathak, A.R.; Pandey, M.; Rautaray, S. Application of deep learning for object detection. *Procedia Comput. Sci.* **2018**, *132*, 1706–1717. [CrossRef]
10. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*; Springer: Zurich, Switzerland, 2014; pp. 740–755.
11. Miller, G.A. WordNet: a lexical database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
12. Kanjir, U.; Greidanus, H.; Oštir, K. Vessel detection and classification from spaceborne optical images: A literature survey. *Remote. Sens. Environ.* **2018**, *207*, 1–26. [CrossRef]
13. Arshad, N.; Moon, K.S.; Kim, J.N. Multiple ship detection and tracking using background registration and morphological operations. In *Signal Processing and Multimedia*; Springer: Jeju Island, Korea, 2010; pp. 121–126.
14. Wijnhoven, R.; van Rens, K.; Jaspers, E.; de With, P.H.N. Online learning for ship detection in maritime surveillance. In Proceedings of the 31th Symposium on Information Theory in the Benelux, Rotterdam, The Netherlands, 11–12 May 2010; pp. 73–80.
15. Lee, S.J.; R., M.I.; Lee, H.W.; Ha, J.S.; Woo, I.G. Image-Based Ship Detection and Classification for Unmanned Surface Vehicle Using Real-Time Object Detection Neural Networks. In Proceedings of the 28th International Ocean and Polar Engineering Conference, International Society of Offshore and Polar Engineers, Sapporo, Hokkaido, Japan, 10–15 June 2018.
16. Redmon, J.; Farhadi, A. YOLO9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 7263–7271.
17. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; ACM: Montreal, QC, Canada, 2015; pp. 91–99.
18. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A. Ssd: Single shot multibox detector. In *European Conference on Computer Vision*; Springer: Amsterdam, The Netherlands, 2016; pp. 21–37.
19. Zheng, Y.; Zhang, S. Mcships: A Large-Scale Ship Dataset for Detection Furthermore, Fine-Grained Categorization in the Wild. In Proceedings of the 2020 IEEE International Conference on Multimedia and Expo (ICME), London, UK, 6–10 July 2020; pp. 1–6.
20. Lukezic, A.; Vojir, T.; Cehovin Zajc, L.; Matas, J.; Kristan, M. Discriminative correlation filter with channel and spatial reliability. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 6309–6318.

21. Padilla, R.; Netto, S.L.; da Silva, E.A. A survey on performance metrics for object-detection algorithms. In Proceedings of the 2020 International Conference on Systems, Signals and Image Processing (IWSSIP), Niteroi, Brazil, 1–3 July 2020; pp. 237–242.
22. Tan, M.; Pang, R.; Le, Q.V. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2020; pp. 10781–10790.
23. Dai, J.; Li, Y.; He, K.; Sun, J. R-fcn: Object detection via region-based fully convolutional networks. In *Advances in Neural Information Processing Systems*; ACM: New York, NY, USA, 2016; pp. 379–387.
24. TensorFlow Object Detection API. Available online: https://github.com/tensorflow/models/tree/master/research/object_detection (accessed on 4 February 2021).
25. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; et al. Imagenet large scale visual recognition challenge. *Int. J. Comput. Vis.* **2015**, *115*, 211–252. [CrossRef]